Visualization Methods to Support Guideline-Based Care Management

Wolfgang AIGNER [a], Katharina KAISER [b], Silvia MIKSCH [a,b,1]
[a] *Danube University Krems, Austria*
[b] *Vienna University of Technology, Austria*

Abstract. Authoring clinical guidelines as well as observing the execution and the maintenance of these is a time-consuming and cumbersome task. Usually, clinical guidelines are represented in conceptual models, which are very hard to understand by domain experts. Furthermore, to analyze the effectiveness and usefulness of clinical guidelines they need to be shown in connection with the patients' data. In this overview book chapter we present different methods to visualize clinical guidelines, patients' data, and the connection thereof. Finally, we illustrate how the different visualization methods support the various tasks in plan management.

Keywords. Visualization, Guideline-based Care

Introduction

Current research into the use of Information Visualization in guideline-based care focuses on support for the knowledge acquisition process (the authoring of computer-interpretable guidelines and protocols) and on ways to help explore plans and monitor and communicate plan execution (over time). In the following we define the concepts of Information Visualization (InfoVis) and the core tasks of plan management to illustrate how InfoVis can support these processes and tasks.

Information Visualization

Growing use of modern information technologies in clinical care is increasing both the amount and complexity of information and data accessible to health professionals. By providing interactive visual representations of data and information, Information Visualization aims to deepen exploration of the "information space", support optimal use of data and information, and help avoid overload. InfoVis is concerned with the development of interactive visual representations of abstract, multidimensional data, information, and knowledge to help users gain a deeper understanding of the contents of a domain by revealing, for example, new insights, previously unknown facts and relationships, or providing explanations for complex situations [4,34].

Chittaro [5] summarizes some of the goals of InfoVis technologies for healthcare:

[1] Corresponding Author: Danube University Krems, Department of Information and Knowledge Engineering (ike), Dr.-Karl-Dorrek-Strasse 30, A-3500 Krems, Austria; E-mail: silvia.miksch@donau-uni.ac.at.

1. To allow "*users to explore available data at various levels of abstraction*".
2. To give "*users a greater sense of engagement with data*".
3. To give "*users a deeper understanding of data*".
4. To encourage "*the discovery of details and relations which would be difficult to notice otherwise*".
5. To support "*the recognition of relevant patterns by exploiting the visual recognition capabilities of users*".

Visualization methods and tools have been used in the medical domain for many years. The majority of applications have been in the field of scientific visualization, for example, 3D volume visualization tasks, X-ray, or computer tomography visualizations. However, tasks involving abstract data (such as patients' data, treatment data, or lab results) or computerized guidelines have not been targeted by InfoVis researchers until quite recently.

Tasks in Plan Management

In the last years different approaches to manage clinical guidelines and protocols were introduced. These approaches range from contributions of the planning community to concepts within the Medical Informatics community [14]. As a possible solution to that diversity the concept of *plan management* was introduced, where clinical guidelines are seen as (time-oriented) plans [14].

Plan management involves more than specifying a problem, generating a possible solution path to reach a goal state from an initial state, and executing this solution path. Plan management includes everything from designing a particular plan or a hierarchy of plans to the real-world execution and evaluation of such plans (compare [18]). Plan management consists of various tasks. These tasks are not performed in a strict sequence (one by one) and can be clustered into groups according to their purposes. Many of these groups overlap in their functionalities. Additionally, we can distinguish tasks, which need to be performed mainly at design time and those which are done mainly during execution time of plans (as shown in Table 1).

According to Table 1 two tasks are mentioned which address the visualization of clinical guidelines explicitly: (1) *plan visualization during design time*, and (2) *plan and data visualization during execution time*. The first refers to authoring of computer-interpretable clinical guidelines, where the main focus lies on the communication of the different clinical guideline components to domain experts. The second one handles the visual representation of clinical guidelines in connection with patients' data.

However, all other tasks in Table 1 can be supported by visual methods, too. *Advanced Plan Editing* and *Domain-Specific Annotations* can be seen as extended authoring of clinical guidelines, which can be eased by visualization methods. *Plan-Scenario Testing* can be seen as off-line execution of clinical guidelines, which can utilize the plan visualization during execution. Furthermore, *Plan Verification and Validation* could indirectly be supported by visual methods, however, these are quite complicated tasks and verbal annotations seem more suitable. All the remaining tasks during execution time can be supported by visual methods in one way or another. One outstanding task is *Plan Modification / Alternatives*: This covers, on the one hand, the maintaining of clinical guidelines when new medical knowledge is discovered and needs to be incorporated in the clinical guidelines (in the sense of "*living guidelines*"). On the other hand, changes

Table 1. Tasks in Plan Management. A list of tasks defined in plan management according to their primary application time [14].

Tasks mostly done at design time	
Plan Generation	starts from an initial state description and creates a path of activities to reach the desired goal (progressive way), or starting from the goal (regressive way)
Advanced Plan Editing	provides guided support to author plans and helps to browse plans or a plan hierarchy
Domain-Specific Annotations	provides structured support to write domain assumptions or domain activities
Plan Verification	examines the method semantics of plans and plan hierarchies
Plan Validation	examines the domain semantics of plans and plan hierarchies
Plan-Scenario Testing	enacts groups of plans by applying definite domain scenarios
Plan Visualization	communicates efficiently sets of plans to domain experts
Tasks mostly done at execution time	
Plan Selection	assigns task according to state/situation
Plan Adaptation	adjusts plans or plan hierarchies to distinct situations at the starting time
Plan Execution	performs selected activities
Plan Monitoring	compares assumptions with reality
Plan Modification / Alternatives	handles changes in the environment
Plan Evaluation / Critiquing	analyzes executed plans or plan hierarchies according to their goals and intentions
Plan and Data Visualization	supervises and communicates plans or plan hierarchies in connection with the patients' data
Plan Rationale / History	explains executed plans or plan hierarchies

in health condition of the patients or in the medical environment can force a modification of the therapeutic activities.

We structure the visualization methods according to abstractions of these tasks: (1) visualizing clinical (computer-interpretable) guidelines seen as plans or activities, (2) visualizing patients' data seen as multidimensional information space, and (3) visualizing patients' data in connection with clinical guidelines. We illustrate afterwards, how the visualization methods can be utilized in design time as well as execution time (compare Table 2).

1. Visualizing Clinical Guidelines seen as Plans or Activities

Different frameworks have been developed to implement clinical guidelines in a computer-interpretable format, such as Asbru, EON, GLIF, Guide, Prodigy, and PROforma (compare [16,7] and Chapter 2 in this book). These frameworks are tailored for specific classes of guidelines, specific users, and specific organizations. Each framework supports specific guideline representation languages and various tools and techniques have been developed to ease the guideline modeling and visualization process. They can be roughly classified into (1) *model-centric* and (2) *document-centric* approaches [13]. In the model-centric approach, a conceptual model is formulated by domain experts. Thus, the relationship between the model and the original document of the clinical guideline is only indirect. In the document-centric approach, markup-based tools are used to systematically mark up the original guideline in order to generate a semi-formal model of the marked text part. The first category covers many approaches and is more visual-oriented, the latter is more text-based and only a few approaches are available. In the following we present a selection of examples according to these two categories. There are different flowchart-based visualizations of clinical guidelines available and Protégé (compare Section 1.1.3) is used in different frameworks. We only present representatives of these.

