Using a task classification in the visualisation design process for task understanding and abstraction: an empirical study

N. Kerracher[1], J. Kennedy[1] and K. Chalmers[1]

[1]Edinburgh Napier University, United Kingdom

Abstract

Task classifications are widely purported to be useful in the design process, with various suggestions having been made for their use at the different stages. However, little has been written regarding the actual use of task classifications in these design scenarios or reflection on the success (or otherwise) of employing them in this respect. In this paper we explore the use of a task classification at the task understanding and abstraction stages of the design process. Specifically, we use a task classification to overcome some of the known problems of eliciting tasks from domain experts during requirements gathering and as a lexicon for task abstraction. Our initial findings suggest that using a task classification helps domain experts to articulate tasks which they may not otherwise have identified. Using a task classification for task abstraction allowed us to characterise tasks in a consistent manner and organise them to establish the most commonly occurring and important tasks.

CCS Concepts

•***Human-centered computing*** → ***Visualization design and evaluation methods;*** *Empirical studies in visualization;*

1. Introduction

Understanding which analytical tasks a visualisation system needs to support and describing these domain specific tasks in the abstract language of visualisation are two important parts of the design process [Mun09, SMM12] . Understanding domain tasks is crucial as they are the foundations for determining the design of a visualisation system. The primary threat at the domain characterisation stage of the nested model is mischaracterisation: that target users do not have the problems which a visualisation tool has been developed to support [Mun09]. In addition to the wrong problems being targeted, where important tasks are missed during requirements gathering, we may end up with only a partial or trivial solution to our domain experts' problems. Task abstraction is used as the basis for selecting visual encodings. It reveals similarities between tasks that may initially appear to be rather different [Mun14] and allows us to identify categories of frequently occurring tasks, which helps determine which tools should be included in our system. It also allows us to draw on related work: it is only by abstracting tasks that we can compare our problems with those of other domains and reuse and share visual methods [WL90, MSQM13, RAW*15]. Abstraction is therefore also essential when reporting the results of design studies, in order to make findings accessible to other domains [SMM12].

Both task understanding and task abstraction are non-trivial problems [VW06, Mun09, SMM12, MSQM13, LTM18]. While many generative methods [MMAM14] for eliciting possible tasks of interest exist, each has known limitations, and little has been written regarding methods or processes designed to support task abstraction. Kerracher and Kennedy [KK17] suggest that using a task classification as a generative method for task understanding may mitigate some of the known problems in eliciting tasks, and highlight the role which task classifications can play in task abstraction as a lexicon for coding [BSIM14], to reveal frequently occurring tasks. However, little has been written regarding the actual use of task classifications in these scenarios or reflection on the success (or otherwise) of employing them in this respect.

In this paper we consider the use of task classifications as part of a systematic approach to task gathering and abstraction in the visualisation design process. We review related work in using a task classification in this way (Section 2). In Section 3, we present a case study in which we employed a task classification in the design of a visualisation system for understanding co-authorship networks. We reflect on our experiences in Section 4, highlighting opportunities for further research and setting out considerations for selecting a task classification for use in task generation and abstraction.

2. Related Work

Kerracher and Kennedy [KK17] review the role of task classifications in visualisation, including their use in the design process. They suggest that task classifications can be gainfully employed in the first three stages of Munzner's nested model [Mun09], but highlight that few examples of using task classifications for task understanding and abstraction exist in the literature.

Rind et al. [RAW*15] define a three dimensional conceptual space of user tasks and use it to classify and compare task classifications. They distinguish *perspective* (*objectives* or *actions*); *composition* (high, intermediate, or low level specification); and *abstraction* (generic or tailored towards a data type, domain, or tool architecture). Task abstraction often involves moving along both the abstraction and composition dimensions of this conceptual model: abstracting from a concrete task couched in specific domain terminology to the generic language of visualisation, and decomposing a high-level task into smaller subcomponents. How this process is carried out is not well documented. The sequence of task abstraction and decomposition is an interesting question: do we abstract or decompose first, or zig-zag between each? While not prescriptive, the description in [MSQM13] implies that decomposition is carried out before abstraction. Lam et al. [LTM18] also propose this order.

