A Provenance Task Abstraction Framework

Christian Bors
TU Wien

John Wenskovitch
Virginia Tech

Michelle Dowling
Virginia Tech

Simon Attfield
Middlesex University

Leilani Battle
University of Maryland

Alex Endert
Georgia Tech

Olga Kulyk
InnoValor

Robert S. Laramee
Swansea University

***Abstract*—Visual analytics tools integrate provenance recording to externalize analytic processes or user insights. Provenance can be captured on varying levels of detail, and in turn activities can be characterized from different granularities. However, current approaches do not support inferring activities that can only be characterized across multiple levels of provenance. We propose a task abstraction framework that consists of a three stage approach, composed of 1) initializing a provenance task hierarchy, 2) parsing the provenance hierarchy by using an abstraction mapping mechanism, and 3) leveraging the task hierarchy in an analytical tool. Furthermore, we identify implications to accommodate iterative refinement, context, variability, and uncertainty during all stages of the framework. We describe a use case which exemplifies our abstraction framework, demonstrating how context can influence the provenance hierarchy to support analysis. The article concludes with an agenda, raising and discussing challenges that need to be considered for successfully implementing such a framework.**

Visual analytics tools support exploration and reasoning over relatively large datasets using visual representations of data for rapid, incremental interaction. With an emphasis on enabling analytical reasoning, Visual Analytics places the user in the loop of analysis. Within the field, there has been increasing interest in the idea of recording both data exploration and accompanied human reasoning. Referred to as insight provenance[1] or analytic provenance,[2] such

Digital Object Identifier 10.1109/MCG.2019.2945720
Date of publication 10 October 2019; date of current version 1 November 2019.

information presents opportunities for presenting interaction suggestions to the analyst, retrospectively auditing the quality and coverage of existing analyses, tracing the origins of insights and assumptions, supporting collaboration between analysts, or simply providing an analyst with a record as a source of reflection and planning.

Gotz and Zhou[1] argue that scalable approaches to representing complex analyses are likely to involve the automated capture of low-level interaction histories. However, processing such extensive and detailed histories into hierarchical representations from which analysts might derive meaning presents a further step. In considering this problem, they point out that activities can be characterized at multiple levels of granularity, and, hence, they frame the problem as one of inferring (semantically richer) higher level tasks from large numbers of lower level actions, i.e., a problem of task abstraction. For such abstraction to be automated, one must hypothesize a mechanism by which low-level operations or actions can be inferentially mapped to higher level intents.

In this article, we propose our vision for how this problem could be addressed in the future. Our proposed approach involves analyzing low-level events into higher level actions and activities. We posit that by considering provenance and the nature of task abstraction more generally, analytical systems can better model and leverage interaction provenance. In the Task Abstraction Framework section, we describe our proposed framework, beginning with assumptions or constraints on what we see as good solution. These include the idea that high-level actions can be realized in multiple ways, and that the role of low-level actions depends on the context of those actions. As a result, mapping using *a priori* task hierarchies would be overly simplistic. We propose what we refer to as an abstraction mapping mechanism (AMM) to enable *ad hoc* parsing of interaction streams into abstract tasks and inferring upcoming actions.

We present the approach as a three-stage framework: a) *Initializing*—developing a mapping mechanism (rules or model); b) *Parsing*—applying the mechanism to a given interaction stream to form an interpretation; and, c) *Leveraging*—applying the resulting interpretation in some useful way. The proposed framework design supports variability allowing the integration of context (e.g., in the form of externalized domain knowledge) and iterative improvement. When parsing and leveraging the framework in a live scenario, users could be prompted if the recommended actions were actually useful, and feedback could contribute in changes of AMM task probabilities. In online hierarchy parsing and leveraging scenarios, users might be prompted if the recommended actions were actually useful, and feedback could contribute in changes of AMM task probabilities. We conclude by motivating an agenda that points out the shortcomings of current approaches toward the development of such a framework, elaborating research needed to accomplish it.

In summary, we make the following contributions:

1) We further clarify and define the problem of inferring the users reasoning process as a hierarchy of the user's data analysis tasks and subtasks.
2) We propose a conceptual framework for inferring the user's data analysis tasks from log data and relevant metadata, involving three stages: initializing, parsing, and leveraging.
3) We provide concrete examples demonstrating how our proposed framework could be developed using existing techniques, connecting our ideas to related problems in other research areas (e.g., natural language processing). The proposed framework utilizes context, variability, and iterative refinement to more appropriately map the reasoning process.
4) We discuss opportunities to advance visual and interactive analytics research with our proposed framework.

BACKGROUND

Task Abstraction

Within the visualization literature, the notion of "task abstraction" is often discussed in the context of taxonomies of task descriptions, which might be generalizable and yet specific enough to support the analysis of user activity and design.[3] At the heart of task abstraction is the idea that low-level operations can be grouped into sets that can themselves be usefully considered as unified, purposeful units of action. These units of action may then be grouped into still larger units of action and so on. Hence, any given coherent sequence of operations can be described in terms of an abstraction hierarchy in which higher level actions *supervene* over lower level action. Implicit in this is the idea that coherent user-activity can be analyzed at multiple levels of granularity, yet can also be decomposed into means (i.e., "how") and aggregated into motivations (i.e., "why").

The idea of task embedding has a long history in ergonomics and HCI. Possibly the best known example appears in hierarchical task analysis (HTA), which according to Stanton[4] was first described by Annett *et al.*[5] HTA uses the idea of task embedding to underpin an approach to task analysis that organizes clusters of ordered activity into hierarchically structured models of goals and subgoals. Lower level subgoals are expansions of higher ones, with goal statements augmented with plans to specify subgoal order and conditions. The uses of HTAs range from the design of interfaces, operating procedures, and training to the analysis of workload and manning levels.[4]

Task abstraction is also a central concept in the abstraction hierarchy, which forms part of cognitive work analysis (CWA).[6] CWA is a framework for modeling complex socio-technical systems, emphasizing the integration of technical functions with human cognitive capabilities to support the design of interfaces, communication systems, training teams, and management systems.