1.1. Model-Centric Approaches

The *model-centric* approaches to author clinical guidelines focus on the creation of a conceptual model of the original guideline without keeping the direct connection between these two representations. We illustrate this approach by various examples ranging from simple flowcharts to sophisticated visual representations of guidelines.

1.1.1. Clinical Algorithm Maps

The most widely used visual representation of clinical guidelines are so-called *flowchart algorithms*, also known as *clinical algorithm maps* [9]. 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* [27]: *"However, since algorithmic logic is wired implicitly into a protocol, it is difficult to learn an algorithm from a protocol. By contrast, flowchart 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."* The proposed standard includes a small number of different symbols and some rules on how to use them. One additional feature to standard flowcharts are annotations that include further details, e.g., citations to supporting literature, or clarifications for the rationale of decisions.

A big advantage of using flowcharts is that they are well known among physicians and require minimal additional learning effort. A drawback of basic flowchart 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, flowcharts cannot be used to represent concurrent tasks or the complex conditions as used in Asbru [24] 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.

1.1.2. Nassi-Shneiderman Diagrams, PERT Charts, Gantt Charts, Petri Nets, and State Transition Diagrams

Other possibilities to visualize logical sequences besides flowcharts are *Structograms (Nassi-Shneiderman Diagrams)*, *PERT charts*, *Gantt charts*, *Petri nets*, and *State Transition Diagrams*. These techniques focus on other purposes and some of them are more powerful and expressive than flowcharts. 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 clinical guidelines.

1.1.3. *Protégé*

Protégé is an open source ontology development and knowledge acquisition environment [8]. Protégé is a Java tool, which provides an extensible architecture for the creation of customized knowledge-based tools. It assists users in the construction of large electronic knowledge bases. It has an intuitive user interface that enables developers to create and edit domain ontologies and supports customized user-interface extensions, incorporates the Open Knowledge Base Connectivity (OKBC) knowledge model, and interacts with standard storage formats such as relational databases, XML, and RDF. Protégé supports to author guidelines in various models (e.g., EON, GLIF, Prodigy, Proforma). See Figure 1 for an example.

1.1.4. *VisiGuide (Part of DeGeL)*

The VisiGuide is part of the DeGeL (Digital electronic Guideline Library) project [25] and is a multi-ontology guidelines browser. Its purpose is to present a large amount of guidelines clustered by the semantic indices and allow the user focusing on a single guideline by exploring its parts (see Figure 2).

1.1.5. *AsbruView - SopoView*

AsbruView [12][2] is a graphical user interface developed in the Asgaard/Asbru project to support the development of guidelines and protocols in Asbru [24]. Asbru is a complex

[2]http://www.asgaard.tuwien.ac.at/tools/asbruview.html [accessed May 2007]

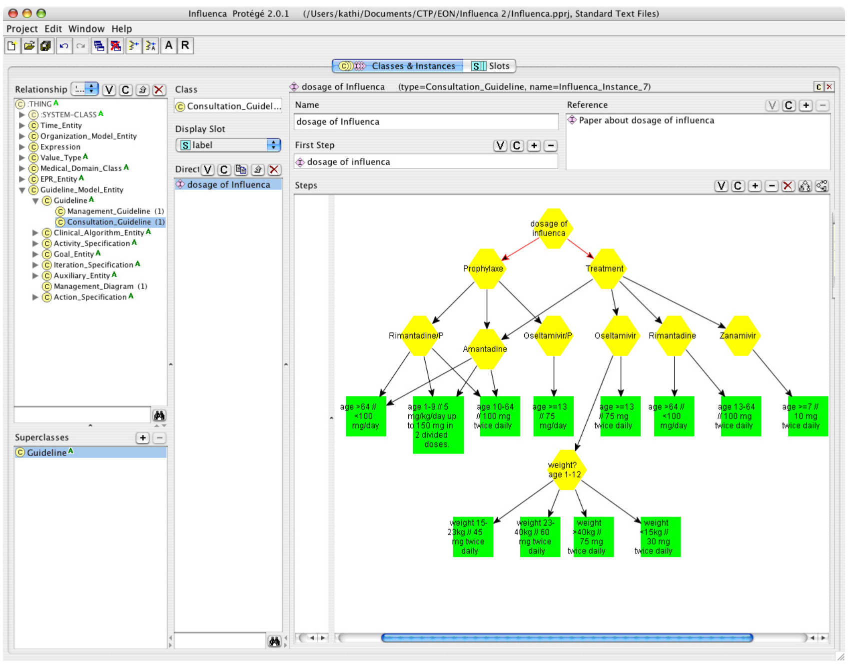


Figure 1. Protégé – A knowledge acquisition tool to author a guideline for managing chronic cough. The guideline model being used in this application is Dharma, part of the EON framework.

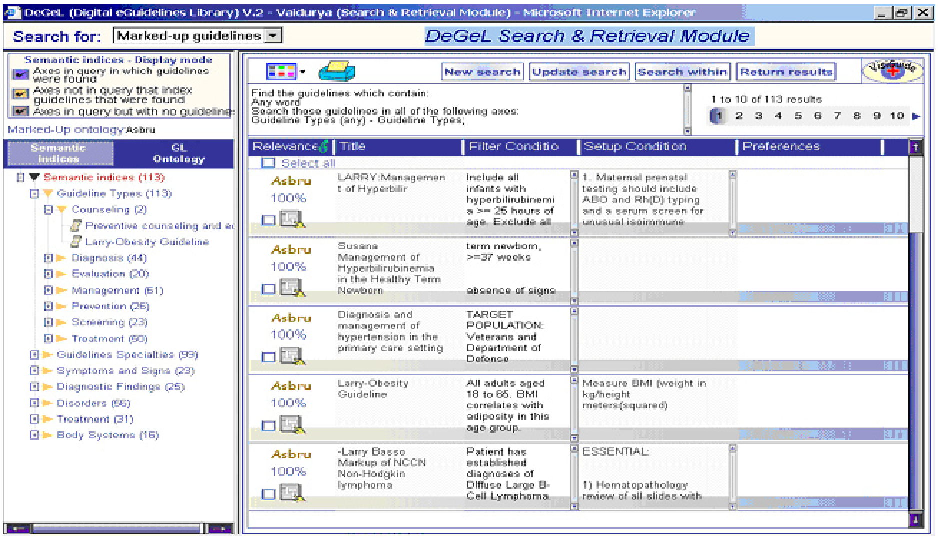


Figure 2. VisiGuide [25] – A visualization and browsing tool for multiple guidelines.

language, which cannot be fully understood by physicians who have no or hardly any training in formal methods. AsbruView is a tool to make Asbru accessible to physicians, and to provide a visual overview of the guideline hierarchy and other Asbru-specific components. AsbruView is based on visual metaphors of running tracks and traffic signs to make the underlying concepts easier to grasp. Currently, AsbruView (see Figure 3) provides four views: *Topological View* (see upper part of Figure 3) displays the relationship between guidelines seen as plans without a precise time scale, *Temporal View* (see lower part of Figure 3) concentrates on the temporal dimensions of plans and conditions, *SOPOView* gives another view of the temporal dimensions of plans, and *XML View*. The metaphors and graphical representations of *AsbruView* have proved to be useful in communicating Asbru's concepts to physicians. Users get a better overview of the therapy steps than from tables, while at the same time being able to see the precise temporal constraints of plans (which is not the case with flowcharts).

