A Design Space of Visualization Tasks

Hans-Jörg Schulz, Thomas Nocke, Magnus Heitzler, and Heidrun Schumann

Abstract—Knowledge about visualization tasks plays an important role in choosing or building suitable visual representations to pursue them. Yet, tasks are a multi-faceted concept and it is thus not surprising that the many existing task taxonomies and models all describe different aspects of tasks, depending on what these task descriptions aim to capture. This results in a clear need to bring these different aspects together under the common hood of a general design space of visualization tasks, which we propose in this paper. Our design space consists of five design dimensions that characterize the main aspects of tasks and that have so far been distributed across different task descriptions. We exemplify its concrete use by applying our design space in the domain of climate impact research. To this end, we propose interfaces to our design space for different user roles (developers, authors, and end users) that allow users of different levels of expertise to work with it.

Index Terms—Task taxonomy, design space, climate impact research, visualization recommendation

1 Introduction

As the field of information visualization matures, a phase of consolidation sets in that aims to pull together multiple individual works of research under a common conceptual hood. This hood can take on different shapes and forms, one of which is the *design space*. Such a design space realizes a descriptive generalization that permits to specify a concrete instance – be it a layout [8], a visualization [46], or a combination of visualizations [28] – by making design choices along a number of independent design dimensions. Even last year's InfoVis conference recognized the increasing importance of design spaces by dedicating an entire session to them.

Yet, information visualization is more than the visual representation alone. It also takes into account the tasks the user wishes to pursue with the visual representation. The literature contains a wealth of classifications, taxonomies, and frameworks that describe these tasks: lists of verbal task descriptions, mathematical task models, domain-specific task collections, and procedural task combinations into workflows. All of these serve the respective purpose well for which they have been developed. However, the research question of how to consolidate them under the hood of one common design space is still open, even though it has been shown on a smaller scale that such a combination into a common framework can be a useful endeavor [9, 21].

In this paper, we aim to give a first answer to this research question by contributing such a design space for visualization tasks. This contribution is twofold. First, it derives an abstract design space that brings together the different aspects of the existing task taxonomies and models. It serves to clarify the somewhat fuzzy notion of visualization tasks and by that it permits judging the suitability and compatibility of individual task taxonomies for a given case at hand. Second, this abstract design space can also be instantiated in its own right for its utilization in concrete use cases. The latter is achieved by providing role-dependent interfaces to the design space that allow *developers* to derive application-dependent design subspaces for *authors* to compose compound tasks and workflows in them, so that *end users* can select them to customize their visualization outcome. We exemplify this by a visualization task design space for climate impact research based on structured interviews with eight domain experts and two visualization developers. This design space is then utilized to recommend visualizations that are suitable to pursue a given task in that field.

The remainder of this paper is organized as follows: The related work is summarized in Section 2 and from its discussion, we derive our task design space in Section 3. We then debate its properties, limitations, and applications in Section 4. This also includes examples of how some of the existing task taxonomies can be expressed as parts of our design space. After this conceptual part, Section 5 details the use case example of how to apply the general design space to the application domain of climate impact research and how to draw concrete benefits from it. With this example, we aim to show a feasible way for the adaptation of the design space that can be transferred to other application domains as well. We conclude this paper by briefly sharing our personal experience from working with the design space and pointing out directions for future work in Section 6.

- *Hans-Jörg Schulz is with the University of Rostock. E-mail: hjschulz@informatik.uni-rostock.de.*
- *Thomas Nocke is with the Potsdam Institute for Climate Impact Research. E-mail: nocke@pik-potsdam.de.*
- *Magnus Heitzler is with the Potsdam Institute for Climate Impact Research. E-mail: heitzler@pik-potsdam.de.*
- *Heidrun Schumann is with the University of Rostock. E-mail: schumann@informatik.uni-rostock.de.*

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.

2 Related Work

The concept of *tasks* exhibits numerous facets that are also reflected in the existing body of research on that topic. Commonly, *visualization tasks* are understood as activities to be carried out interactively on a visual data representation for a particular reason. The investigation of visualization tasks has the aim to **establish recurring tasks** in order to use the knowledge about them for improving the **design and evaluation of visualizations**. Existing research for both of these aspects is briefly summarized in the following.

2.1 Establishing Recurring Visualization Tasks

The literature describes a number of different ways of how to obtain recurring visualization tasks. The most prevalent methods are to simply **survey** a large enough number of visualization-savvy individuals to list their tasks [4], to conduct a full-fledged **task analysis** by observing visualization users [25], or to **infer** from various existing visualization systems which tasks they facilitate [2]. Regardless of the chosen method, the end result is a set of frequent tasks or appropriate generalizations thereof. The scope of this set depends on the concrete notion of "visualization task" that is employed. In the literature, one finds a spectrum ranging from rather strict interpretations that only permit perceptual tasks [16, 26] to broader interpretations that even include non-visual analytical tasks [49].

Once collected, there exist various ways to describe visualization tasks. Most task descriptions are **verbal**, dedicating a paragraph or two to each task's explanation – e.g., [5, 24, 49]. Others are **functional** with tasks being either functions [12] or queries about functions [7]. Further ways to describe tasks are in a **logical** [16] or a **faceted** manner [10, 40]. The latter breaks down the description into various elementary facets that together specify a concrete task. Most of these descriptions are in addition also **hierarchical**, which means that they

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

allow for representing larger tasks as a sequence of smaller subtasks, as it is common in task modeling [22, 50, 51].

With so many different task taxonomies in existence, the question of their consolidation arises. If at all, this question is answered in two fundamental ways in the literature: either **top-down** by putting the taxonomies in the context of a larger meta-taxonomy [13, 14, 42], or **bottom-up** by grounding them in the concrete necessities of a particular type of data [31] or a particular application domain [37]. Most top-down approaches rely on the *Task-by-Data-Type Taxonomy (TTT)* [47], whereas the bottom-up approaches are either based on extensions of the taxonomy by Wehrend and Lewis *(TWL)* [60] or they explicitly mention a striking similarity to it [31]. Both choices make sense and do not pose a contradiction, as the TTT is known to be more high-level and system-centric, while the TWL is rather low-level and user-centric. Their complementary use is further underlined by the research described in [9] and [21], which independently try to combine both into a common task taxonomy.

This observation that the TTT and the TWL form quasi-standards in the field has to be taken with a grain of salt, though, as they may owe their status simply to the fact that they have been around for some time. Recently, more modern task taxonomies appear to supersede them – e.g., the more formal task description by [7] as a low-level taxonomy and the broader list of tasks by [43] as a high-level taxonomy. Both exceed a pure visualization task taxonomy by embracing analytical tasks and become increasingly popular as starting points for task research in the context of Visual Analytics.

2.2 Utilizing Tasks for Visualization Design and Evaluation

Tasks stand in relation to the input data they are concerned with and to the visual representation they are performed on. In the same spirit as for task taxonomies, there also exists literature on taxonomical research for data [45, 64] and for visualizations; the latter being split into characterizations of visual representations [15, 17, 20, 38] and of the interaction with them [18, 30, 59, 62]. It is thus only natural, that these concepts have been combined in two ways:

- **Data + Task = Visualization?** This combination asks which visualization is best suited to pursue a given task on given input data. It caters directly to the **visualization design**.
- **Data + Visualization = Task?** This combination asks which tasks can be pursued (how well) on a given visualization for a given dataset. It caters directly to the **visualization evaluation**.

As for the visualization design, there are a number of different aspects of a visualization that can be chosen or parametrized with the help of task knowledge. This ranges from informal guidance on the overall visualization design [53, 55] to very concrete suggestions for individual aspects, such as appropriate mappings [64] or color scales [3, ch.4.2.2]. Some works, such as [29], even provide a look-up for all possible data/task combinations to recommend visualization techniques to the user. Despite the research in this area, such automated visualization design recommendations have never been achieved to fullest extent and are an open research question until today.

The evaluation of visualizations is the second important use of tasks. By evaluating visualizations with respect to an agreed upon set of tasks, they become comparable. Such an evaluation is usually not a binary one of whether a visualization supports a particular task or not, but rather how well it supports it as measured through completion times and error rates. The most common way of employing this approach is to use it on a per-technique basis, as facilitated by [6, 37, 57]. Yet, the literature also presents more holistic approaches, such as using the tasks to "sketch the space of the possible" [56] to observe which parts of this space are already covered and which parts still need further research or design efforts [2].