CWA prescribes a series of analysis and modeling steps, each with its own modeling conventions, in which a socio-technical design problem is described in progressively finer detail. This description progresses from an explanation of the work domain as a whole to an analysis of individual competencies and decision making. The abstraction hierarchy is a multilevel representation framework combining both physical and functional models of a cognitive worksystem at the top level is functional purpose. Below this are abstract functions, a decomposition from the system to the subsystem level. Following this are further decompositions to generalized functions, then physical functions (states of individual components), and finally physical form, describing the appearance, condition, and location of the components.[7] Aside from the final layer, each level represents a purpose, which is realized by the layer below and explained by the layer above. One interesting departure from the conventions of HTA is that in the abstraction hierarchy, any functions can be linked to multiple explanatory or supporting functions.

Similar analyses have been described within the visualization literature. For example, to better understand how analysts decompose analysis goals into tasks, Lam *et al.*[8] designed a representation (in this case, a framework) of a goals-to-task decomposition, based on a review of visualization design papers.[9] However, current provenance-based analyses focus on extracting only the data needed to achieve very specific goals. For example, Lam *et al.* intentionally focus on higher level goals and highly focused tasks, leaving out other information such as individual interactions. In contrast, Brown *et al.*[10] focus on low-level interaction patterns in their provenance analysis, and ignore hierarchical structure and higher level tasks. Battle *et al.*[11] consider the relationship between goals and interaction sequences in the context of panning and zooming interactions. Due to the difficulty of inferring tasks, the instrumentation of provenance in visual analytics applications often concerns only subtasks of analysis.

Provenance

Ragan *et al.*[12] characterized various types of provenance used in visualization and data analysis as well as their application for analysis purposes. They distinguished between data, visualization, interaction, insight, and rationale provenance, as a way of delineating different types of provenance arising from the use of analytic systems. In computational workflows, there is some history of recording provenance information. Davidson and Freire,[13] for example, argued for storing and leveraging provenance information from different sources, including information about where it was generated and what type of provenance it is. In considering the recovery of the reasoning information from provenance interaction, Dou *et al.*[14] argued that in highly interactive systems, interactions alone lacked the context of the visual representations present to infer underlying reasoning.

Herschel *et al.*[15] presented a survey to identify what types of provenance are captured and for what purpose. However, they did not address the issue of how multiple types of provenance can be combined to determine dependencies of individual tasks that generate or utilize provenance. Andrienko *et al.*[16] suggested methods for conducting model externalization, including provenance collection among others. While they acknowledged different types of provenance, they also argued that knowledge derived through annotation only represents a fraction of the intent and mental model of the user. They described the difficulties of externalizing the entire mental model and motivated the automated construction of such knowledge models. Further, they identified the need for distinguishing between building, evaluating, developing, and reflecting on the model with formal measures of both the effectiveness of the model and user judgments.

A TASK ABSTRACTION FRAMEWORK

Like any (more or less) coherent and purposeful activity, visual analysis can be described as multiple, hierarchically connected levels of description. However, current approaches for capturing provenance information tend to focus only descriptions with limited depth. Using a single organizational structure that represents the full hierarchy of an analysis session, including the high-level goals, the intermediate tasks, and the individual interactions, is helpful for describing an analysis in a more complete yet comprehensible way. We refer to this structure as a *task hierarchy*. Referring back to work noted in Section Task Abstraction, having direct access to this task hierarchy for a series of analysis sessions could have provided direct access to the goals and tasks of interest to Lam *et al.*,[8] the sequences of interactions of interest to Brown *et al.*,[10] and the relationships between the two sought by Battle *et al.*[11] In this section, we present such a task abstraction framework. We argue that this framework can aid system designers in understanding how user intent is inferred and how provenance can be extracted from the structure, ultimately enabling systems to understand and support users in their tasks. Having direct access to such an underlying task hierarchy would be of great use to the visualization community.

A Hierarchical Provenance Structure

Provenance data can take on many forms characterized by both modality and resolution. For example, provenance can be recorded in the form of a log file. A low-cost, low-resolution version may record user input events or screen captures at predetermined time intervals in text format, while other approaches require integrating provenance capture directly into the analytical system. Regardless of the method used to record and archive provenance data, this data must be abstracted in order to make it both accessible and manageable to users and designers.

Provenance can be recorded at various levels of abstraction, whereby higher level provenance can be inferred from lower level abstractions, resulting in a hierarchical structure (see Figure 1). At the lowest level (**Level 0**) of abstraction lies the original, machine-recorded archive of both user and software behavior (cf., the physical form of the CWA), for example, a log file containing thousands or millions of events. This lowest level may contain information about all activities executed while the system was active, representing basic data and interaction provenance. A level above (**Level 1**) can then group such activities, associating them to functions performed by the system (cf., physical functions of the CWA). A still higher level (**Level 2**) could cluster these sequences into coherent actions from function calls (cf., Generalized Functions of the CWA), generating visualization provenance via derivation of the previous data and interaction provenance. At this level, we may start to infer very basic user tasks, such as a drag-and-drop operation. The next level (**Level 3)** groups these basic user tasks into higher level user tasks, such as a series of drag-and-drop operations, to a higher level of abstraction, such as editing a figure or diagram (cf., abstract functions of the CWA). The highest levels in the provenance abstraction hierarchy (**Level m ... Level n**, cf., functional purpose of the CWA) represent the user intent or goals. For example, the user is creating a figure or writing a report. Information could be drawn from the overall views that are used by the users. These

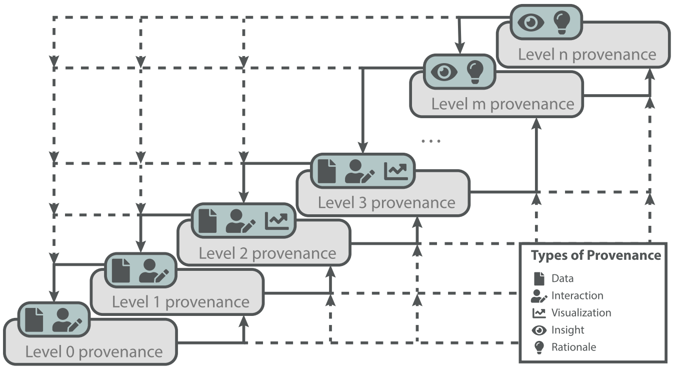

Figure 1. Provenance can be recorded in different types[12] and at different levels of granularity.[1] The dependencies between the different granularities can be mapped into a hierarchical structure.

higher levels of structure can contribute to give lower levels of provenance meaning, and in turn lower level provenance also can give more meaning to insight and rationale provenance (as defined by Ragan *et al.*[12]).

