A Workflow Model for the Asgaard Project: A Time-Oriented, Skeletal Planning Workbench in Medicine

Klaus Hammermüller and Silvia Miksch

Vienna University of Technology, Institute of Software Technology,
Resselgasse 3/E188, A-1040 Wien, Austria
{klaus, silvia}@ifs.tuwien.ac.at

Abstract. Planning medical therapies is a very creative task depending on the patients' situation and the skill of the responsible physicians. Much time is spent to develop standard procedures, which can be seen as skeletal plans. Such plans are very useful in order to reduce the work to the essential individual adaptations reusing existing domain-specific knowledge. We are presenting a workflow model of medical therapy planning and showing the integration of the plan-representing language *"Asbru"* in different user aspects. The information overload, generated by modern devices and the high speed of medical progress, has to be structured to stay manageable. It is important to keep the physician in contact with the patient and leave the last decision in human hands. Therefore, this paper is the draft of the general framework of how *"Asbru"* would help to connect on-line patient data with state-of-the-art skeletal therapy plans to guide through complex decisions and decompose related information.

1 Introduction

To support therapy planning, clinical guidelines are established in many areas of medical care. Generally speaking they are represented as free text, tables, or flow-charts. These documents are far from perfect, because they do not integrate the input from different sources to a consistent workflow participating different actors and do not allow automatical support for verification or quality assessment.

In the Asgaard/Asbru[1] project [1, 14], a number of methods are being developed to deal with problems of clinical therapy planning. To make these methods usable for the medical staff there is the need for an intuitive visualization, discussed in [9]. This part of the Asgaard project is named *"AsbruView"*. Asbru itself is a language used for representing therapy plans in a LISP-like syntax which is outlined in section 2.2.

[1] In Norse mythology, *Asbru* (or *Bifrost*) was the bridge from our world to *Asgaard*, the home of the gods.

To enable an optimal support of different users, we have to reach the needs in their tasks. Different point of views must be generated upon the common record and be integrated in an effective workflow [7]. In this paper we would like to present how this aim is supported by the methods we are developing in the Asgaard project. This paper analyses the target environment and outlines a general framework for Asgaard, by linking different technical aspects into a user-driven workflow. Technical terms are mostly defined by the *"unified modelling language" (UML)[13]* notation.

In the following section we give a short introduction to the key concepts of the project. Our workflow model, plus some possible solutions integrating Asbru into work are given in section 3. We end up with an overview about related work in section 4) and add a conclusion in section 5.

2 The Asgaard Project

2.1 System Design

In Figure 1 we give an overview about the communication model of the Asgaard-project. Based on skeletal plans for medical therapy (based on clinical guidelines) and the collection of executed therapies reusable knowledge is collected in a shared plan-library. They are accessible for medical staff through an "intelligent" interface which offers information related to the current task by task-specific problem solving methods "PSM" [4].

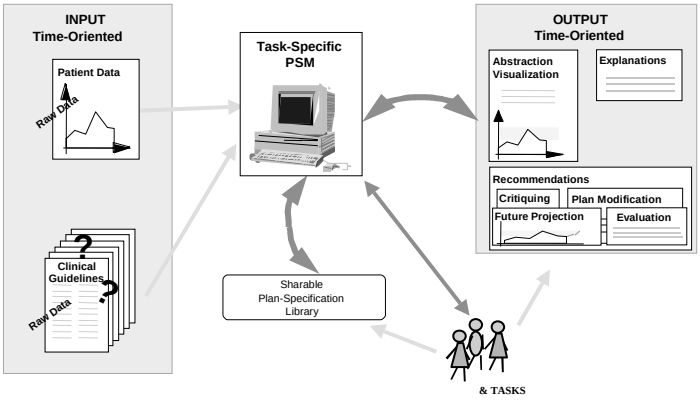

Fig. 1. Outline of the Asgaard system

Skeletal plans may be processed in discrete steps and can be matched to patient data for instance to preselect a set of useful plans for a concrete case. The user is implicitly integrated into this model with reference (a "hyperlink" feature of Asbru connecting context-information to a Asbru-statement) to the

documentation of user-statements in the library and explicitly supported by different task-specific visualization of the output.

