Improving the Analysis Process of Computerized Protocols and Temporal Data by Integrated Visualization Techniques

Evaluation Aspects

Wolfgang Aigner
Silvia Miksch
Bettina Thurnher
Stefan Biffl

Vienna University of Technology
Institute of Software Technology & Interactive Systems (ISIS)

Asgaard-TR-2005-9 August 2005

Authors: **Wolfgang Aigner**
Silvia Miksch
Bettina Thurnher
Stefan Biffl

{aigner, silvia, thurnher, biffl}@ifs.tuwien.ac.at
http://ieg.ifs.tuwien.ac.at

Contact: **Vienna University of Technology**
Institute of Software Technology & Interactive Systems (ISIS)

Favoritenstraße 9-11/188
A-1040 Vienna
Austria, Europe

Telephone: +43 1 58801 18833
Telefax: +43 1 58801 18899
Web http://ieg.ifs.tuwien.ac.at

Improving the Analysis Process of Computerized Protocols and Temporal Data by Integrated Visualization Techniques - Evaluation Aspects

Wolfgang Aigner, Silvia Miksch, Bettina Thurnher, Stefan Biffl

Institute of Software Technology & Interactive Systems (ISIS)
Vienna University of Technology
Favoritenstraße 9-11/188, A-1040 Vienna, Austria, Europe
Tel: +43 1 58801 18833, Fax: +43 1 58801 18899

Abstract

Currently, visualization support for patient data analysis is mostly limited to the representation of directly measured data. Contextual information on performed treatment steps is an important source for finding reasons and explanations for certain phenomena in the measured patient data. But this kind of information is mostly spared out in the analysis process.

We describe interactive visualization methods to integrate and combine classical data visualization with the visualization of treatment information in terms of logic and temporal aspects called *CareVis*. We provide multiple simultaneous views to cover different aspects of a complex underlying data structure of treatment plans and patient data.

The user-centered development approach applied for these interactive visualization methods has been guided by user input gathered via a user study, design reviews, and prototype evaluations. Furthermore, the core aspect of visualizing treatment plans in the temporal domain has been investigated via a comparative empirical user study. Main results are that PlanningLine users make fewer mistakes and are faster in conducting tasks than users of a traditional visualization technique.

Key words: protocol-based care, information visualization, treatment plans, evaluation, user-centered design, temporal uncertainty

Email address: `{aigner, silvia, thurnher, biffl}@ifs.tuwien.ac.at` (Wolfgang Aigner, Silvia Miksch, Bettina Thurnher, Stefan Biffl).
URL: `http://www.ifs.tuwien.ac.at` (Wolfgang Aigner, Silvia Miksch, Bettina Thurnher, Stefan Biffl).

Asgaard-TR-2005-9

1 Introduction

Visualization plays an important role in the task of intelligent data analysis, either as integral part by using human perception for driving the analysis process, for presenting results, or both. In the medical domain, mostly patient data measurements, either high-frequency data for intensive care settings, or low-frequency data i.e. for long term studies are used as basis for the analysis. Due to that, current visualization methods are mostly bound to the representation of such measured patient data, which could be subsumed under the term "data visualization".

But there is much more information to be taken into consideration in the analysis process. One of these informational pieces is treatment information. That is basically information about which treatment steps have been taken at which time, for how long, how often, and the like. So far, contextual information on treatment steps and performed treatments is mostly excluded from first hand data analysis. The integration is either only performed mentally by physicians or worse, contextual information is lost completely. But such information could be an important source for finding reasons and explanations for certain phenomena in the measured patient data. The goal of this work is the integration and combination of various kinds of data as well as information and presenting it in a coherent way for supporting the data analysis process.

Computer-supported protocol-based care is a field of research that aims for supporting semi-automatically the treatment process along protocols by the use of information technology. The core entity, medical treatment plans, are complex documents, currently mostly in the form of prose text including tables and figures [1]. Protocol-based care utilizes clinical protocols to assist in quality improvement and reduce process irregularities. Such clinical protocols are a standard set of tasks that define precisely how different classes of patients should be managed or treated. They can be seen as reusable definitions of a particular care process. Not much work has been done in order to communicate the computerized treatment plans to the medical staff and even less for combining this with the presentation of patient data when treating a patient along a plan for monitoring and analytic tasks. The integrated visualization of medical treatment plans and patient data could be of great assistance to ease the complex task of analyzing medical data and protocols.

2 Related Work

The most widely used visual representation of clinical guidelines are so-called *flow-chart algorithms*, also known as *clinical algorithm maps* [2]. A standard for this kind of flow-chart representation has been proposed by the *Committee on Standardization of Clinical Algorithms* of the *Society for Medical Decision Making* [3]. The proposed standard includes a small number of different symbols and rules on how to use them. One additional feature to standard *flow-charts* are *annotations* that include further details, i.e. citations to supporting literature, or clarifications for the rationale of decisions. A big advantage of using flow-charts is that they are well known among physicians and require minimal additional learning effort. A drawback of basic flow-chart representations is their immense space consumption if more complex situations are depicted where overview is lost easily. Temporal information can only be represented implicitly on a very coarse level in terms of an item's relative position within a sequence. Furthermore, flow-charts cannot be used to represent concurrent tasks or the complex conditions as used in *Asbru*. Clinical algorithm maps were intended to be used on paper and have never been enriched by computer support such as navigation or versatile annotation possibilities.

Other scientific work [4–6] on visual representations focused on visualizing patient data over time or plan execution over time. Research projects dealing with protocol-based care include *GLARE*, *GUIDE*, *Protégé*, *GLIF*, *PROforma*, and *GASTON*. (A comprehensive overview of related protocol-based care projects can be found in [7] and [8].) Only some of the available projects dealing with protocol-based care provide any graphical representations. The listed ones include such graphical representations, but most of them only focus on authoring plans. They use a flowchart- or workflow-like presentation depicting the elements used in their formal representation. A more detailed discussion of the quoted projects can be found in [9].

LifeLines [6] utilize horizontal bars to represent the temporal location and duration of data elements. They were applied for representing personal histories and patient records. For organizing the elements, so-called "facets" are introduced for grouping the data which can be expanded and collapsed. When collapsed, only a very small and geometrically as well as semantically downscaled version without textual labels is shown. Furthermore, information can be encoded via the height and color of individual bars. Additional information can be provided on demand in a linked view as for example x-ray images or the like. Due to their simplicity they are easy to understand but some important features are missing, such as the ability of depicting hierarchical data. The visualization is mainly used retrospectively for analytic and presentation tasks rather than for planning. Moreover, LifeLines cannot represent temporal indeterminacies.