Abstraction Mapping Mechanism

At the heart of our proposed task, abstraction framework is the abstraction mapping mechanism (AMM). The AMM is a conceptual encoding of task hierarchy, mapping low-level interaction sequences to intermediate user tasks, which in turn support higher level goals. While the AMM is a hierarchy, it is not necessarily a tree structure. Instead, the AMM represents a complex set of relationships between processes and data at any level of abstraction. The AMM is also not a structure that remains constant; newly discovered patterns can be dynamically inserted into the AMM to update the structure in an iterative refinement process. In the following sections, we discuss the creation of an AMM from interaction data, complementary bottom-up and top-down parsing procedures, and interaction with the AMM.

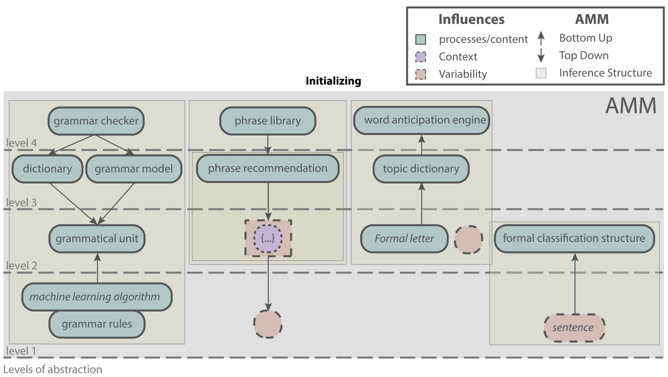

Figure 2. In the *initializing* stage, captured provenance is annotated, mapped into a task hierarchy—by constructing an AMM, and associated across different granularity levels into a hierarchical structure.

Initializing

Provenance log data does not automatically come with tasks and goals labeled for analysis. Instead, a task hierarchy must be inferred from the logs. Thus, the first goal is to develop a process for inferring the AMM from low-level interaction data. To inform the development of such a process, we look to existing work for the guidance.

As part of analyzing visualization system interactions, researchers often manually annotate low-level provenance data with higher level information about intents. This annotation process may be informed by extant models used as coding frameworks, analysis-specific, and emergent issues, as well as a review of task abstraction theory from the literature.[1,9,17] For example, Battle *et al.* used a coding scheme based on the information-seeking mantra[18] to label phases of analysis in their collected log data.[11] As such, researchers often utilize a *top-down* approach in designing rules for constructing task hierarchies.

There also exist examples of analyzing low-level provenance logs to identify interaction subsequences and other low-level patterns, such as through training machine learning models (e.g.,[10,11]). We argue that the initializing process can benefit from both *top-down* and *bottom-up* analysis of provenance data (see Figure 2) unified by the AMM. Combining these corpora of varying granularity into a task hierarchy allows estimating abstract tasks by inferring mappings in a *top-down* and *bottom-up* manner at later stages in our framework. An analysis process of this form requires both contextual input for known, high-level task structures (i.e, inferring encodings of structures from the literature), as well as a sufficiently large corpus of detailed provenance log data to support data-driven analysis techniques, such as training machine learning models (e.g., including interaction information, tool parameters, etc.).

To exemplify these concepts, consider the process of creating an email client capable of forming new sentences, suggesting the next word a user may want to type, and providing detailed email templates. This is not intended to be an analogy of the entire visual analytics process, but rather serves to communicate the main aspects of our framework. A common first step for developing such capabilities is to use machine learning to teach the email client the different ways in which different words are used. Thus, this is a *bottom-up* learning approach in which the email client infers concepts like parts of speech inductively from a series of exemplars. Additionally, informal taxonomic associations between words may be formed, such as *"pig"* and *"horse"* as types of farm animal. In contrast, providing a formal classification structure to learn from reflects a *top-down* approach. For instance, providing the algorithm with grammar rules would then cause it to learn how to determine the semantic parts of a sentence. Words from provided sentences would then be sorted into parts of speech in a concrete manner. Both bottom-up and top-down approaches may be interleaved to provide robust results.

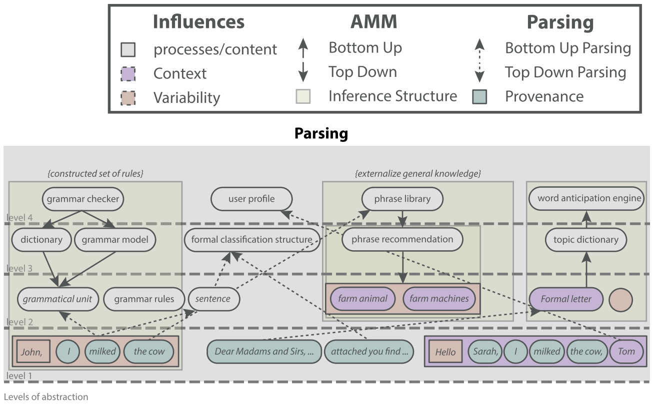

Figure 3. In the *parsing* stage, the AMM constructed in the *initializing* stage is used to estimate pursued tasks, based on the captured provenance and known context.

Parsing

The second stage involves the computational interpretation of logged provenance data into higher level task descriptions. This is essentially a parsing operation in which interaction event sequences are translated into more abstract task categories, utilizing the AMM generated at the initializing stage (see Figure 3). This mechanism can produce anything from a strict set of rules to general knowledge that is externalized.

There are a number of challenges to such interpretation. First, the meaning of any sequence of operations is an emergent property of the sequence. Low-level operations only have determinate meaning to the extent that they are related to other low-level operations. Hence, sequences must be interpreted holistically. Further, sequences themselves depend on other sequences for their interpretation. Any event in the sequence forms part of the context for every other event.

An additional challenge is the often unpredictable nature of user interaction: Interaction with visual analytics systems is frequently opportunistic and exploratory. Users may do things for no apparent reason and with no apparent connection to any previous or future action. They may begin analysis sequences that they do not finish, or they may finish them some time later after an interruption. Hence, the parsing approach strives to be holistic and resilient to incomplete and interrupted tasks. Any interpretation would almost certainly be incomplete, consist of task stubs, and may vary in terms of the level of interpretation achieved.

As a result, we propose an interpretation process that applies both bottom-up and top-down parse strategies, each being called upon opportunistically. As a bottom-up process, sequences of low-level operations are considered as instances of higher level tasks, and sequences of higher level tasks are considered as instances of yet higher level tasks. Partially matched tasks can cue a complementary top-down strategy, driving the search for lower level tasks/operations that would complete them (i.e., given a sequence of user operations, task abstractions can be used to predict and drive the search for the most likely subsequent operation).

