Report from Dagstuhl Seminar 18462

Provenance and Logging for Sense Making

Edited by

Jean-Daniel Fekete[1], T. J. Jankun-Kelly[2], Melanie Tory[3], and Kai Xu[4]

1 **INRIA Saclay – Orsay, FR,** jean-daniel.fekete@inria.fr
2 **Mississippi State University, US,** tjk@acm.org
3 **Tableau Software – Palo Alto, US,** mtory@tableau.com
4 **Middlesex University – London, GB,** k.xu@mdx.ac.uk

Abstract

Sense making is one of the biggest challenges in data analysis faced by both the industry and the research community. It involves understanding the data and uncovering its model, generating a hypothesis, selecting analysis methods, creating novel solutions, designing evaluation, and also critical thinking and learning wherever needed. The research and development for such sense making tasks lags far behind the fast-changing user needs, such as those that emerged recently as the result of so-called "Big Data". As a result, sense making is often performed manually and the limited human cognition capability becomes the bottleneck of sense making in data analysis and decision making.

One of the recent advances in sense making research is the capture, visualization, and analysis of provenance information. Provenance is the history and context of sense making, including the data/analysis used and the users' critical thinking process. It has been shown that provenance can effectively support many sense making tasks. For instance, provenance can provide an overview of what has been examined and reveal gaps like unexplored information or solution possibilities. Besides, provenance can support collaborative sense making and communication by sharing the rich context of the sense making process.

Besides data analysis and decision making, provenance has been studied in many other fields, sometimes under different names, for different types of sense making. For example, the Human-Computer Interaction community relies on the analysis of logging to understand user behaviors and intentions; the WWW and database community has been working on data lineage to understand uncertainty and trustworthiness; and finally, reproducible science heavily relies on provenance to improve the reliability and efficiency of scientific research.

This Dagstuhl Seminar brought together researchers from the diverse fields that relate to provenance and sense making to foster cross-community collaboration. Shared challenges were identified and progress has been made towards developing novel solutions.

Seminar November 11–16, 2018 – http://www.dagstuhl.de/18462
2012 ACM Subject Classification Human-centered computing → Visualization theory, concepts and paradigms
Keywords and phrases Logging, Provenance, Sensemaking, Visualization
Digital Object Identifier 10.4230/DagRep.8.11.35
Edited in cooperation with Christian Bors

1 Executive Summary

Kai Xu (Middlesex University – London, GB)
Jean-Daniel Fekete (INRIA Saclay – Orsay, FR)
T. J. Jankun-Kelly (Mississippi State University, US)
Melanie Tory (Tableau Software – Palo Alto, US)

Sense making is one of the biggest challenges in data analysis faced by both industry and research community. It involves understanding the data and uncovering its model, generating hypothesis, select analysis methods, creating novel solutions, designing evaluation, and the critical thinking and learning wherever needed. Recently many techniques and software tools have become available to address the challenges of so-called 'Big Data'. However, these mostly target lower-level sense making tasks such as storage and search. There is limited support for the higher-level sense making tasks mentioned earlier. As a result, these tasks are often performed manually and the limited human cognition capability becomes the bottleneck, negatively impacting data analysis and decision making. This applies to both industry and academia. Scientific research is a sense making process as well: it includes all the sense making tasks mentioned earlier, with an emphasis on the generation of novel solutions. Similar to data analysis, most of these are conducted manually and considerably limit the progress of scientific discovery.

Visual Analytics is a fast-growing field that specifically targets sense making [6]. It achieves this by integrating interactive visualization with data analytics such as *Machine Learning*. It follows a human-centered principle: instead of replacing human thinking and expertise with algorithms and models, it enables the two to work together to achieve the best sense making result. Fast progress has been made in the last decade or so, which is evidenced by the publications in the Visual Analytics conferences such as IEEE VAST (part of IEEE VIS) and the increasing popularity of visual approaches in many other fields such as Machine Learning, Information Retrieval, and Databases.

One recent advance in Visual Analytics research is the capture, visualization, and analysis of *provenance* information. Provenance is the history and context of sense making, including the "7W" (Who, When, What, Why, Where, Which, and HoW) of data used and the users' critical thinking process. The concept of provenance is not entirely new. In 1996, Shneiderman recognized the importance of provenance by classifying *history* as one of the seven fundamental tasks in data visualization [4]. History allows users to review previous actions during visual exploration, which is typically long and complex. Provenance can provide an overview of what has been examined and reveal the gap of unexplored data or solutions. Provenance can also support collaborative sense making and communication by sharing the rich context of what others have accomplished [7].

The topic of provenance has been studied in many other fields, such as Human-Computer Interaction (HCI), WWW, Database, and Reproducible Science. The HCI research community heavily relies on user information, such as logging and observation, in their study. These closely relate to provenance and share the common goal of making sense of user behavior and thinking. The collaboration between the two fields can potential create novel solutions for some long-standing research challenges. For instance, it has been shown that provenance information can be used to semi-automate part of the qualitative analysis of user evaluation data [3], which is notoriously time-consuming.

The WWW and Database research community has been actively working on provenance for the last decade or so, with a particular focus on tracking data collection and processing.

This has led to the recent publication of the W3C reference model on provenance [1]. A important part of these efforts is to make sense of the source and quality of the data and the analyses base on them, which has a significant impact on their uncertainty and trustworthiness [1]. Similarly, there is a fast growing Reproducible Science community, whose interest in provenance is "improving the reliability and efficiency of scientific research ... increase the credibility of the published scientific literature and accelerate discovery" [2].

There is a trend of cross-community collaboration on provenance-related research, which has led to some exciting outcomes such as the work integrating visualization with reproducible science [5, 8]. However, there are still many challenging research questions and many provenance-related research efforts remain disconnected. This seminar brought together researchers from the diverse fields that relate to provenance. Shared challenges were identified and progress has been made towards developing novel solutions.

The main research question that this seminar aims to address is: **How to collect, analyze, and summarize provenance information to support the design and evaluation of novel techniques for sense making across related fields**. The week-long seminar started with a day of self-introduction, lighting talks, and research topic brain storming. The self-introduction allowed attendees to know each other better, and the lighting talks covered the latest work in the research fields related to provenance. Each participant proposed several research questions, which were then collated and voted on to form the breakout groups. The following are the research areas chosen by the participants:

- Storytelling and narrative;
- Provenance standard and system integration;
- Task abstraction for provenance analysis;
- Machine learning and provenance;
- User modeling and intent.

The rest of the week was breakout session, and each participant had the option to change group halfway. The seminar finished with a presentation from each group and discussions on the next steps to continue the collaboration. Many interesting problems were identified, and progress was made towards new solutions. Please refer to the rest of the report for the details on the identified research questions and the progress made by the end of week.

References

1 Melanie Herschel and Marcel Hlawatsch. Provenance: On and Behind the Screens. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, pages 2213–2217, New York, NY, USA, 2016. ACM.

2 Marcus R. Munafò, Brian A. Nosek, Dorothy V. M. Bishop, Katherine S. Button, Christopher D. Chambers, Nathalie Percie du Sert, Uri Simonsohn, Eric-Jan Wagenmakers, Jennifer J. Ware, and John P. A. Ioannidis. A manifesto for reproducible science. *Nature Human Behaviour*, 1(1):0021, January 2017.

3 P. H. Nguyen, K. Xu, A. Wheat, B. L. W. Wong, S. Attfield, and B. Fields. SensePath: Understanding the Sensemaking Process Through Analytic Provenance. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):41–50, January 2016.

4 Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages*, pages 336–, Washington, DC, USA, 1996. IEEE Computer Society.

5 C. T. Silva, J. Freire, and S. P. Callahan. Provenance for Visualizations: Reproducibility and Beyond. *Computing in Science Engineering*, 9(5):82–89, September 2007.

[1] https://www.w3.org/TR/prov-overview/

6 James J. Thomas and Kristin A. Cook, editors. *Illuminating the Path: The Research and Development Agenda for Visual Analytics.* National Visualization and Analytics Centre, 2005.

7 Kai Xu, Simon Attfield, T. J. Jankun-Kelly, Ashley Wheat, Phong H. Nguyen, and Nallini Selvaraj. Analytic provenance for sensemaking: A research agenda. *IEEE Computer Graphics and Applications*, 35(3):56–64, 2015.

