



A Provenance-Based Infrastructure for Creating Executable Papers (Abstract)

David Koop^a, Emanuele Santos^a, Phillip Mates^a, Huy T. Vo^a, Philippe Bonnet^b, Bela Bauer^c, Brigitte Surer^c,
Matthias Troyer^c, Dean N. Williams^d, Joel E. Tohline^e, Juliana Freire^a, Cláudio T. Silva^a

^aUniversity of Utah

^bIT University of Copenhagen

^cETH Zürich

^dLawrence Livermore National Laboratory

^eLouisiana State University

1. Introduction

While computational experiments have become an integral part of the scientific method, it is still a challenge to repeat such experiments, because often, computational experiments require specific hardware, non-trivial software installation, and complex manipulations to obtain results. In this paper, we posit that integrating data acquisition, derivation, analysis, and visualization as executable components throughout the publication process will make it easier to generate and share repeatable results. We describe the infrastructure we have built to support the lifecycle of such executable papers.

A number of tools have been developed that attack sub-problems related to the creation of executable papers. Besides the lack of an end-to-end solution, existing approaches are often limited. For example, Mesirov described a Windows-specific mechanism for connecting Word documents to GenePattern pipelines [1]. VisTrails [2] provides a multi-platform approach which allows the creation of wiki pages as well as LaTeX, Word, and PowerPoint documents, where each result has a deep caption linked to its provenance. This provenance includes the workflow used to derive the result, but this link is only one piece of an executable paper. For example, a reviewer should be able to assess the correctness and relevance of experimental results described in a submitted paper. Furthermore, ideally, upon publication, readers should be able to repeat and utilize the computations embedded in the papers.

Our focus is on designing an infrastructure that caters to a wide range of requirements from a variety of scientific disciplines. It should meet the following goals: *a lower barrier for adoption* to help authors write and assemble their submissions; *flexibility* to allow authors a choice of mechanisms and systems to package their work; and *support for the reviewing process* to provide reviewers with infrastructure to unpack, reproduce, and validate the submissions.

The infrastructure we propose is centered around VisTrails, a provenance-enabled, workflow-based data exploration tool. For the last three years, we have extended it to combine the natural benefits of a provenance infrastructure—systematic capture of useful metadata, including workflow provenance, source code, and library versions—with tools that address different aspects of the executable paper problem. These components include mechanisms to link results to their provenance, reproduce results, explore parameter spaces, interact with results via a Web-based interface, and upgrade computational experiments to use new versions of software. We note that our notion of executable paper is orthogonal to others which focus on semantics and authoring, and our infrastructure can be combined with these.

In the full version of this paper, we will present the stages of a paper's development, the challenges involved in each, and an outline of the solutions we adopted in our infrastructure. In addition, we will detail use cases and discuss both lessons learned and open issues. In the remainder of this abstract, we sketch our design and briefly discuss two case studies that demonstrate different uses of our infrastructure. We invite the judges to consider a video that illustrates some features of our infrastructure in action and a position paper that details the challenges of computational repeatability and the solutions we have developed. Both the videos and paper can be found at <http://www.vistrails.org/index.php/ExecutablePapers>.

2. Infrastructure Overview

Our infrastructure is designed to address challenges throughout the stages of a paper’s development, from writing to reviewing to publishing.

2.1. Writing & Development

An executable paper begins with its author. But often, ideas and results have been generated before the writing begins. Thus, an author benefits from doing work in an environment that simplifies the creation of an executable paper. Provenance is a critical ingredient for such work [3, 4, 5].

Specifying Computations. Because analyses can be conducted using domain-specific tools, command-line scripts, or workflow systems, we need a general architecture for specifying computations. The VisTrails system is written in Python, a widely-used “glue” language, and allows users to create workflows that integrate existing tools and libraries with a simple wrapping system.

