A Multi-Level Typology of Abstract Visualization Tasks

Matthew Brehmer and Tamara Munzner, *Member, IEEE*

Abstract—The considerable previous work characterizing visualization usage has focused on low-level tasks or interactions and high-level tasks, leaving a gap between them that is not addressed. This gap leads to a lack of distinction between the ends and means of a task, limiting the potential for rigorous analysis. We contribute a multi-level typology of visualization tasks to address this gap, distinguishing *why* and *how* a visualization task is performed, as well as *what* the task inputs and outputs are. Our typology allows complex tasks to be expressed as sequences of interdependent simpler tasks, resulting in concise and flexible descriptions for tasks of varying complexity and scope. It provides abstract rather than domain-specific descriptions of tasks, so that useful comparisons can be made between visualization systems targeted at different application domains. This descriptive power supports a level of analysis required for the generation of new designs, by guiding the translation of domain-specific problems into abstract tasks, and for the qualitative evaluation of visualization usage. We demonstrate the benefits of our approach in a detailed case study, comparing task descriptions from our typology to those derived from related work. We also discuss the similarities and differences between our typology and over two dozen extant classification systems and theoretical frameworks from the literatures of visualization, human-computer interaction, information retrieval, communications, and cartography.

Index Terms—Typology, visualization models, task and requirements analysis, qualitative evaluation

1 Introduction

Consider a person who encounters a choropleth map while reading a blog post in the aftermath of last year's American presidential election. This particular map is static and visually encodes two attributes, candidate and margin of victory, encoded for each state using a bivariate colour mapping. This person decides to compare the election results of Texas to those of California, motivated not by an explicit need to generate or verify some hypothesis, nor by a need to present the visualization to an audience, but rather by a casual interest in American politics and its two most populous states. How might we describe this person's *task* in an abstract rather than domain-specific way?

According to the nested model for visualization design and validation [43], abstract tasks are domain- and interface-agnostic operations performed by users. Disappointingly, there is little agreement as to the appropriate granularity of an abstract task among the many extant classifications of user behaviour in the visualization, human-computer interaction, cartography, and information retrieval literature [2, 3, 5, 10, 11, 12, 13, 14, 15, 19, 23, 29, 31, 37, 39, 42, 50, 51, 56, 57, 59, 61, 64, 66, 72, 73, 75, 78, 82, 83]. One of the more frequently cited of these [2] would classify the above example as being a series of value retrieval tasks. This low-level characterization does not describe the user's context or motivation; nor does take into account prior experience and background knowledge. For instance, a description of this task might differ if the user was unfamiliar with American geography: the user must locate and identify these states before comparing their values. Conversely, high-level descriptions of exploratory data analysis and presentation emanating from the sensemaking literature [3, 11, 31, 51] cannot aptly describe this user's task.

The gap between low-level and high-level classification leaves us unable to abstractly describe user tasks in a useful way [41], even for the simple static visualization in the above example. This gap widens when interactive visualization is considered, and the complexity of its usage is compounded over time. We must move beyond describing a single task in isolation, to a description that designates when one task ends and another begins. To close this gap, visualization tasks must be describable in an abstract way across multiple levels.

The primary contribution of this paper is a multi-level typology of abstract visualization tasks that unites the previously disconnected scopes of low-level and high-level classification systems by proposing multiple levels of linkage between them. Our typology provides a powerful and flexible way to describe complex tasks as a sequence of interdependent simpler ones. While this typology is very much informed by previous work, it is also the result of new thinking and has many points of divergence with previous models. Central to the organization of our typology are three questions that serve to disambiguate the means and ends of a task: *why* the task is performed, *how* the task is performed, and *what* are the task's inputs and outputs. We have found that no prior characterization of tasks satisfactorily answers all of these questions simultaneously at multiple levels of abstraction. Typically, low-level classification systems provide a sense of *how* a task is performed, but not *why*; high-level classification systems are the converse. One major advantage of our typology over prior work is in providing linkage between these two questions. Another advantage is the ability to link sequences of tasks, made possible by the consideration of *what* tasks operate on.

Our typology provides a consistent lexicon for description that supports making precise comparisons of tasks between different visualization tools and across application domains. Succinct and abstract descriptions of tasks are crucial for analysis of visualization usage. This analysis is an essential precursor to the effective design and evaluation of visualization tools, particularly in the context of problem-driven design studies [60]. In these studies, visualization practitioners work with users to identify *why* and *what*, subsequently drawing from their specialized knowledge of visual encoding and interaction techniques to design *how* that task is to be supported [41]. A need for task analysis also arises in visualization evaluation [34], particularly in observational studies of open-ended visualization usage. Our typology provides a multi-level code set for qualitatively describing user behaviour in such studies.

As we expect some readers to be unfamiliar with the context that motivated this work, we begin with a brief discussion of our current inability to succinctly describe and analyze visualization tasks. In Section 3 of this paper, we introduce our multi-level typology of abstract visualization tasks. In Section 4, we demonstrate the benefits of our approach with a detailed case study, in which we describe a sequence of interdependent tasks. In Section 5, we summarize our typology's connections to related work and to its theoretical foundations. In Section 6, we discuss the value and usage of this typology, as well as our plans for its further validation and extension. In Section 7, we offer our concluding remarks.

- *Matthew Brehmer is with the University of British Columbia. E-mail: brehmer@cs.ubc.ca.*
- *Tamara Munzner is with the University of British Columbia. E-mail: tmm@cs.ubc.ca.*

Manuscript received 31 March 2013; accepted 1 August 2013; posted online 13 October 2013; mailed on 4 October 2013.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

1077-2626/13/$31.00 © 2013 IEEE Published by the IEEE Computer Society

2 Background Context

The primary limiting factor in using extant classification systems as tools for analysis is that we cannot easily distinguish between the ends and means of tasks. Making this distinction is a central problem for practitioners during the abstraction phase of design studies [60] and during the analysis phase of qualitative user studies [34].

For instance, a number of extant classification systems mention the word *derive* [2, 14, 23, 37, 50, 66]. Is *derive* a task, or the means by which another task is executed? A user may derive data items as an end in itself, for example to reduce the number of dimensions in a dataset, or as a means towards another end, such as to verify a hypothesis regarding the existence of clusters in a derived low-dimensional space. The ends-means ambiguity exists for many terms found in extant classification systems: consider *filter* [2, 11, 19, 23, 29, 31, 37, 42, 50, 51, 57, 61, 82], *navigate* [23, 64, 75], or *record* [23, 42, 66]. The first step towards distinguishing ends from means involves asking *why* a task is performed separately from *how* a task is performed, a question that is central to the organization of our typology.

The separation of *why* and *how* does not in itself resolve all confusion. Consider *sort*, another term appearing in extant classification systems [2, 19, 23, 37, 50]. Sorting has an input and an output; in some cases, it is items of data within a single view [55]; in others, views themselves may be sorted [8]. In both cases, the sorted output can serve as input to subsequent tasks. The next step in distinguishing ends from means is thus characterizing *what* the task's inputs and outputs are, allowing us to describe sequences of interdependent tasks.

To illustrate how the ends-means ambiguity arises during the course of analysis, we will now attempt to use representative extant classification systems to describe two example tasks:

Example #1: recall the example stated in the paper's introduction, that of a casual encounter with an electoral map in which a user compares two regions; election results for each state are encoded as a choropleth map based on two attributes, candidate and margin of victory. Furthermore, we know that this user is familiar with American geography and its regions; this prior knowledge dictates the type of search.

Using the typology of Andrienko and Andrienko [5], we might describe this example as an *elementary direct comparison task*. While richer than a series of *retrieve value* tasks [2], this description tells us little about *why* and *how* this comparison was performed. Low-level descriptions derived from a number of other classification systems are similarly impoverished [12, 19, 59, 73, 78, 82, 83].

We might enrich our description of this task using Roth's recent *taxonomy of cartographic interaction primitives* [57], a much more comprehensive approach that distinguishes between *goals*, *objectives*, *operators*, and *operands*. Using his taxonomy, this task would be described as follows:

goals:	objectives:	operators:	operands:
procure	*compare*	*retrieve*; *calculate*	*attribute–in–space* (search target); *general* (search level)

While the dimensions of this description are similar to the questions of *why*, *how*, and *what*, the description is incomplete, particularly in its classification of *goals* and *objectives*. Roth's taxonomy provides us only with a partial sense of *how* the comparison is performed: *retrieve* does not tells us about whether the user knows the spatial location of the regions to be compared a priori. The goal, *procure*, does not provide us with any higher-level context or motivation for *why* the user is *procuring*; specifically, the user's casual interest in these two regions is lost. Finally, Roth's taxonomy imposes a spatial constraint on *operands*, leaving us unable to fully articulate *what* is being compared.

