Visualizing Complex Process Hierarchies during the Modeling Process

Andreas Seyfang, Katharina Kaiser, Theresia Gschwandtner, and Silvia Miksch

Institute of Software Technology & Interactive Systems
Vienna University of Technology, Vienna, Austria
{seyfang,kaiser,gschwandtner,miksch}@ifs.tuwien.ac.at

Abstract. Clinical practice guidelines are documents that include recommendations describing appropriate care for the management of patients with a specific clinical condition, such as diabetes or chronic heart failure. Several representation languages exist to model these documents in a computer-interpretable and -executable form with the intention of integrating them into clinical information systems. *Asbru* is one of these representation languages that is able to model the complex hierarchies of these medical processes (called *plans* in Asbru). To allow their efficient evaluation and manipulation, they must be visualized in a compact and still clear form. This visualization must be integrated into an editing environment which makes changes to the process hierarchy easy and gives immediate feedback on the changes.

In this paper, we present a novel visualization, Plan Strips, which represents the hierarchy of plans, i.e., processes, as a set of nested strips. It represents the synchronization of the plans by colour-coding the strips and by the ordering of the strips. This saves considerable space compared to graph representations. The visualization is integrated into an editing environment which allows the immediate modification of the plan hierarchy, but also changes to all other aspects of the plan.

Keywords: Process modeling, block-oriented process hierarchies.

1 Introduction

Clinical Guidelines and Protocols (CGPs) are established means of improving health care quality and limiting cost. Modeling them in a computer-executable form is a prerequisite to integrating them into the electronic data flow at the place of care, which again improves adherence to guidelines and which reduces the workload of care staff by showing only relevant recommendations for the case at hand.

Several languages for executable or computer-interpretable CGPs have been developed (see [1] for an overview and comparison). They are tailored to the medical domain and share many features with process modeling and workflow languages. One of them is *Asbru* [2], a language using the block-oriented paradigm. Each block of actions, called plan, has child blocks, called sub-plans, which are ordered in one of the following fashions: sequential, parallel (manadory simultaneous start), unordered (no timing constraints), or any-order (only one sub-plan is active at any time, the order of execution is not defined). In addition, there is the cyclical plan to implement loops. Besides these modes of ordering, conditions to start and finish each plan influence the course of performed actions.

Due to the block-oriented paradigm the hierarchy can become quite complex containing a high number of hierarchy levels. Thus, for the knowledge engineer it is important to maintain an overview over the whole hierarchy, even when it is big and deeply nested. The example guideline model used in this paper comes from the field of breast cancer treatment.

CPG modeling, in general, remains a practical knowledge modelling challenge to this day, requiring the collaboration of a knowledge engineer and domain experts (physicians, guideline developers), because expertise from both computer sciences and medicine must be combined. In the modeling process, it is crucial for the knowledge engineer to present the resulting model in an easily comprehensible form to the domain experts, who are not at all familiar with complex representations of graphs such as hierarchies of treatment steps.

In the context of modeling CGPs in Asbru, we used various representations in the past which all satisfied to some degree (compare Section 2), but still there was a gap left defined by the following requirements.

- *Dense presentation.* Many graph-like presentation use arcs between boxes, which consume considerable space. Also decorations on boxes and arcs tend to increase the space consumed because they need to be printed at a certain size to be readable while they only occupy a small part of the box border or area, or the arc, preventing the utilization of considerable areas along them.
- *Intuitively arrange parallel plans and sequences.* Declaring one axis the time axis and arranging alternatives and parallel plans along the other axis is a well-accepted and immediately comprehensible organization of content.
- *Qualitative presentation of the temporal dimension.* In contrast to other approaches which focus on scale representation of duration and temporal uncertainty, we focus on the mere sequence of plans here. This is attractive if one or more of the following is given: a) The duration holds no interest for the editing task at hand. b) The duration is unknown. c) The durations of different plans are very dissimilar (weeks versus years).
- *Easy to explain to a non-IT person.* Domain experts such as physicians have limited time and little motivation for dealing with IT concepts. For a presentation to be well-received, it is crucial to demonstrate from the start that it is simple. At the same time, we do not see physicians as those modeling CGPs themselves. Therefore, it is not required that they understand our visualisation without an aide, it is only important that it does not appear overly complex or technical.

