Model-Driven Design for the Visual Analysis of Heterogeneous Data

Marc Streit, *Student Member*, *IEEE*, Hans-Jörg Schulz, Alexander Lex, *Student Member*, *IEEE*, Dieter Schmalstieg, *Member*, *IEEE*, and Heidrun Schumann

Abstract—As heterogeneous data from different sources are being increasingly linked, it becomes difficult for users to understand how the data are connected, to identify what means are suitable to analyze a given data set, or to find out how to proceed for a given analysis task. We target this challenge with a new model-driven design process that effectively codesigns aspects of data, view, analytics, and tasks. We achieve this by using the workflow of the analysis task as a trajectory through data, interactive views, and analytical processes. The benefits for the analysis session go well beyond the pure selection of appropriate data sets and range from providing orientation or even guidance along a preferred analysis path to a potential overall speedup, allowing data to be fetched ahead of time. We illustrate the design process for a biomedical use case that aims at determining a treatment plan for cancer patients from the visual analysis of a large, heterogeneous clinical data pool. As an example for how to apply the comprehensive design approach, we present Stack'n'flip, a sample implementation which tightly integrates visualizations of the actual data with a map of available data sets, views, and tasks, thus capturing and communicating the analytical workflow through the required data sets.

Index Terms—Visual analytics, analysis guidance, model-driven design, multiple data sets.

1 Introduction and Related Work

The advantages and challenges of multiple, heterogeneous data sets are widely recognized in the field of Visual Analytics. Thomas and Cook recommend the creation of "methods to synthesize information of different types and from different sources into a unified data representation" [1, p. 11]. Such a unified representation can be envisioned as a heterogeneous *information landscape*, in which information foraging and sense-making take place. Like a person exploring an unknown territory in the real world, a user navigating the high-dimensional, multifaceted and overwhelmingly large, combined data space of multiple, heterogeneous data sets must be provided with some means of orientation. This challenge has been described in the context of information retrieval as early as 1993 [2]. This publication also coined the notion of *information landscapes* and identified different strategies commonly used to gather information within them—e.g., exploring the data in an undirected fashion or following a concrete plan for finding the desired information. In this paper, we take the next step towards assisting these analytical strategies not only by means of orientation within the data, but also within the large and diverse set of available analysis, methods, and visualization techniques.

In this context, data analysis can be aided by providing two different levels of *analytical support*:

- *Orientation* that communicates the current position within the information landscape, the path of analysis steps that led there (history), and possible directions for further investigation (e.g., related data sets).
- *Guidance* that suggests concrete analysis steps to be taken in order to get from an analysis hypothesis to an analysis result.

The contribution of this paper is a model-driven approach to achieve these two levels of analytical support in two steps: First, a model in the sense of the aforementioned unified representation is constructed via an authoring process. This model goes beyond the sole definition of the information landscape, as it also contains details on suitable visualization and computation methods to access the data sets. In a second step, this model is utilized to provide orientation by means of making the model explicit to the user, and to provide step-by-step guidance by inferring possible paths within the modeled analysis setup, which will lead the user to a desired analysis result.

Several ways of providing such orientation and guidance were described in the field of visualization. On a conceptual level, the literature offers two prevalent strategies. The most common strategy is the data-driven, bottom-up strategy, which gathers data and distills it into navigational cues. For example, the *VisSheet* system generates a number of previews for a range of possible visualization parameter changes and presents them to the user to choose from [3]. Another example is the approach of *HARVEST*'s behavior-driven visualization recommendation which analyzes the

- *M. Streit, A. Lex, and D. Schmalstieg are with the Institute for Computer Graphics and Vision, Graz University of Technology, Graz 8010, Austria. E-mail: {streit, lex, schmalstieg}@icg.tugraz.at.*
- *H.-J. Schulz and H. Schumann are with the Department of Computer Graphics, Institute of Computer Science, Faculty of Computer Science and Electrical Engineering, University of Rostock, Rostock 18051, Germany. E-mail: {hjschulz, schumann}@informatik.uni-rostock.de.*

Manuscript received 27 July 2010; revised 22 Dec. 2010; accepted 12 Apr. 2011; published online 16 June 2011.
Recommended for acceptance by K.-L. Ma.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2010-07-0156.
Digital Object Identifier no. 10.1109/TVCG.2011.108.

1077-2626/12/$31.00 © 2012 IEEE Published by the IEEE Computer Society

user's analytic activity [4]. When employing the concept of *Social Navigation* user data, but crowdsources it from multiple users and displays it as usage statistics indicating popular or neglected user choices [5]. Similar data-driven techniques can be employed on a higher level as well, as it is done by the *VisComplete* system which mines a database of existing visualization pipelines to aid the user in constructing new ones by suggesting possible completions [6].

An approach used more rarely than the data-driven strategy is the workflow-driven, top-down strategy, which derives navigational assistance by instantiating a predefined, abstract best-practice solution with concrete visual and computational techniques. One example is the *Systematic-Yet-Flexible* system, which gives a step-by-step guidance along a high-level workflow, while leaving the choice of concrete techniques that achieve the higher level objectives to the user [7].

The approach presented in this paper falls into this second category, but has a much larger scope: besides the actual workflow, it also utilizes the aforementioned model of the analysis setup, which includes the available data sets and their interrelations, the available algorithmic and visual methods and packages, as well as their applicability to achieve individual steps of the workflow. As a result of making this additional information available, it is possible to automatically determine suitable analytical techniques and subsequently use this information to provide navigational cues on a much more specific, lower level along a given analysis path. This proves especially useful in interactive systems that exhibit a large number of possible continuations at any given point—so that the decision which functionality to use on which part of the data and in which order is particularly challenging.

Our concept of a model-driven design for visual analysis support draws upon first ideas from our earlier position paper [8].

2 Conceptual Foundations

Large information landscapes with multiple, heterogeneous data sets and numerous visual and computational interfaces to access them require means of support to ensure their timely and accurate analysis. Providing such user support is not a trivial task, as the degree of support required by the user may vary during the analysis session—the user may need concrete guidance during one part of the analysis session and only means for orientation during other parts. To realize such a smooth back and forth between these two levels of support, a visual analysis system must have considerable knowledge about the available data sets, the goals of its user, as well as its analytical capabilities. Our approach encapsulates these aspects in three models:

- a domain-independent **model of the setup** in which the interactive visual analysis takes place—describing the data sets, the visual and computational interfaces to the data, and the analytical operations that can be performed with them,
- a **model of the domain** that captures what can be done with a given setup in the context of a specific domain—describing the numerous domain-specific tasks and relating them to the data sets and analytical operations of a given setup model,
- and a **model of the analysis session** that lists what has to be done to pursue a given analysis goal—describing the analysis workflow as a sequence of domain-specific tasks from a given domain model.

The knowledge specified by these models requires an authoring phase in which the models are put together. It is obvious that the overhead of such an elaborate modeling phase is not justified for straightforward setups with a manageable complexity. However, with increasingly complex models, the benefits soon outweigh the initial modeling costs. This is especially true for highly repetitive analysis sequences, which have to be modeled only once and can then be reused over and over again. For such routine tasks, the guidance ensures that every repetition is done with the same care as the very first analysis and without forgetting a crucial intermediate step. A guided analysis thus provides a high degree of reproducibility and traceability, which makes most sense for application fields in which a faulty analysis may lead to dire situations, such as the diagnosis of patients or the analysis of safety hazards in airplane inspection. Nevertheless, if the user wants to deviate from the workflow of a guided analysis to freely roam the information landscape in a more explorative, unplanned fashion, he can do so at any point, resulting in a fall back from guidance to orientation support. Transitioning back from such an exploratory side step onto the planned analysis path means that guidance can then continue with step-by-step instructions again.

The setup model is authored once and needs to be adapted or extended only when new data sets or tools become available. With this underlying, domain-independent model, different domain models can be associated, as different application domains may use the same setup to carry out the analysis. This can be frequently observed, e.g., in the field of life sciences, where a geneticist and a biochemist may use the same data sources and interfaces, but perform completely different tasks. In the last step, a concrete analysis workflow is formulated, which is then tailored to the availability of data and analysis methods for a given case, by pruning tasks that cannot be performed. This yields a streamlined analysis workflow, which contains only those analysis paths that can be realized with the given data and tools.

