CareVis: Integrated Visualization of Computerized Protocols and Temporal Patient Data

Wolfgang Aigner and Silvia Miksch

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

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 the development of CareVis – interactive visualization methods to integrate and combine classical data visualization with the visualization of treatment information in terms of logic and temporal aspects. We provide multiple simultaneous views to cover different aspects of a 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.

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.

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 semiautomatically 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 [8]. 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.

The Plan Representation Language Asbru

Asbru is a time-oriented, intention-based, skeletal plan-specification representation language that is used in the *Asgaard* Project[1] to represent clinical guidelines and protocols in XML. *Asbru* can be used to express clinical protocols as skeletal plans that can be instantiated for every patient. It was designed specific for the set of plan-management tasks [13]. The major features of *Asbru* are that;

- prescribed actions and states can be continuous;
- intentions, conditions, and world states are temporal patterns;

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`).

- 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.

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.

Data Characteristics

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

Starting from this, we examine related work as highlighted in the upcoming section. Following that, we introduce our multiple view approach and explain its design as well as prototype implementation. Then, we describe the user-centered design approach we undertook during development. Finally, we sum up our findings and present work left to be done in future.

Related Work

We investigated related work in the areas of medical treatment planning, information visualization, and commercial medical software as described in the following.

Medical Treatment Planning

Clinical Algorithm Maps. The most widely used visual representation of clinical guidelines are so-called *flow-chart algorithms*, also known as *clinical algorithm maps* [9]. 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* [17]. 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.

AsbruView. AsbruView [11] is a graphical tool that supports authoring and manipulation of *Asbru* plans. AsbruView utilizes metaphors of running tracks and traffic control to communicate important concepts and uses glyphs to depict the complex time annotations used in *Asbru*. The interface consists basically of two major parts or views respectively – one captures the topology of plans, whereas the second one shows the temporal dimension of plans but no depiction of plan and patient data is possible. The intention of AsbruView is to support plan creation and manipulation. It is not supposed to communicate the combination of logic, structure, and temporal aspects of an *Asbru* plan and patient data during execution or analysis.

Other Scientific Projects. Other scientific work [18, 5, 15] 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 [14] and [19].)

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 [1].

Information Visualization Methods

Visualizing Logical Sequences. Other possibilities to visualize logical sequences away from flow-charts are *Structograms*, *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 depicting hierarchical decomposition, flexible execution order, and the state characteristic of conditions together in their basic forms as needed for representing *Asbru* plans.

Visualizing Hierarchical Data. The most popular techniques for visualizing hierarchical data are *Trees*. Further techniques for that matter are *Treemaps* [10] that introduce an additional dimension by proportional space assignment. But these 2D techniques have no notion to depict logical sequences, concurrency, or states.

Visualizing Time-Oriented Data. Time is a very important data characteristic but methods for visualizing time other than in time-series plots are not well known. The probably best known method among them are *GANTT charts* and their utilized *Time Lines*. An extension of *Time Lines* are *LifeLines* [15] that have been used for example to visualize personal histories. A drawback of these methods is that they mostly work retrospectively, thus only depict temporal attributes in the past. To overcome this limitation, other visualization techniques like *Temporal Objects* [7], *Paint Strips* [6], and *SOPOs* [12] were developed. These techniques can be used to visualize complex notions of time like temporal uncertainties that can be utilized to depict future planning data. The main flaw of the presented techniques is that, except *GANTT charts*, they cannot depict hierarchies and logical sequences can only be represented implicitly.

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 that are familiar to most physicians [1]. 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 treatment planning logic at all.

CareVis: Our Visualization Approach

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 decided to use 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 [16].

Having introduced the domain prerequisites, data characteristics, and related work, we now present the different views in detail.

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 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).

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* [9] 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 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-hand side. Right besides this icon, the abort condition is printed textually. This condition has the following semantic – if the condition evaluates to TRUE, the current plan gets aborted. Furthermore, this condition is evaluated and checked during the entire execution of all steps in the plan body. 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 semantic of this condition is – if and only if this condition evaluates to TRUE, the plan can complete successfully. 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^{TM} system.

