Experiences in the formalisation and verification of medical protocols

Mar Marcos[1], Michael Balser[2], Annette ten Teije[3], Frank van Harmelen[3],
Christoph Duelli[2]

[1] Universitat Jaume I, Dept. of Computer Engineering and Science
Campus de Riu Sec, 12071 Castellón, Spain
[2] Universität Augsburg, Lehrstuhl Softwaretechnik und Programmiersprachen
86135 Augsburg, Germany
[3] Vrije Universiteit Amsterdam, Dept. of Artificial Intelligence
De Boelelaan 1081a, 1081HV Amsterdam, Netherlands

Abstract. Medical practice protocols or guidelines are statements to assist practitioners and patient decisions about appropriate health care for specific circumstances. In order to reach their potential benefits, protocols must fulfill strong quality requirements. Medical bodies worldwide have made efforts in this direction, mostly using informal methods such as peer review of protocols. We are concerned with a different approach, namely the quality improvement of medical protocols by formal methods. In this paper we report on our experiences in the formalisation and verification of a real-world medical protocol. We have fully formalised a medical protocol in a two-stage formalisation process. Then, we have used a theorem prover to confirm whether the protocol formalisation complies with certain protocol properties. As a result, we have shown that formal verification can be used to analyse, and eventually improve, medical protocols.

1 Introduction

Medical practice protocols or guidelines[1] are "systematically developed statements to assist practitioners and patient decisions about appropriate health care for specific circumstances" [1]. They contain more or less precise recommendations about the diagnosis tests or the interventions to perform, or about other aspects of clinical practice. These recommendations are based on the best empirical evidence available at the moment. Among the potential benefits of protocols, we can highlight the improvement of health-care outcomes [2]. In fact, it has been shown that adherence to protocols may reduce the costs of care upto 25% [3]. In order to reach their potential benefits, protocols must fulfill strong quality requirements. Medical bodies worldwide have made efforts in this direction, e.g. elaborating appraisal documents that take into account a variety of aspects, of both protocols and their development process (see e.g. [4]). However, these initiatives are not sufficient since they rely on informal methods and notations.

We are concerned with a different approach, namely the quality improvement of medical protocols through formal methods. Currently, protocols are described using a

---

[1] In this paper we use the terms guideline and protocol indistinctively. However, the term protocol is in general used for a more specific version of a guideline.

combination of different formats, e.g. text, flow diagrams and tables. The idea of our work is translating these descriptions into a more formal language, with the aim of analysing different protocol properties. In addition to the advantages of such kind of formal verification, making these descriptions more formal can serve to expose problematic parts in the protocols.

In this paper we report on our experiences in the formalisation and verification of a medical protocol for the management of jaundice in newborn babies. This work is part of the IST Protocure[2] project, a recently concluded project which has consisted in the assessment of the application of formal methods for protocol quality improvement.

The formalisation of medical protocols can be tackled at different degrees. Since we aim at a formal verification, we have chosen the logic of a theorem prover –KIV [5]– as target formalism. Prior to the KIV formalisation step, we have carried out a modelling step using a specific-purpose knowledge representation language for medical protocols –Asbru [6]. This gradual formalisation strategy has made the formalisation task feasible, which in turn has enabled us to use theorem proving.

The structure of this paper roughly follows the *Asbru modelling-KIV formalisation-KIV verification* process that the protocol has undergone. First section 2 introduces the jaundice protocol. Then section 3 describes the Asbru language and the model of the protocol in this language. The next step has been the translation of the Asbru protocol into the formal notation of KIV. Section 4 describes this step, and section 5 presents the results of the subsequent verification step. Finally, section 6 concludes the paper.

2   The jaundice protocol

Jaundice, or hyperbilirubinemia, is a common disease in newborn babies which is caused by elevated bilirubin levels in blood. Under certain circumstances, high bilirubin levels may have detrimental neurological effects and thus must be treated. In many cases jaundice disappears without treatment but sometimes phototherapy is needed to reduce the levels of total serum bilirubin (TSB), which indicates the presence and severity of jaundice. In a few cases, however, jaundice is a sign of a severe disease.