Since the ultimate aim is to support the knowledge engineering task, such a presentation must be tightly integrated with an editing tool, in which the user modifies the relevant parts of the plans, and which immediately updates the graphical presentation.

In the following, we first discuss related work (Section 2), then present our new representation, Plan Strips, in Section 3, together with the editing tool built around them, and conclude the paper in Section 5.

2 Related Work

The complex nature of clinical practice guidelines demands for a plain and compact visualization of the underlying information to facilitate the generation of a formal guideline model. This includes the temporal ordering of plans and the mode of operation of their sub-plans. Dealing with logical sequences, hierarchical data, as well as time-oriented data, the visualization of clinical treatment plans relates to several specific fields of Information Visualization.

According to [3] 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 patient data.

We are here focusing on the first task where a variety of techniques exists aimed at visualizing logical sequences, such as Flow Charts [4], Clinical Algorithm Maps [5], and Petri Nets [6]. On one hand, tree diagrams and Treemaps [7] are well known techniques to visualize the specific characteristics of hierarchical data, using both dimensions of the plane to spread out the hierarchy. In our case, we need to limit the hierarchy to a single dimension only, in order to show temporal ordering on the second dimension.

On the other hand, several visualization methods have been developed to depict time and time-oriented data (e.g., Time Lines, GANTT Charts, Pert Charts, and Temporal Objects [8]) and clinical time-oriented data in particular (e.g., LifeLines [9] and LifeLines2 [10], Paint Strips [11], and Interactive Parallel Bar Charts (IPBC) [12]).

However, communicating the logics of clinical treatment plans in order to facilitate the modeling of clinical practice guidelines require a visualization method with respect to all of these specific data characteristics. In recent years sophisticated approaches to support the modeling and handling of the complex underlying information were introduced.

VisiGuide, part of the DeGeL (Digital electronic Guideline Library) project [13], is a web-based architecture aimed at facilitating the transformation of a textual guideline into a formal model. The VisiGuide tool is used to browse guidelines and to visualize their structure. It supports the presentation of large amounts of guidelines organized by indexing semantic axes as well as the exploration of the different components of a single guideline.

Protégé [14] is an extensible Java tool for the development of customized knowledge-based systems. The flexible development environment allows for ontology development and knowledge acquisition in order to facilitate the authoring of clinical guidelines in various guideline representation languages. The graphical user interface illustrates the clinical algorithm in a way similar to Flow Charts by using different shapes for plans, decisions, actions, enquiries, and root tasks which are connected by arrows. The Tallis Toolset [15] and the domain-independent GLARE system [16] represent the flow of clinical guidelines in a similar way.

GUIDE [17] was developed at the University of Pavia as part of a guideline modeling and execution framework. It serves a three-fold purpose, i.e., integrating a modelled guideline into clinical workflow, using decision trees and influence diagrams to visualize complex relations, and using extended Petri Nets for the simulation of guideline implementation. Additionally, the GUIDE tool allows for graphically authoring the workflow of guidelines.

AsbruView [18], part of the Asgaard/Asbru project [2], is a graphical user interface to visualize the logical and temporal information of treatment plans expressed in the Asbru modeling language. It uses visual metaphors such as traffic signs and running tracks to communicate complex information, i.e., the hierarchical composition of plans, temporal order of plans, conditions, precise temporal constraints, temporal uncertainties, etc. AsbruFlow (part of the CareVis prototype [19]) is based on Clinical Algorithm Maps [5] extended by Focus+Context techniques to avoid an overcrowded appearance and elaborated symbols indicating the execution order of plans.

None of these approaches deals with compressing the representation to show large plan hierarchies on limited space while maintaining the greatest possible overview. In particular, graph-based representations need space for the arcs in addition to the space used for the boxes. Also, most of the above approaches only show two of the following three aspects in a single diagram: plan decomposition, temporal dimension, and parallel or alternative plans.

There are several fields related to our domain. Extensive research goes into the display of hierarchies much larger than the model of a clinical guideline, e.g., phylogenetic trees. See [20] for an overview. These approaches do not deal with a time dimension but promise to complement our approach. E.g., the magnifying glass effect of hyperbolic trees [21] could be added to our representation of the hierarchy.

The phonetically related field of process modelling guidelines [22,23] provides guidelines to model processes, rather than visualisations. However, applying the principles of business process modelling to guideline development does fertilize the field of guideline modelling in general, albeit beyond the scope of this paper.