Example #2: in evaluation studies [34], it is sometimes necessary to perform a comparative analysis of a task being performed using different visualization systems. Consider a user of a tree visualization tool who whose interest relates to two nodes in a large tree, and her intent is to present the path between these nodes to her colleagues. SpaceTree [20] and TreeJuxtaposer [46] are two tree visualization tools that allow users to locate paths between nodes by means of different focus + context techniques. Both systems allow for path selection, in which the encoding of selected paths differs from that of non-selected paths. The systems differ in *how* elements of the visualization are manipulated: TreeJuxtaposer allows the user to arrange areas of the tree to ensure visibility for areas of interest, while SpaceTree couples the act of selection by aggregating and filtering unselected items.

As in the previous example, task descriptions from extant classification systems seldom answer all three questions: *why*, *how*, and *what*. Using Heer and Shneiderman's *taxonomy of interactive dynamics for visual analysis* [23], we might describe this task as being an instance of *data and view specification* (*visualize* and *filter*) as well as *view manipulation* (*navigate* and *select*). This description tells us *how*, but it doesn't specify *why* the task is performed.

We might complement Heer and Shneiderman's description with one based on Lee et al.'s taxonomy of graph visualization tasks [37], in which this task would be classified as a *topology task*, namely one of *determining connectivity* and subsequently *finding the shortest path*. As the scope of Lee et al.'s taxonomy is specialized, we are provided with a clear indication of *what* the user's interest is, this being a *path*. Unfortunately, this description provides only a partial account of *why* the task is performed; we are not provided with a high-level motivation beyond *determining* and *finding*.

Both descriptions do not relate the user's actions to the high-level goal of *presenting* information to others. Second, and more importantly, these descriptions fail to distinguish *how* this task is performed using SpaceTree from *how* it is performed using TreeJuxtaposer.

Summary: these examples demonstrate our inability to comprehensively analyze tasks using state-of-the-art systems for classifying user behaviour. Note that we are not directly criticizing these classification systems; we acknowledge that their scope is often deliberately constrained, with some focusing on low-level tasks, interactions, or operations [2, 5, 10, 12, 13, 14, 15, 19, 29, 37, 59, 61, 72, 73, 75, 78, 82, 83], while others focus on high-level tasks or goals [3, 11, 31, 39, 51], or on user behaviour in specialized domains [37, 57, 65]. We lack guidance on how to integrate these disjoint bodies of work, to compose task descriptions that draw from all of them. This integration is the aim of our typology, which will allow practitioners to describe tasks that address critical questions posed during visualization design and evaluation, namely *why*, *how*, and *what*.

It could be argued that a classification of *tasks* should focus solely on the goal of the user, or *why* visualization is used; users are often not immediately concerned with *how* a task is performed, as long as their task can be accomplished. We argue that by classifying tasks according to *how* they are performed, *in addition* to *why* they are performed and *what* they pertain to, we can improve communication between visualization practitioners working in different application areas, facilitating tool-independent comparisons, the analysis of diverging usage strategies for executing tasks [74, 84], and improved reasoning about design alternatives.

3 Typology of Abstract Visualization Tasks

Our multi-level typology of abstract visualization tasks, represented in Figure 1, is encapsulated by three questions: *why* the task is performed, *how* the task is performed, and *what* does the task pertain to (Figures 1a-c). Complete task descriptions, such as those for Examples #1-2 (represented in Figure 2), must include nodes from all three parts of this typology. We denote this work as a *typology*, rather than a *taxonomy*, as the former is appropriate for classifying abstract concepts, while the latter is appropriate for classifying empirically observable events [6].

This structure, while unusual relative to many extant classification systems, mirrors the analytical thinking process undertaken in design studies [41, 43, 60]. *Why*, *what*, and *how* are also used in *Cognitive Work Analysis* [74], particularly for relating abstractions within a work domains, as well as in Aigner et al.'s analysis of techniques and systems for visualizing time-oriented data [1], which asks *what is presented?*, *why is it presented?*, and *how is it presented?*.

We will introduce *why* before *how*, as this order reflects the translation of empirically observable domain problems into abstract tasks and subsequently into design choices: practitioners first identify *why*

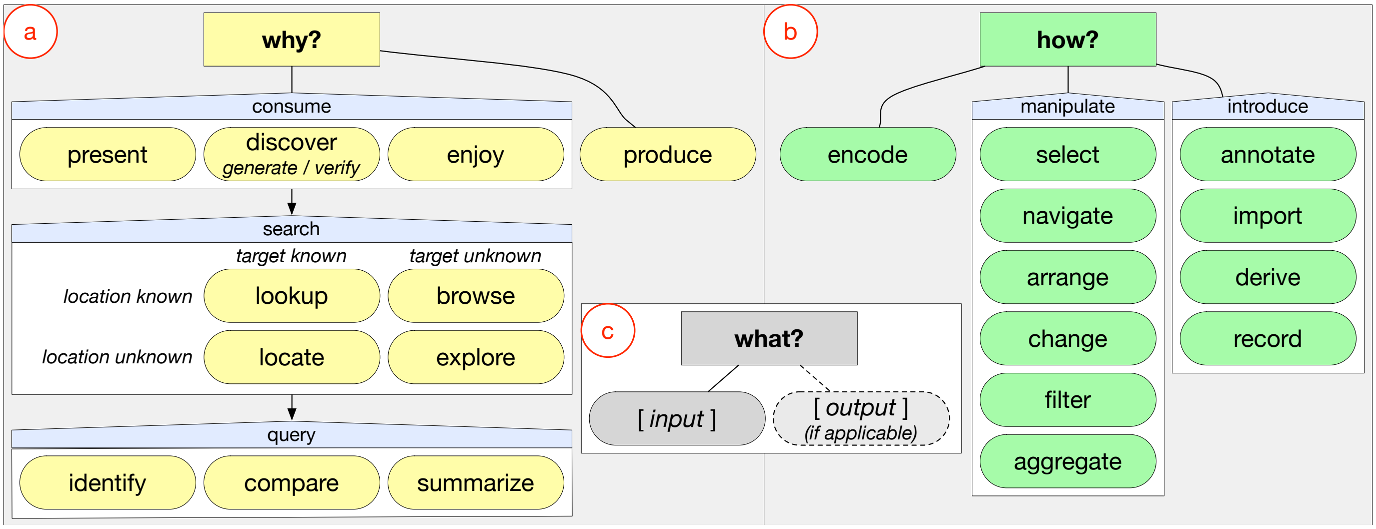


Fig. 1. Our multi-level typology of abstract visualization tasks. The typology spans *why*, *how*, and *what*; task descriptions are formed by nodes from each part: a) *why* a task is performed, from high-level (consume vs. produce) to mid-level (search) to low-level (query). b) *how* a task is executed in terms of *methods*, defined as families of related visual encoding and interaction techniques. c) *what* the task inputs and outputs are.

a task is to be performed, and then must decide upon *how* the task is to be supported. We then discuss the *what* part of our typology, which considers the input and output of tasks. Our typology supports the description of complex visualization usage as a sequence of interdependent tasks, where the output of a prior task may serve as the input to a subsequent task, as we demonstrate in the case study of Section 4.

For clarity, we first present our typology in its entirety with minimal discussion of the previous work that informed its organization, and then focus on these connections in Section 5 and in Table 1.

3.1 Why?

The *why* part of our typology, shown in Figure 1a, allows us to describe *why* a task is performed, and includes multiple levels of specificity, a narrowing of scope from high-level (consume vs. produce) to mid-level (search) to low-level (query).

Consume: Visualizations are used to consume information in many domain contexts. In most cases this consumption is driven by either a need to present information or to discover and analyze new information [79]. However, there are many other contexts in which visualizations are simply enjoyed [17, 54, 65], where users indulge their casual interests in a topic.

`Present` refers to the use of visualization for the succinct communication of information, for telling a story with data, guiding an audience through a series of cognitive operations. Presentation using visualization may take place within the context of decision making, planning, forecasting, and instructional processes [18, 40, 57]. Presentation brings to mind collaborative and pedagogical contexts, and the means by which a presentation is given may vary according to the size of the audience, whether the presentation is live or pre-recorded, and whether the audience is co-located with the presenter [32].

`Discover` is about the generation and verification of hypotheses, associated with modes of scientific inquiry [50]. Scientific investigation may be motivated by existing theories, models, and hypotheses, or by the serendipitous observation of unexpected phenomena [4].

`Enjoy` refers to casual encounters with visualization [54, 65]. In these contexts, the user is not driven by a need to verify or generate a hypothesis; novelty stimulates curiosity and thereby exploration [16, 65, 67, 69]. This motivation is notably absent from previous classification systems, as shown in Table 1. Casual encounters with visualization can be fleeting, such as in the earlier example of encountering a static choropleth electoral map while reading a blog post. Conversely, these encounters might be immersive and time-consuming experiences, such as in museum settings [54].