The jaundice protocol of the American Association of Pediatrics [7] is intended for the management of the disease in healthy term[3] newborn babies. The guideline is a 10 pages document which contains knowledge in various notations: the main text; a list of factors to be considered when assessing a jaundiced newborn; two tables, one for the management of healthy term newborns and another for the treatment options for jaundiced breast-fed ones; and a flowchart describing the steps in the protocol.

The protocol consists of an evaluation (or diagnosis) part and a treatment part, to be performed in sequence. During the application of the protocol, as soon as the possibility of a more serious disease is uncovered, the recommendation is to exit without any further action.

3   Modelling the jaundice protocol in Asbru

In the first stage of protocol formalisation we have used a specific-purpose knowledge representation language. Different languages have been proposed to represent medical

---

[2] http://www.protocure.org/

[3] Defined as 37 completed weeks of gestation.

protocols and their specific features (see [8]). Most of them consider protocols as a composition of actions to be performed and conditions to control these actions [9]. However, although the trend is changing lately, many of the protocol representation languages in the literature are not formal enough. For instance, they often incorporate many free-text elements which do not have clear semantics. Exceptions to this are PROforma [10] and Asbru [6]. In this work we have used Asbru, mainly because it is more precise in the description of a variety of medical aspects.

3.1 Asbru: a knowledge representation language for protocols

The main aspects of Asbru are: (i) in Asbru a medical protocol is considered as a plan skeleton with sub-plans in the sense of AI planning, (ii) it is possible to specify the intentions of a plan in addition to the actions of a plan, (iii) it is possible to specify a variety of control structures within a plan, and (iv) it provides a rich language to specify time annotations. Below we give a short description of the main constructs of the Asbru language (see [6] for more details).

A medical protocol is considered in Asbru as a hierarchical **plan**. The main components of a plan are intentions, conditions, effects, and plan-body. Furthermore, a plan can have arguments and has the possibility to return a value. Next we briefly discuss some of these components.

Intentions are the high-level goals of a plan. Intentions can be expressed in terms of achieving, maintaining or avoiding a certain state or action. Such states or actions can be intermediate or final (overall). For example, the label "achieve intermediate-state" means that sometime during the execution of the plan, a certain state must be achieved. In total there are twelve possible forms of intention: [achieve/maintain/avoid] [intermediate/overall]-[state/action].

A variety of **conditions** can be associated with a plan, which define different aspects of its execution. The most important types of conditions are the following:

– filter conditions, which must be true before the plan can be started.
– abort conditions, which define when a started plan must be aborted.
– complete conditions, which define when a started plan can complete successfully.
– activate conditions, with possible values "manual" or "automatic". If the activate mode is manual, the user is asked for confirmation before the plan is started.

The **plan-body** contains the actions and/or sub-plans to be executed as part of the plan. The main forms of plan-body are the following:

– user-performed: an action to be performed by the user, which requires user interaction and thus is not further modelled.
– single step: an action which can be either an activation of a sub-plan, an assignment of a variable, a request for an input value or an if-then-else statement.
– subplans: a set of steps to be performed in a given order. The possibilities are: in sequence (sequentially), in parallel (parallel), in any possible sequential order (any-order), and in any possible order, sequential or not (unordered).
– cyclical plan: a repetition of actions over time periods.

In the case of subplans, it is necessary to specify a waiting strategy, which describes the plans that must be completed so that the parent plan can be considered successfully

completed. It is possible to specify e.g. whether all the subplans should be executed ("wait-for ALL") or not (e.g. "wait-for ONE", or "wait-for" some specific plan).

Time annotations can be associated to different Asbru elements (e.g. intentions and conditions). A time annotation specifies (1) in which interval things must start, (2) in which interval they must end, (3) their minimal and maximal duration, and (4) a reference time-point. The general scheme for a time annotation is ([EarliestStarting, LatestStarting] [EarliestFinishing, LatestFinishing] [MinDuration, MaxDuration] REF-ERENCE). Any of these elements can be left undefined, allowing for uncertainty in the specification of time annotations.