It is not always clear what level of composition is appropriate during task abstraction. Amar and Stasko [AS05] warn against representational primacy—where designers provide support for low-level visual tasks such as 'correlate', 'identify' or 'distribute'—as this may fail to provide support for important high-level analysis tasks, such as facilitating hypothesis testing or exposing uncertainty in the data. Mapping visual encodings to tasks specified at this higher level could of course prove challenging. Meyer et al. [MSQM13] call for intermediary level task classifications to bridge this gap. However, support for low-level tasks may still require consideration when designing a visualisation system as they often facilitate understanding of our wider goals: Andrienko [AA06] note that while low-level tasks do not play a primary role in exploratory data analysis they may contribute significantly to our overall understanding when exploring data. Lam et al.'s analysis goals framework [LTM18] is specifically designed to bridge between high-level analysis goals and low-level tasks of existing classifications.

In terms of using task classifications for abstraction, Brehmer et al. [BSIM14] suggest using their classification as a "lexicon" for coding and translating observed domain tasks when understanding existing work practices during the design process. However, they do not report in detail about the process of using their classification in this way. Lam et al. [LTM18] propose using their framework of high-level analysis goals to identify units of analysis relating to individual goals. These can be broken into analysis steps at an appropriate level for translation using existing task classifications.

The use of task classifications in supporting the generative phase of task understanding to mitigate some of the known problems associated with existing methods—such as gathering the wrong, or an incomplete set of tasks [KK17], difficulties capturing 'undreamed of' requirements [LD11], or discussions being derailed into a focus on visual solutions [SMM12]—has recently been suggested [KK17]. Tory and Möller [TM04] highlight the problem of eliciting tasks through introspection, noting that "*users may not notice what they do, may not know how to articulate what they do, and may misrepresent reality*". They therefore suggest carrying out task analysis in the context of real work. However, this requires some method for tackling the problem to already exist. This is a general problem for observational methods of task gathering, which also relies upon people being able to articulate their internal mental processes when performing cognitive tasks [SMM12]. An alternative to introspection and observation is to extract tasks from the literature, but this requires that a similar problem has already been tackled, and may require familiarity with domain terminology. Using task classifications to set out the range of possible tasks may help overcome these problems [KK17]: Rind et al. [RAW*15] suggest the use of abstract task lists to check if any relevant tasks still need to be addressed when designing and evaluating visualisations. Generating tasks using a task classification is essentially the opposite of task abstraction, in that we move from an abstract task to a domain specific task. However, it does not normally involve movement along the composition dimension.

McKenna et al. [MMAM14] present a framework which introduces four 'design activities' which map to the levels of the nested model [Mun09]. They review over 100 methods, many of which are *generative* (intended to be divergent and create many outcomes e.g. brainstorming) and appropriate to the *understand* activity which encompasses both task generation and abstraction. However, they do not consider using task classifications for this purpose.

Two studies use task classifications as a generative method: [AS05] and [APS14]. Amar and Stasko [AS05] demonstrate the use of six knowledge precepts in a hypothetical design scenario as "*a systematic basis for thinking about and identifying issues in the data set*". Part of this process uses the precepts to posit potential tasks of interest. However, they do not involve domain experts in this process to confirm whether these tasks are of interest, therefore it is difficult to assess the effectiveness of using their classification as a generative method. Part of Ahn et al.'s evaluation of their temporal graph task taxonomy [APS14] seeks to assess its usefulness to domain experts in discovering tasks of which they had not originally thought. Twelve experts drawn from different research areas were asked to list their tasks, then compare these to the task taxonomy. They were asked to review any newly discovered tasks and use a Likert scale to grade the taxonomy in terms of its ability to support task discovery. Around 70% of experts rated the taxonomy positively in this regard. While this study uses a task classification as a generative method, it does so in the context of an evaluation of a taxonomy rather than in a design scenario. The experts were drawn from different domains and they consider the tasks of interest to their own individual fields. The taxonomy is used directly by the experts which is not typical of a design scenario, as task classifications are often designed for use by—and therefore may only be intelligible to—those with a visualisation background [SNHS13].

3. Case Study

In this section we present an empirical study where we investigate the use of a task classification as a coding scheme during task abstraction, and as a generative method to produce tasks of potential interest to domain experts. We used a task taxonomy for temporal graphs [KKC15], which is an extension of a formal task framework [AA06] and is classified by Rind et al. [RAW*15] as dealing with objectives, data specific in abstraction, and specified at a low to intermediate level of composition. The taxonomy is structured around three main task types (lookup, comparison, and relation seeking), the data items which may participate in the tasks (elements, graph at a single time, element over time, graph over time or time over graph, relational behaviours), and whether structure and/or attributes are involved. Visual tools corresponding to the structure of this taxonomy are set out in [KKCG15].