With *Paint Strips* [10] the idea of Timelines is enriched by a painting metaphor indicating that the displayed bars are drawn by a paint roller. A paint roller at an end of a bar means that this line can expand by moving the roller until a wall is reached. This way the maximum duration and earliest start or latest end, depending on which end of the painting strip the paint rollers are attached to, are defined and indeterminacies shown. Another addition is the possibility to combine strips. The relationship of Paint Strips can be fixed, which means that if one strip moves, the other one moves in the same extent as well. This relationship is indicated graphically by connecting the involved paint rollers and attaching them to a weight at the end of a "rope" which is able to move the rollers. Paint Strips were especially developed for medical applications but can be used elsewhere as well. Due to the simplicity of the paint strip metaphor, some time annotation attributes such as durations independent of the differences between start and end points, different granularities, undefined values, or a reference point cannot be visualized.

Temporal Objects [11] were developed for depicting temporal data with different granularities. Temporal data that is defined in a coarser granularity level than the visual representation is depicted by two encapsulated bars for minimum and maximum duration with a cap at each end for the start and end intervals. Although being visually similar to PlanningLines, this technique has been developed to serve a fundamentally different purpose (granularity vs. indeterminacy), is of static nature, and less flexible.

The *Time Annotation Glyph* [12] is based on the same attribute set as the PlanningLine, but uses the metaphor of bars that lie on "pillar". Four vertical lines on the base specify earliest and latest starting and ending times. These pillars support a bar that represents the maximum task duration. On top of the maximum-duration bar, a minimum-duration bar lies upon two diamonds for latest start and earliest end. Furthermore, undefined parts are displayed in gray and different temporal granularities are indicated by using zigzag lines. Because of this metaphor, a few simple time-attribute constraints can be understood intuitively. For example, the minimum duration cannot be shorter than the interval between latest start and earliest end – if it was, the minimum duration bar would fall down between its supports. All attributes may be defined relative to a reference point that is also represented graphically. Disadvantages of Time Annotation Glyphs are their relatively steep learning curve, difficult integration into currently used and well-known techniques, and less visual cues in order to help maintaining time-attribute constraints as with PlanningLines.

3 Integrated Visualization Approach

3.1 Data Characteristics

For representing treatment plans, the plan representation language *Asbru* is used. It is a time-oriented, intention-based, skeletal plan specification representation language that is used in the *Asgaard* Project to represent clinical guidelines and protocols in XML [13].

Basically, we want to integrate three different kinds of information:

- treatment plan specification data
- treatment plan execution data (instantiation and execution of a treatment plan)
- patient data (time oriented)

Analyzing the type and structure of this data formulated in *Asbru* yields a number of visualization relevant characteristics:

- time-oriented data (execution and planning data including a rich set of time attributes to represent uncertainties)
- logical sequences
- hierarchical decomposition
- flexible execution order (sequential, parallel, unordered, any-order)
- non-uniform element types
- state characteristics of conditions

3.2 Multiple Simultaneous Views

As described above, the underlying data structure we want to communicate to medical domain experts is very complex. Since none of the examined visualization methods can be used to represent all needed data characteristics, we had to decide whether to introduce a new visualization method that allows to depict every data aspect in one view or using the approach of *multiple views*. Multiple views are a well known information visualization technique, whereby a number of representations that focus on different aspects of the data are provided for a common underlying data structure [14,15].

Several reasons led to the clear decision of using multiple simultaneous views. Since we were putting forward a user-centered approach, the goals of providing representations that are easy to comprehend and require as little learning effort as possible were paramount. Therefore, using representations familiar to the domain experts was obvious. Furthermore, we perceived that a single representation would be far too complex, cognitively overwhelming, and surely not optimal to fulfill our prerequisites. Our user study clearly showed that *clinical algorithm maps* are frequently used in daily work and education of physicians to represent treatment plans. *GANTT-charts* and *LifeLines* were identified as quite well known techniques for representing temporal aspects. Since these methods in combination are capable to serve our needs, we chose them as basis for our design.

A further important factor related to this are the different tasks users want to accomplish by using our interactive visualization methods. The three primary tasks are to become acquainted with a specific treatment method, to get guidance in the treatment process, and to analyze the treatment process. Furthermore, these three tasks are temporally as well as semantically intertwined, which led to the decision of using multiple *simultaneous* views rather than sequential ones.

A successful introduction of a multiple view approach in the medical domain has been demonstrated by Zeng and Cimino [16]. They developed a web-based hypermedia system for physicians and clearly showed the advantages of multiple views in the medical domain.

Having introduced the data characteristics and reasons why we have chosen to use multiple simultaneous views, we present these views in detail next.

3.3 Views

Basically, we divided the underlying data structure along the lines of logical structure and temporal aspects. Hence, in *CareVis* we provide a *Logical View* and a *Temporal View* along with a *QuickView Panel*. These distinct views are presented simultaneously and divide the screen in the following manner (see Fig. 1). The QuickView Panel is located on top of the screen displaying the most important patient parameters and plan variables at a prominent position. Below that, the screen is divided vertically by the logical view on the left-hand side and the temporal view on the right-hand side. The logical view presents treatment plans in terms of their logical structure (hierarchical decomposition, plan elements, execution order, conditions). The temporal view on the other side focuses on the temporal aspects of treatment plans and measured patient data as well as plan variables (temporal aspects of plan elements, temporal uncertainties, hierarchical decomposition).

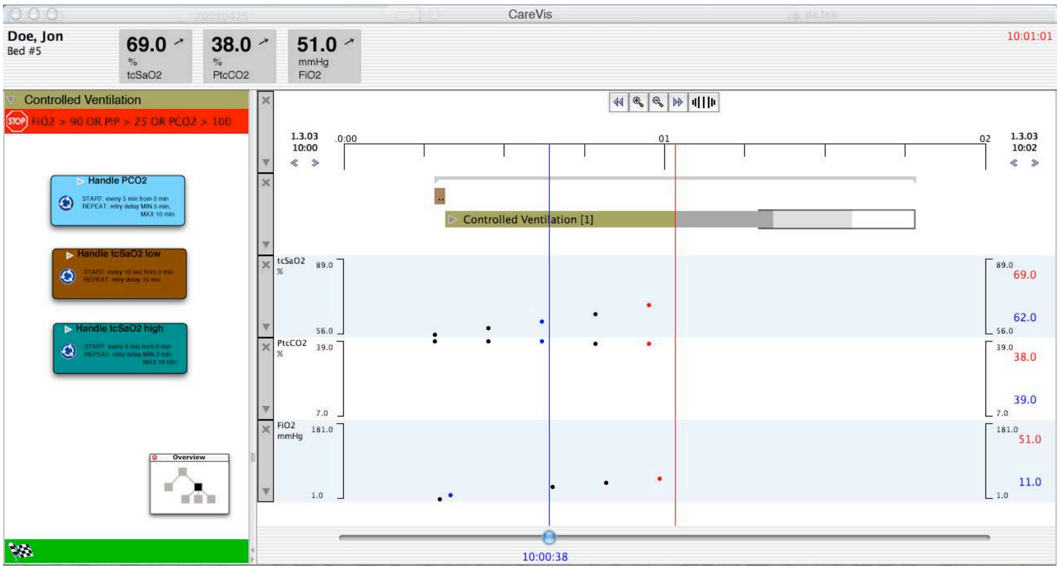


Fig. 1. Application window (top: QuickView Panel, lower left: Logical View, lower right: Temporal View) (Mockup).

