Examining Statistics of Workflow Evolution Provenance: A First Study

Lauro Lins, David Koop, Erik W. Anderson, Steven P. Callahan,
Emanuele Santos, Carlos E. Scheidegger, Juliana Freire, and Cláudio T. Silva

SCI Institute & School of Computing, University of Utah

1 Introduction

Provenance (also referred to as audit trail, lineage, and pedigree) captures information about the steps used to generate a given data product. Such information provides documentation that is key to determining data quality and authorship, and necessary for preserving, reproducing, sharing and publishing the data. Workflow design, in particular for exploratory tasks (e.g., creating a visualization, mining a data set), requires an involved, trial-and-error process. To solve a problem, a user has to iteratively refine a workflow to experiment with different techniques and try different parameter values, as she formulates and test hypotheses. The maintenance of detailed provenance (or history) of this process has many benefits that go beyond documentation and result reproducibility. Notably, it supports several operations that facilitate exploration, including the ability to return to a previous workflow version in an intuitive way, to undo bad changes, to compare different workflows, and to be reminded of the actions that led to a particular result [2].

As provenance-enabled systems are deployed, and increasing volumes of provenance information are collected, there is a unique opportunity to leverage and obtain useful knowledge from this data. In this paper, we take a first step at analyzing this data. We present a preliminary analysis of workflow evolution provenance generated by thirty subjects who worked on six distinct exploratory tasks over the period of four months. This initial analysis shows that useful statistics can be extracted from this data that provide insights into how different people interact with workflow systems to solve problems.

2 Workflow Evolution Provenance: Background

Because scientific tasks evolve as users switch input data, vary parameters, and investigate different approaches, scientists often need to manage a large collection of workflows. The change-based provenance model [2] treats a workflow specification as a first-class data item and captures the provenance of its evolution by recording every change to the specification. As a user modifies a workflow (e.g., by adding a module, changing a parameter or deleting a connection), the provenance mechanism transparently records each change, akin to a database

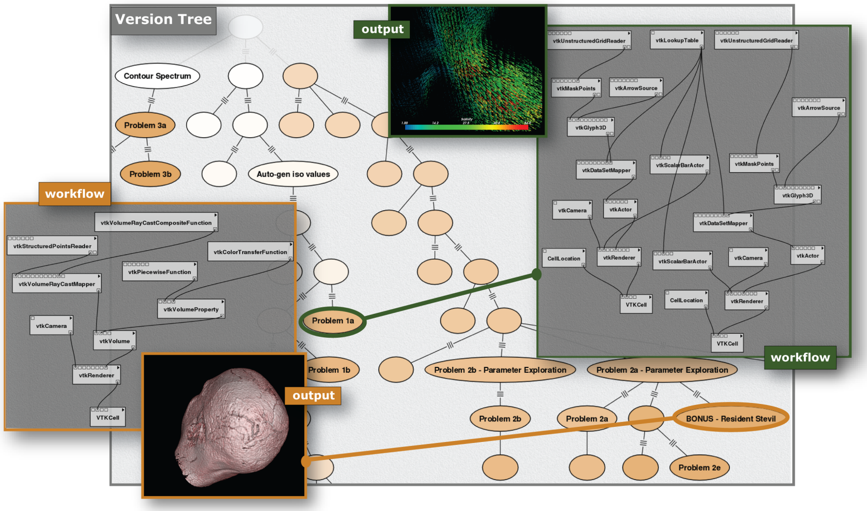

Fig. 1. A version tree with two workflow specifications and their outputs

transaction log. We can then reconstruct any workflow by replaying the sequence of captured changes from an empty specification to the desired version. In contrast to previous models which only capture provenance of data products (i.e., information about how a given data product was generated) [5], the change-based model captures both workflow and data provenance: it maintains a detailed record of the trail created by a user while solving a problem. In addition, this representation is concise and requires substantially less space than the alternative of storing multiple versions of a task specification.

Because the change-based provenance model captures the derivation of workflows, we can represent workflow evolution as a tree where each node is a *version* of the workflow specification and each edge coincides with an action. Given an edge from a parent node w_p to a child node w_c, its corresponding *action* is the sequence of changes necessary to transform w_p into w_c. Figure 1 shows an example of a workflow version tree, a couple of the workflow specifications, and the corresponding outputs from these workflows. Note that to reduce visual clutter, only important nodes of the tree are displayed by default, including those the user has tagged.

We have shown that maintaining detailed provenance of workflow evolution has many benefits and supports various activities that are crucial for performing reflective reasoning and obtaining insights, such as for example, following chains of reasoning backward and forward and comparing different results [3]. The tree-based view allows users to work collaboratively, to return to a previous version in an intuitive way, to undo bad changes, to reuse workflows and workflow fragments, to compare different workflows and their results, and to be reminded of the actions that led to a particular result [2, 4].

Task	Description	Difficulty	Open-Endedness
Task 1	Introduction	1	1
Task 2	2D Visualization Techniques	3	2
Task 3	Scalar & Vector Field Visualization	3	2
Task 4	Isosurfacing & Volume Rendering	4	3
Task 5	Diffusion Tensor Imaging & InfoVis	4	4
Task 6	Open-Ended Visualization	5	5

Table 1. Description of the six tasks involved in the study with the instructor's expectation of difficulty and open-endedness on a scale from 1 to 5.