The traditional representation of business process models uses node-arc diagrams. The representation of nested sub-diagrams is very limited under such schemes. Also, arranging the nodes in such a way that arcs do not cross more than necessary is an important challenge. See [24] for further reading.

3 Plan Strips

During the modelling process of a CGP in a formal representation such as Asbru, users often lose the overview on the hierarchy of plans, what plans have already been modelled and how plans are synchronized altogether.

With Plan Strips we want to provide a simple and intuitive as well as space-saving means to allow users to get an overview on the hierarchy of plans during the editing process. To keep it small and simple the visualization has to represent the following information:

– The timely order of the plans has to be represented.
– The hierarchy of plans has to be shown.
– The kind of synchronization with other plans has to be displayed: serial, parallel, or cyclical order, alternative plans as well as plans where no synchronization is assigned at all.

3.1 The General Concept

In order to display all the information mentioned above, we use the following methods:

Representing Plans and Their Temporal Order. Plans are represented by rectangles or strips (see Figure 1). Time is presented along the X axis in a qualitative way. This means that sequences of plans are arranged horizontally from left to right. The length of the strips representing them does not relate to the duration, but is optimized for presentation. Thereby, Plan Strips are also applicable with uncertain and undefined timely information. Plans executed in parallel are arranged along the Y axis.

Representing the Hierarchy of Plans. Child plans are stacked on top of their parent plans, with a certain inset. Therefore, the colour of the parent forms a frame around and a link between the children.

Representing the Kind of Synchronization among Plans. We use colour for representing the kind of synchronization of the plans. colour is a well accepted and powerful means to encode different data attributes. In Plan Strips it is used to show the order of plans:

- *Parallel* plans are defined – in Asbru – to start together.
- For *any-order* plans the relative order is not known, but it is known that only one of them can be active at any time, as defined by the Asbru syntax.
- For *unordered* plans nothing is known about the timing of the children.
- For *sequential* plans only one can be active at any time; the order of execution is predefined.
- *Cyclical* plans are repeated several times.

3.2 Finding the Right Colours

In our (western) or any other culture there is no colour-coding that refers to the ordering of plans or processes.

Our initial idea was to map the semantics of plan ordering to colours using the traffic light analogy. Under this scheme, the parallel plan, where everything is clear, was associated with green; the unordered one, where nothing is known, with red; and the any-order plan, which lies inbetween, with yellow. However, initial feedback showed that this scheme was not found intuitive by the target audience.

Next, we use a perception-based colour scheme as suggested in [25] and [26], which have proven to be effective. From a collection of such schemes we choose the most appropriate one with respect to the users' visualization goal, which consists of five different qualitative values of plan ordering. The second dimension to visualise is whether a plan was selected or not. Thus, we need five colour pairs. We used the ColorBrewer2 tool[1] [27,28] with ten different qualitative classes and chose the *Paired* colour system, which consists of five pairs of colours in shades of green, blue, red, orange, and violet. Each colour pair has a very similar hue and differs in saturation and brightness. With these colours the less saturated versions were clearly to distinguish from each other and all the fully saturated colours (compare Figure 1).

[1] `http://www.colourbrewer2.org`, last accessed: June 3rd, 2012

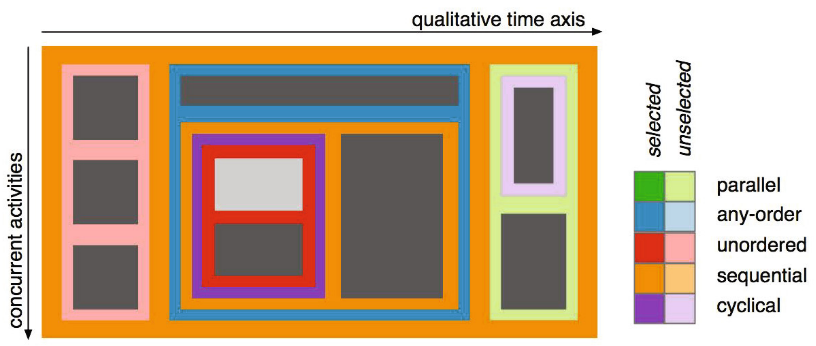


Fig. 1. Explanatory sample of Plan Strips. The light gray box in the center is the currently selected user-performed plan. Its ancestors are shown with normal bright colours. All other plans are shown by colours with less saturation and brightness. The example shows a sequential plan which consists of an unordered plan, an any-order plan, and a parallel plan. The *unordered plan to the left* consists of three user-performed plans. The *any-order plan in the center* consists of a user-performed plan and a sequential plan. This nested sequential plan consists of a cyclical plan and a user-performed plan. The cyclical plan contains a second unordered plan containing two user-performed plans (one of which is the currently selected one). The *parallel plan to the right* contains a cyclical plan containing a user-performed plan, and a user-performed plan.