Complex therapy tasks and relationships between actions that may be too complex to be outlined in flow-diagrams efficiently, because they lack for instance a clear concept of time [6], can be modelled in Asbru. All therapy tasks can be decomposed. The question why a plan is generated and how it should work can always be explained. The patient data is domain-specific converted to abstract qualitative descriptions which can be intuitively visualized and are easier to maintain in case of changing domain knowledge. These qualitative descriptions are used for referring to a skeletal plan.

2.2 Asbru

The framework shown in the following section is primary structured by the functional needs of its users and by the capabilities of the underlying representation, which is determined by the Asbru-syntax. The plan-definition language Asbru has a hierarchically structure which decomposes a plan (labeled in Figure 2 with AA) into subplans unless atomic actions (labeled with letters A to I) are found, the plan-components (e.g. preferences) are optional. Asbru is used for defining skeletal plans and its instances are linked to the related patient data [11].

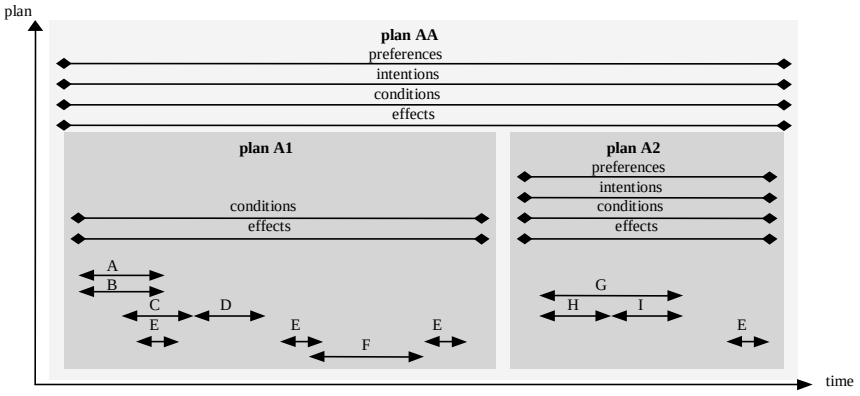

Fig. 2. Outline of the major Asbru components

There are five major plan-components in Asbru:

- *Preferences* describe the plan's behavior (e.g. the strategy) and constrain its applicability (e.g. select-criteria);
- *Intentions* define high-level overall aims which should be achieved during plan execution. Intentions are used for selecting an adequate plan and also for reviewing as part of improving the medical knowledge;

– *Conditions* are the control-mechanism for executing a plan and its sub-plans. A plan can be in a finite state like *started, suspended, reactivated, aborted,* or *completed*. Two different kinds of conditions (called preconditions) control the start of a plan: *filter-preconditions* cannot be achieved, whereas *setup-preconditions* must. Is a plan is suspended it needs true *restart-condition* otherwise it has to be aborted by the *abort-condition*. In case a plan reaches its goal the *complete-condition* is true;
– *Effects* describe the target result, manifested by the change of a parameter or plan state by means of mathematical functions including a probability of occurrence;
– The *plan-body* designs the workflow of the sub-plans or actions (atomic plans). The layout of these sub-plans is restricted to *sequence, any order, parallel, periodically* execution. They are performed if their preconditions are satisfied.

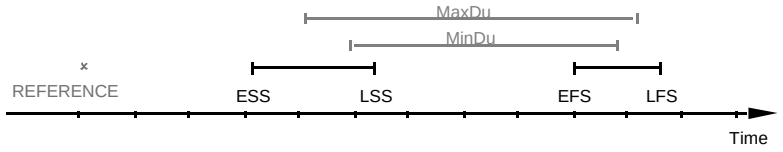

Fig. 3. Time Annotation used in Asbru

In figure 3 the concept of time in Asbru is defined, which may incorrectly be used in plans. Relative to a *reference* point *earliest* - (ESS), and *latest starting shift* (LSS) defining the start-window of an plan-element. *Earliest* - (EFS), and *latest finishing shift* (LFS) defining the finishing-window. The effective duration is limited to the *minimum* - (MinDu) and *maximum duration* (MaxDu).