Produce: we use `produce` in reference to tasks in which the intent is to generate new artefacts. These artefacts include but are not limited to: transformed or derived data, annotations, recorded visualization interactions, or screen shots of static visualizations. Examples of `produce` in previous work include the production of graphical annotations and explanatory notes to describe features of time-series graphs [81], or the production of *graphical histories* in Tableau [21], the latter being meta-visualizations intended to improve users' analytical provenance. Additional examples of `produce` involving derived data and annotations are featured in the case study of Section 4.

It is important to note that the products of a `produce` task may be used in some subsequent task that may or may not involve visualization. For example, some visualization tools for analyzing high-dimensional data allow users to `produce` new categorical attributes for labelling clustered data points in a dimensionally-reduced coordinate space; these attributes might be used later for constructing a predictive model.

Search: Regardless of whether the intent is to `present`, `discover`, or merely `enjoy`, the user must find elements of interest in the visualization. While terms relating to search and exploration are often conflated [40, 69], we have imposed a characterization of search that depends on *what* is being sought. We classify them according to whether the identity or location of the search target is known a priori. Whether the identity of the search target is known recalls Andrienko and Andrienko's concept of *references* and *characteristics* [5]: searching for known reference targets entails `lookup` or `locate`, while searching for targets matching particular characteristics entails `browse` or `explore`. Consider our earlier example of a user who is familiar with American geography and is searching for California on an choropleth map; we would describe this as an instance of `lookup`. However, a user who is *unfamiliar* with American geography must `locate` California.

In contrast, the identity of a search target might be unknown a priori; the user may be searching for characteristics rather than references [5]; these characteristics might include particular values, extremum, anomalies, trends, or ranges [2]. For instance, if a user of a tree visualization is searching within a particular subtree for leaf nodes having few siblings, we would describe this as an instance of `browse` because the location is known a priori. Finally, `explore` entails searching for characteristics without regard to their location, often beginning at an overview level of the visualization [37]. Examples include searching for outliers in a scatterplot, for anomalous spikes or periodic patterns in a line graph of time-series data, or for unanticipated spatially-dependent patterns in a choropleth map.

Query: Once a target or set of targets has been found, a user will identify, compare, or summarize these targets. If a search returns known or *reference* targets [5], either by lookup or locate, identify returns their *characteristics*. For example, a user of a choropleth map representing election results can identify the winning candidate and margin of victory for the state of California. Conversely, if a search returns targets matching particular *characteristics*, either by browse or explore, identify returns *references*. For instance, our election map user can identify the state having the highest margin of victory.

The progression from identify to compare to summarize corresponds to an increase in the amount of search targets under consideration [5, 10, 72], in that identify refers to a single target, compare refers to multiple subsets of targets, and summarize refers to a whole set of targets. As with explore, summarize is also often associated with overviews of the data [37]. Continuing with the choropleth map example, a user identifies the election results for one state, compares the election results of one state to another, or summarizes the election results across all states, determining how many favoured one candidate or the other, or the overall distribution of margin of victory values.

3.2 How?

We now turn our consideration to the *how* part of our typology, which contains *methods*, defined as families of related visual encoding and interaction techniques [45]. This part of our typology, shown in Figure 1b, is likely to be most familiar to readers, as it contains a number of *methods* associated with interaction techniques that are well-represented by several extant classification systems [19, 42, 57, 82]. We distinguish between three classes of methods: those for encoding data, those for manipulating existing elements in a visualization, and those for introducing new elements into a visualization.

Encode: the majority of visualization tasks rely on *how* data is initially encoded as a visual representation. A full enumeration of visual encoding techniques for various data types in terms of methods is beyond the scope of this paper and appears elsewhere [45].

Manipulate: the following methods affect existing elements of a visualization, modifying them to some extent. These methods represent families of interrelated techniques incorporating both interaction and visual encoding. We consider visual encoding and interaction techniques in a unified way because many methods incorporate aspects of both [41, 43], such as focus + context techniques [20, 46].

Select refers to the demarcation of one or more elements in a visualization, differentiating selected from unselected elements [56]. Examples range from directly clicking or lassoing elements in a single visualization to brushing methods used to highlight elements in visualization systems incorporating multiple linked views [77].

Navigate methods include those that alter a user's viewpoint, such as zooming, panning, and rotating. Other methods trigger details-on-demand views, combining navigate and select [61].

Arrange refers to the process of organizing visualization elements spatially. Some of these methods arrange representations of data [39, 42, 80], such as reordering the axes in a parallel coordinates plot or the rows and columns of a scatterplot matrix. Other methods allow users to coordinate the spatial layout of views [23, 77].

Change pertains to alterations in visual encoding. Simple examples include altering the size and transparency of points in a scatterplot or edges in a node-link graph, altering a colour-scale or texture mapping, or transforming the scales of axes. Other methods have more pronounced effects, changing the chart type altogether, such as transitioning between grouped and stacked bar charts, or between linear and radial layouts for time-series graphs. Pronounced changes in visual encoding such as these are often facilitated by smoothly animated transitions, which reduce their disruptive effects [22].

Filter methods adjust the exclusion and inclusion criteria for elements in a visualization. Some methods allow for elements to be temporarily hidden from view and later restored, while other methods are synonymous with outright deletion. As an example of temporary filtering, consider a user examining an age histogram based on population census data. First, she decides to exclude males, then further adjusts her filter criteria to focus solely on unemployed females. Finally, she revises the gender criteria to focus on unemployed males.

A common example of permanent filtering, or deletion, is that of manually selecting and removing outliers resulting from errors in data entry. Alternatively, consider a scatterplot in which some points are labelled with manually generated categorical tags. Deleting a tag would remove this categorical label from all points having that tag.

Aggregate concerns methods that change the granularity of visualization elements. As such, we also consider its converse, segregate, as being associated with this family of methods. For example, a user may adjust the granularity of a continuous time scale in a line graph, aggregating daily values into monthly values, or segregating annual values into quarterly values. Alternatively, a user may aggregate a clique within a node-link diagram into a representative glyph, or segregate clique glyphs into their component nodes.

Introduce: While manipulate methods alter existing elements of the visualization, introduce methods add new elements.

Annotate refers to the addition of graphical or textual annotations associated with one or more visualization elements. When an annotation is associated with data elements, an annotation could be thought of as a new attribute for these elements. The earlier example of manually tagging points in a scatterplot with categorical labels is one such instance of annotating data.

Import pertains to the addition of new elements to the visualization, including new data elements. In some environments, these new data elements might be loaded from external sources, while others might be manually generated.

Derive methods compute new data elements given existing data elements. Aggregating data often implies deriving data, however this may not always be true: we further specify that derived data must be persistent, while aggregated data need not be. For instance, a user might derive new attributes for tabular data using a multidimensional scaling algorithm.

Finally, record methods save or capture visualization elements as persistent artefacts. As such, record methods are often associated with produce. These artefacts include screen shots, annotations, lists of bookmarked elements or locations, parameter settings, or interaction logs [63]. An interesting example of record is that of assembling a *graphical history* [21], in which the output of each task includes a static snapshot of the visualization, and these snapshots accumulate in a branching meta-visualization. Recording and retaining artefacts such as these are often desirable for maintaining a sense of analytical provenance, allowing users to revisit earlier states or parameter settings.

3.3 What?

Previous work has reached no agreement on the question of *what* comprises a visualization. Many classification systems do not address it at all; others discuss *what* implicitly, as indicated by the parenthetical terms in Table 1. Of those that classify *what*, some focus on the level of the entire dataset, such as *tables* composed of *values* and *attributes* or *networks* composed of *nodes* and *links* [61]. Others allow more precise specification of data–attribute semantics, such as *categorical*, *ordinal*, and *quantitative* [11]. A few classification systems include not only *data* but also *views* as first-class citizens [13, 14, 23, 75]. Specific examples of *what* as classified in previous work include:

- Values, extremum, ranges, distributions, anomalies, clusters, correlations [3].
- *Graph-specific objects* [37]: nodes, links, paths, graphs, connected components, clusters, groups.
- *Time-oriented primitives* [1]: points, intervals, spans, temporal patterns, rates of change, sequences, synchronization.
- *Interaction operands* [75]: pixels, data [values, structures], attributes, geometric [objects, surfaces], visualization structures.

In this typology, we have chosen a flexible and agnostic representation of *what* that accommodates all of these modes of thinking: in short, we have a "bring your own *what*" mentality. The only absolute requirement is to explicitly distinguish a task's `input` and `output` constraints when describing sequences of interdependent tasks [72]. An extensive discussion of *what* that dovetails well with this typology appears elsewhere [45], but it cannot be effectively summarized in a few paragraphs; thus, it is beyond the scope of this paper.

3.4 Concise Task Descriptions

Our multi-level typology can be used to concisely describe visualization tasks. Each task is defined by *why* it is executed, the method(s) by which it is executed (*how*), and by *what* the task pertains to. Single tasks may involve multiple nodes from each part of the typology, as shown in Figure 2.