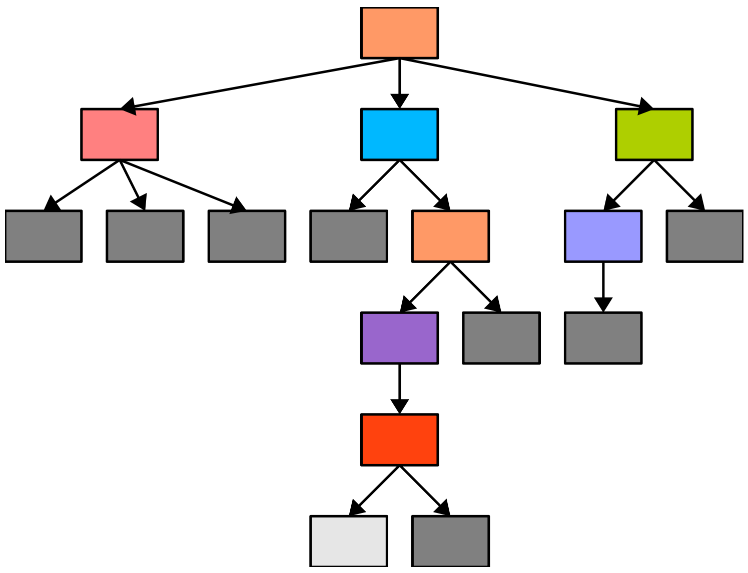


Fig. 2. The same hierarchy as shown in Figure 1, displayed as a conventional tree. The information regarding the ordering of sub-plans and the current plan in focus is represented by colours similar to those in the previous figure.

3.3 Integration with Plan Editing

In order to support productive knowledge acquisition session, our tool not only visualizes the plan hierarchy. It is integrated into an editing environment which permits changes to the hierarchy on the fly, with immediate feedback in the presentation of the Plan Strips.

Figure 3 shows the window of the editing tool into which the Plan Strips visualization is integrated.

When the user clicks on a plan, it is shown in bright colours together with all its ancestors. In contrast, plans currently not selected are shown in colours with lower saturation and higher brightness. After clicking the button "Show all instances", all instances of this plan in the plan hierarchy are shown in bright colours in the Plan Strips display.