3 Functional Framework

3.1 The Workflow's General Outline

Most of the following figures are drawn in the UML notation [13], containing user-roles, use-cases, objects and relationships between them. A (user-)*role* may be processed by one or more persons who may perform different roles, for instance a physician determines the therapy plan and may also execute some of the therapy actions. An *use-case* is a task which may contain a set of further tasks related to a closed field of application. *Objects* are physical manifestations of a result produced by a use-case.

Figure 4 gives an overview about the workflow in a typical planning-process. Two circular workflows are performed: One is the circuit between the use-cases *"diagnose"*, *"planning therapy"* and *"execute therapy"* involving the patient and

medical staff, this circuit is driven by the responsible physician. The second one are *"author guidelines"* and their use in *"planning therapy"* based on the state-of-the-art know-how and the *"analyzes"* collected by domain experts. These guidelines are driven by medical domain experts and may be used by the responsible physician and the therapist. Both processes involve different people but they should be based upon a common documentation. The requirements of the roles are very different. We are convinced of the need of a machine-readable representation for this documentation that supports different aims and user-views, which can be used in different workflow.

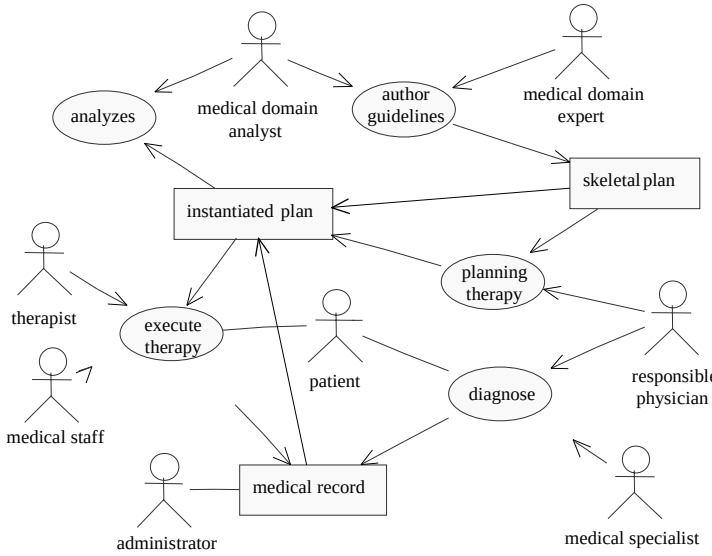

Fig. 4. Use-Case-Model of the workflow in therapy planning (legend: ellipse is a use-case, rectangle is an Object, arrows are relationships between them; user-role)

The critical point is the connection from the (existing) real-data of the patient's *"medical record"* linked together with a chosen (and maybe individually adapted) *"skeletal plan"* producing an *"instantiated plan"*, which is executed in the real-world. That would open up two timelines with different dimensions: A short-circuit data is floating continuously between the patient, the medical staff executing the therapy and the responsible physician, in contrast to a discrete, long-term process refining medical know-how.

3.2 Information Flow

Before going more into detail, we would like to emphasize to the involved information, flowing on two different timelines, showed in Figure 5. The different

dimensions of time and the different triggers of tasks have great impact on the further workflow.

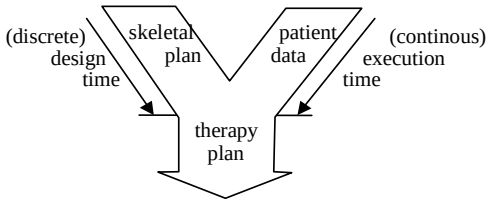

Fig. 5. Discrete design time vs. continuous execution time

Design Time. The first major line is following the design-process of the skeletal plans. Starting with the acquisition of domain knowledge and ending with the output as a set of skeletal plans. The different phases (discussed in the next section) of this work are discrete and triggered by the intervention of a member of the medical staff.