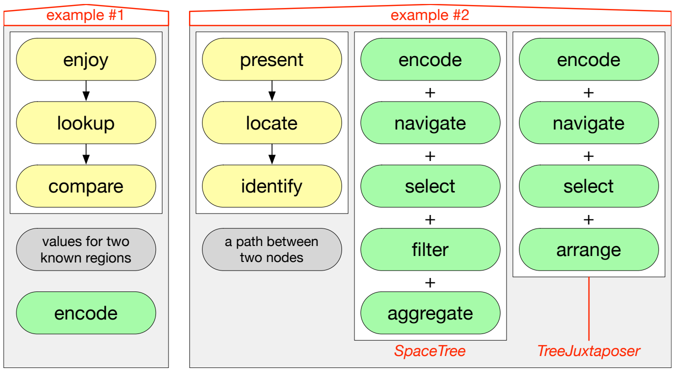


Fig. 2. Task descriptions for Example #1 (left): casually encountering an choropleth electoral map and comparing election results for two regions; and Example #2 (right): presenting a path between two nodes in a large tree using SpaceTree [20] and TreeJuxtaposer [46].

We have chosen to present these descriptions using a simple and flexible visual notation, rather than with a formal grammar [5, 35, 72, 80]; in doing so, creating and iterating on task descriptions can be easily integrated into existing collaborative design and ideation activities, making use of materials such as coloured sticky notes and whiteboards. A crucial aspect of these descriptions is that sequences of interdependent tasks can be chained together, such that the `output` from earlier tasks forms the `input` to later tasks, as discussed in the following case study and as represented in Figure 3.

4 Case Study: A Sequence of Interdependent Tasks

Visualization tasks are seldom executed in isolation, and the output of one task may serve as input to a subsequent task. To illustrate this type of dependency, we present an case study in which our typology is used to describe a sequence of interdependent tasks.

Overview [25] is a visual analytics tool for exploring large text document collections. Overview supports the task of `producing` a set of tags for semantically related documents as a new categorical attribute, using clusters of documents as scaffolding for the `discovery` process. Tags are assigned to clusters by means of `annotation`. However, one must first `explore` visualizations of the document space and `identify` clusters of interest. The user is presented both with two linked visualizations, one in which documents are `encoded` as points in a scatterplot, and another in which clusters of documents are `encoded` in a hierarchical tree layout. `Identifying` clusters is facilitated by `navigating` and `selecting` clusters in both visualizations; upon `selection`, frequent document or cluster terms and raw individual document text are shown in secondary displays. This task too has a dependency on the result of an earlier task. Before the the data is `encoded` in the two visualizations, a set of two-dimensional distances between documents must be `produced`: they are `derived` and `aggregated` from the original high-dimensional distance matrix using a multidimensional scaling algorithm.

Using our typology, we can express dependencies in which the output of one task serves as the input of another, such as the relationship between how data is `derived` and the choice of visual encoding technique [43]. Such dependencies are represented in Figure 3.

As in the examples of Section 2, we can compare our description to those generated by extant classification systems. Consider Chuah and Roth's classification of basic visualization tasks [14], which distinguishes between three categories of *operations*. Using this classification, this sequence of tasks could be described as having *data operations* (*derived attributes*), *graphical operations* (*encode data*), and *set operations* (*create set, express membership*). This description does specify *how* and *what*, but it does not express the interdependencies within a sequence of tasks, nor does it tell us *why* these tasks were performed. Neither can we easily distinguish when sets (or clusters) are created, as this might occur before the data is encoded, such as in the case of the cluster hierarchy, but not in the case of the scatterplot.

While the description based on Chuah and Roth's [14] classification is atemporal, Chi and Riedl's *operator interaction framework* [13] defines stage- and transformation-based operators occurring along the visualization pipeline. Their framework does not contain a comprehensive list of operators, so we draw from the example *operators* cited in their paper to describe this sequence of tasks as follows:

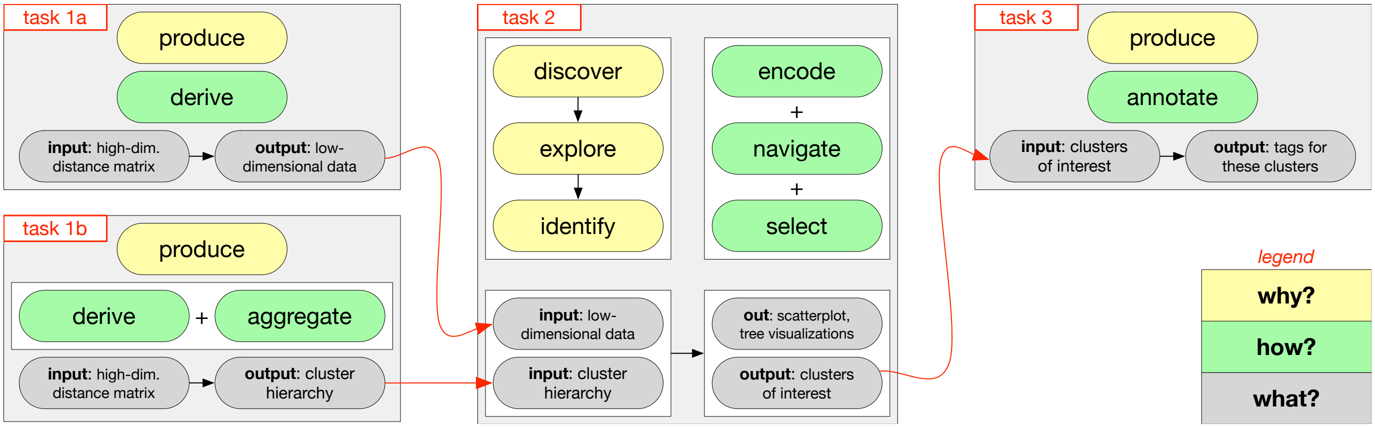
1. *visualization transformation operators*:
   dimension reduction, cluster
2. *visual mapping transformation operators*:
   tree layout, scatterplot
3. *view stage operators*:
   zoom, focus, brush, details-on-demand, pick

This description does capture the interdependencies for this sequence of tasks, though it mischaracterizes the processes of dimension reduction and clustering as transformations on a visualization, rather than transformations on data, a distinction central to our definition of `derive`. While this description captures up until the second task in our description, it does not capture the final task of producing cluster tags by means of annotation.

The description based on our typology retains the separability of these tasks, ensuring the distinction between interim inputs and outputs. Another problem with descriptions generated by extant classification systems was that of coverage; the *how* part of our typology includes both `derive` and `annotate`, while descriptions generated by other classification systems could not account for the latter [2, 13], or both [50, 61, 73, 78, 82, 83]. Finally, our description also accounts for both *why* data is `derived` and *why* clusters are `annotated` with tags, whereas descriptions generated using extant classification systems mention *how* a task is performed in relation only to *when* it is performed [13] or to *what* it is performed on [14]. We maintain that a task description requires *why*, *how*, and *what*; the question of *when* for a sequence of interdependent tasks is best served by denoting task input and output.

5 Connections to Previous Work

Our typology was informed in part by related work, including extant classification systems and established theoretical models, and in part by new thinking with many points of divergence from previous work. We surveyed work relating to tasks and models of user behaviour, spanning the research literature in visualization, visual analytics, human-computer interaction, cartography, and information retrieval. We focus on two subsets that informed the configuration of our typology: thirty works that explicitly contribute a taxonomy, typology, characterization, framework, or model of tasks, goals, objectives, intentions, activities, or interactions [2, 3, 5, 10, 11, 12, 13, 14, 15, 19, 23, 29, 31, 37, 39, 42, 50, 51, 56, 57, 59, 61, 64, 66, 72, 73, 75, 78, 82, 83], along with twenty other references that make compelling or noteworthy assertions about user behaviour [1, 4, 16, 17, 18, 27, 40, 44, 52, 54, 58, 65, 63, 67, 70, 69, 71, 79, 76, 80]. The similarities between the individual nodes of our typology and those of extant classification systems and other related work are presented in detail in Table 1.



task 1a
produce
derive
input: high-dim. distance matrix
output: low-dimensional data
task 1b
produce
derive
+
aggregate
input: high-dim. distance matrix
output: cluster hierarchy
task 2
discover
explore
identify
encode
+
navigate
+
select
input: low-dimensional data
input: cluster hierarchy
out: scatterplot, tree visualizations
output: clusters of interest
task 3
produce
annotate
input: clusters of interest
output: tags for these clusters
legend
why?
how?
what?

Fig. 3. Use of Overview [25], a visual analytics tool for exploring large text document collections, entails a sequence of interdependent tasks. Note that task 1a and task 1b can occur in either order, and that task 2 has a dependency on both of these tasks.

