Communicating the Logic of a Treatment Plan Formulated in Asbru to Domain Experts

Wolfgang AIGNER and Silvia MIKSCH
Institute of Software Technology & Interactive Systems
Vienna University of Technology
Favoritenstraße 9-11/188
A-1040 Vienna, Austria, Europe
{aigner, silvia}@asgaard.tuwien.ac.at

Abstract.
This paper presents an interactive visualization for medical treatment plans that are formulated in the plan representation language Asbru.

So far, most attention of the protocol-based care community was focused towards formal guideline representation and authoring partly supported by graphical tools. The intention of this work is to go the opposite way and communicate the logic of a computerized treatment plan to physicians, nursing-, and other medical personnel visually.

The visualization is based on the idea of *flow-chart algorithms* widely used in medical education and practice. This concept has been extended in order to cope with the powerful and expressive guideline representation language Asbru. Furthermore, a number of interactive navigational and overview extensions are used to intuitively support the understanding of the logic of plans.

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 as described in this document.

1 Introduction

Various researchers have put a great deal of work in their efforts on supporting protocol-based care by the means of information technology. The high-level goals of this efforts are to support planning, executing, and analyzing treatment plans[1] to increase the quality of care.

Most of the work has been dedicated to the extremely difficult task of capturing all aspects of a medical treatment plan into a guideline representation language. The real world domain medicine incorporates a series of complex aspects like time constraints, temporal uncertainties, intentions, plan conditions, and so forth, information systems have to deal with. This task of modeling medical knowledge and guidelines has been solved by several approaches as the *Asgaard project* with its guideline representation language *Asbru* [17, 29].

The next step is to make use of this formalized medical knowledge by executing plans, monitoring data, actions, and plans, data abstraction, and many other kinds of (semi-) automatic knowledge & information processing.

But all that is only one side of the story. As important as the task of feeding real world information into a computer system in a structured and meaningful way and processing it, is

[1] Throughout this paper, the expressions *clinical guideline*, *guideline*, *treatment plan*, *protocol*, and *plan* will be used interchangeably.

presenting and communicating this information to human domain experts, in our case physicians, nursing-, and other medical personnel. This presentation and communication has to be done in a clear, simple, and comprehensible way, preferably familiar to the end users in order to keep the learning effort as low as possible.

This work is aimed towards visualizing the logic of a treatment plan (plan composition, execution sequence, control structures, annotations, ...). We left out the important aspect time in this representation in the first place because incorporating this additional parameter would lead to a too complex visualization not familiar to domain experts. The parameter time in relation to plans is visualized in a separate, coupled view described in [1].

The following section introduces the main features of the guideline representation language *Asbru*. Section 3 contains a compilation and assessment of related work and following that, we present the user study we conducted along with its results to supplement the starting point for our development. Our solution of an interactive visualization environment is presented and discussed in Section 5. Information about the implemented prototype and its evaluation is given in the following section. Finally, we sum up our findings and provide an outlook onto future work in Section 7.

2 The Guideline Representation Language Asbru

Asbru is a time-oriented, intention-based, skeletal plan-specification representation language that is used in the *Asgaard* Project[2] to represent clinical guidelines and protocols in XML. *Asbru* can be used to express clinical protocols as skeletal plans [9] that can be instantiated for every patient (for an example see Fig. 1). It was designed specific to the set of plan-management tasks [16]. *Asbru* enables the designer to represent both the prescribed actions of a skeletal plan and the knowledge roles required by the various problem-solving methods performing the intertwined supporting subtasks. 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 plans or some plans in parallel, all plans or some plans in a particular order or unordered, or periodically;
- particular conditions are defined to monitor the plan's execution; and
- explicit intentions and preferences can be stated for each plan separately.

We will explain the structure and concepts used in *Asbru* in more detail in Section 5.

[2] 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`).

2.1 Example

Figure 1 shows parts of an *Asbru* plan for artificial ventilation of newborn infants. The guideline is represented in XML and contains domain definitions and a set of plans. The *ventilation plan* consists of conditions and the plan body including a sequential execution of the *initial plan* and *controlled ventilation plan*.

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

2.2 Basic Visualization Requirements

Visualizing the logic of *Asbru* plans imposes four fundamental problem characteristics on the representation that have to be considered:

- logical sequences
- hierarchical decomposition
- flexible execution order (sequential, parallel, unordered, any-order)
- state characteristics of conditions

Our research for related work in medical treatment planning, information visualization, medicine, and commercial medical software products was grounded on looking for graphical representations that are able to visualize the listed characteristics. The results of this research are presented in the following section.