We follow this idea of the *space of the possible* by establishing our design space of visualization tasks in the following section. For its specification, we use a *faceted task description* (cp. Sec. 2.1) that combines the most important aspects of visualization tasks from various different task taxonomies in one integrated task concept.

3 Construction of our Design Space

The existing taxonomies have been established with different goals in mind. For example, the quasi-standard taxonomies TTT and TWL both tie-in with certain aspects of the data and they both aim to capture the user's action – with the TWL being more on the intentional, user's side and the TTT being more on the technical, system's side. On top of that, the TTT even provides a *process-centric* view by defining a particular sequence of tasks as a *mantra* to codify it as a common recurring workflow in visualization. Each of these aspects captures an important part of what a visualization task is and it is our aim to identify these different aspects and bring them together as dimensions of an integrated visualization task design space.

3.1 Preliminaries

Design spaces are hypothetical constructs that follow the mechanistic belief that the whole is the sum of a number of independent parts. The identification of these parts and their interplay is challenging, as the mechanistic world view rarely fully matches reality. Yet once they are established, the parts can be put together in any possible way. By this, they do not only describe the existing, but also the (so far) not existing through novel combinations of the parts. In this aspect, design spaces inherently differ from taxonomies, which are by definition classifications of what actually exists.

A design space consists of a finite number of design dimensions, which each capture one particular design decision that has to be made to fully specify the whole. For each such design dimension, a possibly infinite number of design choices are available to choose from. Design spaces have been applied in visualization research for:

- consolidating existing research under one common hood, where they form individual designs or design subspaces,
- identifying blank areas as potential research opportunities – i.e., design combinations that have not yet been investigated, and
- externalizing the often implicit design decisions for improving communication – with students (teaching), with clients (requirements analysis), and with software (API specification).

A design space of visualization tasks provides all of the above, but it is also useful at a fundamental level for working with tasks, as its design dimensions are **customizable** and thus **inclusive**. This means that while the design space as a whole is fixed, individual design choices can be added or further subdivided on any design dimension, as it may be needed to achieve a necessary level of granularity for a particular application (cp. Sec. 4.1.2). As a result, the design space captures all tasks – whether they are abstract, specific, or downright unusual.

3.2 The Design Space Dimensions

The dimensions of our design space basically relate to the "5 W's" of WHY, WHAT, WHERE, WHO, and WHEN, as well as to the often appended HOW. These aspects are frequently used to describe a matter from its most relevant angles in technical documentation and communication, and they have been used in visualization [63], as well as for task analysis [23] and for grasping user intentions [1]. For describing tasks, they call for answers to the following questions:

- WHY is a task pursued? This specifies the task's **goal**.
- HOW is a task carried out? This specifies the task's **means**.
- WHAT does a task seek? This specifies the data **characteristics**.
- WHERE in the data does a task operate? This specifies the **target**, as well as the **cardinality** of data entities within that target.
- WHEN is a task performed? This specifies the order of tasks.
- WHO is executing a task? This specifies the (type of) user.

The latter two aspects are clearly not inherent fundamental properties of an individual task itself, but aim to relate a task to its context. For the WHEN, this is the context of preceding and succeeding tasks in a task sequence, and for the WHO, this is the context of capabilities and responsibilities to perform tasks in a collaborative environment. That is why we discuss these two aspects in later sections that consider workflow (Sec. 3.3.3) and user context (Sec. 4.1.3) in their own

right. This leaves the five dimensions of goal, means, characteristics, target, and cardinality, which are introduced in the following, before combining them into our design space in Sec. 3.3.

3.2.1 Goal

The *goal* of a visualization task (often also *objective* or *aim*) defines the intent with which the task is pursued. We differentiate between the following three high-level goals:

- **Exploratory analysis** is concerned with deriving hypotheses from an unknown dataset. It is often equated with an *undirected search*.
- **Confirmatory analysis** aims to test found or assumed hypotheses about a dataset. In analogy to an undirected search, it is sometimes described as a *directed search*.
- **Presentation** deals with describing and exhibiting confirmed analysis results.

It is important to note that these goals specify the motive of a task's action and not an action itself, which is defined by the "means" in the following section. These aspects are independent, as the same motive can be pursued through different actions and the same action can be performed for different motives.

3.2.2 Means

The *means* by which a visualization task is carried out (often also *action* or *task*) determines the method for reaching the goal. It is challenging to present a definite list of such means to achieve a task. We have extracted the following list of means from the literature and while it may not cover each and every possible way of conducting a task, it serves well as an initial set that can be extended if needed:

- **Navigation** subsumes all means that change the extent or the granularity of the shown data, but that do not reorganize the data itself. For the extent this is done, for example, by *browsing* or *searching* the data, whereas for the granularity this is done by *elaborating* or *summarizing* the data.
- **(Re-)organization** includes all means that actually adjust the data to be shown by either reducing or enriching it. Common means of data reduction are *extraction* (e.g., filtering or sampling) and *abstraction* (e.g., aggregation or generalization) [33], while often used means of enrichment are *gathering* additional data from external sources and *deriving* additional metadata.
- **Relation** encompasses all means that put data in context. This is done by seeking similarities through *comparison*, by seeking differences when looking for *variations* or *discrepancies*, or by their inverse query known as *relation-seeking* [7].

We have purposefully chosen a more abstract terminology for our means, so that they are independent of their concrete technical realization in a visualization system. For example, an "elaborate" can be realized through a *zoom-in*, a *magic lens*, a *drill-down*, or any other available interactive feature that has the desired effect.

3.2.3 Characteristics

The *characteristics* of a visualization task (often also *feature* or *pattern*) capture the facets of the data that the task aims to reveal. While these depend highly on the type of data that is being visualized, we can distinguish between two general kinds of characteristics:

- **Low-level data characteristics** are simple observations about the data, such as *data values* of a particular data object (look-up, identification) or the *data objects* corresponding to a particular data value (inverse look-up, localization). These can be commonly obtained by looking at legends, labels, or coordinate axes.
- **High-level data characteristics** are more complex patterns in the data, such as *trends*, *outliers*, *clusters*, *frequency*, *distribution*, *correlation*, etc. Obtaining such characteristics takes usually more effort to acquire the necessary "big picture" of the data.

Thus the low-level data characteristics are what can simply be "read" from the visualization (cp. visual literacy), while the high-level data characteristics must be "deduced" from it (cp. visual analysis).

3.2.4 Target

The *target* of a visualization task (often also *data facet* or *data entity*) determines on which part of the data it is carried out. In order to accommodate a broad range of data types, we build on the ideas of [7] and take a generic relational perspective on data to describe its different aspects that may be referred to by a task. This way, it is compatible with a variety of data models on the technical level, ranging from traditional relational databases to more contemporary triple-based storages. The different aspects of data can be any of the following relations:

- **Attribute relations** link data objects with their attributes. These include in particular:
  - **Temporal relations** linking data objects to attributes that are time points or time intervals, and
  - **Spatial relations** linking data objects to attributes that are points, paths, or areas in (geographical) space.
- **Structural relations** link data objects with each other, which can have various reasons, such as causal relations, topological relations, order relations, equivalence relations, etc.

This relational view on data puts a particular emphasis on the relational means discussed in Sec. 3.2.2. No longer are these only concerned with deriving relations, but also with querying the relations given in the data. For example in a social network, it is an interesting task in itself to investigate which kinds of relationships among persons (friendship, kinship, partnership, etc.) are already inherent in the data.

3.2.5 Cardinality

The *cardinality* of a visualization task (often also *scope* or *range*) specifies how many instances of the chosen target are considered by a task. This distinction is important, as it makes a difference whether only an individual instance is investigated or all of them. It is notable that a number of existing taxonomies deem the cardinality of a task to be an important aspect as well. Its notion can be found, for example, in Bertin's *Levels of Reading* [11], in Robertson's distinction between *point, local, and global* [44], in Yi et al.'s *individual, subgroup, group* [61], and in Andrienkos' *elementary vs. synoptic tasks* that even form the topmost classes in their task categorization [7]. Hence, we singled out this aspect of the WHERE to form its own design dimension, differentiating between the following options:

- **Single instance**, e.g., for highlighting details.
- **Multiple instances**, e.g., for putting data in context.
- **All instances**, e.g., for getting a complete overview.

