

CareVis: Integrated Visualization of Computerized Protocols and Temporal Patient Data

Wolfgang Aigner * & Silvia Miksch

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

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

Abstract

Objective: 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 to find reasons and explanations for certain phenomena in the measured patient data, but is mostly spared out in the analysis process. This work aims to fill this gap via integrating classical data visualization and visualization of treatment information.

Methods and Material: We considered temporal as well as logical data aspects and applied a user-centered development approach that was guided by user input gathered via a user study, design reviews, and prototype evaluations. Furthermore, we investigated the novel *PlanningLine glyph*, that is used to represent plans in the temporal domain, via a comparative empirical user study.

Results: Our interactive visualization approach *CareVis* provides multiple simultaneous views to cover different aspects of the complex underlying data structure of treatment plans and patient data. The tightly coupled views use visualization methods well-known to domain experts and are designed to facilitate users' tasks. The views are based on the concepts of clinical algorithm maps and LifeLines which have been extended in order to cope with the powerful and expressive plan representation language *Asbru*. Initial feedback of physicians was encouraging and is accompanied by empirical evidence which verifies that PlanningLines are well suited to manage temporal uncertainty.

Conclusion: The interactive integration of different visualization methods forms a novel way of combining, relating, and analyzing different kinds of medical data and information that otherwise would be separated.

Key words: protocol-based care, information visualization, clinical guidelines, treatment plans, patient data, user-centered design, temporal uncertainty

1 Introduction

Visualization is an integral part of intelligent data analysis. On the one hand, visualization is applied for presenting results and on the other hand, human perception might be utilized for driving the analysis process. In the medical domain, mostly patient data measurements are used as basis for analysis either in form of high-frequency data of intensive care settings or low-frequency data, e.g., for long term studies. Current visualization methods are mostly bound to the representation of such measured patient data only, which can be subsumed by 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, data on which therapeutic steps have been taken at what 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. 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 the treatment process along protocols semi-automatically by the use of information technology. The core entity, medical treatment plans, are complex documents, 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. Moreover, a combination with the presentation of patient data when treating a patient along a plan for monitoring and analytic tasks has been considered even less. However, the

* Corresponding author. Tel: +43 1 58801 18833; Fax: +43 1 58801 18899
Email address: aigner@ifs.tuwien.ac.at (Wolfgang Aigner).
URL: <http://ieg.ifs.tuwien.ac.at> (Wolfgang Aigner).

integrated visualization of patient data and medical treatment plans could be of great assistance to ease the complex task of analyzing medical data and protocols.

In the upcoming section, we will provide a task and data analysis of the problem domain followed by a discussion of related work in Section 3. After this, we will describe the design of our visualization approach CareVis in Section 4. Subsequently, we discuss our user-centered design methodology including prototype implementation, evaluation, and user study results in Section 5. Finally, we describe how users' tasks are supported and sum up our findings.

2 Problem Analysis

In this section, we analyze the problem domain from a data as well as user's perspective. At the beginning, we will shortly explain the plan representation language Asbru that our project is based upon. Following this, we illustrate use-cases in form of scenarios and identify major task classes. Finally, we will combine the constraints and requirements to a summary of data characteristics that need to be dealt with.

2.1 The Plan Representation Language Asbru

Asbru is a time-oriented, intention-based, skeletal plan-specification representation language that is used in the *Asgaard* Project¹ to represent clinical guidelines and protocols in eXtensible Markup Language (XML). Asbru can be used to express clinical protocols as skeletal plans that can be instantiated for every patient. It was designed specifically for the set of plan-management tasks [2]. The major features of Asbru are that

- prescribed actions and states can be continuous;
- intentions, conditions, and world states are temporal patterns;
- uncertainty in both temporal scopes and parameters can be flexibly expressed by bounding intervals;
- plans might be executed in sequence, all or some plans in parallel, all or some plans in a particular order or unordered, or periodically; and
- particular conditions are defined to monitor the plan's execution.

¹ In Norse mythology, Asgaard was the home of the gods. It was located in the heavens and was accessible only over the rainbow bridge, called Asbru (or Bifrost). For more information about the Asgaard project see <http://www.asgaard.tuwien.ac.at> (Accessed: 26 March 2006).

Basically, an Asbru plan can be seen as a template. This template gets instantiated whenever the plan gets executed. Additionally, more than one instance might be created for a single plan. This pattern can be seen as an analogy to the class-instance relationship in Object-Oriented Programming.

Since a plan is represented in XML, it is basically readable to humans. But understanding a plan in such a representation needs a lot of training as well as semantic and syntactic knowledge about the representation language. It is cumbersome and surely not suited for physicians. Therefore, the formal representation needs to be translated into a form familiar to domain experts to be able to communicate the logic of a computerized treatment plan.

2.2 User Tasks and Scenarios

To illustrate the different tasks of medical personnel, we created three typical use-scenarios of physicians in protocol-based care.

2.2.1 Scenario 1

Markus Zolte, assistant doctor in training in internal medicine, will be working in the pediatrics department for the next few months and is exploring various treatment methods for new born infants. He informs himself about hyperbilirubinemia by walking through the related treatment protocol. He is interested in the logical workflow and explores the treatment plan. After the first walkthrough of the hyperbilirubinemia protocol, Markus Zolte goes back to the intensive photo-therapy part and wants to know in which cases this plan is stopped. He is also interested in which part of the complete treatment plan he is viewing right now. Furthermore, he wants to see all other parameters and variables that are getting used in this treatment plan.

2.2.2 Scenario 2

Andrea Habacher, assistant medical director of internal medicine, just completed the treatment of a patient using the “Controlled Ventilation” plan. Now, she wants to analyze different parts of the treatment along with measured patient data. She starts by examining how long different phases of the plan took in relation to others. The “Handle PCO₂” plan is of particular interest to her. She also wants to see the transcutaneous partial pressure of carbon dioxide (PtcCO₂) for examining relations between plan execution and PtcCO₂ values. Because there is a significant discontinuity of the PtcCO₂ value within this plan, she recalls the sub-steps taken in the “Handle PCO₂” plan. Furthermore, she wants to see when the particular steps were conducted.

After that, she is interested in if and how the $PtcCO_2$ values influenced the “patient-state” parameter.

2.2.3 Scenario 3

Heinrich Kovanic, assistant medical doctor in an intensive care unit (ICU), is currently treating a patient who suffers from hyperbilirubinemia. He examines the total serum bilirubin (TSB) and TSB-change values and wants to review the patient record for getting basic patient information. After that, he investigates all incoming parameters and encounters a rapid increase of the TSB value that happened two hours ago. He wants to find out which plan or action took place at that time. Furthermore, he examines the parameter constraints defined by the plan conditions. After encountering the reason for the value change, he wants to go back to the current position of plan execution.