8 Zheguang Zhao, Lorenzo De Stefani, Emanuel Zgraggen, Carsten Binnig, Eli Upfal, and Tim Kraska. Controlling false discoveries during interactive data exploration. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, pages 527–540, New York, NY, USA, 2017. ACM.

2 Table of Contents

3 Overview of Talks

3.1 Provenance in Databases

Leilani Battle (University of Maryland – College Park, US)

Recording the history (or provenance) of how a dataset was processed and making the history accessible through an intuitive interface is a challenging problem. From a database management systems (DBMS) perspective, the focus is on performance: efficiently recording information about how database queries process input data, enabling fast analysis of the results via database queries over the provenance data, and exploiting recorded provenance in downstream applications connected to the DBMS. In this presentation, I discuss the data management motivations, challenges and techniques for supporting provenance-based analysis.

3.2 Sensemaking, and the Analytic Provenance of Sense

Alex Endert (Georgia Institute of Technology – Atlanta, US)

People perform exploratory data analysis to gain insight and make sense of their data. This process is often called "sensemaking", and conceptual models exist that help describe the formation of mental models. The process of performing sensemaking often involves many operations, tasks, and interactions. "Analytic Provenance" is a term that describes the study of this iterative interactive process. This talk will introduce these two concepts, how they relate to each other, and what primary research challenges are in each of them.

3.3 (A Blazingly Fast Intro to) Reproducible Science

Claudio T. Silva (New York University, US)

In this ten minute talk, we give a short introduction to reproducible science. Using two examples of our research, we show the success and failure of producing results that can be reproduced. Our experience is similar to others, as noted in the reproducibility survey done by Nature [1]. We use as examples the recently established ACM Artifact Review and Badging policy, the ACM SIGMOD reproducibility effort, and the Graphics Replicability Stamp Initiative to highlight issues related to repeatability, replicability, and reproducibility of scientific results. We end with a brief discussion of the "reproducible paper", where we use the VisTrails system to propose a provenance-based infrastructure to support the life cycle of papers [2].

References

1 Monya Baker. Is there a reproducibility crisis? a nature survey lifts the lid on how researchers view the'crisis rocking science and what they think will help. *Nature*, 533(7604):452–455, 2016.

2 David Koop, Emanuele Santos, Phillip Mates, Huy T Vo, Philippe Bonnet, Bela Bauer, Brigitte Surer, Matthias Troyer, Dean N Williams, Joel E Tohline, et al. A provenance-based infrastructure to support the life cycle of executable papers. *Procedia Computer Science*, 4:648–657, 2011.

3.4 Provenance and Logging: HCI Perspectives

Melanie Tory (Tableau Software – Palo Alto, US)

This talk explores provenance and logging from the perspective of two distinct user groups: data analysts and researchers. Analysts need provenance information to support their analytical workflow, including review and reflection, collaboration, learning, and storytelling. HCI researchers need provenance information to understand user behavior and infer barriers to workflow where design changes may be warranted. In both cases, I argue that raw provenance information is not enough; it may need to be transformed and summarized to support user needs.

4 Working groups

4.1 Storytelling

Daniel Archambault (Swansea University, GB), T. J. Jankun-Kelly (Mississippi State University, US), Andreas Kerren (Linnaeus University – Växjö, SE), Robert Kosara (Tableau Software – Seattle, US), Ali Sarvghad (University of Massachusetts – Amherst, US), and William Wong (Middlesex University – London, GB)

In the context of storytelling, trust is how confident the audience/author is of the story contents. Depending on the audience, more or less provenance information would be required to establish trust. The first step in establishing trust is establishing authority. One way to establish authority is to demonstrate the provenance of the story to the audience. In particular, we show the data and the provenance information of how we got there. If both of these are available in a complete way to the audience, this can be one way to convince the audience that the author is trustworthy. A transparent communication of this information allows the audience to verify and replicate the steps leading to the creation of this story.

Counterstories are stories that run contrary or lead to the opposite conclusion of the proposed story given the data. With interaction and data provenance information, counterstory development can be supported. If during the presentation of a story, the author can convey that the counterstories are less probable than the proposed story, this can provide evidence that the proposed story is more trustworthy. Conversely, if an author finds the counterstory more credible, it can be promoted to the proposed story.

An ideal system would be able to propose areas of the data set that have not been explored in an intelligent way to verify that the exploration is complete. This process can be partially automated by metrics on the data set and can be influenced by interactive pruning of the provenance tree – the user annotates sections of the tree as uninteresting, providing some text to state why it is not relevant to the exploration. Thus, the creation of a credible story could follow this procedure:

Auto suggestions → Story ← User notes on analysis

For completeness, automated tools suggests areas of underexplored data guided by user annotations on the provenance tree to indicate why certain branches are not relevant to this analysis. The process converges on a credible story.

4.2 VAPS: Visual Analytics Provenance Standard for Cross-Tool Integration of Provenance Handling

Daniel Archambault (Swansea University, GB), Jean-Daniel Fekete (INRIA Saclay – Orsay, FR), Melanie Herschel (Universität Stuttgart, DE), T. J. Jankun-Kelly (Mississippi State University, US), Andreas Kerren (Linnaeus University – Växjö, SE), Robert S. Laramee (Swansea University, GB), Aran Lunzer (OS Vision – Los Angeles, US), Holger Stitz (Johannes Kepler Universität Linz, AT), and Melanie Tory (Tableau Software – Palo Alto, US)

The state of the art in capture and presentation of provenance for visual analytics is exemplified by a number of systems that support activities within a single tool, or at best across a set of tools that are constrained to work within a single predefined pattern of activity. For example, VisTrails [1], while equipped with provenance management and a provenance editor that support a class of workflow-style activities, cannot collect provenance outside its own world. However, visual analytics activities seldom take place within a single tool; it is typical for an analyst to refer to, and combine findings from, several tools that are running in parallel. These may include data wrangling programs, databases, visual analytics tools, presentation tools, and reporting systems. Managing provenance from within only one of these tools is hiding a large portion of the analytical work and—more importantly—opportunities to make sense of the user's activity through the observation of the provenance data. For example, a user taking a screen capture of the visualization application and pasting it into a presentation tool is an obvious sign that an insight (or a bug) has been discovered, but the visualization application has no way to know that a screen capture has been performed and will not see that insight event. Therefore, we believe a provenance management system should handle provenance data from multiple applications, and also show inter-application activities (e.g., cut/paste, screen-capture/paste). To achieve this vision, we should address multiple problems that are described in the next section. Some are challenging, and the challenges are listed and discussed in the second section. Fortunately, we envision a possible solution that we discuss in the third section. It will not solve the problem completely and will need time to improve, but we believe that existing technologies can be used to address the main issues in a realistic way.

4.2.1 Problem Statement

Managing provenance from multiple applications requires that all the applications record their provenance in a way that can be interpreted by the provenance visualization and management tool (PVMT) that is being used to oversee the analysis session. This suggests the need for a standard format for provenance data. While no such format exists at present, the striking similarity between the trace formats used by existing applications leads us to believe that a concrete standard format can be specified with little controversy.

In addition, to be useful, a concrete trace format should present activity information at several levels of precision/granularity. The raw format is generated directly by the tool using its internal actions and parameters, but a more usable format would used abstract actions understandable regardless of the actual tool, such as "select", "pan", and "zoom". We therefore need to create a vocabulary of these abstract actions that all the PVMT(s) will recognize and visualize in an understandable way. Among these abstract actions, some virtual actions should also be defined. For example, it should be possible to annotate the provenance trace with screenshots to keep track of the visible state of the program at some key points. These annotations can be used by the PVMT to help navigate the provenance tree.

Managing provenance at a higher level also requires the PVMT to be able to talk back to applications and control them to navigate the provenance graph for undo/redo, or more generally jump to multiple steps of the graph. Therefore, in addition to a standard format, we need a standard service to manage the communication between program generating provenance traces and the PVMT(s). Again, this service will require some new information in the provenance trace. For example, the "undo" function returns to a particular state in the past, which may encompass several actions from the provenance trace. Marking the scope of "atomic" actions in the trace is essential to visualize a meaningful chunking of states that can be targets for navigation.

