Improving medical protocols by formal methods

Annette ten Teije, Mar Marcos, Michael Balser, Joyce van Croonenborg, Christoph Duelli, Frank van Harmelen, Peter Lucas, Silvia Miksch, Wolfgang Reif, Kitty Rosenbrand, Andreas Seyfang

Angaben zur Veröffentlichung / Publication details:

Teije, Annette ten, Mar Marcos, Michael Balser, Joyce van Croonenborg, Christoph Duelli, Frank van Harmelen, Peter Lucas, et al. 2006. "Improving medical protocols by formal methods." Artificial Intelligence in Medicine 36 (3): 193–209. https://doi.org/10.1016/j.artmed.2005.10.006.

Improving medical protocols by formal methods☆

Annette ten Teije [a,*], Mar Marcos [b], Michel Balser [c], Joyce van Croonenborg [d], Christoph Duelli [c], Frank van Harmelen [a], Peter Lucas [e], Silvia Miksch [f], Wolfgang Reif [c], Kitty Rosenbrand [d], Andreas Seyfang [f]

[a] *Vrije Universiteit Amsterdam, Department of Artificial Intelligence, De Boelelaan 1081a, 1081HV Amsterdam, The Netherlands*
[b] *Universitat Jaume I, Department of Computer Engineering and Science, Campus de Riu Sec, 12071 Castellón, Spain*
[c] *Universität Augsburg, Lehrstuhl Softwaretechnik und Programmiersprachen, 86135 Augsburg, Germany*
[d] *Dutch Institute for Healthcare Improvement CBO, Churchilllaan 11, 3527GV Utrecht, The Netherlands*
[e] *University of Nijmegen, Institute for Computing and Information Sciences, Postbus 9010, 6500GL Nijmegen, The Netherlands*
[f] *Vienna University of Technology, Institute of Software Technology, Favoritenstraße 9-11/188, A-1040 Vienna, Austria*

Received 1 October 2004; received in revised form 21 October 2005; accepted 25 October 2005

☆ This work has been done in the framework of the EU project Protocure (IST-2001-33049 — Protocure, Improving medical protocols by formal methods). See http://www.protocure.org/ (last accessed 1 October 2005).
* Corresponding author. Tel.: +31 5987721; fax: +31 5987653.
E-mail address: annette@cs.vu.nl (A. ten Teije).

1. Introduction

During the last decade, evidence-based medicine has given rise to an increasing number of clinical practice guidelines and protocols.[1] Medical guidelines are ''systematically developed statements to assist practitioners and patient decisions about appropriate health-care for specific circumstances'' [1]. They provide clinicians with health-care recommendations based on valid and up-to-date empirical evidence. In this way, guidelines facilitate the spreading of high-standard practices that otherwise would have much less impact. Moreover, it has been proved that adherence to guidelines and protocols may reduce health-care costs up to a 25% [2].

A high number of medical guidelines have been published in the literature and on internet, making them more accessible. However, the work done on developing and disseminating guidelines far outweighs the efforts on guaranteeing their quality. Indeed, anomalies like ambiguity and incompleteness are frequent in medical guidelines [3]. Furthermore, guidelines can be inconsistent because of the lack of familiarity of the designers with certain principles and notations. These problems preclude the effective application of guidelines, and hinders their potential benefits.

Recent efforts have tried to address the problem of guideline quality improvement. The medical community has sought to organise and integrate guidelines into compendiums, to make them more accessible, usable and comprehensible. With the aim of ensuring a high degree of quality, the organisations promoting these initiatives have also set minimal standards for the inclusion of guidelines and protocols in the compendiums. These standards take into account, for instance, the relevance and validity of the sources employed in their development. We believe that these approaches are not sufficient, since they rely on informal processes and notations. As a result, many practical guidelines and protocols still contain ambiguous, incomplete or even inconsistent elements. Even when such ambiguity is intentional, in order to allow for differences in organisational or personal practice, it is important to state this explicitly.

An appropriate representation language, with clear and well-defined semantics, would enable the systematic verification of guidelines and protocols by formal methods. In addition, unlike the post-dissemination activities mentioned before, this approach would make quality improvement possible during the early stages of development. Research from the fields of computer science and artificial Intelligence can be of help in both the definition of an adequate representation language and the development of techniques for the formal analysis of guidelines. The representation language must give means to represent, explicitly and in a non-ambiguous way, the relevant knowledge about guidelines and protocols. Based on the formal semantics of this language, the analysis techniques should allow for the determination of consistency (no contradictions) and correctness (objectives are satisfied).

Taking this into account, the solution we suggest to the quality problem of current guidelines and protocols relies on the utilisation of formal methods. Currently, protocols are described using free-text, flowcharts and/or tables. Our proposal consists in translating these informal descriptions

[1] Although the terms protocol and guideline differ (the term protocol is in general used for a specialised version of a guideline), in this paper we use them indistinctively.

into a formal language, with the aim of analysing different protocol properties. In order to assess the feasibility of this novel approach, we have carried out a case study in the formalisation and verification of medical protocols. Throughout our case study we have shown not only that medical protocols can be expressed in terms of logic, but also that they can be subject to formal verification. Even more, we have demonstrated that both formalisation and verification can serve to detect critical flaws in existing protocols. Indeed, the idea of improving medical protocols by formal methods has shown to be both feasible in practice and useful.

This paper reports on the different phases and results of our case study. The structure of the paper is as follows. Section 2 describes our approach in terms of the steps we have carried out. The structure of the paper follows these steps. We start with the selection of the case studies in Section 3. Section 4 discusses the modelling of the selected cases in a special-purpose knowledge representation language, namely Asbru. We give the formal underpinning of Asbru in Section 5 , such that we are able to formalise our protocols in a logic in Section 6. We identify a number of properties of guidelines and give their formal counterparts (Section 7). We then report on the verification results in Section 8 , and continue with the evaluation thereof by a team of experts (Section 9). Section 10 contains related work, and finally Section 11 concludes and discusses future challenges.