1.1.6. AsbruFlow (Part of CareVis)

AsbruFlow[3] is part of CareVis prototype [2] and helps to communicate the content and logic of Asbru treatment plans to medical domain experts [24]. AsbruFlow is based on the idea of flowchart-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.

In order to prevent getting lost within a guideline 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 Figure 4, left (a)). The second technique is a *Fisheye view* which distorts elements that

[3] http://ieg.ifs.tuwien.ac.at/projects/carevis/ [accessed June 2007]

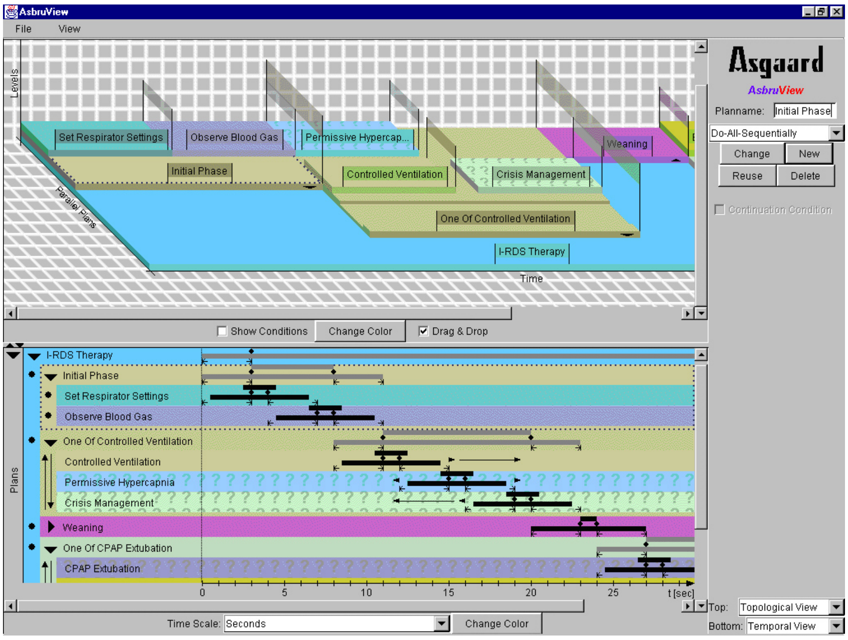


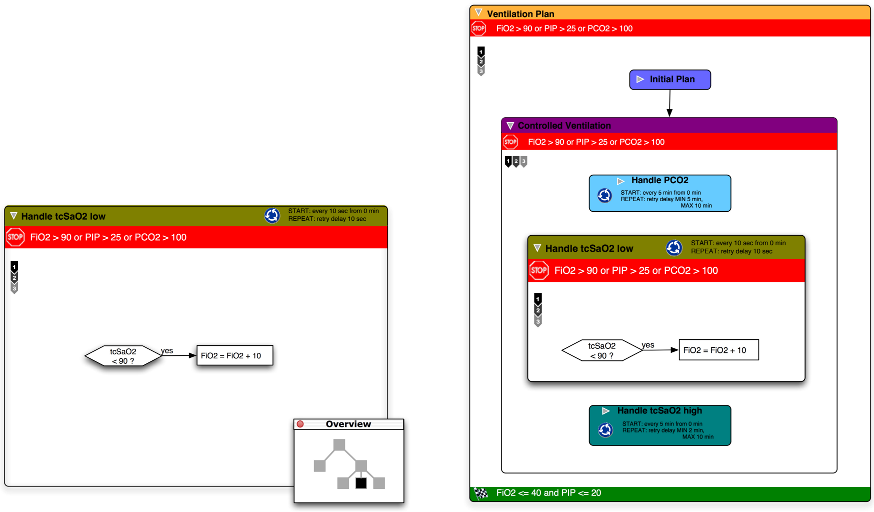
Figure 3. AsbruView [12] – A graphical user interface for the guideline representation language Asbru showing a part of a guideline for infants' respiratory distress syndrome (I-RDS) of new-born infants. *Topological View* is in the upper part and *Temporal View* in the lower part [12].

are out of the current focus geometrically by shrinking and moving (see Figure 4, right (b)) based on the work of Schaffer et al. on hierarchically clustered networks [22].

1.1.7. Arezzo – Tallis Toolset

The first implementation of software to create, visualize, and enact PROforma guidelines [29] was Arezzo. Its successor is the Tallis Toolset [28].[4] It includes software and training materials to create, publish, and enact clinical knowledge applications over the web. It is based on the PROforma language [29] for modeling clinical processes. The toolset consists of three components that interact with each other: the Composer, the Tester, and the Engine. PROforma tasks (i.e., plans, decisions, actions, enquiries) can be connected to form a network. Such a network is sometimes called a "workflow". The Composer provides a graphical interface to support the generation of such task networks. The development of a network is a two-step process: (1) a high-level structure of the process is laid out and assembled as a network; (2) detailed knowledge that is required to enact each component task is entered as task attributes. The Tallis Tester is a tool for testing and debugging the logic of a developed PROforma application. A tested and debugged application can then be enacted by the Tallis Engine.

[4]http://www.acl.icnet.uk/lab/tallis [accessed May 2007]



(a) Overview+Detail. (b) Fisheye view.

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

1.1.8. GLARE

GLARE [30] is a domain-independent system for acquiring, representing, and executing clinical guidelines. The graphical interface of the authoring tool is quite similar to the Tallis Toolset and the applications of Protégé. The different language elements are coded by simple graphical icons and the flow of the guideline is represented similar to a flowchart.

1.2. Document-Centric Approaches

The *document-centric* approach to author clinical guidelines preserves the connection of the original guideline written in text and its semi-formal model. This approach is more text-based and only a few examples exist. To illustrate this approach we give two examples, next. Similar markup tools are provided in the Stepper project [21] and the Uruz project, which is part of the DeGeL project [25].

1.2.1. GEM Cutter

The GEM Cutter [19][5] facilitates the transformation of clinical guidelines into the Guideline Elements Model (GEM) [26], which is an XML-based guideline document model. GEM Cutter's main screen consists of three vertical segments showing the original text of the guideline, a tree view of the developing GEM file, and additional information about the GEM file. It is a pure text-based approach.

