Verification of medical guidelines with KIV

Jonathan Schmitt, Micheal Balser, Wolfgang Reif

Angaben zur Veröffentlichung / Publication details:

Schmitt, Jonathan, Micheal Balser, and Wolfgang Reif. 2008. "Verification of medical guidelines with KIV." In *Computer-based medical guidelines and protocols: a primer and current trends*, edited by A. ten Teije, S. Miksch, and P. Lucas, 253–62. Amsterdam: IOS Press. https://doi.org/10.3233/978-1-58603-873-1-253.

Verification of Medical Guidelines with KIV

Michael Balser and **Jonathan Schmitt** and **Wolfgang Reif**[1]

Abstract. Medical guidelines are useful to standardize health care. As thousands of patients are treated according to these guidelines, the quality of guidelines is an important issue. In this paper, we present the first tool which directly supports interactive verification of medical guidelines.[2]

1 Introduction

There is a tendency in the medical domain to standardize health care. This is because the amount of medical studies conducted per year has long passed a point, where every doctor can keep track of all results and also judge their quality. Guidelines represent a summary of the best evidence concerning the interventions to manage a particular clinical condition. As thousands of patients are treated according to these guidelines, the quality of guidelines is an important issue.

Up to now, medical guidelines are informal documents. An informal text possibly contains inconsistencies and is often incomplete. Guideline development groups employ the AGREE instrument to ensure the quality of the guideline development *process*. Our goal is to provide instruments to improve the quality of the *contents* of a guideline. In a first step, we model the guideline in a formal language. Already, the modeling process points to errors in the informal guideline. In a second step, we formally verify properties of the guideline to validate the formal model and to further improve the quality of the original guideline. Formal verification of properties requires sophisticated tool support in practice.

In this paper, we present the first tool which supports interactive verification of medical guidelines. We have added proof support for guidelines modeled in the Asbru language to the KIV theorem prover [4]. *Challenges which have been solved are as follows. (I) Instead of encoding Asbru medical guidelines in a different formalism already supported in the theorem prover, we have defined the Asbru semantics in KIV. Asbru plans translate almost one-to-one to the interactive theorem prover. (II) The intuitive strategy of symbolic execution as proof method has been carried over to reasoning in Asbru. As a consequence, proofs are straightforward and can be automated to a large extend. This is important for verification in practice.*

A number of languages exist to model medical guidelines. Asbru [7], EON [8], GLIF [9], GUIDE [11], PRODIGY [10], and PROforma [6]. We have chosen Asbru, because Asbru allows for complex timing conditions and because a formal semantics has been defined [2]. As an alternative to interactive verification, automatic techniques, especially model checking, can be used to analyze medical guidelines. We have already utilized model checking for the verification of guidelines in [5]. Interactive verification complements the use of model checking to also verify large and complex medical guidelines.

[1] University of Augsburg, Germany, email: {balser, schmitt, reif}@informatik.uni-augsburg.de

[2] This work has been partially funded by the European Commission's IST program, under contract number IST-FP6-508794 Protocure II.