2. Methodology

Our approach to solve the quality problem of medical guidelines and protocols relies on the use of formal methods. The activities through which we have carried out the assessment of this approach are the following:

1. Choose two real-world reference protocols which cover a wide variety of characteristics.
2. Formalise these reference protocols.
3. Check the formalised protocols through the verification of medically relevant protocol properties.

An evaluation of the obtained results has been carried out with the help of a panel of domain experts. The experts have categorised the uncovered errors according to their criticality, ranging from serious errors to trivial ones. In this way, the detection of either a serious error or several trivial ones can be considered as a success of our project.

The formalisation of medical protocols can be tackled at different degrees. Since we aimed at a formal verification, we chose the logic of the theorem prover KIV [4] as our target formalism. Prior to the KIV formalisation step, we have carried out a modelling step using Asbru, a specific-purpose knowledge representation language for medical protocols [5]. Asbru was selected among the existing protocol representation languages (see [6] for a review), because it allows for the description of various medical aspects in a very precise way. The gradual formalisation strategy (first Asbru, then KIV) has made the formalisation task feasible.

Fig. 1 illustrates in more detail the process of our approach. First we have selected two medical protocols which cover together a wide range of protocol characteristics (see the *selection* step in figure 1). Then the two selected protocols have undergone a gradual transformation into a formal representation. Starting from the original texts (*original protocol*in the figure), the protocols have been modelled first in the Asbru language (the *modelling* step) and then translated into the KIV formal representation (the *formalisation of protocol*). The results of this transformation process have been, respectively, a collection of Asbru plans representing the protocol (the *Asbru plans*) and a

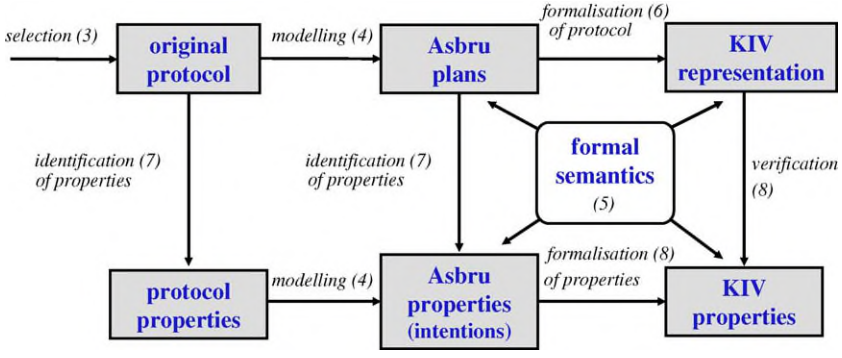

Figure 1 Formalisation and verification of protocols in the Protocure project. Numbers refer to the section where each step is discussed.

set of KIV programs encoding these plans (the *KIV representation*). In order to make this formalisation possible, a formal semantics has been defined for the main parts of the Asbru language (the *formal semantics*). This is a crucial point in the process, since the formal verification that we aim at is only possible with a precise semantics.

In parallel to the above transformation, a number of medically relevant properties has been identified from an analysis of both the original protocols and their Asbru version (the *identification of properties* step). The result of this step are the *protocol properties* and their reformulation as *Asbru properties*. Then we have selected a subset of the properties for which a formal proof seemed feasible, we formalised them (the *formalisation of properties* step), and we have carried out the necessary proofs to verify them (the *verification* step). Finally, the results of the different steps have been evaluated by a panel of medical experts, in order to assess the overall utility of our approach.

The rest of the paper reports on the main results of the process, which is structured according to the steps described above. The relevant section numbers are indicated in Fig. 1.

3. Selection of reference protocols

Concerning the protocols that are the object of our study, we selected two examples that cover a wide range of protocol characteristics. The first protocol is for the management of jaundice in healthy term newborns [7] developed by the American Academy of Pediatrics (AAP). The second protocol is for the management of diabetes mellitus type 2 [8] , developed by the Dutch Association of General Practitioners (NHG). Next we give some background information on the protocols and motivate their choice.

Medical protocols differ along several dimensions, which can be related to the contents of the protocol or to its form, amongst others. Important dimensions are the **target users** of the protocol (e.g. general practitioners (GPs) or nurses) and the **aspects of clinical practice** it covers (e.g. diagnosis and/or treatment). The **period of time** to which the protocol applies is also characteristic, varying between short time-span (hours to days) and long time-span protocols (weeks to years). Concerning the **form**, protocols are usually more or less structured textual descriptions, and sometimes combine alternative formats, such as tables, lists or flowcharts.

The two selected protocols not only come from different organisations but also differ in their characteristics. The diabetes protocol is aimed at GPs and covers both diagnosis and treatment over a long period of time. The jaundice protocol is also devoted to both diagnosis and treatment but, unlike the diabetes protocol, is intended for use in a hospital and over a short time-span. Regarding the form, the careful layout of the diabetes protocol contrasts with the rather unstructured text of the jaundice protocol. The latter, however, includes complementary descriptions such as flow diagrams describing the protocol procedure. Finally, the two protocols can be considered of high quality: one is in daily use by Dutch GPs and the other is included in the repository of the National Guideline Clearinghouse.[2] Criteria for inclusion in the NGC are that the guideline contains systematically developed statements, was produced under the auspices of medical specialty associations, is based on literature, and is developed, reviewed, or revised within the last 5 years. Furthermore, both protocols have been published in a peer-reviewed scientific journal. In summary, the chosen protocols cover different aspects while fulfilling high quality standards, which makes them good candidates for our case study.

In the following sections we will use the jaundice protocol as the main example in our explanations, and we therefore give a brief description of this protocol.

3.1. The jaundice protocol

Jaundice, or hyperbilirubinemia, is a common disease in newborns which is caused by elevated bilirubin levels in the 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). In a few cases, however, jaundice is a sign of a severe disease.