[5] http://gem.med.yale.edu/GEMCutter/gemcutter.htm [accessed: June 2007]

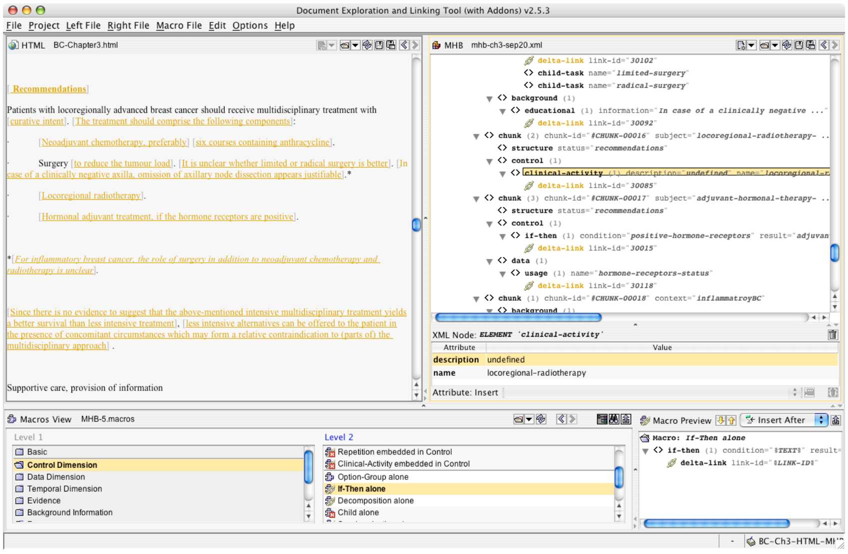


Figure 5. Document Exploration and Linking Tool / Addons (DELT/A) – The two top panes show linked views of the same guideline in different formats and the lower pane is the Macro view to ease authoring of guidelines.

1.2.2. Document Exploration and Linking Tool / Addons (DELT/A)

DELT/A [33] [6] provides two main features: (1) linking between a textual guideline and its formal representation, and (2) applying design patterns in the form of macros. DELT/A's user interface (see Figure 5) consists of three panes. The top left and right panes provide equivalent views to either edit XML files or HTML files. The Macros pane provides either a structure view, search view, or insertable macros view, as well as a preview of the current macro.

DELT/A allows the definition of links between the original guideline and the target representation, which gives the user the possibility to find out where a certain value in the XML-language notation comes from. Using macros allows creating and extending specific XML files more easily through the usage of common design patterns.

In this section we have illustrated how clinical guidelines seen as plans and processes can be visualized. In the next section we give examples of visualizing patients' data.

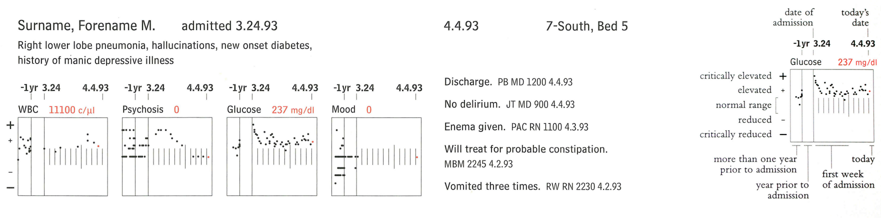
2. Visualizing Patients' Data seen as Multidimensional Information Space

In medicine, large amounts of information are generated and have to be processed mostly by humans. Graphical representations help to make this myriad of information graspable and are a crucial part in the workflow of healthcare personnel. These representations range from classical Fever Curves and EEG Time-Series Plots as found in many commer-

[6] http://ieg.ifs.tuwien.ac.at/projects/delta/ [accessed May 2007]

cial patient data monitoring systems to information rich patient's status overviews (see Figure 6). The graphical summary of patient's status by Tufte and Powsner [20] makes use of concepts like small multiples, Focus+Context, or the integration of textual and graphical information. It manages to display information on a single page that normally fills up entire file folders and would require serious effort to summarize it.

Conceptually, patients' data can be seen as multidimensional information spaces. These information spaces are heterogeneous in multiple ways – quantitative and qualitative data; a mixture of numerical values, text, and images; high-frequency and low-frequency data; raw data and data abstractions. Particularly, this heterogeneity and the need to provide integrated views to create a comprehensive picture of a patient's status as well as its evolution over time impose substantial challenges to visualization design. Besides the visual representation itself, interactivity is a prime concern. It allows for an active interplay of the user and the visualization in order to adapt to the user's information needs for particular tasks and provide additional detail where needed.



(a) Part of overview screen.

(b) Legend.

Figure 6. Graphical Summary of Patient's Status [20] – Concise summary of patient's information. Uses *small multiples*, *Focus+Context*, and integrates textual as well as graphical information.

2.1. Data and Information Visualization

In the upcoming section, we present information visualization techniques for patient's data that go beyond simple line plots of patient's data monitoring systems.

2.1.1. Time Lines and LifeLines

A simple and intuitive way of depicting time-oriented activities is by drawing a horizontal line for the time span the activities took. This form of visualization is called *Time Line* which is a very powerful visualization technique used long before computers even appeared [32]. An extension of Time Lines are *LifeLines* [17] which utilize horizontal bars to represent the temporal location and duration of data (see Figure 7). They were applied to represent personal histories and patient's 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.

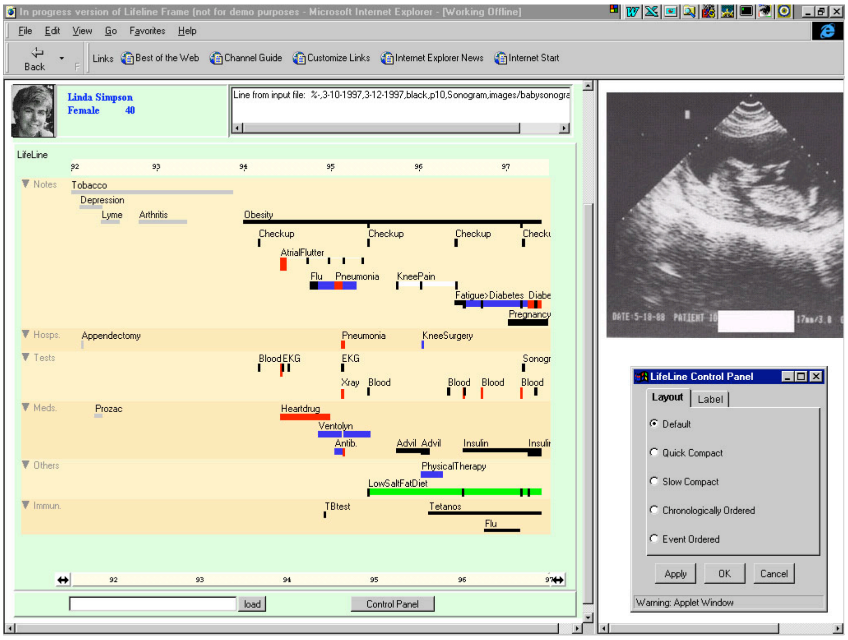


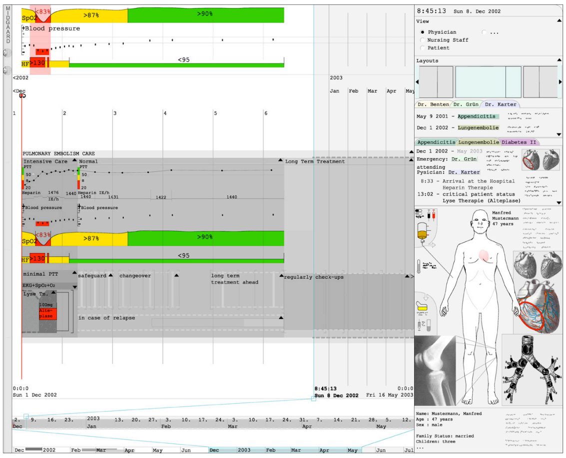
Figure 7. LifeLines [17] – Horizontal bars are used to show the temporal location and duration of information elements. Example shows patient's information.