2 Medical Guidelines in KIV

2.1 Asbru modeling language

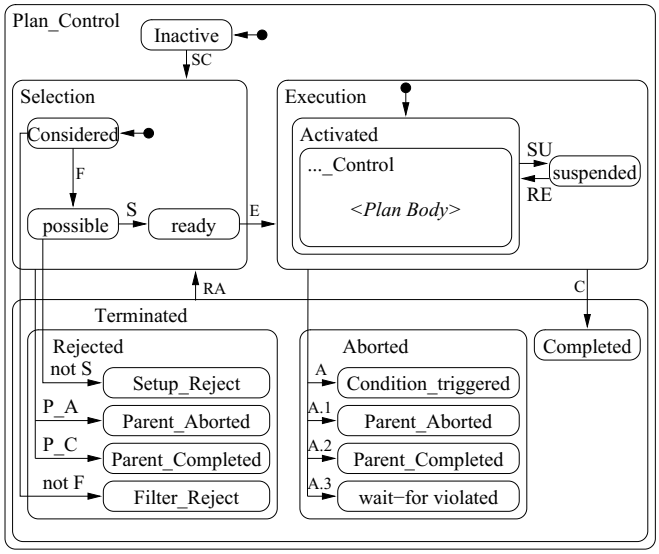

Figure 1. Plan state model of a single Asbru plan

Asbru [7] is a plan oriented language. Several plans are organized in a hierarchy of plans. A plan can refer to a number of sub plans in its plan body. Plan execution of a single Asbru plan adheres to a so called plan state model, which is visualized in Figure 1. Filter and setup conditions (F and S) are used to control the applicability of a plan, abort and complete conditions (A and C) are used to monitor execution. As a specialty of Asbru, conditions can be monitored over time according to so called time annotations. The sub plans in the plan body can be organized using different body types (e.g., sequential, any-order, or parallel). The current state of a plan – especially if a plan has been rejected, aborted, or completed – is propagated according to the plan hierarchy to its super

and sub plans. If a plan is mandatory, it must be completed, otherwise it may also be rejected or aborted. The semantics of Asbru is formally defined in [2].

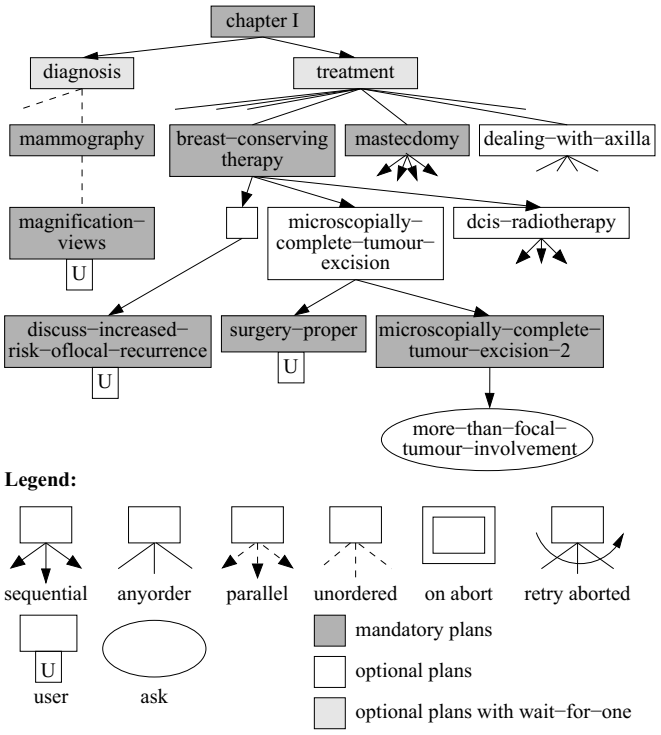

Figure 2. Part of chapter 1 of 2002 version of Dutch breast cancer guideline

Figure 2 shows part of the Asbru model of the (outdated) 2002 version of a Dutch breast cancer guideline. The top-level Asbru plan sequentially executes diagnosis and treatment. The different treatment options, breast-conserving therapy, mastectomy, treatment of the axilla, and others, are executed in any order. The different options are guarded by filter conditions (not shown in Figure 2). The treatments are further refined. Overall, the Asbru model of the first chapter consists of about 30 plans.

2.2 Asbru in KIV

KIV [4] is a tool for formal systems development. It provides strong proof support for all validation and verification tasks. KIV can handle large scale formal models by efficient proof techniques, multi-user support, and an ergonomic user interface. The specification language of KIV is based on higher-order algebraic specifications. Special support has recently been added to verify concurrent systems in temporal logic [1] [3].

In order to verify medical guidelines in KIV, an algebraic specification has been constructed for the Asbru language. This specification formally defines the semantics of Asbru. Based on this specification, an Asbru guideline can be directly translated to KIV. Figure 3 gives an example Asbru plan in KIV. The equation defines plan breast-conserving-

```
  asbru(bct)
= mk-asbru-def(
    /* filter condition with default time annotation */
    mk-aasbruc(
      mk-acond(λ pd. pd.patient-prefers-bct, default-ata),
      false, false),
    /* other conditions as default conditions */
    setup_condition,
    suspend_condition, resume_condition,
    abort_condition, complete_condition,
    /* sequential body, aborted sub plans are not retried */
    sequential, false,
    /* list of sub plans */
    bct1 + mcte + dcis + [],
    /* sub plans are optional, body waits for optional */
    wait-for-n(0, []), false);
```