2.2.4 Tasks

Summarizing the essentials of these scenarios, three fundamental user tasks can be identified.

- becoming acquainted with a specific treatment method and observed patient’s parameters;
- guidance in the treatment process (run-time support while treating a patient via monitoring patient status, presenting upcoming treatment steps, and providing a treatment history); and
- analyzing the treatment process (observed data together with treatments).

2.3 Data Characteristics

The underlying data for the tasks identified above can be broken down into three categories.

- treatment plan specification data (Asbru plans);
- treatment plan execution data (instantiation and execution of a plan, output of protocol execution unit); and
- patient data (time oriented, e.g., blood oxygen measurements).

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

- time-oriented data (execution data and planning data including a rich set of time attributes to represent temporal uncertainties);

- logical sequences;
- hierarchical decomposition;
- flexible execution order (sequential, parallel, unordered, any-order);
- non-uniform element types; and
- state characteristics of conditions.

3 Related Work

In the following, we discuss related work in the areas of medical treatment planning, Information Visualization, and commercial medical software.

3.1 Medical Treatment Planning

3.1.1 Clinical Algorithm Maps

The most widely used visual representation of clinical guidelines are so-called *flow-chart algorithms*, also known as *clinical algorithm maps* [3]. A standard for this kind of representation has been proposed by the *Committee on Standardization of Clinical Algorithms* of the *Society for Medical Decision Making*:

“However, since algorithmic logic is wired implicitly into a protocol, it is difficult to learn an algorithm from a protocol. By contrast, flow-chart algorithms, or clinical algorithm maps, are uniquely suited for explicitly communicating conditional logic and have therefore become the main format for representing a clinical algorithm clearly and succinctly.” [4]. The proposed standard includes a small number of different symbols and some rules on how to use them. One additional feature to standard flow-charts are annotations that include further details, e.g., 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 (before, after). Furthermore, flow-charts cannot be used to represent concurrent tasks or the complex conditions as used in Asbru due to their state-like semantics. 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.

3.1.2 Further Scientific Projects Related to Protocol-Based Care

Other scientific work [5–7] 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 [8,9] and [10].)

Only some of the available projects dealing with protocol-based care provide graphical representations. The listed ones include such graphical representations, but most of them focus on authoring of plans only. They use flowchart- or workflow-like presentations that depict the elements used in their formal representation. An Asbru-specific tool for authoring and manipulating plan specifications is *AsbruView* [11]. It utilizes metaphors of running tracks and traffic control to communicate important concepts and utilizes glyphs to depict the complex time annotations used in Asbru.

These tools make authoring clinical protocols easier, especially for non-computer-scientists, but authoring clinical guidelines, on the one hand, and communicating patient data and complete protocols to domain experts, on the other hand, are two rather different tasks with different goals. Additionally, the presented techniques use graphical representations which are not familiar to domain experts and mix state and flow-chart characteristics within a single diagram. Understanding such representations and using them for plan authoring requires a considerable amount of learning effort. A more detailed discussion of the quoted projects can be found in [12].

In contrast to authoring plans, our goal is the intuitive communication of logical in conjunction with temporal aspects of a treatment plan and patient state parameters. Whereas the presentation of and navigation within protocols are paramount along with offering easy access to linked information and in-depth explanations.

3.2 Information Visualization Methods

Other possibilities to visualize logical sequences besides flow-charts are *Structograms (Nassi-Shneiderman Diagrams)*, *PERT charts*, *Petri nets*, and *State Transition Diagrams*. These techniques focus on other purposes and some of them are more powerful and expressive than flow-charts. But none of them offers a notion for a combined depiction of hierarchical decomposition, flexible execution order, and the state characteristic of conditions in their basic forms as needed for representing Asbru plans.

Time is an important data characteristic but methods for visualizing time

other than in time-series plots are not well known. The probably best known method for representing intervals are *Gantt charts* and their utilized *Time Lines*. An extension of Time Lines are *LifeLines* [7] which utilize horizontal bars to represent the temporal location and duration of data elements. They were applied to represent personal histories and patient records. In order to organize 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 such as the ability to depict hierarchical data are missing. The visualization is mainly used retrospectively for analytic and presentation tasks rather than for planning. Moreover, LifeLines cannot represent temporal indeterminacies. Because of these limitations, LifeLines are not suited to represent indeterminate planning data as needed for Asbru plan specifications. Other visualization techniques like *Paint Strips* [13], *Temporal Objects* [14], the *Time Annotation Glyph* [11], and *SOPO View* [15] were especially developed to represent indeterminate temporal data.

Paint Strips [13] have been developed for visualizing queries on medical databases. The idea of Timelines is enriched by a painting metaphor which indicates that the displayed bars are drawn by a paint roller. A paint roller at the beginning or 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 are 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. Due to the simplicity of the Paint Strips metaphor, some time annotation attributes such as durations independent of the differences between start and end points, undefined values, a reference point, or different granularities cannot be visualized. Especially independent durations and undefined values are very common in Asbru plan specifications and therefore need to be supported.

Temporal Objects [14] were developed for depicting temporal data specified at 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 filled rectangles at each end representing the start and end intervals. The visual appearance of a Temporal Object emphasizes the extent of the glyph as a whole because the glyph consists of multiple connected and solidly filled rectangles (also see Gestalt principles

[16]). Particularly when primarily representing indeterminacy or uncertainty in planning, it is of importance to emphasize the dynamics and degrees of freedom of intervals.

The interactive 2D technique *SOPO View* [15] utilizes Rit’s sets of possible occurrences (SOPOs) [17]. The axes of the diagram are used to depict the start interval (x-axis) and end interval (y-axis). Minimum and maximum duration are the constraining borders parallel to the 45° time flow axis. The area that a SOPO covers contains all intervals that fit the specification given by means of earliest start, latest start, earliest end, latest end, minimum, and maximum duration. Hence, any point in this diagram represents a complete interval, specified by its start (x-coordinate) and end time (y-coordinate). SOPOs were designed for an easy graphical propagation of temporal constraints, but not for making a complex notion of time easy to understand. Specifically, parallel plans and hierarchical decomposition are very hard to depict and work with. Moreover, a notion for undefined parts is missing in the original design.