It is not yet clear if one level of abstraction is sufficient, but it is clear that a PVMT designed for debugging will want to show low-level trace actions, whereas one dedicated to support users might only show abstract ones. Summaries of large amounts of traces might need an event more abstract level; this will be investigated later.

4.2.2 Related Work

There has been substantial work in the domain of provenance capture, visualization, analysis, and reuse for visualization and visual analytics. VisTrails [1] showed the way with an integrated environment in Python, able to run complex analytics pipelines, keeping track of their development over time, editing the history of the evolution of the pipeline as a tree of pipelines evolutions, and allowing analysis to navigate in the provenance graph, replaying or continuing analyses. Other systems such as CZSaw [2] have built upon the experience of VisTrails to let graph visualization and analysis benefit from provenance management. More recently, Caleydo [4] has started to offer provenance capture and management for Web-based analytics systems. However, the landscape of data analytics has evolved and it is clear that analysts are always working with multiple applications and instrumenting only one will not capture the whole analytics process.

On the other side, there is a whole community working on provenance management. This community is related to Databases and Operating Systems, and has been running the International Provenance and Annotation Workshop (IPAW) [3] since 2002, as well as the "Workshop on the Theory and Practice of Provenance" (TAPP) Usenix conference until 2009.

The PROV standard has been published by the W3C[2] and later the ProvONE standard more suited for Scientific Dataflow systems[3].

Meanwhile, the visualization and visual analytics community have published many articles explaining systems and techniques for managing provenance and visualizing it. The LIVVIL workshop, organized at VIS 2014[4], has been the inspiration of the Dagstuhl seminar on Provenance and Logging for Sensemaking[5].

However, despite all these work related to provenance, the data analysis community is still left without practical solutions to match the promises showcased by VisTrails two decades ago. The community should get inspiration from all the research and experiments done so far to come out with a solution to manage provenance for data analysis, leading to sensemaking, storytelling, and hopefully many other outcomes demonstrated by VisTrails and its long list of related publications. Challenges There are a number of challenges involved in building such a system. We focus on the principal challenges involved in a standardized file format for logging provenance in a tool-agnostic way. Such a format, would need to capture the following in a scalable way: high-level events and low-level events that are common to many tools. The low-level events could be implemented in a tool-agnostic way whereas the high-level events would be application specific and would need to be defined in a way that extends the standard. There is the potential to not only define these events but provide a classification of such events. The standard should be easy to implement and extend so that new visualization tools can implement this standard.

4.2.3 Possible Solution and Opportunities

Transform RAW provenance to provenance standard Might be provided by the tool author In the future tools can directly provide provenance information that complies to the standard Provide a service that manages the standardized provenance Store/retrieve provenance information Execute actions from the provenance graph on application Result are multiple provenance trees from different applications that can be combined Provides a holistic view on the analysis process and might offer better insights Hence, new visualization approaches to mine the provenance information are required

Tasks and Users of the Provenance

The suggested approaches and capabilities were informed by a series of tasks and a group of three roles (i.e., users, designers, others) that could require those tasks. These tasks were: Remembering where one was in the exploration (primarily for users), explaining the exploration and findings for others (by the users for others), navigation during the exploration (users), meta-analysis of the exploration (for designers to see how the tool is used or others to see many traces), debugging/optimization of traces (for designers to retool their systems), and for reproduction of the results from the same data at different times (users and others), and using the trace as a template on different data (for users and others). For a deeper exploration of these needs, see Ragan [7].

[2] https://www.w3.org/TR/prov-overview/
[3] http://vcvcomputing.com/provone/provone.html
[4] https://livvil.github.io/workshop/
[5] https://www.dagstuhl.de/18462

What about the Vis?

While the discussion primarily focused on what is needed to support visualization tools for provenance, some time was given to what types of visualizations could be used. Traditionally, tree-like visualizations have been used to depict the branching trace structure (e.g., VisTrails [1]). However, each of the different roles we touched upon could use novel (or at least, not strictly tree-based) visualizations. Linear temporal sequences provide a clear sense of the direct history of the exploration. EventFlow-like compressed trees [6] lose the temporal aspects, but can highlight patterns. Graphs of the parameter similarity or exploration depth built upon metrics can also be considered [5]. When comparing or trying to apply different trees, means of overlaying or visually querying them are needed. Other novel approaches can be explored given the suggested framework. Extension: What activities could be supported using interactive visualizations? Delivering a visualization of the activities that have been carried out during some analysis session is only part of the story. With the appropriate API-based connections to the applications that generated the provenance records, a visualization can provide the means for a user to revisit the state of the session at any point that has been logged. This would mean, for example, re-establishing the full context of data selections and chart settings in an application such as Tableau, ideally including the full interaction history so that the user is faithfully transported back to the set of decision possibilities – including visualization adjustments, and undo/redo opportunities – that were available to the original analyst in that moment. One way for the developers of an application to support at least a limited form of such state revisiting would be through parameterized URLs, that are cheap to embed in the provenance stream and whose parameters will typically be tailored to the needs of the specific application. The more general solution, of course, would be for the application to accept from the visualization a sub-range of the entire provenance stream from the start of the recorded session up to the point of interest, and to use this stream to rebuild its state. Our hope is that application developers would see it as being in their interest to provide this maximal form of faithful reinstatement.

References

1 Louis Bavoil, Steven P. Callahan, Carlos Eduardo Scheidegger, Huy T. Vo, Patricia Crossno, Cláudio T. Silva, and Juliana Freire. Vistrails: Enabling interactive multiple-view visualizations. In *16th IEEE Visualization Conference, VIS 2005, Minneapolis, MN, USA, October 23-28, 2005*, pages 135–142. IEEE Computer Society, 2005.

2 Yingjie Victor Chen, Zhenyu Cheryl Qian, Robert Woodbury, John Dill, and Chris D. Shaw. Employing a parametric model for analytic provenance. *ACM Trans. Interact. Intell. Syst.*, 4(1):6:1–6:32, April 2014.

3 Juliana Freire, David Koop, and Luc Moreau, editors. *Provenance and Annotation of Data and Processes, Second International Provenance and Annotation Workshop, IPAW 2008, Salt Lake City, UT, USA, June 17-18, 2008. Revised Selected Papers*, volume 5272 of *Lecture Notes in Computer Science*. Springer, 2008.

4 Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Nicola Cosgrove, and Marc Streit. From visual exploration to storytelling and back again. *Computer Graphics Forum (EuroVis '16)*, 35(3):491–500, jun 2016.

5 T. J. Jankun-Kelly. Using visualization process graphs to improve visualization exploration. In Freire et al. [3], pages 78–91.

6 Megan Monroe, Rongjian Lan, Hanseung Lee, Catherine Plaisant, and Ben Shneiderman. Temporal event sequence simplification. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2227–2236, 2013.

7 Eric D. Ragan, Alex Endert, Jibonananda Sanyal, and Jian Chen. Characterizing provenance in visualization and data analysis: An organizational framework of provenance types and purposes. *IEEE Trans. Vis. Comput. Graph.*, 22(1):31–40, 2016.

4.3 A Novel Approach to Task Abstraction to Make Better Sense of Provenance Data

Christian Bors (Technische Universität Wien, AT), Simon Attfield (Middlesex University – London, GB), Leilani Battle (University of Maryland – College Park, US), Michelle Dowling (Virginia Polytechnic Institute – Blacksburg, US), Alex Endert (Georgia Institute of Technology – Atlanta, US), Steffen Koch (Universität Stuttgart, DE), Olga A. Kulyk (DEMCON – Enschede, NL), Robert S. Laramee (Swansea University, GB), Melanie Tory (Tableau Software – Palo Alto, US), and John Wenskovitch (Virginia Polytechnic Institute – Blacksburg, US)

In any exploratory activity, in which we include sensemaking with Visual Analytics tools [3], emerging and exploratory paths of interaction can lead to new understandings which can be at the same time surprising and unexpected. Indeed, the often inductive, exploratory quality of Visual Analytics is part of its premise. Hence, there are seldom set plans or procedures. In any exploration, it can be important to reconstruct what was done in order to fully interpret an output [1], to judge its validity, perhaps to see what other ground may be covered [4], or simply to learn [2]. Hence, there is value in recording and reconstructing provenance trails. Uninterpreted interaction logs, however, are typically detailed, low-level, and fail to provide ease of overview and rapid insight. Low level provenance data is limited by its lack of an organizing structure and hence a framework with which to make sense of this data. Providing a robust task abstraction framework (or interpretive structure) has the potential to provide the means of using low-level provenance data to construct higher-level task hierarchies (explicit tasks, and relationships between them), allowing users to interpret and gain the benefit from provenance data more easily.

