Reviewing and Preserving Experimental Software Toolkits through SHARE

Pieter Van Gorp

School of Industrial Engineering, Eindhoven University of Technology (TU/e),
De Lismortel 2, 5612AR Eindhoven, The Netherlands

Abstract

The WASDeTT workshop series promotes academic tool development by providing a discussion forum as well as a journal special issue dedicated to Experimental Software Toolkits (ESTs). WASDeTT papers should discuss software tool requirements, tool design rationale and tool design reflections. Papers in the EST issue and their associated tools should additionally satisfy strict requirements regarding the quality of the source code and documentation, the convenience of the installation scripts and finally the applicability and relevance to the intended domain. We argue that the maturity of EST sources, installation scripts and manuals are academically less relevant than the relevance and usability of an EST. Moreover, we have observed that it is often unrealistic to provide and maintain EST sources, scripts and documentation that are compatible with the execution infrastructure of a large target audience. We therefore invite the WASDeTT authors, reviewers and organizers to adopt a more light-weight approach to reviewing and preserving ESTs. The approach consists of installing the ESTs and their documentation in shareable online Virtual Machines (VMs). VMs do not pose restrictions with regards to APIs or programming languages. Moreover, they can be preserved more easily than more fine-grained artifacts such as Eclipse plug-ins. The SHARE system provides special purpose online workflow support to facilitate the management of VMs in this context.

Keywords: Academic Tool Development, Open Access, Software Engineering

1. Open Access Trends

Why hasn't the software engineering community yet adopted more advanced engineering tools at large? Perhaps the reason is that many software vendors (and academic authors) have been too optimistic on paper about the benefits of their tools. False expectations may have led to frustrations among practitioners

Email address: `p.m.e.v.gorp@tue.nl` (Pieter Van Gorp)
URL: `http://is.ieis.tue.nl/staff/pvgorp/` (Pieter Van Gorp)

who discovered that academic tools are often not that effective as claimed in their corresponding research papers. Over time, prospective users of academic tools may even lose faith in the reliability of papers about such tools. As an escape from this negative spiral, there are various initiatives where the wheat can be separated from the chaff. The EST special issue related to WASDeTT for example requires authors to publish not only the claims about their tools but also evidence to back up the claims that are made. A similar example is the Transformation Tool Contest (TTC), which was spun off from the Graph Based Tools (GraBaTs) workshop in 2008, when the related community felt a need to go beyond reading and writing papers.

WASDeTT and TTC both stimulate better access to research tools (i.e., to academic software). Open access to research software is a special case of open access to research data. Open access is receiving much attention from scientific publishers, from policy makers, and even from the popular press. One of the striking motivators is fraud prevention. At the time of writing, a story related to the National Fund for Scientific research (FWO) is in the Belgian news headlines[1]: all FWO-funded researchers will have to make their raw research data available to their university. While this indeed may discourage fraud related to surveys and lab measurements for example in life sciences, it may have little impact in software engineering. Software tools are such complex digital artifacts that universities will not have the manpower to use raw binaries or sources for verifying systematically the validity of experimental software engineering results.

This issue is addressed by the EST special issue editors by requiring that paper submissions are complemented by elaborate installation documentation and scripts. In this paper, we argue that such requirements are valid only in rare cases while the systematic requirement of using SHARE for open access to ESTs does make sense. SHARE open access also discourages fraud and facilitates knowledge transfer with little effort from authors, reviewers and editors.

The remainder of this paper is structured as follows: Section 2 provides an introduction to SHARE, Section 3 describes the historic load of the system, Section 4 describes how WaSDeTT stakeholders can start practically using it, Section 5 discusses related platforms and Section 6 presents open issues.

2. Sharing Hosted Autonomous Research Environments (SHARE)

The SHARE project aims to build an online library of virtual machines (VMs) related to scientific publications (papers, dissertations, project reports)[2]. The purpose of these virtual machines is to “share” securely and conveniently all data, software and configuration parameters related to scientific experiments.

The online library is supported by specialized infrastructure: a web portal provides its users a convenient and secure means to “share” a virtual copy of

[1] http://www.standaard.be/artikel/detail.aspx?artikelid=DMF20130422_00551279

[2] is.ieis.tue.nl/staff/pvgorp/share/

their own research computer without having to make their code or executables available for download. Moreover, the portal makes it convenient for reviewers to verify the reported results: they can launch right from the browser one or more remote virtual machines that may connect to large datasets and complex software environments behind the scenes. The remote virtual machines also support the secure execution of licensed software and ensure that such software is not used improperly for industrial purposes. This has already enabled various researchers to use each other's data and software without download, installation, configuration, or licensing barriers. Architecturally, the portal and its hosted virtual machines are designed to remain available for future generations of researchers from a wide range of disciplines.