The next subsection outlines the overall authoring process together with the different roles involved in each individual authoring step.

2.1 Overall Authoring Process and Involved Roles

The description of complex, possibly cross-domain analyses require a good deal of expertise in all fields involved. As the assumption of an omniscient expert is unrealistic, we elicit different roles for the authoring of the different models. Table 1 lists the three roles involved in the authoring, as well as two possible roles for analysts using the models.

The process of authoring the different models is best described as a step-wise procedure, which sequentially adds to the complexity of the models until they are fully specified. This is shown schematically in Fig. 1. The authoring process consists of the following sequence of steps, each being the responsibility of one of the three expert roles from Table 1:

I. Developing the data model: this is the responsibility of the **data manager**, who describes the data sets and their interrelations.

TABLE 1
Roles in Authoring and Using Models of Setup, Domain, and Analysis Session

Role	Description	Category
Data Manager	responsible for building and maintaining the data model	author
Visual Analysis Expert	responsible for compiling interfaces and their operators	
Domain Expert	responsible for compiling tasks and analysis workflows	
Informed Analyst	works on open research questions with no predefined analysis workflow	user
Guided Analyst	works on answering a routine question along a predefined analysis workflow	

II. Enriching the data model with interfaces: this is done by the **visual analysis expert**, who annotates the data sets with information about how to access each of them—via graphical interfaces (visualizations) or through computational interfaces (query languages, statistics packages).

III. Compiling a list of operators for each interface: this lies within the responsibility of the **visual analysis expert**, who denotes which interface is suitable to perform which operations, as some interfaces may be more fitting than others.

IV. Connecting tasks to the data model: for this, the **domain expert** identifies the required data sets for each of the high-level analysis tasks that are commonly performed in a given scenario and relates them to the task.

V. Associating operators with the tasks: this is specified by the **domain expert** who links concrete operators to carry out the given tasks on the associated data. As the operators are domain independent, the translation from domain-specific tasks to operators should be supported by the **visual analysis expert** who contributes knowledge about suitable analysis methods.

VI. Specifying a workflow of analysis tasks: in this step, the **domain expert** details concrete analysis sessions for pursuing a given goal by defining an analysis workflow using the tasks defined.

VII. Pruning the workflow according to the actually available data sets and tools: as a final step, it is **automatically** determined, which paths within the workflow cannot be performed for a concrete instance of data and analysis tools. These are then pruned from the workflow.

The first three steps of this process describe the rather static setup of the analysis: data sources, ways to access these data sources, and analytical operators to run on them. Steps IV and V concern the domain model, as they add the domain-specific tasks on top of the setup model. The last two steps connect these tasks to meaningful analysis sessions and prune these sessions to use only the data and tools available at analysis time.

With all these models available, we have identified two different roles of users that can benefit from the explicitly modeled setup and analysis session. The first is the **informed analyst** who analyzes the data freely, without following a predefined analysis path. For the informed analyst, the key benefit is the provision of orientation, which allows him throughout the entire exploration process, to pinpoint exactly which part of the information landscape is currently under investigation, which methods are available to analyze this particular information, and which other parts of the information landscape may be related and thus be of interest. For informing an analyst, only the model of the setup with all its data sets and different visual and computational operators is needed.

The second role is the one of a **guided analyst**, who follows a given analysis path and possibly conducts similar analyses routinely. The guided analyst benefits from the formal model of the analysis session, as it provides exactly the step-by-step guidance on how to pursue an analysis path to achieve a given analysis goal with the data at hand. If necessary, the guided analyst may also deviate from the proposed workflow, in which case the user's role switches to an informed analyst.

It should be noted that a one-to-one mapping of a specific person to a role is not required. Depending on the use case and its complexity, the responsibilities of one role can be performed by multiple individuals. Also, one person can fulfill multiple roles—for instance, the domain expert may fulfill one or both user roles. It is also possible to further extend or subdivide the suggested roles, for example, with more concrete user profiles for specific applications.

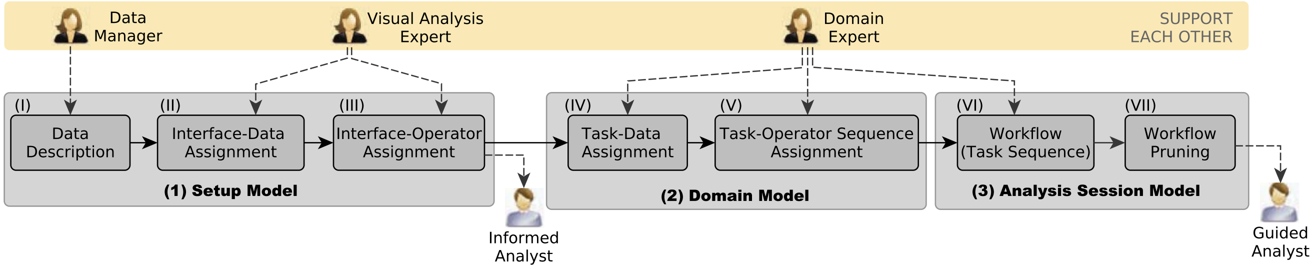

Fig. 1. The authoring process shown as a sequence of authoring steps (I-VII) carried out by data, visual analysis, and domain experts. The authored models can be taken as aids for providing analytical support on two levels: the setup model (1) for informing (providing orientation to) a user within the sets of data, computational procedures, and visualizations, the intermediary domain model (2), and the model of the analysis session (3) relying on the setup and domain model for guiding (providing step-by-step directions) the user.

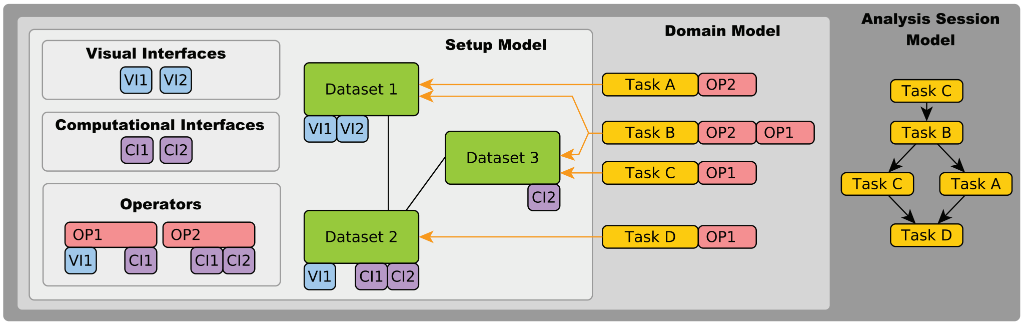

Fig. 2. The different parts of the setup, the domain and the analysis session: interfaces (blue and purple), operators (red), data sets (green), and tasks (yellow). These parts are described and interrelated during the authoring process. An example of a fully authored model of analysis setup, analysis domain, and analysis session is shown in Fig. 3.

Having sketched the overall authoring process, it remains to detail the individual authoring steps and how they build upon one another.

The models are not targeted toward orientation and guidance per se, but can potentially be used to optimize all kinds of processes, such as the treatment of missing data or collaborative analysis, as we have envisioned in [8]. We embrace this generality as a strong argument for such a comprehensive authoring approach. To reflect the clear distinction between the general models and their specific application, we keep the following explanation of the authoring steps general as well. Nevertheless, we use the domain of biomedicine to give examples.

2.2 Authoring the Setup Model

The setup model captures the basic infrastructure in which the analysis takes place. Besides all the different data sources being available (Step I), this includes the software infrastructure for accessing the data (Step II), as well as the available software tools, such as visualization frameworks or statistics libraries, for analyzing the data (Step III).

2.2.1 Step I: Developing the Data Model

The data model captures all data sets (shown green in Fig. 2) available in an analysis setup. This can include local data sets (i.e., an electronic patient file), data sets available from online databases (i.e., pharmaceutical lists or digital anatomical atlases), streamed data (i.e., a patient's vital signs coming from intensive care), etc. Additionally, the different data sets contain different types of data, such as imaging data from body scans, gene expression data from microarray analyses, text data from electronic documents, etc. The data sets are then related via common keys or identifiers where this is possible. In our biomedical use case, this can be, for example, the patient's name or social security number, thus identifying a patient's records across different data sets. In the case of different conventions being used for identifiers among multiple data sets, an ontology can often be used to map them.