Execution Time. The second major line is the processing of the patient's data. Existing medical patient-data-management systems may be integrated as well as direct automatic or manual input is possible. Raw-data is preprocessed to filter false or noisy data producing reliable data. These quantitative values are transformed to qualitative descriptions [10], which are used in the skeletal plan. The most important differences to design time are:

- Execution time is continuous and patient data is arriving continuously;
- The data processing once established is working automatically.

During therapy-execution both dimensions of time are involved. Discrete triggered actions by the medical staff and monitoring the continuous flow of patient information are creating a recursive cycle of diagnosis, therapy planning, and therapy execution.

3.3 Use-Cases in Therapy Planning

In Figure 6 we are structuring the workflow to the major use-cases and have added only the most important user roles. The two time-lines outlined in Figure 5 are present too, in this section we are concentrating on design time.

Author Concept. In this task all meta-information is defined, setting up a common plan-environment as a general outline for all skeletal plans concerned,

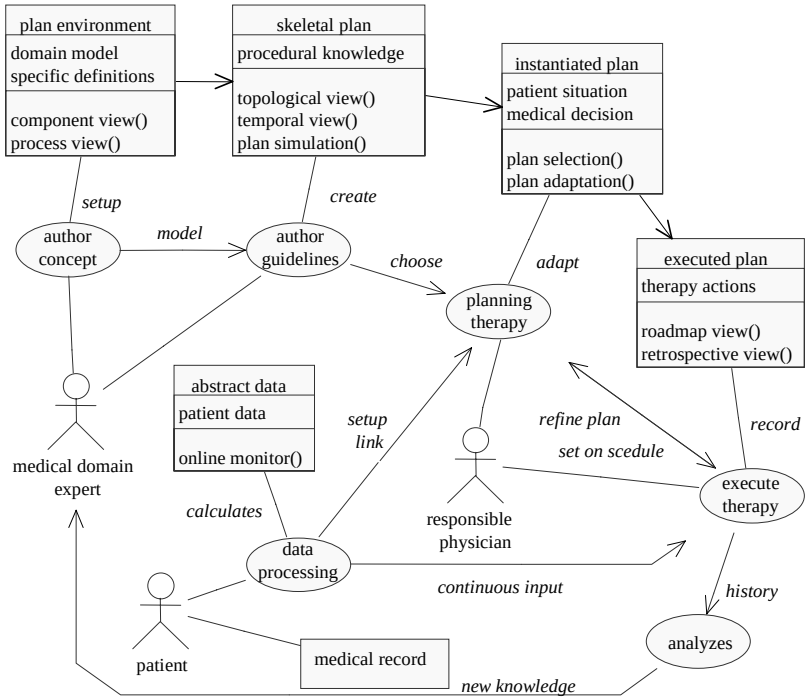

Fig. 6. Major Use-Cases at Design Time at Therapy Planning (legend: cycle is a use-case, rectangle is an Object with attributes and methods, arrows are relationships between them; user-role)

which are designed later on (compare the top of Figure 6). It represents an overall domain model related to the target problem domain. Two issues are discussed in this task by an expert board:

- As Asbru is a generic language, there is need of extending the syntax to enable acquired domain-specific knowledge;
- Defining the general process model containing the structure of the plan hierarchy and defining a data-processing-model for computing patient data.

Domain-specific definitions are used in Asbru as keywords and are linking related domain-specific knowledge rules, which are implemented on a knowledge base. Both definitions are composed with an editor and parser to guarantee consistence generating Asbru-components (valid partial language constructs). An example of the Asbru syntax shown below is generated automatically:

```
(DOMAIN-DEPENDENT TIME-ASSIGNMENT    ;; define a new keyword
    (SHIFTS DELIVERY <- 38 WEEKS)    ;; time shift from CONCEPTION
    (POINT CONCEPTION <- (ask
        (ARG 'what is the conception-date?')))
```

These Asbru-components are later used to carry their properties in different views. The user should do their work visually as far as possible. First of all, there is the outline of the overall plan hierarchy as a general process model during the design time, which is supported by a topological plan-view discussed in [9], see also Figure 7.