The jaundice protocol of the AAP is intended for the management of the disease in healthy term[3] newborns. The guideline is a 10 page 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 newborns; 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

[2] See http://www.guideline.gov/ (accessed October 2005)
[3] Defined as 37 completed weeks of gestation.

serious disease is uncovered, the recommendation is to exit the protocol without any further action.

We have now discussed the *selection* step from Fig. 1 , and in the next section we continue with modelling the protocol using the special-purpose language Asbru.

4. Modelling of the jaundice protocol in Asbru

Different languages have been proposed in the literature to represent medical protocols. Most of them consider protocols as a composition of actions to be performed and conditions to control them [9]. However, although the trend is changing lately, many of the protocol representation languages in the literature are not formal enough for our purposes. For instance, they often incorporate free-text elements which do not have clear semantics. Exceptions to this are PROforma [10] and Asbru [5].

In this section, we present the main features of Asbru, we report on the Asbru modelling of the protocols, and we conclude with lessons learned.

4.1. Asbru: a knowledge representation language for protocols

The main features of Asbru are: (i) medical protocols are viewed as hierarchical skeletal plans in the sense of AI planning, i.e. plans with subplans; (ii) it is possible to specify the intentions of a plan, in addition to the action composition; (iii) a wide variety of control structures can be used to specify the control within a plan; and (iv) a rich language can be used to describe temporal annotations. Below we give a short description of the main constructs of the language (see [11] for more details).

A medical protocol is considered in Asbru as a hierarchical **plan**. The main components of an Asbru plan are intentions, conditions and plan-body. We briefly discuss these components.

The **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. The states or actions to which intentions refer can be intermediate or final (overall). Thus, 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 subplans 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 subplan, 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 case a plan has subplans, in addition to the specification of the ordering (sequentially, parallel and so forth), it is necessary to specify a waiting strategy to describe which of the subplans 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 with different Asbru elements (e.g. intentions, conditions and even plan activations). A time annotation specifies: (1) in which interval things must start, (2) in which interval things must end, (3) their minimal and maximal duration, and (4) a reference time point. Any of these elements can be left undefined, allowing for the specification of incomplete time annotations. The general scheme for a time annotation is:

([EarliestStartingShift, LatestStartingShift],
[EarliestFinishingShift, LatestFinishingShift],
[MinimalDuration, MaximalDuration],
Reference)

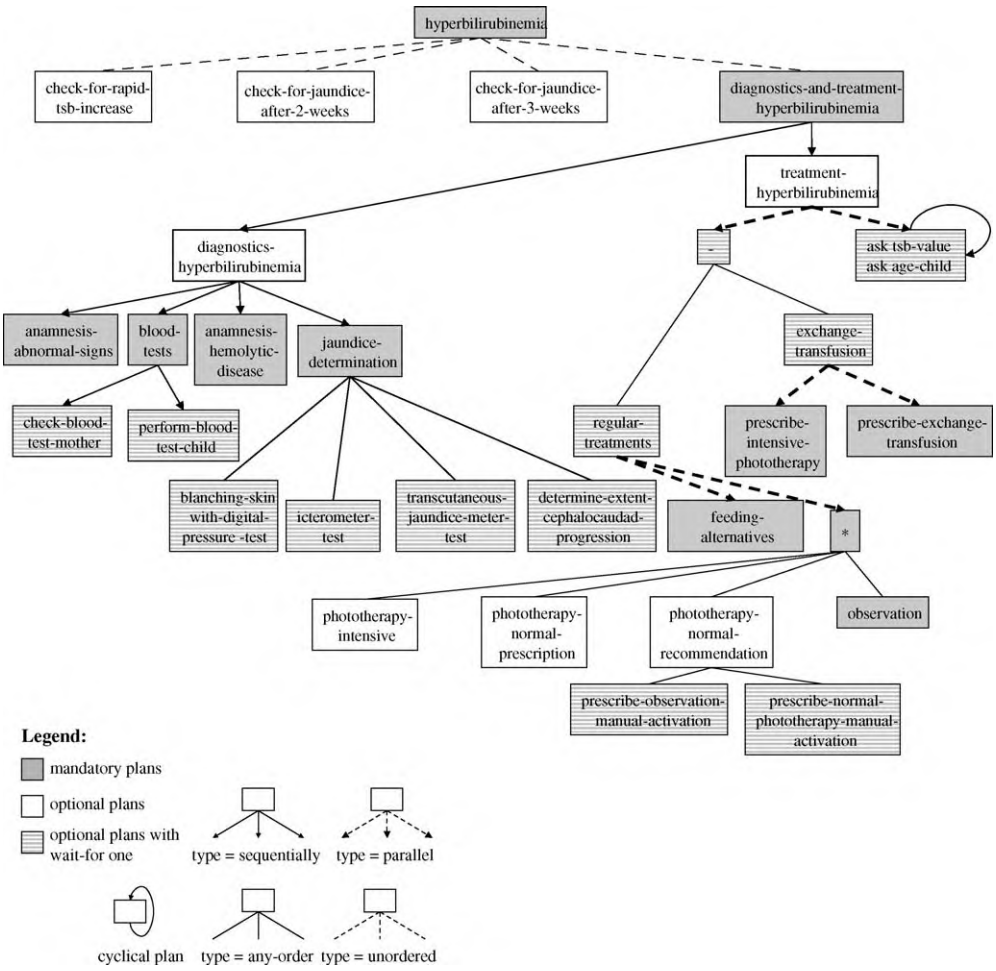

Figure 2 *Overview of the jaundice protocol in Asbru.* The main entry point of the protocol is the ''diagnostics-and-treatment-hyperbilirubinemia'' plan — the three ''check-for- . . .'' plans model a continuous monitoring of TSB level and two check-ups at specific time intervals. The plan ''diagnostics-and-treatment-hyperbilirubinemia'' is divided into two subplans, for diagnostics and a treatment, to be executed sequentially. (0) and (∗) are labels for group of plans.