A data model of this sort is commonly used to plan and implement the combination of large database collections [9]. Large organizations, such as hospitals, usually have employees dedicated to define and refine such models, to validate and cross-reference entered data, and to supply necessary metadata. Hence, many larger setups and even many freely available data collections, such as *linkeddata.org* or *data.gov*, do already have a data model of some sort. Yet beyond the pure organization of data sets, such data models are rarely used. A first approach utilizing a data model for visual analysis was only recently given by Lieberman et al. [10]. They use well-established, standard data models (e.g., ERM [11]), which our approach also relies on. This makes it easy to reuse or adapt existing data models for our setup model.

2.2.2 Step II: Enriching the Data Model with Interfaces

A first step to enhance the data model beyond what is stored is to add information about how to access each data set. The access is conceptually performed through interfaces, which can be

- computational interfaces (purple in Fig. 2), that fetch the data either directly from the source (low-level, query interfaces—e.g., *SQL* or *MapReduce* [12]) or calculate derived data, such as clusterings or correlations (high-level, algorithmic interfaces—e.g., *R statistics toolkit*[1] or *WEKA*[2])
- visual interfaces (shown blue in Fig. 2), e.g., scatter plots or parallel coordinates, allowing for access using interactive, graphical methods, such as visual queries or query by example.

These interfaces are provided by the software infrastructure of the analysis setup—database front ends, statistical libraries, visualization frameworks, etc. As different types of data require or permit different interfaces, the information about which method of access is available for each data set is added to the data model. This is done through one-to-many assignments, as a data set may require a combination of multiple visual interfaces to be properly displayed, or as an algorithmic interface may need several data sources to derive the desired information.

2.2.3 Step III: Compiling a List of Operators for the Interfaces

Operators (shown red in Fig. 2) are domain-independent analysis actions that describe in general terms what each available interface can be used for. For example, an *SQL* interface is perfect for querying individual data items, a statistics library is well suited for correlation analyses and clustering, and a parallel coordinates view is ideal for interactive filtering.

The list of operators for each interface is usually based on the experience of the visual analysis expert, as well as on

[1] www.r-project.org.
[2] www.cs.waikato.ac.nz/ml/weka.

domain-specific conventions and recommendations from the literature. Hence, this step encodes common knowledge and the current state of research in the field of Visual Analytics in general.

This completes the modeling of the setup. It effectively describes the information landscape, and computational as well as visual access methods, as they are needed for guiding the informed analyst.

2.3 Authoring the Domain Model

The domain model adds a layer of domain-dependent knowledge on top of the setup model. It does so by associating tasks being formulated in terms of the domain with the appropriate data (Step IV) and operators (Step V).

2.3.1 Step IV: Connecting Tasks to the Data Model

As Munzner points out, the term "task" is overloaded in the visualization literature [12]. Hence, it should be made clear that the term "task" is being used here for domain-dependent, textual descriptions of what an analysis step should achieve on which data set. An example of a domain-specific task is "Find all patients with a common characteristic." At this stage, tasks (yellow in Fig. 2) are described and linked to the data sets they are performed on. In the example given, patient characteristics may be scattered across multiple data sets. As no concrete characteristic is specified, the task would be connected to all of these data sets.

Tasks are closest to the actual analytical process and describe, in the words of the domain expert, what is being analyzed with which goal. They are used later as the building blocks of analysis sessions.

2.3.2 Step V: Associating Operators with the Tasks

While Step IV models what to do with which data set, Step V finally defines how to do it, in order to actually be able to carry out a task. This is achieved by mapping the tasks to the domain-independent operators. The mapping can either assign a single operator or a few operators to be carried out subsequently. Otherwise, in the case of tasks getting too complex, they can always be broken down into multiple more fundamental tasks. In the case of the example task "Find all patients with a common characteristic," this would be a single *filter* operator that filters the data set of patients by the given characteristic. If a data set provides multiple interfaces to perform the filter operator with, e.g., an *SQL* interface and a parallel coordinates visualization, then the task is connected to all operators provided by the different interfaces. Which one to choose is for the user to decide.

This completes the modeling of the domain. It bridges the domain-dependent analysis steps and the domain-independent analysis setup, and effectively yields a graph that connects data sets and tasks via domain-independent operators. The last authoring steps define the missing workflows on top of the domain model.

2.4 Authoring the Analysis Session Model

Often, an analysis session is seen as being equivalent to performing a sequence of analytical tasks. Yet in our concept, analysis sessions are more abstractly defined, also capturing different analytical possibilities, in order to ensure their reusability for other instances of data (e.g., other patients). Specifically, an analysis session model consists of two parts: the actual analysis workflow (Step VI) and the constraints imposed on the workflow due to unavailability of data sources or analysis tools (Step VII).

2.4.1 Step VI: Specifying Workflows of Tasks

This authoring step assembles analysis workflows using the available tasks as building blocks in whichever order they are needed. In addition to simply appending tasks in a purely sequential order, Step VI also makes it possible to model more complex analysis patterns than a linear, step-by-step composition of tasks. In order to capture the involved and convoluted nature of analysis, branching, looping and forward jumping is possible as well—in the very same spirit, as task models [13] or user-task models [14] are authored in the field of interface design.

The analysis workflows are modeled as directed graphs with tasks as nodes and edges as transitions from one task to the next. Alternative analysis paths leading to the same analysis goal are rather common, so the branching of a workflow is an important property that makes it possible to capture and combine multiple possible analysis paths in one analysis session model. Likewise, the incorporation of forward jumps as shortcuts allows the same session model to be used for novice and professional users, alike. The guided analysis can switch between a detailed step-by-step walkthrough for the former and a less elaborate, shorter "todo-list" for the latter—even in the middle of the analysis. On top of that, loops make it possible to encode any number of task repetitions by revisiting a task (sequence) until its result is refined enough to be taken as an input for the next task. Moreover, it is possible to define preconditions per task to specify certain requirements to be met, e.g., a hierarchical clustering or aggregation to be performed before visually analyzing the results of the processed data. Likewise, postconditions can be formulated that impose requirements on the result of an analysis task, e.g., with regard to accuracy.

While the definition of the analysis workflows is usually done by the domain expert, it is also possible, to leave this to an informed user, who can also define paths for the guided, routine users.

2.4.2 Step VII: Pruning the Workflows According to the Available Data Sets and Tools

As a final step, the analysis session model is adapted to the constraints imposed by the unavailability of data (e.g., as not all theoretically collectible data may have been gathered for a given patient or the analyst may not have the clearance to view them) and of the analysis tools (e.g., licensing issues may prevent their use or an analyst may not be properly trained to use them). This adaptation is done by automatically pruning all tasks that rely on unavailable data or interfaces from the workflows. As a result, the remaining workflows cover all currently possible analysis paths which can be chosen as the analysis progresses.

This completes the overall authoring process. It may seem quite elaborate at first, but the modularity of the three models ensures a high level of reusability. The same setup model can be used to build different domain models on top of it, and the same domain model can in turn be used to author numerous workflows utilizing it. This makes a lot of sense, as the definition of workflows is usually more short-lived and prone to be changed and optimized more often than the basic setup model or the domain model. The following section briefly explores the final use of the models for providing analytical support, which motivated the externalization of the experts' knowledge about infrastructure, domain, and workflows in the first place.

2.5 Utilizing the Models for Analytical Support

The use of the setup model for orientation support is rather straightforward, as the model itself already provides a map in which to pinpoint the current analysis step and determine possible next steps. Using the analysis session model for the guidance support requires some extra computation.

At first, it needs to be determined, if any continuous analysis path is left after pruning. This allows to check, whether or not an analysis goal can be pursued at all by the specific analyst on the given data within the current setup. If not, one could, for example, request the collection of additional data in order to obtain enough information to be able to complete an analysis path. In our use case, this can be additional tests or screenings for a patient. The session model makes it possible to determine the smallest gap among the analysis paths, which can then be bridged at minimal costs—financially or in terms of the stress a patient has to go through. It thus realizes the opposite direction of the pruning: the pruning ensures that nothing is (intended to be) used that is actually unavailable by removing these parts from the model, whereas the reachability check makes sure that everything is available that is needed at the bare minimum to pursue the intended analysis goal.