Provenance Capture. Common problems in reproducing a result include steps omitted from published code and parameter choices that are not specified. We use VisTrails to automatically and unobtrusively capture a spectrum of provenance, from workflow evolution to data lineage [6, 7]. The *persistence package* supports data provenance by creating strong links which identify data by its content (via hashing), its use (via the workflow that generated it), and its history (via a version control system) [8]. This ensures that an author can always retrieve data used in previous work, even if the original file has been changed or even removed. Besides, it permits efficient re-use of data that has already been generated, for example, as the result of long-running computations.

Result Integration. To support including reproducible results in papers, we have *developed code and plug-ins for LaTeX, MediaWiki, Microsoft Word and PowerPoint*. This allows authors to easily embed and regenerate results when creating their executable paper, and readers to link back to and explore the actual computations.

Execution Infrastructure. While VisTrails is cross-platform, the code, libraries, and other dependencies underlying a workflow are not necessarily cross-platform. To address this problem, we support remote, server-based execution and employ virtual machines to mimic a specific environment, but we also encourage authors to include special modules that check pre- and post-conditions as part of the workflow.

2.2. Review, Validation, & Interaction

An executable paper has the potential to improve the quality of reviews because reviewers have the ability to explore and validate conclusions. However, it can also present challenges including the need to reproduce and test computations that may have been developed on different hardware or software. In addition, reviewers need infrastructure to help them test different configurations and settings.

Local, Remote, and Mixed Execution. In some settings, results require proprietary data, or special hardware and architectures that are not available for the reviewers. For open systems, it may be possible for authors to grant reviewers access, but for closed systems, it may be necessary to scale the problem to a smaller size or assume certain preconditions. Another solution is to treat results from long-running computations, like those obtained using *high-performance computing resources*, as raw experimental data; the papers will only contain full post-processing information. In our infrastructure, we have developed a VisTrails server that works via XML RPC as well as modules to access remote data through relational database (or proprietary) interfaces.

Testing and Validating Results. Workflows provide an abstract, uniform structure for interacting with results, reducing the need for the reviewer to learn new interfaces for each submission. This allows them to more easily understand and explore the computations in the papers. VisTrails also provides a parameter exploration interface to quickly select ranges of parameters to test. These results can be displayed and compared in an intuitive spreadsheet interface [6].

2.3. Publishing, Maintenance, & Re-Use

After an executable paper is accepted, it is important that the executable nature of the publication is maintained. The format of data and computations used will be important in ensuring the longevity of the paper.

Data Access and Formats. Some papers present results that use proprietary data, and others may use data that is too large to duplicate. In some cases, authors can host their own data and/or provide methods to run experiments locally. Another frustration is that because data can exist in many formats, published data may not be easily integrated in other computations. VisTrails provides data conversion modules and allows users to write conversion methods that can be integrated into workflows.

Maintenance and Longevity. As systems evolve, archiving executables will not suffice; archiving code can also be problematic. VisTrails provides a mechanism to store provenance that includes the version of each module used. If a reader downloads an old paper, VisTrails uses *workflow upgrades* to convert the paper to the reader’s environment [9].

Querying and Re-Using Results. Because executable papers contain more than text and metadata, it is important that the executable elements can also be queried. Such queries allow users to more easily find papers that use a specific dataset or computational technique. VisTrails provides a mechanism for building queries by example [10], and we have also worked to develop search engines that allow users to filter results based on computational structure [11]. Furthermore, authors can choose to provide high-level interfaces to simplify re-use and interaction via VisMashups [12].

3. Case Studies

3.1. The ALPS Project

The ALPS project (Algorithms and Libraries for Physics Simulations) is an open-source initiative for the simulation of large quantum many-body systems [13, 14, 15, 16], and has been used in close to two-hundred research projects over the past six years. One of its goals has been to simplify archival longevity and repeatability of simulations by standardizing input and result file formats.

