Communicating Time-Oriented, Skeletal Plans to Domain Experts Lucidly

Silvia Miksch and Robert Kosara

Vienna University of Technology, Intitute of Software Technology (IFS)
Resselgasse 3/188, A-1040 Vienna, Austria, Europe
{silvia, rkosara}@ifs.tuwien.ac.at, www.ifs.tuwien.ac.at/~{silvia, rkosara}

Abstract. Practical planning systems for real-world environments imply a striking challenge, because the planning and visualization techniques available are not that straightforwardly applicable. Skeletal plans are an effective way to reuse existing domain-specific procedural knowledge, but leave room for execution-time flexibility. However, the basic concepts of skeletal plans are not sufficient in our medical domain. First, the temporal dimensions and variability of plans have to be modelled explicitly. Second, the compositions and the interdependencies of different plans are not lucid to medical domain experts. The aim of our paper is to overcome these limitations and to present an intuitive user interface to the plan-representation language *Asbru*. We explored different representations and developed a powerful plan visualization, called *AsbruView*. AsbruView consists of two views, first, a *topological* view, which utilizes the metaphor graphics of "running tracks" and "traffic" and, second, a *temporal* view, which utilizes the idea of LifeLines.

1 Introduction

Currently, there are some plan-representation languages and various planning techniques available. However, for someone who wants to engineer a *practical* planning system for a particular real-world domain, it is not that straightforward to simply select a representation and a technique and proceed.

We are trying to bridge the gap between theory and practice providing concrete support for *medical treatment planning*. We have encountered two main problems when applying planning techniques in real-world environments: (1) traditional plan-representation languages are not appropriate in dynamically changing environments, like medicine; (2) it is next to impossible to communicate such complex abstract concepts to domain experts, like physicians. Therefore, we focused on a plan-representation language and a suitable visualization of this language. Our final aim is to support the authoring and the execution of clinical protocols seen as time-oriented, skeletal plans. Such skeletal plans turned out to be beneficial to capture domain-specific procedural knowledge.

Section 2 discusses the problem area and the related techniques. Section 3 gives an overview about the time-oriented, skeletal plan-repesentation language, called *Asbru*. Section 4 explains the main features of our plan visualization, called *AsbruView*, using the medical scenario of mechanical ventilation. Finally, we end up with conclusion and future plans.

2 Problem Area and Approaches

Medical Treatment Planning. In recent times, physicians have tried hard to improve the quality of health care through increased awareness of proper disease management techniques while simultaneously the physicians have to reduce costs without adversely affecting the quality of patient care. Treatment planning from scratch typically is not necessary, as general procedures exist which should guide the medical staff. These procedures are called *clinical guidelines* or *protocols*.

Authoring clinical protocols is a non-trivial task. Mostly, these protocols are expressed in natural language or flow diagrams, but these kinds of representation can not easily be transformed into a formal and structured framework [7]. The benefits of existing representations are: (1) writing in free text is easy; (2) medical experts are used to working with free text or flow diagrams; (3) flow diagrams are useful for representing sequential states and actions in a graphical way.

However, these techniques have significant limitations when used in *practical* planning systems: (1) existing clinical protocols are partly vague concerning their intentions and their temporal, context-dependent representations; (2) the variability of clinical protocols is hard to represent in a structured way (e.g., a medical goal can be achieved by different therapeutic actions); (3) it is quite difficult to cope with all possible orders of plan execution and all the exception conditions that might arise; (4) it is hard to represent the concurrent actions, the different temporal dimensions, the high numbers of possible transitions, and the mutual dependencies of parameters in an easy to comprehend way. Therefore, hardly any of the existing protocols are formulated in an appropriate way, which would facilitate computer support.

Representing Time-Oriented Plans. Clinical protocols can be seen as plans, procedures, or algorithms, which need to be executed depending on a patient's health conditions within a certain time interval.

The planning and scheduling problems have been attacked in two major ways: approaches which try to understand and solve the general problem without using of domain-specific knowledge (*domain-independent* approaches) and approaches which use domain heuristics directly (*domain-dependent* approaches). The "classical" domain-independent approach that many planners use describes states and operators in a restricted language known as the STRIPS language [3], or in extensions thereof. The STRIPS language is based on situation calculus [10]. Therefore, all approaches that descended from STRIPS are unable to handle durative events and actions, uncertainty and variability in the utility of available actions, and concurrent and cyclical execution of plans. STRIPS's search space is close to situation space. The usual assumptions are that only the agent affects the state of the world, that all actions occur instantaneously, that effects of actions are instantaneous, and that all actions follow one another with no break inbetween. Finally, classical planning and scheduling assume complete and deterministic information about the world's states and the effects of actions. These assumptions are inappropriate in medical domains.