The descriptions of these choices already hint at numerous common visual analysis patterns, such as *Overview and Detail* or Shneiderman's *Information Seeking Mantra* [47] (cp. Sec. 3.3.3) that are closely related to a task's cardinality.

3.3 The Design Space as a Whole

The design dimensions do not stand for themselves, but are used in conjunction forming the design space. An *individual task* can be represented as a point in this design space by specifying a design choice in each of the five design dimensions. In the spirit of hierarchical task descriptions, these individual tasks form the building blocks of more high-level *compound tasks*. Individual tasks and compound tasks can then in turn be strung together to form *workflows*. These three levels of describing tasks with our design space are introduced in the following.

3.3.1 Individual Tasks as Points in the Design Space

An individual task is constructed from five design choices – one for each of the design dimensions that span the design space. It can be represented as a 5-tuple (goal, means, characteristics, target, cardinality) and be interpreted as a singular point in the design space. A simplistic example (see Sec. 5 for more realistic examples) would be to subsume research in climatology by the following individual task:

(exploratory, search, trend, attrib(temperature), all)

It means that the user is *searching* for a *trend* among *all* available *temperature attribute values*. This task is *exploratory*, as the user does not yet know if he is looking for an upward trend, a downward trend, or whether a trend exists at all. Only with these five aspects given, the task is fully specified. For instance, omitting the cardinality in the example would leave it open, whether to perform the task on all values or maybe only on those of the last week. So, this seemingly small information determines whether the task is a climatological or a meteorological one, each of these fields implying their very own set of terminology, conventions, and visual representations.

3.3.2 Compound Tasks as Subspaces in the Design Space

A compound task is constructed from a number of individual tasks. It can be represented as a non-empty set of tasks and be interpreted as a subspace in the design space. There are two ways of defining compound tasks: as an enumeration of individual tasks (bottom-up) or as a cut through the design space (top-down).

An example for an enumeration would be a compound task that does not merely look for a trend in the temperature, but also investigates outliers that might be used to refute a trend:

$$\{\textit{(exploratory, search, }\textbf{\textit{trend}}\textit{, attrib(temperature), all)},\ \textit{(exploratory, search, }\textbf{\textit{outliers}}\textit{, attrib(temperature), all)}\}$$

As a shorthand, we can also write the following:

$$\textit{(exploratory, search, }\textbf{\textit{trend}}|\textbf{\textit{outliers}}\textit{, attrib(temperature), all)}$$

Such an enumeration is useful when combining only a few different task options into a compound task. Yet, if the task definition shall be extended even further to comprise more or less any data feature that meets the eye, we can use a cut through the design space:

$$\textit{(exploratory, search, } * \textit{, attrib(temperature), all)}$$

This example would be a 4-dimensional cut (four design dimensions are specified) through the 5-dimensional design space. This is thought of as a top-down definition, which starts with the complete design space (*,*,*,*,*) and specifies one design decision after the other until everything that is known about a task has been described. The remaining unspecified dimensions then form the subspace that gives a compound task its flexibility. Note, that both types of definition can also be mixed to be as specific as needed on some design decisions and as flexible as possible on others:

$$\textit{(confirmatory, compare, } * \textit{, }\textbf{\textit{attrib(temperature)}}|\textbf{\textit{attrib(precipitation)}}\textit{, all)}$$

This compound task describes a possible confirmatory step after a pattern has been found in the temperature data by carrying out the above exploratory task. Since it is known that patterns in temperature data are reflected in precipitation data [34, 54], a pattern in one should reappear in some form in the other. Hence, this task aims to confirm the finding in the temperature data by comparing it to the precipitation data. This example also shows the limits of compound tasks, which can only capture individual steps of a visual analysis, but not their order or dependencies on one another, as they are described by workflows.

3.3.3 Workflows as Paths/DAGs in the Design Space

A workflow is constructed from a number of points (individual tasks) and/or subspaces (compound tasks). The dependencies between the tasks can be represented as a directed acyclic graph (DAG). This means if one task precedes another task, a directed edge connects them. If the workflow does not contain any branches or alternatives, it degenerates into a (possibly self-intersecting) path in the design space. Picking up the example from the previous section, the order of the two tasks can be expressed by the following two-step workflow:

$$\textit{(exploratory, search, } * \textit{, attrib(temperature), all)} \Rightarrow$$
$$\textit{(confirmatory, compare, } * \textit{, attrib(temperature)|attrib(precipitation), all)}$$

By this, we model that the user has to search for a feature in the temperature data first, before it can be confirmed by comparing it with the precipitation data. Yet, workflows go well beyond capturing domain-dependent step-by-step instructions for how to use particular aspects of data. With a bit of notation from the existing literature on process modeling and more flexibly defined compound tasks, workflows can even express general guidelines and fundamental visualization principles, such as the Information Seeking Mantra [47]:

$$\textit{(exploratory, summarize, } * \textit{, } * \textit{, all)} \Rightarrow$$
$$\textit{(exploratory, elaborate|filter, } * \textit{, } * \textit{, multiple)}^{+} \Rightarrow$$
$$\textit{(exploratory|confirmatory, gather, look-up, } * \textit{, single)}$$

The first step performs an *overview* (summarization of all data) on any particular data relation of interest and without a concrete data characteristic to look for, yet. The second step describes the *zoom+filter*, which cuts down on the cardinality of the shown data by using navigational means (elaborate) or means of reorganizing the data (filter). This task is performed iteratively (as denoted by the $^{+}$) until additional *details on demand* can be gathered on a single data object in the third step. This last step can be an exploratory one, if nothing is known about the data object's details, or it can be a confirmatory one, if the details were already encoded in position/size/color and thus the user already has a vague idea of their concrete values.

Note that in the same spirit as compound tasks combine individual tasks, and workflows combine compound tasks, further combinations can be derived in a similar manner. For example, it is easily imaginable to hierarchically combine workflows to express even more advanced concepts from task modeling, such as *ConcurTaskTrees* [41]. Yet, instead of building ever more complex constructs on top of our task definition, the following section proceeds to take a look at the capabilities and limitations of our fundamental design space that forms the basis of these derived concepts.

4 Discussion of our Design Space

The need for a model that clearly charts visualization tasks and puts them in perspective of each other is underlined by the many task taxonomies that have already been put forth in the past. In constructing such a model, a balance between its **completeness** and **consistency** has to be found: The more aspects are included in the model, the more potential inconsistencies in the form of invalid design combinations are introduced. Yet, the fewer aspects are included in the model, the fewer of the existing tasks can be described with it. With our five-dimensional design space, we have struck a particular balance between these two aspects, which we will discuss in the following.

4.1 Completeness of our Design Space

When surveying the existing task taxonomies, one will find three types of tasks:

- those that can be **captured by our design space** as it is, because these tasks are defined on the same high level of abstraction as our design space,
- those that can be **captured by adapting our design space** to their level of specificity, targeting it towards a particular application domain or software system, and
- those that **lie outside of our design space**, as they concern external aspects that we have not included in our design space – e.g., tasks that do not concern data, but other tasks.

In the following, we will go through each of these cases, in order to provide a better grasp of how far our design space reaches.

4.1.1 Tasks captured by our Design Space

Many existing tasks can be expressed in a straightforward manner, as it was exemplified in Sec. 3.3.3 with the "overview", "zoom+filter", and "details-on-demand" tasks from the TTT. In other cases, expressing existing tasks is not always as straightforward, as the following examples show.

A common challenge is that existing taxonomies often mix aspects that we have carefully separated into different design dimensions, most notably means and characteristics. An example are the *operation classes* of the TWL, which on the one hand contain tasks such as "cluster" *(∗,∗,clusters,∗,all|multiple)* and "identify" *(∗,∗,data value,∗,single)*. Even though these have been expressed as verbs, they do not specify a concrete means of how to go about finding these characteristics – they just say "do something (whatever it is) to obtain that clustering/value". On the other hand, it also lists tasks such as "compare" *(∗,compare,∗,∗,all|multiple)*, which is clearly a means in our terminology, as it does not specify which characteristics (e.g., clusters or values) to compare. Note that the cardinality in the tuple notation was not explicitly stated by the TWL, but since it was certainly implied, we took the liberty of filling it in.