Motivated both by demands for repeatable simulations and the ETH Zurich research ethics guidelines [17], which require that all steps from input data to final figures needs to be archived and made available upon request, the recent 2.0 release takes an important further step. ALPS 2.0 [16] makes use of the provenance infrastructure provided by VisTrails to support data analysis and exploration, as well as the publication of reproducible results. For small simulations, VisTrails combined with ALPS allows for complete reproducibility of any result in a paper, as we have demonstrated in the ALPS 2.0 paper [16], which is already an “executable paper”. Clicking on a figure activates a “deep caption” that retrieves a workflow and executes the underlying calculation on a user’s machine using the VisTrails and ALPS systems. For large-scale simulations, ALPS 2.0 projects are typically split up into two parts: a time-consuming set of simulations and a faster set of analysis workflows. The simulations are executed on high-performance machines whose results are archived with provenance information (using the VisTrails persistence package); the analyses begin with these results and generate the figures that can be included in executable papers.

The ALPS libraries and applications support full backward compatibility to older versions of the input and output files, thus enabling executable ALPS papers to be run on future versions. In addition, the compatibility of past and current ALPS workflows is supported through the workflow upgrade mechanism in VisTrails.

3.2. The SIGMOD Repeatability Effort

As a step towards the goal of computational repeatability, the ACM SIGMOD conference has offered, since 2008, to verify the experiments published in accepted conference papers. A committee is in charge of reproducing the experiments provided by the authors (repeatability), and exploring changes to experiment parameters (workability). In 2010, 20% of accepted papers received the repeatability label. Proper verification requires that reviewers have access to necessary software and data, and that they can determine the accuracy of the submitted experiments. Without proper infrastructure or guidelines, the task can be frustrating and take considerable time. At the same time, authors cannot be expected to spend significant time and effort porting their experiments to a specific infrastructure.

To aid reviewers, we are encouraging SIGMOD authors to adhere to the following guidelines:¹

- (a) Rely on our provenance-based workflow infrastructure to automate experimental setup and execution tasks.
- (b) Use a virtual machine (VM) as the environment for experiments.
- (c) Explicitly represent pre- and post-conditions for setup and execution tasks.

¹See the Repeatability section of the ACM SIGMOD 2011 home page: http://www.sigmod2011.org/calls_papers_sigmod_research_repeatability.shtml

A common infrastructure guarantees the uniformity of representation across experiments so reviewers need not relearn the experimental setup for each submission. The structure of workflows helps reviewers understand the design of the experiments as well as determine which portions of the code are accessible. While virtual machines ensure the portability of the experiments so reviewers need not worry about system inconsistencies, explicit pre- and post-conditions makes it possible for reviewers to check the correctness of the experiment under the given conditions.

To help authors, we have extended VisTrails to better support *remote* executables in a workflow via XML RPC². This allows authors with proprietary software or data, or even specific hardware to make portions of their experimental framework accessible remotely to reviewers. In addition, if authors use our infrastructure throughout the lifecycle of their papers, satisfying the repeatability criteria should be automatic. Note that our infrastructure can also be used to archive results internally, allowing them to be reproduced later for a journal version of a given paper or after the person who designed and ran the experiment has left the group.

4. Conclusion

To the best of our knowledge, our infrastructure is the first approach that provides an end-to-end solution to the problem of computational repeatability. It relies on a provenance-based workflow system, VisTrails, that we have extended to meet the requirements of an executable paper lifecycle. An important feature of this infrastructure is that it was designed with extensibility in mind; different components can be added and combined to support a wide range of requirements for scientific publications. There are several open problems we are currently investigating including methods for improved testing and debugging, easier management of complex workflows, and better support for remote resources.