Our design scenario is that of developing a visualisation tool to help academics explore their department's co-authorship network in order to better understand collaborative working practices and publishing rates within their department. 12 academics—who could be considered domain experts—participated in the study. They were provided with a data set consisting of publications data for approximately two-hundred authors and two thousand publications, spanning a period of over thirty years. A description of the data and an illustrative excerpt were included in the instructions to participants (see Annex).

The study was divided into two parts, both of which were conducted by email. In part 1, the experts were presented with the analysis scenario and data set. They were asked to consider the data and note any questions of interest to them, and rate how interesting each of their questions were on a scale of 1-4 (where 1=least interest and 4=most interest). We abstracted these questions using the high-level categories of the task classification as a coding scheme. This process revealed a number of task 'gaps'—task categories for which none of the experts had identified a task. For part 2, a selection of the identified task gaps were presented, and the experts were asked to rate how interesting they found them using the original scale, with the addition of 0 to indicate that a task was of no interest.

The study was designed to evaluate the use of a classification in the design process with respect to two scenarios: (1) Use in the process of task abstraction: *can the task classification act as a useful means for abstracting and organising domain specific tasks?* and (2) Use as a generative method for task understanding: *can the task classification be used to discover tasks of interest to experts which they had not considered?*

3.1. Task Abstraction

The experts' tasks returned in part 1 were categorised according to high level dimensions of the framework. A total of 72 tasks were identified (mean=6; max=12; min=0). Just over half (36/71; 51%) were rated as very or extremely interesting. However, five experts explicitly commented that the data used in the study was of limited interest, as it lacked information on research topics and publication quality. This group of experts ("limited interest experts") returned fewer questions (mean of 3.2 vs. mean of 6 per expert overall, and 8 in the "interested expert" group). They also generally rated them to be of less interest, with only one third of questions (5/15; 33%) rated as very or extremely interesting vs. over 50% (31/56; 55.36%) in the interested expert group (Annex Figure 1).

The experts' questions were abstracted using the task classification as a coding scheme. The list of experts' tasks, their categorisation under the task framework and, where necessary, explanations of how the categorisation was reached, are included in the Annex. During task abstraction, contra to [MSQM13] and [LTM18], our general strategy was to abstract before decomposing, first identifying and mapping domain specified data components to their abstract counterparts e.g. authors became nodes in a graph. We then considered the task in relation to the types of tasks in the framework e.g. for the task "*at what point in their time within the department do authors start producing collaborative work with others?*" we abstracted "*start producing collaborative work with others*" as "*the appearance of a pattern of an increase in connectivity in the network over time*" which we mapped to a pattern search task involving the configurations of nodes over time. When decomposing tasks, it was often possible to describe them either as a sequence of low level tasks or as a single higher level (synoptic) task. We gave preference to the latter, as this better reflects what an analyst is trying to achieve (e.g. pattern understanding as opposed to comparison of individual data values) and the visual techniques which are known to support these tasks e.g. the task, "*is anyone in the wrong research centre going by their paper collaborations?*" can be abstracted as looking for nodes in the graph which are mostly connected to nodes having different attribute values. This could either be decomposed into multiple individual comparisons between the attribute values of connected nodes, or defined as looking for a pattern of attribute values over the graph structure; the latter definition is more likely to lead us to select e.g. a visualisation which shows the network with attributes encoded on the nodes. Eight tasks were not coded: four required further clarification and four involved attributes not included in the data. The process of task abstraction was not trivial, and required several iterations to ensure consistency, however by using the classification in this way, it was possible to organise tasks into similar types and identify several "task gaps".

3.2. Task Generation and Discovery

For each task gap, a generic task description was constructed, along with illustrative concrete examples. Images were also used to help describe what was intended by the task description (see Annex). Experts were instructed that these were for illustration only; they had been constructed using synthetic data and there may be more appropriate ways to visualise the data. Using the framework to generate tasks was more straightforward than abstracting tasks: we systematically constructed tasks for each attribute in turn. However, this produced a huge number of potential tasks. It became apparent during piloting that the amount of time required to consider every task gap was far more than could reasonably be expected of our volunteers, therefore we selected a subset of 16 task gaps to explore.

10 of the original 12 experts completed part 2 of the study, as two (from the 'interested' group) were unavailable. Of the 16 proposed tasks, all were found to be of some level of interest to the experts collectively (Annex Figure 2). Of the 159 ratings returned, over one third (38%) were very or extremely interesting. Only 8 ratings (5%) of no interest were returned. 47% of the limited interest group's ratings were very or extremely interesting, compared to 29% in the interested group. All "no interest" ratings were returned by the interested group. As would be expected, some tasks were found to be more interesting than others (Figure 1 & Annex Figure 3). All tasks were thought to be of some level of interest, with the "no interest" ratings spread over eight separate tasks (as opposed to being directed at a single task of very limited interest to participants). Over a third of the tasks were thought to be very or extremely interesting by half of the experts.