We anticipate complementary bottom-up and top-down processes operating at multiple levels of description concurrently, with each level suggesting higher level interpretations and each interpretation suggesting lower level events. This is essentially a hermeneutic circle, required in order to achieve a holistic and context-sensitive analysis. Among others, the context for a given event (or group of events) are its neighbors plus any candidate interpretations that each *n*th event can contribute to the resolution of competing interpretations. The result of the analysis is a

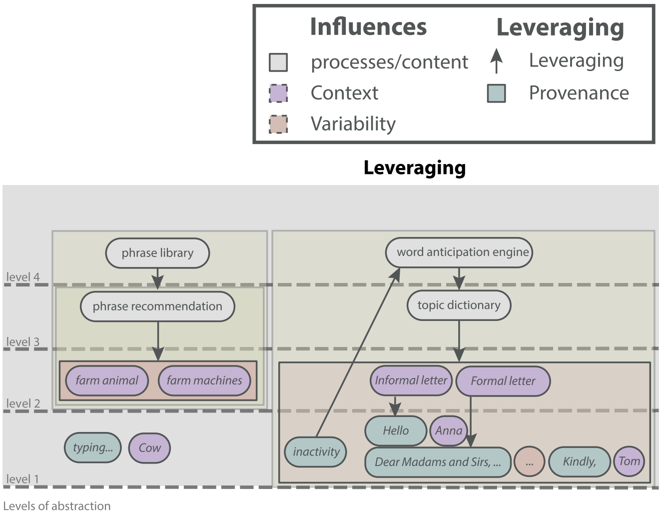

Figure 4. In the last stage of *leveraging* the task hierarchy, actions can be initiated based on inferences, leveraging detected analysis tasks. Depending on the levels of provenance and detected tasks, actions can then be initiated on the same level but also on higher/lower levels accordingly.

hierarchical structure akin to a parse tree in which low-level operations are mapped to higher level interpretations.

Continuing with the email client analogy from the initializing phase, this stage equates to an intelligent email client that is able to use a grammar model, drawing inferences about how words in a message link together to form embedded grammatical units. A part of this process would be the client interpreting sequences of words into *"known"* embedded structures that it knows about, thereby using these structures to anticipate subsequent words from the user.

Leveraging

The third stage centers around the means by which a user interacts with the AMM generated from the previous two stages while using the system (see Figure 4). Because each user of an analytical system will have different goals, different levels of expertise, and different expectations of support, the way in which information and suggestions are presented to users will differ.

Starting with a top-down level of support intended for novice users, the system could generate a set of templates for how to complete the required task of the user. The user is then guided through the steps to, for example, select the data that they wish to analyze, choose the method by which they want the data to be evaluated, and identify the visual structure for presenting the results of that analysis. The AMM can further assist in redirecting and correcting the user if they begin to drift from the template, using the knowledge stored from previous interactions to detect when the user begins to perform unexpected or unhelpful actions. A similar argument can be made for detecting the frustration of a user, who may begin to exhibit such unexpected behavior as a demonstration of their frustration.

Similarly, the AMM can support bottom-up processes, building upon individual interactions. In this case, the system permits the user to begin interacting with the system, and will provide suggestions for next operations that will guide the user toward the completion of their task. These operations could be suggested at varying levels within the AMM's hierarchy, ranging from individual clicks (e.g., suggesting interactions) to higher level analysis phases (e.g., offering the next step in analysis). Here, the AMM uses its stored knowledge to detect the interactions of the user, infer the next step in their analytical process, and then suggest future interactions based upon the training data.

Further, an analytical system can combine the knowledge of the user and the history of previous tasks to optimize the visual interface, attempting to maximize the efficiency of the user by hiding unnecessary functionality and focus their attention on the interactions that will enable them to reach their goal. The AMM could also be presented to the system developers, permitting them a glimpse at how users are actually behaving in the system. If a developer can identify the pain points in their current implementation by seeing where users most often run into difficulty, they can work to resolve these issues in future versions of the system.

Using our intelligent email client example to demonstrate these ideas, the system at this stage would be able to anticipate the next word that the user may want and suggest it automatically. That is, the word processor does not just understand what new sentences may be formed, but it is also capable of determining probable words or phrases that may be used next. Other examples of this can be found in the tab completion functionality of IDEs and software such as overleaf, which are capable of learning and adapting to user-defined

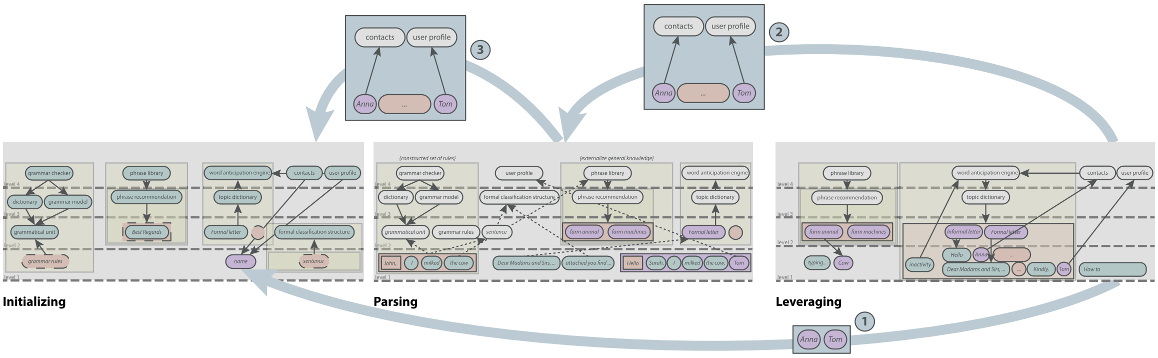

Figure 5. Overview of the framework's three stages and how discovered patterns can be processed and iteratively change the hierarchy: 1) New context is added into the parsing stage, 2) the model is extended with a new activity, which is 3) consecutively parsed into a new rule.

functions and commands. Alternatively, the email client may provide a template for users to leverage, walking them through the template interactively to form a well-written email. These examples involve both top-down and bottom-up processes, leveraging the capabilities and understanding formed by the previous two steps. However, this step does not involve changing the AMM hierarchical structure; rather, the AMM is simply leveraged here to support advanced features. This does not mean that such changes are impossible; it merely means that such changes must be accomplished by revisiting the previous steps in the process as appropriate.