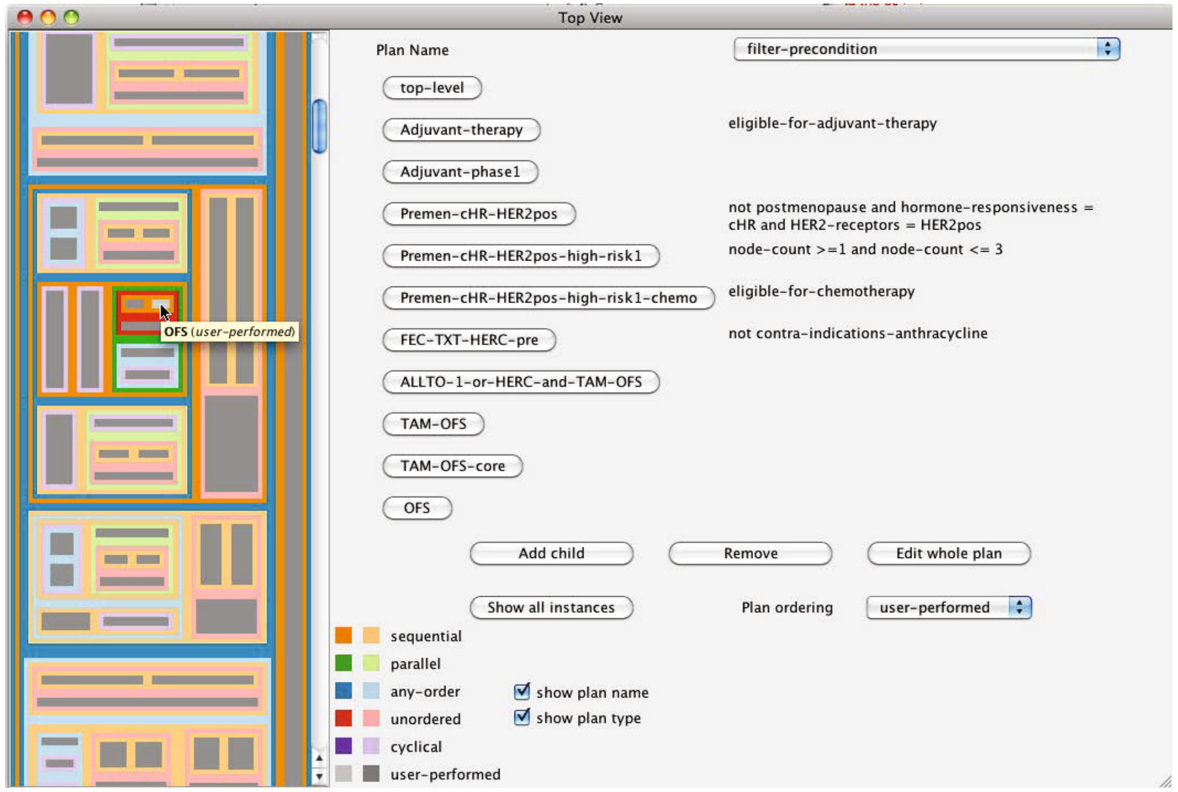
When the user moves over a plan with the cursor, tooltips are shown that display the plan type (parallel, any-order, unordered, sequential, cyclical, user-performed), the name of the plan, or a combination of both. These tooltips assist the users especially in the beginning of their handling with the editor in memorizing the colours that refer to a certain plan type. A legend of the colour palette is also permanently displayed at the bottom of the Plan Strips.

Beside the Plan Strip, the list of ancestors of this plan is shown (including the name of the selected plan itself). Clicking on one of the "Select" buttons in this list selects the corresponding ancestor plan, thereby truncating the ancestor list below the corresponding line. This relieves the user from clicking on the rather small margins between user performed plans to select a plan higher in the hierarchy.

Editing the plan properties takes place on three levels of sophistication.

1. Some plan properties are directly manipulated using the controls in the dialog windows. This comprises adding a child plan, removing the selected plan (via the corresponding buttons), changing the plan ordering (via the combo box showing "user-performed" in Figure 3, and editing the comment attached to the top-level plan element in Asbru in the text area at the bottom.
2. Selected knowledge roles are displayed as abstractions generated by XSL scripts, to the right of the "Select" buttons. This gives them a much more compact presentation compared to the XML code. For each of them, there is a dedicated "Edit" button, which brings up a window showing the XML code of this knowledge role in a tree notation, and allowing the modification of the content.
3. The whole plan can be edited in a similar manner without limits after clicking the "Edit whole plan" button. This allows for more complex modification of the plan body, and of those knowledge roles which are not in the focus of this tool, such as intentions and resources.

The knowledge roles in the second group are the filter precondition, setup precondition, suspend condition, reactivate condition, abort condition, complete condition, continuation specification, and propagation specification. They are displayed for the whole list of ancestors of the selected plan. This is necessary because of their effect. E.g., the activation of a plan only becomes possible if the filter preconditions of all its ancestors are fulfilled. Without a list as shown in Figure 3, it is easy to introduce redundant conditions


Top View

Plan Name	filter-precondition
top-level	
Adjuvant-therapy	eligible-for-adjuvant-therapy
Adjuvant-phase1	
Premen-cHR-HER2pos	not postmenopause and hormone-responsiveness = cHR and HER2-receptors = HER2pos
Premen-cHR-HER2pos-high-risk1	node-count >=1 and node-count <= 3
Premen-cHR-HER2pos-high-risk1-chemo	eligible-for-chemotherapy
FEC-TXT-HERC-pre	not contra-indications-anthracycline
ALLTO-1-or-HERC-and-TAM-OFS	
TAM-OFS	
TAM-OFS-core	
OFS	