To overcome some of these limitations, approaches as the planning initiative "Shared Planning and Activity Representation" (SPAR [17]), the procedural reasoning systems (PRSs, [6]), situated and reactive planning ([4], [16], [20]) or O-Plan [18] were proposed. However, we need to have greater temporal reasoning power and to focus on new issues such as temporally extended goals [1].

Another way is representing procedural knowledge as a library of skeletal plans. Skeletal plans are plan schemata at various levels of detail that capture the essence of procedures, but leave room for execution-time flexibility in the achievement of particular goals [5]. However, the basic concepts of skeletal plans are not sufficient in the medical domain, either. The temporal dimensions and variability of clinical protocols have to be modelled explicitly in skeletal plans.

Plan Visualization. Graphical representations support the understanding of complex coherences. In the last years, many visualization techniques were introduced to improve the understanding of the relationship between several variables (e.g., [13], [8], [19]). Cole and Stewart [2] suggested to use metaphor graphics to display a collection of different parameters over time (e.g., minute-ventilation rectangles representing the mechanical ventilator data) and found that human performance on interpreting mechanical ventilator data can be improved significantly [2]. This approach assumes that *"metaphor graphics are custom tailored visual displays designed to look like the real world situation from which the data is collected, but not in a literal sense of 'look like' "* [2]. We extended the idea of metaphor graphics in the literal sense of "metaphor" [9]. A metaphor supports comprehending an unknown complex concept using a well-known concept.

We are more interested in plan visualization than in visualization of multidimensional data. Besides flow charts, graphical animation languages, visual (programming) languages, and process modeling techniques, LifeLines [12] provide an excellent way to represent data and actions over time. LifeLines are diagrams with time lines proceeding from left to right. These lines are drawn in different vertical areas, with a label to the very left-hand side of the area. While events whose dates are known (e.g., past events) are captured very well by this approach, it does not deal with temporal uncertainty, different temporal orders of plans, and compulsory or optional plans.

Which Features Do We Need? We have already developed a plan-representation language, called Asbru, which explicitly defines all the necessary knowledge roles. However, we ended up with a quite complicated and difficult to comprehend language (compare Section 3). We could not communicate the basic concepts to the domain experts – the physicians. Therefore, we need a plan visualization, which is able to capture:

1. *hierarchical decomposition* of plans (which are uniformly represented in a plan-specification library);

2. *compulsory* and *optional* plans;
3. *temporal order*: sequential, concurrent, and cyclical execution of plans;
4. *temporal uncertainty*;
5. *continuous* (durative) states, actions, and effects;
6. *intentions* considered as high-level goals; and
7. *conditions*, that need to hold at particular plan steps.

3 Asbru Language

Considering all shortcomings of traditional plan-representation languages, we defined a temporal, skeletal plan-specification language, called *Asbru* [11]. Asbru is part of the Asgaard project[1] [15], in which we are developing task-specific problem-solving methods (PSMs) based on such time-oriented, skeletal plans written in the Asbru notation. These PSMs will support the design and the execution of skeletal plans by a human executing agent other than the original plan designer (e.g., plan verification, plan selection, plan revision).

Components of Asbru. A plan consists of a name, a set of arguments, including a time annotation (representing the temporal scope of a plan), and five components: *preferences*, *intentions*, *conditions*, *effects*, and *a plan body (layout)*, which describes the actions to be executed. The plan name is compulsory and all other components are optional. Table 1 explains the different components of Asbru.

Table 1. Components of Asbru.

Component	**May Consists of**	**Explanation**
Preferences		*constrain the selection of a plan*
	strategy	a strategy for dealing with the problem
	utility	a set of utility measures
	select-method	a matching heuristic for the applicability of the whole plan
	resources	a set of prohibited, recommended, discouraged, and obligatory resources
	responsible-actor	a set of actors, who are entitled to adapt the protocols (e.g., physician, nurse)
Intentions		*are high-level goals at various levels of the plan, an annotation specified by the designer; intentions are temporal patterns that should be maintained, achieved, or avoided*

continued on next page

[1] In Norse mythology, *Asgaard* was the home and citadel 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.ifs.tuwien.ac.at/asgaard/`).