IMPROVING THE FRAMEWORK

Previous sections described the major stages of the framework. There are inherent complexities in the design and implementation of a multistage task hierarchy and the AMM described in this article: 1) iterative refinement of the AMM, and 2) the explicit mapping of context and variability. Specifically, users and systems are situated in a variety of contextual settings that include tasks, dataset uncertainty, and others. In this section, we discuss the influencing factors on our proposed approach. Continuous improvement and refinement are necessary to accurately determine these factors, and minimize erroneous mapping.

Iterative Refinement of the Task Hierarchy

It is rare to be able to accurately infer a user's intent on the first try, and the same is true of provenance analysis. Andrienko *et al.*[16] expressed the demand for performing model evaluation and adaption to reflect the accuracy of mental models, specifically. As such, the process of constructing an AMM will need to be performed iteratively, where each iteration will involve revisiting all stages of the framework in both the top-down and bottom-up approaches. For example, new patterns discovered in the low-level interaction sequences may lead the application to identify a more relevant high-level task structure from the literature (e.g., perhaps a particular goals-to-tasks structure suggested by Lam *et al.*[8] is now more relevant than the information-seeking mantra[18]). Similarly, when mapping a new high-level task structure to the low-level data, the task structure can help the application predict what low-level tasks or interactions may appear next in the dataset. Figure 5 provides an example demonstrating how newly discovered patterns can feed back into the different stages of the framework. Successive iterations could be computed until a convergence threshold is reached, representing only minimal changes to the current task hierarchy data structure in later iterations. Ambiguity and misclassifications can lead to indecision in the system. By prompting users with disambiguation efforts during online provenance collection and parsing, this ambiguity can be resolved. While this is not possible for offline AMM parsing, ambiguous patterns can be disambiguated by developers anticipating actions.

Revisiting our example of developing an email client capable of suggesting the next word to type, the trained machine learning algorithm may misclassify or misunderstand certain words. For example, the meaning of the word “land” can vary greatly depending on context, ranging from a plot of land that an individual owns to simply meaning firm ground anywhere on Earth (i.e., not water). Therefore, if the algorithm is only given example sentences where words like “land” are used in a single context, the machine learning algorithm may misinterpret the higher level meaning of a new sentence that uses the word in a different context. If the algorithm is provided feedback to correct this error, then it can learn this new context.

The Role of Context, Variability, and Uncertainty

In all three stages of initializing, parsing, and leveraging the task hierarchy, we noted that information can be inferred from either a top-down or bottom-up structure. Depending on the information that can be obtained and meaningfully interpreted, there is the danger of the AMM to be insufficiently expressive. We argue that the expressiveness of the AMM is associated with the ability to consider context, variability, and uncertainty.

Accounting for context to disambiguate outcomes in the AMM can be based on a number of factors, including but not limited to individual usage (e.g., level of expertise using a VA tool), environmental dependencies (e.g., various views or restrictions of VA systems), the application/analysis domain, or the user profile operating the system. In this way, enriched sensemaking of the provenance data allows for the inference of various interaction patterns, permitting a system to better interpret the multidimensional data and user actions. Over time, context-enriched provenance history data can refine the system’s understanding of what the user is trying to accomplish. However, it also implies that provenance captured in the task hierarchy has a context dependence, which makes the interpreted structure inherently biased to its source context. In our running email client example, the phrase library can only store phrases captured from previous sessions and categorize them with user actions or clustering.

To maintain robustness of the task abstraction framework, the variability of a system, actors, and domains should be represented in the AMM and during all stages of the framework. Iterative refinement can appropriately deal with this variability if the AMM is accurately mapping these factors. Within an analysis system, the set of possible actions and interactions is predetermined, and hence variability can be narrowed down to smaller sets. To give these actions additional meaning, context and variability can be used to derive a more expressive structure. For example, expanding a context menu will limit the user to execute one of the available menu item operations. Capturing, relating, and leveraging these dependencies in the AMM can lead to more appropriate identification and generalization of user intents and actions.

Returning to the email client example which has been trained to suggest the word “land” only within the context of a plot of land, assume that a new user of the email client is a flight instructor. As a result, the emails from this user only incorporate “land” in the context of “how to land a plane.” Without factoring in these context and variability factors, the machine learning algorithm would repeatedly misinterpret the use of the word. However, if the framework supports iterative, contextual refinement, feedback from the user can be collected to adapt the meaning of the word, as well as to prioritize the new meaning over the old.

We now discuss details on influences of variability and context in the different stages of the framework.

In Initializing In order for context to be beneficial in the initializing stage of the framework, capturing and inferring structure can account for domain-specific customs. In order to more accurately find and subsequently interpret patterns, information about user profiles, application context, and analysis goals can be employed in initializing the hierarchy. For example, provenance data captured from both analysis and system interactions can be annotated with information provided by system developers (e.g., computing additional performance metrics, filtering input, associating a function or method with a specific task/goal).

Variability is a further factor that can influence framework initialization. In visual analytics systems, variability is influenced by 1) the set of available interactions, 2) user expertise and guidance, 3) the set of pursued and recognized, and tasks, and 4) data characteristics and dimensionality.

We propose that influences of variability can be quantified in the AMM in the form of probabilities, e.g., uncertainty measures. During initializing, these uncertainties can be used to direct inferences, yielding a more flexible representation of user intents.

In Parsing In the previous initializing stage, context and variability were introduced to aid interpretation, and uncertainties comprise the probability of an action being part of a sequence that subsequently comprises a task. Furthermore, contextual cues can be used to concisely eliminate ambiguity (e.g., selecting a specific data point and closing a view indicates that the user deliberately ended the selection process). In the parsing stage, methods can be employed that identify, capture, and explicitly incorporate context in the AMM. This can be done on various levels, e.g., estimating analysts' expertise based on the pace of interactions and overall time spent in the VA system.

In contrast, variability must be adapted constantly when interpreting the hierarchy. In the previous phase, variability could only be estimated. When parsing, estimated probabilities can be validated and adapted based on captured provenance, and ambiguous hierarchies can be altered. User intents can vary widely among a collection of users, including in their methodology, expertise, and initiative. Most commonly, pursued tasks can be suspended or dropped in favor of another due to a branch in the analysis process. However, the possibility that a user will return to a certain task cannot be outruled, so task estimations can run in parallel to estimate upcoming actions. This also introduces a temporal aspect into the task hierarchy, where variability in actions progressively influences ambiguity of tasks that are possibly pursued.