The change-based model was originally implemented in the VisTrails system.[1] More recently, other workflow systems, including Taverna [6] and Kepler [1], have started to capture workflow evolution provenance.

3 Extracting Statistics from Workflow Evolution

Workflow evolution provenance makes it possible to analyze, in an unobtrusive manner, different aspects of workflow design. Furthermore, it provides a means to evaluate the utility of workflow systems and provenance to users, as they solve problems using workflows. In this section, we present an initial case study and discuss some statistics that can be extracted from this kind of provenance.

3.1 The Data

Our dataset was collected during a scientific visualization course.[2] A total of thirty students took the course. Throughout the semester, they were assigned six different tasks with fixed deadlines. Table 1 provides a short description as well as a subjective evaluation by the course instructor of the difficulty and open-endedness of each task.

Students used VisTrails to complete the tasks and for each task, they submitted a file containing all the actions they performed. These actions are transparently captured by VisTrails and stored according to the change-based model. Each action has a unique identifier; the identifier of its parent action; the user who performed the action; a timestamp indicating when the action took place; an optional tag; free-text annotations; and the required information to reproduce the action.

3.2 Analyzing Evolution Provenance at Different Levels

Because our provenance data encompasses a range of tasks completed by a set of users, it can be analyzed on different levels. Globally, we can observe trends across all tasks and users. At the task level, we can attempt to characterize tasks by the types of actions involved. Finally, for a specific user, we can drill down to assess progress, work habits, and strategies used for different tasks.

Because we know exactly when each action occurred, it is possible to plot the total workload against time. The activity histogram in Figure 2 shows that, unsurprisingly, most work was condensed into the few days preceding the task

[1] http://www.vistrails.org

[2] http://www.vistrails.org/index.php/SciVisFall2007

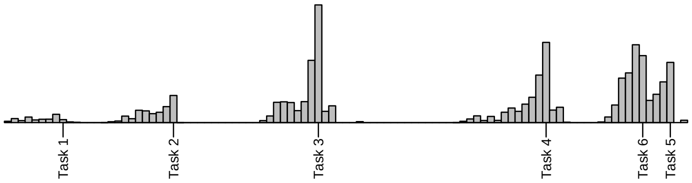

Fig. 2. Activity histogram binned by date with due dates indicated

deadlines. Besides that, the activity histogram also gives a good sense of which tasks required more effort. Although this measure may not match the assessment of the instructor, it gives a better measure of the effort the students put forth.

Global Analysis.

One useful feature of workflow evolution provenance is that users can interact with this provenance as they work. For example, in VisTrails users can at any time access the version tree and select any existing workflow to execute it, to inspect its specification or to modify it. In this last case, a new *branch* with the modified workflow specification is created as a new leaf of the tree. In order to help users to identify workflow specifications, VisTrails allows them to *tag* the nodes in the tree. In our analysis, we found that the number of branches in the version tree is correlated with the number of tagged nodes, as shown in Figure 3. This indicates that, as users have to revisit a previously defined workflow, they would select a tagged node because it is easier to identify.

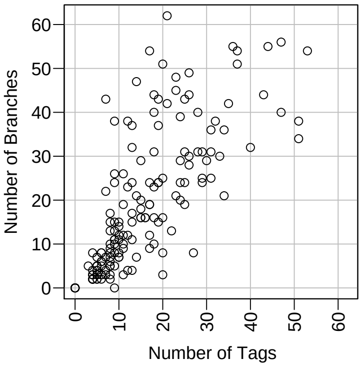

Fig. 3. The correlation between the number of branches and the number of tags per user-task.

Analysis of Tasks. Workflow evolution information can also be helpful to characterize tasks. As noted in Table 1, the tasks assigned to the scientific visualization students varied in their goals, difficulty, due date, and how open-ended they were. To illustrate how workflow evolution data can be used to gain some insight into the types of work involved in a task, we classified the actions involved in workflow development into: *structural actions* (addition and deletion of modules and connections in the workflow); *parameter actions* (modification of parameter values in the workflow); and *layout actions* (changes to the locations of modules in visual programming interface).

Figure 4 shows an attempt to characterize tasks by the breakdown of actions involved. For all users, we calculated the overall percentage of actions that were structural, parameter and layout actions across all tasks (Figure 4(a)). In addition, we computed these percentages for each task, as shown in Figure 4(b), (c) and (d). The distributions of these percentages were plotted as boxplots. Note that the percentage of actions spent changing parameters has the greatest