continued from previous page

Component	**May Consists of**	**Explanation**
	intermediate-state	the state(s) that should be maintained, achieved, or avoided *during* the applicability of the plan
	intermediate-action	the action(s) that should take place *during* the execution of the plan
	overall-state-pattern	the overall pattern of state(s) that should hold *after* finishing the plan
	overall-action-pattern	the overall pattern of action(s) that should hold *after* finishing the plan
Conditions		*are temporal patterns, sampled at a specified frequency, that need to hold at particular plan steps to induce a particular state transition of the plan instance*
	filter-preconditions	the preconditions which need to hold initially if the plan is applicable, but can not be achieved and are necessary for a plan to become `possible`
	setup-preconditions	the preconditions which need to be achieved to enable a plan to start and allow a transition from a `possible` plan to a `ready` plan
	activate-condition	a token which determines if the plan should be started manually or automatically
	suspend-conditions	the conditions which determine when an `activated` plan has to be `suspended`
	abort-conditions	the conditions which determine when an `activated`, `suspended`, or `reactivated` plan has to be `aborted`
	complete-conditions	the conditions which determine when an `activated` or `reactivated` plan can be completed successfully
	reactivate-conditions	the conditions which determine when a `suspended` plan has to be `reactivated`
Effects		*describe the possible effects of plans*
	functional relationship	relationship between the plan arguments and measurable parameters
	overall effect	overall effect of a plan on parameters independent of plan's arguments
Plan-body (layout)		*is a set of plans to be executed in sequence, in parallel, in any order, or in some frequency*
	sequential (`do-all-sequentially`)	a set of plans that are executed in sequence (executed in total order)
	concurrent : parallel (`do-all-together`)	a set of plans that are executed in parallel – all plans must start together; no `continuation-condition`

continued on next page

continued from previous page		
Component	**May Consists of**	**Explanation**
	concurrent : parallel (`do-some-together`)	a set of plans that are executed in parallel – some plans must start together; the `continuation-conditions` specified as subset of plans, which must be completed
	concurrent : any order (`do-all-any-order`)	a set of plans that are executed in any order – all plans must be completed; no `continuation-condition`
	concurrent : any order (`do-some-any-order`)	a set of plans that are executed in any order – some plans must be completed; the `continuation-conditions` specified as subset of plans, which must be completed
	cyclical (`every`)	a repeated plan with optional temporal and continuation arguments that can specify its behavior

Hierarchical Decomposition of Plans. The aim is to specify different plans, which are uniformly represented in a plan-specific library. Therefore, all plans and their subplans have the same structure. A *plan* in the plan-specification library is composed hierarchically of a set of plans with arguments and time annotations. The execution interpreter always attempts a decomposition of a plan into its subplans, unless the plan is not found in the plan-specification library, thus representing a nondecomposable plan (informally, an *action*). This is called *"semantic" stop-condition.*

Time Annotations. Intentions, states, and prescribed actions are temporal patterns. A simple temporal pattern is a *parameter proposition*: a parameter (or its abstraction), its value, a context, and a time annotation (e.g., the *state*

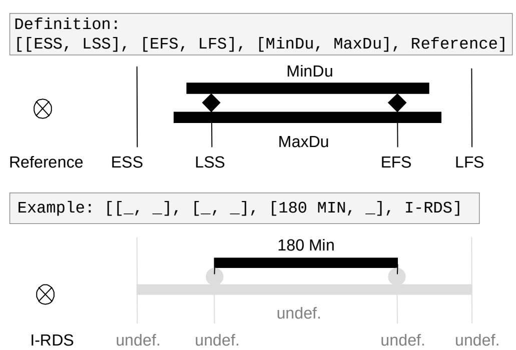

Fig. 1. Asbru's Time Annotations. The upper part of the figure presents the generic annotation and the lower part shows an example.

abstraction of the blood-gas parameter is *normal*, as defined in the context of weaning therapy, during a certain time period).

The time annotations allow us to represent uncertainty in starting time, ending time, and duration [14]. This time annotation supports multiple time lines by providing *reference annotations*. Temporal shifts from the reference annotation are defined to represent the uncertainty in starting time, ending time, and duration, namely earliest starting shift (ESS), latest starting shift (LSS), earliest finishing shift (EFS), latest finishing shift (LFS), minimal duration ($MinDu$), and maximal duration ($MaxDu$). The temporal shifts are associated with time units (e.g., minutes, days) and can be "unknown" or "undefined" to allow incomplete time annotation, denoted by an underscore "_". To allow temporal repetitions, sets of cyclical time points and cyclical time annotations are defined. Figure 1 illustrates these time annotations.