2.1.2. Midgaard

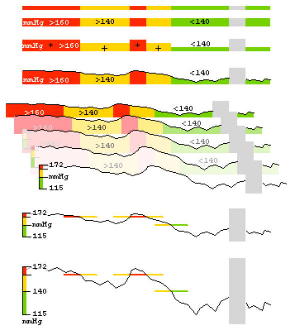
Midgaard is an interactive and adaptive visualization technique for intensive care data (see Figure 8(a)). It provides a number of user interactions such as browsing, pan+zoom, focus+context, and smoothly integrated semantic zoom functionality (see Figure 8(b)). In addition to the visualization of quantitative time-series data, qualitative scales as well as qualitative/quantitative hybrids might be displayed. Moreover, Midgaard allows for simultaneous representation of high- and low frequency data typically found in medical care through intelligent zooming and aggregation techniques. In the value domain, methods for coping and representing measurement deviation, trustability of data points, and missing data are provided. Furthermore, Midgaard visualizes patients' data in combination with assigned clinical guidelines.

2.1.3. VIE-VISU

An interactive glyph technique that is used for time-oriented analysis of electronic patient's records is *VIE-VISU* [11]. A glyph is a graphical object using different geometric and visual attributes to encode multidimensional data structures in combination [34]. The motivation for VIE-VISU's development was the fact that paper-based analysis of patient's records is very hard to conduct because many parameters are involved and an overall assessment of the patient's situation is hard to maintain. The glyph display helps to combine different measurements, maintain their relationships, show their development over time, and make specific, possibly life threatening situations, easy to spot. The used glyph basically consists of three parts that represent circulation, respiration, and fluid

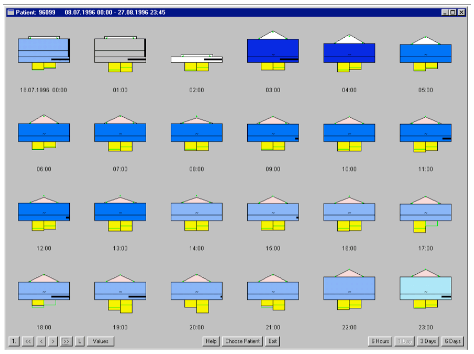


(a) Main screen.

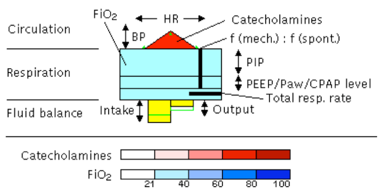


(b) Semantic zoom which adapts the type of representation depending on the available screen space.

Figure 8. Midgaard [3] – Visualization of intensive care data. Integrated visualization of quantitative time-series data, qualitative scales as well as qualitative/quantitative hybrids.



(a) VIE-VISU Interface – Each glyph represents a one hour period and 24 glyphs are combined in a single screen.



(b) Glyph Parts – The glyph basically consists of three parts that represent circulation, respiration, and fluid balance parameters (top to bottom. All in all, 15 different patient's parameters are combined to form this glyph.

Figure 9. VIE-VISU [11] – Combines different measurements, maintaining their relationships, shows their development over time, and make specific, possibly life threatening situations, easy to spot. Example shows neonatal patient's data.

balance parameters (top to bottom, see Figure 9(b)). All in all, 15 different patient's parameters are combined to form this glyph. Each glyph represents a one hour period and 24 glyphs are combined in a single screen (see Figure 9(a)). Different time frames can be selected and displayed (from 6 hours to 6 days).

2.1.4. Interactive Parallel Bar Charts (IPBC)

Interactive Parallel Bar Charts (IPBC) [6] have been designed to analyze hemodialysis data. It uses rows of histogram cuboids to represent the data in 3D (see Figure 10). In terms of interacting with the technique, several useful features, as, for example, the "tide mode" that allows for highlighting specific volumes and brushing of the representation, are included. Color mappings can be changed interactively and pattern matching can be performed based on examples not only for exact matches, but also with configurable tolerance levels. Moreover, the problem of occlusions in the 3D representation is addressed by providing the possibility to flatten occluding elements. All interactions are smoothly integrated by animated transitions when changing view parameters or interacting with the representation.

2.1.5. Gravi++

A visual method to analyze data gathered from patients via questionnaires is *Gravi++* [10]. Gravi++ was designed to find predictors during the treatment planning for anorectic girls. It utilizes human capabilities by positioning icons on the screen: icons representing patients and questions from the questionnaires. This is modeled with a spring-based system where every question is connected with every person by an (invisible) spring. Every person's icon position on the screen identifies how she answered the respective questions. This leads to the formation of clusters of persons who gave similar answers (see Figure 11). To visualize the changing values over time, Gravi++ uses animation whereas the position of each person's icon changes over time allowing to trace, compare, and analyze the changing values. Alternatively, the change over time can also be represented by traces.

2.2. Data Abstraction

In the previous section, we have presented a number of interactive techniques for visualizing and analyzing patients' data. In the course of studying and analyzing patients' data, users are mostly interested in the meaning that can be derived from the raw data, that is, the *information* that can be extracted. For example, the exact reading of the body temperature of a patient might not be as important as the fact whether the patient has "low fever" or "high fever". Additionally, the abstraction of raw-data to cognitively higher concepts yields data reduction of often huge amounts of data. This data reduction, in turn, eases visualization and analysis tasks. The objective of data abstraction in general is *"to create an abstraction that conveys key ideas while suppressing irrelevant details"* [31].

Examples for such systems used for temporal data abstraction (TDA) are *KNAVE II* and *VIE-VENT*. KNAVE II [23] is a tool that supports the visualization, summarizing, (intelligent) interpretation, explanation, and context-sensitive navigation of time-oriented raw clinical data sets and higher-level concepts abstracted from time-oriented data. *VIE-VENT* [15] addresses context-sensitive and expectation-guided temporal abstraction methods in a medical application domain. The developed methods incorporate knowledge about data points, data intervals, and expected qualitative trend patterns to arrive at unified qualitative descriptions. Smoothing and adjustment mechanisms are used to keep qualitative descriptions stable in case of shifting contexts or data oscillating near thresholds. For example, during intermittent positive pressure ventilation (IPPV),

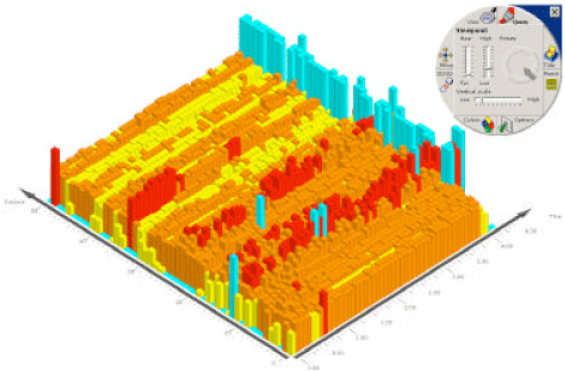


Figure 10. Interactive Parallel Bar Charts (IPBC) [6] – Rows of histogram cuboids for each series of measurements represent the data in 3D.