3.3.1 Logical View

The logical view on the left-hand side of the screen provides a representation of the treatment plan specification data. The applied visualization technique *AsbruFlow* is based on the idea of flow-chart-like *clinical algorithm maps* [2] that are well known amongst physicians. This concept has been extended in order to be able to depict the characteristics of a treatment plan formulated in *Asbru*.

A set of six visual elements is used to depict the single steps within the body of an *Asbru* plan - Plan, User-performed plan, Ask element, Cyclical plan, If-Then-Else Element, and Variable assignment. For depicting plan conditions and the execution order of the plan steps, an enclosing frame was created. The largest part of the representation is dedicated to the plan body of the depicted plan along with the *execution sequence indicator*. Its four possible symbols specify the execution order of the elements within the plan body – sequentially, parallel, any-order, or unordered.

The visual exploration of a treatment plan is supported by several interactive features. Plan elements that contain sub-elements are indicated by small gray triangles right in front of their labels. By clicking the triangle, the user navigates down the hierarchy, revealing the child elements of the chosen element. This navigational technique is well known from file system viewers as for example the *Finder* of the Macintosh™ system.

In order to prevent getting lost within a plan by navigation, two *focus+context* techniques are applied. Firstly, there is the *overview+detail* technique that uses a small window containing a downscaled, simplified tree overview where the current position within a plan is highlighted. This small overview window can be toggled on or off. The second technique used is the *fisheye view* which distorts elements that are out of the current focus geometrically by shrinking and moving them.

For a comprehensive description of the visualization methods used within the logical view refer to [17].

3.3.2 Temporal View

The temporal representation of treatment plans is based on the idea of *LifeLines*. This concept has been extended for enabling the display of hierarchical decomposition as well as the complex time annotations used in *Asbru*. These new visual elements are called *LifeLines+* and *PlanningLines*, respectively. LifeLines+ allow the interactive representation of temporal intervals with hierarchical decomposition and simple element characteristics. On top of that, PlanningLines allow the depiction of temporal uncertainties via a novel glyph.

The temporal view is used to display the temporal aspects of plans and patient data in the past, present, and future, whereas only plans can be shown in future including temporal uncertainties.

Fisheye deformation is used to magnify the focus part of the time scale while the context part is demagnified. This fisheye functionality can be turned on and off via a button above the time scale. Furthermore, the time scale can be zoomed and shifted interactively.

The facets below the temporal treatment plan representation are used for displaying measured patient data and plan variables. This work focuses on the integrative aspect and representing treatment plan information. Several novel approaches for visualizing time-oriented data that can be used for the graphical representation of patient data are described in [18].

3.4 PlanningLine Glyph

Before presenting the PlanningLine glyph in detail, we introduce the design goals that drove the development. In principle, these design goals can be divided into two major areas – single-glyph-related and multiple-glyph-related goals. Particularly, goals related to a single glyph are to provide a visual representation of temporal indeterminacies of a single activity, facilitate the identification of (un)defined attributes, support in maintaining logical constraints, and to give a visual impression of how distinctive the individual and overall uncertainties are. Goals related to multiple glyphs (parts of a plan or complete plans) are foremost to support the identification of critical areas, facilitate the understanding of activity interrelationships and hierarchy as well as the comparison of activities. Overall design goals are to provide an intuitive visual representation with low learning effort that can easily be integrated into current techniques.

3.4.1 Design Concept

For our glyph, the concept of LifeLines [6] has been extended to enable the display of hierarchical decomposition as well as a set of complex time annotations to reflect temporal indeterminacies. These new visual elements are called *PlanningLines* and allow for the interactive representation of temporal intervals with hierarchical decomposition and simple element characteristics. The glyph consists of two encapsulated bars, representing minimum and maximum duration, that are bounded by two caps that represent start and end intervals (see Fig. 2).

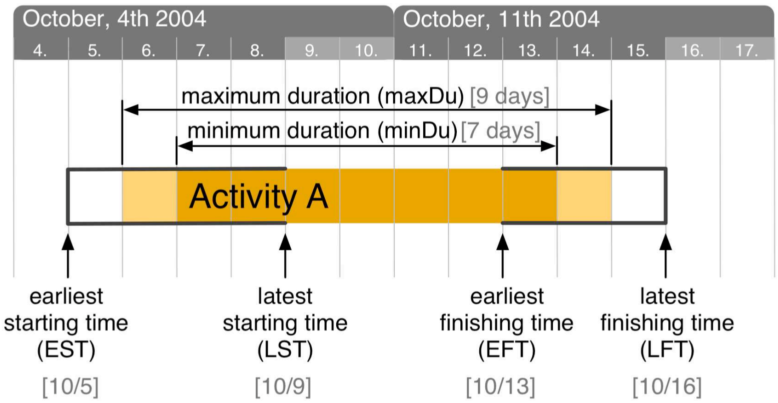


Fig. 2. PlanningLine and Represented Time Attributes.

Temporal Attributes. For reflecting temporal uncertainties, begin, end, and duration of activities are modeled as intervals including a set of six attributes:

- start interval
  - · earliest starting time [EST]
  - · latest starting time [LST]
- end interval
  - · earliest finishing time [EFT]
  - · latest finishing time [LFT]
- duration
  - · minimum duration [minDu]
  - · maximum duration [maxDu]

This implies that the actual start of an activity might be any instant within the start interval and an activity's end any instant within the end interval while the duration of the activity might be any span between minimum and maximum duration. Moreover, the start and end attributes can either be defined absolutely on the time scale (e.g., Dec 10th) or as shifts relative to a reference point (e.g., two days after the end of Activity A).

Mental Model. In order to facilitate a straight forward explanation of the visual representation we use a simple mental model. The two black caps representing begin and end interval are mounted at the time scale. These caps are holding two encapsulated bars that represent minimum and maximum duration (see Fig. 2). Furthermore, the bars can be shifted within the constraints of the two mounted caps.