3 Related Work

3.1 Medical Treatment Planning

Flow-chart Algorithms. The most widely used visual representation of clinical guidelines are so-called *flow-chart algorithms* also known as *clinical algorithm maps* [11]. 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* [31]:

"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." [31] The proposed standard includes a small number of different symbols and some rules on how to use them (see Fig. 2). 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. Furthermore, flow-charts cannot be used to represent concurrent tasks or the complex conditions used in *Asbru*. Clinical algorithm maps were intended to be used on paper

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plan-library SYSTEM "asbru_7_3.dtd">
<plan-library>
    <domain-defs>
        <domain name="controlled_ventilation_domain">
            ...
        </domain>
    </domain-defs>
    <plans>
        <plan-group>
            <plan name="ventilation_plan">
                <intentions> ...  </intentions>
                <conditions>
                    <complete-condition>
                        <constraint-combination type="and">
                            <parameter-proposition parameter-name="FiO2">
                                <value-description type="less-or-equal">
                                    <numerical-constant value="40"/>
                                </value-description>
                            ...
                        </constraint-combination>
                    </complete-condition>
                    <abort-condition>
                        <constraint-combination type="or">
                            <parameter-proposition parameter-name="FiO2">
                                <value-description type="greater-than">
                                    <numerical-constant value="90"/>
                                </value-description>
                            ...
                        </constraint-combination>
                    </abort-condition>
                </conditions>
                <plan-body>
                    <subplans type="sequentially">
                        ...
                        <plan-activation>
                            <plan-schema name="initial_plan"/>
                        </plan-activation>
                        <plan-activation>
                            <plan-schema name="controlled_ventilation_plan"/>
                        </plan-activation>
                    </subplans>
                </plan-body>
            </plan>
            ...
            <plan name="controlled_ventilation_plan">
                <plan-body>
                    <subplans type="parallel">
                        ...
                        <plan-activation>
                            <plan-schema name="handle_PCO2_plan"/>
                        </plan-activation>
                        <plan-activation>
                            <plan-schema name="handle_tcSaO2_low_plan"/>
                        </plan-activation>
                        <plan-activation>
                            <plan-schema name="handle_tcSaO2_high_plan"/>
                        </plan-activation>
                    </subplans>
                </plan-body>
            </plan>
            ...
        </plan-group>
    </plans>
</plan-library>
```

Figure 1: An example of Asbru 7.3 code: Parts of a clinical treatment plan for artificial ventilation of newborn infants.

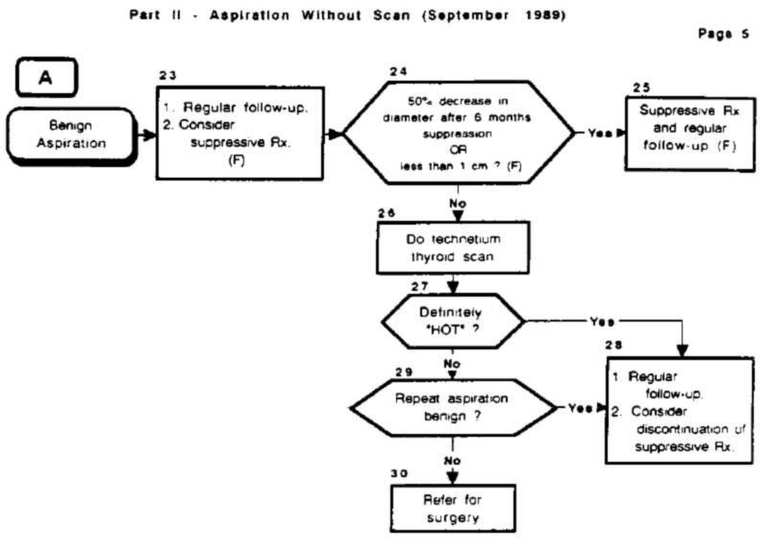


Figure 2: Clinical algorithm map [31].

and have never been enriched by computer support as for example navigation or versatile annotation possibilities.

Visualizing Logical Sequences. Other possibilities to visualize logical sequences away from flow-charts are *Structograms* [18], *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 in their basic forms.