Another point of view is the definition of usable data-processing models which are linked to Asbru via an abstraction-function. This model controls the continuous computing of the patient's data at the execution time to generated abstract values used in Asbru. Therefore a visual model builder can be used to connect the different components carrying modular abstraction-functions for different input data [10] and generating the Asbu-code as shown in the following example:

```
(COMPLETE-CONDITIONS                    ;; shows use of abstraction
    (delivery TRUE GDM-Type-II *        ;; function GDM-Type-II
        (SAMPLING-FREQUENCY 30 MINUTES)))
```

We distinguish this task "author concept" from the next one "author guidelines" to prevent from the well-known problem of inconsistence within definitions and knowledge rules, which should be maintained very carefully. A bad plan may be ignored by the physicians, a bad definition may spoil the whole plan library.

Author Guidelines. In this task different ways to handle a problem presented as skeletal plans can be modelled, based on the conceptual model defined (compare Figure 6). The focus is now on the dynamic aspects in order to perform a therapy. No more "global" definitions should be done in this step, e.g. "What does temperature mean?" should be defined in the previous task and stay consistent within all plans. "Local" knowledge used only in a specific therapy may be added to interpret the incoming patient data. A guideline may be authored by an expert board or a local medical domain expert.

The main part is defining skeletal plans, discussed in [14] and supported by three different views:

- The topological outline is similar to the view used in "author concept" but more detailed and with a tool to maintain the different properties of the Asbru-components, see figure 7;
- A temporal view illustrates the interactive impact of different sub-plans to the dimensions of time [8]. The development of a therapy in the real-world is related to various periods of time actions so as to become effective;
- Different plans may be simulated to calculate the impact of the therapy as a "playground" for the domain engineer.

A background task is to verify and validate the authored plans automatically. That means that the plans get a syntax check and a semantic check, which should guarantee the consistence of the plan within the plan library. [3]

```
(DO-ALL-SEQUENTIALLY                    ;; defines the outline of the
    (initial-phase)                     ;; plan-body controlling the
```

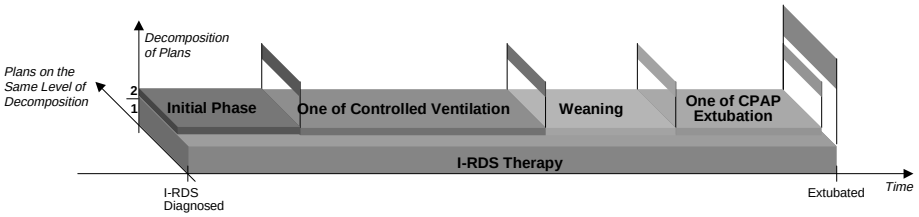

Fig. 7. An Asbru Plan in topological View, adapted from [6]

```
(one-of-controlled-ventilation)  ;; therapy roadmap
(weaning)
(one-of-cpap-extubation)))
```

The output of this task is a reusable skeletal plan which includes all information necessary to perform a particular therapy for a typical class of patients. The information is structured within the major components in Asbru described above (see Figure 2). This plan represents one way of handling a medical problem and is part of a plan-library which offers other comparable plans to the responsible physician in the next task.

Therapy Planning. This task makes the matching of the patient data to the skeletal plan by the responsible physician. The access to related reusable and standard procedures helps to save time and to concentrate on remarkable developments. Two sub-tasks are performed:

- Selecting an adequate skeletal plan and connecting this plan to the patient data.
- Adapting the plan to the individual needs of the patient and the restrictions of the situation.

To select a plan, all plans are matched with the patient's data preselecting those, which fulfill filter-conditions of Asbru. We also support a simulation of different plans with the known patient-data to show different alternatives in the therapy path.