3.4.2 Attribute Constraints

For maintaining a valid attribute set, a number of logical constraints have to be followed:

- The interval between the latest starting time [LST] and the earliest finishing time [EFT] defines the smallest possible and the interval between the earliest starting time [EST] and the latest finishing time [LFT] defines the largest possible time window for the duration of an activity.
- For each single time point in the starting interval [EST, LST], there must exist at least one duration out of [minDu, maxDu], which allows the finishing interval [EFS, LFS] to be reached.
- Each single time point in the finishing interval [EFT, LFT] must be reachable by at least one duration out of [minDu, maxDu] from the starting interval [EST, LST].
- Each duration must connect one instant in the starting interval with one instant in the finishing interval.
- EST $\leq$ LST, EFT $\leq$ LFT, EST $\leq$ EFT, LST $\leq$ LFT, minDu $\leq$ maxDu

Our glyph helps to maintain these constraints visually. First of all, the possible durations have to be longer than the interval between latest start and earliest finish [LST, EFT] – if this would not be the case, the inner bars would fall out of the holding caps. Secondly, the possible durations cannot be longer than the interval between earliest start and latest finish [EST, LFT] – otherwise, the inner bars would not fit into the caps. Furthermore, the inner bars have to be long enough to reach the end cap if shifted completely to the left which satisfies constraint number two – otherwise, the bars would fall out of the right cap. Analogous, this is applied for shifting the bars to the right which satisfies constraint number three. Several other implicit constraints, as for example that the earliest finishing time, might not be before the earliest starting time can also easily be maintained and spotted visually at a glance.

3.4.3 Special Constellations

All temporal attributes can be specified optionally since they may not all be known. However, the remaining ones still have to maintain the constraints that are applicable and attributes might be calculated (e.g., the minimum duration by the interval between latest start and earliest end). Undefined attributes are not drawn at all and attributes that have been calculated are represented in lighter colors (e.g., gray instead of black for start and end interval attributes). If only the latest start (without earliest start) or earliest end (without latest end) are known, they are represented as diamonds (filled, rotated squares) that support the duration bars.

3.4.4 View coupling

Logical view and temporal view are tightly coupled in three different ways.

(1) A *common color palette* is used among the views for coloring plan elements.
(2) *Linking + brushing* through synchronous selection. If an element is selected in either the temporal or the logical view, the corresponding element(s) are selected in both views. This ensures a quick recognition and comparison of an element of interest in both views.
(3) *Navigation Propagation.* In contrast to the already presented methods, navigational procedures within a plan are not propagated to the coupled view, thus providing no automatic synchronization. Instead, view synchronization is user triggered via drag and drop. If the user wants to propagate the current position within a plan from one view to the other, she selects the desired element, moves it to the other view and drops it there. This user interaction initiates a navigation of the selected view to the desired position.

Figure 1 shows the *CareVis* application window during analysis of a ventilation plan. The “tcSaO2” facet indicates that the corresponding parameter is increasing. When referring to the PlanningLine display located above in the temporal view, we find that an instance of the “Controlled Ventilation” plan was performed while the parameter was increasing. To get more detailed information about this plan, we can drag the PlanningLine into the *AsbruFlow* panel (logical view) on the left-hand side, where the logical substeps of the plan are revealed.

4 Prototype Implementation

The main aim of implementing our prototype was to demonstrate the most important characteristics of the design, to proof that our concept works at all, to get a better impression of the look and feel, and to see how the interaction patterns are working. Furthermore, it should act as basis for further evaluation by potential users.

The prototype was implemented using the programming language *Java* (JDK 1.4.1). The visual elements of the prototype are based on Java's *Swing* standard component library.

4.1 Approach

For implementing the prototype we applied a *rapid prototyping* approach with small development cycles (about two weeks). Hence, the prototype evolved step by step whereas the analysis and design steps were done only for the next development cycle and existing parts were getting constantly refactored.

This approach has been taken from the *Extreme Programming (XP)* technique [19]. Furthermore, another XP technique, namely "Unit Testing", was applied for the classes implementing core functionality of our prototype. This means writing tests first and doing the actual implementation after that, which has a lot of advantages. First of all one can see instantly if the implemented part works by running the test and is forced to work out the external behavior clearly upfront by writing the test.

4.2 MVC Paradigm

The basic structure of the prototype resembles the *MVC* paradigm. *MVC* is the acronym for the three components: *Model*, *View*, and *Controller* (see Fig. 3). Whereas the *model* is the core element building up the system structure. The *model* can have one or more *views* associated to it. *Views* are passive, meaning that they do not change or manipulate the *model*. *Views* represent the *model* or parts of it (mostly visually). *Controllers* in contrary are associated elements that are responsible for manipulating (parts of) the model. In return, changes caused by *controllers* are reflected by the representing *views*.

Furthermore, the coupling of these three parts is very loose: The *model* does not and should not know anything about its associated *views* or *controllers*. Often, the separation between *views* and *controllers* is not that strict and

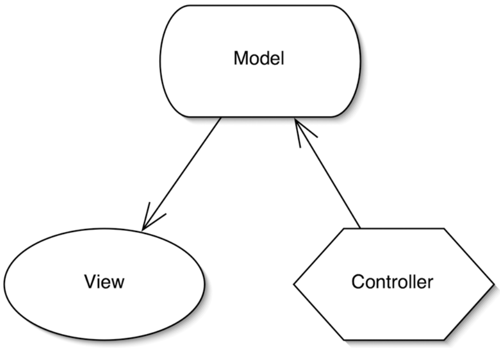


Fig. 3. Model View Controller paradigm.

elements act both, as *views* and *controllers*.

In our case, *Asbru* plans as well as parameters and variables are the *models*, the *Logical View* and the elements of the *Temporal View* are *views* on them.

4.3 Logical View Implementation

The *Logical View* of our tool is used to visualize the logical structure of an *Asbru* plan. For displaying the flowchart-like part of our representation to depict plan step elements, we use the graph drawing framework *JGraph* [20].

JGraph is a general purpose graph drawing framework developed by Gaudenz Alder [20]. It is a flexible, small, and powerful package using the MVC model (see 4.2). It is structured analogous to the standard *Swing* component *javax.swing.JTree*.

JGraph uses a general model of graphs consisting of *nodes* and *edges* connecting them. Views are defined for representing graph cells (nodes and edges) which in turn utilize renderers that do the actual screen painting work (using the *Flyweight* pattern [21]).

4.4 Temporal View Implementation

All elements of the *Temporal View*, except diagrams, are generally views of temporal objects (objects implementing the *Timed* interface or one of its subinterfaces).