3.2 Asbru model of jaundice protocol

Like the original document, the Asbru model of jaundice protocol has as main components a diagnostics part and a treatment part. It is made up of about 40 plans and has a length of 16 pages in a simplified Asbru notation. Figure 1 shows the overall structure of the protocol as a hierarchy of plans.

The treatment phase, in which we focus here, consists of two parallel parts, namely the actual treatments and a cyclical plan asking for the input of new age and TSB values every 12 to 24 hours. Regarding the treatments (label (-) in figure 1), either the regular ones ("Regular-treatments") or an exchange transfusion ("Exchange-transfusion") can take place depending on the bilirubin level. The "Regular-treatments" plan contains the main treatment procedure. It consists of two parts to be performed in any possible order (unordered): the study of feeding alternatives and the different therapies (see label (*)). The plans in group (*) can be tried in any order, one at a time.

Figure 2 shows the "Phototherapy-intensive" plan, which describes one of the therapies. Its plan-body simply contains a sub-plan activation pointing to a user-performed action. One of its intentions is attaining normal (or "observation") bilirubin levels. It also contains different conditions, e.g. one of the abort conditions specifies that the plan should abort as soon as it fails to reduce the bilirubin levels in 4 hours.

## 4 Formalising the jaundice protocol in KIV

In the second stage of the formalisation process we have used the KIV verification tool [5]. KIV is an interactive theorem prover with strong proof support for higher-order logic and elaborate heuristics for automation. Currently, special proof support for temporal logic and parallel programs is being added. In contrast to fully automatic verification tools, the use of KIV interactive tool allows for the verification of large and complex systems, as it has been shown by its application to a number of real-world systems (distributed systems, control systems, etc).

4.1 KIV

KIV supports the entire software development process, i.e. the specification, the implementation and the verification of software systems. Next we briefly describe the relevant aspects of KIV for Asbru specification and verification needs.

For **specification**, three aspects are important: specifications can be structured, and both functional and operational system aspects can be described. A specification is broken down into smaller and more tractable components using structuring operations such