Visualizing Hierarchical Data. The most popular technique for visualizing hierarchical data are *Trees*. A further technique for that matter are *Treemaps* [12] introducing an additional dimension by proportional space assignment. But these 2D techniques have no notion to depict logical sequences, concurrency, or states.

AsbruView [13–15] 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, respectively views: One captures the topology of plans, whereas the second one shows the temporal dimension of plans. The intention of *AsbruView* is to support plan creation and manipulation but neither to communicate the logic of an *Asbru* plan during execution or analysis of a plan nor for educational reasons as our work is aiming towards.

Other Scientific Projects. Further scientific work [4, 25, 32] on visual representations focused on patient data over time or plan execution over time. Other research projects dealing with protocol-based care include *GLARE* [10], *GUIDE* [26], *Protégé* [30], *GLIF* [20], *PROforma* [8], and *GASTON* [6]. (A comprehensive overview of related protocol-based care projects can be found at [21] and [34].)

Only some of the available projects dealing with protocol-based care provide graphical tools at all. The just listed ones include such graphical tools, but most of them only for authoring plans. They use a flowchart- or workflow-like presentation depicting the elements used in their formal representation. A more detailed examination of the quoted projects can be found in [1].

These tools make authoring clinical protocols easier especially for non computer scientists but they use a not very familiar graphical representation and mix state and flow-chart characteristics within a single diagram. Thus, understanding this representation and using it for plan authoring requires a considerable amount of learning effort.

Authoring clinical guidelines and communicating complete protocols to domain experts are two rather different tasks with different goals. For guideline authoring, first of all one can assume a more thorough knowledge of the user in the computer domain and a higher threshold towards acceptable learning effort is likely. In terms of aid for achieving the goal of a completely specified guideline, the user has to have an overview of what elements are available for constructing it as well as means for data input have to be provided. Moreover, mechanisms for preventing mistakes in the authoring process should be present. This is in contrast to the goal of communicating the logic of a treatment plan where the presentation of and navigation within guidelines is paramount along with providing easy access to linked information and in depth explanations.

3.2 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 for aiding treatment planning.

We examined a number of non-administrative software products that use graphical representations: *IntelliVue* [22] (Philips Medical Systems) formerly known as *CareVue* (Hewlett-Packard), *Chart+* [23] (Picis), *Visual Care* [24] (Picis), *QCare* [5] (Critical Care Company), *Coronary Risk Profile (CRP)* (Wellsource) [33], *SOAPware* (Docs, Inc.) [7] and *Clicks Medical Information System* [27] (Roshtov Software Ind. Ltd.). We investigated medical software products having graphical representation 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.

All of the examined products are rather data-centric and the most popular form of data representation is using tables where numerical respectively textual data is organized in spreadsheets. None of the listed products offered a way of visualizing treatment planning logic at all.

We think that besides this research of related work on a scientific basis and examining commercial products it is absolutely necessary to involve end-users from the very beginning because only this measure can ensure the incorporation of the users' valuable experience, knowledge, and desires, thus increasing quality and acceptance dramatically. This user-centric development was begun by carrying out a user study as described in the following section.

4 User Study to Acquire Physicians' Needs