4.2. Asbru model of the jaundice protocol

In this section we show the main structure of our Asbru model of the jaundice protocol. This illustrates the need for the variety of Asbru's control structures. We also give an example of a plan description that serves to illustrate the main Asbru features, such intentions, various conditions and time annotations.

As the original document, the Asbru model of the jaundice protocol has as main components a diagnostics part and a treatment part. It consists of about 40 plans and has a length of 18 pages in a simplified Asbru notation. Fig. 2 shows the overall structure of the protocol as a hierarchy of plans.

The treatment phase, on 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—24 h. Regarding the treatments (label (—) in Fig. 2), either the ''regular-treatments'' or an ''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 feeding alternatives and the different therapies (see label (∗)); the plans in this group can be tried in any-order, one at a time.

Fig. 3 shows the ''phototherapy-intensive'' plan, which describes one of the therapies. Its plan-body simply contains a subplan activation pointing to a user-performed action. One of its intentions is attaining normal 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 h.

4.3. Lessons learned

The task of modelling the jaundice protocol in Asbru has required a significant effort. It is a typical

```
plan Phototherapy-intensive
    intentions
        achieve overall-state: (bilirubin = normal)
        maintain intermediate-state: (and
            (TSB-decrease = yes) in ([4h, -] [-, 6h] [-, -] SELF)
            (TSB-change ≥ 1) in ([4h, -] [-, 6h] [-, -] SELF) )
    conditions
        filter-precondition: (or (bilirubin = phototherapy-intensive) in NOW
            normal-phototherapy-failure )
        abort-condition: (or (and (bilirubin ≠ 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
```

Figure 3 "Phototherapy-intensive" plan.

knowledge engineering task, where the knowledge in the original document is transformed into the Asbru notation, which implies devising an overall plan structure and adding all the necessary details to make it executable. As result, the Asbru version of a protocol is much more lengthy than the original text: the original jaundice protocol is 10 pages, while the Asbru code (consisting of 40 plans) fills 18 pages in the simplified notation, and 30 pages in its full XML format.

More importantly, numerous anomalies became apparent during the Asbru modelling of the jaundice protocol. We use the term anomaly to refer to any issue preventing a satisfactory interpretation of the original protocol. Below we give examples of the different anomalies we have found (see [12] for more details). For presentation purposes they have been grouped into four general categories: ambiguity, incompleteness, inconsistency and redundancy.

4.3.1. Ambiguity

Ambiguous protocol parts are inherently hard to interpret. For instance, a problem we encountered while modelling the jaundice protocol was determining whether the terms "jaundiced", "clinically jaundiced" and "clinically significant jaundice by medical judgement" had the same meaning or not. These are terms that are used in the flowchart form of the original protocol, but not defined elsewhere.

The problem with ambiguity is that it leaves space for wrong interpretations by the medical practitioner, and therefore may have a negative effect on the decisions made.

4.3.2. Incompleteness

Incompleteness can be related either to insufficient information or to entirely missing pieces of information. In either case, incompleteness hinders a correct interpretation of the protocol. For example, the original protocol contains a table with "factors to be considered when assessing a jaundiced infant". One of these factors is "rapid increase in the TSB level after 24—48 h". However, what "rapid" exactly means is missing in the protocol.

As an example of insufficient information we can cite a sentence of suggesting a procedure for the choice of TSB monitoring times but without further details on it: "determination of the rate of rise of TSB and the infants age may help determine how often to monitor bilirubin levels and whether to begin phototherapy".

4.3.3. Inconsistency

Inconsistencies are elements that may result in different (and even conflicting) decisions given the same patient data. The problems derived from inconsistent elements are very serious and as such must be avoided.

The jaundice protocol presents several inconsistencies. For instance, we found an inconsistency related to the applicability conditions of the protocol, which is meant for "healthy newborns" according to the title. The protocol also specifies that "infants who are clinically jaundiced at ≤ 24 h old are not considered healthy". However, elsewhere in the protocol an action is advised for exactly these children (i.e. babies to whom the protocol should not be applied): "a TSB level needs to be

Table 1 Anomalies found in the jaundice and diabetes protocols

	Ambiguity	Incompleteness	Inconsistency	Redundancy
Jaundice	1	10	6	0
Diabetes	4	38	0	2

determined in infants noted to be jaundiced in the first 24 h of life''.

4.3.4. Redundancy

Whenever some part of the protocol can be removed without any (noticeable) effect in the resulting recommendations, it can be considered redundant. Although redundant elements are in principle harmless, the main problem is that they may cause unnecessary inconveniences for the patient and efficiency loss in the care process. We did not find any occurrence of this type of anomaly in the jaundice protocol. However, we did find redundancies during the Asbru modelling of the diabetes protocol.

Table 1 shows the number of uncovered anomalies referring in the two protocols of our case study. By far the most common anomaly in both protocols is incompleteness. Other anomalies only occur in one of the protocols. No inconsistency was found in the diabetes protocol whereas several were found in the jaundice protocol, mainly between alternative formulations of the same recommendations between text, tables and/or flowcharts. Athough the two protocols differ in characteristics, these differences in types of anomalies do not seem related to these different characteristics.

As Table 1 shows, a significant number of anomalies can be detected through Asbru modelling [12] , [13]. These anomalies are protocol parts in which potential problems might arise, and therefore indicate points where improvements are possible. The fact that these anomalies were detected during the modelling of two high quality protocols is already an important contribution of our approach.

We now continue with the formalisation stage (see Fig. 2), namely from Asbru to formal logic. To make this step possible, a formal semantics of Asbru is needed. This is discussed in the next section.