Second, it must be determined which analysis path to actually use for guiding the analyst among all the possible analysis paths contained in the analysis session model. For this, it is important to observe that the paths differ in terms of their *seamlessness* and *effectiveness*. A path is considered to be *effective* when it is short compared to other possible analysis paths. A path is called *seamless* if for each transition from one task to the next, there exists a relation (edge) between the data sets that the tasks are connected with as well. A seamless analysis path would allow the analyst to proceed from one task to the next without destroying the mental map, as the data sets used by both tasks are related via a common identifier. The more discontinuities between data sets an analysis path has to bridge, the less seamless it is. For a traceable and swift analysis, paths that are more seamless and effective are generally preferred and thus chosen for the guidance.

To bring this whole process to life, the following section gives an example for authoring the three models and using both forms of analyst support.

3 Applying the Design Process to a Biomedical Use Case

Based on the theoretical foundation laid in the previous section, we demonstrate how to apply the concept to a real use case. The use case covers a comprehensive analysis of patient-related data. Our long-term collaboration partners from the Institute of Pathology at the Medical University of Graz approached us with a need for visual analysis: they try to base a decision of how to treat a newly diagnosed cancer patient on a wider array of available data. In such a scenario, they would like to analyze the patient's basic data, anamnesis, tissue data, gene expression data, etc., and relate it to other reference patients. Moreover, they want to be able to explore information about genes, proteins or pathways, which they encounter during an analysis. Hence, it is a prime example of visual analysis across multiple, heterogeneous data sets.

3.1 Creating the Setup Model

The starting point for creating the setup model (*cf.*, Section 2.2) is a well-defined data model, which in an optimal case can be based on an existing hospital data management system. In this scenario, many of the data sets are directly linked to the patient. This is reflected in Fig. 3 by the high degree of connectivity from the patients' basic information to other data sets. The patient-related data sets include:

MR/CT/X-ray. Magnetic resonance (MR), computer tomography (CT), and X-ray data are acquired using imaging techniques. For cancer patients, a tumor might be visible in one, several or all of the imaging data sets. In some cases, computer-based analysis, such as automatic tumor segmentation, is employed.

Tissue samples. When a tumor is discovered, the standard procedure is to take a biopsy. The acquired tissue is investigated under the microscope. High-resolution scans are acquired and stored in a database.

Gene-/protein expression. High-throughput techniques like DNA microarrays enable the biomedical expert to measure the regulation of ~omics data (genomics, proteomics, metabolomics, etc.—for details see [15]) for a patient at a specific point in time. This snapshot of the expression tells a life scientist how active a gene/protein is, which influences the cellular processes and in turn the disease itself. A common procedure to analyze expression data is to cluster a group of patients with known features and try to find similarities and/or differences between their profiles which allow to draw conclusions in the analysis.

Anamnesis. The anamnesis is a patient's medical history—including illnesses, allergies, etc.

Lab results. Lab results include blood levels, urine, and stool sample results, etc.

The patient-independent data sets are:

Pathways. Pathways are models of the biochemical processes in cells. They are especially valuable in combination with ~omics expression data [15].

Disease database. Diseases and health-related conditions are classified according to various disease schemes (e.g., ICD).

Gene/protein database. Information about genes and proteins is stored in public databases as, for example, *GeneCards*[3] and *EntrezGene*.[4] These web sources include metainformation such as short names, alternative identifiers, a detailed description, references to publications, and disease classifications.

Publications. Articles published about genes, proteins, pathways, and diseases play an important role during the

[3] www.genecards.org.

[4] www.ncbi.nlm.nih.gov/Entrez.

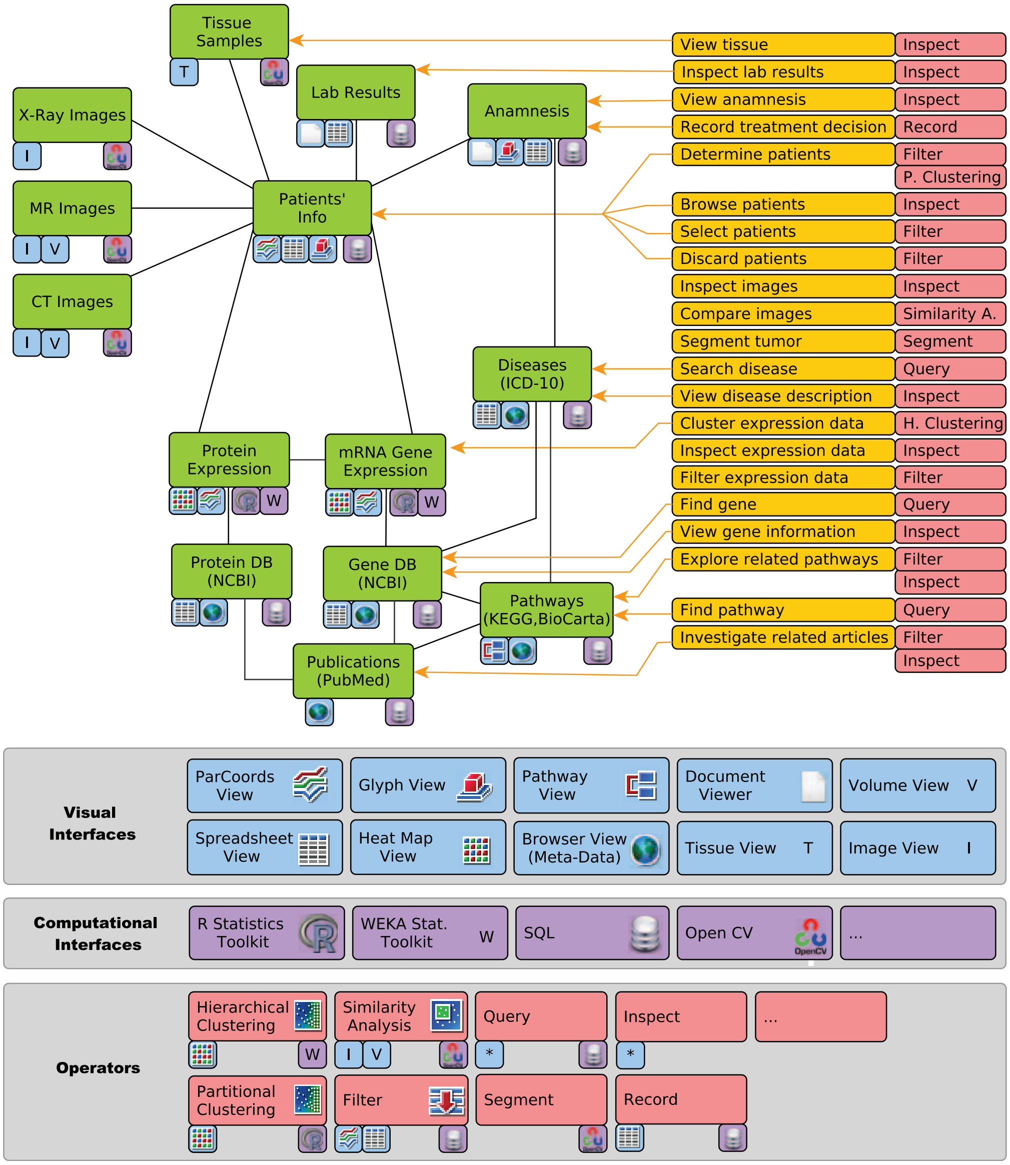

Fig. 3. Setup and domain model of the biomedical use case. The data sets (green)—either from local or online sources—are connected when they share a common identifier. The interfaces (blue for visual interfaces, purple for computational interfaces) are compiled from several tools and assigned to the data sets. For the analysis session description, tasks (yellow) and operators (red) are added and connected to the data sets.

analysis, as an analyst can gain deeper knowledge on the topics if needed. The most commonly used database for literature research in the biomedical domain is *PubMed*.[5]

To create the data model, the clinical data manager collects those data sets (green in Fig. 3) and defines their relations (*cf.*, authoring Step I). Having the data model at hand, the design responsibility is handed over to the visual analysis expert who chooses or develops suitable visual as well as computational interfaces and assigns those interfaces to the data sets (Step II). This step requires in-depth knowledge about the tools available for conducting the analysis. In our scenario, we use the Caleydo visualization framework [16], [17] for biomolecular-, tissue-, patient-, and metadata. We use a commercial volume visualization tool for MR/CT and X-ray data. The visual analysis expert starts by compiling a list of the available (visual as well as computational) interfaces, as shown in Fig. 3 at the bottom (visual interfaces are shown in blue and computational interfaces in purple). The available visualization techniques are suitable for depicting data with specific properties. For example, parallel coordinates are capable of visualizing multidimensional data. Therefore, this visual interface can be assigned to expression data as well as patient information. Other visual interfaces are the document viewer, heat map, web-browser, pathway viewer, etc. Caleydo's computational interfaces include the *R statistics toolkit*, *WEKA* and *SQL*, which are assigned to the data sets using the same procedure as before.

