EuroRV3: EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization (2016)
K. Lawonn, M. Hlawitschka, and P. Rosenthal (Editors)

Towards Multi-user Provenance Tracking of Visual Analysis Workflows over Multiple Applications

C. Hänel[1], M. Khatami[1], T. W. Kuhlen[1], and B. Weyers[1]

[1] Visual Computing Institute, JARA - High-Performance Computing, RWTH Aachen University, Germany

Abstract

Provenance tracking for visual analysis workflows is still a challenge as especially interaction and collaboration aspects are poorly covered in existing realizations. Therefore, we propose a first prototype addressing these issues based on the PROV model. Interactions in multiple applications by multiple users can be tracked by means of a web interface and, thus, allowing even for tracking of remote-located collaboration partners. In the end, we demonstrate the applicability based on two use cases and discuss some open issues not addressed by our implementation so far but that can be easily integrated into our architecture.

Categories and Subject Descriptors (according to ACM CCS): C.2.4 [Computer Systems Organization]: Distributed Systems—Client/Server; D.2.13 [Software]: Reusable Software—Reusable Models; H.3.0 [Information Systems]: Information Systems Applications—General

1. Introduction

Reproducibility in interactive visual analysis is key to gather valid and verifiable scientific results in data-driven and empirical research (see [SPG05, dMdCDLC*13]). Analysis workflows which include visual analysis methods are determined not exclusively by the scientific domain in which they are applied but also by the scientist's objectives and the data to be analyzed. Neuroscience is a scientific domain which raises very special requirements regarding data to be analyzed as well as regarding the structure of the analysis workflows. Neuroscientific data is rather heterogeneous and emerges from various sources, such as simulations on different biological scales or experimental data gathered from in-vivo or post-mortem experiments. The workflows combine a tight integration of pure data-based analysis, such as calculating statistics, and visual analysis using interactive tools. Furthermore, data analysis workflows in neuroscience tend to be very dynamic and change over time. Finally, various (remote) partners are involved in the analysis of neuroscientific data sets, such as it is crucial in the Human Brain Project (HBP). Therefore, the HBP will develop six information and communications technology (ICT) platforms for Neuroinformatics, Brain Simulation, High Performance Computing, Medical Informatics, Neuromorphic Computing and Neurorobotics, all developing capabilities ultimately aimed at enabling Neuroscience to understand the human brain.

Based on the work by Ragan et al. [RESC16] who discuss a set of types and purposes of provenance tracking, we structure our requirements for tracking interactive visual analysis workflows, which can be part of analysis workflows as mentioned above. Therefore, we reason why data, visualization, interaction, and insight are provenance types we want to address and why especially collaboration is an important purpose. In this regard, the main goal we want to achieve with provenance tracking is to capture all visualization states that occur during exploration that show relevant views onto the data. This is closely linked to provenance of interaction. During an interactive exploration more insights and findings can be made which cannot be represented by means of static images. For instance, animation is crucial for time-variant data, navigation along a path allows for a better spatial understanding, or the interactive change of a lookup table can reveal distinct structures.

Provenance of insights is highly relevant for interactive analysis. By use of annotations a researcher can label important analysis steps or partial results to give a fast guide to relevant points in the tracking history. Provenance of rationale extends this by giving an understanding why the analysis was performed in a certain way. However, this is a rather complex aspect in its realization due to application and domain-specific requirements and, therefore, will not be explicitly addressed in this scope but is meant to be addressable by the extendibility of the presented concept.

The relevance of data provenance is strongly application-dependent. While sometimes one data set is generated and is not changed during the whole analysis process, running simulations can continuously change the data basis. There are several tools that are specialized to track this information (see [RESC16]) and, therefore, could be applied for according needs. For the visual analysis workflow itself, we want to focus only on the tracking of the actual data, for example, if permanent data modifications are applied to it (also this is usually not the case for visual analysis tools), or if any data like a screenshot are generated.

DOI: 10.2312/eurorv3.20161112

Accessing the provenance by means of recall is a purpose that nearly all tracking tools address. More specific is the support of collaboration. It can be distinguished between synchronous and asynchronous, and co-located and distributed settings. The multi-view and multi-application tracking as required for neuroscientific workflows generates the need for synchronous tracking independent of the location of the scientists. Furthermore, remote partners should also be able to work together and create a combined provenance.

