Environment-Driven Skeletal Plan Execution for the Medical Domain

Peter Votruba[1], Andreas Seyfang[1], Michael Paesold[1], Silvia Miksch[1,2]

Abstract. An important application of both data abstraction and plan execution is the execution of clinical guidelines and protocols (CGP), both to validate them against a large set of test cases and to provide decision support at the point of care. CGPs can be represented and executed as a hierarchy of skeletal plans. To bridge the gap between low-level data and high-level concepts in the CGP, intelligent temporal data abstraction must be integrated with plan execution.

In this paper we describe a solution to this challenge which was implemented as part of the European project Protocure II to improve the quality of CGPs. They are translated to the high-level plan representation language Asbru which again is compiled into a network of abstraction modules by the system. Then this network performs the content of the plans triggered by the arriving patient data.

By this, we seamlessly integrate the synchronisation of guideline execution with observed patient state, complex temporal abstractions and execution of complex plans without requiring the user to handle the low-level details. Instead, user-friendly tools are used to create and maintain the guideline.

1 Introduction

In the field of medicine, the application of clinical guidelines and protocols helps to improve the quality of care by ensuring the optimal choice of treatment. A precondition for the successful application of clinical guidelines and protocols is the automatic abstraction of context-dependent time-annotated raw-data (e.g., percent of oxygen in blood at a certain second) to high-level medical concepts (e.g., sufficient oxygen saturation during an extended period of observation). This is performed by temporal data abstraction.

Within the domain of medicine, intensive care poses additional challenges to plan execution. The nature of the data raise the need for elaborate pre-processing of the data, including complex knowledge-based plausibility checks and correction heuristics which involve multiple channels and time windows and the synchronisation of low-frequency and high-frequency inputs. This processing together with the sometimes complex propagation of plan states needs to be performed in a timely manner to meet the real-time requirements of online patient monitoring.

To meet these requirements, we integrate time-oriented, skeletal planning using the Asbru representation with real-time monitoring and temporal data-abstraction.

There are many guideline modelling approaches today [4, 6, 1], but only few integrate strong data abstraction resources. This may be caused by the fact that in most settings, data is entered manually which allows to delegate the data abstraction task to the user by demanding qualitative high-level input instead of the original data. However, the integration of guideline execution into the clinical data flow becomes more and more important in order to apply decision support systems in clinical daily practice [2].

[1] Institute of Software Technology, Vienna University of Technology, Austria
[2] Department of Information & Knowledge Engineering, Danube-University Krems, Austria

2 Plan Representation in Asbru

Asbru is a time-oriented plan representation language that represents clinical guidelines as skeletal plans [5]. Asbru's distinguishing features are that intentions, conditions, effects and world states have a temporal dimension and are continuous (durative). Because of the advanced temporal data abstraction capabilities, diagnosis and treatment can be tightly integrated allowing each one to support the other one.

All conditions for the transition from one plan state to another are expressed in terms of temporal patterns. A temporal pattern consists of one or more parameter propositions or plan-state descriptions. Each parameter proposition contains a value description, a context, and a time annotation. The time annotation used allows a representation of uncertainty in starting time, ending time, and duration of an interval. Start and end are defined as shifts from a reference point. Reference points can be defined as sets of cyclical time points or references to parameter changes, allowing repeated temporal patterns.

3 The Asbru Interpreter

Conceptually, the Asbru Interpreter consists of three basic units: data abstraction, monitoring, and plan execution. In the data abstraction unit, various temporal or atemporal abstractions are applied to the patient data to gain information at higher conceptual levels. The provided quantitative or qualitative data is monitored to detect temporal patterns in the abstracted data. This information is used to control the selection and execution of plans. This data flow is not unidirectional, instead, the execution unit can interact with both monitoring and abstraction unit to adjust the monitored patterns and to adapt the abstraction process to the context given by the current plan states.

The interpreter has two different modes of operation: Batch mode and interactive mode. In batch mode, a large set of records is read to validate a guideline against patient data or to create complex abstractions of the data for later analysis. In interactive mode, data can be read from monitoring devices in addition to the user input.

Figure 1 shows the parts of the interpreter on the implementation level. The main parts are the Asbru Compiler and the Execution Manager. The three conceptual components mentioned above – data abstraction, monitoring, and plan execution – are seamlessly integrated in the module graph.

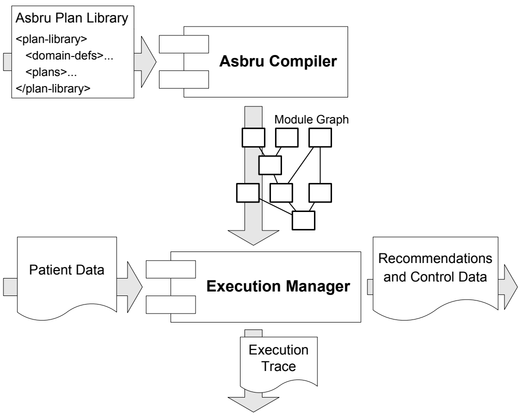

Figure 1. System architecture and data flow in the Asbru Interpreter. The Asbru plan library is compiled into a directed graph of modules by the Asbru Compiler. The Execution Manager uses this module network to process patient data and execute the plans representing the guideline.