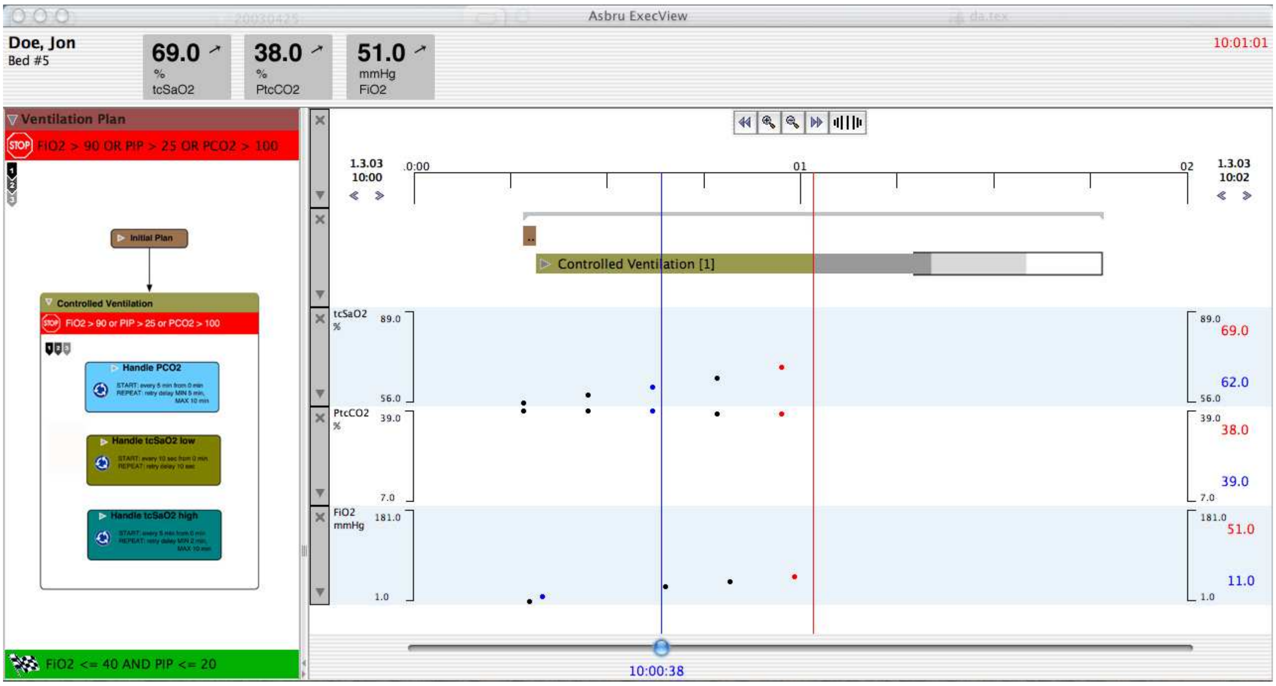

Figure 1: *CareVis application window (top: QuickView Panel, lower left: Logical View, lower right: Temporal View).*

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 [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 glyph consisting of two encapsulated bars, representing minimum and maximum duration, that are bounded by two caps that represent the start and end intervals. Encapsulated bars that can be shifted within the constraints of two mounted caps resemble the glyph's mental model.

The navigation is achieved analogous to the logical view by using small gray triangles which expand and collapse elements. In order to prevent visual overload and an overly cluttered display, expanded elements are shrunk to summary lines and colored in light gray.

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.

The temporal view is divided into collapsable facets which can be added and removed dynamically. The most important element of this view is the time scale. It determines the portion of time being displayed. Below that, one facet is displayed containing the temporal aspects of the treatment plan elements followed by several facets containing 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. Another focus+context technique is applied to the time axis itself. *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 [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.

