Demo: Composing, Reproducing, and Sharing Simulations

Debashis Ganguly, William C. Garrison III, David Wilkinson,
Bruce R. Childers, Adam J. Lee, and Daniel Mosse
Department of Computer Science
University of Pittsburgh
Pittsburgh, Pennsylvania USA

Every year, research groups around the world contribute papers and artifacts to the computer science literature. In many areas, simulation and modeling play key roles in bringing about these new contributions. Simulation is used to test and validate new ideas prior to their implementation, and thus, the artifacts (software, data sets, benchmarks, etc.) used in simulation are fundamental to the empirical evaluation of a research hypothesis.

Often, the primary focus of a paper is on the validation of a central hypothesis or building proof-of-concept software, and the details surrounding the artifacts used during and experiments used for this process are often scarce. Many researchers do not intend to build a foolproof software component to share with the community. Artifacts may end up limited in scope or usability, and hidden or unstated assumptions may make the artifact difficult (if not impossible) to reuse, extend, or compose. Many artifacts take a tremendous amount of effort to build and validate and, as such, may remain private to the research groups that invested in developing them in the first place. This limits their availability, increases the difficulty of validating claims made in papers based on these artifacts, and limits the ability of others to build upon prior work.

Addressing this situation necessitates sharing and reproducibility[1]. While this problem cuts across most CS disciplines, the modeling and simulation community has a unique advantage in addressing it. Namely, modeling and simulation rely on well-defined artifacts to carry out some activity; a model, simulation component, initial conditions, input stimuli, etc., must be specified and encapsulated in some form as part of an evaluation. To this end, our participation at WSSSPE 2016 will concretely demonstrate our approach to sharing, reproducing, and composing simulations toward accelerating research productivity while also improving accountability and credibility. Specifically, we have developed a case study in which we compose and share access control simulations in the form of shareable data store units for cloud systems. This case study is openly hosted in the OCCAM collaborative repository (http://occam.cs.pitt.edu) and integrated with Sandia's Structural Simulation Toolkit (http://sst-simulator.org).

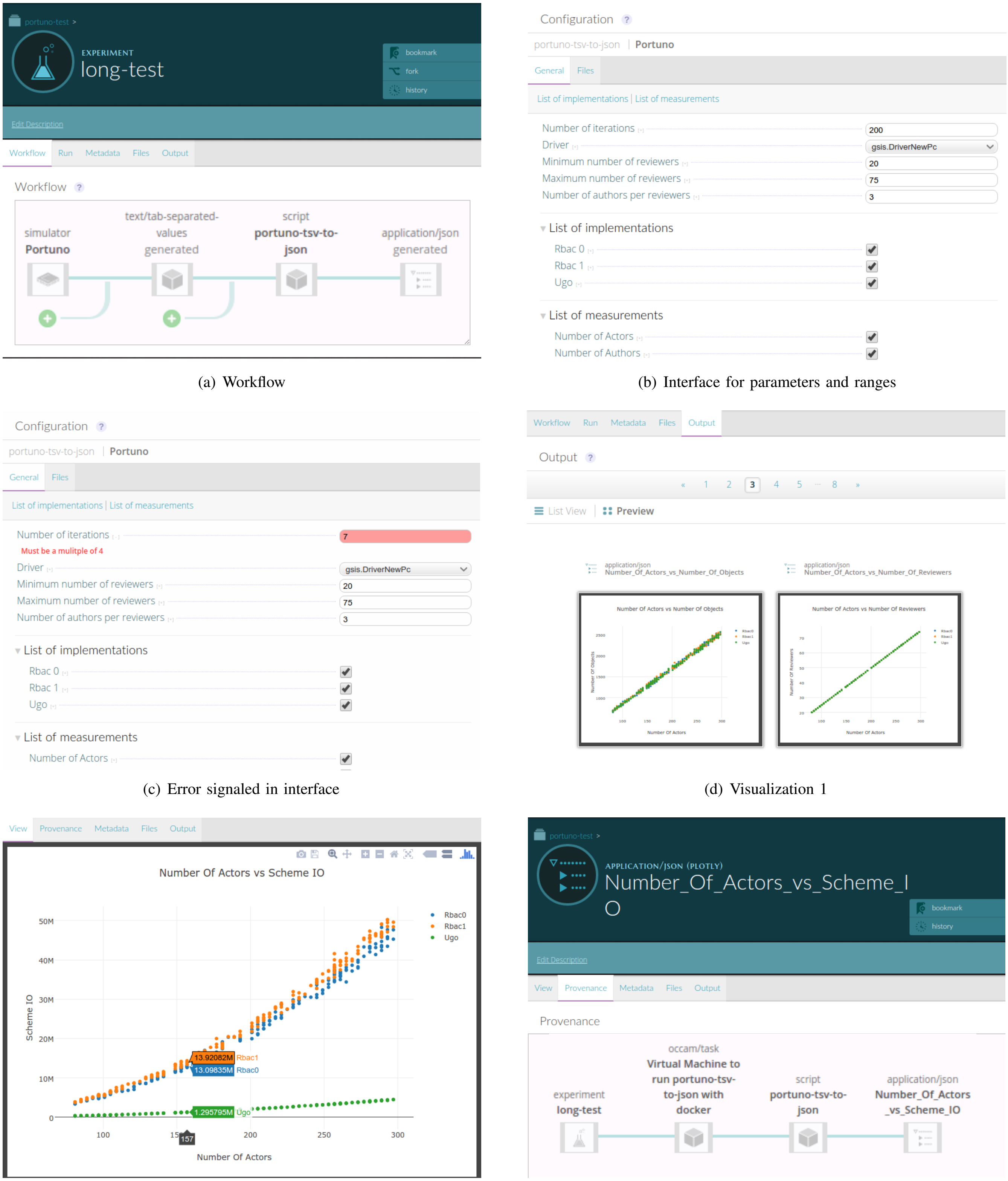
[1]Note that *reproducibility* in this paper is defined as experimentation that is both repeatable *and* modifiable.