4 AsbruView: Topological and Temporal View

Our plan-visualization approach was influenced by the idea of metaphor graphics ([2], [8]), LifeLines [12], and the graphical-timetable design of Shinkansen Lines (Japanese National Railroad) described in [19]. However, we are utilizing the idea of metaphors more literally. Instead of using an abstract diagram or object (e.g., rectangles, growing and shrinking circles), we are applying signs from the (more-or-less) daily life to communicate the various components of our plan-representation language Asbru.

Our approach, called AsbruView, consists of two views, a topological view and a temporal view. The topological view is eligible to depict the overall flows of the different plans, the hierarchical decomposition of plans, the compulsory and optional plans, and several time-oriented components of the plans. However, this view is incapable for representing the temporal uncertainty in an appropriate way. Therefore, we needed the temporal view to embody this dimension, too.

4.1 Topological View

The topological view is qualified to communicate the basic concepts of Asbru and the overall control and decompositions of plans to be executed. We are utilizing the metaphor of "running tracks" to visualize such a plan. The 3-dimensional objects sketch the "running tracks". The width represents the time axis, the depth represents parallel plans on the same level of decomposition, and the height represents the decomposition of plans into subplans. The cube is rotated to the left to ensure readability in case of multiple tracks. Figure 2 presents a screenshot of the AsbruView program. The upper part of Figure 2 shows parts of a treatment protocol for infants' respiratory distress syndrome (I-RDS). The general rule of *undefined components* is that these icons appear in gray.

Plans can be stacked on top of each other to represent the hierarchical decomposition. For example, the plan *One-of-Controlled-Ventilation* is decomposed into three subplans, called *Controlled-Ventilation*, *Permissive-Hypercapnia*, or

Crisis-Management (Figure 2). These three subplans will be executed in any order (`do-some-together`). However, *Controlled-Ventilation* is compulsory (displayed with plain background) and the other two subplans are optional (displayed with question-mark texture). The finishing-line (flag) stands for the `complete-conditions`.

We are using the metaphor of "traffic" to visualize the other five kinds of conditions (Figure 2). The sign "No Entry with Exceptions" symbolizes the `filter-preconditions`. The supplementary sign stands for the exeptions, like "Except Buses", which we are using to name the `filter-conditions` (e.g., "Except Females" allows only females to enter the track). A barrier, which illustrates the fact that this condition can be achieved, embodies the `setup-preconditions` (and thus the barrier will be opened). The traffic light includes three kinds of conditions: the red light symbolizes the `abort-conditions`, the yellow light the `suspend-conditions`, the green light the `reactivate-conditions`.

4.2 Temporal View

The strengths of the temporal view are to grasp the temporal uncertainty and to explain the plans, their subplans, and their components in more detail. We have adapted the idea of LifeLines (compare Section 2) to represent temporal uncertainty, different temporal orders of plans, and compulsory or optional plans.

The lower part of Figure 2 shows the temporal view. We are using facets [12] for all of Asbru's components: plan body (layout), preferences, intentions, conditions, and effects. Facets can be opened and closed at any time, and share a common time axis. Thus, the relation between different parts of the display is very easy to understand. Vertical scrolling of the different facets is independent. However, vertical scrolling within the topological or temporal view is dependent.

A plan is represented with uncertainty in starting time, ending time, and duration. The time-annotation can be constrained implicitly by the plan's conditions (e.g., `complete-conditions`) or explicitly by defining starting time, ending time, or duration (compare Section 3). Since time annotations play an important role in all aspects of Asbru, the same kind of representation can be used in all facets (explanations of the time annotations are given in Figure 1). If all the different components are defined, the upper bar has to lie at least on the two diamonds, because the minimal duration must be equal or less than the difference between latest starting shift and earliest finishing shift ($MinDu \leq EFS - LSS$). If the LSS or the EFS are undefined, the black diamonds are converted to gray circles. The two diamonds have to stay on the lower bar accordingly. The visualization of the time annotations is the only metaphor, which breaks our rule of using only signs of the daily life instead of abstract objects.