This way, an application independent implementation is ensured because views are coupled to *Asbru* plan model classes via time interfaces rather than directly.

For coupling the views, a managing entity is needed. This element is embodied by a *view manager class* that holds references to all views of *Asbru* plans in the system. These views do not represent plans themselves visually but use other view elements for that matter: *LifeLines+* and *PlanningLines* in the temporal view, and the *PlanGraph* element in the logical view.

A heavily used architectural element for interaction event notification is the *Observer pattern* [21]. The user interface (UI) event model has the following event types:

- *select*: A UI element was getting selected.
- *expand*: A UI element was getting expanded.
- *collapse*: A UI element was getting collapsed.
- *propagate*: The propagation of the current selection has been triggered.

These events are encapsulated in a *ViewSelectionEvent* class and sent to all registered listeners. The event class holds references to the object sending or resending the event as well as to the object originally firing the event. Classes that are interested in receiving such events have to implement a particular listener interface. This interface defines a set of listener methods that are called upon when the associated event types are fired.

Due to the structure of views, a layered dispatch is used for delivering events. This means that events are passed up in the hierarchy as long as they affect the next level. When the root view manager is reached, it passes the event down to all other plan views registered for receiving UI events (see Fig. 4).

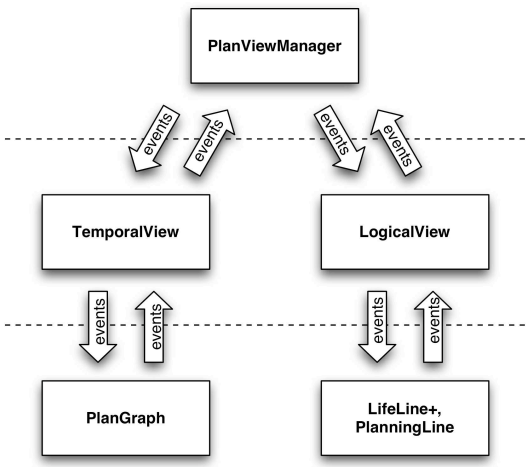


Fig. 4. Layered dispatch for views.

5 User-Centered Design

"New medical information systems, no matter how fast, inexpensive, and easy to use, will not be used more widely until it has been demonstrated to practitioners that these systems provide answers that help solve the problems of patient care." [22]

When developing our interactive visualization methods, we put forward a user-centered design approach. This included a user study, the discussion of the designed methods in a review step, and the evaluation of our Java prototype as described in the upcoming sections. All of these steps were carried out in a qualitative manner in form of guided interviews. The prototype evaluation was done scenario-based using an example protocol.

Figure 5 shows our development process graphically as a set of interconnected tasks around the central entity of design, the user. Points of information exchange with the user in the development are signalized by arrows to and from the user.

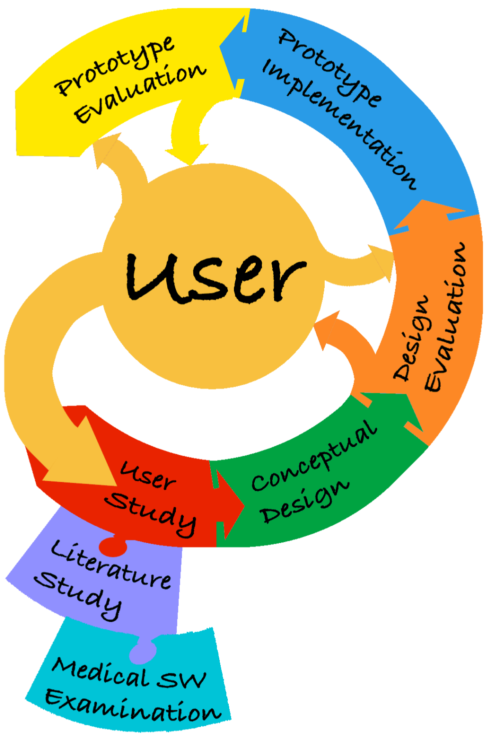


Fig. 5. User Centered Design.

5.1 User Study to Acquire Physicians' Needs

A step of major importance for requirement analysis in our development process was to conduct a user study with eight physicians of the General Hospital of Vienna (AKH Wien) to gain deeper insights into the medical domain, work practices, application of guidelines in daily work, users' needs, expectations, and imaginations.

It became apparent that clinical guidelines are generally depicted by a special form of flow-charts named *clinical algorithm maps* as proposed in [3] and are widely known. GANTT charts were known among most of our interview partners and half of the interviewed physicians knew LifeLines and PERT charts. LifeLines however, were understood much more easily when asking for the possible meaning of an example.

When summarizing and evaluating the results of our user study, the following fundamental characteristics can be recognized – a simple and transparent structure, intuitive interaction (easy to learn and comprehend), a cleaned up interface, a high level of application safety (undo where possible), time saving (allowing quick and effective work), fast, and flexible. (Detailed results and interview guidelines can be found in [9].)

5.2 Design Review

When having completed the first "release" version of the conceptual design, we conducted a review session for getting early feedback regarding our design by two experts (visualization expert and medical expert). This early evaluation process was very valuable and reduced the risk of investing time and effort in unfruitful initiatives.

5.3 Prototype Evaluation

A scenario-based, qualitative prototype evaluation was carried out by conducting interviews with physicians working in intensive care units. Five of the eight physicians who already participated in the user study at the beginning of this work took part in the evaluation. The interviews consisted of the four main parts: Introduction, Prototype Presentation, Prototype Testing, and Feedback/Questionnaire [9].

The feedback regarding our design and prototype, given by the interviewed physicians, was very positive. All of them considered the overall structure clear, simple and not overloaded. The graphical representations and symbols have been judged to be intuitive and clear, keeping the learning effort relatively low. The interviewed doctors considered the two different views very helpful in working with and exploring treatment plans as well as patient data. Difficulties in relating the views to each other were not perceived.