To deal with uncertainty arising from potential incompleteness and ambiguity, interpretations may be assigned probabilities depending on factors such as how completely they integrate task primitives (i.e., how much of the "evidence" has been accounted for) and the relative depth of an analysis. These probabilities may be used as heuristics for the adjudication between competing interpretations for further search/expansion. An additional possibility to improve interpretation plausibility is to permit users to explicitly assess the mapping outcome following a set of interactions. Such feedback can be used for continuous adaption, thereby supporting contextual circumstances that would lead to different interaction-intent mappings.

In Leveraging When using the task hierarchy, variability and uncertainty can be leveraged to improve an analysis outcome. For example, this can be accomplished by methods ranging from actively recommending likely outcomes of tasks to anticipating costly computations and running them in advance. Continuously improving the task hierarchy by adapting variability of outcomes and corresponding uncertainties can lead to more complete and effective analysis, supporting users by anticipating future actions through the hierarchy. When parsing and leveraging the framework in a live scenario, users could be prompted if the recommended actions were actually useful, and feedback could contribute in changes of AMM task probabilities. In online hierarchy parsing and leveraging scenarios, users might be prompted if the recommended actions were actually useful, and feedback could contribute in changes of AMM task probabilities. Domain knowledge can be actively added into the system by users. Referring to our email client example, users might add slang words to the dictionary they frequently use that should not be autocorrected. Another aspect can be the retrospective analysis of provenance logs by developers and designers to understand how context and variability affected probabilities in the AMM in practical use cases, and if both existing tasks are matched without ambiguity and new tasks can be inferred from ambiguous sequences.

USE CASE

This section describes a use-case scenario to demonstrate the application of the conceptual task abstraction framework, using it to interpret a provenance dataset. Alice and Bob are hobby cyclists who seek to get into shape for the next season. As a first step, they want to

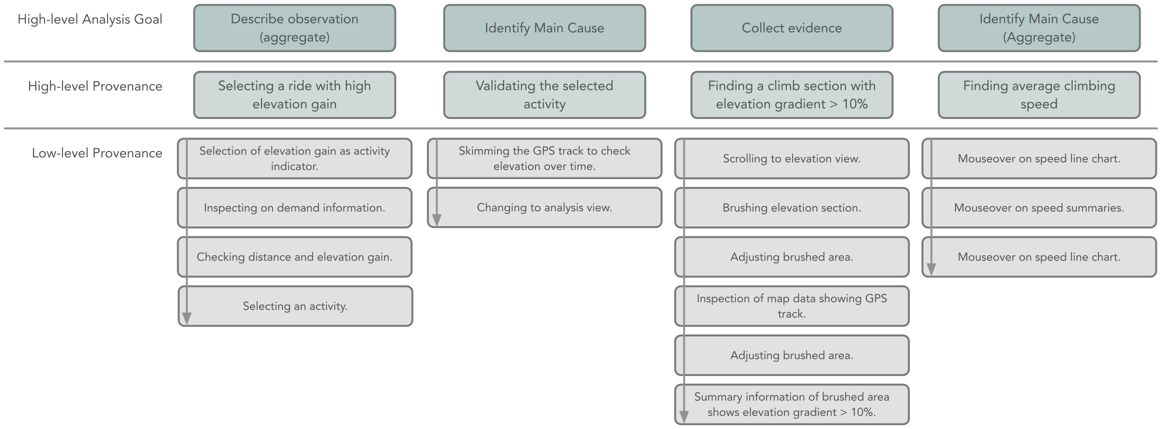

Figure 6. Overview of the activity protocol from the Strava activity analysis. It illustrates two levels of provenance (high and low), with goals abstracted from the high-level provenance data. The high-level analysis goals are annotated, using analysis goals from Lam *et al.*[8]

determine their base fitness level. They are using activity trackers to record their rides, which are then synchronized to the Strava social sports tracking platform. Strava supports detailed analysis of their activities, including comparing their performances to that of other riders on some segments of the rides. We recorded Bob's analysis trail to determine the average climbing speed from a past ride in order to compare it to Alice's climbing speed, a common task that will be added as an experimental feature to Strava with the help of our provenance abstraction framework. This recorded trail of interaction includes both log data as well as a video of Bob exploring the self-recorded activity data using Strava. The provenance data captured from the analysis are semi-structured due to the analysis being broken down into multiple webpages.

Figure 6 shows an overview of the captured provenance information, structured into low-level actions and corresponding high-level tasks. The high-level tasks are used to abstract the user's actions in a top-down fashion. The Strava environment is confined, allowing only small variations in analysis scenarios. The provenance hierarchy is enhanced to accommodate context and variability in analyses:

1) Switching the activity indicator changes the scope to elevation analysis.
2) Applying a clustering algorithm on the GPS track shows three segments: two segments without elevation gain and one with elevation gain.
3) Brushing indicates an area of interest. The selected area is evaluated, and the estimated trend calculation result is zero, indicating a circular activity. In the second brushing action, the trend calculation shows a positive trend. Since we are looking at elevation over time, this implies a climb.
4) The mouseover information shows that the user was interested in the average speed data. This indicates the user's interest in average speed for a selected climb.

Next, we use recommendations to guide Bob through the analysis of a time trial section, which shows different characteristics from a climb section. These recommendations are incorporated as a new feature in the Strava web interface, recommending interesting features in the GPS tracks based on the constructed task hierarchy. At the beginning, the activity type is selected to be distance as opposed to elevation, context information that implies that elevation gain was not determined to be relevant to the user. Based on this assumption, the algorithm segments three uninterrupted flat sections in the selected activity, suggesting them to the user to perform a detailed analysis. The user dismisses two of them, but accepts the third. The elevation within this segment is unchanged throughout; however, automatic clustering of the trajectory data reveals a recurring pattern, implying more relevant contextual information. After the user again interacts with the average speed and estimated power values, the algorithm determines that the recurring pattern corresponds to a segment (a predefined section to compare performances of users). The task hierarchy is adapted to incorporate the newly discovered context, and to search in the available segments for sections representative of the current analysis scenario (time trial, or elevation gain/climbs).

DISCUSSION

Our proposed conceptual task abstraction framework enables a meaningful mapping between raw provenance trails and higher level descriptions of tasks. What distinguishes it from current approaches is that the mapping allows data analysts and end users to use provenance data by leveraging the hierarchy during their analysis being able to derive tasks and goals. Decomposed high-level tasks and goals, like by Lam *et al.*,[8] and low-level interaction patterns[10] can be combined, facilitating task inference. Thus, users can be directed to more meaningful and accurate data insights.