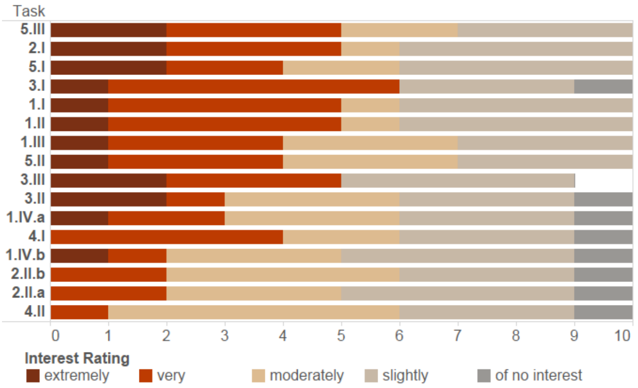

Figure 1: *Count of interest ratings for each proposed task, in descending order by interest level. Task details can be found in Annex Figure 3. All tasks were of some level of interest, with over one third rated very or extremely interesting by half of the participants. Eight "no interest" ratings were spread over eight separate tasks.*

4. Discussion

We managed to abstract and categorise tasks using the classification as a coding scheme, however, we did not find this process straightforward and it was time consuming. This could in part be due to the size and complexity of the task classification we used, and its low level of composition. It may also have been due to conducting the study via email: as we did not directly discuss the experts' tasks, we had to interpret their meaning. We identified a difference between decomposition—breaking a complex task into its constituent parts—and clarification—which requires more information e.g. for the task "*who would I be able to help*", are the authors who could be helped those with few co-authors or low publication counts etc.? We found using a task classification particularly helpful in highlighting potential ambiguities, an artefact of natural language e.g. "*who is still currently in the department?*" could mean '*which members of staff are currently in the department?*' or '*which current members of staff were also present in previous year(s)?*'. It also helped identify where tasks were not fully specified e.g. many tasks made no reference to time: "*who's working with whom?*" could potentially be asked of a specific year or a time period. In some cases, tasks required data which was not available in our dataset. While the process was not easy, whether we could have effectively grouped similar tasks without the use of a classification as a coding scheme we cannot say. We can say with confidence that without the task classification, it would have been difficult to identify task gaps.

All of the task gaps were found to be of some level of interest to participants. At least one third were rated as very or extremely interesting by at least half of the experts; this indicates that using the classification in this way can discover not only tasks of passing interest, but also those which could potentially be important to people carrying out an analysis. One unexpected observation was that experts who expressed only limited interest in the data in part 1 rated the proposed tasks as more interesting than participants in the interested group. Further, only those in the interested group returned ratings of "no interest". One possible explanation is that the interested participants had a clearer idea of tasks of interest at outset, therefore they had already articulated the tasks of most interest to them. Another is that those less interested in the data had not been able to anticipate the range of possible questions it might help them answer, as per the known difficulty of capturing 'undreamed of' requirements [LD11].

While it may pay to invest time in task understanding and we found using the task classification as a generative method straightforward, one issue is the volume of tasks which can be constructed, and therefore the time it takes to specify exemplar tasks and for experts to consider them. Again, this may be due in part to the level of detail in the classification we used, and may be less of an issue for simpler frameworks or those specified at a higher level of composition. However, further research is required to establish whether it is possible to generate specific concrete tasks using a classification specified at a high level. Taking a staged approach which first establishes the general tasks of interest before considering specific tasks may help. Our study was email based, but the tasks generated using a task classifications could potentially be used in conjunction with other methods e.g. as the basis for discussions in structured interviews or focus groups, or as inputs for card sorting [SA15].

When selecting a task classification for use in the design process, classifications dealing with *objectives* are most useful in the early stages where tasks are established, whereas those dealing with *actions*—the discrete steps necessary to address objectives, such as mapping data items to specific visual encodings—are more appropriate later in the process, when designing visual encodings and interactions. Selecting a task taxonomy specifically developed for the type of data with which we are dealing or our problem domain would be a sensible choice, however these may not exist for all data types and domains. As discussed above, the level of composition of the task classification is another important consideration. When selecting a classification we should also consider its validity in terms of its properties such as its descriptive powers, comprehensiveness, and usability [KK17]. For example, classifications which are incomplete may lead to missing tasks when used as a generative method for task understanding, while large or complex classifications may come with a significant learning overhead. Unfortunately, given the lack of current evaluation practices, it may be difficult to make these decisions prior to adopting a classification.