The next step would be the adaptations of plans in a way that the responsible physician can do a quick navigation by starting at the current state of the therapy path. Those site-specific adaptations may be changed in the near future, for instance changing the drug dosage. Changes are verified and validated, and cross-checked by a background task, including all available facts to calculate whether the therapy plan stays in the defined borders or not. Monitoring the progress of the therapy in real-world may also include replanning events, driven by the physician to react to an abnormal development of the patient.

The output of this task is an instantiated plan, carrying the link to all available and related patient data. The skeletal plan should be "filled" and modified to the needs of the individual situation and ready for execution in reality.

Execute Therapy. This task implements the actions defined in the therapy plan in reality by the therapy staff. The *therapist* is supported by different means.

- Visualizing the roadmap of the plan therapy, related to the current situation, including which action has to be performed next and which impact previous actions have had (and should have). Adapted to this situation we use a special topological view (see Figure 7);
- A monitoring tool visualizes the data-processing pipeline from raw data to qualitative description used in Asbru. Different time scales are needed for prospective and retrospective aspects [10];
- A simulation of the available information to show trends about the therapy's impact to the patient and to generate warnings and messages through this process.

As life is not as simple as a plan may suggest, there has to be a replanning capability. If a plan is rejected the focus goes back to the therapy planning task to check a new setup. It is aimed to keep the therapy on track as long as possible to reduce the count of necessary steps backwards. Finally, a record of the executed actions (which may differ to the planned actions) containing all related information is filled to be used for future analyses and further context information about the implemented case.

Analyzes. In this task we close our world with the analyses of historical data and plans by a domain-expert who may integrate new knowledge into the system by analysing existing plans, refining rules or authoring new plans. Therefore we are implementing tools [10] for data analyses and plan-analyses.

4 Related Work

During the past 15 years, there have been several efforts to create automatic reactive planners to support the process of protocol-based care over significant periods of time. In the prescriptive approach, an active interpretation of the guidelines is given; examples include *ONCOCIN* [16] in the oncology domain, *T-HELPER* [12] in the AIDS domain, and *DILEMMA* [5], and the European *PRESTIGE Health-Telematics* project, as general architectures.

In the critic approach, the program criticizes the physician's plan rather than recommending a complete one of its own. This approach concentrates on the user's needs and assumes that the user has considerable domain-specific knowledge. A task-specific architecture implementing the criticizing process has been generalized in the *HyperCritic system* [17]. Task-specific architectures assign well-defined problem-solving roles to domain knowledge and facilitate acquisition and maintenance of that knowledge.

Several approaches to the support of guideline-based care encode guidelines as elementary state-transition tables or as situation-action rules dependent on the electronic medical record [15], but they do not include an intuitive representation

of the guidelines clinical logic, and have no semantics for the different types of clinical knowledge represented. Other approaches permit hypertext browsing of guidelines via the World Wide Web [2], but do not use the patient's electronic medical record.

None of the current guideline-based-care systems have a sharable representation of guidelines that has knowledge roles specific to the guideline-based-care task, is machine and human readable, and allows data stored in an electronic patient record to invoke an application that directly executes the guidelines logic and related tasks, such as critiquing. Such a machinereadable language and the task-specific problem solving methods need to solve that problem described in [14]. However the workflowmodel needs classification to be applicable to real-world environments.

5 Conclusion

We have demonstrated how the workflow in medical therapy planning can be supported by generating different views using the plan-representing language *"Asbru"* which integrates the input from different actors.

Existing knowledge can be reused, at each step of the process only the "delta" must be added to create skeletal plans. Patient data can be fitted automatically into those plans. Anyhow, the responsibility of executing actions rests with the physician, but information is related to the current situation. Therapeutic plans can be adopted individually during therapy generating as an accurate roadmap for the next actions, which includes clear and immediate aims for the next steps.

Each task can be supported optimally and specifically to its needs. Finally a backward chaining is always possible in order to decompose available information related to a concrete question.

Acknowledgements

We wish to thank Georg Duftschmid, Werner Horn, Robert Kosara, Katharina Renath, Andreas Seyfang, and Christian Popow for valuable suggestions and discussions and for the chance to work with them. This project is supported by *"Fonds zur Förderung der wissenschaftlichen Forschung" (Austrian Science Foundation)*, P12797-INF.