Table 1 serves three purposes: it documents our choices of terms for the purpose of reproducibility, it illustrates the influence of previous work on our thinking, and it summarizes overrepresented and underrepresented areas in the literature, such as consume and enjoy in the *why* part of our typology. Note that non-leaf nodes in the *how* part of our typology are poorly represented, serving to indicate the gap between low and high levels of abstraction.

5.1 Extant Classification Systems

The scope of extant classification systems can be categorized in three different ways: level, temporality, and specificity.

High vs. low: much of previous work can be divided into those having a low or high level of abstraction, with very little falling in between. Relying solely on either type of classification leads to the aforementioned ends-means confusion, thereby limiting the potential for rigorous analysis. Low-level visualization usage is well represented in related work [2, 5, 10, 12, 13, 14, 15, 19, 29, 37, 59, 61, 72, 73, 75, 78, 82, 83]. Elements common to many of these classification systems include *select*, *filter*, and *navigate*. Following Lee et al. [36], we note that low-level classifications of tasks are often conflated with those of interaction techniques. At the high level, abstract tasks can be found in the context of theoretical models, but without explicit connections to low-level visualization tasks [3, 11, 31, 39, 51]. Examples of these include *confirm hypotheses*, *present*, and *explore*. Low-level classification systems often provide a sense of *how* a task is performed, but not *why*; high-level models are the converse. Our focus on multi-level descriptions of visualization tasks is intended to close this gap and resolve the ends-means confusion.

Sequences vs. constraints: most classification systems are atemporal in that they do not have any way to express sequences or dependencies between different stages. A few classification systems are explicitly temporal in that they divide user behavior into larger stages that occur in specific sequences or cycles. Examples include pipeline models for visualization construction [13] or data analysis [27], or cyclic models such as knowledge crystallization [11] or information foraging and sensemaking [51]. However, empirical observations of visualization usage have indicated a mismatch between the specific cyclical or sequential patterns proposed by these models and actual user behaviour [26]. Vicente argues that sequence-based approaches to task analysis are overly rigid and thus inappropriate for describing such open-ended tasks [74], and that *constraint*-based approaches to task analysis allow for more flexibility in terms of *how* a task is performed. Descriptions based on our typology do not force any strict *global* temporal orderings, as imposed by sequence- or cycle-based models; instead, they accommodate *local* interdependencies within sequences of tasks by way of *constraints* on task input and output.

General vs. specific: many visualization classification systems represented in our survey are generally applicable, though specifically-targeted classification systems and models do exist. Examples include Lee et al.'s task taxonomy for graph visualization [37] and Lammarsch et al.'s task framework for time-oriented data [35].Our typology encompasses and complements these specific classification systems, and we encourage further development of more like these.

We also identify five general classification systems that span low-level and high-level tasks. These classification systems had the highest contributions to the organization of our own typology:

Springmeyer et al. (1992) [66]: This classification of scientific data analysis covers both *how* and *why*, but these aspects are not clearly distinguished within its hierarchical structure, that which begins with a high-level distinction between *investigation* and *integration of insight*.

Mullins and Treu (1993) [42]: This extensive taxonomy contains over 150 elements: an exhaustive list of high-level *mediation* and *coordination* tasks, which overlaps with our classification of *why* and *how*, as well as many low-level object-oriented interactions relating to physical interface input and output. We do not attempt to specify tasks at this lowest level, though we have adopted a consideration of input and output in the *what* part of our typology.

Pike et al. (2009) [50]: Their characterization of *analytic discourse* draws from earlier work, distinguishing high-level modes of inquiry [3] as *goals*, from low-level tasks [2] and interactions [82]. These are in turn distinguished from the separable *intents* of representation and interaction techniques. Bringing these formerly disjoint classifications together is laudable, though the integration of this information for the purpose of analyzing tasks was not the focus of Pike et al.'s article. The aim of our typology of abstract tasks is to make this integration explicit, relating these intents and techniques (*how*) to modes of inquiry, goals, and tasks (*why*).

Heer and Shneiderman (2012) [23]: Their *taxonomy of interaction dynamics* provides a top-level distinction between *data-*, *view-*, and *process*-centric tasks. The focus of their taxonomy is on interactive elements and operations; ten of the twelve task types they identify are encompassed by the *how* part of our typology. The two remaining *process and provenance* tasks, *share* and *guide*, are captured by the definition of present in the *why* part of our typology.

Roth (2012) [57]: Roth's taxonomy, based on Norman's *Stages of Action* model [47], classifies *cartographic interaction primitives* as *objectives*, *operators*, and *operands*. Norman's model describes a series of translations between the user's goal, an immediate intention (or *objective*), and a series of actions (*operators*) performed on an environment (of *operands*). Roth's taxonomy is closely aligned with our notions of *why*, *how*, and *what*, and thus has a high-level structure similar to that of our own taxonomy. However, Roth's taxonomy imposes a spatial constraint on where *operands* are located in space, as discussed in Section 2. In contrast, we restrict our classification of *what* to that of input and output; the location of *operands* is represented by the search node in the *why* part of our typology.