We note that our proposed framework is conceptual and has not yet been proven in practice. However, we illustrated the possible application of the framework in an activity analysis scenario by showing how the task hierarchy and AMM are constructed based on a provenance generated from low-level actions observed from a video inspection and derived high-level tasks. The constructed hierarchy was then applied to support analysis in a recommendation engine of the analysis tool.

Significant challenges remain in practical implementation of the framework, particularly in learning from a large number of log files or from historical provenance data in various contexts. Another challenge is integrating this task abstraction framework into a visual analytics system and determining sensible levels of granularity for capturing provenance from interactions, and then creating expressive links between these levels. Deep learning and artificial intelligence techniques can help us to solve this challenge by iteratively improving the AMM.

We argue that the integration of context and variability allows for a more flexible and concise representation when parsing the task hierarchy. Iterative refinement will ensure unexpected interactions and provenance will be used for further optimizing the hierarchy. Gathering and externalizing this contextual information and variability is highly domain- and application-dependent. As a result, developers of visual analytics systems will need to account for these factors in a context-dependent method. Assigning uncertainties to tasks in the AMM is one approach for handling ambiguity.

We further note that using high-level tasks for recommending actions to users could be detrimental, as many systems are available to users with various levels of experience, expecting and accepting different levels of support. The difficulty of deriving high-level tasks also poses an interesting challenge, with developers having to resort to demonstrations or training videos to determine how users actually generate insight in existing systems and applications. We hope to provide a framework that accommodates appropriate high-level inferences to render this method of determining high-level goals unnecessary, thereby making goal estimations more accurate.

AGENDA

We have outlined our framework and presented a concrete use case that exemplifies how a task abstraction framework from provenance can be utilized to facilitate analysis, leveraging interactions and context to derive user intents. Here, we list some research considerations and challenges that must be addressed in order to implement such a task abstraction framework.

Formalizing Levels of Provenance

Perhaps the most significant challenge involves designing the syntax for describing tasks, particularly if an AMM is designed with maximum flexibility in mind. Task abstraction is an information hiding process in which some features are used for classification and others are ignored. However, details can be retained to a certain extent through the use of arguments. For example, if a user adjusts a filter, there is the question of which filter was the adjustment. Hence, a task language might feature a statement of the form `filter_adjust(name, start, end)`. The optimal syntax of such a language is an open question.

A Hierarchical Provenance Standard

Current approaches for capturing and externalizing provenance in visual analytics systems resort to idiosyncratic provenance structures and lack hierarchical structure. Context is often implicitly considered, but is rarely externalized as provenance. Further, the incorporation of variability when feeding provenance back in the system is rarely supported. As a result, incomplete or ambiguous activities cannot be appropriately mapped to high-level tasks. Establishing a generic provenance model that can accommodate a hierarchical structure and supports the integration of context and variability would allow developers to leverage this formalized information, permitting the implementation of more flexible solutions based on logged provenance.

High-Level Goals

Constructing a provenance hierarchy has the goal of determining overall analysis tasks, which often may be reduced to one single goal. However, this is rarely the case, especially when considering feature-rich visual analytics systems. Capturing insight and rationale provenance requires collecting additional information.[12] Training videos or paper/publication videos often demonstrate how to use a tool to accomplish a specific task. While this task is typically held outside of any broader goal or context, it provides a simple yet effective way of capturing insight and rationale provenance for high-level tasks. However, better ways for automatically externalizing tasks as provenance could be immensely beneficial, because availability of such demonstrations is limited and highly system specific.

Granularity of Mappings between Levels of Hierarchy

A challenge in implementing an AMM is determining the granularity of the mappings between each level of the hierarchy. Developers trying to implement an AMM must consider how granular the mappings between levels of the hierarchy should be. Some mappings between level pairs may be finer or coarser than others, depending on how important the finer-grained information is and how it might be used. The hierarchy needs to support such varying granularities of the mappings.

Automated Capture of Context

While we have discussed different means of determining and integrating context and variability into a provenance framework, we still consider the automatic detection of relevant context as a challenge for future research. Automatic retrieval of context could significantly increase the amount of data collected, and subsequently meaningful information could be concealed by noisy, irrelevant data, so this retrieval must be considered carefully. Simultaneously, automatically assessing the variability of provenance can significantly improve the accuracy of detected tasks, distinguishing users who are pursuing different analysis approaches. Accounting for this could reduce ambiguity and interpretation bias in the iterative provenance hierarchy.

Guidance from the Task Hierarchy

At a higher level, possibilities for task hierarchy usage speak to broad UX and UI challenges. A significant challenge to user-centered design is identifying when it is appropriate to present guidance and feedback to users, as well as what form that guidance should take. The overarching goal of these interventions is to minimize interruptions and frustrations to the analysis process of the user, while still remaining helpful enough to correct any issues faced by the user during their interactions. The task hierarchy could be leveraged to optimize guidance and feedback more appropriately with less interaction interruption.

CONCLUSION

We presented a conceptual framework that leverages a hierarchical provenance structure to generate effective task abstraction across multiple levels of provenance. The creation of this provenance structure, which we termed an abstraction mapping mechanism, consists of three stages: initialization of the provenance hierarchy, the parsing of it into a task abstraction hierarchy, and the leveraging of this task abstraction hierarchy to aid users of visual analytics systems. We discussed the effects of context, variability, and uncertainty of this framework, and demonstrated a use-case scenario to illustrate the possible application of the framework. We conclude by outlining an agenda that discusses challenges associated with the implementation of the framework in practice.

ACKNOWLEDGMENTS

This work was inspired by conversations initiated during the Dagstuhl seminar #18462: Provenance and Logging for Sensemaking. This work was supported by the Austrian Science Fund (FWF), Project No. I 2850-N31, Lead Agency Procedure (DACH) "Visual Segmentation and Labeling of Multivariate Time Series (VISSECT)".