References

1. The Asgaard-project: http://www.ifs.tuwien.ac.at/šilvia/projects/asgaard/.
2. M. Barnes and G. O. Barnett. An architecture for a distributed guideline server. *Proceedings of the Annual Symposium on Computer Applications in Medical Care (SCAMC-95)*, pages 233–7, 1995.
3. G. Duftschmid, S. Miksch, Y. Shahar, and P. Johnson. Multi-level verification of clinical protocols. *The Workshop on Validation and Verification of Knowledge-Based Systems in conjunction with the Sixth International Conference on Principles of Knowledge Representation and Reasoning, (KR'98)*, June 1998.

4. H. Erikson, Y. Shahar, S. W. Tu, A. R. Puerta, and M. A. Musen. Task modeling with reusable problem-solving methods. *Artificial Inteligence*, 2:293–326, 1995.
5. S. I. Herbert, C. J. Gordon, A. Jackson-Smale, and J.-L Renaud Salis. Protocols for clinical care. *Computer Methods and Programs in Biomedicine*, 48:21–6, 1995.
6. Robert Kosara, Silvia Miksch, Yuval Shahar, and Peter Johnson. AsbruView: Capturing complex, time-oriented plans — beyond flow-charts. In *Thinking with Diagrams 98*, 22–23 August 1998.
7. P. Lawrence. *Workflow Handbook*. John Wiley & Sons, West Sussex, England, 1997.
8. S. Miksch and R. Kosara. Visualization techniques for time-oriented, skeletal plans in medical therapy planning. In W. Horn, Y. Shahar, G. Lindberg, S. Andreassen, and J. Wyatt, editors, *Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making, AIMDM'99*, Berlin, 1999. Springer. in print.
9. S. Miksch, R. Kosara, Y. Shahar, and P. Johnson. AsbruView: Visualization of time-oriented, skeletal plans. In *Proceedings of the 4th International Conference on Artificial Intelligence Planning Systems 1998 (AIPS-98)*. Carnegie Mellon University, AAAI Press, June 7–10 1998.
10. S. Miksch, A. Seyfang, W. Horn, and Popow C. Abstracting steady qualitative descriptions over time from noisy, high-frequency data. In W. Horn, Y. Shahar, G. Lindberg, S. Andreassen, and J. Wyatt, editors, *Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making, AIMDM'99*, Berlin, 1999. Springer. in print.
11. S. Miksch, Y. Shahar, and P. Johnson. Asbru: A task-specific, intention-based, and time-oriented language for representing skeletal plans. *7th Workshop on Knowledge Engineering: Methods and Languages (KEML-97)*, pages 22–24, 1997.
12. M. A. Musen, C. W. Carlson, L. M. Fagan, S. C. Deresinski, and E. H. Shortliffe. T-helper: Automated support for community-based clinical research. *Proceedings of the Sixteenth Annual Symposium on Computer Applications in Medical Care (SCAMC-92)*, pages 719–23, 1992.
13. J. Rumbaugh, I. Jacobson, and Booch G. *Unified modeling language : reference manual*. Addison-Wesley, Harlow, England, 1998.
14. 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.
15. E. H. Sherman, G. Hripcsak, J. Starren, R. A. Jender, and P. Clayton. Using intermediate states to improve the ability of the arden syntax to implement care plans and reuse knowledge. *Proceedings of the Annual Symposium on Computer Applications in Medical Care (SCAMC-95)*, pages 238–42, 1995.
16. S. W. Tu, M. G. Kahn, M. A. Musen, J. C. Ferguson, E. H. Shortliffe, and L. M. Fagan. Skeletal-plan refinement on temporal data. *Communications of the ACM*, 32:1439–55, 1989.
17. J. Van der Lei and M. A. Musen. A model for critiquing based on automated medical records. *Computers and Biomedical Research*, 24(4):344–78, 1991.