Figure 3. Example Asbru plan in KIV

therapy (bct). The plan is selectable, if patient-prefers-bct evaluates to true for a given patient data *pd* (filter condition). Other conditions are defined as default conditions. The plan body contains three sub plans which are executed in sequential order. The sub plans are optional. We have taken special care to define Asbru plans in KIV such that translation from Asbru to KIV is almost one-to-one. Translation is fully automatic. The algebraic specification for Asbru defines the formal semantics of functions mk-asbru-def, mk-aasbruc, etc. More details can be found in [12].

Properties can be expressed in full first order (linear) temporal logic. Proof support has been added based on symbolic execution with induction: an Asbru guideline is executed step by step and induction is applied if execution loops.

3 Example

Figure 4 displays the structured algebraic specification of the first chapter of the Dutch breast cancer guideline in KIV. Every node in the upper half of the graph defines one or more Asbru plans. In the lower half, the data types, e.g., tumor and diagnosis scale, are described. Standard data types such as integers and lists are imported from a library. The sub tree *Asbru-Abstracted* defines the semantics of the Asbru language.

An Asbru guideline defines a concurrent system, therefore temporal logic is used to express properties. An example proof obligation is shown in Figure 5. Initially, the Asbru plan mcte is inactive and the consider signal has been received. The procedure inactive#(...) defines the system behavior for an inactive Asbru plan. This procedure implements the plan state model of Figure 1 and defines the system behavior as a relation between unprimed and primed temporal variables. We assume that the environment does not interfere with the internal state of plan mcte and all its sub plans. An environment assumption is expressed as a relation between primed and double primed temporal variables. Our goal is to verify that always if mcte remains activated then the parameter more-than-focal-tumour-involvement is false.