Another challenge is posed by taxonomies, which have a somewhat different idea of what tasks are. For example, Ahn et al. [2] propose a task taxonomy tailored to dynamic network visualizations. Fundamentally, they describe a design space consisting of the dimensions *entities*, *properties*, and *temporal features*. In doing so, their approach is quite close to our faceted task definition along five independent design dimensions. In our terminology, entities and properties are subsumed under the target dimension as structural relation and attribute relation, respectively, and temporal features are data characteristics specialized to this application domain. What turns their design space into a taxonomy (cp. Sec. 3.1) is that they consider the means to be a dependent dimension, while we consider it to be independent: Their taxonomy gathers from publications and software packages which means are actually used in practice for which combinations of entities, properties, and temporal features [2, Fig.3]. It thus focuses on surveying the tasks that actually exist (taxonomy) and not so much the tasks that could possibly exist (design space). Nevertheless, our design space is able to express this notion of a task in the form of a subspace that combines targets and characteristics with appropriate means to pursue them:

(∗, find|identify|compare, peak|valley, attrib($*_{time}$)|attrib($*_{struct}$), ∗)

In this example, their taxonomy determines the means "find", "identify", and "compare" as being used in publications and/or as being supported by software systems for the combination of the characteristics "peak" and "valley" with the targets "activity" and "structural metrics", which map to our temporal and structural attribute relations.

4.1.2 Tasks captured by adapting our Design Space

Other tasks are not necessarily incompatible with our design space, but they are defined on a much more specialized lower level than our rather abstract, high-level task definition. This is mainly the case for two types of tasks: *system-dependent tasks* that are tailored towards concrete interaction features and *domain-dependent tasks* that are proposed to fit the needs of a particular application domain. As we have found it impossible to exhaustively include all system-dependent and domain-dependent design choices that were ever published, we chose to give high-level instances of possible design choices for each individual design dimension. At the same time, we emphasized that these design choices are not to be treated in a dogmatic way, but rather as starting points for one's own adaptation of our customizable and thus inclusive design space. Such an adaptation leaves the overall structure of the design space with its five dimensions untouched and only changes the design choices in any of the following ways:

- a simple **renaming** of design choices to be better understood by end users of a particular system or from a certain domain,
- a **refinement** of the abstract high-level design choices by subdividing them into a number of more concrete low-level choices to more precisely capture particular aspects of a software system or of a domain,
- an **extension** of the set of available design choices by incorporating additional unforeseen choices if these are necessary to describe tasks in a particular system or domain,
- a complete **substitution** of all design choices on a selected dimension to switch to an entirely different categorization that better fits the system or domain.

System-dependent tasks are usually used when referring to concrete interaction handles in a software, i.e., "zoom-out" instead of "summarize". This technical level of defining tasks is useful if one wants to automatically link a user's task with a concrete way of performing it in a software system. Knowledge about such a connection is beneficial in both ways: when performing a task, the user can be guided towards particular interactions that will help him to achieve it, but also when interacting with the visualization software, the system can attempt to determine which task the user is currently pursuing. The latter is helpful for keeping a record of the visualization session in a "history". System-dependent tasks are generally realized by simply concretizing the abstract means defined in our design space by refining and substituting them with the concrete interaction to be performed.

Domain-dependent tasks are very similar in this respect, only that they tailor the tasks towards a particular application domain. Note that both can go hand in hand if an application domain uses a rather fixed set of software systems. The most commonly used adaptation to a particular domain is the renaming of tasks and compound tasks to match the domain's terminology. On top of that, specific conventions of a domain can be modeled in our design space either as compound tasks (means A always involves target B) or as workflows (task X always precedes task Y). Often, a particular domain also implies a particular type of data that comes with it and that is not adequately described by our simplistic target dimension. This is a prime example for the case where a complete substitution of choices on a design dimension is the most convenient approach to cope with it. Data types for which this would be a reasonable choice are, for example, document collections or imaging data.

4.1.3 Tasks that lie outside of our Design Space

To keep our design space concise, we decided for a rather narrow understanding of tasks that treats some aspects as external influences and not as integral parts, as which they appear in other task taxonomies. Besides the aforementioned aspect of WHO performs a task in a collaborative environment [27], there are various other aspects that describe tasks in the much broader scope of visual analysis and visual data mining [40]. It is noteworthy, that these aspects designate in most cases entire research directions in their own right, which interact with our design space on various levels. This forbids their oversimplification as additional linear dimensions and thus, these aspects cannot be expressed within our design space. For example, an established model for contexts, such as a user's domain context, has itself already 12 different dimensions [32].

One of these aspects deserves to be mentioned individually, as it appears throughout the existing literature on tasks, yet is rarely recognized as being special: the self-referential aspect of tasks, which leads to the definition of *meta-tasks*. These are tasks to manage other tasks, such as "undo/redo", "revisit", or "task division". As they do not refer to data, but to tasks themselves, they stand outside of our design space. Interestingly, many task taxonomies, such as the TTT, list them in the same breath with regular tasks on data. Note that meta-tasks are of particular importance in collaborative scenarios, in which a large part of working with a visualization actually consists of dividing and managing tasks among the multiple users [27]. An especially useful meta-task is the "annotate" task that is not only of particular use in such collaborative scenarios [35], but also permits for renaming tasks to adapt them to particular domains and systems (cp. Sec. 4.1.2).

4.2 Consistency of our Design Space

In principal, our design space is consistent, as each design dimension answers to a different question of the "5 W's" and thus addresses a different aspect of a task. While this ensures that the design dimensions do not overlap, this cannot be claimed for all possible design choice combinations on these dimensions. For example, the means "compare" is connected with the cardinality, which must be either "multiple" or "all" for this design choice to make sense. If these inherent connections are not observed, inconsistencies can arise – for example, in the above case by choosing "compare" as a means together with the cardinality "single". Fortunately, our 5-tuples allow us to directly

encode these dependencies so that they can be observed. Usually, a means like "elaborate" that does not have any dependencies on other dimensions would be encoded as *(∗,elaborate,∗,∗,∗)*. Whereas "compare" would additionally mark down the constraint it poses on the cardinality dimension and be written as *(∗,compare,∗,∗,many|all)*. This interesting use of compound tasks works of course also for other design dimensions and gives us a general way to encode constraining semantics of design choices within the design space itself. By adhering to the encoded constraints, inconsistencies can be ruled out.

Note that many constraints that one may discover are actually not constraints at all. It is the nature of a design space to extend the meaning of design choices that were formerly only used in very confined settings also to other scenarios, simply by allowing for novel and so far unexplored combinations. Thus, what looks like an inconsistency in the design space may actually be just a very creative and unusual combination of design choices. An example would be the combination of "presentation" as a goal with the full spectrum of navigational means. This may look like an invalid combination to anybody who equates "presentation" with a printed picture or a linearly structured slideshow at most. Yet in the light of interactive storytelling and serious gaming, "presentation" takes on a much broader meaning, which actually fits well with an extended set of means to pursue it.

Another way in which constraints are falsely imposed is simply by wording. For example, the characteristic "trend" is usually associated with a temporal target. If one was to find a "trend" in a spatial target, one would rather call it "distribution". Yet, this connection between the characteristic and the target is only imposed by the used terminology. If one chooses a term like "tendency" to express the characteristic one is looking for, there would be no such strong implied connection with a particular target and the perceived constraint would be resolved. For such issues in wording, one could make use of visualization ontologies [19, 48, 58] to retrieve more suitable synonyms or generalizations of terms.

Even more than these general remarks, the next section will illustrate how to make use of our design space in the very concrete setting of climate impact research.

5 Use Case: Climate Impact Research

In its generality and abstractness, the proposed design space is applicable by a limited number of visualization experts only. To make it accessible and useful to users from an application domain, it has to be concretized by instantiating it for such a domain. In our case, this domain is climate impact research. As a first step towards understanding the tasks and the task terminology of users from this domain, we conducted a survey with eight end users from the field of climate impact research and with two experts developing visualization solutions in this field. This survey captures the most common tasks in the domain and frames them in terms of our design space (Sec. 5.1). Once established, they can be utilized for choosing visualization frameworks (Sec. 5.2) that are suitable to pursue a given set of tasks. In a similar way, they can also be used to suggest suitable visualizations for a task at hand to an end user (Sec. 5.3), if visualization developer and author have annotated the individual visualization techniques with the task subspace for which they are applicable.