WHY?	
consume	–
→ present	_present_, [63, 79], _author, compose_ [11]*, _build (case), tell (story)_ [51]*, _depict_ [50]*, _express (ideas), describe_ [66]*, _guide, share_ [23]* _inform, elaborate_ [83]*, _report_ [27],
→ discover (generate, verify, hypotheses)	_discover_, [40], _explore_ [83]* [79], _verify_ [12]* [40], _synthesize_ [42]*[40], _investigate, integration (of insight)_ [66]*[40], _frame operations: construct, elaborate, question, reframe_ [31]*, _assimilate, assess, understand_ [50]*, _infer_ [73]*, _analyze_ [42, 50]*[40], _support, reevaluate (hypotheses)_ [51]*, _monitoring_ [76], _confirm (hypotheses), expose (uncertainty), formulate (cause and effect), concretize (relationships), learn (domain parameters), multivariate explanation_ [3]*, _evaluate, learn, investigate_ [40], _open-ended exploration, diagnosis_ [52], _abduction, deduction, induction_ [50], _generate, confirm (hypotheses)_ [4, 18], _integrate, interpret_ [18], _exploratory and confirmatory data analysis_ [71]
→ enjoy	visualization use in casual contexts [54, 65], _strolling_ [17]
produce	_export_ [57]*, _store_ [42]*, _save_ [39, 57]*, _extract_ [11, 61]*, _generating (images)_ [52], (a _classification_) [11, 66]*, (a _categorization_) [42, 78, 83]*, (a _record_ of one's history / process) [23, 61, 66]*
search	_search_ [11, 51, 83]*, _acquire_ [42]*, _visual queries_ [76]
→ lookup	_lookup_ [12]* [40], _identify: lookup (value)_ [78]*, _(value) lookup_ [59]*, _retrieve (value)_ [2, 37, 50, 57]*[70], _procure_ [57]*
→ browse	_browse_ [11, 42, 50, 64]*[17, 69], _search_ [57]*, _finding (gestalt)_ [10]*, _browsing tasks: follow (path)_ [37]*
→ locate	_locate_ [39, 42, 73, 78, 83]*[1, 18], _search_ [12]*[17], _search (for known item)_ [40], _seek_ [64]*, _pathfinding_ [76]
→ explore	_explore_ [39, 50, 82]*[76, 80], _forage_ [11, 39, 51]*, _finding (gestalt)_ [10]*, _(overview) tasks_ [37]*, _find (clusters, correlations, extremum, anomalies)_ [2, 37, 50]*, _determine (correlations)_ [59]*, _determine (clusters)_ [78]*
query	_query_ [56]*, _posing queries_ [10]*, _elementary and synoptic tasks_ [5]*, _levels of questions_ [72]*, _question answering_ [40]
→ identify	_identify_ [37, 42, 50, 57, 73, 78, 83]*[1, 58], _reading (the data)_ [18], _read (fact, pattern)_ [11]*, _lookup_ [5]*, _examine_ [66]*, _determine (range)_ [2, 37, 50]*, _determine / characterize (distribution)_ [2, 37, 50, 78]*, _recognize_ [31]*
→ compare	_compare_ [5, 31, 42, 50, 57, 66, 72, 73, 83]* [40], _compare (within a relation vs. across / between relations)_ [59, 78]*, _relation seeking_ [5]*, _read comparison_ [11]*, _making comparisons_ [10]*, [76], _discriminate_ [42]*, _associate_ [57]*
→ summarize	_summarize_ [83]*, _summarize (set), enumerate (set objects)_ [14]*, _overview_ [11, 15, 61]*, _(overview) tasks_ [37]*, _scan_ [37, 42]*, _connectional tasks_ [5]*, _count_ [37, 61]*, _visualization_ [17], _review_ [63]
HOW?	
encode	_encode_ [14, 50, 82, 83]*, _create mapping_ [14]*, _visualize_ [23, 73]*, _generate_ [66]*, _transform (visual mapping)_ [13]*
manipulate	_manipulate_ [80], _(object) manipulation_ [42]*, _modify_ [56]*, _(data) manipulation loop_ [76]
→ select	_select_ [23, 42, 50, 56, 72, 75, 82]*, _brush_ [19, 29, 50]*[13, 76, 80], _distinguish_ [78, 83]*, _emphasize_ [83]*, _differentiate_ [50]*, _highlight_ [15, 23, 56]* [76], _identify: portray, individualize, profile_ [83]*, _indicate_ [42, 56]*, _mark_ [42, 82]*, _reference_ [42]*, _outline (clusters)_ [83]*, _promote_ [11]*, _track_ [82]*, _pick_ [42]*[13], _express (set membership)_ [14]* _connect_ [50, 82]*
→ navigate	_navigate_ [23, 64, 75]*[40, 44, 52, 76, 80], _focus_ [10, 15]* [13], _details-on-demand_ [11, 61]*, [13], _flip through_ [13] _zoom_ [10, 11, 15, 19, 29, 42, 50, 57, 61, 82]*[13, 44, 80], _pan_ [10, 19, 42, 50, 57, 82]*[80], _elaborate_ [50, 82]*, _abstract_ [50, 82]*, _change (range)_ [19]*, _drill down_ [15]*, _maneuver / navigate_ [66]*, _rotate_ [13, 80] _revisit_ [19, 37]*
→ arrange	_arrange_ [10, 57]*, _sort_ [2, 19, 23, 37, 50]*[44], _rank_ [57, 78, 83]*, _coordinate_ [23]*, _delineate, sequence_ [57]*, _index_ [59]*, _move_ [42, 56]*, _edit_ [42]*, _organize_ [23]* [63], _orient, permute, position, translate_ [13], _reorder_ [11, 80], _configure_ [73]*, _reconfigure_ [50, 82]*, _restructure_ [39]*
→ change	_change (parameters)_ [15]* [13], _change (metaphor)_ [19]*, _change (representation)_ [15]*, _change (vis. encoding)_ [44], _transform_ [56]*[40, 80], _transform (mapping), shift, scale, set (graphical value)_ [14]*, _rotate, scale_ [13], _configure_ [73]*, _animate_ [13, 80], _distort_ [29, 75]* [13], _orient / transform_ [66]*, _(object) manipulation: transform, stretch, shape_ [42]*, _re-express_, _re-symbolize_, _re-project_ [57]*, _edit_ [42, 57]*, _activate_ [56]*
→ filter	_filter_ [2, 11, 19, 23, 29, 31, 37, 42, 50, 51, 57, 61, 82]*[44, 70, 80], _subsetting, (value) filtering, (view) filtering_ [13], _exclude_ [40, 70], _screen: filter, suppress, conceal_ [42]*, _maneuver: (data) management / culling_ [66]*, _configure_ [73]*, _delete (objects, sets, graphical objects)_ [14]*, _delete_ [11, 19, 56]*, _overlay_ [57]*, _restore_ [19, 42]*
→ aggregate	_aggregate_ [42]*[13, 44], _cluster_ [11]*[13], _associate_ [42, 78, 83]*, _simplify_ [13], _link_ [10, 15, 29, 42]*[63, 80], _merge_ [19]*, _generalize / merge_ [83]*, _assemble_ [42]*, _create (set)_ [14]*, _split_ [19]*, _disassemble_ [42]*, _disassociate_ [42]*, _reveal: itemize, separate_ [83]*, _segregate: ungroup, unlink_ [42]*, _withdraw, overlay_ [42]*
introduce	_introduce_ [42]*
→ annotate	_annotate_ [19, 23, 57]*, _add placemark_ [82], _create (anchors)_ [39]*, _create / copy (graphical objects)_ [14]*, _create / modify (note)_ [19]* _externalize (analysis artefacts)_ [63], _give a meaningful name to (groups / clusters)_ [37]*,
→ import	_import_ [57]*, _add (objects)_ [14]*, _create_ [11, 42]*, _generate_ [56]*, _(data) entry_ [42]*, _load_ [39]
→ derive	_derive_ [23]*, _derived (attributes)_ [14]*, _derive (new conditions)_ [66]*, _compute (derived value)_ [2, 37, 50]*, _copy_ [56]*, _compute_ [83]*, _calculate_ [42, 57, 66]*, _configure, determine_ [73]*, _average_ [11]* _computation operators_ [12]* _transform (data)_ [13]*, _estimate, generate (statistics)_ [66]*, _extrapolate_ [42]*[18], _interpolate_ [42]*[18]
→ record	_record_ [23, 42, 66]*, _bookmark_ [19]*, _history_ [61]*, _redo, undo_ [19, 82]*

Table 1. Nodes in our typology of abstract visualization tasks and their relation to the vocabulary used in previous work. The table serves to document our typology in detail, to illustrate the influence of previous work, and to show overrepresented and underrepresented areas in the literature. Underlining is used where a term used in our typology appears in related work. Terms in parentheses are encompassed by the _what_ part of our typology. Previous work that explicitly contributes a classification system is denoted by *; other sources incidentally make compelling or noteworthy assertions about visualization usage.

What these five classification systems have in common is that they are atemporal, and most span our characterization of *why* and *how*. Our typology integrates and extends this work, adding a specification of *what*, the input and output tasks. As a result, our typology can be used to describe sequences of tasks, in that the output of one task may serve as the input of another.

5.2 Theoretical Foundations

Our typology was also informed by four theoretical frameworks:

Distributed cognition: The distributed cognition literature offers us a useful distinction between *pragmatic* and *epistemic* actions [30, 38]. Pragmatic actions are explicitly and consciously goal-directed, while epistemic actions refer to processes of coordination between internal mental models and external representations of information [38], often performed in support of pragmatic actions. This distinction is lost in low-level classification systems; in isolation from higher-level goals we are unable to discern between pragmatic and epistemic actions. Our typology accommodates this distinction. External representations are the graphical elements of visualizations displayed to or created by the user. Pragmatic actions are correspond to the *why* part of our typology, while epistemic actions are captured by the *how* part of our typology. The set of manipulate methods are particularly well-suited for the purpose of describing epistemic actions and their role in coordinating between internal and external representations.

Stages of action: Norman's *Stages of action* model [47] and its influence on Roth's *objective-operand-operator* classification of extant classification systems [58] helped shape the *why-what-how* organization of our typology. In the process of evaluating visualizations, we can discuss the Norman's *gulf of execution* with respect to the *how* part of the typology, asking if the user has the *means* to execute the task. Also central to Norman's model is the *gulf of evaluation*, useful for reasoning about whether the output of a task matches the user's expectation. However, this gulf is more applicable when reasoning about specific interaction techniques, which are not directly addressed by our typology. More recently, Lam extended the model with a *gulf of goal formation* [33], relevant whenever a user articulates their own questions pertaining to a visualization, thereby specifying the *ends* of a task. This gulf corresponds to the *why* part of our typology, which allows us to abstractly describe these questions.

Sensemaking: The *why* part of our typology overlaps with and bridges to high-level processes of decision making and prediction described in theories of information foraging and sensemaking, both temporal stage-based models [11, 51] and atemporal data-frame models [31]. In particular, sensemaking models connect at the levels of `discover`, denoting hypothesis generation and formation, `present`, and the types of `search`: `lookup`, `locate`, `browse`, and `explore`.

Play theory: Casual interactions with visualization pose another set of problems for many extant classification systems, in that task specifications for these contexts are not easy to motivate by a need to present, discover, or produce [65]. We included `enjoy` in the *why* part of the typology to encompass casual consumption of information, curiosity-driven tasks without expectations or predicted outcomes [16, 69]. The choropleth map example from the paper's introduction is an instance of this type of task. As visualization becomes increasingly pervasive in casual contexts, we may turn to theories of casual information seeking and newsreading behaviour, such as Stephenson's *Play theory* [67] to motivate visualization tasks in these contexts. This theory accounts for media consumption activities that bring no material gain, serving no "work" functions, but instead induce moments of absorption and self-enchantment. Casual media consumption relies upon serendipitous apperception, a readiness to interact with information relating to existing interests. Studies of newsreading behaviour indicated that people read most avidly what they already know about [67], a seemingly irrational activity that cannot be described as an explicit *need* to *discover* new information. We posit that this behaviour is also true of some visualization consumption, particularly in non-work contexts [54, 65].

6 Discussion

Our motivation to develop a multi-level classification of abstract tasks grew in part from our own needs. We have specifically noted that our ability to rigorously analyze tasks in the context of the design and evaluation of visualization in general [41, 45] and of design studies in particular [60] has been constrained by its lack. While we well aware of the limits of the "reductionist assumption that complex sensemaking tasks can be broken down into low-level components" [43], we have already found that this typology has led us to conduct more precise analysis than we could do before; a forthcoming book [45] will feature many additional examples of analysis using it. We offer this new typology as a next step in the ongoing discussion in the literature, rather than as a final answer. Our efforts also serve the broader purpose of strengthening the science of analytical reasoning [68] by further uniting the frameworks and methodologies of the cognitive sciences with those in the field of visualization [53]. This work also calls for a wider range of evaluation techniques centred around task analysis, with a feedback loop in which tasks observed in field settings can inform subsequent visualization design and evaluation. Our multi-level task typology will serve to expedite this translation and analysis.