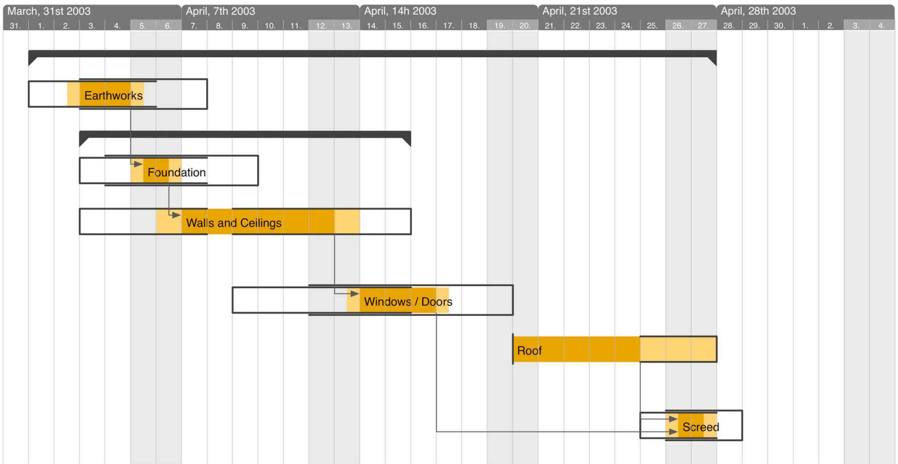
Temporal aspects and particularly temporal uncertainties play a crucial role treatment planning and medical data analysis. This involves supporting the detection of possibly critical situations as well as macro readings in the temporal domain. In order to investigate if our novel PlanningLine glyph supports these characteristics, we conducted a thorough empirical evaluation in form of a comparative study.

6 Empirical Evaluation

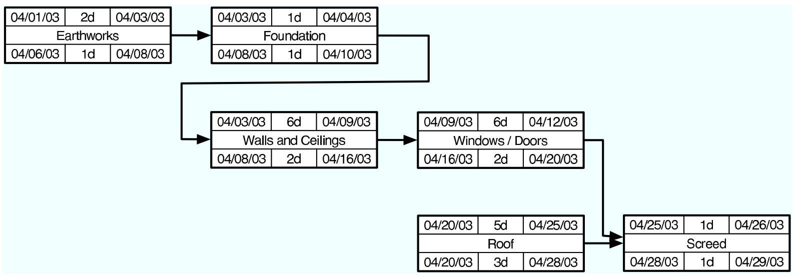
In contrast to the methods applied in our user centered design approach, we focus here on the cognitive level of the used visual representation. Since the temporal characteristics and attributes of treatment plans are very similar to those of project plans in project management, we generalized the concept in order to be able to compare our approach with a well-known technique in the field. Goal of the study was to compare the performance of individuals using PlanningLine or PERT representations depicting temporal attributes and relationships of project tasks (see Fig. 6). The reason for the decision to compare PlanningLines with PERT is based on the capability of PERT to represent temporal uncertainties. The experiment design is paper-based and analogous to related studies evaluating LifeLines [23] and Paint Strips [10]. For brevity you find both experiment hypotheses and results of statistical tests in Section 6.8. The fact that our study was carried out at a cognitive rather than a domain specific level allows for applying its results to the medical domain as well.

6.1 Subjects

The subjects in the study were 48 undergraduate and graduate students of informatics and business informatics in a usability engineering workshop. The subjects exhibit rather heterogeneous knowledge and experience levels, and had no knowledge on the PlanningLine method. Since the subjects in our study have varying degrees of experience with the PERT method, we conducted a tutorial that briefly repeated how to use PERT and introduced the new PlanningLine method to ensure a minimal common level of knowledge for the experiment, namely chart reading and basic calculation skills.



(a) PlanningLines.



(b) PERT.

Fig. 6. Project Plan Example (Construction Works) in PlanningLine and PERT Representation.

6.2 Experiment Objects and Procedures

In the following, we give a short overview of the experiment objects used in the empirical study (refer to [24] for the detailed material). All experiment participants received the following material:

(1) *Background Questionnaire:* At the beginning of the study, a one-page questionnaire acquired the experience with PERT and other representations used in project management.
(2) *Answering Sheets for task solutions in three parts:* Four different versions of these sheets were available, for the combinations of the two treatments (PlanningLine, PERT) and data sets (1, 2).

- *Part A:* This part contained a three-page answering sheet for questions and tasks, concerning the usage of PlanningLines or PERT (see Fig. 7(a) for a sample question).
- *Part B:* Contained an example project plan and a five-page answering sheet for questions on the project plan (see Fig. 7(b) for a sample question).
- *Part C:* A one-page answering sheet for drawing a PlanningLine/PERT chart, based on textual task description (see Fig. 7(c) for a sample question).

(3) *Feedback Questionnaire* on the ease of use and perceived usefulness of both approaches [25].

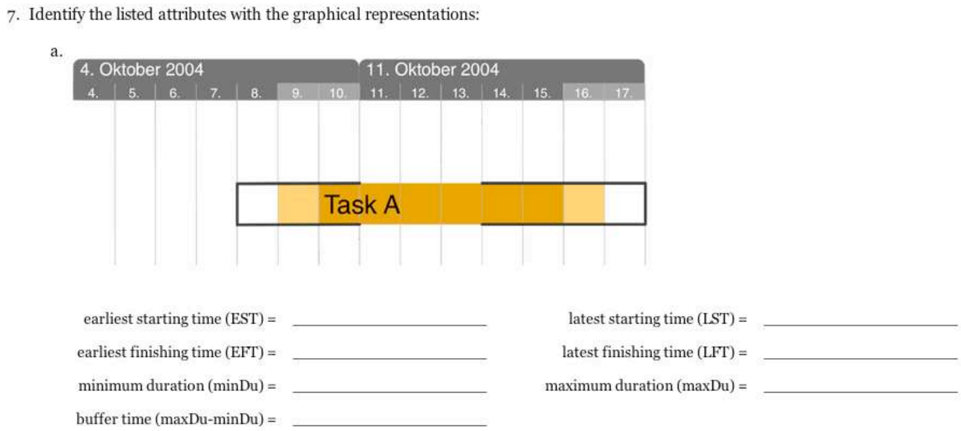
After the tutorial that briefly repeated how to use PERT and introduced PlanningLines, the participants received the experiment material. The participants had 45 minutes to fill in the questionnaires and answering sheets. Subjects were asked to take time stamps at the start and end of each part of the answering sheet. These time stamps allow to measure the time needed to work on the tasks in a part.

6.3 Experiment Design

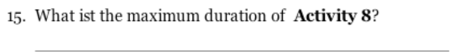
We randomly selected students for the two groups in the study – initial PlanningLine and PERT users. By randomization we forced unknown source of discrepancy to contribute homogeneously to the treatments, following the suggestion presented in [26]. During the experiment each individual independently worked on the experimental material. In addition to the two treatments, PlanningLine and PERT, we used two project data sets to investigate whether the treatments performed similarly with different data sets.