Add child | Remove | Edit whole plan

Show all instances | Plan ordering: user-performed

🟧	🟧	sequential
🟩	🟩	parallel
🟦	🟦	any-order
🟥	🟥	unordered
🟪	🟪	cyclical
⬜	⬛	user-performed

✓ show plan name
✓ show plan type

Fig. 3. Plan editing tool featuring Plan Strips to the left, with the selected plan in light gray inside orange; the list of ancestors of the selected plan in the middle; and a selected knowledge role (here, the filter-precondition) for each of them to the right. Below the list of plans there is the legend of the colour palette showing both the colours for selected and unselected plans according to their plan type. Next to the legend there are two check boxes where the user can select the content of the tooltips. As in Figure 1, the time axis runs from left to right. See Section 3.3 for further details.

into a plan hierarchy, or to omit one by mistake, assuming that it is present in another layer of the hierarchy.

The "Settings" menu takes the user to various configurable aspects of the presentation, such as zoom factor of the Plan Strips.

A range of a user-supplied XSL scripts can be invoked from entries in the "XSL" menu, producing custom excepts of various aspects of the plan library, such as lists of plan names, parameter names, cross-reference tables, and similar.

4 Evaluation

We conducted a qualitative user evaluation with four users (computer scientists with and without experience in guideline modelling) to gain insight how well Plan Strips are suited to explore the logic of a guideline's plan hierarchy.

We presented the Plan Strips editor to the users using a protocol for the medical treatment of breast cancer. The modelled protocol consists of 247 distinct plans which are invoked in 440 different places. In the adjuvant therapy part, which constitutes the largest part of the plan library, nesting depth is around 10. The users used Plan Strips to interactively explore the guideline's plan hierarchy and to analyse the dependencies among the plans. The evaluation was carried out separately with each user; in all cases, the interaction with Plan Strips took about half an hour. Users not familiar with guideline modelling were introduced in CPGs and guideline modelling. After a short introduction of the visualization and its features, all users used the Plan Strips editor autonomously to investigate the plan hierarchy. After the interaction phase, the users were interviewed using a mix of open and closed questions.

All users reacted in a positive way to using Plan Strips. It took all of them very little time to learn how to use Plan Strips. When being asked about shortcomings of the visualization, they suggested the following additions and changes: adding the ability for zooming the Plan Strips pane horizontally; adding boundaries between plans and sub-plans of the same type; short term highlighting of the plan in the Plan Strips pane that is selected in the plan list on the right side; adding the ability for collapsing plans; and providing more selectable details to be shown in the tooltips (e.g., conditions of a plan).

The interviews consisted of eight questions, which can be summed up in the following two main questions: (1) Does Plan Strips help to faster assess and model the plan hierarchy of a guideline? (2) Which concepts of the Plan Strips help you most in identifying a plans type?

All users gave affirmative answers to practically all of these questions. When being asked what makes one aware of a plan's type, all of them answered that the ordering on the time axis was the first issue. Further details (whether a plan's type is any-order, parallel, or unordered) were then seen by the colour. Tooltips were also appreciated, although only for seeing the plan's name and not the plan's type, as the colour-coding was quickly memorised and therefore sufficient for identifying the type of ordering.

5 Conclusion

Plan Strips are designed for visualizing process hierarchies in block-oriented process modeling languages, such as Asbru. Plan Strips provide a dense and translucent presentation of plan (or process) hierarchies, omitting quantitative aspects of the temporal dimension. Their usage is advised under the following conditions:

- The hierarchy is highly nested and large.
- There is no interest in the metrics of the temporal dimension, nor in displaying the temporal uncertainty.
- The distinction between parallel or alternative tasks on the one hand and sequential ones on the other hand is crucial.
- The user of the visualization has a basic understanding of the five types of plan ordering in Asbru, but does not want to delve into further details.

Graphical presentations like AsbruView and CareVis are not dense enough for such a high amount of information. Likewise, standard graph representations proved to be too space-consuming (compare Figures 1 and 2). While the amount of space-saving is hard to quantify in a general way, it is obvious that the arcs between the boxes need space to be legible, and this space is wider than the frame formed by the parent plan in Plan Strips.

Initially, we created text-based representations of the call graph, using different types of frames to represent the different plan orderings, but this was less intuitive. It also consumed considerable space – too much to be integrated into the side bar of an editing window.