A step of major importance for requirement analysis in our development process was to conduct a user study [19] with eight physicians to gain deeper insights into the medical domain, work practices, application of guidelines in daily work, users' needs, expectations, and imaginations.

Most of the interviewed physicians work at different departments for critically ill patients at the General Hospital of Vienna (AKH Wien). The AKH Wien is a university clinic which means that employed physicians also work scientifically. Conducting an interview took on average about 45 minutes and lead to interesting, but not too surprising results and insights. (Detailed results and interview guidelines can be found in [1].)

Fundamental issues for the interviewed physicians were rather practical ones. Most importantly the system has to save time – no one would use a system if it would take more time as working without it. Another major issue is that learning effort for using the system has to be minimal. The system should be intuitive, simple, and clearly structured without complex menu structures or functions.

It became apparent that clinical guidelines are generally depicted by a special form of flow-charts as proposed in [31] and are widely known. Relatively unknown to our interview partners were Structograms, and Glyphs as for example Chernhoff Faces.

When summarizing and evaluating the results of our user study the following desired 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.

5 Visualizing the Logic of an Asbru Plan

As our research showed, there are no graphical methods suiting our needs available for communicating the logic of computerized medical treatment plans to domain experts. Related projects and information visualization methods do not offer applicable concepts to represent *Asbru* plans. Weighing up the results of the conducted research in combination with the key aspects from the end users' point of view delivered by our user study lead to the decision of using *clinical algorithm maps* as basis for our visualization. We extended this concept and added a number of interactive features to enable intuitive access to the logic of treatment plans formulated in *Asbru*:

5.1 Asbru Prerequisites

In the following, a simplified description of the structure of *Asbru*[3] plans is extracted:

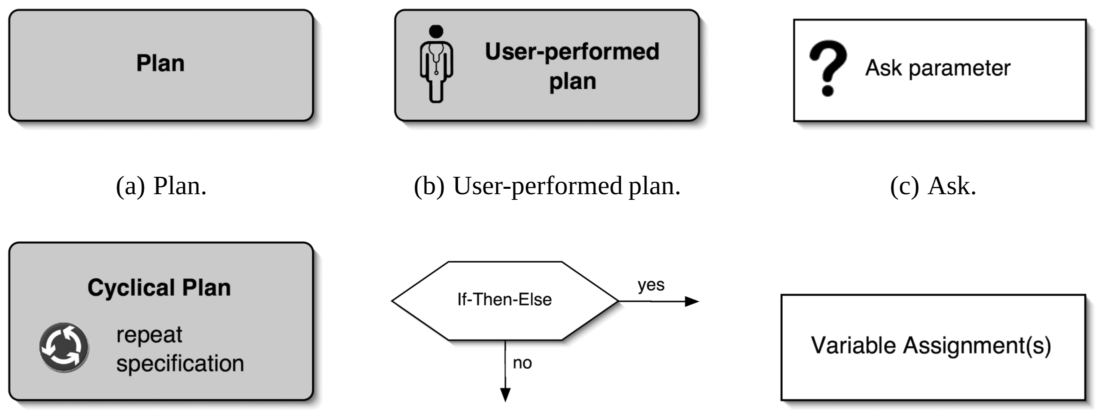
- An *Asbru* plan may contain the following conditions:
  - **filter precondition:** Only if this condition evaluates to *true*, the plan gets executed.
  - **abort condition:** If this condition evaluates to *true*, the whole plan aborts. This condition is valid and checked all throughout plan execution and is getting forwarded to subplans.
  - **complete condition:** If and only if the elements within the plan body are completed as intended and the complete condition evaluates to *true*, the plan can complete successfully.

[3] This work is using a subset of Asbru called *Asbru Light+*.

- An *Asbru* plan has a plan-body containing *single-steps* that are executed in one of the following *execution sequences*:
  - **sequentially:** The contained steps are processed one after the other in the given order.
  - **parallel:** All steps get initialized at the beginning and are processed in parallel.
  - **any-order:** Same as *sequentially* except that the execution order is arbitrary.
  - **unordered:** The contained steps can be executed in any arbitrary way.
- A *single-step* is one of the following:
  - **Variable assignment:** An expression is getting assigned to a plan variable.
  - **If-Then-Else:** If the condition of the construct evaluates to *true*, the *then-branch* otherwise the *else-branch* gets executed if present.
  - **Ask:** An external, typically user entered value is assigned to the specified parameter.
  - **Plan activation:** The specified plan gets activated.

5.2 Plan Step Elements

The used visual plan step elements are based on the elements of the flowchart-like representation of the *Committee on Standardization of Clinical Algorithms* [31].