6.4 Threats to Validity

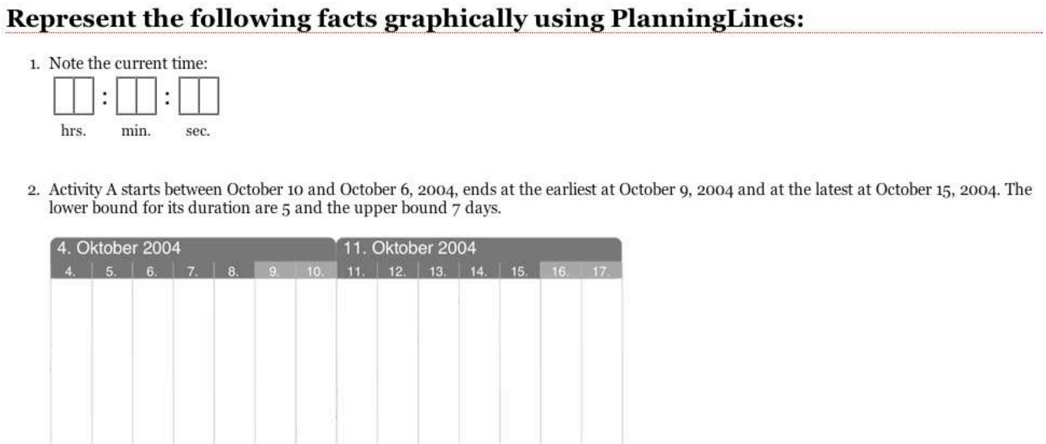
In every empirical study there are possible threats to the validity of the study which need to be acknowledged and mitigated with appropriate countermeasures. With the experiment design we prevented threats to internal validity: history, maturation, selection, and process conformance [27]. Regarding external validity we took a control method (PERT) that is widely used in practice and we investigated mainly cognitive abilities of subjects rather than their project management abilities. Thus using students for the study is not a problem.



(a) Part A Sample Question.



(b) Part B Sample Question.



(c) Part C Sample Question.

Fig. 7. Sample Questions of Study Tasks (PlanningLines) (translated from German).

6.5 Experiment Variables

The goal of the empirical study is to infer causality or to analyze relationships between variables. The dependent variables measure the effect of manipulating the independent variables [28] – subject performance on number of mistakes when answering a standard questionnaire and duration for answering these questions. Independent variables are defined as factors believed to influence the results of the experiment [8], in our case the treatments (PlanningLine and PERT) and the project data sets.

6.6 Data Analysis Approach

In this paper we use the following notation to describe a combination of treatment (PlanningLine, PERT) and data sets (1, 2): PlanningLine1 means using PlanningLines with data set 1. We used statistical differences to determine significant differences in the dependent variables' means caused by the independent variables. In most cases the parametric t-test or its non-parametric counterpart, the Mann-Whitney test can be used to compare two sample means [27]. The statistical tests were performed with an α-level of 0.05.

6.7 Test on Similarity of Data Sets

First we tested the similarity of performance with the two project data sets. We have four groups of combinations of treatments and data sets (PlanningLine1, PlanningLine2, PERT1, and PERT2). We evaluated the performance (relative number of mistakes and duration) of groups that use the same data set but different representation techniques (PlanningLine1-PlanningLine2, PERT1-PERT2). There is no significant difference in the performance of the project data sets regarding mistakes (p=0.501 for PERT and p=0.431 for PlanningLines) and duration (p=0.601 for PERT and p=0.401 for PlanningLines). After establishing that the users of different data sets perform in a similar way, we can compare the performance of different representations regardless of the data set used (PlanningLine1+2 - PERT1+2). The hypotheses correspond to testing the user groups and data set combinations regarding data from different parts and sections of the experiment material.

6.8 Hypotheses and Test Results

As t-test and Mann-Whitney test consistently showed similar results, we report the p values from the t-test. The hypotheses correspond to testing the above user groups and data set combinations (PlanningLine1+2 vs. PERT1+2) regarding results from the different parts and sections of the experiment material. In the following, we state the hypothesis and the results.

(1) *The PlanningLine representation is as simple and intuitive to use as the PERT representation.*
Regarding the performance of both mistakes (p=0.468) and time (p=0.323) there is no significant difference between PlanningLine and PERT users.

(2) *The classical PERT chart is more appropriate for answering detailed questions on single attributes of a project plan than PlanningLines.*
While PERT users make significantly fewer mistakes than PlanningLine

users (p=0.016), the task duration of both technologies is not significantly different (p=0.087).

(3) *The PlanningLine representation is better suited to deal with temporal uncertainties regarding the duration, start, or end of activities or plans.* PlanningLine users do not make significantly fewer mistakes than PERT users (p=0.086), but the task duration of PlanningLine users is significantly shorter (p=0. 012).

(4) *Possible critical sections in a project plan can be spotted easier and more correctly using PlanningLines as with PERT charts.* PlanningLine users make significantly fewer mistakes than PERT users (p=0.089).

(5) *The layout and meaning of individual parts of the PlanningLine glyph are recalled easier as of the PERT representation.* PlanningLine users make significantly more mistakes (p=0.000) and take longer than PERT users (p=0.000).

(6) *PlanningLines are perceived subjectively positive.* PlanningLines users are more content using their method than PERT users (p=0.005).

6.9 Discussion

Overall, the experiment results confirm our assumptions regarding PlanningLines. The fundamental assumption that PlanningLines are generally not harder to use for typical project management tasks as PERT charts was supported by the study results. Furthermore, PlanningLine users are faster in answering questions on temporal uncertainties which clearly reflects our intentions. Only the fifth hypothesis on the recall of the representation was not supported by the study results which might be caused by the fact that most subjects used PERT before. As predicted, PlanningLines are harder to use for reading exact attributes but are best for overall analysis of temporal uncertainties and can be augmented with dynamic display of explicit detail data as needed. Besides the quantifiable results, users subjectively judged PlanningLines positively.

7 Conclusions

Analyzing time-oriented medical data is a very challenging task, because various dimensions need to be taken into account to explore the data in-depth. Data visualization techniques provide different visualization and interaction techniques to explore such data, but mostly neglect the context the data was acquired. We tried to overcome that limitation and provided in interactive visualization - called *CareVis* - which combines the pure time-oriented data with the medical treatments steps to ease the exploration process in a multi-view approach.

During the design and implementation of CareVis we concentrated on the user's demands and needs and performed a user-oriented design. Therefore, we extended and enhanced visualization techniques the medical staff is used to work with and which are easy to comprehend, namely clinical algorithm maps and LifeLines. During the last evaluation phase, we discovered that temporal uncertainty and planning techniques are an essential component in the clinical daily routine. Therefore, we conducted a comparative study to assess the power of our PlanningLines representation which we designed to represent and visualize temporal uncertainty. This empirical study demonstrated the usefulness of our approach on a cognitive level.