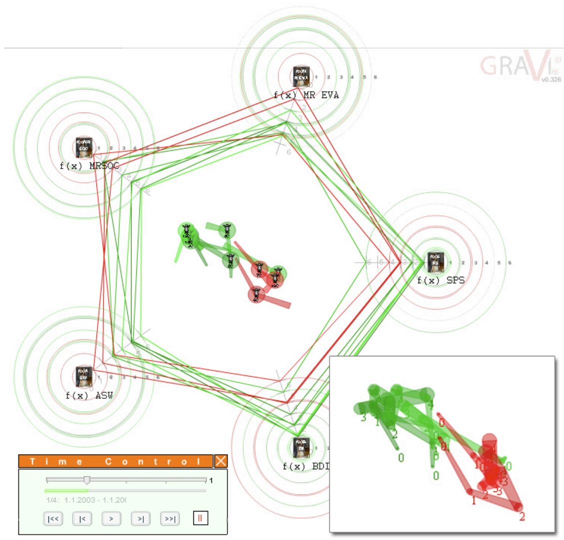


Figure 11. Gravi++ [10] – Patient's icons in the middle of the display are positioned according to a spring-based model relative to the surrounding question icons. The representation might be stepped through time manually or animated via the time control panel on the lower left. Furthermore, traces might be displayed that convey information about the evolution of values over time as shown in detail on the lower right.

the transformation of the quantitative value $P_{tc}CO_2 = 56mmHg$ results in a qualitative $P_{tc}CO_2$ value of "substantially above target range".[7] During intermittent mandatory ventilation (IMV), however, $56mmHg$ represents the "target value".

In the previous two sections we have shown how guidelines and patients' data can be visualized separately. In the next section we depict how these two views can be connected with in one.

[7] $P_{tc}CO_2$ = transcutaneous partial pressure of carbon dioxide

3. Visualizing Patient Data in Connection with Clinical Guidelines

Visualizing clinical guidelines in combination with patients' data is a challenging task, because heterogeneous and time-oriented data and information need to be visualized in an intuitive way. Usually, the visual representations of clinical guidelines are simply combined with text-based explanations of the course of the patient. However, very few contributions exist, which try to propose more sophisticated solutions. In the following we describe three representatives.

3.1. Tallis Tester

In Section 1.1.7 we explained the Arezzo and Tallis tools to model PROforma guidelines. One component of the Tallis Toolset is the Tallis Tester, which supports the enacting of guidelines. The Tallis Tester keeps track of which tasks need to be performed and provides information to external data and events regarding the current state of the execution.

3.2. CareVis

CareVis is an interactive visualization method to support the visualization of Asbru's plan execution and monitoring [2]. CareVis provides multiple simultaneous views to cover different aspects of a complex underlying data structure of treatment plans and patient's data.

Basically, CareVis divides the underlying data structure along the lines of logical structure and temporal aspects. Hence, CareVis provides a *Logical View* (compare *AsbruFlow* in Section 1.1.6) and a *Temporal View* along with a *QuickView panel*. These distinct views are presented simultaneously and divide the screen in the following manner (see Figure 12). The QuickView panel is located on top of the screen displaying the most important patient's 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-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).

3.3. Guideline Overview Tool - GOT

The main purpose of the *Guideline Overview Tool (GOT)* [1] is to provide a compact overview that is easy to read and comprehend in order to support physicians in executing and analyzing therapies with clinical guidelines (see Figure 13). GOT's overview has two main functions: (1) showing the actual state of a patient in relation to the assigned clinical guideline and (2) displaying several patients at one view in order to compare them.

In the next section, we illustrate how the different visualization methods facilitate the tasks in plan management.

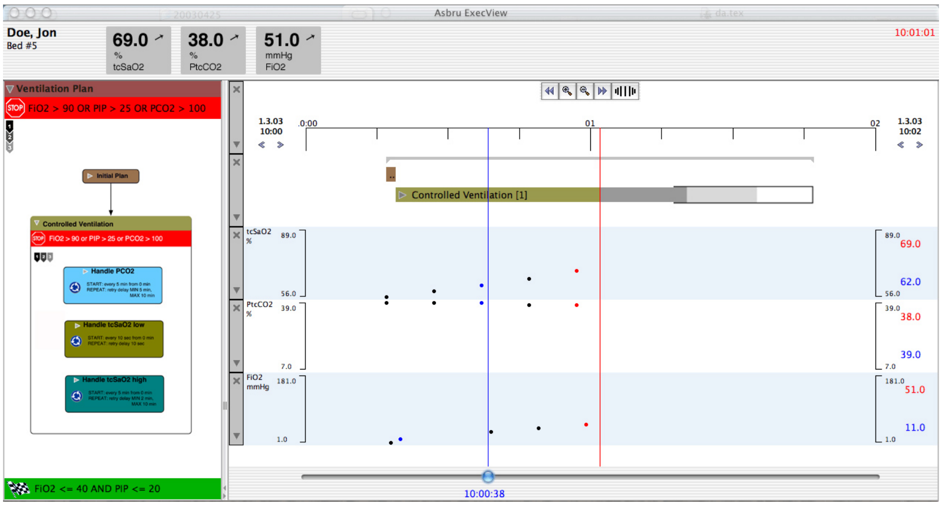


Figure 12. CareVis prototype [2] – Showing *Logical View* on the left-hand side, *Temporal View* on the right–hand side, and *QuickView panel* above.

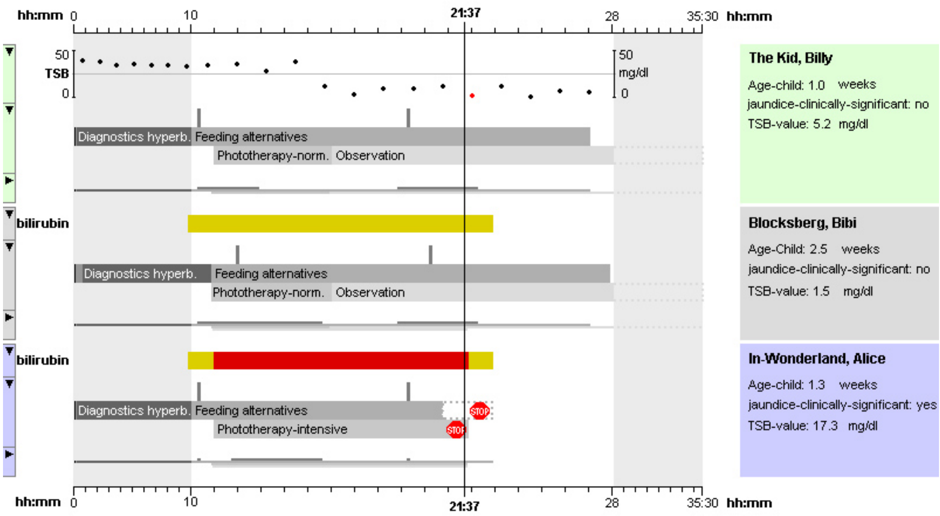


Figure 13. GOT [1] – It shows different patients treated with the same guideline for Hyperbilirubinemia.