5. Formal semantics of Asbru

With the definition of a formal semantics of Asbru, we aim to achieve two goals: first, the semantics should document the language, users should be able to understand the details of Asbru; second, the semantics should be formal enough to derive a calculus. In order to achieve the second goal, we use structural operational semantics (SOS) [14] to define Asbru. However, SOS rules are difficult to understand for people without a formal background. Therefore, we use automata, which we depict as state charts [15] , to informally illustrate the semantics. In this section, we give an overview of the semantics. A more complete definition of the semantics can be found in [16].

5.1. Overview

Asbru is a plan oriented language, and Asbru protocols are organised in a hierarchy of plans. A parent plan can refer to other sub plans in its plan-body. Conditions are used to control the applicability of a plan and to monitor its execution. Conditions can be monitored over time according to the so called time annotations. The sub plans in the plan body can be organised using different body types (sequential, parallel, unordered, or any-order). The current state of a plan — especially if a plan has been rejected, aborted, or completed — is propagated to its parent and sub plans. If a plan is mandatory, it must be completed; otherwise, it may also be rejected or aborted.

The formal semantics of Asbru consists of two parts. An operational semantics defines the plan execution engine, which is triggered with signals from an underlying data abstraction unit. All Asbru plans run in parallel and communicate via shared variables and signals. The data abstraction unit takes different types of patient data and provides signals to trigger plan conditions. Signals are also used by plans to control the execution of sub plans.

5.2. Plan execution

Plans behave as depicted in the state chart of Fig. 4. The *plan control* is divided into the selection phase *selection* and the execution phase *execution*. Initially, a plan is *inactive*. An external signal *consider* triggers the selection phase (transition SC). In state *considered*, the filter condition $filter_p$ is checked. If the condition is satisfied, control advances to state *ready* (transition F). Transitions E.1 and E.2 model manual and automatic activation and are omitted here. If, during the selection phase, the filter condition is not fulfillable any more, the plan is *rejected* (transition R). In state *activated*, the sub plans of the current plan are executed. How the execution of

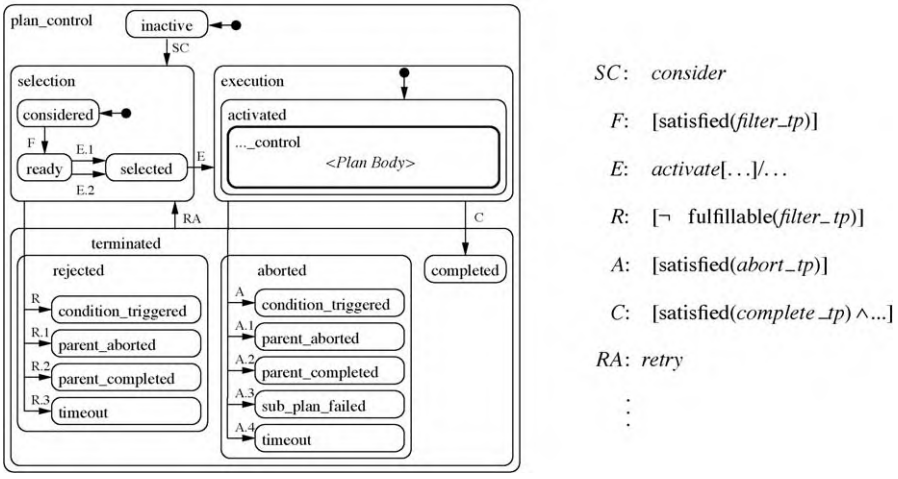

Figure 4 Plan execution.

the sub plan is controlled by the parent depends on its body type, i.e. the children can be executed sequentially, in parallel, unordered, or in any-order. For each body type there is a corresponding controlling state chart (see Section 5.3). The execution of sub plans can be either completed successfully (transition C) or aborted, e.g. in the case of emergency patient readings (transition A). Further reasons to reject or abort plans (transitions R.1—R.3, A.1—A.4) are omitted here.

Roughly, each transition corresponds to an SOS rule. For example, the rule:

$$\frac{[[\mathsf{P.state} = \mathsf{considered}]]_\sigma\ \mathsf{da}(\Phi,\sigma) \rightarrow *\mathsf{da}(\mathsf{true},\sigma')}{\mathsf{psm}_P(\sigma) \rightarrow \mathsf{psm}_P(\sigma'[\mathsf{P.state} \leftarrow \mathsf{ready}])}$$

where Φ is the filter condition of plan P, expresses that if P is in state *considered* and the filter precondition Φ is evaluated to *true*[4] by the data abstraction unit *da*, plan P may progress to state *ready*.

5.3. Controls

The different body types may empose restrictions on the execution of sub plans. This is done by deferring the generation of the events *consider*, *activate*, and *retry*. Fig. 5 contains a controlling state chart for a parent plan which executes its sub plans in any-order. The selection phases are executed in parallel. The execution phases of the sub plans are synchronized such that at most one sub plan is active at the same time. The first plan to become selectable is activated (transition *i.a*). Only if this plan terminates (transition *i.T*) another one can be activated. If several sub plans reach

[4] σ is the current valuation, and σ' is the next valuation.

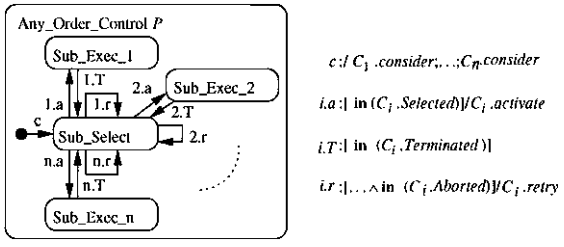

Figure 5 Any-order control.

state *selected* simultaneously, the choice is non-deterministic. If option "retry-aborted" is chosen, then transition *i.r* is used to initiate a retrial. See [16] for complete set of controlling state charts for the Asbru control structures.