2. Related Work

Based on the classification of provenance types and purposes by Ragan et al. [RESC16], we take a look at relevant work that addresses our three main requirements for tracking visualization and interaction, as well as collaborative analysis (distributed and multi-application scenarios). A well-established tool for tracking provenance in scientific workflows is VisTrails [BCC*05]. Computational and visualization workflows can be easily reproduced by means of a graphical representation. The current version of VisTrails has a decent feature set which covers, for example, a multi-view visualization of different branches in the tracking history. Furthermore, it supports collaboration by differentiation between changes applied by multiple users and the possibility to export the tracked data to other systems. However, VisTrails cannot completely fulfill our requirements. One important aspect is the limitation in the representation of interactions in their provenance model. All interactions are ultimately interpreted as separate states such that a change in the color lookup table is, for example, always discrete and a transition between two states cannot be represented. Furthermore, the use of the provenance tracking is only possible within the VisTrail environment. VisTrails enables the user to include her own visualization algorithm by means of a library, but this hinders the use of existing or more individual applications as, for example, for stereoscopic rendering or ones that use specific interaction devices. Last, we would like to point out the deficit in the collaboration aspects which is not supporting people in different locations to work on the same provenance track without explicitly transferring the history.

Heer et al. [HMSA08] presented a tool for tracking and visualizing provenance in the area of information visualization. Interactions in different graphical views can be tracked and later on presented or manipulated. Also collaborative issues are addressed. Unfortunately, these functions are limited within the use of the proprietary software Tableau and, therefore, we cannot extract and generalize them for our needs.

GraphTrail is a visualization application by Dunne et al. [DHRL*12] that supports the analysis of large heterogeneous networks. A key feature is the visualization of the exploration history that allows for a recapitulation of the analysis results of previous sessions or a combination of different analysis branches. The use of the provided pivot mechanism is labeled in GraphTrail to make transitions between two states in the analysis more comprehensible. However, the tool is limited to graph visualizations, only a selected set of interactions is saved in their history model, and the collaboration aspect is insufficiently elaborated.

As exemplary shown on the tools above, no existing provenance tracking systems exists that fulfills our complex requirements to track interactive scientific visual analysis workflows with multiple applications and users. Therefore, we describe our prototypical approach to resolve the technical limitations in the following.

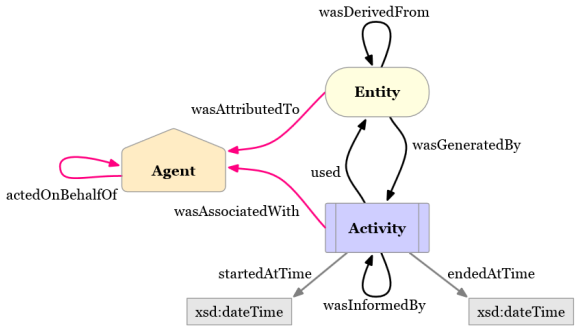

Figure 1: *PROV model schema by Belhajjme et al. [BCC*13].*

3. Architecture

In this section, based on the requirements introduced above, the architecture of our prototype for a provenance tracking system is derived. One major aspect is that our provenance model has to be capable of tracking interactions during visual analysis. When considering general provenance tracking models like the Open Provenance Model (OPM) [MCF*11] or the PROV model [MGC*15], we see a flexible use and, more importantly, being compatible to other tracks using the same format. However, in comparison to OPM, the PROV model provides a change of entities over time what is a relevant aspect for our required provenance of interaction and, therefore, we decided to base our implementation on the latter. In Figure 1 it can be seen how this feature is represented by starting and ending time of an activity. In this schema, entities refer to physical, digital, or conceptual objects and they can be described from different perspectives by the specification of attributes. Activities like a process, action, or procedure represent the dynamic aspects in the model. They can change attributes of entities to derive a new one or create entirely new entities. An agent can be, for example, a human or software that is at least partially responsible for an occurring activity.