Clearly, to promote productivity in knowledge acquisition, any visualization must be integrated with a tool that allows the immediate modification of the content presented. In the case of the Plan Strips, we achieved this by showing them side by side with the editing window.

To offer a more versatile solution to knowledge engineers facing the task of creating and maintaining plan hierarchies, the integration of the Plan Strips visualization with other approaches like CareVis and AsbruView is desirable, as well as the implementation on a client-server based architecture which allows the remote collaboration of different users.

Future extensions to Plan Strips include the integration of fish eye perspective, where the mouse acts as a magnifying glass when moved over the Plan Strips; and adding a list of plan names with fish eye perspective, where all the plan names are given beside the Plan Strips, but only those under focus are zoomed sufficiently to be read easily, with the plan under focus clearly standing out.

For the swift modification of plan hierarchies, a drag-and-drop feature would be useful, preferably utilizing a range of clipboards to which parts of the hierarchy can be moved temporarily. Also, the right mouse button could bring up a pop-up menu allowing the user to change the plan ordering of the plan at the mouse pointer, and to remove or add child plans.

We have also evaluated the usefulness in the modelling process. The outcome of the evaluation was very positive; apart from some minor deficiencies that can and will easily be corrected in the near future, the overall outcome of the evaluation showed that

Plan Strips is a valuable enhancement of editing environments for such complex tasks like guideline modelling.

Acknowledgements. The research leading to these results has received funding from the “Fonds zur Förderung der wissenschaftlichen Forschung FWF” (Austrian Science Fund), grant TRP71-N23, and from the Centre for Visual Analytics Science and Technology CVAST (funded by the Austrian Ministry of Economy, Family and Youth in the exceptional Laura Bassi Centres of Excellence initiative, project number 822746).

The plan library shown in the figures was developed during the OncoCure project, which was partly funded by Fondazione Caritro of Trento.

References

1. Peleg, M., Tu, S.W., Bury, J., Ciccarese, P., Fox, J., Greenes, R.A., Hall, R., Johnson, P.D., Jones, N., Kumar, A., Miksch, S., Quaglini, S., Seyfang, A., Shortliffe, E.H., Stefanelli, M.: Comparing computer-interpretable guideline models: A case-study approach. JAMIA 10(1), 52–68 (2003)
2. Shahar, Y., Miksch, S., Johnson, P.: The Asgaard Project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines. Artificial Intelligence in Medicine 14, 29–51 (1998)
3. Aigner, W., Kaiser, K., Miksch, S.: Visualization techniques to support authoring, execution, and maintenance of clinical guidelines. In: ten Teije, A., Lucas, P., Miksch, S. (eds.) Computer-Based Medical Guidelines and Protocols: A Primer and Current Trends, Health Technology and Informatics, pp. 140–159. IOS Press (2008)
4. Goldstine, H.H., von Neumann, J.: Planning and coding problems for an electronic computing instrument, part II, vol. 1, U.S. Ordinance Department (1947); Reprinted in von Neumann, J.: Collected Works, vol. 5, pp. 80–151. McMillian, New York (1963)
5. Hadorn, D.C.: Use of algorithms in clinical practice guideline development: Methodology perspectives. Clinical Practice Guideline Development: Methodology Perspectives 9(95), 93–104 (1995)
6. Petri, C.A.: Fundamentals of a theory of asynchronous information flow. In: Proceedings of IFIP Congress, vol. 62, pp. 386–390. North Holland Publ. Comp., Amsterdam (1963)
7. Shneiderman, B.: Tree visualization with tree-maps: 2-d space-filling approach. ACM Trans. Graph. 11(1), 92–99 (1992)
8. Combi, C., Portoni, L., Pinciroli, F.: Visualizing Temporal Clinical Data on the WWW. In: Horn, W., Shahar, Y., Lindberg, G., Andreassen, S., Wyatt, J.C. (eds.) AIMDM 1999. LNCS (LNAI), vol. 1620, pp. 301–314. Springer, Heidelberg (1999)
9. Plaisant, C., Mushlin, R., Snyder, A., Li, J., Heller, D., Shneiderman, B.: Lifelines: Using visualization to enhance navigation and analysis of patient records. In: Proceedings of the American Medical Informatics Association Fall Symposium (AMIA 1998), pp. 76–80 (1998)
10. Wang, T.D., Plaisant, C., Quinn, A., Stanchak, R., Shneiderman, B., Murphy, S.: Aligning temporal data by sentinel events: Discovering patterns in electronic health records. In: Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2008), pp. 457–466 (2008)
11. Chittaro, L., Combi, C.: Visual Definition of Temporal Clinical Abstractions: A User Interface based on Novel Metaphors. In: Quaglini, S., Barahona, P., Andreassen, S. (eds.) AIME 2001. LNCS (LNAI), vol. 2101, pp. 227–230. Springer, Heidelberg (2001)