We propose a conceptual task abstraction framework as an approach to enabling meaningful mapping between raw provenance trails and higher-level descriptions of tasks. We assume first a recorded trail of interaction. Next we assume that a task abstraction from this trail can be understood such that higher-level, more abstract task descriptions supervene over lower-level events or actions and provide a shorthand for sequences at the lower-level. Further, we assume that such relationships can be embedded in multiple layers, and hence a multi-leveled hierarchy. One premise of our approach, however, is the claim that abstracted descriptions are both interpreted and dependent on context. In many ways, we take the situated nature of language, and its translation of a series of contextually bound low-level phonemes into a higher-level message which can be further summarized and so on, as a metaphor for the interpretation of provenance data. As a consequence, we do not assume any fixed task hierarchies for a given string of low-level actions, but rely instead on the idea of interpretation as constituted from the construction of ad hoc hierarchies depending on context.

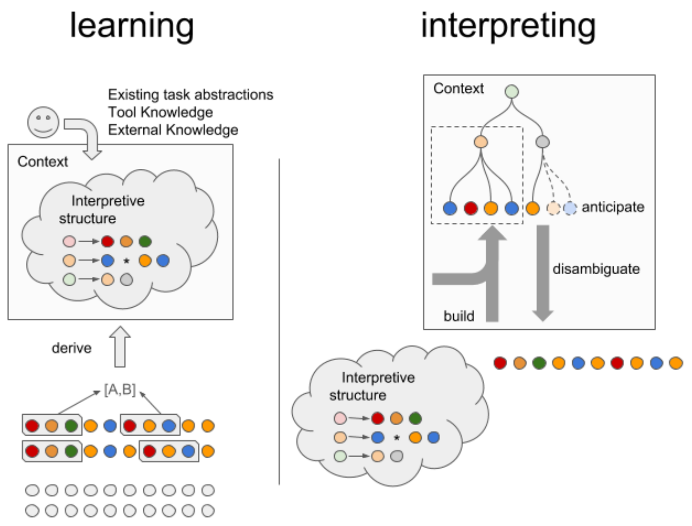


Figure 1 The three stages for task abstraction based on mapping low level provenance to a higher-level interpretive structure. While one objective of the task abstraction is building a structure for later use, the other is applying this structure for applying it to infer user goals.

4.3.1 A Task Abstraction Framework

Our framework can be divided into three major phases: Learn, Interpret, Use. Sequentially, the abstraction will impact these phases differently and changes in any of them will also propagate into the others. Additionally, we exemplify the levels of abstraction in the interpretation of a natural language.

4.3.1.1 Learn

The first phase is the creation of a specific hierarchical structure of *actions*, *tasks*, and *intents* (i.e., an interpretive structure). Such a structure can be created manually, either a priori or in an iterative manner, by deriving it from pre-defined interactions and tasks available in the system that the log traces are recorded from. Alternatively, the structure can be learned/acquired from a (large) number of logged sessions through various means, like sequence mining, process mining, ontology learning, etc.
Following the example of language, an analogy for the learn phase of task abstractions would be determining parts of speech that individual words belong to, based on their use within the context of the language (cf. Figure 2). Consecutively, when learning a language, single words are identified and combined into phrases, determining in which context the phrase can be used.

4.3.1.2 Interpret

The second phase comprises of the interpretation of logged data/provenance information by matching it to available structures determined during the Learn phase. Factors to consider for this are the expressiveness and flexibility of the interactive visualization tools provenance is obtained from, and the peculiarity of the user's task with respect to the concreteness. However, interpretation is also influenced by context that requires a formalization within the framework/structure.

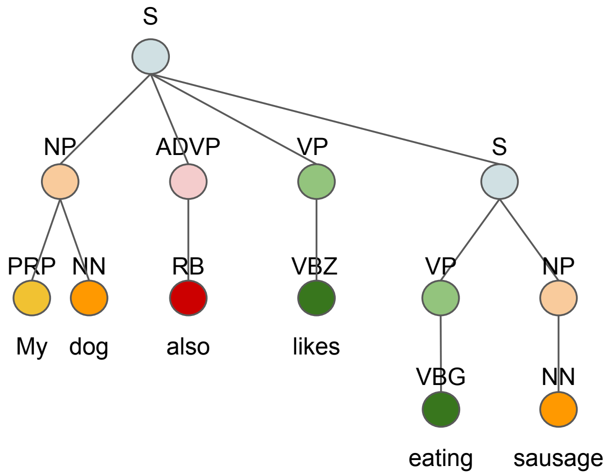


Figure 2 A parse tree exemplifying our natural language analogy. We parse words into logical phrases. From known phrases we can derive the meaning of unknown words or phrases in the context of the remaining sentence. We can furthermore apply this acquired knowledge to deduce further context (e.g., linking the phrases dog and sausage positively).

Continuing with our natural language analogy, we can use the phrases that we have so far learned to derive the meaning of new vocabulary or phrases that co-occur with known ones.

4.3.1.3 Use

Given a task hierarchy interpreted from an analysis session, there are different uses of the interpretive structure that can be presented to users (e.g., categories of guidance, automatic processing for presentation, ...). Based on the different goals of users, we have to apply the interpretive structure differently to account for the different expectations. For instance, a system could: (a) present templates for how to complete a task; (b) suggest next operations that guide the user to complete a task; (c) provide suggestions on what could be the next step as a decision making guidance; (d) optimize other aspects of the visual data exploration process based on the understanding of a user's process / behavior interpreted from the learned task abstraction; or (e) give an overview to a developer on how users actually conducted a task or tried completing it. Included in these complex UI and UX goals are also challenges with understanding when to present guidance and feedback to users in an appropriate way to minimize interruption and frustration.
Coming back to our example, reading sentences written in our natural language, we can exploit our prior experience with different language conventions and structures to try to deduce the meaning of unfamiliar words and phrases through their positioning, relative to known phrases. The structure, tenses, prepositions of the sentences change, based on how the information is intended to be conveyed.

4.3.2 The Role of Context and Uncertainty

Interpretation of user tasks based on low-level interactions is an imperfect process, often with incomplete or uncertain results. For example, some observed provenance records may not "fit" within the system's current interpretation of the user's behavior, and multiple interpretations may appear equally plausible for a given set of provenance records. Furthemore, context can influence interpretability and omit unlikely outcomes. Context can disambiguate outcomes by narrowing down possibilities based on usage (e.g., differences in individual users' tool expertise), environment variations (e.g., different designs of visualization systems, employed visual encodings), or the application/analysis domain. Additionally, we can incorporate notions of confidence or error into plausible interpretations and update these confidence measures as the user performs more interactions with the system, again drawing additional information from provenance. These measures can then be used to refine the system's understanding of what the user is trying to accomplish over time.

A user may repeat some interaction patterns in their future analysis, but they could also start exhibiting new patterns of analysis. As such, the system should be able to adapt by continuing to refine the interpretive structure given additional input. An important consideration for an active learning approach is the matter of temporal context: when should a system adapt the interface because of learning? When should a system remain the same? Previous work on adaptive menus highlights this problem: Users often rely on consistency to enhance performance, and thus user performance can suffer when consistency is ignored. The Show Me automated presentation feature in Tableau [5] is an example of how consistent recommendations help users to reason about and utilize recommendations efficiently. Alternatively, the uncertainty of outcomes can be actively communicated to the user to clarify expected outcomes and improve accuracy of the interpretive structure.