- [1] J. Mesirov, Accessible reproducible research, *Science* 327 (5964) (2010) 415–416.
- [2] VisTrails, <http://www.vistrails.org>.
- [3] C. Silva, J. Freire, S. P. Callahan, Provenance for visualizations: Reproducibility and beyond, *IEEE Computing in Science & Engineering* 9 (5) (2007) 82–89.
- [4] J. Freire, E. Santos, C. Silva, Provenance-enabled data exploration and visualization with vistrails, in: *SciDAC*, Vol. 125, 2009.
- [5] C. Silva, J. Freire, E. Santos, E. Anderson, D. Koop, Provenance-enabled data exploration and visualization, in: *IEEE Visualization Conference, 2009*, refereed tutorial.
- [6] J. Freire, C. Silva, S. Callahan, E. Santos, C. Scheidegger, H. Vo, Managing rapidly-evolving scientific workflows, in: *International Provenance and Annotation Workshop (IPAW)*, LNCS 4145, Springer Verlag, 2006, pp. 10–18.
- [7] J. Freire, D. Koop, E. Santos, C. T. Silva, Provenance for computational tasks: A survey, *Computing in Science and Engineering* 10 (3) (2008) 11–21.
- [8] D. Koop, E. Santos, J. F. Bela Bauer, Matthias Troyer, C. T. Silva, Bridging workflow and data provenance using strong links, in: *SSDBM*, 2010, pp. 397–415.
- [9] D. Koop, C. Scheidegger, J. Freire, C. T. Silva, The provenance of workflow upgrades, in: *IPAW*, 2010, pp. 2–16.
- [10] C. E. Scheidegger, H. T. Vo, D. Koop, J. Freire, C. T. Silva, Querying and creating visualizations by analogy, *IEEE Transactions on Visualization and Computer Graphics* 13 (6) (2007) 1560–1567.
- [11] T. Ellkvist, L. Strömbäck, L. D. Lins, J. Freire, A first study on strategies for generating workflow snippets, in: *Proceedings of the ACM SIGMOD International Workshop on Keyword Search on Structured Data (KEYS)*, 2009, pp. 15–20.
- [12] E. Santos, L. Lins, J. Ahrens, J. Freire, C. T. Silva, Vismashup: Streamlining the creation of custom visualization applications, *IEEE Transactions on Visualization and Computer Graphics* 15 (6) (2009) 1539–1546.
- [13] The ALPS project, <http://alps.comp-phys.org>.
- [14] F. Alet, P. Dayal, A. Grzesik, A. Honecker, M. Körner, A. Läuchli, S. R. Manmana, I. P. McCulloch, F. Michel, R. M. Noack, G. Schmid, U. Schollwöck, F. Stöckli, S. Todo, S. Trebst, M. Troyer, P. Werner, S. Wessel, The ALPS Project: Open Source Software for Strongly Correlated Systems, *Journal of the Physical Society of Japan* 74S (Supplement) (2005) 30–35. doi:10.1143/JPSJS.74S.30.
- [15] A. Albuquerque, F. Alet, P. Dayal, A. Feiguin, S. Fuchs, L. Gamper, E. Gull, S. Gürtler, A. Honecker, R. Igarashi, M. Körner, A. Kozhevnikov, A. Läuchli, S. R. Manmana, M. Matsumoto, I. P. McCulloch, F. Michel, R. M. Noack, G. Pawłowski, L. Pollet, T. Pruschke, U. Schollwöck, F. Stöckli, S. Todo, S. Trebst, M. Troyer, P. Werner, S. Wessel, The ALPS project release 1.3: Open-source software for strongly correlated systems, *Journal of Magnetism and Magnetic Materials* 310 (2, Part 2) (2007) 1187–1193.
- [16] B. Bauer, L. D. Carr, A. Feiguin, J. Freire, S. Fuchs, L. Gamper, J. Gukelberger, E. Gull, S. Guertler, A. Hehn, R. Igarashi, S. V. Isakov, D. Koop, P. N. Ma, P. Mates, H. Matsuo, O. Parcollet, G. Pawłowski, J. D. Picon, L. Pollet, E. Santos, V. W. Scarola, U. Schollwöck, C. Silva, B. Surer, S. Todo, S. Trebst, M. Troyer, M. L. Wall, P. Werner, S. Wessel, The ALPS project release 2.0: Open source software for strongly correlated systems, paper available at <http://arxiv.org/pdf/1101.2646> and underlying workflows at <http://arxiv.org/abs/1101.2646> (Jan. 2011). arXiv:1101.2646.
- [17] Guidelines for Research Integrity and Good Scientific Practice at ETH Zurich, <http://www.vpf.ethz.ch/services/researchethics/Broschure>, accessed January, 2011.

²See the Tuning case on the Sigmod home page – <http://effdas.itu.dk/repeatability/tuning.html>