The visual analysis expert then compiles a list of operators and assigns interfaces from the compiled list (*cf.*, Step III). In Fig. 3, the operator pool is presented as a series of red blocks. Operators in our use case are for instance query, similarity analysis of images as well as partitional and hierarchical clustering, where partitional

[5] www.ncbi.nlm.nih.gov/pubmed.

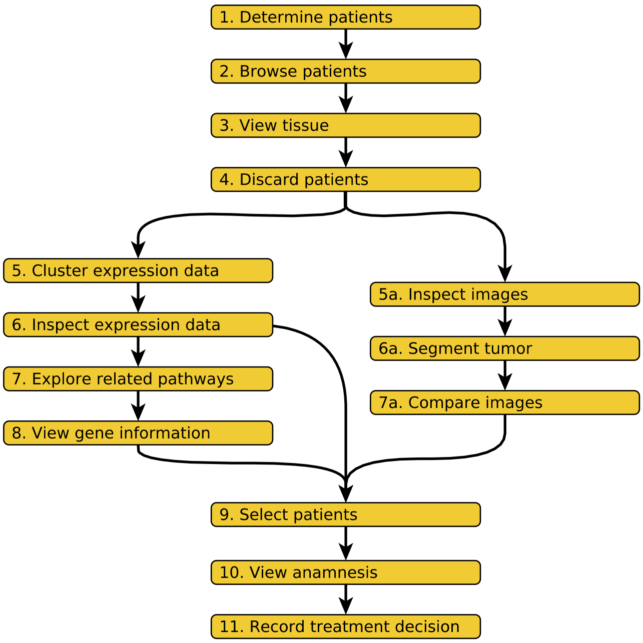

Fig. 4. The workflow of finding a treatment plan for a newly diagnosed cancer patient.

clustering is realized through the *R* interface, and hierarchical clustering through *WEKA*. Note that the operators provided in Fig. 3 are only a sample compilation for the workflow of patient treatment planning.

3.2 Creating the Domain Model

In Step IV, the domain expert, in this case our partner from the Medical University, defines a set of tasks (yellow in Fig. 3) and assigns the tasks to the data on which they operate. A sequence of operators which enable the fulfillment of the task is associated with each task (Step V). One example for our use case is the "Find gene"-task, which is assigned to the gene database and can be accomplished using the Query operator. Note that this step does not include ordering or connecting the tasks.

3.3 Creating the Analysis Session Model

In Step VI, the domain expert defines the workflow as a sequence of tasks, which is the basis for guidance. The following workflow, depicted in Fig. 4, is an example aimed at the goal described before: determining a treatment plan for a patient diagnosed with cancer. Patients are known to respond differently both to therapy and the disease itself based on several factors, including their genetic traits. Therefore, it is crucial to identify the likely course of the disease for a patient under different treatments.

1. **Determine similar patients.** First, the guided analyst filters patients based on their anamnesis (for example, in terms of age, gender, blood values) using a computational approach.
2. **Browse patients.** The analyst explores the patients that remain in the sample and tries to find differences in their conditions.
3. **View tissue.** For those patients, he explores the tissue images, on which the initial diagnosis was based. This is done to make sure that the patients actually present similar manifestations.
4. **Discard patients.** Remove patients with different manifestations in terms of the tissue samples.
5. **Cluster expression data.** To be able to identify patients with similar gene expression patterns, which might indicate common traits and therefore a similar course of the disease, the data are clustered.
6. **Inspect expression data.** The analyst inspects the clustering results to find patterns where the patient under investigation is similar to one group of patients, while different to others. He then selects a group of genes that clearly distinguishes the patient group from others. If the genes' functions are clear to the analyst (e.g., a well-known protoonco or tumor suppressor gene), he can directly jump to Task 9. If this is not the case, he can proceed with the next task to find out more about their function.
7. **Explore related pathways.** To understand the found genes' function, the analyst explores the pathways containing the genes.
8. **View gene information.** Further information about a particular gene is gathered by inspecting its entry in an online database.
9. **Select patients.** With the knowledge that the genes are in fact relevant for the condition, the analyst goes back to the gene expression view; where he selects those patients that are in the same group as the patient under investigation.
10. **View anamnesis.** The analyst then views the anamnesis to judge whether previous courses of actions were successful for similar cases and bases his treatment decision on the findings.
11. **Record treatment decision.** He records the treatment decision in the patient's anamnesis.

Alternatively, instead of conducting an analysis based on gene expression data (Tasks 5 to 8 in the left branch of Fig. 4), the guided analyst can choose to conduct the selection of patients in Task 9 based on an exploration, segmentation, and comparison of tumor images (*cf.*, Tasks 5a to 7a). However, the right branch is only feasible if the disease under investigation causes tumors, visible in imaging data.

Preconditions are defined optionally for each task. For instance, before viewing the tissue slices in Task 3, the analyst needs to filter below 20 patients.

Before the models can be utilized by an analysis system they need to be tailored to the given constraints (*cf.*, Step VII). In our scenario, we do not have the patients' protein expression profiles available, which makes the protein database obsolete. Furthermore, due to access restrictions at the hospital, lab results cannot be a part of the setup. Based on the remaining available setup resources, the automatic pruning of paths is performed. As the exemplary defined workflow samples are rather small, all tasks of the workflow are possible and consequently remain in the analysis session model.

4 Implementation in Real Systems

Before a real system can be developed based on the described model, we first have to briefly discuss ways to create such a model (i.e., authoring). This is followed by an example of a concrete visual analysis system which can make use of the gained information.

4.1 Authoring

To be of use in actual systems, the models described must be available in machine-readable form: either by explicitly creating the model offline, or by capturing interface actions and associating them with tasks at runtime. The interactive method is only suitable for the domain and the analysis session model, since it requires the setup model to be performed.

Tools for offline creation of the model range from dedicated authoring solutions[6] to simple XML editors. While these external tools can be used out-of-the-box, an integrated solution is potentially more powerful: on-the-fly editing and refinement can be tightly bound to the visual data analysis. It enables users to create and refine models—making a live role switch possible—i.e., the analyst becomes the author.

The choice between these two variants is a tradeoff between flexibility and costs. This tight integration of data analysis and authoring requires high initial costs in terms of software engineering. As authoring interfaces are not the focus of this paper, we have defined the models covering our biomedical use case directly in XML.

4.2 Implementation Example: Stack'n'flip

In this section, we give a practical example that realizes a system using a previously authored model.

The "Stack'n'flip" system is grouped into two parts: a space for data visualization, similar to what Shrinvasan and van Wijk [18] call the *Knowledge View*, and a space showing the relations between data, views, and analysis paths, similar to their *Navigation View*. While our realization of this system and application goals are very different from those proposed in [18], the views are conceptually similar. Therefore, these terms were adopted. Two factors distinguish Stack'n'flip from other systems: first, the navigation and the knowledge view are seamlessly integrated, and second, the kind of support (guidance) based on the developed setup model goes well beyond provenance and history.