A visualization technique representing the same temporal attributes as SOPOs, is the *Time Annotation Glyph* [11]. But in contrast to the geometric SOPO technique, the Time Annotation Glyph is timeline- and glyph-based. Specifically, it uses the simple metaphor of bars that lie on “pillars”. Four vertical lines on the base specify earliest and latest starting and ending times. Supported by these pillars lies a bar that is as long as the maximum duration. On top of the maximum duration bar, a bar that represents the minimum duration lies upon two diamonds for latest start and earliest end. Furthermore, undefined parts are displayed in gray and different granularities are indicated by using zigzag lines. Because of this metaphor, a few simple time attribute constraints can be understood intuitively. For example that 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 might 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 relatively few visual cues in order to help maintaining time-attribute constraints. Together with SOPO View, the Time Annotation Glyph is applied to represent the time annotations of medical treatment plans within the AsbruView application [11].

The main focus of our work is to communicate and support the analysis of patient data and medical treatment plans. Formal considerations of temporal data modeling or formal timeline manipulations are out of scope of this article. For further information, see for example *HMAP* [18], which is a temporal data model for managing intervals with different granularities and indeterminacies. A formal system for timeline elements and timeline operations has been developed by Cousins and Kahn [19] which also includes a *Time Line Browser*

for displaying intervals or events of multiple timelines and their manipulation.

3.3 Visualization in Commercial Medical Software

A very high portion of the offered commercial software products in medicine deal with administrative issues such as patient data management or billing. Only very few include any visualization parts and even less offer functionality to aid treatment planning.

We examined a number of non-administrative software products that use graphical representations in general (not only focused on protocol-based care), for the reason of compiling a set of graphical representations most commonly used and familiar to most physicians [12]. Investigated software products include IntelliVue / CareVue [20], Chart+ for Critical Care [21], Visual Care [22], and QCare [23]. All of the examined products are rather data-centric and the most popular form of data representation is using tables where numerical and textual data is organized in spreadsheets. None of the investigated products offered a way of visualizing the logic of a treatment plan.

4 Design of the Integrated Visualization CareVis

The underlying data structure that we want to communicate to medical domain experts is very complex. Since none of the existing visualization methods can be used to represent all needed data characteristics, we decided to use the approach of *multiple views* [24]. Multiple views are a well known Information Visualization method, whereby a number of representations that focus on different aspects of the data are provided for a common underlying data structure [25].

4.1 Views

Basically, we divided the underlying data structure along the lines of logical structure and temporal aspects. Hence, 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 distinct position. Below that, the screen is divided vertically by the logical view on the left and the temporal view on the right 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). Table 1 summarizes which data characteristics are visualized by the different views.

	Logical View	Temporal View	QuickView Panel
Asbru plans	•	•	
Time-oriented data		•	
Logical sequences	•		
Hierarchical decomposition	•	•	
Non-uniform element types	•	◦	
Conditions	•		
Parameters and variables		•	•

entirely represented (•), partly/implicitly represented (◦), or not represented (empty).

Table 1
Data characteristics in views.

4.1.1 Logical View

The logical view on the left-hand side of the screen provides a representation of the treatment plan specification data [26]. The applied visualization technique *AsbruFlow* is based on the idea of flow-chart-like clinical algorithm maps [3] 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.

An Asbru plan has a plan-body containing *single-steps* that are executed in one of the execution orders *sequentially*, *parallel*, *any-ordered*, or *unordered*. A single-step is either a *variable assignment*, a *if-then-else* construct, an *ask* element, or a *plan activation*. Furthermore, an Asbru plan may contain three conditions – *filter precondition*, *abort condition*, and *complete condition*. These conditions are not just evaluated at a specific time-step during plan execution, but have state characteristic and are monitored throughout the execution of the associated plan.

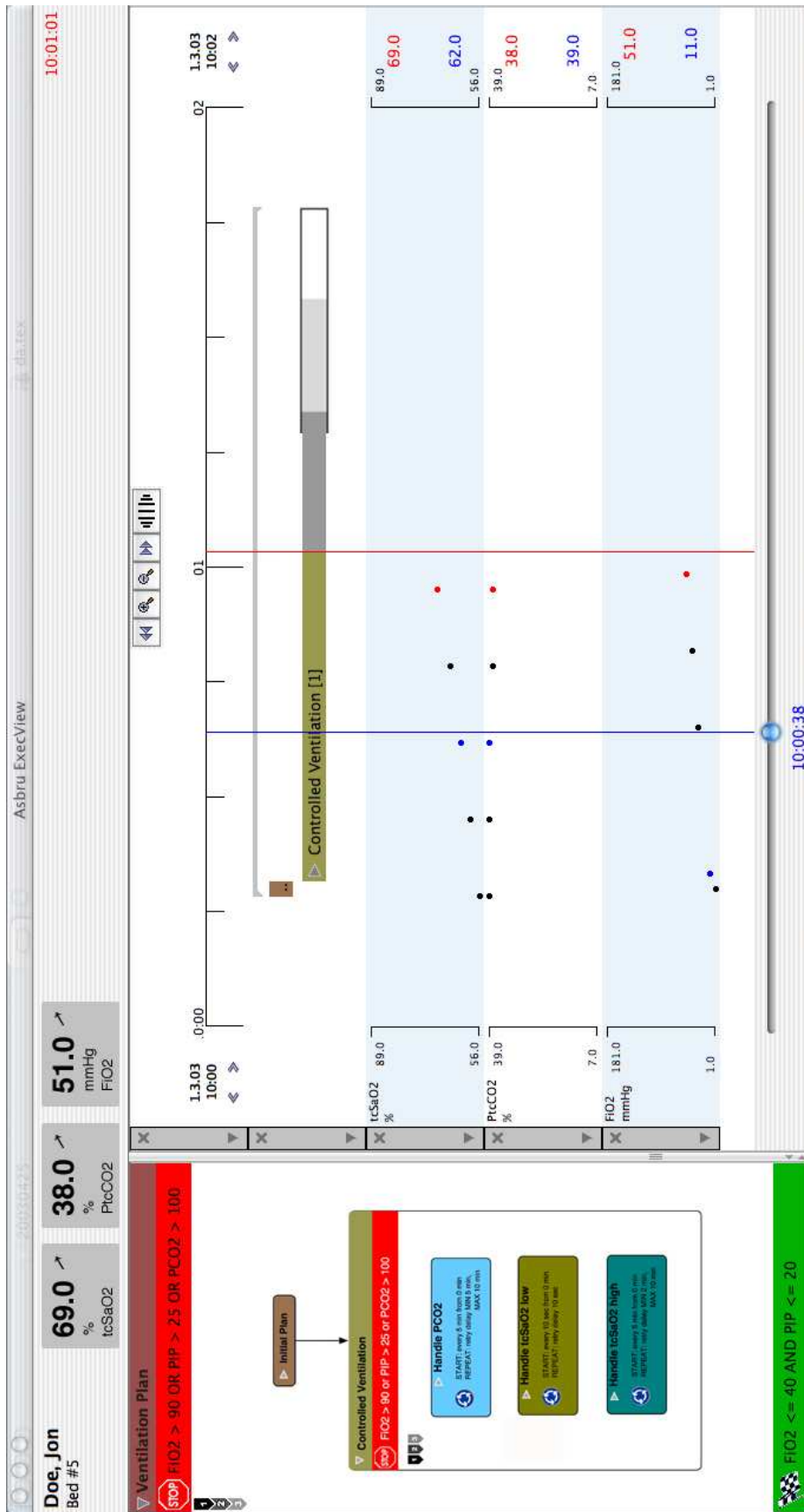


Fig. 1. Application window showing the execution of a plan.