5.1 A Domain-Specific Instance of our Design Space

From our experience, we observe a general gap between existing visualization techniques and systems, and the mental maps and practical demands of domain users. For many practical tasks, this gap results from hampered intuitive access to the required visualization functionality. Thus, to support complex visualization tasks and to establish sophisticated visualization techniques in application domains such as climate impact research, a translation of wording, visual metaphors, and interaction techniques is required, which considers the existing visualization and domain knowledge of the users. This knowledge has to be incorporated into the design of visualizations to smoothly integrate them into the scientists' ongoing, genuine research workflows.

Table 1. Collected Tasks from Domain Users based on Survey Answers

	task complexity	
terminology	**compound tasks**	**general workflow tasks**
visualization	explore the data, present data, find similarities/differences, visualize parameter distributions, visualize parameter variations, visualize outliers, gain overview over dataset, find correlations, create high-quality presentations, search for characteristic features, find relations/phenomenona/effects	present data for professional audiences, present data in a popular scientific way
intermediate	generate Fourier spectrum, visualize network bundles, visualize attributes aggregated by sector/region	verify theories using observation data, assess quality of model output in comparison to observation data, evaluate methods using test examples, validate experiment environments
application	find climatological means, find strong deviations of climatological means, visualize surface temperature, visualize wind patterns, find extreme wind field patterns, show water levels for area	perform planetary wave analysis and find resonance events, visualize the atmosphere, give information about water

In particular in the context of climate impact research, visualization design and visualization software must be adaptable to heterogeneous user groups, including users with different skills and qualification grades (from students to senior scientists), with different objectives (from scientific analyses to communication and policy making), and from different disciplines (e.g., meteorology, hydrology, socioeconomics, ecology, physics) that each come with different terminologies.

To concretize their often vaguely formulated domain-specific visualization tasks, we conducted a survey in which we first asked the domain users for a plain listing of tasks for which they are using visualization in their daily work. The outcome of this survey is summarized in Table 1, which classifies the gathered tasks into their level of complexity (i.e., compound vs. workflow tasks) and the kind of terminology used (close to the visualization context, close to the application domain, and intermediate being neither too visualization-specific nor too application-specific).

In a second part of the survey, we provided a list of seven predefined visualization tasks to our users, which are based on a previous study that we conducted in the field of climate data visualization [39]. They reflect the current perspective of visualization authors on which tasks are relevant in this context. To validate these tasks with the users and to define a task design subspace for this domain, we asked them to assess their importance and usefulness, as well as to list important tasks that were missing. The list of tasks presented to them was:

T1. compare variable distributions
T2. find model input/output relations
T3. gain overview of whole dataset
T4. present temporal trends
T5. visual (climate) model validation
T6. visual data cleansing / find data inconsistencies
T7. visualize periodicities

The result of this second study was that the importance and helpfulness of certain tasks were often directly linked to the scientific background of the interviewed user. While some tasks (T1, T3, T6) were considered useful by the majority of users, other tasks, such as T2 and T5, were only considered relevant by users with a modeling background. T4 was considered useful, but many users criticized that the task is too narrowly focused on temporal and presentation aspects. T7 was also considered useful, but being too constricted on visualization and it was asked for a more general variant of this task for periodicity analysis. Due to this feedback on T7, we renamed "visualize periodicities" to the more general "analyze periodicities", which seems to

Table 2. Resulting Design Sub-space for Climate Impact Research Tasks

task	notation
compare variable distributions	*(confirmatory, search|compare|navigate, distributions, attrib(attribute$_1$)|attrib(attribute$_2$), all)*
find model input/output relations	*(exploratory|confirmatory, search|relate|enrich, trends|correlations, attrib($*_{input}$)|attrib($*_{output}$), all)*
gain overview of whole dataset	*(exploratory, summarize, $*$, $*$, all)*
analyze trends	*(exploratory|confirmatory, $*$, trends, attrib($*$)|attrib($*_{time}$), all)*
visual (climate) model validation	*(confirmatory, search|enrich|query|relate, $*$, attrib($*$)|attrib($*_{space}$)|attrib($*_{time}$)|struct($*$), all)*
visual data cleansing / find data inconsistencies	*(exploratory, search|filter|extract, outliers|discrepancies, attrib($*$), all)*
analyze periodicities	*(exploratory|confirmatory, $*$, frequencies, attrib($*$)|attrib($*_{time}$), all)*
analyze outliers	*(exploratory, search|filter|query, outliers, attrib($*$), all)*
compare measurements with simulation data	*(exploratory|confirmatory, compare|enrich, $*$, attrib($*_{measurement}$)|attrib($*_{simulation}$) |attrib($*_{space}$)|attrib($*_{time}$), all)*
present data for general audiences	*(presentation, summarize, high-level data characteristics, attrib($*$), multiple)*

better capture the range of what the users want to do.

While almost all interviewed users gave suggestions for further tasks, most of them were too specific to be included – e.g., "present accumulated CO_2-emissions using integration". However, three missing tasks have been mentioned frequently and have therefore been added to the list: "analyze outliers", "compare measurements with simulation data", and "present data for general audiences". Based on the resulting extended list of tasks, we further concretized them by instantiating each task as a concrete 5-tuple in our general design space. The result of this process can be seen in Table 2.

The resulting list contains general purpose tasks (e.g., "gain overview of whole dataset" and "analyze trends"), as well as application-specific tasks (e.g., "visual (climate) model validation" and "compare measurements with simulation data"). Together they form a set of typical tasks for climate impact scientists from different disciplines. Note that some tasks that were named by the domain experts are very general and vague (e.g., "visual (climate) model validation" or "present data for general audiences"). This is due to the lack of standardized visual analysis workflows in these cases, which leaves it open to the researcher how to perform these tasks. Yet it is of course always possible to further concretize such tasks for smaller user groups that have a more homogeneous way of pursuing them.

From this list and the process of generating it, we learned a number of domain-specific requirements for the visualization. First, it appeared that comparison tasks ("relate", "compare", "trends") are usually very important to the users as they permit for relating certain time periods with a reference period, which is a very common analysis approach in climatology. Second, data aggregation ("summarize" or "enrich") is very typical within this field for deriving climatological information from meteorological data – e.g., by calculating seasonal and/or decadal data aggregations. Third, domain experts tend to think of analysis tasks from the presentation point-of-view, in the sense that they strongly prefer to pursue tasks with visualization techniques that they can directly share with their colleagues and use in publications. As a result of this observation, we removed the presentation aspects from the tasks in Table 2, as for climate impact researchers these are intrinsic up to a certain point. Instead, we added a general presentation task ("present data for general audiences") to capture the notion of communicating data to the general public, which is a separate notion from the intrinsic presentation. A last observation was again the heterogeneity of domain backgrounds that mix in the field of climate impact research. One result that stems from this observation is the differentiation of the tasks "analyze trends" and "analyze periodicities", even though they are in a design space sense structurally very similar (cp. Table 2). Yet, depending on the user groups, they are used in very different time scales: While trends are usually analyzed in decades or centuries close to the present, periodicities are commonly related to questions about much longer time scales – typically paleo data, which are climatological measurements from field data, i.e., drilling cores.

With the list of tasks provided in Table 2, we have concretized and instantiated our formal design space for the concrete scientific objectives in the field of climate impact research. As the most common uses of task taxonomies are to evaluate existing visualization techniques/systems and to recommend visualization techniques to users (cp. Sec. 2.2), we aim to exemplify the established task subspace along the same lines. Thus, the following section deals with evaluating visualization systems by asking which of the identified domain-specific tasks are supported by a particular software.

5.2 Choosing Suitable Visualization Frameworks

While the tasks in Table 2 describe what a domain expert may want to do, we can similarly describe what a specific visualization framework allows him to do. Table 3 illustrates two such feature-based task subspaces for the visualization frameworks GGobi [52] and ComVis [36], which are being used for climate impacts analyses.

