A User Interface for Executing Asbru Plans

Robert Kosara Silvia Miksch

Institute of Software Technology
Vienna University of Technology, Austria
{rkosara, silvia}@ifs.tuwien.ac.at
http://www.ifs.tuwien.ac.at/~{rkosara, silvia}/

Abstract. Asbru is a language for specifying treatment plans. These plans are then used by an execution unit to give advice to the medical staff what actions to take (open-loop system). In order to do this, the execution unit has to monitor patient data and needs to be informed about the duration and success of actions.
This communication requires a flexible and adaptive user interface that 1) makes it possible to understand which information is currently needed most urgently, and 2) allows the user to tell it the outcome of treatment steps he or she has performed. It must also be possible to simulate patient parameters for testing treatment plans prior to actual clinical trials.
We present a user interface, called AsbrUI, that meets these criteria.

1 Introduction

Asbru [5, 6] is a plan-specification language for defining clinical protocols. A clinical protocol is usually translated into many Asbru plans that contain small units of the treatment and information about how they work together. This knowledge acquisition and translation is supported by a visual tool named AsbruView [3].

The execution of these plans is monitored and controlled by an execution unit that gives advice to medical staff based on the procedures laid down in Asbru plans. This is called an *open-loop system*, as opposed to a *closed-loop system* where no human is involved who can check proposed settings or steps before they are performed.

The execution unit needs information about the patient, only a part of which can be acquired automatically. It also needs information about actions the medical staff has performed, and their outcome. The user interface for the execution unit is called AsbrUI, and is described in this paper. Because we are still in the design phase of AsbrUI, all images presented in this paper are mock-ups.

2 Related Work

There is a user interface for PROforma [1], which allows the definition and execution of plans. But it does not show when a data item that it requests is needed, nor does it provide a way of aborting or suspending user actions.

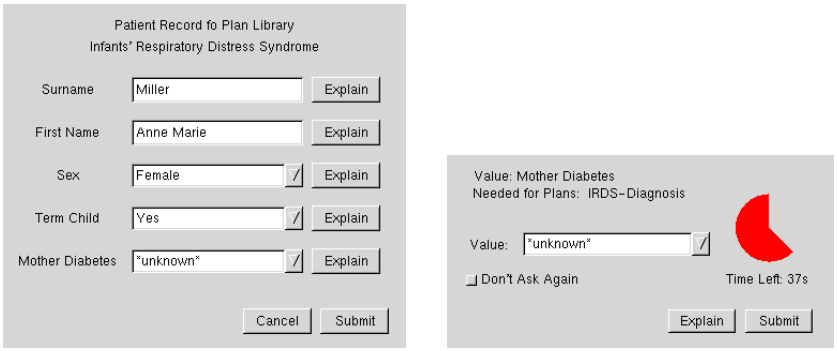


Fig. 1. Patient record input windows. Left: The complete patient record as shown when the first plan ist started. Right: The execution unit asks for a specific value.

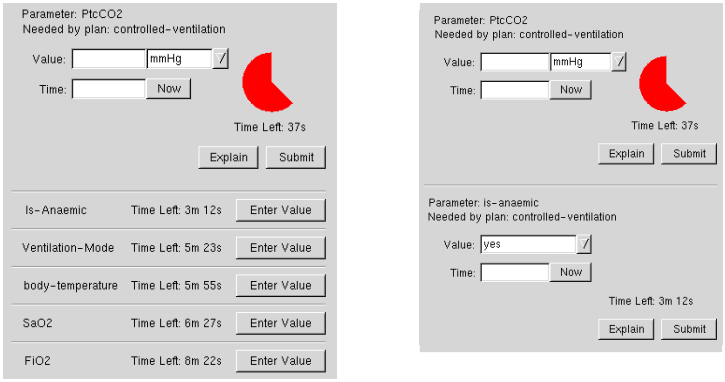
There is also a strict distinction between diagnosis and treatment phases in PROforma, which is the reason for parameter input during plan execution not being supported in this interface. To the best of our knowledge, there is no runtime support appropriate for the medical staff for either GLIF [4] or the Arden Syntax [2].

3 Data Acquisition

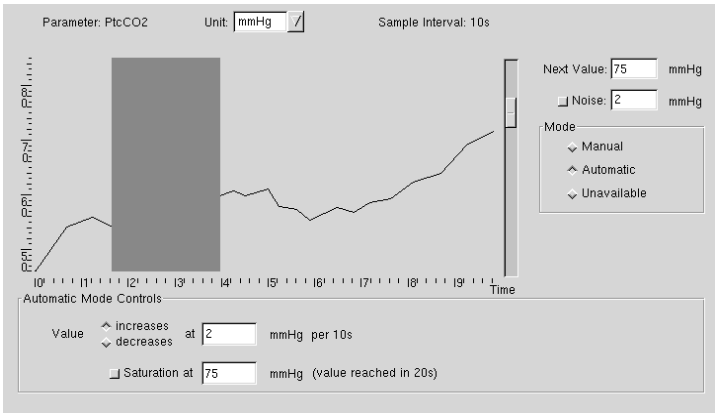
Two different types of data are acquired by AsbrUI: Entries in the patient record and parameters.