A systematic overview of the TTC-based use cases and non-functional requirements that have driven the SHARE development has been published in 2010 [6]. Amongst others, these requirements included the free and secure creation, sharing and online publication of remote demonstration virtual machines. The use of virtual machines was originally driven by the fact that TTC involved complex artifacts such as interactive visualization components, desktop-based debugging user interfaces, etc. It was unrealistic to expect from TTC authors that they would make these artifacts executable on any type of client machine. Similarly, it was unrealistic to expect from TTC reviewers that they would follow detailed installation/configuration instructions and solve incompatibility issues that were unforeseen by the authors. Before introducing SHARE, the TTC organizers observed that results were therefore rarely reproduced.

The 2010 publication clarifies that in comparison to platforms such as Eclipse, jETI and Sourceforge, SHARE was unique in its holistic and light-weight online deployment approach. In comparison to more generic platforms for virtual machine management (e.g., Eucalyptus or Amazon EC2), SHARE is unique in its default use of stateless, Internet connectionless, virtual machine sessions, which is important in the context of VM preservation and license management [6]. To the best of our knowledge, that related work analysis is still up-to-date.

The relation between SHARE and external requirements from a Grand Challenge in the Computational Sciences community has been published in 2011 [2]. Amongst others, this Grand Challenge involved new requirements such as provenance and integration in a publishing workflow.

If the project manages to attract funding for professional support, it will enable the systematic preservation of the complete working environments from large scale research projects. Thus, these objects would no longer get lost on the volatile machines of grant holders. So far, we have had two very positive evaluations of SHARE proposals but eventually were not ranked high enough for national funding in The Netherlands. Via this paper and the intended workshop discussions, we aim to increase the visibility of the project and demonstrate the effectiveness of SHARE in new application domains, regardless of funding.

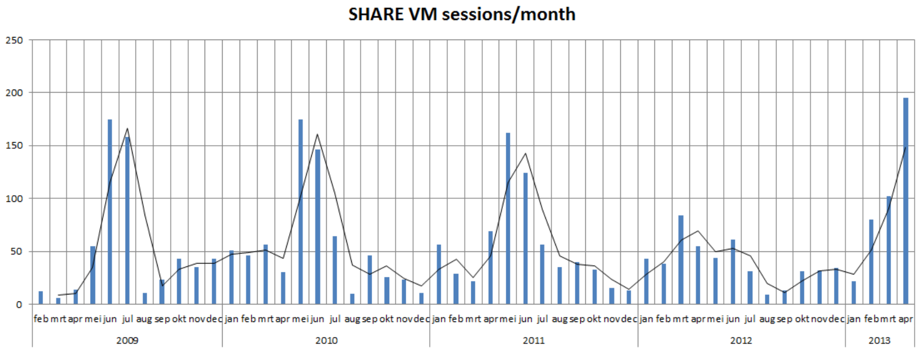

Figure 1: Number of VM sessions per month, excluding sessions from the author of this paper.

3. Adoption of SHARE

SHARE has been built to solve organisational problems of early editions of the Transformation Tool Contest (TTC), a workshop affiliated with the ECMFA/ICGT/ICMT/MoDELS communities. The system has been used successfully for various TTC editions and other events. Also, it has served as a local bridge between the research and teaching at TU/e.

Figure 1 shows how the number of virtual machine sessions per month has evolved since 2009. The trendline is based on a 2 month moving average and clearly illustrates that there has been one peak every year so far. This peak corresponds to the TTC 2009, 2010 and 2011 editions as well as a related journal special issue in 2012. The activity for TTC 2013 is ongoing[3]. The trendline also shows that during other periods of the year, about 25 VM sessions are used per month. These sessions are primarily from about 2-3 master and PhD students per year who use SHARE both for the convenient evaluation of academic tools and for the publication of their thesis results.