We now discuss the capabilities and potential usage of our typology in terms of its descriptive, evaluative, and generative power [7, 9].

6.1 Using the Typology to Describe

The typology's descriptive power is in its provision of a consistent lexicon for tasks in terms of *why*, *how*, and *what* in a way supports precise comparisons across different visualization tools and application domains. This lexicon can be used to describe and compare user tasks as they occur in situ, of particular use to those analyzing current work practices and use of visualization tools "in the wild". This form of inquiry is often performed within a single domain, such as enterprise data analysis [27] or intelligence analysis [28], wherein tasks are described in a domain-specific way. An interface- and domain-independent vocabulary for multi-level tasks allows practitioners to perform comparative analyses of tasks involving different visualization tools occurring in different disciplines.

Only the descriptive aspect of the typology has been directly validated in this paper; we used our typology to describe several empirical cases including single tasks and a sequence of interdependent tasks. We also demonstrated its ability to facilitate the comparison of tasks as they are performed using different visualization tools. Future work includes a further examination of its descriptive power by analyzing whether it covers the full set of abstract tasks described in previous published design studies. In addition, we acknowledge that the typology does not at present explicitly address collaborative use of visualization, although we did consider some of the issues involved during its development [26]. Future work will verify if the typology can sufficiently describe collaborative tasks, or if extensions are needed.

6.2 Using the Typology to Generate

The typology's generative power stems from its ability to prescribe and inform design. In particular, the typology is well-suited to support task analysis occurring throughout the formative *discover* and *design* stages of the nine-stage design study framework [60]. In the *discover* stage, the practitioner must transform a domain problem into an abstract task description; the typology provides an explicit set of choices for *why* the task is performed, possibly making this difficult aspect of design studies more tractable [41, 43]. In the *design* stage, the practitioner then chooses *how* the task will be supported, calling upon the existing repertoire of encoding and interaction techniques or inventing new ones [44]. During both stages, the practitioner must consider *what* comprises the inputs and outputs of these tasks, remaining aware that these tasks may have interdependencies. We aim to validate the typology's generative power in our future design studies.

6.3 Using the Typology to Evaluate

Part of the motivation for developing this typology arose from our struggle to characterize and compare the tasks of different users in an ongoing post-deployment qualitative user study evaluating

Overview [25], the system described in Section 4. The typology is intended to facilitate the evaluation of user experiences as well as field studies examining visualization usage [34]. We plan to validate the typology's evaluative power by assessing its utility a multi-level code set for labelling user behaviour, a common practice in observational studies of open-ended visualization use; these include longitudinal insight-based studies [48] and multidimensional in-depth long-term case studies (MILCs) [62]. A MILC study of SocialAction [49], a social network visualization tool, incorporated Yi et al.'s [82] categorization of interaction techniques into the analysis of *how* users tasks were performed; we envision a similar use for our task typology, in which the scope of analysis will broaden to include *why* and *what*. Mixed-method qualitative evaluation studies allow practitioners to identify *how* a task is performed along with its inputs and outputs via interaction logs and observational analysis; we can also identify *why* a task is performed via interviews, think-aloud protocols, and artefact analysis.

Task descriptions generated by our typology could be also used to better understanding users' individual analytical strategies and the context-dependent variability with regards to *how* a task is performed [74, 84]. Understanding individual problem solving strategies in terms of mental model formation and coordination is also an ongoing goal of distributed cognition research [24, 39]. Our typology and its accommodation of pragmatic and epistemic actions may serve to further this research in the context of visualization usage.

Finally, while our typology does not provide the low level of specification required for empirical experiments aimed at evaluating user performance with respect to specific interaction and visual encoding techniques [34], it may be useful in suggesting external validity for these low-level benchmark experimental tasks.

7 Conclusion

Our primary contribution is a multi-level task typology that relates both *why* and *how* a task is performed to *what* the task pertains to in terms of inputs and outputs. The typology allows for the precise description of complex tasks as sequences of simpler tasks, with their interdependencies made explicit. One major advance of the new typology is that it bridges the gap between the low-level and high-level tasks of previous work by providing linkages between them, distinguishing the ends and means of a task. Our typology integrates new thinking with extant systems for classifying tasks, and with previously established theoretical frameworks spanning multiple literatures. The multi-level task typology presented here is another step towards a systematic theoretical framework for visualization, helping us to describe existing user experiences, evaluate them, and generate new ones.

Acknowledgements

We thank NSERC for funding. We thank Jessica Dawson, Joel Ferstay, Stephen Ingram, Joanna McGrenere, Miriah Meyer, Michael Sedlmair, and Colin Ware for their feedback. We also thank François Guimbretière for very early discussions on the topic. We acknowledge Ron Rensink for the original idea of classifying lookup, browse, locate, and explore according to target identity and location.

References

[1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data.* Springer, 2011.

[2] R. Amar, J. Eagan, and J. T. Stasko. Low-level components of analytic activity in information visualization. In *Proc. IEEE Symp. Information Visualization*, pages 111–117, 2005.

[3] R. Amar and J. T. Stasko. A knowledge task-based framework for design and evaluation of information visualizations. In *Proc. IEEE Symp. Information Visualization*, pages 143–149, 2004.

[4] P. Andre, J. Teevan, and S. T. Dumais. Discovery is never by chance: Designing for (un)serendipity. In *Proc. ACM Conf. Creativity and Cognition*, pages 305–314, 2009.

[5] G. Andrienko and N. Andrienko. *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach.* Springer-Verlag, 2006.

[6] K. D. Bailey. *Typologies and Taxonomies: An Introduction to Classification.* Sage Publications, 1994.

[7] M. Beaudouin-Lafon. Designing interaction, not interfaces. In *Proc. ACM Conf. Advanced Visual Interfaces (AVI)*, pages 15–22, 2004.

[8] R. A. Becker, W. S. Cleveland, and M.-J. Shyu. The visual design and control of trellis display. *Computational and Statistical Graphics*, 5(2):123–155, 1996.

[9] B. B. Bederson and B. Shneiderman. Theories for understanding information visualization. In *The Craft of Information Visualization: Readings and Reflections*, pages 349–351. Morgan-Kaufmann, 2003.

[10] A. Buja, D. Cook, and D. F. Swayne. Interactive high-dimensional data visualization. *Computational and Graphical Statistics*, 5(1):78–99, 1996.

[11] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think.* Morgan Kaufmann, 1999.

[12] S. M. Casner. A task-analytic approach to the automated design of graphic presentations. *ACM Trans. Graphics*, 10(2):111–151, 1991.

[13] E. H. Chi and J. T. Riedl. An operator interaction framework for visualization systems. In *Proc. IEEE Symp. Information Visualization*, pages 63–70, 1998.

[14] M. C. Chuah and S. F. Roth. On the semantics of interactive visualizations. In *Proc. IEEE Symp. Information Visualization*, pages 29–36, 1996.

[15] A. Dix and G. Ellis. Starting simple: Adding value to static visualization through simple interaction. In *Proc. ACM Conf. Advanced Visual Interfaces (AVI)*, pages 124–134, 1998.

[16] M. Dörk, S. Carpendale, and C. Williamson. The information flaneur: A fresh look at information seeking. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 1215–1224, 2011.

[17] M. Dörk, N. H. Riche, G. Ramos, and S. Dumais. PivotPaths: Strolling through faceted information spaces. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 18(12):2709–2718, 2012.

[18] S. N. Friel, F. R. Curcio, and G. W. Bright. Making sense of graphs: Critical factors influencing comprehension and instructional implications. *Research in Mathematics Education*, 32(2):124–158, 2001.

[19] D. Gotz and M. X. Zhou. Characterizing users' visual analytic activity for insight provenance. In *Proc. IEEE Symp. Visual Analytics Science and Technology (VAST)*, pages 123–130, 2008.

[20] J. Grosjean, C. Plaisant, and B. B. Bederson. SpaceTree: supporting exploration in large node link tree, design evolution and empirical evaluation. In *Proc. IEEE Symp. Information Visualization*, pages 57–64, 2002.

[21] J. Heer, J. D. Mackinlay, C. Stolte, and M. Agrawala. Graphical histories for visualization: supporting analysis, communication, and evaluation. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 14(6):1189–1196, 2008.

[22] J. Heer and G. G. Robertson. Animated transitions in statistical data graphics. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 13(6):1240–1247, 2007.