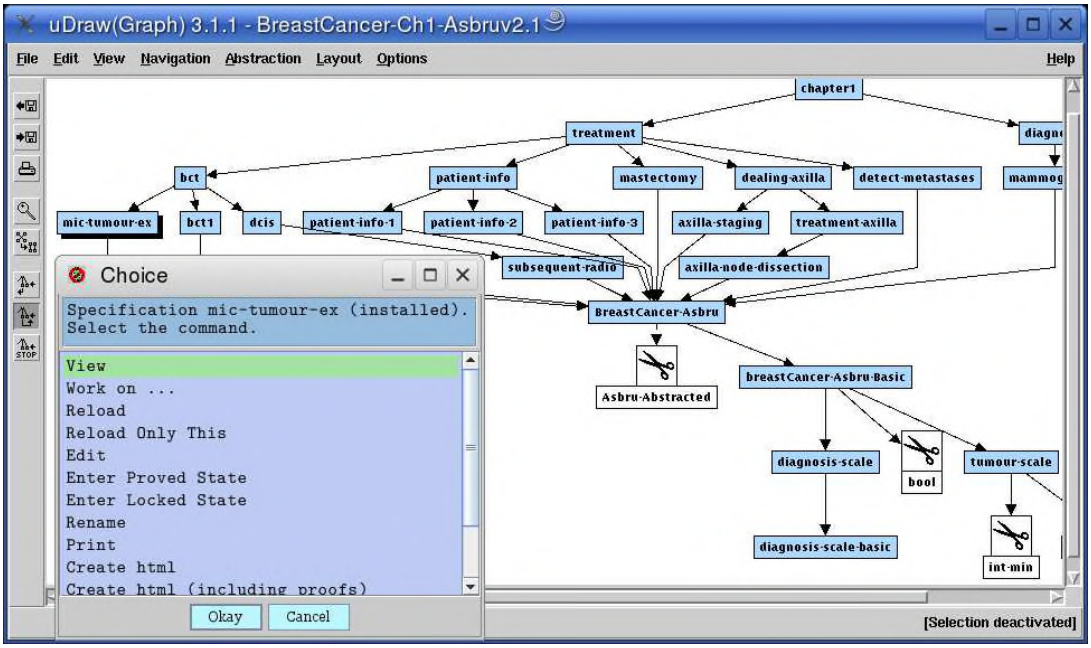

Figure 4. Development graph with specification of Breast Cancer guideline

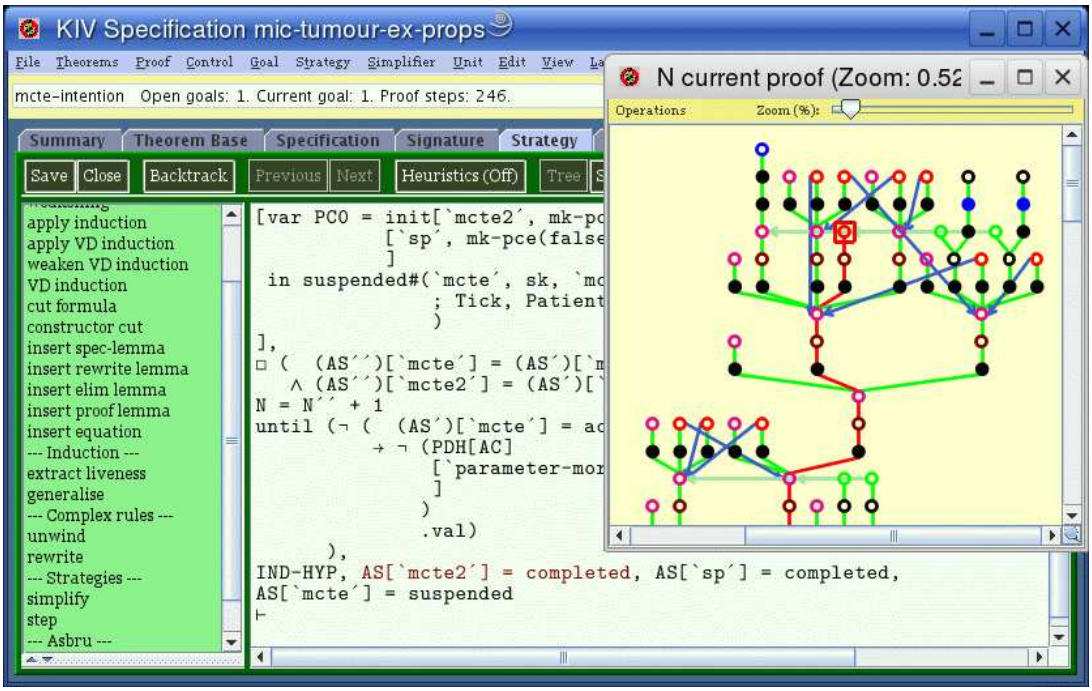

Figure 6. Example proof in KIV

$$
\begin{array}{l}
\quad /^{*}\ \textit{system configuration}\ ^{*}/ \\
\quad AS[\text{mcte}] = \text{inactive}, PC[\text{mcte}].\text{consider}, \\
\quad /^{*}\ \textit{system behaviour}\ ^{*}/ \\
\quad [\text{inactive}\#(\text{mcte}, sk; \mathit{Tick}, PDH, AC, AS, PC, \ldots)], \\
\quad /^{*}\ \textit{environment assumption}\ ^{*}/ \\
\quad \Box\ (AS''[\text{mcte}] = AS'[\text{mcte}] \wedge AS''[\text{sp}] = AS'[\text{sp}] \wedge \ \ldots) \\
\vdash\ /^{*}\ \textit{property}\ ^{*}/ \\
\quad \Box\ (\quad AS'[\text{mcte}] = \text{activated} \wedge \ \ldots \\
\qquad \rightarrow \neg\ PDH[AC].\text{more-than-focal-tumour-involvement})
\end{array}
$$

Figure 5. Example proof obligation in KIV

In order to prove the property, the Asbru plan is executed step by step. In the first state, plan mcte is inactive. Execution adheres to the plan state model of Figure 1. After the first transition, plan mcte is considered. If the filter condition F is satisfied, then the plan is possible in the next state, otherwise it is rejected. If the plan is activated, the sub plans are considered, and so on. Execution results in a proof tree depicted on the right hand side of Figure 6. The final tree represents all of the possible paths of execution. For infinite execution paths, induction is used such that the overall proof tree is always finite. During execution, the current state is displayed in the large area of the main proof window. In Fig. 6, plan mcte is currently suspended while execution of the sub plans have been completed. On the left hand side of the main proof window, a list of applicable proof rules are displayed, the most important rules being *step* to execute a system step and induction rules to reason with loops. System execution is automatic to a large extent, however, invariants for induction must be manually provided.