Patient Record. In Asbru, the patient record not only contains information like the name, date of birth, etc. of the patient, but can also contain additional information like if the patient has a genetic predisposition to a certain disease. A form (Figure 1, left) is shown at the start of the first plan, where the information for the current patient can be entered. Information that is not entered here but is later on needed is requested with the form shown in Figure 1, right.

Manual Parameters. Parameters are all data that comes from the patient, rather than being generated in the execution unit (program variables). Manual parameters are parameters that cannot be acquired automatically. The execution unit asks the user interface for values, and gives it the latest point in time, when the value must be available. This "latest point in time" is determined depending on the context the parameter is used in. For example, if a parameter has a certain sampling period, the end of the current sampling interval is the deadline for this value. AsbrUI then provides an input mask (Figure 2a, right) which tells the user about the time constraint and allows input of the value and the time when the value was actually acquired (this is important for tests that take longer, like blood samples). The users are warned if the deadline for a value is less than a certain pre-specified amount of time away. This is done by displaying a circle



a) The list of currently needed parameters (left) and input fields for single values (right)



b) The environment simulator in automatic mode.

Fig. 2. Parts of AsbrUI: manual parameters and the environment simulator.

for each minute in that time span, with the last one being "cut" to reflect the amount of seconds still left in that minute.

Usually, of course, there will be more than just one parameter that the execution unit is waiting for. AsbrUI sorts the list of asked parameters by their urgency (Figure 2, left) and also lists the plans that need that parameter, so that the user knows what the value is needed for.

Automatic Parameters and the Environment Simulator. Automatic parameters are usually acquired by devices rather than entered manually. For testing of plans, it is useful to be able to also control the values of these parameters, so that different paths through the plan can be tried out. Each automatic parameter is displayed in the environment simulator (Figure 2b) as a function over time. The value that will be fed into the abstraction unit at the next sampling point can be set manually, automatically, or be set to "unavailable".

4 Plan Execution Control

Manual plans need approval by a user before they can be started. AsbrUI displays simple dialog boxes for this case, where the user can allow or deny a plan from running. This is more complicated when such plans are part of a parallel plan. In this case, a dialog is shown that allows the user to select which subplans of that parallel plan to start (not shown due to space restrictions).

The execution unit monitors patient data while a user-performed plan is active, but does not necessarily know when it is complete or when it has to be aborted, because not all conditions might be filled with enough information or the user might decide to abort or complete a plan regardless of fulfilled conditions. For this purpose, AsbrUI uses a dialog that allows the user to suspend, abort and complete user-performed plans (not shown due to space restrictions).

5 Conclusions and Future Work

We have presented a part of a user interface design for the execution of Asbru plans. It is able to provide the user with information he or she needs to make needed data available on time, and to control the control flow for manual plans.

The obvious next step is to implement AsbrUI and test it together with the execution unit that is in development in real-world scenarios.

Acknowledgments

This work is part of the Asgaard Project, supported by "Fonds zur Förderung der wissenschaftlichen Forschung" (Austrian Science Fund), grant P12797-INF.

References

1. John Fox and Subrata Das. *Safe and Sound: Artificial Intelligence in Hazardous Applications*. MIT Press, 2000.
2. George Hripcsak, Peter Ludemann, T. Allan Pryor, Ove B. Wigertz, and Paul D. Clayton. Rationale for the Arden syntax. *Computers and Biomedical Research*, 27:291–324, 1994.
3. Robert Kosara and Silvia Miksch. Metaphors of movement: A visualization and user interface for time-oriented, skeletal plans. *Artificial Intelligence in Medicine, Special Issue on Information Visualization in Medicine*, 22(2):111–131, 2001.
4. Lucila Ohno-Machado, John H. Gennari, Shawn Murphy, Nilesh L. Jain, Samson W. Tu, Diane E. Oliver, Edward Pattison-Gordon, Robert A. Greenes, Edward H. Shortliffe, and G. Octo Barnett. The guideline interchange format: A model for representing guidelines. *JAMIA*, 5(4):357–372, 1998.
5. Andreas Seyfang, Robert Kosara, and Silvia Miksch. Asbru's reference manual, Asbru version 7.2. Technical Report Asgaard-TR-2000-3, Vienna University of Technology, Institute of Software Technology, 2000.
6. Yuval Shahar, Silvia Miksch, and Peter Johnson. The Asgaard project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artificial Intelligence in Medicine*, 14:29–51, 1998.