We are aware of existing research to address aspects of the challenges outlined above, but we see shortcomings in combining them into a singular organizing framework that leverages these approaches. For example, how to manage the uncertainty of interpretation and inherent variability in analysis sessions across user expertise, system design, etc., is currently unclear. Thus, there exist exciting opportunities and challenges along these lines of research that can advance our understanding in how to learn useful structures from provenance. Possible interpretations of our conceptual framework could be implemented by probabilistic parsing, building grammars (NLP), or machine learning (cf. [6, 7, 8, 9, 10]).

References

1 Bodesinsky, Peter, et al. "Exploration and assessment of event data." Proceedings of EuroVis Workshop on Visual Analytics. 2015.

2 E. T. Brown et al., "Finding Waldo: Learning about Users from their Interactions," in IEEE Transactions on Visualization and Computer Graphics, vol. 20, no. 12, pp. 1663-1672, 31 Dec. 2014.

3 Pirolli, Peter, and Stuart Card. "The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis." Proceedings of International Conference on Intelligence Analysis. Vol. 5. 2005.

4 H. Stitz, S. Gratzl, H. Piringer, T. Zichner, und M. Streit, "KnowledgePearls: Provenance-Based Visualization Retrieval", IEEE Transactions on Visualization and Computer Graphics, p. 1–1, 2018.

5 Tableau Software:. URL: https://www.tableau.com/. Accessed: Jan, 2019.

6 I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, Data Mining, Fourth Edition: Practical Machine Learning Tools and Techniques, 4th ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2016.

7 L. Bradel, C. North, L. House, S. Leman, Multi-model semantic interaction for text analytics, in: 2014 IEEE Conference on Visual Analytics Science and Technology (VAST), 2014, pp. 163–172. doi:10.1109/VAST.2014.7042492.

8 J. Z. Self, R. K. Vinayagam, J. T. Fry, C. North, Bridging the gap between user intention and model parameters for human-in-the-loop data analytics, in: Proceedings of the

Workshop on Human-In-the-Loop Data Analytics, HILDA '16, ACM, New York, NY, USA, 2016, pp. 3:1–3:6.doi:10.1145/2939502.2939505.
9 J. Wenskovitch, C. North, Observation-level interaction with clustering and dimension reduction algorithms, in: Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics, HILDA'17, ACM, New York, NY, USA, 2017, pp. 14:1–14:6. doi:10.1145/3077257.3077259.
10 M. Dowling, J. Wenskovitch, J. T. Fry, S. Leman, L. House, C. North, Sirius: Dual, symmetric, interactive dimension reductions, IEEE Transactions on Visualization and Computer Graphics 25 (1) (2019) 172–182. doi:10.1109/TVCG.2018.2865047.

4.4 Machine Learning and Provenance in Visual Analytics

Christopher Collins (UOIT – Oshawa, CA), Sara Alspaugh (Splunk Inc. – San Francisco, US), Remco Chang (Tufts University – Medford, US), Phong H. Nguyen (City – University of London, GB), and Kai Xu (Middlesex University – London, GB)

Provenance and machine learning can be mutually beneficial in two distinct ways: applying machine learning on provenance data to carry out tasks, and using provenance data from machine learning processes to improve the machine learning systems.

In both of these scenarios, the problem of chunking or segmenting event logs is relevant. In particular, we are interested in the opportunities for using interaction records, or logs of user actions, in analytic systems. Interaction records are different from other types of events that a system may record, such as receipt of data from an asynchronous process, or exceptions in network communication. Interaction records relate to task analysis in visual analytics in that a series of low-level interactions may represent a higher level analytic process [5]. For example, hovering on an item, panning the screen, then hovering another item may represent an explore action. We are interested in determining the appropriate chunking of low-level actions, such that they can be used in the variety of applications described below.

4.4.1 Machine Learning on Provenance Data

Applications of machine learning on provenance data includes using machine learning to summarize event sequences to tell stories of analytic processes. For example, machine learning could be used to curate provenance data to abstract sequences into higher level tasks to describe the analysis. Machine learning could be used on provenance logs to identify canonical analytic workflows and generalize them through fuzzy matching. This could be useful for creating data-driven descriptive theory about real-world analytics work, to compare to idealized models of visual analytics [3].

If models of analytic workflows can be created from event logs, machine learning might then be helpful in mixed-initiative interfaces providing guidance to analysts and users of visualization systems. Such guidance may come in the form of recommended next steps in an analysis, based on learned effective analytic sequences (data support) or on learned usage patterns for specific analytic software (interface support).

Furthermore, interface and analytic process customization may be possible if machine learning on provenance logs can be leveraged to draw conclusions about user characteristics, both long-term traits (e.g., personality factors, locus of control, level of analytic expertise) and short-term transient traits (level of stress, level of cognitive load). If these user characteristics could be determined to a reasonable level of confidence, a system may make recommendations of analytic next steps, or interface support, which may be specific to the user.

4.4.2 Provenance for Machine Learning

The second way that provenance and machine learning can come together in visual analytics is in generating provenance data during the generation of machine learning models, essentially to support model creation, refinement, and deployment. For example, it could be useful to record the steps to develop a model, the parameter settings used both in experiments and in the final model, the features used to train the model, and the steps of model validation.

Furthermore, provenance of machine learning models could be useful in the creation of explainable machine learning visualizations. A classifier, for example, may reveal the features and input data examples which are most influential in a given classification trial. Another example would be a visualization which reveals the sensitivity of the classifier to the specific value of parameters driving the model.

4.4.3 Case Study: Log Event Chunking

A fundamental problem underpinning many applications of visual analytics provenance is summarizing the high-volume, low-level event data into a smaller, more cognitively manageable, and semantically meaningful set of chunks, which we will refer to as tasks [1]. Given a sequence of interaction events from a visual analytics system, the problem is to group, segment, or chunk the events into subsets corresponding to higher-level tasks (see Figure 3). Here we focus on the problem of automated or semi-automated chunking and leave the problem of labeling or identifying the higher-level meaning of the resulting chunks to future work.

There are several challenging aspects of the chunking problem:

- The notion of higher-level tasks (or goals or intent) is not precise or may not be known a priori, making top-down aggregation difficult.
- Tasks may be hierarchical, with tasks being themselves part of higher-level tasks or consisting of subtasks, with the steps that correspond to the interaction events in the log being just the lowest level.
- Tasks may be interleaving (e.g., the user starts a task, interrupts it to switch to a different task, then switches back to the original task) or overlapping, so that the boundaries between tasks may be fuzzy, and tasks may consist of non-adjacent events.

4.4.3.1 Possible Approaches

One potential formulation of the problem is to label each event with the chunk that it belongs to. However, this solution would suggest stricter boundaries between chunks and more certainty in the chunk assignments than is present in reality, given the inherent ambiguity in tasks, as described above. A way of remedying this is to augment each event's chunk assignment with a probability that corresponds to the confidence of that assignment. Going one step further, providing the probability, for each event, that the event belongs to each cluster, would provide even more information and would more accurately reflect the characteristic ambiguity described above.

With this formulation we can consider whether to use a supervised, unsupervised, or semi-supervised machine learning approach.

Event	Group
0	0
1	0
2	1
3	1
4	1
5	1
6	1
7	2
8	2
9	2

(a)

Event	Group	Confidence
0	0	1.00
1	0	0.66
2	1	0.32
3	1	0.58
4	1	0.93
5	1	0.83
6	1	0.67
7	2	0.49
8	2	0.73
9	2	0.76

(b)

Event	Group 0	Group 1	Group 2	...
0	0.38	0.00	0.00	...
1	0.56	0.00	0.00	...
2	0.06	0.02	0.01	...
3	0.00	0.18	0.03	...
4	0.00	0.43	0.01	...
5	0.00	0.23	0.00	...
6	0.00	0.67	0.16	...
7	0.00	0.14	0.22	...
8	0.00	0.01	0.34	...
9	0.00	0.00	0.23	...

(c)

Figure 3 Three ways of grouping events. In (a) events are each assigned to a group. In (b), these assignments are augmented with a confidence level, to reflect the inherent ambiguity of the task. In (c), events are given a probability of belonging to each group, allowing for overlapping groups.