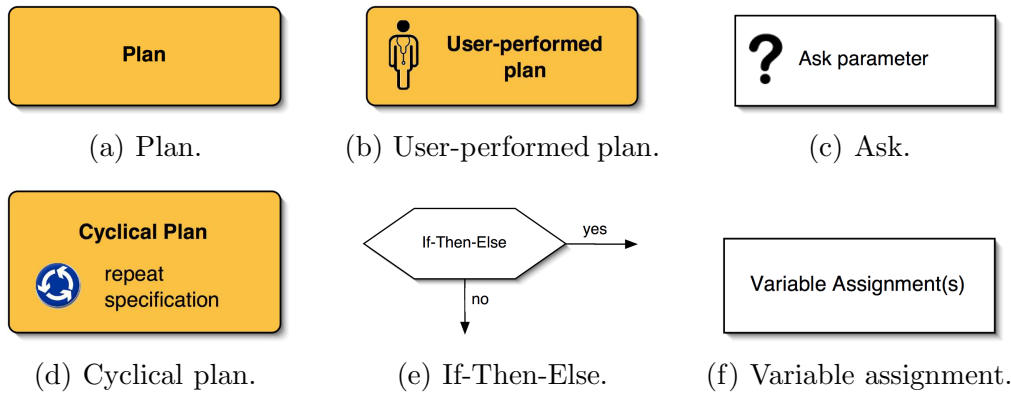


Fig. 2. Plan step elements.

A set of six visual elements is used to depict the single steps within the body of an Asbru plan.

- *Plans (plan activations)* are represented by a rounded rectangle filled with the plan color² (see Fig. 2(a)). In case of being a *cyclical plan*, an additional roundabout icon as well as the repeat specification in textual form are presented within the rectangle (see Fig. 2(d)). Furthermore, a physician icon appears within the element if the plan is *user performed* (see Fig. 2(b)).
- *Ask* steps of a plan are represented by a rectangle including a question mark (“?”) symbol and the text “Ask” followed by the parameter to be entered into the system (see Fig. 2(c)).
- *If-Then-Else* constructs are shown as hexagons having the condition displayed textually (see Fig. 2(e)).
- *Variable assignments* are represented by a rectangle containing the assignment textually (see Fig. 2(f)).

For depicting plan conditions and the execution order of the plan steps, an enclosing frame was created (see Fig. 3). The topmost bar is filled with the plan color and contains the title of the plan. Below the plan title, the abort condition is shown. It is represented by a red bar having a stop sign icon at the left side. Right besides this icon, the abort condition is printed textually. The green bar at the bottom of the plan represents the complete condition. It has a checked finish flag icon at its left and contains the complete condition textually. 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

² A distinct color is assigned to each plan, which makes it easier to distinguish plans from other elements and helps to recognize them in other parts of the representation.

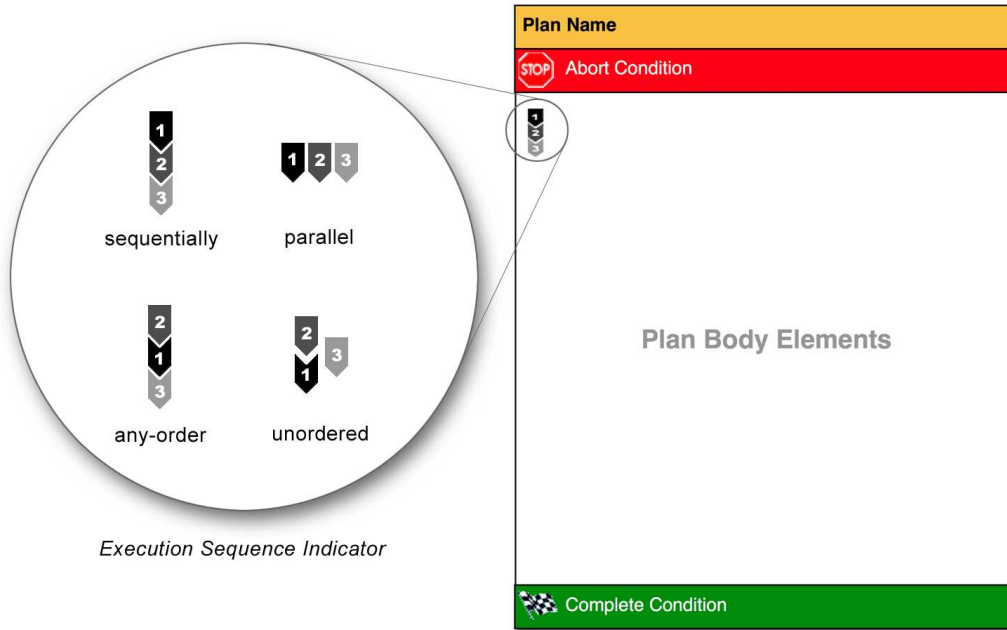


Fig. 3. Basic structure and execution sequence symbols.

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 browsers as for example the Finder of the MacintoshTM operating system.

In order to prevent getting lost within a plan by navigation, two *Focus+Context* techniques are applied. Firstly, there is the *Overview+Detail* mode that uses a small window containing a downscaled, simplified tree overview where the current position within the plan is highlighted. This small overview window can be toggled on or off (see Fig. 4, left). The second technique is a *Fisheye view* which distorts elements that are out of the current focus geometrically by shrinking and moving (see Fig. 4, right) based on the work of Schaffer et al. on hierarchically clustered networks [27].