4. Visual Supporting Tasks

The main tasks of plan management were discussed in the first section (compare Table 1). Afterwards we gave an overview about the different visualization methods. Table 2 sketches how these tasks can be best supported by the different visualization methods.

Table 2. Visualization Methods and Tools Used for Tasks in Plan Management.

Tasks mostly done at design time	Visualization methods and tools
Plan Generation	*only for visualizing the output*
Advanced Plan Editing	*Protégé, VisiGuide, AsbruView, SopoView, AsbruFlow, Tallis Composer, GLARE, GEM Cutter, DELT/A*
Domain-Specific Annotations	*Protégé, VisiGuide, AsbruView, SopoView, AsbruFlow, Tallis Composer, GLARE, GEM Cutter, DELT/A*
Plan Verification	*only for visualizing the output*
Plan Validation	*only for visualizing the output*
Plan-Scenario Testing	*Protégé, AsbruFlow, Tallis Tester, GLARE, CareVis, GOT*
Plan Visualization	*clinical algorithm maps, Nassi-Shneiderman diagrams, PERT charts, Gantt charts, and Petri nets, Protégé, VisiGuide, AsbruView, SopoView, AsbruFlow, Tallis Composer, GLARE, Tallis Tester, CareVis, GOT*
Tasks mostly done at execution time	**Visualization methods and tools**
Plan Selection	*clinical algorithm maps, Protégé, VisiGuide, AsbruView, SopoView, AsbruFlow, Tallis Tester, GLARE, CareVis*
Plan Adaptation	*clinical algorithm maps, Protégé, VisiGuide, AsbruView, SopoView, AsbruFlow, Tallis Composer, DELT/A, Tallis Tester, GLARE, CareVis, GOT*
Plan Execution	*AsbruFlow, Tallis Tester, GLARE, CareVis, GOT*
Plan Monitoring	*AsbruFlow, Tallis Tester, GLARE, CareVis, GOT*
Plan Modification / Alternatives	*Protégé, VisiGuide, AsbruView, SopoView, AsbruFlow, Tallis Composer, GLARE, GEM Cutter, DELT/A*
Plan Evaluation / Critiquing	*AsbruFlow, Tallis Tester, GLARE, CareVis, GOT*
Plan Visualization	*Protégé, GLARE, Tallis Tester, CareVis, GOT*
Data Visualization	*graphical summary of patient's status by Tufte/Powsner, Time Lines, LifeLines, Midgaard, VIE-VISU, IPBC, Gravi++, VIE-VENT, KNAVE II, Tallis Tester, CareVis, GOT*
Plan and Data Visualization	*Tallis Tester, CareVis, GOT*
Plan Rationale / History	*AsbruFlow, Tallis Tester, GLARE, CareVis, GOT*

4.1. Tasks During Design Phase

Plan Generation does not really ask for visualization support in its pure sense, because it tries to create a path of activities in an automatic, semi-automatic, or manual way. However, the process of plan generation and the output of this task can strongly be eased by visual methods, because it is quite difficult to communicate the different steps and the output to domain experts. The same holds for *Plan Verification and Validation*. *Advanced Plan Editing* and *Domain-Specific Annotations* can be supported in two ways: firstly, document-centric approaches (compare Section 1.2) can be used to add structured text and secondly, visual methods can be applied to author and communicate the added or needed parts and to illustrate changes. The other tasks can be supported by selected visualization methods as shown in Table 2.

4.2. Tasks During Execution Phase

Only a few visualization methods are available, which support both the *Plan and Data Visualization* during execution phase (compare the few examples in Section 3). As mentioned earlier in Section , *Plan Modification / Alternatives* is an outstanding task, because it handles maintaining of clinical guidelines and changes in the environment. Therefore, on the one hand, visualization methods are needed, which support adapting the guidelines. On the other hand, visualization methods should ease the monitoring of the patients' health course, the medical environment, and changes thereof in connection with applicable therapeutic actions. The remaining tasks during execution phase could be eased by appropriate methods for *Plan and Data Visualization*.

In summary, there are fewer visual contributions available to support the tasks during design time than during execution time and the visual methods needed to ease the tasks during execution time ask for more features than the other tasks. However, the different tasks can mutually benefit from visual methods of the others.

5. Research Agenda

In this book chapter we presented methods to visualize (1) clinical guidelines seen as plans or activities, (2) patients' data seen as multidimensional information space, and (3) patients' data in connection with clinical guidelines. Contributions of the first two categories are manifold. However, visualizing patients' data in connection with clinical guidelines is a challenging task and only a few approaches are currently available. Many of the visualization methods were developed within the framework of a particular guideline representation language. Therefore, the available visualization methods are mainly oriented towards the specific functionality of the guideline representation language (compare AsbruView, GLARE, and the Tallis Toolset).

According to our findings and various discussions with colleagues coming from different research fields and industry, we can formulate the following research directions:

Visualising the various dimensions of guideline-based care management. We presented different methods to visualize clinical guidelines, patients' data, and the connection thereof (as mentioned above). As we have shown, various approaches exist to visualize patients' data. However, more research needs to be done in the other two categories. On the one hand, visualizing clinical guidelines seen as plans or activities is too much oriented towards the particular features of the guideline representation languages, which asks for more research in the directions of the particular tasks in plan management. On the other hand, visualizing patients' data in connection with clinical guidelines is still an open, but challenging issue.

Designing a science or model of interactions. One very important element of the visual exploration process is interaction(s). Different contributions on various levels exist in the visualization and human-computer interaction community. However, there is no well-accepted science or model of interactions. The guideline-based community could contribute to develop such models for their particular tasks.

Supporting different users, tasks, and data. Visualizations should to be designed and developed according to the different users, their tasks, as well as data and information available. However, the presented visualization methods do not differentiate between these dimensions. Therefore, research should consider these aspects. For example, guideline developers need other visualizations than physicians who are debugging guidelines or real end-users of guidelines.

In-depth evaluation of the visualization methods. As also observed in other fields, the assessment of the usability and utility of the designed visualization methods is partly a neglected issue. However, to illustrate the benefits, a more in-depth evaluation of the visualization methods with particular consideration of the different tasks in plan management is needed.

Acknowledgements

This work is supported by "Fonds zur Förderung der wissenschaftlichen Forschung FWF" (Austrian Science Fund), grant L290-N04.

References

[1] Wolfgang Aigner. Guideline Overview Tool (GOT). Technical Report Asgaard-TR-2001-4, Vienna University of Technology, Institute of Software Technology and Interactive Systems, Austria, 2001.

[2] Wolfgang Aigner and Silvia Miksch. CareVis: Integrated Visualization of Computerized Protocols and Temporal Patient Data. *Artificial Intelligence in Medicine (AIIM)*, 37(3):203–218, May 2006.

[3] Ragnar Bade, Stefan Schlechtweg, and Silvia Miksch. Connecting Time-oriented Data and Information to a Coherent Interactive Visualization. In *Proc. of the 2004 Conf. on Human Factors in Computing Systems (CHI04)*, pages 105–112. ACM Press, 2004.

[4] Stuart Card, Jock Mackinlay, and Ben Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, San Francisco, 1998.

[5] Luca Chittaro. Information Visualization and its Application to Medicine. *Artificial Intelligence in Medicine (AIIM)*, 22(2):81–88, 2001.