Table 3. Supported Tasks in GGobi and ComVis

	supported subspace of visualization tasks
GGobi	*(exploratory|confirmatory, relate|enrich|query|browse|extract| summarize|filter, frequencies|outliers|discrepancies, attrib($*$), multiple|all)*
ComVis	*(exploratory|confirmatory, relate|enrich|query|extract| summarize|filter, frequencies|trends|outliers|discrepancies, attrib($*$), multiple|all)*

These feature subspaces are based on our knowledge of the most recent versions of the two frameworks. They are the result of us analyzing their support for all individual tasks, in which the domain-specific compound tasks of Table 2 break down. It is obvious from the resulting list of tasks, that both frameworks cater to a very similar spectrum of use, as they both have their strengths in analyzing multivariate data ("attrib($*$)"). They provide a high level of interactivity with brushing and linking ("query", "filter") in multiple views. In addition, both are strong in deriving data ("enrich") and extraction of filtered datasets for new analyses ("extract"), thus allowing for both "exploratory" and "confirmatory" analyses. By providing scatterplot matrices (GGobi) and parallel coordinates (GGobi, ComVis), overview tasks ("summarize") are supported as well. However, both have their weaknesses in (geo-)spatial data representation ("attrib($*_{space}$)"). Their differences lie in their ability for analyzing temporal relations ("attrib($*_{time}$)"), as GGobi provides very restricted support to identify "trends", while ComVis does not provide support for "browsing" the data in the same way as GGobi's Grand Tour mechanism does.

These two feature subspaces can now be matched with the task design space of the domain to aid in a decision for or against a particular framework. Both frameworks support the majority of tasks (for small and medium sized datasets) from Table 2. For "analyzing trends", ComVis fits somewhat better due to its flexible time series visualization techniques that directly support this task. Yet, tasks that are more specific to climatological questions ("visual (climate) model validation", "compare measurement with simulation data") are not well supported by either of the two frameworks for their lack of geospatial visualization functionality. Furthermore, since both focus on exploratory and confirmatory analyses, they are very restricted in "presenting data for general audiences".

The result of such an analysis may not only be a decision for or against a particular visualization framework, but also the identification of necessary plugins or extensions to broaden the scope of a framework in just the direction that is necessary for a particular application. In the case of pursuing climatological research with ComVis and GGobi, a geospatial visualization technique and a package to produce high-end presentation images have been identified as necessary extensions.

This example shows that our design space allows for a systematic assessment of required functionalities of visualization tools with respect to a domain-specific task subspace, i.e., the set of identified domain-specific tasks. It thus provides a conceptual basis for further visualization development and investment decisions in order to meet the feature requirements. Such a systematic evaluation of which tasks can be performed with which visualization tools is only one use of the concretized domain-specific task subspace. Its other common practical use is to recommend visualization techniques that are suitable for a given task. This is covered in the following section.

5.3 Choosing Suitable Visualization Techniques

Our design space, even in its domain-specific form, is not directly usable by *visualization end users*. To achieve this, it has to pass through two additional stages: an initial encoding in a machine-readable format by the *visualization developer* and a subsequent fine-tuning and concretization to match the requirements of a particular visualization session by the *visualization author*. Towards this end, we provide three different views on the abstract design space, each featuring a decreasing amount of flexibility to match the particular role.

For the visualization developer, we provide a basic XML notation to encode the 5-tuples of our tasks. This notation is a straightforward realization of our design dimensions and an example is given in Fig. 1. From the feedback provided by the two (geo-)visualization developers that we interviewed as well, we gather that this is an appropriate representation for developers, who are willing to work on such a lower technical level to maintain the full flexibility.

```
<TASK name = "find model input/output relations">
  <GOAL v0 = "exploratory analysis" v1 = "confirmatory analysis"/>
  <MEANS v0 = "relate" v1 = "search" v2 = "enrich"/>
  <CHARACTERISTICS v0 = "trends" v1 = "correlations"/>
  <TARGET v0 = "attributes(input)" v1 = "attributes(output)"/>
  <CARDINALITY v0 = "all"/>
</TASK>

<TASK name = "compare variable distributions">
  <GOAL v0 = "confirmatory analysis"/>
  <MEANS v0 = "search" v1 = "compare" v2 = "navigate"/>
  <CHARACTERISTICS v0 = "distributions"/>
  <TARGET v0 = "attributes(1)" v1 = "attributes(2)"/>
  <CARDINALITY v0 = "all"/>
</TASK>

<TASK name = "analyze trends">
  <GOAL v0 = "confirmatory analysis" v1 = "exploratory analysis"/>
  <MEANS v0 = "*"/>
  <CHARACTERISTICS v0 = "trends"/>
  <TARGET v0 = "attributes(*)" v1 = "attributes(time)"/>
  <CARDINALITY v0 = "all"/>
</TASK>
```

Fig. 1. XML notation of three example tasks taken from Table 2.

This stands in contrast to the visualization author, who expects to be handed a tool or customized editor to conduct his adaptations of the predefined bare design space coded by the developer. The author must translate the concrete specifications of planned or recurring analysis sessions/workflows to the design space. For this, we provide an interface that permits him to perform and test his adaptation, but only giving access to the design subspace that was created by the developer. Instances of this interface can be seen in Fig. 2, exemplified for the task "compare variable distributions". In a similar way, the author can describe the capabilities of concrete visualization techniques by listing the design options for which they are suited and for which not. Both descriptions, the task description and the technique description can then be matched using a simple rule-based mechanism, which we

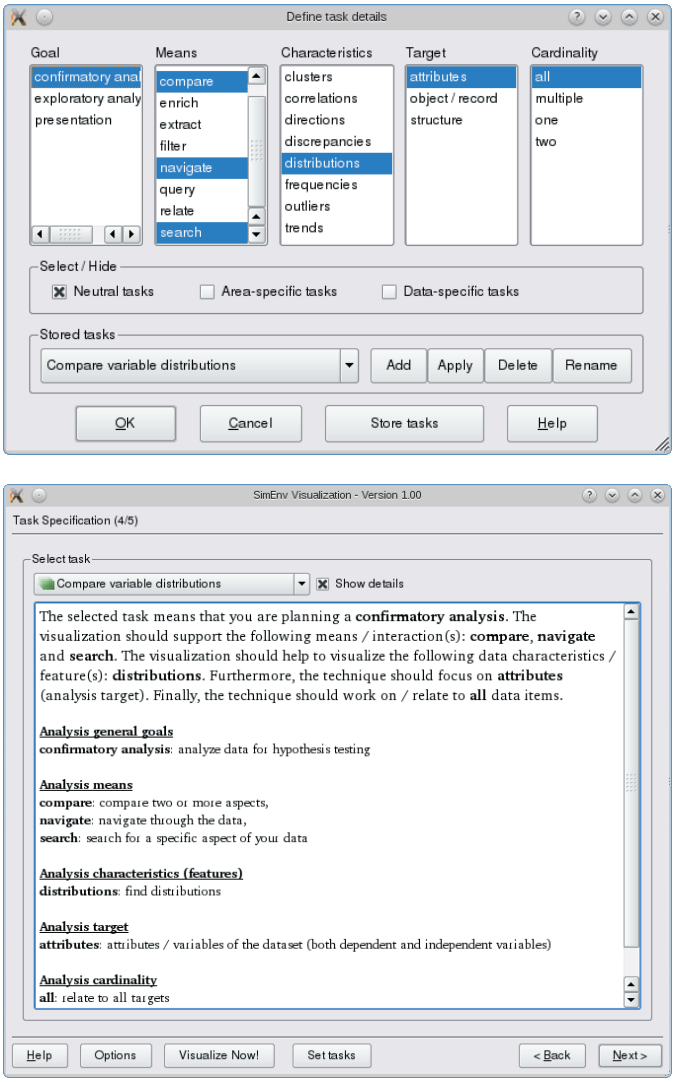

Fig. 2. An authoring tool permits for simple customizations of existing compound tasks. It gives access to the design dimensions and the design choices can be refined and altered to adapt a given compound task (top). The compound task can also be renamed to be meaningful to the domain users. Detailed descriptions are stored for each of the individual design choices, so that a description of a compound task can be auto-generated from them and shown to the user (bottom).

have introduced in an earlier publication [39] and which has been constantly used and improved in tight interplay with the users from the climatology domain for more than five years.

The end result of this authoring step is presented to the user to aid him in choosing a visualization technique based on his task. After loading the data and choosing his desired visualization task, which is now appropriately named and described so that the user can relate to it, a simple list of suitable visualization techniques is presented for him to choose from. This list is ordered by a suitability score that is derived by the rule-based matching mechanism. Fig. 3 shows an example for selecting appropriate 2D/2.5D visualization techniques for climate model output data. The available six techniques are provided to the user as a list that is ordered with respect to the user-specified compound task "analyze outliers". The "contribution" column at the right provides further feedback on how well the aspects of the chosen task match with the individual visualization techniques. This provides a small degree of transparency on how the visualization recommendation came about and thus improves the understanding of the visualization techniques and the defined compound tasks.