This provenance model is included into our overall prototype, which is shown in Figure 2. It illustrates its components and interactions between them. Due to one of the main identified challenges, multi-user provenance tracking in remote locations, we decided to employ a server/client architecture. Using a centralized engine does not only construct a platform for provenance of shared workflows, but also limits the maintenance requirements to the server side. In terms of a server/client architecture, the server is an engine that is responsible for the core of provenance, client authorization, and data management, whereas client refers to different kinds of applications, workflows, pipelines, etc. that aim at taking benefit of provenance tracking. The query processing, checking data validity, and all other functions related to data management are accomplished on the server side which leads to less load on the client side. Therefore, for any desired application only a small interface needs

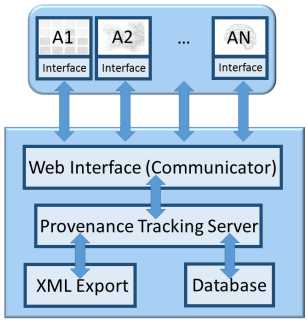

Figure 2: *Architecture of our proposed provenance tracking system. Multiple applications (A1 to AN) are connected via interface to a server providing the actual provenance tracking.*

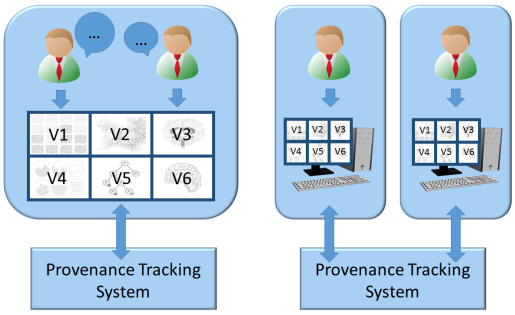

Figure 3: *Schema of two use cases. Use case 1 (left): Two users are collaborating synchronously in the same location. Use case 2 (right): Two users are working asynchronously in distinct locations.*

to be implemented and integrated on the client side that is able to communicate with the server side to send and receive messages which encapsulate provenance information.

In comparison to that, the server application is independent of the visualization applications and the system they are executed on. It consists of four main parts: the core, the communicator, the database handler, and the XML handler. The core, as central service, acts like a hub for other parts and connects different subservices. This component comprises the actual provenance model and provides several common queries and provenance analyses, for example, to obtain affected entities by a specific activity or list all collaborating agents. Besides the key elements in PROV model, a workflow table is designed for multi-user workflow authentication and authorization. A web interface–the communicator–is responsible for sending messages to and receiving them from clients. The database handler takes care of data management such as authorization, storing, or retrieving provenance information. The XML handler converts the database format to be transferable to the client. As a complement to the provided query functions in the core service, transferring the whole provenance information in XML-based format enables the user to apply her analyses and queries on provenance information.

4. Implementation

For the realization of our prototype for tracking interaction for distributed analysis tools (InDiProv), the server application was implemented in C++ and the *ZeroMQ* library was employed as messaging service. As a number of queries can occur simultaneously, the range of ZeroMQ send/receive sockets can be modified according to the number of possible clients. Each socket of the communicator runs on an independent thread to increase performance. We decided to use the JavaScript Object Notation (JSON) [Bra14], a widely used lightweight data-interchange format, as standard communication format to be send via ZeroMQ. JSON is supported by most programming languages and gives the opportunity that clients with different platforms are able to communicate with the server, what fulfills the goal of generalization. Each provenance task, such as workflow creation, storing provenance, or querying the tracking, is encapsulated as a JSON exchange message and send to the server for corresponding response. In the same manner the response from the server is JSON-based.

To store the PROV data, we utilized a relational database schema as, for example, demonstrated by Lim et al. [LLCF11] for OPM, but based on the PROV Model in our case. Therefore, we chose the well-known and free database management system *MYSQL* and the *mysqlcppconn* library for data manipulation. Furthermore, it is common to store provenance data in XML format like shown by the VisTrails export function. To enable this feature, we use the PROV-XML schema that can serialize instances of the PROV data model to XML. The latter is specifically relevant in the case of visual analysis as part of a bigger analysis workflow, as discussed in the introduction. Using standard formats enables the integration of various types of provenance information into one coherent context.

5. Use Case

In order to demonstrate the behavior and flexibility of our prototype, we present two use case scenarios in the following. Thereby, we do not relate to specific visualization applications to give a more generalizable idea of the integration of our InDiProv tool.