Our security simulator, Portuno, conducts cost analyses to explore the suitability of different access control approaches for a given application workload. Portuno has been used in an array of analyses, including evaluating group-centric approaches to information sharing and exploring the communication, computation, and administrative overheads associated with cryptographic enforcement of role-based access controls (RBAC) on untrusted cloud platforms. Portuno uses probabilistic actor-based models of user, administrator, and system behaviors to generate application traces (e.g., open file, edit file, share file, modify permissions, etc). These traces are then mapped into traces in concrete access control systems: those that are candidates for implementing the application. Costs are then aggregated over these candidate system traces. Portuno supports a wide range of design choices in its actor models, initial system states, and other parameters of an experiment. As such, openly sharing the choices that have been made and allowing other researchers to modify these choices can lead to a better understanding of the trade-offs among different access controls techniques. Figure 1(a) shows the workflow of the composition of Portuno into OCCAM.

To compose Portuno with other simulations, share the resulting infrastructure, and disseminate the experimental outcomes, Portuno is integrated with SST and incorporated in OCCAM. SST acts as the driver of the underlying access control models, which are implemented in Java. This is a novel use of SST as a backbone for probabilistic modeling in an area other than computer architecture and computer systems simulation. It also illustrates interoperability between SST and Java models.

The combination of OCCAM, SST, and Portuno leads to a seamless environment that is more capable than the sum of its parts. This integrated approach offers the capability to quickly define, run, visualize, and share simulation artifacts and results over a huge design space. It supports an end-to-end workflow for modeling and analyzing access controls under a variety of scenarios, making it easier to (a) use Portuno for access control analysis, (b) inspect and augment Portuno experiments done by others, and (c) modify Portuno in a contained environment.

OCCAM allows for a dynamic environment where a researcher can explore ranges of inputs and simulation results by (a) specifying ranges and having the system automatically generate, organize, and tag output results (see Figure 1(b) for a sample of parameters, ranges, and web interface); note



(a) Workflow

(b) Interface for parameters and ranges

(c) Error signaled in interface

(d) Visualization 1

(e) Visualization 2

(f) Provenance

Fig. 1. Web interface of OCCAM

that Figure 1(c) shows what happens when a parameter is specified incorrectly when a research is able to specify a range (e.g., number of iterations 4-20 while being a multiple of 4, OCCAM generates automatically 4, 8, 12, 16, and 20), (b) visualizing the results of already executed simulations (see Figures 1(d) and 1(e) for a sample of automatically generated visualization of results, which can be manipulated dynamically through our web interface), and (c) requesting the system to extend the simulation runs for different input ranges. In effect, users of OCCAM can be researchers, developers, experimentalists, or curious users.

Traditional digital archives for publishing experimentation, such as Open Science Framework and Dataverse, focus on sharing data without directly enabling reproducibility. Some archives, such as MyExperiment, specialize their systems further by holding the source code and other offered information, and introducing a means of visualizing the experimentation through a workflow of that experiment. In contrast, OCCAM goes several steps forward by retaining all of the data and code necessary for an experiment to be executed/run/reproduced, in addition to also giving a consistent means of visualizing the workflow of the experiment, deploying it, and viewing the results. In the case of OCCAM, the workflow can be composed of several simulators and not only of steps of a simulation. Simulation results can be viewed and manipulated in a dynamic and interactive analyses, representing the "article of the future". Papers currently and recently have been disseminated as PDFs with limited space, fixed content (e.g., a specific set of results/graphs), and inadequate or incomplete details (e.g., missing setup, limited sweeps, etc.). With OCCAM, the results will be integrated in the articles, which will be enhanced to provide greater transparency, actual reproducibility, and complete provenance of the results. For example, a reader can click on a graph, and is taken to the digital library repository of the data used to produce the graph, including the simulator, the input data, the configuration files for the simulator, etc. See Figure 1(f) for an example of the output of the provenance. In addition, the reader will be able to extend a graph beyond what is shown on the paper, to see trends and other further results the reader wants to see, not only the (fixed) extended results on a website provided by the authors.

Our seamless environment also enables the novel composition of simulators. In particular, we can combine Portuno with other simulations. For example, we are currently investigating how Microns hybrid memory cube (HMC) can decrease the overheads associated with enforcing cryptographic access controls in cloud environments. Recent simulations by our team show that the administrative costs involved in altering cryptographically enforced RBAC policies are prohibitive: e.g., revoking a user from a single role may require thousands of re-encryptions in even a moderately-sized organization. The use of HMCs, perhaps combined with trusted execution environments like Intel's SGX, would allow users to push the re-encryption to the data, rather than bringing bulk data to the processor to re-encrypt. The administrative action traces generated by various Portuno configurations would serve as good candidate inputs for HMC simulators that could help us explore the potential benefits of this architectural enhancement to speed up the management of files on untrusted infrastructure.

At the workshop, we will show how sharing, composing and repeating simulations through a collaborative repository (OCCAM) and a general simulation framework (SST) can accelerate our efforts as a community. Using our work on access controls as a case study, we will explain our technical approach, how our integrated environment facilitates design exploration, and the potential of composing separate models. In the spirit of this abstract, interactive results obtained from Portuno, SST, and OCCAM are available at http://tinyurl.com/hj2oewn.

ACKNOWLEDGEMENTS

The material in this document is based in part upon work supported by the National Science Foundation (NSF) under grant numbers ACI-1535232, CNS-1305220, and CNS-1228697. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.