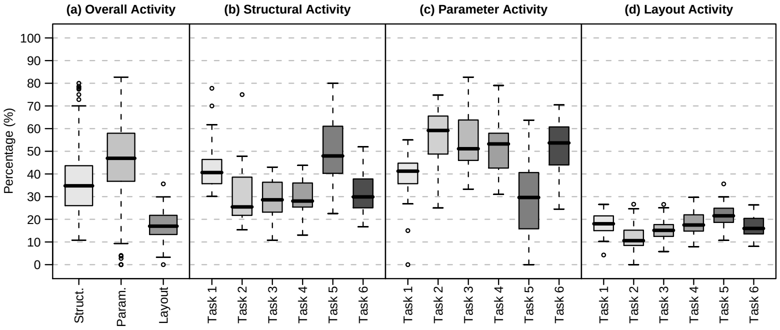

Fig. 4. Workflow Structural, Parameter and Layout Activity

variance for most tasks. This should be expected as some users locate correct parameter values faster than others, and some will also expend more effort tweaking parameters than others. Another interesting feature of these plots is that Task 5 shows more structural activity than Tasks 2, 3, and 4. This is explained by the fact that students were given examples for the previous three tasks, and in Task 5, they were left to discover how to create workflows from scratch.

Analysis of Users. A useful application of workflow evolution provenance is to help in understanding how different users approach a problem. Figure 5 shows two trees created by different users for the same task. User 1 and User 2 clearly have different development styles: the tree derived by User 2 is both shorter and narrower than that of User 1. This figure also shows a plot of the branching factor of the version trees across the tasks for User 1 and User 2. A smaller branching factor indicates that a more direct path was used to obtain a solution. In contrast, a larger branching factor indicates that more trial-and-error steps were followed. There are many cases where branching can be useful, including when a user wishes to develop workflows that share a common sub-workflow: the user designs the first workflow, goes to the version tree, selects the node corresponding to the common sub-workflow and from there branches to the second workflow. We found a range of branching factors that varied across users and tasks.

Branching is just one variable from the workflow evolution provenance data that can be used to identify "user signatures", other variables, such as the time between actions and the umber of sessions may also lead to insights in this respect.

4 Discussion and Future Work

We have shown that workflow evolution provenance allows one to measure, summarize, and analyze new aspects of workflow specification and design. A detailed analysis of how time is spent in workflow design can help to provide an understanding of how users interact with workflow systems. In addition, these statistics can produce insights into the potential bottlenecks and how these systems can be improved. While our results represent only an initial examination, we have discovered a number of areas where comparative statistics offer a window into general workflow design patterns, task characterization, and exploratory styles.

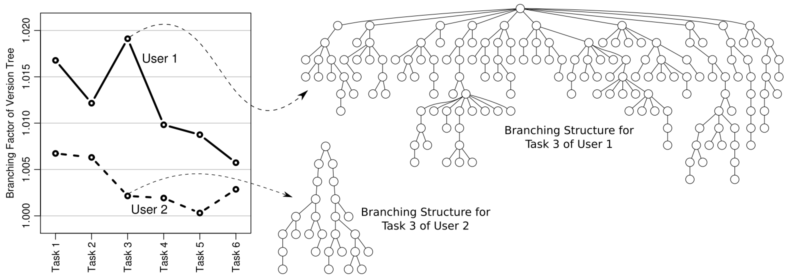

Fig. 5. Plot of Branching Factors for the six tasks from two different users. The branching structure for Task 3 is depicted on the right.

Besides investigating additional measures and statistical analyses, there are several avenues we plan to pursue in future work. In the course of our study, we have identified some limitations of the VisTrails provenance capture mechanism. We plan to improve and augment the variables captured by the change-based model to allow for more accurate and detailed analyses. Specifically, while each change is time-stamped, it is difficult to determine the actual time involved in performing a single action. In addition, information about distinct sessions of work would be useful to better determine the actual time spent accomplishing the computational tasks. We also plan to cross quality or merit data about the workflow specifications with the provenance data to infer information about which practices led to good workflow specification and how time was used in these cases. For our initial analysis we considered only general actions for modifying workflows. In future work, we plan to perform analyses that take into account the semantics of the individual actions. For example, instead of looking at the addition and deletion of modules, for a visualization task, we could consider the addition of a volume renderer or of an isosurface extraction. By doing so, we could measure the effort involved in applying these two different visualization techniques.

References

1. I. Altintas, O. Barney, and E. Jaeger-Frank. Provenance collection support in the kepler scientific workflow system. In *Proceedings of the International Provenance and Annotation Workshop (IPAW)*, pages 118–132, 2006.
2. J. Freire, C. T. Silva, S. P. Callahan, E. Santos, C. E. Scheidegger, and H. T. Vo. Managing rapidly-evolving scientific workflows. In *International Provenance and Annotation Workshop (IPAW)*, LNCS 4145, pages 10–18, 2006.
3. D. A. Norman. *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine.* Addison Wesley, 1994.
4. C. E. Scheidegger, H. T. Vo, D. Koop, J. Freire, and C. T. Silva. Querying and creating visualizations by analogy. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1560–1567, 2007.
5. Y. L. Simmhan, B. Plale, and D. Gannon. A survey of data provenance in e-science. *SIGMOD Record*, 34(3):31–36, 2005.
6. J. Zhao, C. Goble, R. Stevens, and D. Turi. Mining taverna's semantic web of provenance. *Concurrency and Computation: Practice and Experience*, 2007.