5.4. Data abstraction unit

The underlying data abstraction unit is not described in detail here and only its purpose is summarised. The semantics of the abstraction unit is not operational, but functional in nature. As input, measurements of patient parameters are taken. The type of parameters can be very different, ranging from quantitative values, like the bilirubin level in blood, to boolean values, e.g. whether the patient is male or female. Data can be provided as a continuous stream of patient readings (high frequency domains such as artificial ventilation of prematured babies) or as sporadic measurements once every month (low frequency domains like diabetes mellitus).

The incoming data is memorized in a global variable, the patient record. Quantitative values can be abstracted to qualitative values. More importantly, if the data is time-annotated in an Asbru plan, the

abstraction unit evaluates data over a long-time period. For example, the time-annotated condition:

$$(\text{bilirubin decrease} < 1) \text{ in } ([4\,\text{h}, \text{-}], [\text{-}, 6\,\text{h}], [\text{-}, \text{-}], \text{now})$$

requires monitoring the decrease of the bilirubin level over a period of at least 4 h (modelled as EarliestStartingShift) and up to 6 h (modelled as LatestFinishingShift). The data abstraction unit abstracts from all these features and provides signals to trigger the plan execution unit.

This section summarised our formal semantics for Asbru, a crucial element in our process from Fig. 2. In the next section we will formalise a specific protocol (jaundice) in logic, based on the semantics given above.

6. Formalisation of the jaundice protocol in KIV

In the second stage of the process (see Fig. 1) we have used the logic of the KIV verification tool [4]. 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 the KIV interactive tool allows for the verification of large and complex systems, as has been shown by its application to a number of real-world systems (distributed systems, control systems, etc).

Below we first describe the relevant aspects of KIV for Asbru specification and verification, we continue with the formalisation of our case study, and conclude with the lessons learned from this formalisation step.

6.1. KIV

KIV supports the entire software development process, i.e. the specification, the implementation and the verification of software systems. 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 as union , that can be used to combine more simple specifications. For functional aspects, algebraic specifications are used to specify abstract data types.

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 ($\phi \| \psi$) 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 ($\phi \|_{\text{s}} \psi$), as well as any-order execution ($\phi \|_{\text{a}} \psi$), 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) [17] to formulate properties. This logic is first-order and allows finite and infinite intervals. Here we restrict ourselves to the temporal operators always ($\Box\,\phi$), eventually ($\Diamond\,\phi$), next ($\circ\,\phi$), 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 $v = 0 \wedge (\Box\, v' = v + 1) \rightarrow \Diamond\, v = n$ 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.

In KIV, the proof technique for verifying parallel programs is symbolic execution with induction. Details can be found in [18]. Since programs are treated as formulas — for both, the semantics is a set of traces — they can be arbitrarily mixed. This gives rise to a modular proof technique, which is very

Table 2 Translation patterns of some Asbru constructs into KIV

Asbru	KIV
complete condition ϕ body	If ϕ **then** body
filter precondition ϕ body	**Break** body **if** ϕ, update plan-state = ''complete''
abort condition ϕ body	**Break** body **if** ϕ, update plan-state = ''abort''
$\ll$ name $\gg$ (*plan activation*)	$\ll$ name $\gg$ #(...) (*procedure call*)
do type = sequentially P1 ..., Pn	P1; ..., Pn
do type = any-order P1, ..., Pn	P1 $\|_{\text{a}}$ ..., Pn
Wait-for Pi body	**Break** body if *some expression on* Pi-state

important for the verification of Asbru plans as they tend to be large.

6.2. KIV formalisation of the jaundice protocol

In a first attempt to formally analyse Asbru plans, 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 of Fig. 2. This gives the possibility to obtain feedback from the formalisation and verification phases in terms of the Asbru model, and to exploit this structure during proof attempts. Table 2 shows some of the patterns that we used in the translation of Asbru plans.

In many cases the KIV translation closely follows the structure of the original Asbru plan, except for small details. This means that a large part of the translation could be done automatically (up to 80%). Other translations needed additional encodings, and therefore had to be done manually. The example in Fig. 6, corresponding to the plan "phototherapy-intensive", serves to illustrate a mixture of automatic and manual translations that we have obtained. See Fig. 3 for the Asbru model of this plan.

This translation includes an await construct to 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 phototherapy-intensive-activated).

6.3. Lessons learned

The KIV formalisation step has taken considerably less effort than the Asbru modelling step [13]. This is mainly due to the structure-preserving strategy we followed. Thanks to this, the formalisation is largely a mechanical translation of Asbru plans into KIV programs.

We cannot strictly say that the formalisation in KIV has contributed to the improvement of the original protocol, as was the case in the Asbru modelling. Clearly, the most important contribution of this formalisation stage is to produce a version of the protocol that can be used as the basis for verification of protocol properties as discussed in Sections 7 and 8.

Furthermore, the KIV formalisation has served to disambiguate unclarities in the Asbru model that resulted from the modelling stage: a number of semantic problems with Asbru were uncovered and resolved by refining the formal semantics of Asbru in KIV.

7. Identification of protocol properties

In parallel to the formalisation process described so far (the right-pointing arrows in Fig. 1), we have

```
phototherapy-intensive#(var phototherapy-normal-prescription-activated,patient-data,
    time, phototherapy-intensive-activated)
begin
  await  get-bilirubin(patient-data.tsb)  = phototherapy-intensive
       ∨ get-bilirubin(patient-data.tsb)  = phototherapy-normal
           ∧ phototherapy-normal-prescription-activated  ≠ ⊥
           ∧ 4 ≤ time - phototherapy-normal-prescription-activated.value
           ∧ ¬ get-decrease(patient-data.tsb);
  phototherapy-intensive-activated:= mk-value(time);
  break
    prescribe-intensive-phototherapy#(;  time)
  if   get-bilirubin(patient-data.tsb) ≠ phototherapy-intensive
    ∨ get-bilirubin(patient-data.tsb)  = phototherapy-intensive
      ∧ (  4 ≤ time - phototherapy-intensive-activated.value
           ∧ ¬ get-decrease(patient-data.tsb)
        ∨ ...)
end
```

Figure 6 KIV translation of "phototherapy-intensive" plan of Fig. 3.

identified a number of protocol properties which were deemed potentially interesting and medically relevant (the downwards-pointing arrows in Fig. 1).