Fig. 1. *Overview of the jaundice protocol in Asbru.* The main entry point of the protocol is the "Diagnostics-and-treatment-hyperbilirubinemia" plan –the three "Check-for-..." plans are Asbru artifacts to model a continuous monitoring of TSB level and two check-ups at temporally specified intervals. The plan "Diagnostics-and-treatment-hyperbilirubinemia" is divided into a diagnostics and a treatment subplan, to be executed sequentially.
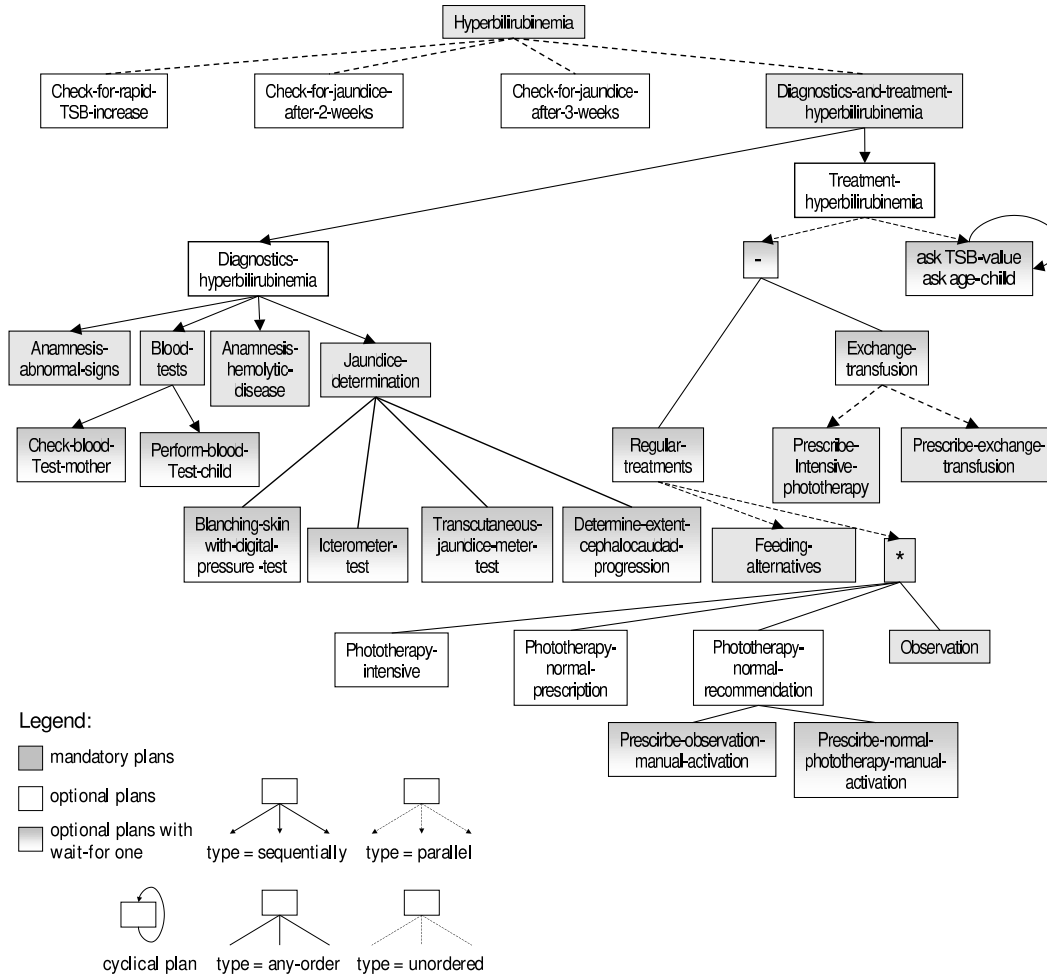
as union and enrichment, that can be used to combine more simple specifications. For functional aspects, algebraic specifications are used to specify abstract data types.

Fig. 2. *"Phototherapy-intensive" plan.*

plan Phototherapy-intensive

    intentions
        achieve overall-state: (bilirubin = observation)
        maintain intermediate-state: (and
            (TSB-decrease = yes) *in* ([4h, -] [-, 6h] [-, -] SELF)
            (TSB-change $\geq$ 1) *in* ([4h, -] [-, 6h] [-, -] SELF) **)**
    conditions
        filter-precondition: (or (bilirubin = phototherapy-intensive) *in* NOW
            `normal-phototherapy-failure`**)**
        abort-condition: (or (and (bilirubin $\neq$ phototherapy-intensive) *in* NOW
            (not `normal-phototherapy-failure`**)** */* and */*
        `intensive-phototherapy-failure`:
        (and (bilirubin = phototherapy-intensive) *in* NOW
            (or (TSB-decrease = no) *in* ([4h, -] [-, -] [-, -] SELF)
            . . .**)** */* or */*
) */* and */*
) */* abort condition */*
    plan-body
    Prescribe-intensive-phototherapy

Complex operational behaviour can be specified using parallel programs. Programs in KIV can contain assignments ($v := \tau$), conditionals (**if** φ_{pl} **then** ψ_1 **else** ψ_2), loops (**while** φ_{pl} **do** ψ), local variables (**var** $v = \tau$ **in** ψ), nondeterministic choices (**choose** φ **or** ψ), interleaving ($\varphi \parallel \psi$) and synchronisation points (**await** φ_{pl}). For a better support of Asbru, additional basic constructs have been implemented: interrupts (**break** ψ **if** φ_{pl}), for modelling different plan conditions; and synchronous parallel execution ($\varphi \parallel_s \psi$), as well as any-order execution ($\varphi \parallel_a \psi$), for a more direct translation of plan-bodies. With the help of these constructs, the main features of Asbru can be directly translated. Others still need to be encoded using additional program variables.