(a) Plan. (b) User-performed plan. (c) Ask.

(d) Cyclical plan. (e) If-Then-Else. (f) Variable assignment.

Figure 3: Plan step elements.

We added one *plan element* and a number of symbols for depicting parts of the *Asbru* language that could not be visualized by the elements of the proposal (see Fig. 3 for an overview):

- **Plans** respectively **plan activations** are represented by a rounded rectangle filled with the plan color[4] (see Fig. 3(a)). In case of being a **cyclical plan**, an additional roundabout

[4] A distinct color is assigned to each plan, making it easier to distinguish plans from other elements and helping to recognize them in other parts of the representation.

icon as well as the repeat specification in textual form are presented within the rectangle (see Fig. 3(d)). Furthermore, a physician icon appears within the element if the plan is **user performed** (see Fig. 3(b)).

- **Variable assignments** are represented by a rectangle containing the assignment textually (see Fig. 3(f)).
- **If-Then-Else** constructs are shown as hexagons having the condition displayed textually (see Fig. 3(e)). The *then-branch* of the construct is always connected via an arrow originating at the right top of the element, and the *else-branch* via an arrow originating at the bottom of the element. The branches are labelled by the word "yes" (*then-branch*) respectively "no" (*else-branch*) right next to their connecting arrow lines.
- **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. 3(c)).

5.3 Anatomy of a Plan

Using the elements just presented, we are able to visualize the single steps within the plan body of an *Asbru* plan. For depicting the conditions and the execution order of the plan steps, an enclosing frame was created, containing the following parts (see Fig. 4).

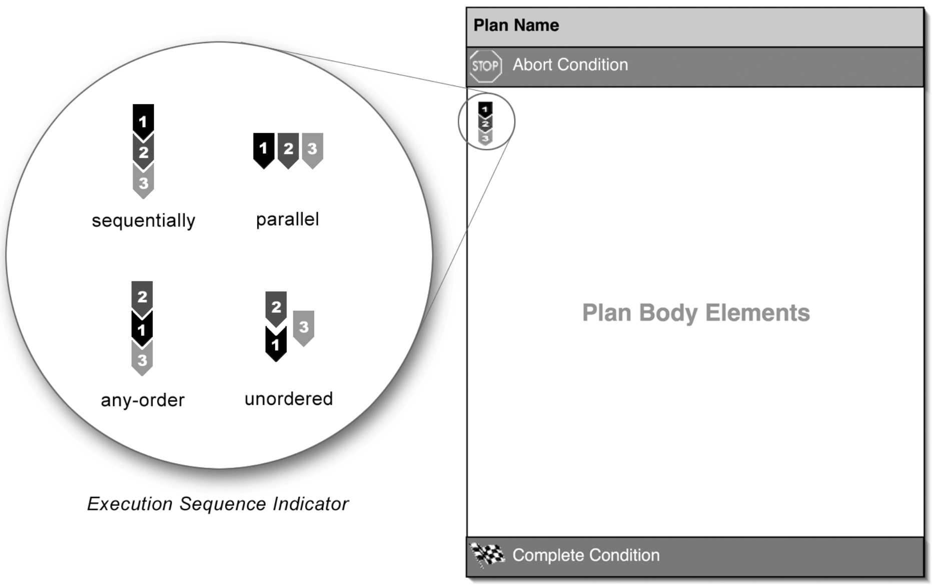


Figure 4: Basic structure and execution sequence symbols.

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. This condition has the following semantics: If the condition evaluates to *true*, the current plan gets aborted. Furthermore, this condition is valid 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 *checkered finish flag icon* at its left and contains the *complete condition* textually. The semantics of this condition is: If and only if this condition evaluates to *true*, the plan can complete successfully.

The biggest part of the representation is dedicated to the *plan body* of the depicted plan along with an icon on top showing the execution order of the elements enclosed. The **execution sequence indicator** has four possible symbols (see Fig. 4).

The rest of the plan body area contains plan elements as described in the last section. If the execution order of the elements is *sequentially*, the elements are additionally connected by arrows.

Note: The **filter precondition** is not represented by a special element but by using an *If-Then-Else* element prior to the related plan element.

5.4 Navigation & Interaction

Regardless of the fact that the static form of the visualization as described so far contains a lot of information and may also be useful in a printed form, adding interactive features increases the user experience much more.

One used element for that purpose not mentioned so far is the **small gray triangle** at plan elements and plan titles (see Fig. 5). This triangle indicates if an element has subelements (triangle pointing to the right) and if the subelements are currently expanded (triangle pointing to the bottom). In case an element has no subelements, no triangle is shown at all. By clicking a triangle pointing to the right, the element is getting expanded, which means navigating down in the hierarchy. When clicking a triangle pointing to the bottom, the element is getting collapsed, which means navigating up in the hierarchy. The use of those triangles is intuitive and based on their application in file system viewers as for example the *Finder* of the Macintosh™ system.