- For supervised algorithms, we would require a set of chunk labels. However, there is no natural source of these labels, and manually generating the many labels that would be needed is tedious. Moreover, labelers will be inconsistent with one another for the reasons described above regarding the inherent ambiguity of tasks, making it important to obtain labels from a sufficiently broad sample of labelers.
- For unsupervised algorithms, we dispense with the need for labels, however, the challenge is to develop a meaningful similarity or distance measure such that events in the same chunk register as more similar or closer together and events in different chunks, further away.
- Semi-supervised learning can combine the strengths of both of these approaches by using both a similarity measure and selective use of labels to reduce ambiguity in borderline cases; in this way fewer labels are needed, and weaknesses of the similarity measure can be compensated for. However, semi-supervised learning techniques can be sensitivity to inconsistent labels (e.g. a labeler providing different labels for the same data item during a session), making the outcomes less repeatable or stable. Another challenge would be in designing a label solicitation interface that provided enough context to the labeler.

4.4.4 Potential Algorithms

Given a machine learning approach (supervised, unsupervised, or semi-supervised), we suggest a number of possible algorithms that could be suitable to perform the segmentation of interaction logs. This list is not meant to be exhaustive, but to provide the initial seeding to the design and adoption of techniques from other communities.

- Frequent sequential patterns
- Bioinformatics sequence algorithms
- Regex pattern generation
- Program synthesis (e.g. use of domain specific language)

- Fuzzy clustering or classification
- Crowdsourcing
- Bayesian modeling

4.4.5 Provenance Data to Machine Learning Features

Determining what features of interaction records and system events may be useful in the chunking requires some feature engineering. Many interactive systems are instrumented for some sort of event logging, but in the following section we enumerate a variety of feature types which could be added to instrumented software to provide a better set of features for chunking. Our list makes use of "basic features" in general interactive systems as well as features that are specific to visualization and visual analytics systems.

4.4.5.1 Basic Features

Interaction records of software, including visualization systems, often record timestamped low-level events such as mouse movements, clicks, keys typed, use of interface functions such as undo, redo, and buttons. Based on these low-level events, additional information can be derived. For example, the velocity of the mouse movement, the fact that a user performed a selection bounding box (through click-and-drag).

These particular features have to be carefully curated so that the logs collect information useful for machine learning on provenance data, but do not contain or reveal personally identifiable information. For example the velocity of typing, or the classes of keys (letters/numbers) or words (stop words/content words) may be recorded without recording the actual content typed into the logs.

4.4.5.2 Visualization Specific Features

Event records specific to visualization applications include, most importantly, those that relate to the data. For example, the use of filter tools to filter a dataset, including the filter parameters, would be a feature to log in the interaction records. In addition, the actual data visible on screen at any moment would be useful to log either in association with all events, or whenever the visible data changes. However, logging all visible data may introduce storage considerations if the dataset is large. Some systems may have succinct ways to describe the dataset (e.g. hashing, lists of constraints and filters, etc.). It may also be possible to capture this information through descriptive features of the data, such as mean, standard deviation, presence of outliers, distribution of node degree (for graphs), etc. Or, a low-resolution screenshot of the visualization state may suffice as a proxy for the list of visible data items. Otherwise, if the entire visible dataset must be recorded, we leave the challenge of addressing the storage problem to future work.

In a coordinated multi-view visualization system, which visualization panel is currently active may be appropriate to place in the log. Along with the active panel, lower-level focus events could be logged, such as mouse hover on visual items (data items and visualization features such as the axes). Depending on the availability of additional interface hardware, such as eye-tracking, dwell time on visual items could also be logged. Analytic actions such as annotation (specific to data items) and note-taking (general about a visualization state) should also be recorded in the log.

4.4.5.3 Novel Features to Log

Moving forward from the more traditional interaction records, machine learning, and specifically chunking, on provenance logs may be more successful if we log new types of features specific to analytics systems. First, we may encode the system state in a feature vector, or potentially reduce the system state and data state to a point [4, 2] which can be compared to other points to determine a distance from previous states.

While recording the data which is on screen and also which has been explicitly of interest through hover or selection events can be useful, it may be possible to discover features in the logs which are more closely tied to the task of analytics. For example, statistical relations of focused (hovered) data to other data could indicate high-level interest patterns. Are outliers being hovered? Are items of interest in a cluster? Another feature could be to track the number of recently visited items affected by an operation. Does the filter remove recently hovered items? Then the filter action is probably part of a sequence. Image-based measures targeted toward visualization could also be informative for discovering important moments in analysis, such as the amount of change in the displayed image (e.g. image-based or model of perceived change). Similarly, back-end logs such as data load operations could indicate major changes in direction in an analysis process.

4.4.6 Summary

In this report, we describe the mutual benefit between provenance and machine learning, and focus on a particular problem that is of relevance for both – chunking of event data. We discuss possible machine learning approaches and list some algorithms that have potential in addressing the problem. More concretely, we brainstorm novel features, besides standard keyboard and mouse events, that could be considered in future chunking solutions.

References

1 Sara Alspaugh. *Understanding Data Analysis Activity via Log Analysis*. PhD thesis, EECS Department, University of California, Berkeley, Aug 2017.
2 Eli T Brown, Sriram Yarlagadda, Kristin A Cook, Remco Chang, and Alex Endert. Modelspace: Visualizing the trails of data models in visual analytics systems. In *Workshop on Machine Learning from User Interaction for Visualization and Analytics*, pages 1–11, 2018.
3 Peter Pirolli and Stuart Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. pages 2–4, 2005.
4 S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk. Reducing snapshots to points: A visual analytics approach to dynamic network exploration. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):1–10, Jan 2016.
5 K. Xu, S. Attfield, T. J. Jankun-Kelly, A. Wheat, P. H. Nguyen, and N. Selvaraj. Analytic provenance for sensemaking: A research agenda. *IEEE Computer Graphics and Applications*, 35(3):56–64, May 2015.

4.5 Storytelling & User Intent

Robert Kosara (Tableau Software – Seattle, US), Ali Sarvghad (University of Massachusetts – Amherst, US), William Wong (Middlesex University – London, GB), and Michelle X. Zhou (Juji Inc. – Saratoga, US)

While visualization research tends to assume that the exploration and analysis of the data come before the presentation is designed, the analysis is often motivated and guided by an assumed story, hunch, or hypothesis. Analysts might go back and forth between story construction and analysis many times along the way, discovering new questions to ask as they create the story, finding supporting or contradicting evidence in their analysis, etc.

4.5.1 Levels of Intent

A central challenge in understanding user intent and incorporating it into the storytelling process is the mapping between low-level interaction events and higher-level goals of an analysis. We described four levels of abstraction in user intent: Operation: The lowest-level of interaction event (mouse click, selection event) in a visual interface Task: The activity of which an operation is a part (e.g., a selection event coupled with drag-and-drop may be part of a "categorize" task) Goal: The reason a task is performed Intent: The highest-level abstraction for the purpose behind an analysis (or analysis subsequence)

Operation and Task together comprise what an analyst does. Goal and Intent together comprise why. A complete taxonomy of intent levels should borrow from the human factors literature on hierarchical task analysis [4].

Stories can communicate the intent behind an analysis (by representing the choices and interpretations an analyst made) to support interpretation by the analysis consumer. Stories can also drive intent in that they provide a template for how a data-driven argument was constructed and allow an analysis producer to reuse a story structure with new data.

4.5.2 From Provenance to Story

Provenance can support the construction of the presentation by surfacing states in the analysis that are likely of interest. Heuristics for selection include node centrality of the state within the provenance graph, repetition of states (a state that was visited more often is more likely to be of importance), amount of change from the previous state, explicit user tags, etc.

Once potentially useful states have been identified, the user can select which ones to include in the story and insert them into a story structure. At this point, the type of story or argument can be used to select pre-defined story templates according to a number of classic rhetorical structures, such as persuasion, argument, analysis, or exposition. More corresponding structures and prototypes need to be identified, but recent work has shown at least one distinct and reusable pattern for data-based arguments in news graphics [2].

Story structure may be specified ahead of time via a template but may also emerge over time as an analyst explores data. Emergent stories themselves have provenance, which reflects the evolution of an analyst's argument or explanation over time.