4.1.2 Temporal View

The temporal view focuses on the time-oriented aspects of Asbru plans as well as the representation of parameters and variables over time. This includes temporal attributes that are defined at design-time and at run-time. Design-time attributes are either defined implicitly by execution order or explicitly via Asbru's time annotations. Run-time attributes include plan execution times (plan start, plan end, etc.), plan variables over time, as well as measured patient data (parameters). The time interval covered by this view can span from a point in time in the past to a point in time in the future, whereas only

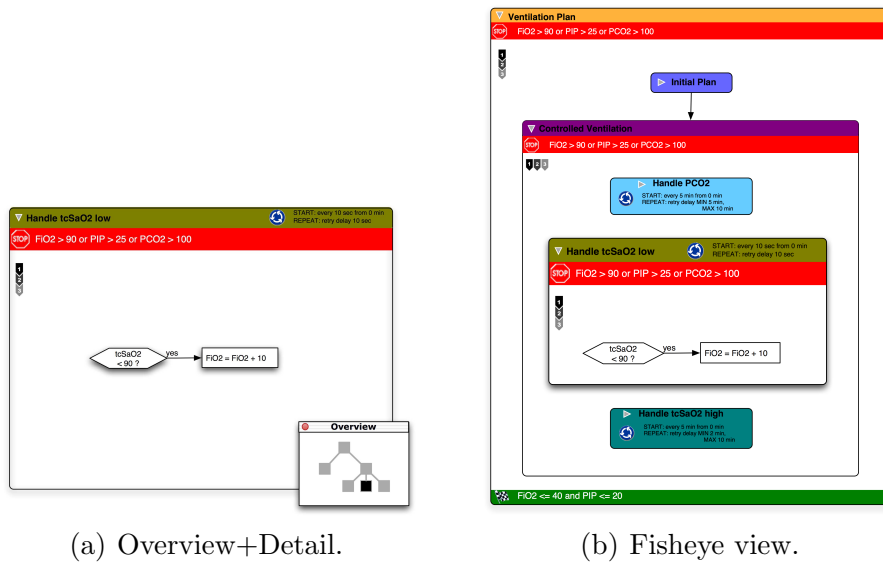


Fig. 4. Logical view showing parts of the Asbru plan for artificial ventilation of newborn infants – Overview+Detail mode (left) vs. Fisheye view mode (right).

plans having time annotations (temporal planning attributes) can be displayed accurately in the future. Parameters and variables can only be displayed for past and present because they are only known and valid, starting right at the point they appear.

The temporal view is divided into collapsible facets which can be added and removed dynamically (see Fig. 1). The time scale on top determines the portion of time being displayed. Below that, one facet that contains the temporal aspects of the treatment plan elements is displayed followed by several facets that contain different plan parameters and variables measured or computed over time. Collapsing facets leads to vertically shrunk and semantically zoomed representations which can be considered as Focus+Context technique (see Fig. 5(a) and Fig. 5(b)).

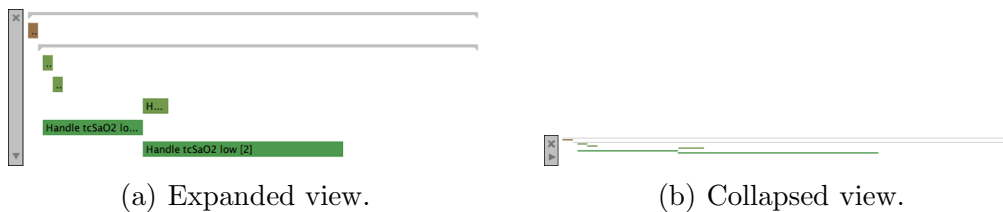


Fig. 5. Asbru plan in expanded and collapsed view.

Time Scale The visible portion of time is determined by the time scale. It provides several interactive features for its manipulation. The viewed interval can be shifted forward or backward in time, zoomed in and out with automatic adjustment of the displayed ticks and labels, and begin and end point might

also be manipulated independently. Moreover, a valuable feature offered by the time scale is the *Fisheye display*. It magnifies a part of the scale interval (focus) while at the same time demagnifying the areas to the left and right of the focus (context) (see Fig. 6). This way, an area of interest is emphasized for detailed examination without hiding information before and after that area, thus preserving the “full picture”. The Focus+Context technique applied here uses a non-continuous transformation function based on the *Bifocal Lens* [28].

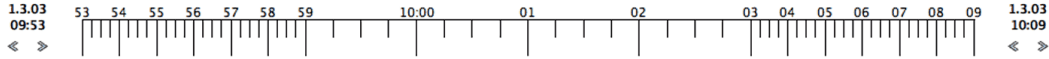


Fig. 6. Fisheye time scale.

Visualization of Plans 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* [29]. LifeLines+ allow the interactive representation of hierarchically structured temporal intervals and simple element characteristics. On top of that, PlanningLines allow the depiction of temporal uncertainties via a glyph consisting of two encapsulated bars, representing minimum and maximum duration, that are bounded by two caps that represent the start and end intervals (see Fig. 7).

A LifeLine+ has a defined beginning and a defined end represented visually by a bar connecting those two points in time. The bar includes the title of the depicted incident and a number of optional elements. Exceed indicators (small arrows) are displayed in case the LifeLine+ exceeds the display interval as an indication that the currently visible line is only a part of the complete element. Furthermore, property symbols in form of small icons might be added on top of the line to indicate simple properties of the depicted incident (e.g., that the displayed plan is cyclical). To indicate hierarchical decomposition, small triangles are displayed in front of the line’s caption in case the element contains child elements (in analogy to the logical view). By clicking this triangle, the element gets expanded. Here, the child elements are getting displayed as LifeLines+ and the expanded element itself is reduced to a gray, so-called *summary line*. By clicking onto this summary line, the reverse effect is triggered and the element gets collapsed into a LifeLine+.

PlanningLines are, as the name indicates, intended for depicting planning data afflicted with temporal uncertainties. Besides all aspects that are visualized by LifeLines+, PlanningLines offer additional support for the following rich set of time attributes.

- start interval (earliest starting shift + latest starting shift);

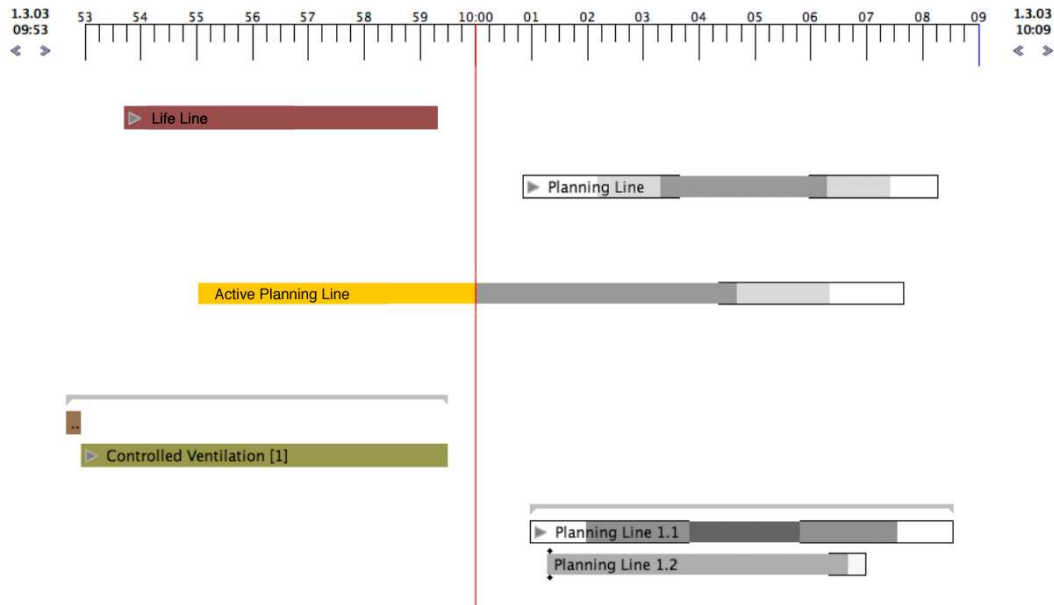


Fig. 7. Temporal view elements (LifeLines+, PlanningLines).