In contrast to interpretation of Asbru guidelines, the proof engine accepts symbolic values for the variables. For example, the patient state can be described with arbitrary first order formulas. Thus, proofs can be constructed for arbitrary patient input. KIV offers strong support for reasoning in first order logic. Details on verifying Asbru in KIV can be found in [12].

4 Conclusion

In KIV, we have implemented support for the verification of medical guidelines modeled in Asbru. The proof support is based on an intuitive proof strategy: symbolic execution with induction. The proof method is automated to a large extent. The guidelines are executed step by step and induction is used to reason about loops. Large guidelines can be verified including complex data types and time annotations. This complements the use of model checking to easily verify simple properties of smaller guidelines with state-finite data types.

The implementation has been evaluated in several case studies: a guideline from the American Academy of Pediatrics for the treatment of jaundiced newborns, a Dutch general practitioners guidelines for diabetes mellitus type II, and a Dutch guideline for the treatment of breast cancer. With KIV, we have found a number of errors in the Asbru models, in the formal semantics of Asbru, and in the KIV implementation. We are currently working on a modular approach based on assumption guarantee reasoning.

A detailed comparison of model checking and interactive verification of medical guidelines remains for future investigation.

REFERENCES

[1] M. Balser. *Verifying Concurrent System with Symbolic Execution – Temporal Reasoning is Symbolic Execution with a Little Induction*. PhD thesis, University of Augsburg, Augsburg, Germany, 2005.

[2] M. Balser, C. Duelli, and W. Reif. Formal semantics of Asbru – An Overview. In *Proceedings of IDPT 2002*. Society for Design and Process Science, 2002.

[3] M. Balser, C. Duelli, W. Reif, and G. Schellhorn. Verifying concurrent systems with symbolic execution. *Journal of Logic and Computation*, 12(4):549–560, 2002.

[4] M. Balser, W. Reif, G. Schellhorn, K. Stenzel, and A. Thums. Formal system development with KIV. In T. Maibaum, editor, *Fundamental Approaches to Software Engineering*, number 1783 in LNCS. Springer-Verlag, 2000.

[5] Simon Bäumler, Michael Balser, Andriy Dunets, Wolfgang Reif, and Jonathan Schmitt. Verification of medical guidelines by model checking – a case study. In *SPIN conference proceedings*, 2006. to appear.

[6] J. Fox, N. Johns, and A. Rahmanzadeh. Disseminating medical knowledge – the PROforma approach. *Artificial Intelligence in Medicine*, 14, 1998.

[7] 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. v. Harmelen, C. Pierret-Golbreich, I. Filby, and N. Wijngaards, editors, *7th Workshop on Knowledge Engineering: Methods & Languages (KEML-97)*. Milton Keynes, UK, 1997.

[8] M. Musen, S. Tu, A. Das, and Y. Shahar. EON: A component-based approach to automation of protocol-directed therapy. *Am Med Inform Assoc*, 3, 1996.

[9] L. Ohno-Machado, J. Gennari, S. Murphy, N. Jain, S. Tu, D. Oliver, and E. Pattison-Gordon. The guideline interchange format: a model for representing guidelines. *Am Med Inform Assoc*, 5, 1998.

[10] I. Purves, B. Sugden, N. Booth, and M. Sowerby. The PRODIGY project–the iterative development of the release one model. In *Proc AMIA Symp*, 1999.

[11] S. Quaglini, M. Stefanelli, A. Cavallini, G. Micieli, C. Fassino, and C. Mossa. Guideline-based careflow systems. *Artifical Intelligence in Medicine*, 20, 2000.

[12] J. Schmitt, M. Balser, and W. Reif. Asbru in KIV v2.1 – a tutorial. Technical Report 2006-03, University of Augsburg, 2006.