Such properties require a deep understanding of both the disease and the protocol dealing with it. These properties focus on aspects like the relevant clinical parameters or the actions that the care providers are required to perform. As an example, when treating diabetes with insulin, a desirable property of the protocol (as recommended by our domain expert) is to distribute the morning and evening insulin doses according to the ratio 2/3—1/3.[5]

The correctness of plan intentions and properties based on medical indicators[6] were deemed of particular interest from a medical point of view. 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 to verify 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[7] Committee [19]. These quality indicators for jaundice are formulated by a different medical institution then the one who developed the jaundice protocol. Therefore these indicators are not specifically defined for our protocol, which make them more attractive to verify.

Notice that we are not claiming that we can verify any intention a plan may have, in general, but rather that formal verification can be applied to determine under which conditions a protocol complies with the intentions the developers had in mind.

8. Formalisation and verification of protocol properties

The formalisation of protocol properties and their verification constitute the final step of the process. In this section we give two examples and summarise the lessons that we have drawn from these verification experiences.

8.1. Verification of an intention

One of the intentions of the plan ''phototherapy-intensive'' is the reduction of the TSB levels by at least 1 mg/dl within 4—6 h. 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:

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

Informally, this formula says that, if ''phototherapy-intensive'' is activated, when 4—6 h have elapsed, there should be a decrease in TSB levels greater or equal than 1. This is exactly the second conjunct of the second intention in Fig. 3.

The property was successfully proved in KIV, in a fully automatic proof of more then a thousand steps. 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 (see Fig. 6). 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.

8.2. Verification of an indicator

The MAJIC indicators that appear in [19] have been refined by the same organisation into different medical review criteria for the evaluation and treatment of jaundice in infant care. This includes a set of 11 criteria which jaundice protocols must comply with. Among these criteria, we selected indicator #7 for verification. The indicator is stated as follows[8]:

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

[5] Of course whether this recommendation should be part of the protocol would have to be discussed with the team of experts that developed the protocol.

[6] Indicators are measures to judge the quality of care, systematically engineered by medical experts.

[7] Making advances against jaundice in infant care

[8] The rationale of this indicator is beyond the scope of this work.

This states that if the decision is made to apply phototherapy (which will involve drawing bilirubin samples to measure the success of the therapy), then such a measurement should no longer be repeated after the therapy is discontinued.

This indicator was translated into the following temporal formula:

$$
\begin{aligned}
&/^{*}\ \mathit{indicator}\ ^{*}/\\
&\Box\ \ \textbf{laststep}\\
&\quad \vee \quad pd\ .\text{under-phototherapy} \wedge \neg\, pd'\ .\text{under-phototherapy}\\
&\qquad \wedge\, pd\ .\text{tsb} = TSB_0\\
&\quad \rightarrow \Box\, pd\ .\text{tsb} = TSB_1 \wedge TSB_1 \neq TSB_0 \rightarrow \Box\, pd\ .\text{tsb} = TSB_1
\end{aligned}
$$

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 the ''regular-treatments'' subplan (Fig. 2) complies with this indicator. These attempts 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 h, allowing ''treatment-hyperbilirubinemia'' to measure TSB twice. The analysis of this proof served to identify the assumptions under which the indicator should be satisfied: (1) if phototherapy is discontinued, bilirubin levels are ''normal''; (2) the observation plan will run for less than 24 h; and (3) after phototherapy and observation, bilirubin levels will still be ''normal''. Given these additional assumptions, the proof could be completed successfully. (Proving the indicator for the ''regular-treatments'' plan in the jaundice protocol involves thousands of proof steps.) 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.

8.3. Lessons learned

The verification attempts have served to clarify any remaining problems in the Asbru model. In particular, we discovered errors in the formulation of intentions which shed light on the semantics of Asbru intentions.

More importantly, we have shown that it is possible in practice to systematically analyse whether a protocol formalised in KIV complies with certain medically relevant properties [20]. Various properties of the jaundice protocol have been the object of formal verification using the KIV system, with different types of results. In some cases the properties have been confirmed, which means that the formalised version of the protocol complies with them. However, in other cases verification was problematic, i.e. it was not possible to complete the proof unless we added additional assumptions describing the conditions needed to make the property true.

The incorporation of the additional assumptions necessary to complete a proof might seem an extremely difficult task, because of the high amount of medical background knowledge that is usually involved — e.g. about medical actions, treatment effects, etc. It is important to note that we have followed a pragmatic approach, namely, assumptions have been identified as they were needed, by exploiting the information of failed proof attempts, rather than elicited *a priori* from domain experts and other valid sources. For example, the assumptions that make indicator #7 true are statements of the kind ''if phototherapy is discontinued, bilirubin levels are normal'', rather different from an exhaustive description of the effects of the different phototherapies. This results in a characterisation of the circumstances under which the property holds. The obtained characterisation, after analysis by medical experts, could be used either to redefine/restrict the property or to improve the original protocol text by documenting the cases under which the property does (or does not) hold.

9. Evaluation of results

In order to assess the overall utility of our approach, a selection of the results of the formalisation and verification steps have been presented for evaluation to a group of medical experts. The evaluation has been done by means of structured interviews in which three different experts judged: (1) the quality of the selected protocols, (2) the criticality of the problems detected in the protocols, and (3) the utility of the approach of quality improvement of protocols by formal methods.

Most of the problems detected in the jaundice protocol were not considered serious from a clinical point of view. However, one of the experts, who is also involved in guideline development, stated that some of the flaws were serious as they indicate lack of care in designing the protocol, and therefore should be avoided. Concerning the diabetes proto-

col, half of the flaws were not considered very serious, but the remainder were judged as very serious, and one even extremely serious [21].