Some approaches, such as *Aruvi* [18], *History Mechanism* [19], as well as Heer et al.'s temporal work for *Tableau* [20] visualize the exploratory process in a history tree. We take this principal idea a step further by not only presenting history information, but also proposing future steps—either following a predefined path, or showing possible next steps independent of the path. However, in contrast to the *VisComplete* approach of *Vistrails* [6], the path suggestions are not purely derived from previous sessions and workflows, but by employing the authored models. In addition, the associations between previous and possible future analysis steps are made explicit on both levels—the navigation and the knowledge view.

The Stack'n'flip implementation is a part of the Caleydo visualization framework. It is developed in Java and uses the Java OpenGL (JOGL) binding for rendering. The authored model is loaded from a predefined XML representation and is stored in a graph data structure. The interactive support of Stack'n'flip is based on simple graph traversal operations.

4.2.1 Data View

Exploring multiple data sets naturally lends itself to the usage of multiple coordinated views. However, traditional systems often present those multiple views either in tiled windows or in tabs. This does not correspond well to an analysis path, where one data set or view is in focus, while the previous (and possibly the next) data set is also contextually relevant.

To take this into consideration, we propose a stacking of views as depicted in Fig. 5a. The views are projected and rendered on 2D planes in a 2.5D scene, making it related to Collins and Carpendale's *VisLinks* [21], the *Bucket* approach [17], or even Apple's *Cover Flow*. The view in focus is in the center and parallel to the screen. Other views are stacked to the left and right of the focus view, tilted toward the user. The adjacent views are either from the same data set, or from a data set explored in a previous (on the left) or upcoming (on the right) analysis step. This makes it possible to easily relate data in adjacent views. Additionally to conventional highlighting of selected items, visual links [21] are shown between related entities in adjacent views.

4.2.2 Navigation View

The contribution of the proposed approach is not primarily the view arrangement, but the orientation provided by a "map" through the information landscape—the navigation view. When designing such a navigation view, it is important to find a balance between the amount of information presented and the requirement to give as much space as possible to the knowledge view, which contains the actual information.

We realized the map by depicting the network of data sets as large symbols (see (1) in Fig. 5a). Transitions in the data model between loaded data sets are visible at any time, while all possible transitions are shown only when hovering over the associated symbol (see (2) in Fig. 5a). A red exclamation mark followed by a short description indicates that a precondition needs to be met before the analyst can continue to a data set. By picking one of those possible next data sets, the associated data are loaded and shown in the knowledge view. Its symbol is added to the navigation view permanently.

The association between interfaces and data sets, contained in the setup model, is shown as icons on top of the data set symbol. Opening a new interface for a particular data set is achieved by clicking the interface icon.

In case of a guided analysis, the information available through the analysis session model is employed to highlight the recommended path, while still showing other options to proceed (i.e., switching from guided to informed support). The highlighting is realized in red ((2) in Fig. 5a). Recommended interfaces for performing the next task are also shown in red and are opened by default when the data set symbol is clicked. A short description of the current task is presented at the bottom of the navigation view.

[6] For example, Altova Authentic (www.altova.com/authentic.html).

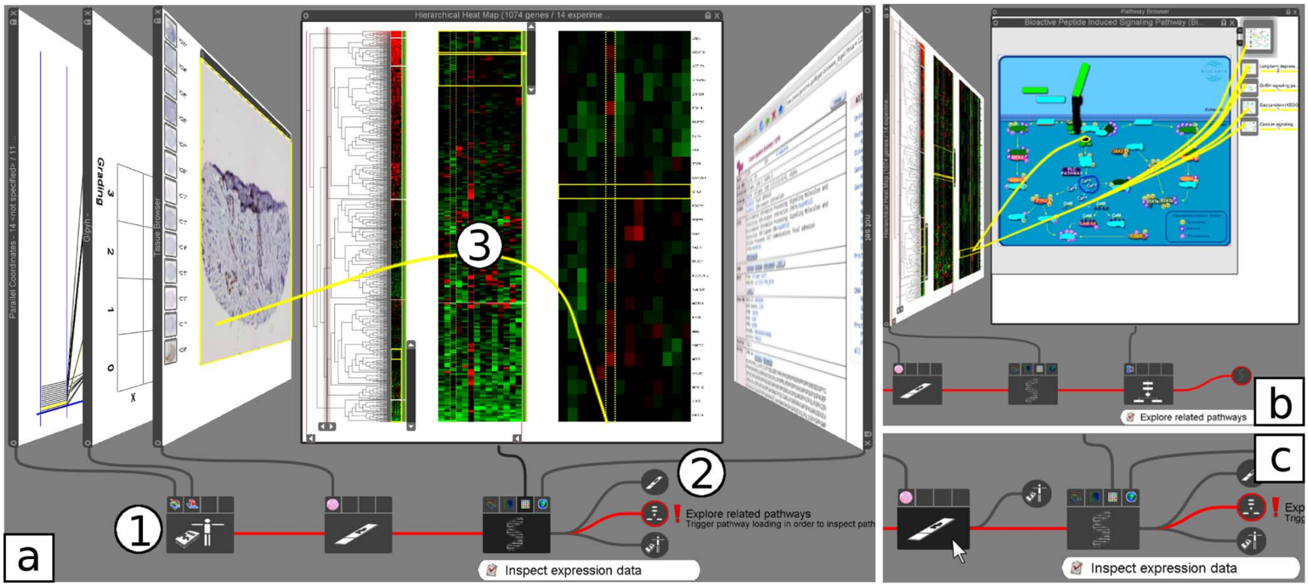
Hierarchical Heat Map (1074 genes / 14 experime...
Parallel Coordinates
Tissue Browser
Grading
Pathway Browser
Bioactive Peptide Induced Signaling Pathway
1
2
3
a
b
c
Explore related pathways
Trigger pathway loading in order to inspect path
Inspect expression data

Fig. 5. (a) Stack'n'flip showing the guided analyst working on Task 6 of the described workflow. A heat map view is shown in the center, a tissue browser on the left, a web-browser on the right and additional stacked views on both sides. The succession of large symbols at the bottom represents the analysis path taken, with each symbol showing a data set (1). On top of the data set symbols, smaller icons show which interfaces are available for the data set. Possible future steps or branches (2) are either highlighted red, symbolizing the suggested analysis path, or gray, showing alternative options. Visual links emphasize relations between the views (3). (b) The analyst has selected a gene and can therefore move on to Task 7 and explore the loaded pathways. Visual links indicate the location of the gene in the pathways. (c) To take an alternative analysis branch or even leave the suggested path the guided analyst can always go back.

4.2.3 Fusion of Navigation and Knowledge View

A key contribution of Stack'n'flip is the seamless integration of navigation view and knowledge view. Open, active views are connected with a curve to their interface symbol on top of the data set symbol, thereby clarifying the relationship between the view and its data set. This association of data sets and views makes it explicit which data set is shown in which view, and also allows the unambiguous use of the same visualization technique for different data sets.

This merging of interactive visualization with analysis context is related to Ma's *Image Graphs* [22], Jankun-Kelly's *P-Set Model* [23] as well as the *Graphical Histories* by Heer et al. [20]. However, Image Graphs and the P-Set Model capture only the analysis process operating on a homogeneous data set. In contrast, Heer's Graphical History view does handle heterogeneous data, but is restricted to history information and therefore does not support real guidance or orientation in the sense of Stack'n'flip.

4.2.4 Discussion of the System

We believe that the Stack'n'flip approach is general enough to be utilized in many different forms. In fact, as it mainly describes how to visually handle transitions in heterogeneous data analysis, it is applicable to a wide range of existing visualization frameworks. We have chosen the 2.5D layout, because we have shown in the past, that it is an effective method for working with multiple, interconnected views (*cf.*, [17]). However, pure 2D layouts, avoiding problems arising from distortion, are of course possible as well.

As such, this system provides orientation when exploring heterogeneous data spaces by showing a history of previously explored data sets, a list of possible connected data sets (in the navigation view) as well as employed visualizations (through the stacking in the knowledge view) and is therefore suitable for the informed analyst. This is especially important in comprehensive analysis of data from different sources, as it requires the analyst to switch back and forth between different views and data sets, refining, for example, selections or filters. Each switch requires mental effort and is potentially confusing for the analyst. By making such switches seamless and keeping the source view as contextual information, the mental effort can be reduced significantly.