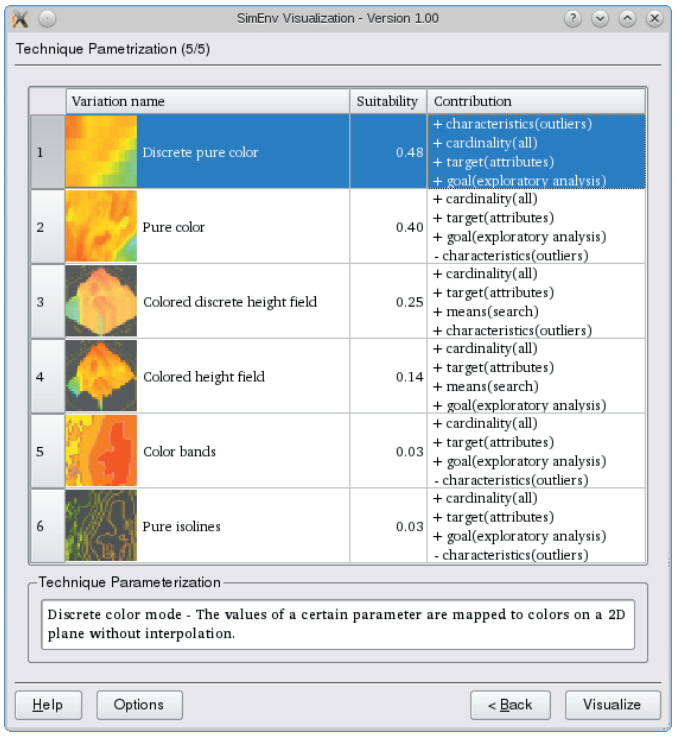

Fig. 3. Visualization recommendations given in the end user interface for the selection of 2D/2.5D visualization techniques based on the user-specified compound task "analyze outliers". It orders the available visualizations according to their computed suitability score that is given in the middle column. The right-most column shows the four most influential design requirements, ordered by the degree of their contribution to the final score. This column is usually hidden for end users, but it helps visualization authors and expert users to debug and further refine the set of compound tasks that form the domain-specific task subspace.

Even if this short section of how we apply the design space to the use case of climate impact research can only give a first glimpse of the many possibilities that arise from it, it certainly demonstrates the utility of our design space. While the recommendation of visualization techniques remains a complex problem, we take our results as a first indication that it is possible to derive recommendations if the task is concretely specified and narrowed down to the context of a particular application domain. Currently, we are still in the process of consolidating and evaluating the derived task design space for climate impact research, as over time feedback from more users is integrated and the list of tasks is further concretized and extended. Yet already at this stage, we can conclude that the promising results we have achieved so far would not have been possible without having the general design space of visualization tasks as a conceptual foundation in which to ground our approaches and software tools.

6 Conclusion and Future Work

With our design space of visualization tasks, we have contributed a first step towards concretizing the "colorful terminology swirl in this domain" [47]. At least from the discussions among ourselves and with the users from the application case, we can state that for us the concept of tasks and the many different notions surrounding it have become much clearer through this systematization. We have noticed that having the design space as a structure to orient on and the different design decisions as an agreed upon terminology enabled us to communicate about tasks more precisely and with less misunderstanding. In addition, it gave us a firm handle on the otherwise overwhelming amount of related work in this area. While they may need to be instantiated differently for other applications or technical realizations, we are confident that the five identified design dimensions will prove to be a useful fragmentation of the otherwise somewhat blurry and overloaded notion of tasks. In future work, we plan to put our optimism in this regard to a further test by applying our design space in the context of other domains. Only then the design space will form a solid and thoroughly tested basis for further extensions. One such extension that we are eager to explore is an interaction design space for visualization to be defined on top of it, which would provide a similar treatment for the currently existing interaction taxonomies.

Acknowledgments

The authors wish to thank the participants of the survey for their time and helpful input. Funding by the German Research Foundation (DFG) and the German Federal Ministry of Education and Research via the Potsdam Research Cluster for Georisk Analysis, Environmental Change and Sustainability (PROGRESS) is gratefully acknowledged.

References

[1] K. Abhirami and K. Vasan. Understanding user intentions in pervasive computing environment. In *Proc. of ICCEET'12*, pages 873–876. IEEE Computer Society, 2012.

[2] J.-W. Ahn, C. Plaisant, and B. Shneiderman. A task taxonomy for network evolution analysis. Technical Report 2012/13, University of Maryland, 2012.

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

[4] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In J. Stasko and M. O. Ward, editors, *Proc. of IEEE InfoVis'05*, pages 111–117. IEEE Computer Society, 2005.

[5] R. A. Amar and J. T. Stasko. A knowledge task-based framework for design and evaluation of information visualizations. In M. O. Ward and T. Munzner, editors, *Proc. of IEEE InfoVis'04*, pages 143–149. IEEE Computer Society, 2004.

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

[7] N. Andrienko and G. Andrienko. *Exploratory Analysis of Spatial and Temporal Data – A Systematic Approach*. Springer, 2006.

[8] T. Baudel and B. Broeksema. Capturing the design space of sequential space-filling layouts. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2593–2602, 2012.

[9] A. Becks and C. Seeling. A task-model for text corpus analysis in knowledge management. In *Proc. of UM-2001*, 2001.

[10] N. J. Belkin, C. Cool, A. Stein, and U. Thiel. Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. *Expert Systems with Applications*, 9(3):379–395, 1995.

[11] J. Bertin. *Graphics and Graphic Information Processing*. Walter de Gruyter, 1981.

[12] C. Beshers and S. Feiner. AutoVisual: Rule-based design of interactive multivariate visualizations. *IEEE Computer Graphics and Applications*, 13(4):41–49, 1993.

[13] A. F. Blackwell and Y. Engelhardt. A meta-taxonomy for diagram research. In M. Anderson, B. Meyer, and P. Olivier, editors, *Diagrammatic Representation and Reasoning*, pages 47–64. Springer, 2002.

[14] M. Bugajska. Framework for spatial visual design of abstract information. In E. Banissi, M. Sarfraz, J. C. Roberts, B. Loften, A. Ursyn, R. A. Burkhard, A. Lee, and G. Andrienko, editors, *Proc. of IV'05*, pages 713–723. IEEE Computer Society, 2005.

[15] S. K. Card and J. Mackinlay. The structure of the information visualization design space. In J. Dill and N. D. Gershon, editors, *Proc. of IEEE InfoVis'97*, pages 92–99. IEEE Computer Society, 1997.

[16] S. M. Casner. Task-analytic approach to the automated design of graphic presentations. *ACM Transactions on Graphics*, 10(2):111–151, 1991.

[17] E. H. Chi. A taxonomy of visualization techniques using the data state reference model. In J. D. Mackinlay, S. F. Roth, and D. A. Keim, editors, *Proc. of IEEE InfoVis'00*, pages 69–75. IEEE Computer Society, 2000.

[18] M. C. Chuah and S. F. Roth. On the semantics of interactive visualizations. In N. D. Gershon, S. Card, and S. G. Eick, editors, *Proc. of IEEE InfoVis'96*, pages 29–36. IEEE Computer Society, 1996.

[19] D. J. Duke, K. W. Brodlie, and D. A. Duce. Building an ontology of visualization. In *Poster Compendium of IEEE Vis'04*, pages 87–88, 2004.

[20] Y. Engelhardt. *The Language of Graphics: A framework for the analysis of syntax and meaning in maps, charts and diagrams*. ILLC Publications, 2002.

[21] I. Fujishiro, R. Furuhata, Y. Ichikawa, and Y. Takeshima. GADGET/IV: A taxonomic approach to semi-automatic design of information visualization applications using modular visualization environment. In J. D. Mackinlay, S. F. Roth, and D. A. Keim, editors, *Proc. of IEEE InfoVis'00*, pages 77–83. IEEE Computer Society, 2000.

[22] D. Gotz and M. X. Zhou. Characterizing users' visual analytic activity for insight provenance. *Information Visualization*, 8(1):42–55, 2009.

[23] G. Hart. The five W's: An old tool for the new task of task analysis. *Technical Communication*, 43(2):139–145, 1996.

[24] J. Heer and B. Shneiderman. Interactive dynamics for visual analysis. *Communications of the ACM*, 55(4):45–54, 2012.