5.1. Use Case 1: Synchronous Work in Same Location

In the first use case, illustrated in Figure 3 (left), two users are synchronously collaborating together at one large tiled display. Various visualization applications (*V1*-*V6*) that display specific aspects of the same raw data are running distributed over the different screens and are all connected to the same provenance tracking system. To start a session, the workflow is being initialized by a name and secured with a password. The system returns an ID that can be used in combination with the password to insert, update, and request provenance tracking data. In the beginning, the provenance model consists of one agent (representing the two collaborating users) and one entity (the raw data). The model is extended when the user triggers the start of the applications and the data is visualized by means of different algorithms. This is represented by the PROV property *wasStartedBy*. New entities (the six visualizations) are created derived from the raw data. Any following interaction activity like changing the lookup table, translation, or starting an

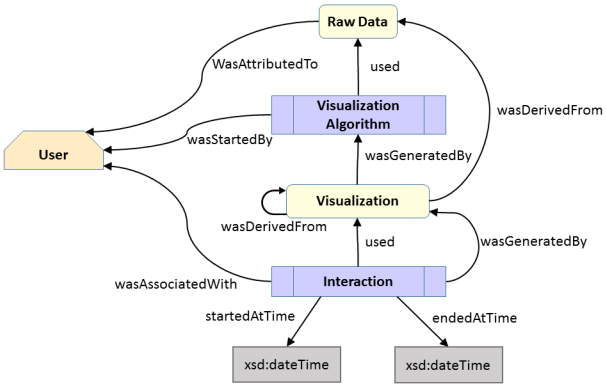

Figure 4: *PROV model schema transferred into Use Case 1 whereat the six visualization applications are represented only once by the entity Visualization, and the activities Visualization Algorithm and Interaction for reasons of simplicity.*

animation affects the visualization entity as shown in Figure 4. This information is stored in the tracking system and the relations corresponding to the PROV model (see Figure 1) are inserted. However, it has to be defined in each application interface which activities are traceable. Due to the threading on the server side, parallel ongoing interactions in different applications can be differentiated. Furthermore, the PROV model provides a functionality to add additional meta data by means of a *value* element to a PROV record to allow, for example, for annotations or storing meta information of the raw data. In our use case, this feature also enables the user to label specific views / time points. Nevertheless, the system cannot natively distinguish which user performed which action–as we have only one agent–and everything is represented in a sequential stream.

5.2. Use Case 2: Asynchronous Work in Remote Locations

In the second use case (see Figure 3 (right)), two persons are working in distinct locations at their workstations. Both work asynchronous on the visual analysis of the same set of data. In addition to the shared workflow that stores the provenance tracking of the collaboration, each user is also able to have her workflow. The workflow is established in the same manner as in Use Case 1, however now the responsibilities of each user are distinguishable. Up to this point the PROV model schema would look for both users the same as in Use Case 1. However, to track communication between the collaborators is an issue not taken into consideration so far. Although, the communication is not explicitly covered within InDiProv there are two possibilities to track this information. Firstly, one could implement an interface to software that is used for communication. By doing so, for example, text messages from a chat can be included in the track as *values*. Secondly, any communication outcomes can be included manually by means of an additional *communication* activity. If, for example, the users were informed about new time step data of a simulation ready to be included in their visual analysis, the *communication* activity changes the raw data. As the *visualization algorithm* activity uses the *raw data* entity, a property *wasInformedBy* is added from the *visualization algorithm* to the *communication* activity. These functionality can also be applied to store rationale of the analysis process to a limited extend.

6. Discussion and Future Work

Provenance is a highly relevant topic for data-driven research including such scientific workflows which facilitates interactive and visual analysis methods. In this paper, we presented an approach that on the one hand addresses the interactive nature of visual analysis tools and applications, and on the other hand on-site and remote collaboration between scientists. Both aspects have been addressed on two abstraction levels; the description of interactive analysis workflows in the PROV model and from a software architectural perspective. In two use cases, the feature set of the prototypical implementation of the architecture called *InDiProv* has been demonstrated.

However, the current prototypical implementation has several limitations, which can be covered in the proposed architecture but are not implemented yet. First, the handling of the server-side provenance track is currently very simplistic. For instance, there is no support for interactively using the tracked provenance information for replay or branching. Replay enables the user to retrace processes of gaining insight and also to find possible errors in the method or results in an analysis. Branching goes a step further by enabling the user to decide at a certain point in the track to pick up the previous analysis and continue it following another idea or goal. This new branch has to be tracked and related to the previous one. In this case, not only the calculation of differences between tracks gets relevant but also the comparison of results and gained insights. Thus, branching affects all types of provenance and has to be elaborated in future work for the types of use cases presented.