[6] Luca Chittaro, Carlo Combi, and G. Trapasso. Visual Data Mining of Clinical Databases: An Application to the Hemodialytic Treatment based on 3D Interactive Bar Charts. In *Proceedings of VDM 2002: 2nd International Workshop on Visual Data Mining*, 2002.

[7] Paul A. de Clercq, Johannes A. Blom, Hendrikus H. M. Korsten, and Arie Hasman. Approaches for Creating Computer-interpretable Guidelines that Facilitate Decision Support. *Artificial Intelligence in Medicine (AIIM)*, 31(1):1–27, May 2004.

[8] John H. Gennari, Mark A. Musen, Ray W. Fergerson, William E. Grosso, Monica Crubézy, Henrik Eriksson, Natalya F. Noy, and Samson W. Tu. The Evolution of Protégé: An Environment for Knowledge-based Systems Development. *International Journal of Human Computer Studies*, 58(1):89–123, 2003.

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

[10] Klaus Hinum, Silvia Miksch, Wolfgang Aigner, Susanne Ohmann, Christian Popow, Margit Pohl, and Markus Rester. Gravi++: Interactive Information Visualization of Highly Structured Temporal Data. *Journal of Universal Computer Science, Special Issue on Visual Data Mining*, 11:1792–1805, 2005.

[11] Werner Horn, Christian Popow, and Lukas Unterasinger. Metaphor Graphics to Visualize ICU Data over Time. In *Proceedings of Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP-98), Workshop Notes of the ECAI-98 Workshop*, August 25 1998.

[12] Robert Kosara and Silvia Miksch. Metaphors of Movement: A Visualization and User Interface for Time-Oriented, Skeletal Plans. *Artificial Intelligence in Medicine, Special Issue: Information Visualization in Medicine*, 22(2):111–131, May 2001.

[13] Tze Yun Leong, Katharina Kaiser, and Silvia Miksch. Free and Open Source Enabling Technologies for Patient-Centric, Guideline-Based Clinical Decision Support: A Survey. *IMIA Yearbook of Medical Informatics 2007, Methods Inf Med*, 46(1):74–86, 2007.

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

[15] Silvia Miksch, Werner Horn, Christian Popow, and Franz Paky. Utilizing Temporal Data Abstraction for Data Validation and Therapy Planning for Artificially Ventilated Newborn Infants. *AI in Medicine*, 8(6):543–576, 1996.

[16] Mor Peleg, Samson W. Tu, Jonathan Bury, Paolo Ciccarese, John Fox, Robert A. Greenes, Richard Hall, Peter D. Johnson, Neill Jones, Anand Kumar, Silvia Miksch, Silvana Quaglini, Andreas Seyfang, Edward H. Shortliffe, and Mario Stefanelli. Comparing Computer-Interpretable Guideline Models: A Case-Study Approach. *Journal of the American Medical Informatics Association (JAMIA)*, 10(1):52–68, Jan-Feb 2003.

[17] Catherine Plaisant, Richard Mushlin, Aaron Snyder, Jia Li, Dan Heller, and Ben Shneiderman. LifeLines: Using Visualization to Enhance Navigation and Analysis of Patient Records. In C. G. Chute, editor, *Proceedings of the 1998 American Medical Informatic Association Annual Fall Symposium*, pages 76–80. AMIA, Bethesda, MD, November9–11 1998.

[18] Martha E. Pollack and John F. Horty. There's More to Life than Making Plans: Plan Management in Dynamic, Multi-agent Environements. *AI Magazine*, 20(4):71–84, 1999.

[19] Kristi-Anne Polvani, Abha Agrawal, Bryant Karras, Aniruddha Deshpande, and Richard Shiffman. GEM Cutter Manual. Technical report, Yale Center for Medical Informatics, New Haven, CT, 2000.

[20] Seth M. Powsner and Edward R. Tufte. Graphical Summary of Patient Status. *The Lancet*, 344:386–389, August 6, 1994.

[21] Marek Ruzicka and Vojtech Svátek. Mark-up Based Analysis of Narrative Guidelines with the Stepper Tool. In Katharina Kaiser, Silvia Miksch, and Samson W. Tu, editors, *Computer-based Support for Clinical Guidelines and Protocols. Proceedings of the Symposium on Computerized Guidelines and Protocols (CGP 2004)*, volume 101 of *Studies in Health Technology and Informatics*, pages 132–136, Prague, Czech Republic, April 2004. IOS Press.

[22] Doug Schaffer, Zhengping Zuo, Saul Greenberg, Lyn Bartram, John Dill, Shelli Dubs, and Mark Roseman. Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods. *ACM Transactions on Computer-Human Interaction*, 3(2):162–188, 1996.

[23] Yuval Shahar, Dina Goren-Bar, David Boaz, and Gil Tahan. Distributed, Intelligent, Interactive Visualization and Exploration of Time-Oriented Clinical Data and Their Abstractions. *Artificial Intelligence in Medicine*, 38:115–135, October 2006.

[24] Yuval Shahar, Silvia Miksch, and Peter Johnson. The Asgaard Project: A Task-Specific Framework for the Application and Critiquing of Time-Oriented Clinical Guidelines. *Artificial Intelligence in Medicine*, 14:29–51, Sept. 1998.

[25] Yuval Shahar, Ohad Young, Erez Shalom, Alon Mayaffit, Robert Moskovitch, Alon Hessing, and Maya Galperin. DEGEL: A Hybrid, Multiple-Ontology Framework for Specification and Retrieval of Clinical Guidelines. In Michel Dojat, Elpida Keravnou, and Pedro Barahona, editors, *Proceedings of the 9th Conference on Artificial Intelligence in Medicine in Europe (AIME 2003)*, volume 2780 of *LNAI*, pages 122–131, Protaras, Cyprus, 2003. Springer Verlag.

[26] Richard N. Shiffman, Bryant T. Karras, Abha Agrawal, Roland Chen, Luis Marenco, and Sujai Nath. GEM: A Proposal for a More Comprehensive Guideline Document Model Using XML. *Journal of the American Medical Informatics Association (JAMIA)*, 7(5):488–498, 2000.

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

[28] Rory Steele and John Fox. Tallis PROforma Primer – Introduction to PROforma Language and Software with Worked Examples. Technical report, Advanced Computation Laboratory, Cancer Research, London, UK, 2002.

[29] David R. Sutton and John Fox. The Syntax and Semantics of the PROforma Guideline Modeling Language. *Journal of the American Medical Informatics Association (JAMIA)*, 10(5):433–443, 2003.

[30] Paolo Terenziani, Stefiana Montani, A. Bottrighi, M. Torchio, G. Molino, and G. Correndo. The GLARE Approach to Clinical Guidelines: Main Features. *Stud Health Technol Inform*, 101:162–166, 2004.

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

[32] Eduard R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, 1983.

[33] Peter Votruba, Silvia Miksch, and Robert Kosara. Facilitating Knowledge Maintenance of Clinical Guidelines and Protocols. In Marius Fieschi, Enrico Coiera, and Yu-Chuan Jack Li, editors, *Proceedings from the Medinfo 2004 World Congress on Medical Informatics*, pages 57–61. AMIA, IOS Press, 2004.

[34] Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, San Francisco, CA, 2004.