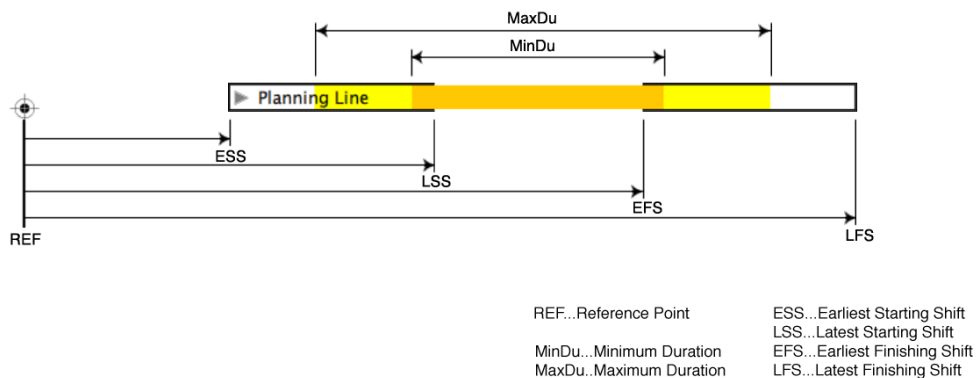


Fig. 8. PlanningLine.

- end interval (earliest finishing shift + latest finishing shift);
- minimum duration; and
- maximum duration.

The glyph itself consists of three main parts: The start cap at the left, the end cap at the right, and the duration bars in between (see Fig. 8). The caps are drawn in black to emphasize their fixed position. The bars in contrary are colored whereas the color of the maximum duration bar has equal hue and saturation but is brighter than the minimum duration bar. Encapsulated bars that can be shifted within the constraints of two mounted caps resemble the

glyph's mental model.

Current Time Indicator and Time Cursor Two elements that have proven to be very useful interaction means are the *current time indicator* and the *time cursor*. The current time indicator is a red vertical line, marking the current time, on the one hand. Furthermore, the current time is displayed precisely in the upper right corner of the application window (see Fig. 1). The time cursor, on the other hand, marks an arbitrary point on the time scale. It is represented by a blue vertical line and can be manipulated by mouse clicking and dragging (see Fig. 1). The precise value of the time cursor is displayed at the bottom of the application window right below the vertical time cursor line. This element might be used to inspect variable and parameter values at certain points in time, measure beginning and ending of plans, or compare various elements.

Visualization of Patient Data The facets below the temporal treatment plan representation are used to display patient data measurements and plan variables. This work focuses on the integrative aspect and the representation of treatment plan information. Several novel approaches for visualizing time-oriented data that can be used for the visualization of patient data are described in [30].

4.1.3 QuickView Panel

A separate possibility to display currently valid variable and parameter values is the so-called QuickView panel in the top part of the application window (see Fig. 1). The panel consists of rectangular areas that can be assigned to the available parameters and variables. A single item shows the current value along with its name, unit, and a trend indicator. Thus, the QuickView panel allows to monitor the most important values by placing them at a prominent position, enlarged in size and without the need to display their complete history in an additional facet.

4.2 View coupling

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

- (1) *Color Cues*. 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 immediately 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, he or 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 transcutaneous measured arterial oxygen saturation (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 sub-steps of the plan are revealed.

5 User-Centered Design, Prototype, and Evaluation

“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.” [31]

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 performed scenario-based using an example protocol. Moreover, the novel PlanningLine glyph has been investigated in-depth by conducting a comparative empirical study in a project management setting.

Figure 9 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 signaled by arrows to and from the user.

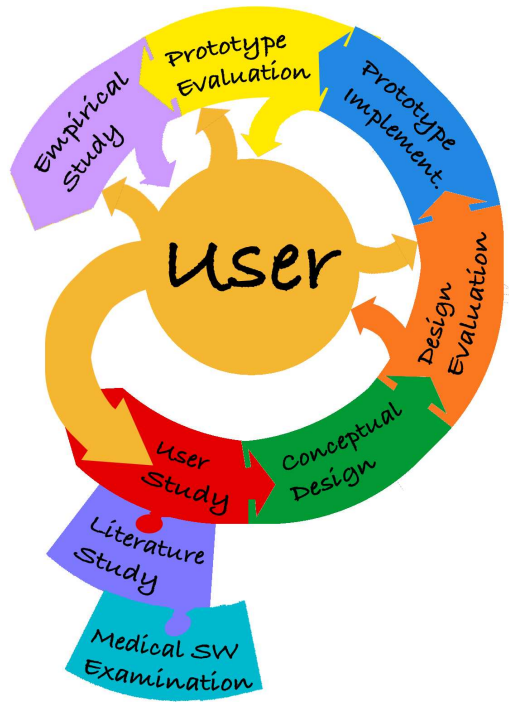


Fig. 9. User-Centered Design.

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. The study was performed and analyzed qualitatively, but structured along an interview guideline. An interview took on average about 45 minutes. It became apparent that clinical guidelines are generally depicted by a special form of flow-charts named clinical algorithm maps as proposed in [4], that 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 initial 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 [12].)

5.2 Design Review

After completing the first “release” version of the conceptual design, we conducted unstructured, qualitative review sessions for getting early feedback regarding our design by two experts (one visualization expert and one medical expert). This early evaluation process was valuable and reduced the risk of investing time and effort in unfruitful initiatives. It showed that we were working in the right direction and pointed out some minor weaknesses of the design we were able to correct quickly. We decided to perform this step as *expert review* rather than by *user testing* because cost and effort are much lower while a lot of valuable information can be gathered by choosing appropriate experts.

5.3 Prototype Implementation

As proof of concept and in order to generate a better impression of interaction issues, we implemented a Java 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 refactored constantly.