Prototype

As a proof-of-concept and in order to generate a better impression of interaction issues, we implemented a Java prototype. For depicting the plan step elements in the flow-chart-like part of our representation, we used the graph drawing framework *JGraph* [3]. This is a flexible, small, and powerful package using the Model-View-Controller 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.

User-Centered Design

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.

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 [17] 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 [1].)

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.

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 [1].

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.

Conclusion

Our goal was to develop visualization and interaction methods that integrate various sources of data and information to support the analysis of patient data. *CareVis* represents relevant information in a coherent way using visualization methods familiar to medical domain experts. To achieve this goal, we had to consider several data aspects like the logic, structure, and temporal constraints of plans as well as patient data in form of parameters and variables. Applying a multiple simultaneous 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 3-step evaluation process including user study, design reviews, and prototype evaluation.

Our visualization and interaction methods were mainly designed for intensive care settings, but most aspects are also applicable to low frequency domains as for example diabetes treatment. In this case, the QuickView Panel might be turned off and other data visualization techniques like LifeLines could be used for visualizing patient data.

CareVis enables a meaningful navigation, provides annotations on demand for not overwhelming the viewer, and helps to keep orientation by using focus+context techniques, thus increasing the flexibility in working with treatment plans and patient data. The introduced views focus on different aspects of the data while being tightly coupled to support physicians at their main work tasks.

Acknowledgements This project is supported by "Fonds zur Förderung der wissenschaftlichen Forschung - FWF" (Austrian Science Fund), grant P15467-INF.

References

[1] 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.

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

[3] G. Alder. Design and Implementation of the JGraph Swing Component. Technical Report 1.0.6, February 2002.

[4] R. Bade, S. Schlechtweg, and 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*, pages 105–112. ACM Press, 2004.

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

[6] L. Chittaro and C. Combi. Visual Definition of Temporal Clinical Abstractions: A User Interface Based on Novel Metaphors. In *Proceedings of the 8th Conference on Artificial Intelligence in Medicine Europe (AIME 01), Lecture Notes in Computer Science*, volume 2101, pages 227–230, 2001.

[7] C. Combi, L. Portoni, and F. Pinciroli. Visualizing Temporal Clinical Data on the WWW. In W. Horn, Y. Shahar, G. Lindberg, S. Andreassen, and J. Wyatt, editors, *Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making (AIMDM'99)*, pages 301–311. Springer, jun 1999.

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

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

[10] B. Johnson and B. Shneiderman. Treemaps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures. In *Proceedings of the IEEE Information Visualization (IV 91)*, pages 275–282. IEEE, 1991.

[11] R. Kosara and S. Miksch. Metaphors of Movement — A User Interface for Manipualting Time-Oriented, Skeletal Plans. *Artificial Intelligence in Medicine*, 22(2):111–132, 2001.

[12] 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.

[13] S. Miksch. Plan Management in the Medical Domain. *AI Communications*, 12(4):209–235, 1999.

[14] 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, and Stefanelli. Comparing Computer-Interpretable Guideline Models: A Case-Study Approach. *The Journal of the American Medical Informatics Association (JAMIA)*, 10(1):52–68, 2003.

[15] C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, and 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*, pages 76–80, November9–11 1998.

[16] J. C. Roberts. On Encouraging Multiple Views for Visualization. In E. Banissi, F. Khosrowshahi, and M. Sarfraz, editors, *IV'98 – Proceedings International Conference on Information Visualization*, pages 8–14. IEEE Computer Society, July 1998.

[17] Society for Medical Decision Making. Proposal for Clinical Algorithm Standards. *Medical Decision Making*, 12(02):149–154, April-June 1992.

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

[19] www.openclinical.org. Open Clinical - Knowledge Management for Medical Care, 2003.

Address for correspondence

Wolfgang Aigner (`aigner@asgaard.tuwien.ac.at`)
Institute of Software Technology & Interactive Systems (ISIS), Vienna University of Technology, Favoritenstraße 9-11/188, A-1040 Vienna, Austria, Europe