Analysis (calculating differences and to comparison) and control (selecting certain steps in the workflow from the provenance track) can be supported by software tools which make this information accessible for the user. In case of the presented architecture, this would be another application connected to the server but in this case not as agent or object but as a controlling instance able to access the provenance track for such purposes.

After implementing these missing features, we want to analyze how InDiProv realistically supports our neuroscientific collaboration partners from Jülich Research Centre (Germany) within their analysis processes. Furthermore, the presented prototype is not only limited to neuroscientific visual analysis. Beside others, one possible example could be surgery planning with multiple imaging modalities (CT, blood flow data, etc.) including the collaboration with other experts to review critical surgery planning.

To make the solution presented in this paper usable for the community, it is accessible as open source project on GitHub (`https://github.com/HBPVIS/InDiProv/`). The repository offers the server implementation as well as a C++-based client implementation and a small example visualization application. We plan to extend the documentation and to promote the current implementation regarding the previous discussed points.

Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 604102 (HBP) and from the Helmholtz Portfolio Theme "Supercomputing and Modeling for the Human Brain".

References

[BCC*05] BAVOIL L., CALLAHAN S. P., CROSSNO P. J., FREIRE J., VO H. T.: VisTrails: Enabling interactive multiple-view visualizations. In *IEEE Visualization 2005* (2005), pp. 135–142. doi:10.1109/VISUAL.2005.1532788. 2

[BCC*13] BELHAJJAME K., CHENEY J., CORSAR D., GARIJO D., SOILAND-REYES S., ZEDNIK S., ZHAO J.: *PROV-O: The PROV Ontology*. Tech. rep., World Wide Web Consortium, 2013. 2

[Bra14] BRAY T. (Ed.): *The JavaScript Object Notation (JSON) Data Interchange Format*. Tech. rep., RFC 7159, 2014. doi:10.17487/RFC7159. 3

[DHRL*12] DUNNE C., HENRY RICHE N., LEE B., METOYER R., ROBERTSON G.: GraphTrail: Analyzing Large Multivariate, Heterogeneous Networks While Supporting Exploration History. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), CHI '12, ACM, pp. 1663–1672. doi:10.1145/2207676.2208293. 2

[dMdCDLC*13] DE MENDONÇA R. R., DA CRUZ S. M. S., DE LA CERDA J. F. S. M., CAVALCANTI M. C., CORDEIRO K. F., CAMPOS M. L. M.: LOP: Capturing and Linking Open Provenance on LOD Cycle. In *Proceedings of the Fifth Workshop on Semantic Web Information Management* (2013), SWIM '13, ACM, pp. 3:1–3:8. doi:10.1145/2484712.2484715. 1

[HMSA08] HEER J., MACKINLAY J., STOLTE C., AGRAWALA M.: Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation. *IEEE Transactions on Visualization & Computer Graphics 14* (2008), 1189–1196. doi:10.1109/TVCG.2008.137. 2

[LLCF11] LIM C., LU S., CHEBOTKO A., FOTOUHI F.: Storing, reasoning, and querying OPM-compliant scientific workflow provenance using relational databases. *Future Generation Computer Systems 27*, 6 (2011), 781–789. doi:10.1016/j.future.2010.10.013. 3

[MCF*11] MOREAU L., CLIFFORD B., FREIRE J., FUTRELLE J., GIL Y., GROTH P., KWASNIKOWSKA N., MILES S., MISSIER P., MYERS J., PLALE B., SIMMHAN Y., STEPHAN E., DEN BUSSCHE J. V.: The Open Provenance Model core specification (v1.1) . *Future Generation Computer Systems 27*, 6 (2011), 743–756. doi:10.1016/j.future.2010.07.005. 2

[MGC*15] MOREAU L., GROTH P., CHENEY J., LEBO T., MILES S.: The rationale of {PROV}. *Web Semantics: Science, Services and Agents on the World Wide Web 35, Part 4* (2015), 235–257. doi:10.1016/j.websem.2015.04.001. 2

[RESC16] RAGAN E., ENDERT A., SANYAL J., CHEN J.: Characterizing Provenance in Visualization and Data Analysis: An Organizational Framework of Provenance Types and Purposes. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2016), 31–40. doi:10.1109/TVCG.2015.2467551. 1, 2

[SPG05] SIMMHAN Y., PLALE B., GANNON D.: *A Survey of Data Provenance Techniques. Technical Report IUB-CS-TR618*. Tech. rep., Computer Science Department, Indiana University, 2005. 1