After showing the usability of one part of our approach, we need to illustrate that CareVis really meets the needs and demands of the medical staff in assisting them in analyzing context-specific and time-oriented data. We have already performed an artificial scenario-based evaluation (compare [9]) where we demonstrated how different analysis tasks can be accomplished (like, a physician who wants to analyze different parts of the treatment along with measured patient data of a treatment protocol she just completed). The next step will be that the medical staff will evaluate, if CareVis meets their needs.

Acknowledgements

Many thanks to Peter Messner, for reviewing the experiment design, to Gernot Goluch, Christoph Gesperger, Manuel Ganglberger, and Christoph Fleury, for their support in preparing the experiment material, and to the experiment participants. This project is supported by “Fonds zur Förderung der wissenschaftlichen Forschung - FWF” (Austrian Science Fund), grant P15467-INF.

References

[1] M. Field, K. Lohr, Guidelines for Clinical Practice: From Development to Use, Institute of Medicine, Washington, D.C. National Academy Press, 1992.

[2] D. C. Hadorn, Use of Algorithms in Clinical Practice Guideline Development: Methodology Perspectives, AHCPR Pub. 0009 (95) (1995) 93–104.

[3] Society for Medical Decision Making, Proposal for Clinical Algorithm Standards, Medical Decision Making 12 (02) (1992) 149–154.

[4] E. Tufte, S. M. Powsner, Graphical Summary of Patient Status, The Lancet 344 (8919) (1994) 386–389.

[5] C. A. Brandt, S. J. Frawley, S. M. Powsner, R. N. Shiffman, P. L. Miller, Visualizing the Logic of a Clinical Guideline: A Case Study in Childhood Immunization, Methods of Information in Medicine 36 (1997) 179–83.

[6] C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, B. Shneiderman, LifeLines: Using Visualization to Enhance Navigation and Analysis of Patient Records, in: Proceedings of the 1998 American Medical Informatic Association Annual Fall Symposium, 1998, pp. 76–80.

[7] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R. Greenes, R. Hall, P. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E. Shortliffe, Stefanelli, Comparing Computer-Interpretable Guideline Models: A Case-Study Approach, The Journal of the American Medical Informatics Association (JAMIA) 10 (1) (2003) 52–68.

[8] www.openclinical.org, Open Clinical - Knowledge Management for Medical Care, http://www.openclinical.org (2003).

[9] W. Aigner, Interactive Visualization of Time-Oriented Treatment Plans and Patient Data, Master's thesis, Vienna University of Technology, Institute of Software Technology and Interactive Systems, Vienna, Austria (May 2003).

[10] L. Chittaro, C. Combi, Visualizing Queries on Databases of Temporal Histories: New Metaphors and their Evaluation, Data and Knowledge Engineering 44 (2) (2003) 239–264.

[11] C. Combi, L. Portoni, F. Pinciroli, Visualizing temporal clinical data on the www, in: W. Horn, Y. Shahar, G. Lindberg, S. Andreassen, J. Wyatt (Eds.), Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making (AIMDM'99), Springer, 1999, pp. 301–311.

[12] S. Miksch, R. Kosara, Communicating Time-Oriented, Skeletal Plans to Domain Experts Lucidly, in: T. Bench-Capon, G. Soda, T. A.M. (Eds.), Database and Expert Systems Applications, Proceedings of the10th International Conference of Database and Expert Systems Applications (DEXA'99), Springer, Berlin, 1999, pp. 1041–1051.

[13] Y. Shahar, S. Miksch, P. Johnson, The asgaard project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines, Artificial Intelligence in Medicine 14 (1998) 29–51.

[14] J. C. Roberts, On Encouraging Multiple Views for Visualization, in: E. Banissi, F. Khosrowshahi, M. Sarfraz (Eds.), IV'98 – Proceedings International Conference on Information Visualization, IEEE Computer Society, 1998, pp. 8–14.

[15] J. C. Roberts, Multiple-View and Multiform Visualization, in: R. Erbacher, A. Pang, C. Wittenbrink, J. Roberts (Eds.), Visual Data Exploration and Analysis VII, Proceedings of SPIE, Vol. 3960, IST and SPIE, 2000, pp. 176–185.

[16] Q. Zeng, J. J. Cimino, Providing Multiple Views to Meet Physician Information Needs, in: Proceedings of the 33rd Hawaii International Conference on System Sciences, IEEE, 2000.

[17] W. Aigner, S. Miksch, Communicating the Logic of a Treatment Plan Formulated in Asbru to Domain Experts, in: K. Kaiser, S. Miksch, S. Tu (Eds.), Computer-based Support for Clinical Guidelines and Protocols. Proceedings of the Symposium on Computerized Guidelines and Protocols (CGP 2004), IOS Press, 2004, pp. 1–15.

[18] R. Bade, S. Schlechtweg, S. Miksch, Connecting Time-Oriented Data and Information to a Coherent Interactive Visualization, in: Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2004, ACM Press, 2004, pp. 105–112.

[19] K. Beck, Extreme Programming Explained, Addison-Wesley, 2000.

[20] G. Alder, Design and Implementation of the JGraph Swing Component, Tech. Rep. 1.0.6 (February 2002).

[21] E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns - Elements of Reusable Object-Oriented Software, Addison-Wesley, Reading, MA, 1994.

[22] P. Gorman, M. Helfand, Information seeking in primary care: How physicians choose which clinical questions to persue and which to leave unanswered, Med Decis Making 15 (1995) 113–9.

[23] D. L. Alonso, A. Rose, C. Plaisant, K. L. Norman, Viewing Personal History Records: A Comparison of Tabular Format and Graphical Presentation using LifeLines, Behavior and Information Technology 17 (5) (1998) 249–262.

[24] W. Aigner, S. Miksch, B. Thurnher, S. Biffl, PlanningLines Usability Studie - User Study zum Vergleich von PlanningLines und PERT Darstellung (in German), Technical Report Asgaard-TR-2005-3, Vienna University of Technology, Institute of Software Technology and Interactive Systems, Vienna, Austria, Europe (2005).

[25] F. D. Davis, Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology, MIS Quarterly 13 (3) (1989) 319–340.

[26] G. E. P. Box, W. G. Hunter, J. S. Hunter, W. G. Hunter, Statistics for Experimenters, John Wiley & Sons, 1978.

[27] C. Wohlin, P. Runeson, M. Hst, M. C. Ohlsson, B. Regnell, A. Wessln, Experimentation in Software Engineering – An Introduction, The Kluwer International Series in Software Engineering, Kluwer Academic Publishers, 2000.

[28] C. M. Lott, H. D. Rombach, Repeatable Software Engineering Experiments for Comparing Defect-Detection Techniques, Empirical Software Engineering 1 (3) (1996) 241–277.