4. Using SHARE for WASDeTT, EST and more

SHARE VMs are organized in groups. Each group has a moderator and each image has an owner. Academic event organizers, PhD students, and others can apply online for becoming the moderator of a new group[4]. Authors who wish to make their results available via SHARE should request a clone of an existing virtual machine image. Typically, new users start from an image in the SHARE *Operating Systems* group[5]. For each clone request, the owner of the

[3] `http://planet-sl.org/ttc2013/index.php?option=com_content&view=article&id=231:important-dates`

[4] `http://is.ieis.tue.nl/staff/pvgorp/share/?page=RequestNewBundle`

[5] `http://is.ieis.tue.nl/staff/pvgorp/share/?action=clone&page=LookupImage&bNameSearch=Operating+Systems`

base image as well as the group moderator will automatically be asked to give their approval.

If, in the context of WaSDeTT for example, one would like to make a new SHARE VM based on Ubuntu 12, then one would select the corresponding image from a drop down menu[6] and one person (who is the moderator of the *Operating Systems* group but also the owner of all enclosed VMs) would give his approval. In parallel, the WASDeTT organizers or an individual author could apply for becoming the moderator of a new group. A special administator can handle such requests with just one click.

Once the author would have finished preparing the new virtual machine (e.g., after uploading data, installing software and placing a basic tutorial on the VM desktop), (s)he would publish the VM via the SHARE portal. The system then stores the VM image in the library. From that moment in time, authorized users will be able to run independent and stateless sessions of the new VM right from their browser. *Independence* refers here to the fact that different users do not see each other's VM state. *Statelessness* on the other hand refers to the fact that subsequent sessions by the same user will always start from the VM state that was prepared by the author.

The browser access to remote virtual machines is supported by a generic Java applet that implements the Remote Desktop Protocal (RDP) specifications. Consequently, authors do not have to write any remote access related code. By default, it is impossible to download content from VM sessions. This supports scenarios where reviewers only get temporary access to software and data that is worth writing a paper about, yet too sensitive for download access.

We invite all authors that submit a paper to the 2013+ editions of the EST special issue to also deploy their artifacts to SHARE. These authors will have prepared already elaborate end-user documentation, installation scripts and example applications. The effort of these EST experts to deploy their artifacts to a VM should be significantly lower than that of special issue readers that have to deploy foreign software. The authors will also benefit from SHARE's preservation service, assuming that the project attracts funds for continued operation. Finally, we also invite the special issue editors to allow submissions from authors that cannot satisfy the traditional requirements for submitting EST artifacts yet that can provide a SHARE demo instead. This may involve ESTs that are closed source as well as ESTs for which robust installation scripts are not (yet) a priority.

5. Related Work

SHARE has won the second prize in the aforementioned Executable Paper challenge, which was organized by Elsevier as a satellite event to the Interna-

[6] `http://is.ieis.tue.nl/staff/pvgorp/share/?page=RequestMutableVirtualDiskImage&vdi=ubuntu_12.04.vdi`

tional Conference on Computational Science (ICCS 2011)[7]. This has been an excellent opportunity to compare SHARE against a variety of related platforms beyond the software engineering community.

The first prize of the Elsevier challenge was awarded to the *Collage* Authoring environment [4]. That approach is based on an online text editor where authors can use specialized widgets for inserting interactive parts into their online papers. The widgets enable calling external web services from the compiled online paper. SHARE does not offer a paper authoring environment. Also, its VM session viewers are not directly embedded in the online version of a paper. Instead, SHARE based papers cite hyperlinks for starting VM sessions. SHARE does include a Java-based session viewer such that readers can also conveniently experiment with software as they read an electronic paper. Also, while *Collage* requires the development of domain specific visualization widgets, SHARE provides online access to any[8] desktop tool without additional programming effort.

Another team of challenge finalists also proposed the use of virtual machines [1]: while their supportive platform did not offer SHARE features such as online VM cloning and online, stateless VM executions, the authors did report on various benefits due to the VM paradigm. Among others, they stressed the advantage of conveniently sharing and preserving *holistic* research environments (i.e., not just source code and documentation, but also contemporary library versions, tool settings, etc.).

Other challenge solutions also demonstrated features which could improve SHARE. For example, SHARE could be improved by automatically issuing a DOI for each of the VMs in its library. For those features, we try to acquire funding in cooperation with the academic data center which already has data archiving services and related DOI services.

Finally, a variety of domain-specific solutions have also participated in the challenge. For example, the R^2 Platform was presented as a successful approach from the statistical community [3]. This specific platform is domain-specific since it assumes the use of the statistical scripting language R. SHARE is by no means domain specific since any API or programming language can be used within its virtual machines. This implies that also R can be used from inside SHARE VMs. This flexibility comes at the cost of a lower level of integration with the related paper. More specifically, while a SHARE based VM is dynamic, the research paper that cites it remains static. In contrast, just like *Collage*-based papers, R^2-based papers have embedded panels for dynamic (i.e., reader-specific) output rendering.