Grice's *conversational maxims* [3] describe other desirable properties of the structure and content of stories. The maxims of quantity, quality, relation, and manner provide guidelines for coherent conversation (such as the asynchronous communication between an analysis producer and consumer) that, when violated, introduce ambiguity or other rhetorical flaws. Storytelling interfaces may enforce these maxims as an aid to constructing strong data-based arguments.

4.5.3 Structuring Arguments and the Circle

Toulmin provides an explanation of how arguments may be set out so that claims that are made have a basis, and that evidence can be provided to support those claims. Toulmin's argumentation model [5] is a useful way of thinking about a generic structure that can be adapted by a variety of different approaches to representing that structure for purposes for communicating and presenting information. The CFO model [2] is one such useful approach to organizing materials for presentation. The Toulmin model is also useful as it includes other factors that could be considered when we collect data and results of analysis to communicate our findings, e.g. the concept of warrants is the authority on which claims are made. These can take the form of higher order assumptions upon which our explanation is based.

One approach taken based on some of these ideas has been reported in Groenwald, et al. [1]. The approach to storytelling in this example is to construct the story made up of data elements and results of analyses into unique sequences that help to explain what the analyst has observed in the data. The ideas generated by the initial sequence of data can be used to tell communicate a story – an explanatory narrative. This initial understanding provides the basis for formulating an early stage tentative hypothesis – a hunch – that can guide further inquiry, and more data collected to prove or disprove the hypothesis.

This process informs the communicator/analyst, develops new or elaborates his understanding, which enables him to seek further data or analyses, question its findings, and to even reframe his conceptualisation of the problem and the way he intends to communicate the message [6].

4.5.4 Next Steps

We plan to further investigate how provenance data and visualisations might be used to support the process of communicating the outcomes from analyses:
- Use provenance trails to highlight nodes of interest
- Select nodes for story
- Assemble the nodes into rhetoric structure
- Add a narrative to create an explanation

Logging user intent provides a promising starting point for a wealth of research into building more effective presentations and stories.

References

1 Celeste Groenewald, Simon Attfield, Peter Passmore, B. L. William Wong, Nadeem Qazi, and Neesha Kodagoda. A descriptive, practical, hybrid argumentation model to assist with the formulation of defensible assessments in uncertain sense-making environments: an initial evaluation. *Cognition, Technology & Work*, 20(4):529–542, Nov 2018.
2 Robert Kosara. An argument structure for data stories. In *Short Paper Proceedings of the Eurographics/IEEE VGTC Symposium on Visualization (EuroVis)*, 2017.
3 Stephen Neale. Paul grice and the philosophy of language. *Linguistics and philosophy*, 15(5):509–559, 1992.
4 Neville A Stanton. Hierarchical task analysis: Developments, applications, and extensions. *Applied ergonomics*, 37(1):55–79, 2006.
5 Stephen Toulmin. The uses of argument. 2003.
6 B. L. W. Wong, P. Seidler, N. Kodagoda, and C. Rooney. *Supporting variability in criminal intelligence analysis: From expert intuition to critical and rigorous analysis*, pages 1–11. Springer International Publishing AG.

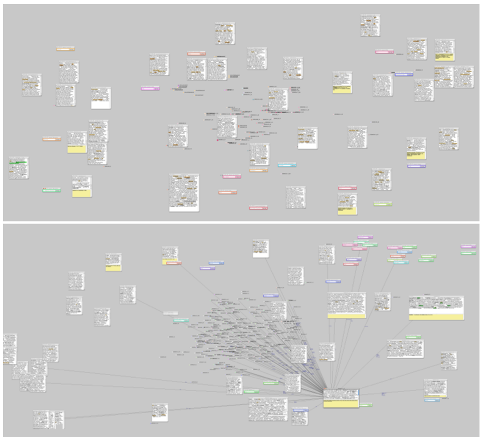
4.6 User Modeling & Intent

John Wenskovitch (Virginia Polytechnic Institute – Blacksburg, US), Remco Chang (Tufts University – Medford, US), Christopher Collins (UOIT – Oshawa, CA), Michelle Dowling (Virginia Polytechnic Institute – Blacksburg, US), Alex Endert (Georgia Institute of Technology – Atlanta, US), Phong H. Nguyen (City – University of London, GB), Kai Xu (Middlesex University – London, GB), and Michelle X. Zhou (Juji Inc. – Saratoga, US)

4.6.1 Motivation

Inferring user intent from interactions is an active problem in visual analytics. Current solutions to this problem in the realm of interactive projections make use of online learning, inferring the intent of individual interactions to incrementally build a user interest model.

For an example, consider a visual text analytics system. In this system, a collection of documents are laid out spatially using a distance similarity metaphor – similar documents are positioned close together, while dissimilar documents are positioned further apart. The system responds to user interactions by updating an interest model (if a user places two documents close together, the system identifies what makes these documents similar and thereby learns what factors the user is interested in), and may forage for new relevant information based upon what the system learned. An example of this foraging can be found in the accompanying figure.



However, this system may need to respond differently depending on the skill level and traits of the users. For example:

- Users could be experienced system users (e.g., power users are ready to be overwhelmed with new information that they can easily process) or they could be novices (e.g., let's not give them too much information at once until they are comfortable with the system).
- Users could be intellectually curious (e.g., open to exploring a number of conflicting hypotheses) or they could want hand-holding (e.g., explore fewer possibilities).

An open question is how the system can obtain or learn this information about its users: whether it comes from monitoring and interpreting user behavior, or is detected from a survey before using the system, or another means entirely.
Our group's discussion on Tuesday sought to explore some of the options in this space.

4.6.2 Modeling Characteristics vs. Modeling Intent

To begin, we developed two separate lists: (1) a list of user characteristics that could affect the system preferences of a user, and (2) a list of goals that a system designer may want to include in their software. These are not necessarily complete lists, but they gave us a starting place to work from. These lists follow:

User Characteristics
1. Independence
2. Expertise
3. Level of stress
4. Adaptability
5. Need for control
6. Intellectual curiosity
7. Dark Triad
8. Locus of control
9. Tolerance of uncertainty

System/Designer/User Goals
1. Avoid biases
2. Information and analysis coverage
3. Efficiency
4. Detect if user needs help/reassurance
5. Understand how users use a system (post-hoc)
6. Storytelling, report generation
7. Delivering different data (in different ways) for different types of users
8. User happiness ("the system gets me")
9. Increase user awareness of the implication of their behaviors/analysis (e.g., ethics, biases, assumptions)

It is worth noting that the user characteristics listed to the left have a variety of temporal spans. For example, a user's level of stress could fluctuate during the time spent using the system, whereas expertise is more constant but dependent on the subject area of analysis, and intellectual curiosity is a still more constant behavior.

The mapping between these characteristics and goals is also a bit nebulous. For example, the (3) level of stress user characteristic could map to a number of goals. Most obvious is (4) detecting if a user needs help of reassurance; if the stress is due to issues with understanding the system or the data, the system may wish to reduce the rate of information flow to a simpler level, or perhaps may pop up some tips for how to continue with the analysis. Reducing the rate of information flow therefore also effects (2) information and analysis coverage, while redirecting the analysis path effects (1) avoid biases and (7) delivering data for different types of users (as a stressed user is different from a non-stressed user). Analysis of how a user responds to this reduction in information flow further triggers goal (5) and possibly also (8).

These interventions in response to user behaviors could either be handled by the front-end or the back-end of a system. There is, of course, a tradeoff inherent in these intervention options:
- **Front-end:** There could be different levels of visibility for the intervention (a spectrum from subtle hints to locking out some system functionality), but each necessitates an interruption to the user's workflow.
- **Back-end:** Low risk and no obvious interruption, but failure of a predicted intervention is not visible to the user.

We also discussed how different user models can be classified and characterized. We decided that there are three types of predictive user models:
1. **Understand intent:** These user models determine the interests of a user based on their interactions. This could serve to adapt the behavior of the system as suggested above (e.g., change the rate or variety of information flow in response to learned characteristics).
2. **Predict future user actions:** These user models predict future interactions for users during their analysis process (e.g., to increase system response time by preprocessing future analysis or to suggest analysis routes to a user).
3. **Classify user characteristics:** These user models can assist with post-hoc analysis of system behavior for future versions (e.g., learn what types of users most often use the system so that menus and toolbars can be organized for ease of access to common features), and can also assist the other two user model types.