In principle, we used the programming language *Java* (JDK 1.4.1) for our implementation. For depicting the plan step elements in the flow-chart-like part of our representation (AsbruFlow, logical view), we utilized the graph drawing framework *JGraph* [32,33]. This is a flexible, small, and powerful package using the model-view-controller (MVC) paradigm and is structured analogous to the standard Swing component *javax.swing.JTree*. All other graphical elements are embedded into the Java Swing standard component framework. For coordination of views, a view manager was built that is in charge of event notifications across the system. Due to the structure of views, a layered dispatch is used for delivering events. This means that events are passed up in the object hierarchy as long as they affect the next level. When the root view manager is reached, it passes the event down to all plan views registered for receiving user interaction events.

5.4 Prototype Evaluation

A scenario-based, qualitative prototype evaluation was carried out by conducting interviews with physicians working in ICU’s. Five of the eight physicians who already participated in the user study at the beginning of this

project took part in the evaluation. The interviews consisted of the four main parts: Introduction, Prototype Presentation, Prototype Testing, and Feedback/Questionnaire [12].

All of the interviewed physicians 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.

5.5 Empirical Comparative Study

Temporal aspects and particularly temporal uncertainties play a crucial role in various application domains. 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. The evaluation was performed in a project management setting involving 48 subjects (undergraduate and graduate students of informatics and business informatics). The reasons for performing this study in a project management setting are – (1) The PlanningLine glyph is well suited for being applied in project management due to similar temporal domain requirements. (2) A quantitative study with a reasonable amount of subjects that have a background in project management could be realized much quicker than involving subjects with a medical and in particular protocol-based care background. Moreover, the experiment was designed on a perceptual and cognitive level rather than a domain-specific one, allowing for generalizing its results to a certain extent.

The empirical study compared the performance of two techniques to visualize task relationships and temporal uncertainties – traditional PERT charts and our novel PlanningLines technique. The reasons for using PERT for comparison are foremost that PERT diagrams are widely known and familiar to most study subjects and that PERT diagrams allow for representing temporal uncertainties. The experiment was paper-based and investigated in two rounds with 48 subjects the time needed to conduct a standard set of graph reading and recognition steps and the number of mistakes on questions regarding these experiment steps. Main results of the study are, that while PERT is well suited for reading single attributes, PlanningLines better support users in judging temporal task uncertainty [34].

6 Supporting Users' Tasks

So how can Markus Zolte, Andrea Habacher, and Heinrich Kovanic benefit from our visualization methods in accomplishing their work tasks as described in Section 2.2?

6.1 Scenario 1

Markus Zolte wants to become acquainted with the hyperbilirubinemia protocol. Therefore, he loads the appropriate Asbru file and maximizes the logical view for examining the logical workflow of the plan. He uses the Fisheye view for keeping an overview while exploring different paths of the plan using the small gray triangles for navigation through the hierarchy. When examining the intensive photo-therapy part, he deactivates the Fisheye view for displaying detail only and reads the abort condition in the red bar on top of the plan to identify cases in which the plan aborts. For getting positional information, he turns on the overview window. Finally, he opens a pull-down menu to see the full list of used parameters and variables.

6.2 Scenario 2

Andrea Habacher just completed treatment along the “Controlled Ventilation” plan and would now like to analyze the treatment history. She adjusts the zoom factor of the time scale in order to display the complete execution interval and explores the duration and position of the different phases. Furthermore, she uses the small gray triangles at the LifeLines+ to navigate to sub-plans. For investigating the “Handle PCO₂” plan, she selects the plan in the logical view and drops it into the temporal view. Subsequently, all instances of the plan are displayed and highlighted. Furthermore, she selects the PtcCO₂ parameter at a pull-down menu for display in the temporal view. When encountering a significant discontinuity of the PtcCO₂ value in one of the plan instances, she recalls the sub-steps of the plan by navigating down the hierarchy in the logical view to investigate which sub-steps of the treatment procedure might have caused this phenomenon. After that, she displays the “patient-state” parameter in the temporal view to examine how the PtcCO₂ value influences it.

6.3 Scenario 3

Heinrich Kovanic is currently treating a patient along the hyperbilirubinemia protocol. He displays the TSB and TSB-change values in the temporal view as well as in the QuickView panel. In order to get basic patient information, he displays the patient record by double clicking the patient's name. After that, he displays all parameters and variables in the temporal view and encounters a rapid increase of the TSB value. He identifies the point in time of this episode by using the time cursor. He selects the plan that has been executed at that time in the temporal view and drops it into the logical view. The logical view navigates to the dropped plan and shows the details of the applied parameter constraints defined by the plan conditions in the upper red and lower green bars. Finally, he double clicks the current time indicator at the upper right of the application window to navigate the temporal view back to the current position of plan execution.

7 Conclusion

Our goal was to develop visualization and interaction methods for supporting medical personnel in computerized protocol-based care. To achieve this goal, we had to consider several data aspects like the logic, structure, and temporal constraints of plans as given at design-time, data of instantiated plans at execution-time, as well as patient data in form of parameters and variables. Applying a multiple views approach helped to master the complexity of the underlying data structure while using visualization methods well known to the domain experts. We have examined the usefulness of our approach performing a multi-step evaluation process including user study, design reviews, prototype evaluation, and a comparative empirical study.

That visualizing the logic of clinical guidelines is useful to support understanding and exploration of protocols has already been proposed and proved years ago [4,3]. Clinical algorithm maps are most widely used in medical education and practice for that matter. This form of representation is clear, simple, and easily graspable – thus served as basis in our visualizations for the representation of a plan's logical structure. But it cannot be applied directly to represent Asbru plans because it lacks a notion for representing hierarchical decomposition, flexible execution order, and state characteristics of conditions. Therefore, we extended this visualization by introducing new element types, an execution sequence indicator, and an enclosing frame containing the plan conditions.

Besides that, visualizing the temporal aspects of already executed plans, cur-

rently running plans, and plans to be executed in future in addition to the logic of treatment plans is vital for analysis and runtime support in medical treatment planning. Key issues of planning are temporal uncertainties inherently related to the temporal dimension. These uncertainties in the form of Asbru’s powerful time annotations are visualized in a simple and meaningful way, fully integrated in the LifeLine-based representation.

The use of software in contrast to paper allows us to support the process of exploring and understanding treatment plans at a higher level. The interactive integration of different visualization methods forms a novel way of combining, relating, and analyzing different kinds of medical data that otherwise would be separated. It enables a meaningful navigation, provides annotations on demand for not overwhelming the viewer, and keeps orientation by using Focus+Context techniques. This increases the flexibility in working with treatment plans. The introduced views focus on different aspects of the data while being tightly coupled to support physicians at their main work tasks.