The guided analyst benefits from the explicit path laid out for him, while the navigation view shows possible alternatives—thereby encouraging a deviation from the predefined path (and therefore a switch from guided analyst to informed analyst) for a deeper understanding of the data.

5 Further Implications

In this section, we discuss how the models can potentially be utilized for different purposes besides orientation and guidance.

Visual Analytics goes well beyond simply providing the necessary tools for an analysis scenario—it also aims at helping the analysts in choosing the appropriate techniques by defining several processes as best practice solutions for given analytical objectives. These are based on high-level guidelines, such as Keim et al.'s Visual Analytics Mantra [24] or Shneiderman's well established Information Seeking Mantra [25]. Both have found their way into the design of Visual Analytics systems, as they give valuable advice on which kind of tools to provide at which point in the analysis. These processes can be understood as abstract design patterns for visual analysis software. However, they

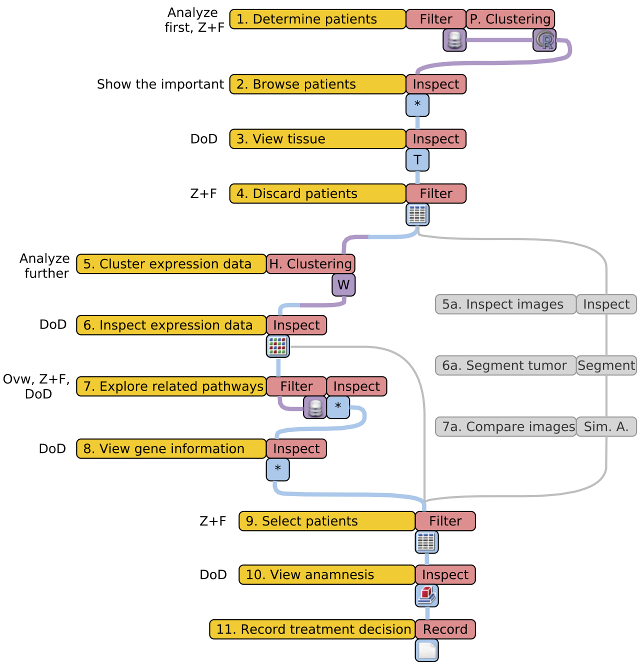

Fig. 6. Sample analysis path showing the chosen interfaces. Jumps between computational (purple) and visual (blue) interfaces denote switches from the data to the view domain and vice versa. High-level interaction mantras can be found as reoccurring patterns.

are too abstract to actually specify visual analysis techniques to be used on a concrete set of data. Hence, most approaches derive suggestions for the analysis from low-level events (mouse clicks, etc.) recorded during previous analysis sessions.

In between high-level mantras and low-level mouse clicks, a gap emerges that neither can fill. A mid-level approach, like the one proposed, makes it possible to formulate analysis sessions as abstractly as needed in order to serve as reusable patterns and at the same time being specific enough to be used for concrete user support, thus merging the best of both worlds. However, in our proposed design approach, high-level mantras are still incorporated. Task 1-6 in Fig. 6 is one example where Keim et al.'s Visual Analytics Mantra—"*analyze first—show the important—zoom, filter and analyze further—details-on-demand*" is evident.

The benefits of using the proposed three-stage model is twofold: on the one hand, it can be employed by a visual analysis system to provide analyst support on different levels, as already discussed in detail; on the other hand, it can help in the design phase of a complex analysis scenario. Benefits gained by the definition of the comprehensive models are:

5.1 Data Selection

The proposed concept makes it possible to dynamically select a set of relevant data sets for a specific analysis goal. Selecting a reduced list of data sets needed in an analysis session makes the analysis more targeted toward the goal. Additionally, the system can anticipate the next steps of an analyst and preprocess, prefetch, or prelayout data in otherwise idle times. For example, fetching of large tissue images from databases can be triggered before the analyst traverses the data set during the interaction. An example for a time-consuming preprocessing step is clustering of gene expression data, based on a selection of patients. Since an analyst can always choose a path different from the preferred one, we propose to prefetch data first for the preferred path, and then for other possible paths, if enough processing power, memory and/or bandwidth is available. By conducting such operations in a separate thread, such a system can utilize modern multicore systems, resulting in a significant speed up.

5.2 Missing Data or Interface Identification

When defining the analysis session model with an analysis goal in mind, interfaces or data sources might be missing from the analysis setup in order to perform a task. Due to the structured authoring process, however, such missing interfaces or data sets are immediately obvious to the domain expert. At this early stage the domain expert can try to fill these gaps by requesting the missing data sets or interfaces from the data manager or visual analysis expert, respectively.

5.3 Post Analysis Optimization

Based on the analysis session model, it is possible to log the workflow path actually taken by a user during an analysis session. Fig. 6 depicts an example path including the interfaces used for each step. Switches between the visual (purple) and computational (blue) domain are of special interest as these are often not seamless and therefore imply a higher mental effort for the user. The extracted data can be utilized to:

- **Optimize the workflow.** By comparing the suggested path with the one taken by the user, a feedback loop can be introduced in the authoring process.
- **Optimize the analysis framework.** Based on the gained insights, the underlying application can be modified to better reflect the user's needs.

5.4 Generalization of Workflows

Analyst support based on history and provenance information is an integral part of various Visual Analytics systems (e.g., [5], [26]). However, by logging low-level application events, the collected information is tightly coupled to one specific setup and cannot be reused for guidance purposes within different applications and tools. With the proposed association of tasks to application and domain independent operators, we detach implementation internal matters from the actual semantic path information. This indirection allows us to employ the collected path information in different analysis setups as well. It is even possible to unhinge the workflow with the associated domain independent operator sequences from a specific setup in order to find an alternative combination of analysis tools.

6 Conclusion and Future Work

In this paper, we have introduced a three-stage, model-driven design process for the interactive visual analysis of heterogeneous data sets, which allows a system to guide a user on two different levels during the analysis. In the first authoring stage, a basic model of the given setup is created which considers data sets from different sources, relations between them and visual as well as computational interfaces operating on them. A visual analysis system employing these models can support the analyst by providing orientation within the conglomerate of data sets, where not only previous but also possible future analysis steps are shown. On top of the setup, a set of domain-specific tasks are defined, forming the domain model. In the last stage, the analysis session model, which contains a workflow with a concrete analysis goal in mind, is defined. These three models are used to actively guide the analyst along a predefined path.

We have demonstrated the concept for a biomedical use case and presented a concrete implementation based on an existing visualization framework. Initial feedback from our project partners at the Medical University of Graz was encouraging and the proposed design approach will become a key component of our joint biomedical research projects.

The current Stack'n'flip implementation provides guidance based on the predefined models via the compact navigation view. In a next step, the navigation view could be switched on demand to a full authoring interface with on-the-fly model editing capabilities. This tight integration of authoring and data analysis has the potential to support a wide range of Visual Analytics applications.

Approaches such as Stack'n'flip demonstrate the usefulness of analyst support and are in fact important for specific data analysis problems. However, as visual analysis often employs highly specialized and expensive tools, using a single superapplication to support users in all their analysis needs is unrealistic. Consequently, there is a strong need for bridging the gaps between these existing, independent tools. In this paper, we have provided the conceptual foundations for doing so. What is left to do is to solve the technical problems of such a multitool scenario. In the spirit of the Snap-Together Visualization [27], we have previously explored possibilities to visually link information across applications [28]. The approach works without a common database on the basis of ID-Strings that are collected from minimally modified applications (e.g., via plug-ins) and matched by a light-weight management application. Thereby identified related entities are connected by visual links. We plan to extend this idea with model-driven guidance and possibly also implementing a Stack'n'flip-like scenario with independent tools.

Collaboration is another important topic for solving complex domain problems. For a complex analysis, experts from multiple domains with different background knowledge are required. We believe that the proposed concept has the potential to also serve as a basis for supporting this collaborative scenario.

Acknowledgments

The authors want to thank the team of Professor Zatloukal from the Medical University of Graz for providing the use case and the valuable insight in their work. This work was funded in part by the Austrian Research Promotion Agency (FFG) through the *InGenious* project (385567) and the Caleydoplex project (P22902) granted by the Austrian Science Fund (FWF), the German Research Foundation (DFG), research training school 1387 *dIEM oSiRiS*, and the *VisMaster Coordination Action* (FET-Open grant number 225429).