4.6.3 Using Semantic Interaction to Model Characteristics

Following our discussion on possible characteristics and system goals that might be included in future systems, we turned our attention to how a system could obtain this information. The example of giving a user a pre-survey to understand the user is useful but also trivial and potentially misleading (e.g., users don't want to admit their lack of expertise or current stress level). Our discussion instead focused on if the Semantic Interaction paradigm, intended to infer user intent, could be adapted to learn user characteristics. For example:

Semantic Interaction	**Modeling Characteristics**
1. Capture an interaction	1. Capture an interaction
2. Interpret intent	2. Interpret *characteristics*
3. Update the model	3. Update the model

Step 2 on the right is the challenge that needs to be addressed in order to build user models that learn these user characteristics.

4.6.4 Mapping Interactions to Intent and Characteristics

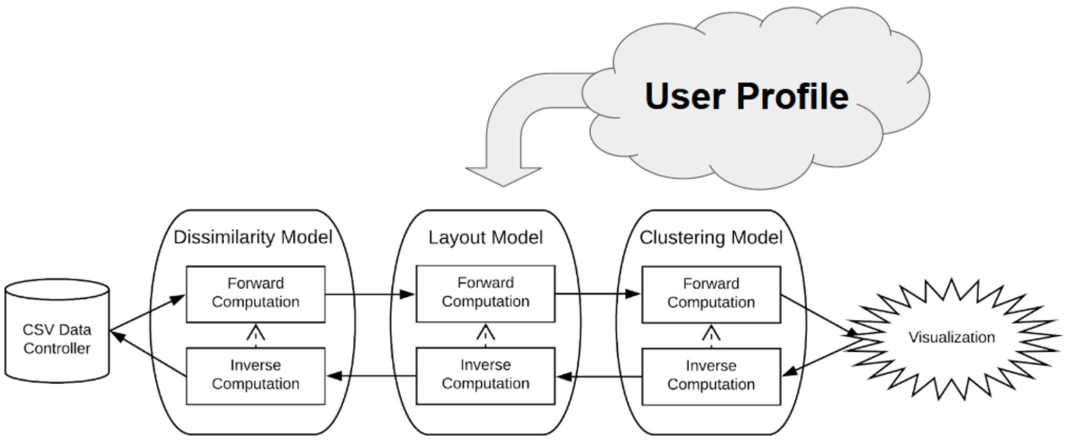
A necessary step in both semantic interaction and in modeling these user characteristics is the inference phase. Without clear-cut rules, inferring the intent of a user based only upon interactions is a clear challenge. In discussing this problem, we structured our discussion by cardinality of both interaction and intent:

- **One interaction implies one intent:** This is the trivial case, and one example would be the direct manipulation of a control widget (click the button to submit the form).
- **Many interactions imply one intent:** This case can be thought of as flexible UI design. For example, there are many different interactions and interaction sequences that can be used to bold text in Microsoft Word. Though the intent is the same, a separate set of interactions may be supported in OpenOffice to achieve the same goal.
- **One interaction implies many intents:** This case is an underspecified interaction: one interaction from a user could be inferred in many different ways. We discussed five different ways to disambiguate this uncertainty in interaction:
  1. **Ask the user to disambiguate:** A simple case of popping up a "What did you intend this interaction to do?" message.
  2. **User provenance to infer the user's intent:** Given a past sequence of interactions, we could attempt to guess at the most likely intent of an interaction.
  3. **Get more examples from the user:** An ambiguous interaction may not alter the underlying model until it is performed many times, or until the user broadens the scope of the interaction (e.g., flash-fill in Excel).
  4. **User data to infer:** Similar to (2), but using the dataset under consideration rather than the past interactions of the user.
  5. **Use user characteristics to infer:** Similar to (2) and (4), but using what the system has learned about the behavior and characteristics of the user to disambiguate.
- **Many interactions imply many intents:** This is the most interesting case, because the obvious (to our discussion) interpretation is flexible interaction design. The user could perform any gesture or interaction, and the system could use a set of meta-rules (or user behavior, or provenance, or any of the above) to infer the user's intent for the interaction.

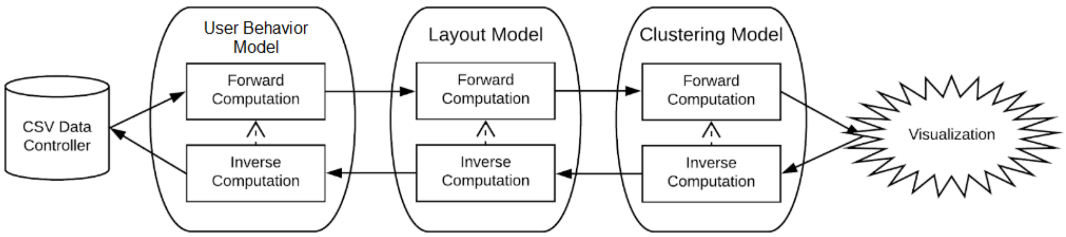
Though we did not discuss this in detail, a similar cardinality breakdown could be used to map interaction to learned user behaviors. For example, shaking the mouse in frustration could easily map to an increase in the inferred frustration level of a user.

4.6.5 How This Could Work / Building a System

After inferring user characteristics, we want to know how best to use these to influence and adapt existing and new systems. Our initial solution was to keep user characteristics as a separate set of parameters that can influence each of the individual components of a system. For example:



However, it is also possible to build this user model into the system as a component that is only processed when it is necessary to reference.



4.6.6 Other Issues Discussed

The discussion provided above is a shortest path through relevant topics that we discussed, but a number of other subjects were briefly discussed in side conversations and parallel threads. A quick summary of these discussions are included here:

- What signals can we and should we collect? There are a lot of signals, and also a lot of garbage. How can we tell the signal from the noise?
- What role does training have on user intent expression?
- Data scarcity is a problem. How many user logs do you have? You don't have tens of thousands of users to evaluate. Can we built a model on a single user while they're learning about a system for the first time?
- What is the purpose of this modeling in general? Do we want to build a single-use model, or a persistent model? This is a key difference between post-hoc and real-time learning.
- How do we avoid being Clippy? Can we always get it right, and without being disruptive?

- Struggle to balance instruction vs. freedom to explore your interfaces.
- Capturing low-level parameters is much more tractable than predicting the user's intent.
- Can we specify how many levels of intent we want? Or just low-level and high-level?

Participants

- Sara Alspaugh
Splunk Inc. – San Francisco, US
- Daniel Archambault
Swansea University, GB
- Simon Attfield
Middlesex University – London, GB
- Leilani Battle
University of Maryland – College Park, US
- Christian Bors
Technische Universität Wien, AT
- Remco Chang
Tufts University – Medford, US
- Christopher Collins
UOIT – Oshawa, CA
- Michelle Dowling
Virginia Polytechnic Institute – Blacksburg, US
- Alex Endert
Georgia Institute of Technology – Atlanta, US
- Jean-Daniel Fekete
INRIA Saclay – Orsay, FR
- Melanie Herschel
Universität Stuttgart, DE
- T. J. Jankun-Kelly
Mississippi State University, US
- Andreas Kerren
Linnaeus University – Växjö, SE
- Steffen Koch
Universität Stuttgart, DE
- Robert Kosara
Tableau Software – Seattle, US
- Olga A. Kulyk
DEMCON – Enschede, NL
- Robert S. Laramee
Swansea University, GB
- Sérgio Lifschitz
PUC – Rio de Janeiro, BR
- Aran Lunzer
OS Vision – Los Angeles, US
- Phong H. Nguyen
City – University of London, GB
- William Pike
Pacific Northwest National Lab. – Richland, US
- Ali Sarvghad
University of Massachusetts – Amherst, US
- Claudio T. Silva
New York University, US
- Holger Stitz
Johannes Kepler Universität Linz, AT
- Melanie Tory
Tableau Software – Palo Alto, US
- John Wenskovitch
Virginia Polytechnic Institute – Blacksburg, US
- William Wong
Middlesex University – London, GB
- Kai Xu
Middlesex University – London, GB
- Michelle X. Zhou
Juji Inc. – Saratoga, US