[23] J. Heer and B. Shneiderman. Interactive dynamics for visual analysis: A taxonomy of tools that support the fluent and flexible use of visualizations. *Comm. of the ACM*, pages 1–26, 2012.

[24] J. Hollan, E. Hutchins, and D. Kirsh. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Trans. Computer-Human Interaction (TOCHI)*, 7(2):174–196, 2000.

[25] S. Ingram, T. Munzner, and J. Stray. Hierarchical clustering and tagging of mostly disconnected data. Technical report, Dept. Computer Science, University of British Columbia, 2012.

[26] P. Isenberg, A. Tang, and S. Carpendale. An exploratory study of visual information analysis. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 1217–1226, 2008.

[27] S. Kandel, A. Paepcke, J. M. Hellerstein, and J. Heer. Enterprise data analysis and visualization: An interview study. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 18(12):2917–2926, 2012.

[28] Y. A. Kang and J. T. Stasko. Characterizing the intelligence analysis process: Informing visual analytics design through a longitudinal field study. In *Proc. IEEE Symp. Visual Analytics Science and Technology (VAST)*, pages 19–28, 2011.

[29] D. A. Keim. Information visualization and visual data mining. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 8(1):1–8, 2002.

[30] D. Kirsh and P. Maglio. On distinguishing epistemic from pragmatic action. *Cognitive Science*, 18(4):513–549, 1994.

[31] G. Klein, B. Moon, and R. R. Hoffman. Making sense of sensemaking 2: A macrocognitive model. *IEEE Intelligent Systems*, 21(5):88–92, 2006.

[32] R. Kosara and J. D. Mackinlay. Storytelling: The next step for visualization. *IEEE Computer*, 46(5):44–50, 2013.

[33] H. Lam. A framework of interaction costs in information visualization. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 14(6):1149–1156, 2008.

[34] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Empirical studies in information visualization: Seven scenarios. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 18(9):1520–1536, 2012.

[35] T. Lammarsch, A. Rind, W. Aigner, and S. Miksch. Developing an extended task framework for exploratory data analysis along the structure of time. In *Proc. Eurographics Intl. Workshop on Visual Analytics (EuroVA)*, 2012.

[36] B. Lee, P. Isenberg, N. H. Riche, and S. Carpendale. Beyond mouse and keyboard: Expanding design considerations for information visualization interactions. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 18(12):2689–2698, 2012.

[37] B. Lee, C. Plaisant, C. S. Parr, J. D. Fekete, and N. Henry. Task taxonomy for graph visualization categories and subject descriptors. In *Proc. Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*, 2006.

[38] Z. Liu, N. Nersessian, and J. T. Stasko. Distributed cognition as a theoretical framework for information visualization. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 14(6):1173–1180, 2008.

[39] Z. Liu and J. T. Stasko. Mental models, visual reasoning and interaction in information visualization: a top-down perspective. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 16(6):999–1008, 2010.

[40] G. Marchionini. Exploratory search: From finding to understanding. *Comm. of the ACM*, 49(4):41–46, 2006.

[41] M. Meyer, M. Sedlmair, and T. Munzner. The four-level nested model revisited: blocks and guidelines. In *Proc. Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*, 2012.

[42] P. M. Mullins and S. Treu. A task-based cognitive model for user-network interaction: Defining a task taxonomy to guide the interface designer. *Interacting with Computers*, 5(2):139–166, 1993.

[43] T. Munzner. A nested model for visualization design and validation. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 15(6):921–928, 2009.

[44] T. Munzner. Visualization. In P. Shirley and S. Marschner, editors, *Fundamentals of Graphics*, chapter 27, pages 675–707. AK Peters, 3rd edition, 2009.

[45] T. Munzner. *Visualization Analysis and Design: Abstractions, Principles, and Methods*. AK Peters, 2014 (forthcoming).

[46] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou. TreeJuxtaposer: scalable tree comparison using focus + context with guaranteed visibility. *ACM Trans. Graphics*, 22(3):462–471, 2003.

[47] D. A. Norman. *The Psychology of Everyday Things*. Basic Books New York, 1988.

[48] C. North, P. Saraiya, and K. Duca. A comparison of benchmark task and insight evaluation methods for information visualization. *Information Visualization*, 10(3):162–181, 2011.

[49] A. Perer and B. Shneiderman. Integrating statistics and visualization for exploratory power: From long-term case studies to design guidelines. *IEEE Trans. Computer Graphics and Applications (CG&A)*, 29(3):39–51, 2009.

[50] W. A. Pike, J. T. Stasko, R. Chang, and T. A. O'Connell. The science of interaction. *Information Visualization*, 8(4):263–274, 2009.

[51] P. Pirolli and S. K. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proc. Intl. Conf. Intelligence Analysis*, 2005.

[52] C. Plaisant, D. Carr, and B. Shneiderman. Image-browser taxonomy and guidelines for designers. *IEEE Software*, 12(2):21–32, 1995.

[53] M. Pohl, M. Smuc, and E. Mayr. The user puzzle: Explaining the interaction with visual analytics systems. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 18(12):2908–2916, 2012.

[54] Z. Pousman, J. T. Stasko, and M. Mateas. Casual information visualization: depictions of data in everyday life. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 13(6):1145–1152, 2007.

[55] R. Rao and S. K. Card. The Table Lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 318–322, 1994.

[56] J. Raskin. *The Humane Interface: New Directions for Designing Interactive Systems*. Addison-Wesley Longman, 2000.

[57] R. E. Roth. An empirically-derived taxonomy of cartographic interaction primitives. In *Proc. GIScience*, pages 18–21, 2012.

[58] R. E. Roth. Cartographic interaction primitives: Framework and synthesis. *Cartographic Journal*, 49(4):376 – 395, 2012.

[59] S. F. Roth and J. Mattis. Data characterization for intelligent graphics presentation. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 193–200, 1990.

[60] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 18(12):2431–2440, 2012.

[61] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proc. IEEE Symp. Visual Languages*, pages 336–343, 1996.

[62] B. Shneiderman and C. Plaisant. Strategies for evaluating information visualization tools: Multi-dimensional in-depth long-term case studies. In *Proc. Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*, 2006.

[63] Y. B. Shrinivasan and J. J. van Wijk. Supporting the analytical reasoning process in information visualization. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 1237–1246, 2008.

[64] R. Spence. *Information Visualization: Design for Interaction*. Prentice Hall, 2nd edition, 2007.

[65] D. Sprague and M. Tory. Exploring how and why people use visualizations in casual contexts: Modeling user goals and regulated motivations. *Information Visualization*, 11(2):106–123, 2012.

[66] R. R. Springmeyer, M. M. Blattner, and N. L. Max. A characterization of the scientific data analysis process. In *Proc. IEEE Conf. Visualization*, pages 235–242, 1992.

[67] W. Stephenson. *The Play Theory of Mass Communication*. University of Chicago Press, 1967.

[68] J. J. Thomas and K. A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE, 2005.

[69] E. G. Toms. Understanding and facilitating the browsing of electronic text. *Intl. J. Human-Computer Studies*, 52(3):423–452, 2000.

[70] M. Tory and T. Möller. Rethinking visualization: A high-level taxonomy. In *Proc. IEEE Symp. Information Visualization*, pages 151–158, 2004.

[71] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[72] L. Tweedie. Characterizing interactive externalizations. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 375–382, 1997.

[73] E. R. A. Valiati, M. S. Pimenta, and C. M. D. S. Freitas. A taxonomy of tasks for guiding the evaluation of multidimensional visualizations. In *Proc. Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*, 2006.

[74] K. J. Vicente. *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-Based Work*. CRC Press, 1999.

[75] M. Ward and J. Yang. Interaction spaces in data and information visualization. In *Proc. IEEE Symp. Information Visualization*, pages 137–145, 2004.

[76] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 3rd edition, 2012.

[77] C. Weaver. Patterns of coordination in Improvise visualizations. In *Proc. SPIE–IS&T Visualization and Data Analysis*, 2007.

[78] S. Wehrend and C. Lewis. A problem-oriented classification of visualization techniques. In *Proc. IEEE Conf. Visualization*, pages 139–143, 1990.

[79] J. J. v. Wijk. Views on visualization. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 12(4):421–432, 2006.

[80] L. Wilkinson. *The Grammar of Graphics*. Springer, 2nd edition, 2005.

[81] W. Willett, J. Heer, and M. Agrawala. Strategies for Crowdsourcing Social Data Analysis. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 227–236, 2012.

[82] J. S. Yi, Y. A. Kang, J. T. Stasko, and J. A. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 13(6):1224–1231, 2007.

[83] M. X. Zhou and S. K. Feiner. Visual task characterization for automated visual discourse synthesis. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 392–399, 1998.

[84] C. Ziemkiewicz, S. Gomez, and D. H. Laidlaw. Analysis within and between graphs: Observed user strategies in immunobiology visualization. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 1655–1658, 2012.