The symbol next to every plan's name shows its type. In the example, *Initial-Phase* is a sequential plan and *One-of-CPAP-Extubation* is an any-order plan; two parallel lines would indicate a parallel plan and a cyclical arrow would illustrate a cyclical plan. The order of execution is also indicated by the position of the time annotation along the time axis. In case of plans that are to be executed

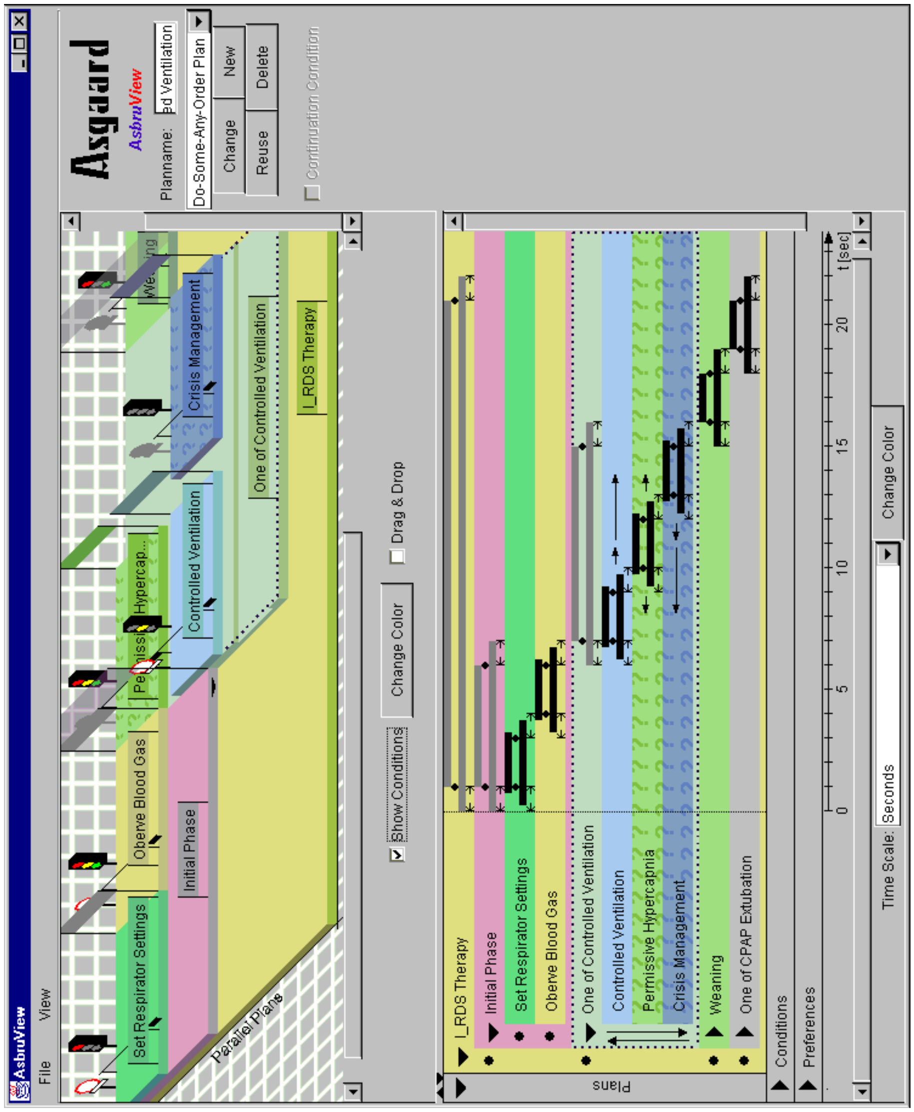

Fig. 2. A screenshot of the *AsbruView* program. The upper part illustrates the topological view showing an example of a real clinical protocol for treating infants' respiratory distress syndrome (I-RDS). The lower part depicts the temporal view.

in any order, the time annotations are displayed with arrows pointing to other possible execution times (e.g., the subplan *Controlled-Ventilation*).

5 Conclusion and Future Plans

We outlined the necessity for suitable plan representation and plan visualization for practical planning systems in real-world domains. Our plan visualization approach is based on metaphor graphics and LifeLines, called AsbruView. AsbruView consists of two views, which support different features of our plan-representation language Asbru. Asbru is a time-oriented and intention-based language to represent skeletal plans. We have utilized the metaphors of "running tracks" and "traffic". These metaphors clarify the complex plan-representation language Asbru in a comprehensible way. We have implemented most features of AsbruView in $Java^{tm}$.