An additional value, besides communicating plans to domain experts, became apparent during development. The visualization of plans helps to spot problems, bugs, and ambiguities in the formal plan representation which are hard to see and detect otherwise. Furthermore, the visualization serves as an important basis for the communication between medical domain experts and computer scientists.

Moreover, we applied a user-centric approach when developing our visual representation – we involved the end-users from the very beginning by carrying out a user study and evaluated our design as well as our prototype. This increases the quality of design, the user acceptance, and serves as an indicator of the maturity of development. We used a well known graphical representation as basis and introduced a cleaned up interface that has a simple and transparent structure with only a handful of different visual elements which are easy to learn and comprehend. The interaction is carried out intuitively by applying well known techniques from standard software supported by different Focus+Context techniques for keeping an overview. The most important user requirement of being time-saving is achieved by combining intuitive navigation and rich information presentation in a structured way.

Acknowledgments

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] S. Miksch, Plan Management in the Medical Domain, *AI Communications* 12 (4) (1999) 209–235.
- [3] D. C. Hadorn, Use of Algorithms in Clinical Practice Guideline Development: Methodology Perspectives, *Clinical Practice Guideline Development: Methodology Perspectives* 0009 (95) (1995) 93–104.
- [4] Society for Medical Decision Making, Proposal for Clinical Algorithm Standards, *Medical Decision Making* 12 (02) (1992) 149–154.
- [5] E. Tufte, S. M. Powsner, Graphical Summary of Patient Status, *The Lancet* 344 (8919) (1994) 386–389.
- [6] 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.
- [7] C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, B. Shneiderman, LifeLines: Using Visualization to Enhance Navigation and Analysis of Patient Records, in: C. G. Chute (Ed.), *Proceedings of the 1998 American Medical Informatic Association Annual Fall Symposium, AMIA, Bethesda, MD, 1998*, pp. 76–80.
- [8] 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.
- [9] P. A. de Clercq, J. A. Blom, H. H. M. Korsten, A. Hasman, Approaches for Creating Computer-Interpretable Guidelines that Facilitate Decision Support, *Artificial Intelligence in Medicine* 31 (1) (2004) 1–27.
- [10] www.openclinical.org, Open Clinical - Knowledge Management for Medical Care, <http://www.openclinical.org> (Accessed: 26 March 2006).
- [11] R. Kosara, S. Miksch, Metaphors of Movement — A User Interface for Manipulating Time-Oriented, Skeletal Plans, *Artificial Intelligence in Medicine* 22 (2) (2001) 111–132.
- [12] 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).
- [13] 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.

- [14] C. Combi, L. Portoni, F. Pincioli, 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-Verlag, New York, 1999, pp. 301–311.
- [15] P. Messner, Time Shapes - A Visualization for Temporal Uncertainty in Planning, Master's thesis, Vienna University of Technology, Institute of Software Technology and Interactive Systems, Vienna, Austria (April 2000).
- [16] S. Card, J. MacKinlay, B. Shneiderman, Readings in Information Visualization: Using Vision to Think, Morgan Kaufmann Publishers, San Francisco, 1998.
- [17] J.-F. Rit, Propagating Temporal Constraints for Scheduling, in: T. Kehler, S. Rosenschein, R. Filman, P. F. Patel-Schneider (Eds.), Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86), Morgan Kaufmann, Los Altos, CA, 1986, pp. 383–388.
- [18] C. Combi, G. Pozzi, HMAP - A Temporal Data Model Managing Intervals with Different Granularities and Indeterminacy from Natural Language Sentences, The VLDB Journal 9 (4) (2001) 294–311.
- [19] S. B. Cousins, M. G. Kahn, The Visual Display of Temporal Information, Artificial Intelligence in Medicine 3 (6) (1991) 341–357.
- [20] Philips Medical Systems, IntelliVue, http://www.medical.philips.com/main/products/patient_monitoring/products/patient_monitors/ (Accessed: 26 March 2006).
- [21] Picis, Chart+ for Critical Care, http://www.picis.com/html/products/module_chart%2Bcritcare.html (Accessed: 24 May 2003).
- [22] Picis, Visual Care, http://www.picis.com/html/products/module_visualcare.html (Accessed: 24 May 2003).
- [23] Critical Care Company, QCare, <http://www.c3.be> (Accessed: 26 March 2006).
- [24] W. Aigner, S. Miksch, Supporting Protocol-Based Care in Medicine via Multiple Coordinated Views, in: J. Roberts (Ed.), Proceedings International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV04, IEEE Press, Los Alamitos, CA, 2004, pp. 118–129.
- [25] 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 Press, Los Alamitos, CA, 1998, pp. 8–14.
- [26] 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, Amsterdam, 2004, pp. 1–15.

- [27] D. Schaffer, Z. Zuo, S. Greenberg, L. Bartram, J. Dill, S. Dubs, M. Roseman, Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods, *ACM Transactions on Computer-Human Interaction* 3 (2) (1996) 162–188.
- [28] Y. K. Leung, M. D. Apperley, A Review and Taxonomy of Distortion-Oriented Presentation Techniques, *ACM Transactions on Computer-Human Interaction* 1 (2) (1994) 126–160.
- [29] W. Aigner, S. Miksch, B. Thurnher, S. Biff, PlanningLines: Novel Glyphs for Representing Temporal Uncertainties and their Evaluation, in: E. Banissi, et al. (Eds.), *Proceedings of the 9th International Conference on Information Visualisation (IV05)*, IEEE Press, Los Alamitos, CA, 2005, pp. 457–463.
- [30] R. Bade, S. Schlechtweg, S. Miksch, Connecting Time-Oriented Data and Information to a Coherent Interactive Visualization, in: E. Dykstra-Erickson, M. Tscheligi (Eds.), *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2004)*, ACM Press, New York, 2004, pp. 105–112.
- [31] P. Gorman, M. Helfand, Information Seeking in Primary Care: How Physicians Choose Which Clinical Questions to Pursue and Which to Leave Unanswered, *Medical Decision Making* 15 (1995) 113–9.
- [32] G. Alder, Design and Implementation of the JGraph Swing Component, Tech. Rep. 1.0.6, JGraph Ltd. (February 2002).
- [33] G. Alder, The Home Page of JGraph, <http://jgraph.sourceforge.net> (Accessed: 26 March 2006).
- [34] S. Biff, B. Thurnher, G. Goluch, D. Winkler, W. Aigner, S. Miksch, An Empirical Investigation on the Visualization of Temporal Uncertainties in Software Engineering Project Planning, in: J. Verner, G. H. Travassos (Eds.), *Proceedings of Empirical Software Engineering 2005 (ISESE 2005)*, IEEE Press, Los Alamitos, CA, 2005, pp. 437–446.