Furthermore, the elements of the representation can be dragged and resized in case the applied automatic layout is not delivering the desired results.

5.5 Annotations

Annotations and notes are a vital part of graphical representations for clinical guidelines [11, 31]. These annotations may include references to literature, web links, precise definition of terms, parameter descriptions, clarifications for the rationale of decisions, and more. We present this kind of information as “Tool Tips” when the mouse is hovering over the related part of the graphical representation or as small additional windows triggered by clicking in case Tool Tips are not suitable to represent certain chunks of information (ie. graphics, web links, long documents).

5.6 Focus + Context

Losing track of the actual position within a plan is quite easy when just using the visualization presented so far.

The first utility overcoming this problem is the *Overview + Detail display*. It shows a small tree-like representation of the whole plan, marking the current view position (see Fig. 5, right column). This *Overview display* is only shown on demand (triggered by the user) for not overloading or cluttering the screen.

The second utility avoiding to get lost within a plan is the *Fisheye display* (see Fig. 5, left column) whereas the current (sub)plan represents the focus which is displayed in full detail.

The surrounding (context) elements are shrunk and displayed with less detail. In contrast to the *Overview + Detail display* where only positional information is shown, surrounding context information is presented without gaps in more detail. Furthermore, smooth, animated transitions are used for fisheye navigation in order to not confuse the user when changing the focus.

In principle, *Asbru* plans can be seen as *hierarchically clustered networks*. Schaffer et al. examined visualization techniques for that kind of systems [28] and show that the *Fisheye display* is particularly useful but for certain purposes (i.e. examining a specific problem within a selected node), *full zoom* is more appropriate. Therefore, we use a button for toggling the *Fisheye* vs. *Full zoom* display.

5.7 Example

Figure 5 shows examples of our graphical representation. It depicts the *Asbru* plan for artificial ventilation of newborn infants as presented in XML in Fig. 1. The left column of figures shows a full navigational sequence when using the *Fisheye display*: Ventilation Plan (top plan level) $\rightarrow$ Controlled Ventilation $\rightarrow$ Handle tcSaO2 low. In the right column the same sequence is shown when *Overview + Detail display* is used.

5.8 Design Evaluation

When having completed the first “release” version of the conceptual design, we conducted an evaluation session for getting early feedback regarding our design. This early evaluation process was very valuable and reduced the risk of investing time and effort for might going in the wrong direction.

The evaluation was done by two experts: one person is a visualization expert having experience with medical software development and the other one is a physician (medical expert) having visualization knowledge.

The result of the evaluation was very positive, validated our concept, and showed that we were working in the right direction. Only some minor issues of the design were objected which led to an improvement of the design.

5.9 Discussion

The flowchart-like representation of so-called *clinical algorithms* [31] is well known among physicians, because it is used frequently in literature and is part of the education of physicians as our user study proved.

Asbru is too powerful to be translated completely into a flow-chart representation. The main difficulty in that sense is the state machine characteristic regarding plan conditions. Therefore, the most accurate visualization for *Asbru* plans would probably be State Transition Diagrams. But this type of visualization is not well known, requires relatively high learning effort and might not be accepted by physicians.