The applicability of AsbruView was evaluated with scenario-based techniques. We applied treatment protocols of mechanically ventilated newborn infants and analyzed AsbruView's expressiveness with collaborating physicians. AsbruView is able to visualize most of the features of Asbru in an easy to understand way and supports the navigation through a complex plan-specification library. Therefore, domain experts need not be familiar with the Asbru syntax to understand a plan.

According to [9], abstract concepts are defined by clusters of metaphors. Each metaphor gives a partial definition and these partial definitions overlap in certain ways. Therefore, better understanding of concepts may best be served by permitting alternative metaphors even at the expense of completeness and consistency. Alternative metaphors of our problem domain could be "road maps" or "golf courses". However, we have first chosen the metaphors of "running tracks" and "traffic", which seemed easier to comprehend and more appropriate for our domain experts.

Acknowledgements. The authors thank Georg Duftschmid, Johannes Gärtner, Klaus Hammermüller, Werner Horn, Peter Johnson, Franz Paky, Christian Popow, Andreas Seyfang, and Yuval Shahar who provided helpful comments and discussions on previous versions of this document. This project is supported by "Fonds zur Förderung der wissenschaftlichen Forschung" (Austrian Science Fund), P12797-INF.

References

1. F. Bacchus and F. Kabanza. Planning for temporally extended goals. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 1215–1222, Menlo Park, CA, 1996. AAAI Press/The MIT Press.
2. W. G. Cole and J. G. Stewart. Human performance evaluation of a metaphor graphic display for respiratory data. *Methods of Information in Medicine*, 33:390–396, 1994.

3. R. E. Fikes and N. J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3-4):189–208, 1971.
4. R. Firby. Adaptive execution in complex dynamic worlds. Ph.D. Thesis YALEU/CSD/RR #672 Thesis, Yale University, 1989.
5. P. E. Friedland and Y. Iwasaki. The concept and implementaion of skeletal plans. *Journal of Automated Reasoning*, 1(2):161–208, 1985.
6. M. P. Georgeff, A. L. Lanskey, and M. J. Schoppers. Reasoning and planning in dynamic domains: A experiment with mobile robots. Technical Report Tech Note 380, SRI International, AI Center, 1986.
7. S. I. Herbert. Informatics for care protocols and guidelines: Towards a european knowledge model. In C. J. Gordon and J. Christensen, editors, *Health Telematics for Clinical Guidelines and Protocols.* IOS Press, Amsterdam, 1994.
8. W. Horn, C. Popow, and L. Unterasinger. Metaphor graphics to visualize ICU data over time. In *Proceedings of the Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP-98).* Workshop Notes of the ECAI-98 Workshop, 1998.
9. G. Lakoff and J. Mark. *Metaphors We Live By.* University of Chicago Press, Chicago, 1980.
10. J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and M. Michie, D. andSwann, editors, *Machine Intelligence 4*, pages 463–502. University Press, Edinburgh, Scotland, 1969.
11. S. Miksch, Y. Shahar, and P. Johnson. Asbru: A task-specific, intention-based, and time-oriented language for representing skeletal plans. In E. Motta, F. van Harmelen, C. Pierret-Golbreich, I. Filby, and N. Wijngaards, editors, *Proceedings of the 7th Workshop on Knowledge Engineering: Methods & Languages (KEML-97).* Open University, Milton Keynes, UK, 1997.
12. C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, and B. Shneiderman. Life-Lines: 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, 1998.
13. S. M. Powsner and E. R. Tufte. Graphical summary of patient status. *The Lancet*, 344(6):386–389, 1994.
14. J.-F. Rit. Propagating temporal constraints for scheduling. In *Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86)*, pages 383–388, Los Altos, CA, 1986. Morgan Kaufman Publishers, Inc.
15. Y. Shahar, S. Miksch, and P. 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, 1998.
16. L. A. Suchman. *Plans and Situated Actions: The Problem of Human/Machine Communication.* Cambridge University Press, Cambridge, 1987.
17. A. Tate. Roots of SPAR - shared planning and activity representation. *The Knowledge Engineering Review*, 13(1), 1998.
18. A. Tate, B. Drabble, and J. Dalton. O-plan: A knowledge-based planner and its application to logistic. In *Advanced Planning Technology*, pages 213–239. Morgan Kaufmann, 1996.
19. E. R. Tufte. *Envisioning Information.* Graphics Press, Cheshire, CT, 1990.
20. D. E. Wilkins and K. L. Myers. A common knowledge representation for plan generation and reactive execution. *Journal of Logic and Computation*, 5(6):731–761, 1995.