References

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

[2] V.L. O'Day and R. Jeffries, "Orienteering in an Information Landscape: How Information Seekers Get from Here to There," *Proc. SIGCHI Conf. Human Factors in Computing Systems (INTERACT '93 and CHI '93),* pp. 438-445, 1993.

[3] T. Jankun-Kelly and K. Ma, "Visualization Exploration and Encapsulation via a Spreadsheet-Like Interface," *IEEE Trans. Visualization and Computer Graphics,* vol. 7, no. 3, pp. 275-287, July-Sept. 2001.

[4] D. Gotz and Z. Wen, "Behavior-Driven Visualization Recommendation," *Proc. Conf. Intelligent User Interfaces (IUI '09),* pp. 315-324, 2009.

[5] W. Willett, J. Heer, and M. Agrawala, "Scented Widgets: Improving Navigation Cues with Embedded Visualizations," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 6, pp. 1129-1136, Nov./Dec. 2007.

[6] D. Koop, C.E. Scheidegger, S.P. Callahan, H.T. Vo, J. Freire, and C.T. Silva, "VisComplete: Automating Suggestions for Visualization Pipelines," *IEEE Trans. Visualization and Computer Graphics,* vol. 14, no. 6, pp. 1691-1698, Nov./Dec. 2008.

[7] A. Perer and B. Shneiderman, "Systematic yet Flexible Discovery: Guiding Domain Experts through Exploratory Data Analysis," *Proc. ACM Conf. Intelligent User Interfaces (IUI '08),* pp. 109-118, 2008.

[8] M. Streit, H. Schulz, D. Schmalstieg, and H. Schumann, "Towards Multi-User Multi-Level Interaction," *Proc. Workshop Collaborative Visualization on Interactive Surfaces (Part of VisWeek '09),* pp. 5-8, 2009.

[9] K.A. Marrs, S.A. Steib, C.A. Abrams, and M.G. Kahn, "Unifying Heterogeneous Distributed Clinical Data in a Relational Database," *Proc. Symp. Computer Applications in Medical Care,* pp. 644-648, 1993.

[10] M.D. Lieberman, S. Taheri, H. Guo, F. Mir-Rashed, I. Yahav, A. Aris, and B. Shneiderman, "Visual Exploration across Biomedical Databases," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 8, no. 2, pp. 536-550, Mar./Apr. 2011.

[11] P.P.S. Chen, "The Entity-Relationship Model toward a Unified View of Data," *ACM Trans. Database Systems,* vol. 1, no. 1, pp. 9-36, 1976.

[12] T. Munzner, "A Nested Process Model for Visualization Design and Validation," *IEEE Trans. Visualization and Computer Graphics,* vol. 15, no. 6, pp. 921-928, Nov./Dec. 2009.

[13] C. Stary, "TADEUS: Seamless Development of Task-Based and User-Oriented Interfaces," *IEEE Trans. Systems, Man, and Cybernetics, Part A,* vol. 30, no. 5, pp. 509-525, Sept. 2000.

[14] A. Puerta, "A Model-Based Interface Development Environment," *IEEE Software,* vol. 14, no. 4, pp. 40-47, July/Aug. 1997.

[15] N. Gehlenborg, S.I. O'Donoghue, N.S. Baliga, A. Goesmann, M.A. Hibbs, H. Kitano, O. Kohlbacher, H. Neuweger, R. Schneider, D. Tenenbaum, and A. Gavin, "Visualization of Omics Data for Systems Biology," *Nature Methods,* vol. 7, no. 3, pp. 56-68, 2010.

[16] M. Streit, A. Lex, M. Kalkusch, K. Zatloukal, and D. Schmalstieg, "Caleydo: Connecting Pathways and Gene Expression," *Bioinformatics,* vol. 25, no. 20, pp. 2760-2761, 2009.

[17] A. Lex, M. Streit, E. Kruijff, and D. Schmalstieg, "Caleydo: Design and Evaluation of a Visual Analysis Framework for Gene Expression Data in Its Biological Context," *Proc. IEEE Symp. Pacific Visualization (PacificVis '10).* pp. 57-64, 2010.

[18] Y.B. Shrinivasan and J.J. van Wijk, "Supporting the Analytical Reasoning Process in Information Visualization," *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI '08),* pp. 1237-1246, 2008.

[19] M. Kreuseler, T. Nocke, and H. Schumann, "A History Mechanism for Visual Data Mining," *Proc. IEEE Symp. Information Visualization (InfoVis '04),* pp. 49-56, 2004.

[20] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala, "Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation," *IEEE Trans. Visualization and Computer Graphics,* vol. 14, no. 6, pp. 1189-1196, Nov./Dec. 2008.

[21] C. Collins and S. Carpendale, "VisLink: Revealing Relationships amongst Visualizations," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 6, pp. 1192-1199, Nov./Dec. 2007.

[22] K. Ma, "Image Graphs—A Novel Approach to Visual Data Exploration," *Proc. IEEE Conf. Visualization (Vis '99),* pp. 81-88, 1999.

[23] T.J. Jankun-Kelly, K. Ma, and M. Gertz, "A Model and Framework for Visualization Exploration," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 2, pp. 357-369, Mar./Apr. 2007.

[24] D.A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler, "Challenges in Visual Data Analysis," *Proc. Conf. Information Visualisation (IV '06),* pp. 9-14, 2006.

[25] B. Shneiderman, "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations," *Proc. IEEE Symp. Visual Languages (VL '96),* pp. 336-343, 1996.

[26] L. Bavoil, S. Callahan, C. Scheidegger, H. Vo, P. Crossno, C. Silva, and J. Freire, "VisTrails: Enabling Interactive Multiple-View Visualizations," *Proc. IEEE Conf. Visualization (VIS '05),* pp. 135-142, 2005.

[27] C. North and B. Shneiderman, "Snap-Together Visualization: A User Interface for Coordinating Visualizations via Relational Schemata," *Proc. ACM Conf. Advanced Visual Interfaces (AVI '00),* pp. 128-135, 2000.

[28] M. Waldner, W. Puff, A. Lex, M. Streit, and D. Schmalstieg, "Visual Links across Applications," *Proc. Conf. Graphics Interface (GI '10),* pp. 129-136, 2010.

Marc Streit received the master's (2007) and PhD (2011) degrees from the Graz University of Technology. He is a postdoctoral researcher at the Institute for Computer Graphics and Vision at Graz University of Technology, Austria. His research focuses on the Caleydo project (www.caleydo.org) where he works on topics including information visualization, Visual Analytics and bioinformatics. He is a student member of the IEEE.

Hans-Jörg Schulz received the diploma (2004) and the PhD degree (2010) from the University of Rostock. At present, he is a postdoctoral researcher at the graduate school "dIEM oSiRiS" in Rostock. There, he develops visualizations for computational systems biology. His main interests concern the visualization of special graph classes, like bipartite or interval graphs, as well as graph representations aside node-link layouts.

Alexander Lex studied in Graz, Austria and Hamilton, Canada and received the bachelor's and master's degree in computer science from Graz University of Technology where he is now involved in the Caleydo project as a PhD candidate. His research interests are InfoVis, Visual Analytics, HCI, and Bioinformatics. When he is not in the lab he enjoys traveling and riding his snowboard in the Austrian mountains. He is a student member of the IEEE.

Dieter Schmalstieg received the Dipl-Ing, Dr techn, and Habilitation degrees from the Vienna University of Technology in 1993, 1997, and 2001, respectively. He is a professor of Virtual Reality and Computer Graphics at Graz University of Technology (TUG), Austria. His current research interests are augmented reality, virtual reality, real-time graphics, 3D user interfaces, ubiquitous computing, and information visualization. He is a member of the IEEE.

Heidrun Schumann graduated at the University of Rostock (1977 Diploma, 1981 PhD degree, 1989 Habilitation) and is heading the Computer Graphics Research Group at the Institute for Computer Science at the University of Rostock since 1992. Her research covers Information Visualization, Visual Analytics, and Rendering. Her current projects, supported by research institutions and industry, include scalable frameworks for information visualization and adaptive visual interfaces.