Furthermore, our user study showed that minimal learning effort and ease of understanding are essential and most important, given that the tool should not be limited to specialists or academic purposes only.

Based on these arguments we decided to use a flowchart-like representation. We are fully aware that the used visualization is not accurately representing how an *Asbru* plan is going to be executed. But we think that the mental model we are trying to create by this visualization is close enough to the actual execution model being at the same time familiar and easy to

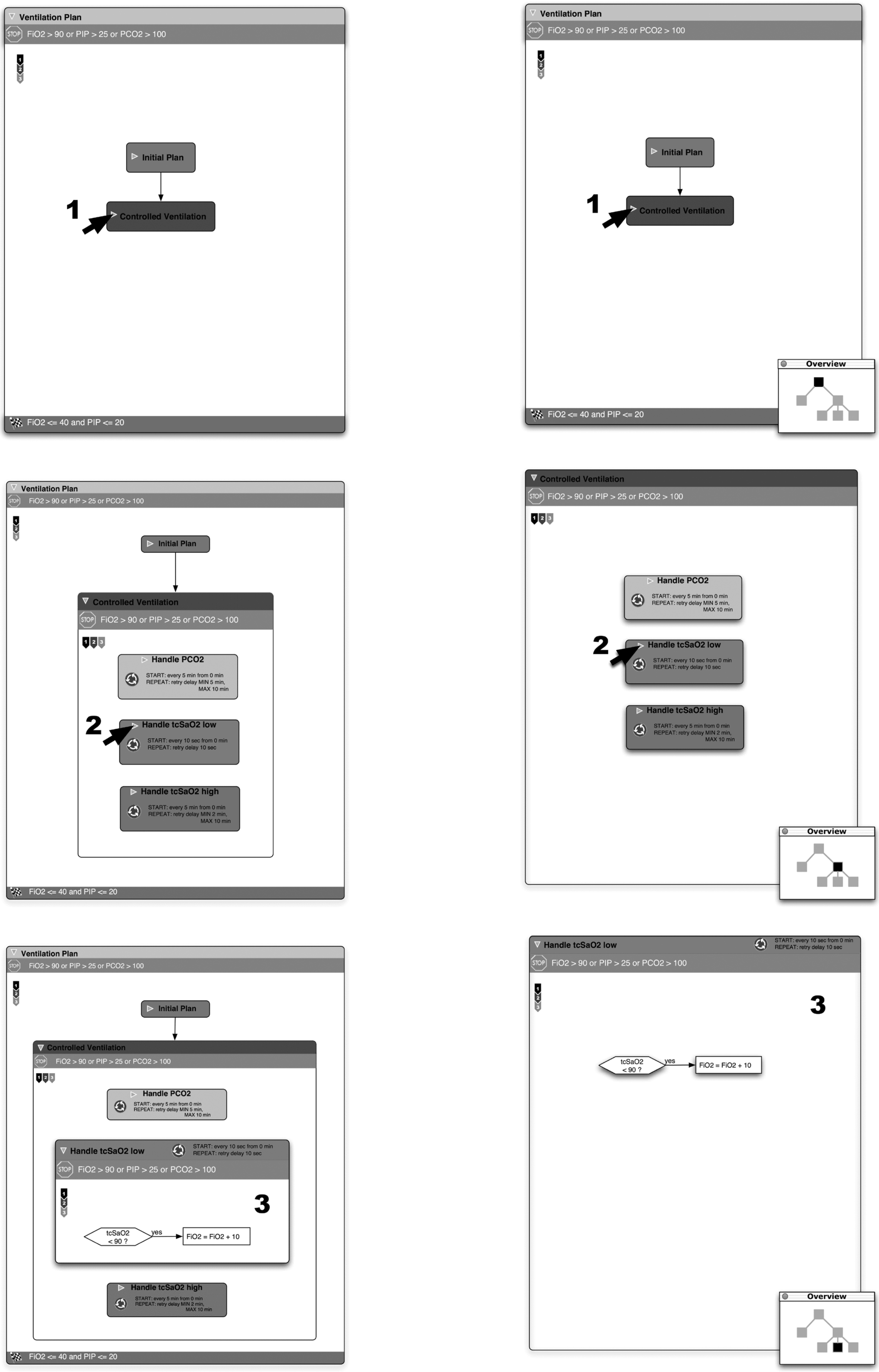


Figure 5: Visualization of the Asbru plan for artificial ventilation of newborn infants (see Fig. 1) – Fisheye Navigation (left column) vs. Overview+Detail Navigation (right column).

understand. An absolutely accurate representation would require a much more complicated and cluttered visualization but only show subtle differences in the used model.

6 Prototype

In order to proof our concept and give as well as get a better impression especially of interaction issues, we implemented a Java prototype. For displaying the flowchart-like part of our representation to depict plan step elements, we use the graph drawing framework **JGraph** [2, 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*.

6.1 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 (see Section 4) 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 generally 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. Detailed information about the evaluation process and its results can be found in [1].

7 Conclusion and Future Work

That visualizing the logic of clinical guidelines is useful to support understanding and exploration of protocols has already been proposed and proved years ago [11, 31]. *Flow-chart algorithms* 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 for our visual representation. But it cannot be applied directly to represent Asbru plans because it does not provide 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 order indicator, and an enclosing frame containing the plan conditions. We have examined and proven the usefulness of our approach performing a 3-step evaluation process including user study, design evaluation, and prototype evaluation.

The use of software in contrast to paper allows us to support the process of exploring and understanding treatment plans at a higher level. It enables a meaningful navigation, providing annotations on demand for not overwhelming the viewer, and keeping orientation by using Focus + Context techniques, thus increasing the flexibility in working with treatment plans.

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 thoroughly. This increases the quality of design, the user acceptance, and serves as an indicator of the maturity of development. We fulfilled the fundamental user requirements as listed in Section 4 by using a well known graphical representation as basis and introducing 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 including annotations and linked documents in a structured way. This is in contrast to working with paper-based treatment protocols that are often a mix of text, tables, and graphics, scattered over various pages, making it hard to keep an overview and conceive the logic of a guideline.

Some more effort has to be put into actually implementing the full set of introduced design concepts. The most important measure for that matter is to directly abstract the visual representation from Asbru plan files. Furthermore, Focus + Context techniques have to be implemented and rich annotation display possibilities should be integrated. A better layout algorithm for plan step elements has to be found as well including a smart aggregation of nodes if appropriate.

Besides that, the software environment should be enriched by smart lookup of plans available on the system, within a network, or even over the internet.