12. Chittaro, L., Combi, C., Trapasso, G.: 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, pp. 97–111 (2002)
13. Shahar, Y., Young, O., Shalom, E., Mayaffit, A., Moskovitch, R., Hessing, A., Galperin, M.: DEGEL: A Hybrid, Multiple-Ontology Framework for Specification and Retrieval of Clinical Guidelines. In: Dojat, M., Keravnou, E., Barahona, P. (eds.) AIME 2003. LNCS (LNAI), vol. 2780, pp. 122–131. Springer, Heidelberg (2003)
14. Gennari, J.H., Musen, M.A., Fergerson, R.W., Grosso, W.E., Crubézy, M., Eriksson, H., Noy, N.F., Tu, S.W.: The evolution of Protégé: An environment for knowledge-based systems development. International Journal of Human-Computer Studies 58, 89–123 (2002)
15. Steele, R., Fox, J.: Tallis PROforma Primer – introduction to PROforma language and software with worked examples. Technical report, Advanced Computation Laboratory, Cancer Research, London, UK (2002)
16. Terenziani, P., Montani, S., Bottrighi, A., Torchio, M., Molino, G., Correndo, G.: The GLARE approach to clinical guidelines: Main features. In: Kaiser, K., Miksch, S., Tu, S.W. (eds.) Computer-based Support for Clinical Guidelines and Protocols. Proceedings of the Symposium on Computerized Guidelines and Protocols (CGP 2004). Studies in Health Technology and Informatics, vol. 101, pp. 162–166. IOS Press, Prague (2004)
17. Quaglini, S., Stefanelli, M., Lanzola, G., Caporusso, V., Panzarasa, S.: Flexible guideline-based patient careflow systems. Artificial Intelligence in Medicine 22(1), 65–80 (2001)
18. Kosara, R., Miksch, S.: 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 (2001)
19. Aigner, W., Miksch, S.: Carevis: Integrated visualization of computerized protocols and temporal patient data. Artificial Intelligence in Medicine 37(3), 203–218 (2006)
20. Pavlopoulos, G., Soldatos, T., Barbosa-Silva, A., Schneider, R.: A reference guide for tree analysis and visualization. BioData Mining 3(1), 1 (2010)
21. Lamping, J., Rao, R., Pirolli, P.: A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 401–408 (1995)
22. Mendling, J., Reijers, H., van der Aalst, W.: Seven process modeling guidelines (7pmg). Information and Software Technology (IST) 52(2), 127–136 (2010)
23. Becker, J., Rosemann, M., von Uthmann, C.: Guidelines of Business Process Modeling. In: van der Aalst, W.M.P., Desel, J., Oberweis, A. (eds.) Business Process Management. LNCS, vol. 1806, pp. 30–49. Springer, Heidelberg (2000)
24. Reichert, M., Kolb, J., Bobrik, R., Bauer, T.: Enabling personalized visualization of large business processes through parameterizable views. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing, SAC 2012, pp. 1653–1660. ACM, New York (2012)
25. Brewer, C.A.: Color Use Guidelines for Mapping and Visualization. In: MacEachren, A.M., Taylor, D.R.F. (eds.) Visualization in Modern Cartography, ch. 7, pp. 123–147. Elsevier Science, Tarrytown (1994)
26. Bergman, L., Rogowitz, B., Treinish, L.: A rule-based tool for assisting colormap selection. In: Proceedings of IEEE Visualization 1995, pp. 118–125 (1995)
27. Brewer, C.A.: Color use guidelines for data representation. In: Proceedings of the Section on Statistical Graphics, pp. 55–60. American Statistical Association, Baltimore (1999)
28. Harrower, M.A., Brewer, C.A.: Colorbrewer.org: An online tool for selecting color schemes for maps. The Cartographic Journal 40(1), 27–37 (2003)