[7] `http://www.executablepapers.com/`

[8] SHARE relies on Oracle VirtualBox for virtualizing desktop environments. A list of supported operating systems is available at `https://www.virtualbox.org/wiki/Guest_OSes`.

6. Open Issues

By default, SHARE VMs are deprived of Internet access. In particular, by default no content can be downloaded from a SHARE VM. SHARE does enable any user to upload content to a VM through a mechanism that still does not require VMs to have Internet access [6, 2]. We can also implement an extension to SHARE that would enable VM authors to put public files in a specialized VM network directory. Files in such directories could then be made automatically accessible for public download via the SHARE portal. Note that authors could then also decide to put only specific files inside these network directories (e.g., only end-user documentation and screencast movies). During WaSDeTT 2013, we will discuss the relevance of such a SHARE extension and solicit also additional requirements.

SHARE provides no built-in content screening features. Instead, the platform leaves content screening to organizers and reviewers. In the context of TTC 2013, we have experimented with a preliminary form of content screening: for that event, authors had to demonstrate that their model transformation artifacts satisfied the constraints of a specific testsuite. Instead of hard-coding the related checks inside the SHARE platform, we decided to deploy the testsuite and test runner inside a VM and solution authors then cloned that base VM. Reviewers were then expected to check whether a submitted VM for a model transformation solution indeed produced correct test behavior. In this approach, no technical measures were taken to prevent fraud but one could explore code signing techniques as one way to prevent fraud in this setting.

SHARE also does not provide built-in functionality for documenting how a VM should be used. Instead, the platform does enable authors to upload among others some screencast movies that were recorded with external software. In practice, our preferred VM documentation method is the use of a companion website that links to the SHARE VM as well as to online screencasts. However, as illustrated by `https://sites.google.com/site/executablepaper/model-transformation-example`, also screencast links on such companion websites may become invalid. Ultimately, this is an argument for consistently deploying all VM documentation inside the VM but we consider the usability of the associated uploading step an open point for improvement.

Finally, SHARE does not yet provide features for the semantic reasoning about VM content. In the context of a tutorial that is co-located with WASDeTT 2013[9], we are presenting ongoing efforts to address this limitation.

Conclusions

This paper calls the experimental software engineering community to action: academic participants to WASDeTT and similar events should make their results available via SHARE. This will save them valuable time compared to the

[9] `http://info-web.lirmm.fr/ec-montpellier-2013/?id=158#tutorial7`

situation where they have to build installation tools that are both user-friendly and future proof. We acknowledge that for mature software suites, the availability of such artifacts is important too but for most research vehicles the effort does not outweigh the benefits.

References

[1] Grant R. Brammer, Ralph W. Crosby, Suzanne J. Matthews, and Tiffani L. Williams. Paper mâché: Creating dynamic reproducible science. In Sato et al. [5], pages 658 – 667. doi: 10.1016/j.procs.2011.04.069.

[2] Pieter Van Gorp and Steffen Mazanek. SHARE: a web portal for creating and sharing executable research papers. In Sato et al. [5], pages 589 – 597. doi: 10.1016/j.procs.2011.04.062.

[3] Friedrich Leisch, Manuel J. A. Eugster, and Torsten Hothorn. Executable Papers for the R Community: The R2 Platform for Reproducible Research. In Sato et al. [5], pages 618–626. doi: 10.1016/j.procs.2011.04.065.

[4] Piotr Nowakowski, Eryk Ciepiela, Daniel Harezlak, Joanna Kocot, Marek Kasztelnik, Tomasz Barty'nski, Jan Meizner, Grzegorz Dyk, and Maciej Malawski. The collage authoring environment. In Sato et al. [5], pages 608–617. doi: 10.1016/j.procs.2011.04.064.

[5] Mitsuhisa Sato, Satoshi Matsuoka, Peter M. A. Sloot, G. Dick van Albada, and Jack Dongarra, editors. *Proceedings of the International Conference on Computational Science, ICCS 2011, Nanyang Technological University, Singapore, 1-3 June, 2011*, volume 4 of *Procedia Computer Science*, 2011. Elsevier.

[6] Pieter Van Gorp and Paul Grefen. Supporting the internet-based evaluation of research software with cloud infrastructure. *Software & Systems Modeling*, 11(1):11–28, 2012. ISSN 1619-1366. doi: 10.1007/s10270-010-0163-y.