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

[3] G. Alder. The Home Page of JGraph, 2002. http://jgraph.sourceforge.net.

[4] 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.

[5] Critical Care Company. QCare, Nov. 2002. http://www.c3.be.

[6] P. A. de Clercq, A. Hasman, J. A. Blom, and H. H. M. Korsten. Design and implementation of a framework to support the development of clinical guidelines. *International Journal of Medical Informatics*, 64(2–3):285–318, December 2001.

[7] Docs Inc. SOAPware, Nov. 2002. http://www.docs.com/ Products/Modules/onlinedemo.htm.

[8] J. Fox and R. Thomson. Decision Support and Disease Management: A Logic Engineering Approach. *IEEE Transactions on Information Technology in Biomedicine*, 2(4):217–228, 1998.

[9] P. E. Friedland and Y. Iwasaki. The Concept and Implementaion of Skeletal Plans. *Journal of Automated Reasoning*, 1(2):161–208, 1985.

[10] A. Guarnero, M. Marzuoli, G. Molino, P. Terenziani, M. Torchio, and K. Vanni. Contextual and Temporal Clinical Guidelines. In *Proceedings AMIA Symposium*, pages 683–7, 1998.

[11] D. C. Hadorn. Use of Algorithms in Clinical Practice Guideline Development: Methodology Perspectives. *AHCPR Pub.*, (No. 95-0009):93–104, Jan. 1995.

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

[13] R. Kosara. Metaphors of Movement — A User Interface for Manipualting Time-Oriented, Skeletal Plans. Master's thesis, Vienna University of Technology, Institute of Software Technology and Interactive Systems, Vienna, Austria, May 1999.

[14] 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.

[15] R. Kosara and S. Miksch. Visualizing Complex Notions of Time. In J. Roberts, editor, *Proceedings of the Conference on Medical Informatics (MedInfo 2001)*, pages 211–215, 2001.

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

[17] S. Miksch, Y. Shahar, W. Horn, C. Popow, F. Paky, and P. Johnson. Time-Oriented Skeletal Plans: Support to Design and Execution. In *Fourth European Conference on Planning (ECP'97)*. Springer, September 24–26 1997.

[18] I. Nassi and B. Shneiderman. Flowchart Techniques for Structure Programming. *SIGPLAN Notices*, 8(8):12–26, 1973.

[19] J. Nielsen. *Usability Engineering*. Academic Press, 1993.

[20] M. Peleg, A. A. Boxwala, O. Ogunyemi, and et al. GLIF3: The Evolution of a Guideline Representation Format. In *Proc. AMIA Annual Symposium*, 2000.

[21] 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.

[22] Philips Medical Systems. IntelliVue, Nov. 2002. http://www.medical.philips.com/main/products/patient_monitoring/products /intellivue/index.html.

[23] Picis. Chart+ for Critical Care, Nov. 2002. http://www.picis.com/html/products/module_chart%2Bcritcare.html.

[24] Picis. Visual Care, Nov. 2002. http://www.picis.com/ html/products/module_visualcare.html.

[25] 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.

[26] S. Quaglini, M. Stefanelli, G. Lanzola, V. Caporusso, and S. Panzarasa. Flexible guideline-based patient careflow systems. *Artificial Intelligence in Medicine*, 22(1):65–80, 2001.

[27] Roshtov Software Ind. Ltd. Clicks Medical Information System, Nov. 2002. http://www.roshtov.com.

[28] D. Schaffer, Z. Zuo, S. Greenberg, L. Bartram, J. Dill, S. Dubs, and M. Roseman. Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods. *ACM Transactions on Computer-Human Interaction*, 3(2):162–188, 1996.

[29] A. Seyfang, R. Kosara, and S. Miksch. Asbru 7.3 Reference Manual. Technical Report Asgaard-TR-2002-1, Vienna University of Technology, Institut of Software Technology & Interactive Systems, Vienna, Austria, Europe, 2002.

[30] R. D. Shankar, S. W. Tu, and M. A. Musen. Use of Protégé-2000 to Encode Clinical Guidelines. In *Proc. AMIA Annual Symposium*, 2002.

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

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

[33] Wellsource. Coronary Risk Profile (CRP), Nov. 2002. http://www.wellsource.com/products/crp/crpopen.htm.

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