Concerning the **verification**, we use a variant of Interval Temporal Logic (ITL) [11] to formulate properties. This logic is first-order and allows finite and infinite intervals. Here we restrict ourselves to the temporal operators always ($\Box \varphi$), eventually ($\Diamond \varphi$), next ($\circ \varphi$), and **laststep** –which is true only in the last step of an interval. Single transitions are expressed as first-order relations between unprimed and primed variables, where the latter represent the value of the variable in the next state. For example, the formula $v = 0 \wedge (\Box v' = v + 1) \to \Diamond v = n$ states that, if variable v is initially 0, and the value v' in the next state is always incremented by one, then eventually the variable will be equal to an arbitrary natural number n. Finally, the proof technique for verifying parallel programs in KIV is symbolic execution with induction.

4.2 KIV formalisation of jaundice protocol

In order to formally analyse Asbru plans in a first attempt, we have translated them into parallel programs. The translation of the Asbru model into KIV has been done in a structure-preserving way, by mapping each Asbru plan into a KIV specification containing a parallel program. Thus, the structure of the jaundice protocol in KIV roughly

mirrors the Asbru model in figure 1. This is one of the key ideas of our work, because it gives the possibility to obtain some feedback from the formalisation and verification phases in terms of the Asbru model, and to exploit this structure during proof attempts. Table 1 shows some of the patterns that we used in the translation of Asbru plans.

Table 1. *Translation patterns of some Asbru constructs into KIV.*

Asbru	KIV
filter precondition φ NOW body	**await** φ; body
filter precondition φ body	**if** φ **then** body
complete condition φ body	**break** body **if** φ
abort condition φ body	**break** body **if** φ
<<name>> (*plan activation*)	<<name>>#(...) (*procedure call*)
do type=sequentially P1,... Pn	P1;... Pn
do type=any-order P1,... Pn	P1 $\|_a$ ... Pn
wait-for Pi body	**break** body **if** *some expression on* Pi-state

In many cases the KIV translation closely follows the structure of the original Asbru plan, except for small details. Other translations, however, needed additional encodings to represent the Asbru elements not directly supported by KIV. The example in figure 3, corresponding to the plan "Phototherapy-intensive", serves to illustrate the kind of translations that we have obtained. This translation includes an await construct to

Fig. 3. *KIV translation of "Phototherapy-intensive" plan.*

Phototherapy-intensive#(**var** phototherapy-normal-prescription-activated, patient-data,
    time, phototherapy-intensive-activated)
begin
  await   get-bilirubin(patient-data.tsb) = phototherapy-intensive
        $\vee$ get-bilirubin(patient-data.tsb) = phototherapy-normal
          $\wedge$ phototherapy-normal-prescription-activated $\neq \bot$
          $\wedge\ 4 \leq$ time - phototherapy-normal-prescription-activated.value
          $\wedge\ \neg$ get-decrease(patient-data.tsb);
  phototherapy-intensive-activated := mk-value(time);
  break
    prescribe-intensive-phototherapy#(; time)
  if   get-bilirubin(patient-data.tsb) $\neq$ phototherapy-intensive
   $\vee$ get-bilirubin(patient-data.tsb) = phototherapy-intensive
      $\wedge$ ($4 \leq$ time - phototherapy-intensive-activated.value
          $\wedge\ \neg$ get-decrease(patient-data.tsb)
       $\vee \ldots$)
end

model the filter preconditions of the plan, as well as an interrupt (break) to monitor the conditions under which the plan should abort. The KIV plan also shows the way in which time annotations can be encoded, with the help of additional variables holding the time at which a plan has been activated (such as photography-intensive-activated).

5 Verifying the jaundice protocol in KIV

After the formalisation of the jaundice protocol, we have worked on the verification of several protocol properties using the KIV system. Protocol properties are expressed in the previously introduced variant of ITL. For instance, $Phototherapy-intensive\#(\ldots) \wedge (\Box\ time'' = time' + 1) \rightarrow \Diamond$ **laststep** states that, if the program $Phototherapy - intensive$ is executed, then execution will eventually reach the last step, i.e. it terminates. However, the plan only terminates under the additional assumption that time is incremented by one in each step.