At program start-up, the Asbru plan library XML file is compiled into a directed graph of modules.

For each time step, the Execution Manager enacts each of the modules in the network to process patient data, monitor temporal patterns, and execute the plans in the guideline. These modules are largely compatible with each other, which allows information extracted by any module to flow back into the abstraction or monitoring process. To handle complex networks with many inputs in high-frequency applications, the Execution Manager ensures that each module is enacted exactly when needed, allowing for small time steps by some modules without the overhead created by other modules which would not provide new information at that moment.

Modules can set alarms, to be triggered when a certain span of time is elapsed. Here we distinguish between *pre-alarms* and *post-alarms* depending on whether the alarm is triggered before or after processing the data for this time step. This distinction allows the implementation of both complex and convex temporal intervals. Alarms are set by various monitoring and value aggregation modules.

The available modules can be divided into the following categories.

Raw data modules. These modules interface the input channels and map the raw-data parameter definitions in Asbru.

Value abstraction modules. This group comprises the logical and arithmetic combination of inputs, and the mapping of quantitative values to qualitative categories.

Value aggregation modules. In order to map high-frequency, error-prone inputs to high-level concepts, it is mandatory to aggregate series of measurements and to derive the abstractions from them, and not from single measurements.

Monitoring modules. These modules handle temporal patterns, such as parameter propositions, which control the state changes of Asbru plans.

Temporal abstraction modules. The patterns detected by monitoring modules and aggregates of the measurements often need one or more steps of temporal abstraction to detect complex patterns in the input data such as "five episodes of apoea followed by hyperoxemia during the previous hour".

Plan modules. Modules in this category represent Asbru plans or single plan steps. The network of parent and child plans is fully integrated with the other modules to from the module graph.

4 Evaluation and Future Work

An extensive real-world guideline [3] was modelled in Asbru and test runs of the resulting model in the interpreter were successful. Running the interpreter on patient cases will allow the comparison between the expected outcome of applying the guideline and the actual outcome according to the model in Asbru.

In practical tests with high-frequency data recorded at an intensive care unit, the interpreter processed input from 10 channels and moderately complex abstractions thereof at a rate of more than 1 kHz on a standard PC. Most clinical data is recorded at 1 Hz, or 200 Hz. We therefore conclude that the computational performance is sufficient for real-time applications in clinical monitoring.

Future work will go into the construction of dedicated modules to interface equipment at intensive care units, where it will be employed in clinical studies. In addition, a graphical user interface to allow the interactive use of the interpreter by non-computer experts is under development.

5 Conclusion

Plan execution in real-world high-frequency domains such as intensive care units demand for tight integration of temporal data abstraction and plan execution to achieve the required intelligent reaction to unpredictable changes in the environment, i.e., the patient state.

While the knowledge in such domains is abstract, partly vague or incomplete, and often complex, the data arrives at a high rate and in a format that is far from the level in which the domain knowledge is specified. Translating the domain knowledge to such low levels by a knowledge engineer leads to well known short-comings regarding maintenance and assuring the correctness of the model.

We therefore designed an interpreter, which takes a high-level specification of skeletal plans and temporal data abstraction and compiles them into a network of abstraction modules. Using elaborate management of data flow, these modules process the data at a high rate, even in complex configurations.

Tests have demonstrated that the interpreter can handle a complete guideline, large sets of patient records, and high-frequency measurements. Applications to a large set of data are currently in progress.

ACKNOWLEDGEMENTS

This work is part of the Protocure II project, which is supported by the European Commissions IST program, under contract number IST FP6-508794.

REFERENCES

[1] P. Ciccarese, E. Caffi, L. Boiocchi, S. Quaglini, A. Kumar, and M. Stefanelli, 'A guideline management system', in *MedInfo 2004*, ed., M. Fieschi et al., pp. 28–32, Amsterdam, (2004). IOS Press.

[2] W. Horn, 'AI in medicine on its way from knowledge-intensive to data-intensive systems', *Artif Intell Med*, **23**(1), 5–12, (2001).

[3] Nationaal Borstkanker Overleg Nederland (NABON), *Guideline for the treatment of breast cancinoma*, Van Zuiden Communications B.V., 2002.

[4] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R. Greenes, R. Hall, P Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E. Shortliffe, and M. Stefanelli, 'Comparing computer-interpretable guideline models: A case-study approach', *JAMIA*, **10**(1), (2003).

[5] A. Seyfang, R. Kosara, and S. Miksch, 'Asbru 7.3 reference manual', Technical report, Vienna University of Technology, (2002).

[6] P. Terenziani, G. Molino, and M. Torchio, 'A modular approach for representing and executing clinical guidelines', *Artif Intell Med*, **23**(3), 249–276, (2001).