REFERENCES

1. D. Gotz and M. X. Zhou, "Characterizing users' visual analytic activity for insight provenance," *Inf. Visualization*, vol. 8, no. 1, pp. 42–55, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1057/ivs.2008.31
2. K. Xu, S. Attfield, T. Jankun-Kelly, A. Wheat, P. Nguyen, and N. Selvaraj, "Analytic provenance for sensemaking: A research agenda," *IEEE Comput. Graph. Appl.*, vol. 35, no. 3, pp. 56–64, May 2015.
3. T. Munzner, *Visualization Analysis and Design*, ser. A. K. Peters visualization series. A. K. Peters, Boca Raton, FL, USA: CRC Press, 2014. [Online]. Available: http://www.crcpress.com/product/isbn/9781466508910
4. N. A. Stanton, "Hierarchical task analysis: Developments, applications, and extensions," *Appl. Ergonom.*, vol. 37, no. 1, pp. 55–79, Jan. 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0003687005000980
5. J. Annett, K. D. Duncan, R. B. Stammers, and M. J. Gray, *Task Analysis*. Department of Employment Training Information, Her Majesty's Stationary Office (HMSO), London, U.K., 1971.
6. K. J. Vicente, *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-Based Work*. Boca Raton, FL, USA: CRC Press, 1999.
7. A. M. Bisantz and K. J. Vicente, "Making the abstraction hierarchy concrete," *Int. J. Human-Comput. Stud.*, vol. 40, no. 1, pp. 83–117, 1994.
8. H. Lam, M. Tory, and T. Munzner, "Bridging from goals to tasks with design study analysis reports," *IEEE Trans. Visualization Comput. Graph.*, vol. 24, no. 1, pp. 435–445, Jan. 2018.
9. M. Brehmer and T. Munzner, "A multi-level typology of abstract visualization tasks," *IEEE Trans. Visualization Comput. Graph.*, vol. 19, no. 12, pp. 2376–2385, Dec. 2013.
10. E. T. Brown *et al.*, "Finding Waldo: Learning about users from their interactions," *IEEE Trans. Visualization Comput. Graph.*, vol. 20, no. 12, pp. 1663–1672, Dec. 2014.
11. L. Battle, R. Chang, and M. Stonebraker, "Dynamic prefetching of data tiles for interactive visualization," in *Proc. Int. Conf. Manage. Data*, 2016, pp. 1363–1375. [Online]. Available: http://doi.acm.org/10.1145/2882903.2882919
12. E. D. Ragan, A. Endert, J. Sanyal, and J. Chen, "Characterizing provenance in visualization and data analysis: An organizational framework of provenance types and purposes," *IEEE Trans. Visualization Comput. Graph.*, vol. 22, no. 1, pp. 31–40, Jan. 2016.
13. S. B. Davidson and J. Freire, "Provenance and scientific workflows: Challenges and opportunities," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2008, pp. 1345–1350. [Online]. Available: http://doi.acm.org/10.1145/1376616.1376772
14. W. Dou, D. H. Jeong, F. Stukes, W. Ribarsky, H. R. Lipford, and R. Chang, "Recovering reasoning processes from user interactions," *IEEE Comput. Graph. Appl.*, vol. 29, no. 3, pp. 52–61, May 2009.
15. M. Herschel, R. Diestelkämper, and H. B. Lahmar, "A survey on provenance: What for? What form? What from?" *The VLDB J.*, vol. 26, no. 6, pp. 881–906, Dec. 2017. [Online]. Available: https://link.springer.com/article/10.1007/s00778-017-0486-1
16. N. Andrienko *et al.*, "Viewing visual analytics as model building," *Comput. Graph. Forum*, vol. 37, no. 6, pp. 275–299, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13324
17. J. Heer and B. Shneiderman, "Interactive dynamics for visual analysis," *Queue*, vol. 10, no. 2, p. 30, 2012.
18. B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Craft Inf. Visualization*. Amsterdam, The Netherlands: Elsevier, 2003, pp. 364–371.

Christian Bors is currently working toward the Ph.D. degree at the Institute of Visual Computing and Human-Centered Technology, TU Wien. His main area of research is developing interactive techniques for data wrangling, profiling, and cleansing systems, utilizing quality metrics and data provenance. Contact him at christian.bors@tuwien.ac.at.

John Wenskovitch is a Visiting Assistant Professor in the Computer Science Department, Virginia Tech. His research seeks to enable human–AI interaction for big data analytics, with specific foci on high-dimensional data, sensemaking, and clustering. Contact him at jw87@vt.edu.

Michelle Dowling is currently working toward the Ph.D. degree with the Computer Science Department, Virginia Tech. Her research focuses on developing symmetric visualization and interaction techniques for high-dimensional data and text data. Contact her at dowlingm@vt.edu.

Simon Attfield is an Associate Professor of User Centred Technology at Middlesex University, London, U.K. His research areas include HCI, information seeking, sensemaking, visualisation, and engagement. He has a particular interest in sensemaking understood from the perspective of Distributed Cognition and the design of resources to support it. He has conducted studies of sensemaking in the wild with journalists, corporate lawyers, military intelligence analysts, and police crime analysts. Contact him at s.attfield@mdx.ac.uk.

Leilani Battle is an Assistant Professor at the University of Maryland, College Park. Her research integrates diverse methodology and techniques across databases, HCI, and visualization to enable large-scale data processing using interactive, exploration-focused interfaces. Her research has garnered awards from the National Science Foundation (NSF GRFP, NSF CISE CRII), Oak Ridge Associated Universities, and Adobe. She completed her postdoctoral work at the University of Washington (UW), received the M.S. and Ph.D. degrees in computer science from MIT, in 2013 and 2017, respectively, and the B.S. degree in computer engineering from UW in 2011. Contact her at leilani@cs.umd.edu.

Alex Endert is an Assistant Professor in the School of Interactive Computing, Georgia Tech. He directs the Visual Analytics Lab, which performs research that explores novel user interaction techniques for visual analytics. His lab applies these fundamental advances to domains including text analysis, intelligence analysis, cyber security, manufacturing, and others. Contact him at endert@gatech.edu.

Olga Kulyk is a Senior Advisor and Project Manager Digital Trust at InnoValor, The Netherlands. She received the Ph.D. degree in computer science and has over nine years of international academic and industrial experience in coordinating R&D projects in agile software engineering, healthcare, life sciences, and education. She is an expert in user-centered design, personalized visualizations, and situational awareness for decision-making support. Her current research areas include privacy-by-design, digital identities, biometrics, user-centric data management, and AI. Contact her at olgakulyk.nl@gmail.com.

Robert S. Laramee is an Associate Professor of data visualization at Swansea University, Wales, U.K. His research interests include scientific visualization, information visualization, and visual analytics. He has published more that 160 scientific, peer-reviewed articles on these topics. Contact him at r.s.laramee@swansea.ac.uk.