As part of Protocure project, we identified a number of protocol properties which were deemed important from a medical point of view. We distinguished properties at the *implementation level*, dependent on the Asbru language, from properties at the *conceptual level*, protocol-dependent. Properties at the conceptual level reflect the verification needs in a practical application and hence have been the target of our formal verification. Among them, the correctness of plan intentions and the properties making reference to indicators were deemed of particular interest. The correctness of intentions aims at ensuring that the intentions of a plan follow from its body. As for the properties about indicators, the goal is verifying that the protocol results in actions that comply with certain quality criteria defined either in the protocol itself or by external sources. In our case, we have exploited the indicators for jaundice treatment defined by the MAJIC (Making Advances against Jaundice in Infant Care) Committee [12].

5.1 Verification of one intention of "Phototherapy-intensive" plan

One of the intentions of the plan "Phototherapy-intensive" is the reduction of the TSB levels in at least 1 mg/dl within 4 to 6 hours. We can view this as a property that the plan should satisfy, i.e. while executing the plan, there should be such a decrease. This property was initially translated to the following ITL formula:

$$
\begin{aligned}
\Box\quad &\textbf{laststep} \\
\vee\quad &\textit{pti-state} \neq \text{activated} \wedge \textit{pti-state}' = \text{activated} \\
\rightarrow\quad &\qquad \textit{/* time annotation */} \\
&\qquad 4 \leq \textit{time} - \textit{pti-acttime}\ .\text{value} \\
&\qquad \wedge\ \textit{time} - \textit{pti-acttime}\ .\text{value} \leq 6 \\
&\quad \rightarrow \textit{/* property */} \\
&\qquad \text{get-decrease}(\textit{pd}\ .\text{tsb}, \textit{pti-acttime}\ .\text{value}) \\
&\qquad \wedge\ \text{get-change}(\textit{pd}\ .\text{tsb}, \textit{pti-acttime}\ .\text{value}) \geq 1 \\
\textbf{unless}\ &\textit{pti-state}' \neq \text{activated}
\end{aligned}
$$

Informally, this formula says that, if "Phototherapy-intensive" is activated, when 4 to 6 hours have elapsed, there should be a decrease in TSB levels greater or equal than 1.

The property was successfully proved in KIV, in a fully automatic proof. However, the proof was successful only after a number of proof attempts. These attempts uncovered errors in the formulation of the intention, which led to an improved formal semantics for Asbru intentions. This property was somehow "easy" to prove, because the

intention is enforced by one of the abort conditions of the plan. In the next section we present a more difficult proof, which required identifying the conditions under which the property should hold. As we will see, these conditions describe the most usual cases of newborn jaundice.

5.2 Verification of MAJIC indicator #7

The MAJIC indicators that appear in [12] have been refined by the same organisation into different medical review criteria for the evaluation and treatment of jaundice. This includes a set of 11 criteria which jaundice protocols must comply with. Among these criteria, we selected indicator #7, which is stated as follows:

INCLUSIONS If any phototherapy initiated
CRITERIA No more than one serum bilirubin level drawn after phototherapy is discontinued

The rationale of this indicator is beyond the scope of this work. It was translated into the following temporal formula:

$$\square \quad \textbf{laststep}$$
$$\vee \quad pd \text{ .under-phototherapy} \wedge \neg\, pd' \text{ .under-phototherapy}$$
$$\wedge\, pd \text{ .tsb} = TSB_0$$
$$\rightarrow \square\, pd \text{ .tsb} = TSB_1 \wedge TSB_1 \neq TSB_0 \rightarrow \square\, pd \text{ .tsb} = TSB_1$$

Informally, when phototherapy is discontinued, if another TSB value is measured, then there will not be more TSB measurements (TSB history will stay the same).