5. Conclusions

In this paper we demonstrated the use of a task classification as a generative method at the domain problem characterisation stage of the design process. We also described how we used the classification to abstract tasks from domain specific language into a format suitable for selecting visual encodings, and documented the challenges we encountered. We have shown that using a task classification to elicit tasks overcomes a number of known problems associated with other strategies, including finding important tasks which people otherwise would not have reported. While these strategies could benefit from further research, we hope that this paper will encourage visualisation designers to consider using task classifications to help them at these important stages of the design process.

References

[AA06] ANDRIENKO N., ANDRIENKO G.: *Exploratory analysis of spatial and temporal data: a systematic approach*. Springer, New York, 2006. 2

[APS14] AHN J.-W., PLAISANT C., SHNEIDERMAN B.: A task taxonomy for network evolution analysis. *IEEE Transactions on Visualization and Computer Graphics 20*, 3 (2014), 365–376. 2

[AS05] AMAR R. A., STASKO J. T.: Knowledge precepts for design and evaluation of information visualizations. *IEEE Transactions on Visualization and Computer Graphics 11*, 4 (2005), 432–442. 2

[BSIM14] BREHMER M., SEDLMAIR M., INGRAM S., MUNZNER T.: Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization* (2014), ACM, pp. 1–8. 1, 2

[KK17] KERRACHER N., KENNEDY J.: Constructing and Evaluating Visualisation Task Classifications: Process and Considerations. *Computer Graphics Forum 36*, 3 (2017). 1, 2, 4

[KKC15] KERRACHER N., KENNEDY J., CHALMERS K.: A Task Taxonomy for Temporal Graph Visualisation. *IEEE Transactions on Visualization and Computer Graphics 21*, 10 (2015), 1160–1172. 2

[KKCG15] KERRACHER N., KENNEDY J., CHALMERS K., GRAHAM M.: Visual Techniques to Support Exploratory Analysis of Temporal Graph Data. In *Eurographics Conference on Visualization (EuroVis) - Short Papers* (Cagliari, Sardinia, 2015), Bertini E., Kennedy J., Puppo E., (Eds.), The Eurographics Association. 3

[LD11] LLOYD D., DYKES J.: Human-centered approaches in geovisualization design: Investigating multiple methods through a long-term case study. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2498–2507. 2, 4

[LTM18] LAM H., TORY M., MUNZNER T.: Bridging from goals to tasks with design study analysis reports. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 435–445. 1, 2, 3

[MMAM14] MCKENNA S., MAZUR D., AGUTTER J., MEYER M.: Design Activity Framework for Visualization Design. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (dec 2014), 2191–2200. 1, 2

[MSQM13] MEYER M., SEDLMAIR M., QUINAN P. S., MUNZNER T.: The nested blocks and guidelines model. *Information Visualization 14*, 3 (2013), 234–249. 1, 2, 3

[Mun09] MUNZNER T.: A Nested Model for Visualization Design and Validation. *Visualization and Computer Graphics, IEEE Transactions on 15*, 6 (2009), 921–928. 1, 2

[Mun14] MUNZNER T.: *Visualization analysis and design*. CRC press, 2014. 1

[RAW*15] RIND A., AIGNER W., WAGNER M., MIKSCH S., LAMMARSCH T.: Task cube: A three-dimensional conceptual space of user tasks in visualization design and evaluation. *Information Visualization* (2015). 1, 2

[SA15] SAKAI R., AERTS J.: Card Sorting Techniques for Domain Characterization in Problem-driven Visualization Research. *Eurographics Conference on Visualization (EuroVis) - Short Papers* (2015). 4

[SMM12] SEDLMAIR M., MEYER M., MUNZNER T.: Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (Dec 2012), 2431–2440. 1, 2

[SNHS13] SCHULZ H.-J., NOCKE T., HEITZLER M., SCHUMANN H.: A design space of visualization tasks. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2366–2375. 2

[TM04] TORY M., MOLLER T.: Human factors in visualization research. *IEEE Transactions on Visualization and Computer Graphics 10*, 1 (2004), 72–84. 2

[VW06] VAN WIJK J. J.: Bridging the gaps. *IEEE Computer Graphics and Applications 26*, 6 (2006). 1

[WL90] WEHREND S., LEWIS C.: A problem-oriented classification of visualization techniques. In *Visualization, 1990. Visualization'90., Proceedings of the First IEEE Conference on* (1990), IEEE, pp. 139–143. 1