[25] S. L. Hibino. Task analysis for information visualization. In D. P. Huijsmans and A. W. Smeulders, editors, *Proc. of VISUAL'99*, pages 139–146. Springer, 1999.

[26] E. Ignatius, H. Senay, and J. Favre. An intelligent system for task-specific visualization assistance. *Journal of Visual Languages and Computing*, 5(4):321–338, 1994.

[27] P. Isenberg, A. Tang, and S. Carpendale. An exploratory study of visual information analysis. In M. Burnett, M. F. Costabile, T. Catarci, B. de Ruyter, D. Tan, M. Czerwinski, and A. Lund, editors, *Proc. of CHI'08*, pages 1217–1226. ACM Press, 2008.

[28] W. Javed and N. Elmqvist. Exploring the design space of composite visualization. In H. Hauser, S. Kobourov, and H. Qu, editors, *Proc. of IEEE PacificVis'12*, pages 1–8. IEEE Computer Society, 2012.

[29] P. R. Keller and M. M. Keller. *Visual Cues: Practical Data Visualization*. IEEE Computer Society, 1993.

[30] R. Kosara, H. Hauser, and D. L. Gresh. An interaction view on information visualization. In C. Montani and X. Pueyo, editors, *Eurographics'03 State of the Art Reports*, pages 123–138. Eurographics Association, 2003.

[31] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry. Task taxonomy for graph visualization. In E. Bertini, C. Plaisant, and G. Santucci, editors, *Proc. of BELIV'06*, pages 1–5. ACM Press, 2006.

[32] D. B. Lenat. The dimensions of context-space. Technical report, Cycorp Inc., 1998.

[33] Y. K. Leung and M. D. Apperley. E^3: Towards the metrication of graphical presentation techniques for large data sets. In L. J. Bass, J. Gornostaev, and C. Unger, editors, *Proc. of EWHCI'93*, pages 125–140. Springer, 1993.

[34] R. A. Madden and J. Williams. The correlation between temperature and precipitation in the United States and Europe. *Monthly Weather Review*, 106(1):142–147, 1978.

[35] N. Mahyar, A. Sarvghad, and M. Tory. Note-taking in co-located collaborative visual analytics: Analysis of an observational study. *Information Visualization*, 11(3):190–204, 2012.

[36] K. Matković, W. Freiler, D. Gračanin, and H. Hauser. ComVis: a coordinated multiple views system for prototyping new visualization technology. In E. Banissi, L. Stuart, M. Jern, G. Andrienko, F. T. Marchese, N. Memon, R. Alhajj, T. G. Wyeld, R. A. Burkhard, G. Grinstein, D. Groth, A. Ursyn, C. Maple, A. Faiola, and B. Craft, editors, *Proc. of IV'08*, pages 215–220. IEEE Computer Society, 2008.

[37] E. L. Morse, M. Lewis, and K. A. Olsen. Evaluating visualizations: using a taxonomic guide. *International Journal of Human-Computer Studies*, 53(5):637–662, 2000.

[38] K. Nazemi, M. Breyer, and A. Kuijper. User-oriented graph visualization taxonomy: A data-oriented examination of visual features. In M. Kurosu, editor, *Proc. of HCD'11*, pages 576–585. Springer, 2011.

[39] T. Nocke, M. Flechsig, and U. Böhm. Visual exploration and evaluation of climate-related simulation data. In S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, editors, *Proc. of WSC'07*, pages 703–711. IEEE Computer Society, 2007.

[40] T. Nocke and H. Schumann. Goals of analysis for visualization and visual data mining tasks. In *Proc. of CODATA'04*, 2004.

[41] F. Paternò, C. Mancini, and S. Meniconi. ConcurTaskTrees: A diagrammatic notation for specifying task models. In S. Howard, J. H. Hammond, and G. Lindgaard, editors, *Proc. of INTERACT'97*, pages 362–369. Chapman&Hall, Ltd., 1997.

[42] D. Pfitzner, V. Hobbs, and D. Powers. A unified taxonomic framework for information visualization. In T. Pattison and B. Thomas, editors, *Proc. of APVIS'03*, pages 57–66. Australian Computer Society, 2003.

[43] P. Pirolli and S. Card. Sensemaking processes of intelligence analysts and possible leverage points as identified through cognitive task analysis. In *Proc. of IA'05*. The MITRE Corporation, 2005.

[44] P. K. Robertson. A methodology for scientific data visualisation: Choosing representations based on a natural scene paradigm. In A. Kaufman, editor, *Proc. of IEEE Vis'90*, pages 114–123. IEEE Computer Society, 1990.

[45] S. F. Roth and J. Mattis. Data characterization for intelligent graphics presentation. In J. C. Chew and J. Whiteside, editors, *Proc. of CHI'90*, pages 193–200. ACM Press, 1990.

[46] H.-J. Schulz, S. Hadlak, and H. Schumann. The design space of implicit hierarchy visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):393–411, 2011.

[47] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. of IEEE VL'96*, pages 336–343. IEEE Computer Society, 1996.

[48] G. Shu, N. J. Avis, and O. F. Rana. Bringing semantics to visualization services. *Advances in Engineering Software*, 39(6):514–520, 2008.

[49] R. R. Springmeyer, M. M. Blattner, and N. L. Max. A characterization of the scientific data analysis process. In A. Kaufman and G. M. Nielson, editors, *Proc. of IEEE Vis'92*, pages 235–242. IEEE Computer Society, 1992.

[50] C. Stary. TADEUS: Seamless development of task-based and user-oriented interfaces. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 30(5):509–525, 2000.

[51] M. Streit, H.-J. Schulz, A. Lex, D. Schmalstieg, and H. Schumann. Model-driven design for the visual analysis of heterogeneous data. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):998–1010, 2012.

[52] D. F. Swayne, D. T. Lang, A. Buja, and D. Cook. GGobi: Evolving from XGobi into an extensible framework for interactive data visualization. *Computational Statistics and Data Analysis*, 43(4):423–444, 2003.

[53] M. Tory and T. Möller. Rethinking visualization: A high-level taxonomy. In M. O. Ward and T. Munzner, editors, *Proc. of IEEE InfoVis'04*, pages 151–158. IEEE Computer Society, 2004.

[54] K. E. Trenberth and D. J. Shea. Relationships between precipitation and surface temperature. *Geophysical Research Letters*, 32(14), 2005.

[55] L. Tweedie. Interactive visualisation artifacts: How can abstractions inform design? In M. A. R. Kirby, A. J. Dix, and J. E. Finlay, editors, *People and Computers X: Proc. of HCI'95*, pages 247–265. Cambridge University Press, 1995.

[56] L. Tweedie. Characterizing interactive externalizations. In S. Pemberton, editor, *Proc. of CHI'97*, pages 375–382. ACM Press, 1997.

[57] 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 E. Bertini, C. Plaisant, and G. Santucci, editors, *Proc. of BELIV'06*. ACM Press, 2006.

[58] M. Voigt and J. Polowinski. Towards a unifying visualization ontology. Technical Report TUD-FI11-01, Dresden University of Technology, 2011.

[59] T. von Landesberger, S. Fiebig, S. Bremm, A. Kuijper, and D. W. Fellner. Interaction taxonomy for tracking of user actions in visual analytics applications. In W. Huang, editor, *Handbook of Human Centric Visualization*, pages 149–166. Springer, 2013.

[60] S. Wehrend and C. Lewis. A problem-oriented classification of visualization techniques. In A. Kaufman, editor, *Proc. of IEEE Vis'90*, pages 139–143. IEEE Computer Society, 1990.

[61] J. S. Yi, N. Elmqvist, and S. Lee. TimeMatrix: Analyzing temporal social networks using interactive matrix-based visualizations. *International Journal of Human-Computer Interaction*, 26(11–12):1031–1051, 2010.

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

[63] Z. Zhang, B. Wang, F. Ahmed, I. V. Ramakrishnan, R. Zhao, A. Viccellio, and K. Mueller. The five W's for information visualization with application to healthcare informatics. *IEEE Transactions on Visualization and Computer Graphics*. to appear.

[64] M. X. Zhou and S. K. Feiner. Data characterization for automatically visualizing heterogeneous information. In N. D. Gershon, S. Card, and S. G. Eick, editors, *Proc. of IEEE InfoVis'96*, pages 13–20. IEEE Computer Society, 1996.