Several attempts were made to prove that "Regular-treatments" complies with this indicator, which uncovered some problems in the formalisation of the protocol. This insight was used to enhance the translation patterns for Asbru plans. Finally, it was proved that the property does not hold. A counter example was found, which consists in applying phototherapy once and then doing observation for more than 24 hours, allowing "Treatment-hyperbilirubinemia" to measure TSB twice. The analysis of proof attempts served to identify the assumptions under which the indicator should be satisfied: (1) If phototherapy is discontinued, bilirubin levels are normal (or "observation"); (2) Observation plan will run for less than 24 hours; and (3) After phototherapy and observation, bilirubin levels will still be "observation". These assumptions were given to medical experts for review, who concluded that they capture the most usual cases, i.e. for most of newborns the assumptions hold and the protocol satisfies the indicator. Thus the assumptions explicitly define the cases in which the indicator is satisfied. They could be used to improve the original protocol, e.g. to document the cases in which the indicator might not be satisfied.

6 Conclusions

In this paper we have shown that it is possible to formalise a significant piece of medical knowledge to such an extent that it can be used as the basis for formal verification, and that this verification is indeed possible. We have fully formalised a real-world medical protocol in a two-stage formalisation process. Then, we have used a theorem prover

to systematically analyse whether the formalisation complies with certain (medically relevant) protocol properties.

The most important contribution of our effort is showing that it is possible to formally analyse medical protocols. If a protocol is developed with certain goals in mind (e.g. intentions or indicators), verification can serve to check whether the protocol actually complies with them. Even if this is not the case, the verification attempts can be of help in obtaining counter examples and/or assumptions, which can be eventually used to improve the original document.

Obviously, this achievement comes at a price: a significant amount of work has been necessary for such an effort. Although we are not in a position to make strong quantitative statements, the formalisation and verification exercise reported in this paper has taken over a person-year to complete. However, this has been our first attempt in the direction of verifying medical protocols with mathematical rigour. We expect that the necessary effort should decrease in the future, e.g. with a more direct KIV support for Asbru protocols. Furthermore, we would argue that the improvement of the quality of medical practice protocols is worth additional effort.

References

1. Field, M., Lohr, K., eds.: Clinical Practice Guidelines: Directions for a New Program. National Academy Press, Washington D.C., USA (1992)
2. Woolf, S., Grol, R., Hutchinson, A., Eccles, M., Grimshaw, J.: Potential benefits, limitations, and harms of clinical guidelines. British Medical Journal **318** (1999) 527–530
3. Clayton, P., Hripsak, G.: Decision support in healthcare. Int. J. of Biomedical Computing **39** (1995) 59–66
4. AGREE Collaboration: Appraisal of Guidelines for Research & Evaluation (AGREE) Instrument (2001) Obtained in `http://www.agreecollaboration.org/`.
5. Balser, M., Reif, W., Schellhorn, G., Stenzel, K., Thums, A.: Formal system development with KIV. In Maibaum, T., ed.: Fundamental Approaches to Software Engineering. Number 1783 in LNCS, Springer (2000)
6. Shahar, Y., Miksch, S., Johnson, P.: The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. AI in Medicine **14** (1998) 29–51
7. AAP: American Academy of Pediatrics. Practice parameter: management of hyperbilirubinemia in the healthy term newborn. Pediatrics **94** (1994) 558–565
8. Elkin, P., Peleg, M., Lacson, R., Bernstam, E., Tu, S., Boxwala, A., Greenes, R., Shortliffe, E.: Toward Standardization of Electronic Guidelines. MD Computing **17** (2000) 39–44
9. Miksch, S.: Plan Management in the Medical Domain. AI Communications **12** (1999) 209–235
10. Fox, J., Johns, N., Lyons, C., Rahmanzadeh, A., Thomson, R., Wilson, P.: PROforma: a general technology for clinical decision support systems. Computer Methods and Programs in Biomedicine **54** (1997) 59–67
11. Moszkowski, B.: A temporal logic for multilevel reasoning about hardware. IEEE Computer **18** (1985) 10–19
12. MAJIC: MAJIC Steering Committee Meets. MAJIC Newsletter **1** (1998)