Finally, in general the three experts regarded the computer-supported systematic analysis of protocols we carried out as an improvement over current methodologies. The experts judged our approach especially promising for guideline development. In addition, they mentioned that it constitutes an important first step to make computerised support for guidelines possible.

10. Related work

In this section we relate our work to currently ongoing work in the literature for computerised medical guidelines and protocols. First of all we would like to emphasize that although there is a lot work on decision support systems in clinical settings with respect to medical protocols, there is only a small amount of work concerning the improvement of medical protocols themselves.

In the last decade a number of languages for medical protocols have been developed with different levels of formality. Although we have used Asbru as protocol representation language in our case study, we believe that the same can be done with other languages like PROforma [22] , EON [23] and others [24,25] , provided that they are extended with a formal semantics (which most of these languages are currently lacking). In this sense, our proposal constitutes a general approach to the improvement of medical guidelines and protocols, and is independent of our specific choice for Asbru.

A rather different approach for improving a guideline is by critiquing the physician. Critiquing involves comparing the actions that physicians perform to solve particular patient cases with the recommendations prescribed by the guideline for those cases [26]. When deviations occur, critiquing can concentrate on the explanation thereof or, alternatively, can provide a valuable input for guideline improvement [27]. If critiquing is performed based on intentions [28] , the physician's intentions are inferred from the patient data and the performed actions. These are then checked against the intentions stated in the guideline. In [28] , satisfaction of intentions depends on the data of a specific patient, whereas in our approach we try to find general assumptions under which the intentions hold for a given guideline, independently of the data of a specific patient.

Another direction of related work is automated quality assessment of clinical actions and patient outcomes. This concerns deriving structured quality indicators from formalised specifications of guidelines used in decision support [29]. In our work we use such indicators as formal properties that the guideline must comply with.

We consider work such as [30—33] as other approaches to the improvement of medical guidelines.

Together, these works cover a whole spectrum of properties of guidelines, ranging from internal coherence properties related to the representation language to properties which depend on the specific guideline, to general properties of guidelines such as safety properties. The work of [30] and [32] fit in the beginning of this spectrum, while the work of [31,33] fits in the end.

The motivation of [30,32,34,35] is slightly different from ours. They identify anomalies in guidelines, but their focus is, as mentioned above, more on language coherence properties, whereas our focus is mainly on properties specific to the protocol.

Miller et al. [30] tackles the problem that clinical logic specified in the guidelines may be incomplete or inconsistent. If ''if-then'' rules are used as representation language for guidelines, incompleteness means that there are combinations of clinically meaningful conditions to which the system (guideline) is not able to respond. The Commander tool supports the verification of rule-based clinical guidelines using semantic constraints. Commander is able to point out clinical situations where the rules are incomplete. In their study of a guideline for childhood immunization, Miller et al. were able to identify a number of missing immunization rules, which required a response by the guideline.

Shiffman and Greens [34] and Shiffman [35] present a method for the verification and simplification of guidelines by applying decision-table techniques. Their assumption is that a guideline can be described as situation/action statements: If the antecedent circumstances exist, then one should perform the recommended actions. They focus on two properties for verification, namely completeness and consistency of a guideline expressed in terms of decision-table. Again, these properties are internal coherence properties, whereas we are focussed on domain specific properties.

Hammond et al. [31] uses another approach to improve medical guidelines, which can be considered as exploiting general safety properties for guidelines, and therefore belonging to the end of the spectrum (i.e. general properties about guidelines). They work on safety reasoning in protocols, and identify a number of generic safety principles. The idea is that the same generic knowledge can be reused to generate specific safety clauses when

designing new treatment plans. These inclusion of specific safety clauses can be considered as an improvement of the protocol.

Finally, it is also worth to mention that our work is consistent with some approaches to protocol improvement from the medical communities [36]. For instance, the appraisal instrument developed by the AGREE Collaboration [37]http://www.agreecollaboration.org/(accessed: 1 June 2005) stresses protocol aspects that can be detected through formalisation, such as the specificity and unambiguity of recommendations.

11. Conclusions and future challenges

11.1. 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 not only is feasible but also useful for quality improvement purposes. We have fully formalised two real-world medical protocols in a two-stage formalisation process. Subsequently, we have used a theorem prover to systematically analyse whether the formalisation complies with certain medically relevant protocol properties. This process has revealed a number of anomalies which were judged to be of medical relevance by a panel of independent medical experts.

Throughout this process we have obtained the following concrete results:

- A consolidated formal language to model medical practice protocols. The Asbru protocol representation language has been extensively tested with two real-life medical protocols. In a later stage, the semantics of the Asbru constructs has been clarified and formally specified — **formal semantics of Asbru**. As result, Asbru has become a **formal language for protocol representation**.
- **Two protocols, each both formalised in Asbru and in KIV.** The two medical protocols used in our study have been gradually transformed into a formal representation. Each of the (intermediate) versions of the protocols constitutes an achievement, since it demonstrates that protocol formalisation is feasible.
- **A list of anomalies** obtained during the formalisation process of our two case studies.
- With the aim of carrying out a series of formal verification exercises, a number of **medically relevant protocol properties** has been identified.
- **Verification proofs** for these protocols and properties in KIV. Some of the previous properties have been selected for verification in KIV, and proofs have been produced for them with different types of results. The fact that these proofs have been completed shows that formal verification of protocols is feasible.

The main contribution of our work is to show that it is indeed possible to formally analyse medical protocols to improve their quality. If a protocol is developed with certain goals in mind (expressed as intentions or indicators), formal 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 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 the first attempt ever in verifying medical protocols with mathematical rigour. We expect that the necessary effort should decrease in the future, e.g. with more direct KIV support for Asbru protocols. Notice that our approach requires to have a team with a number of different skills: a medical expert, a knowledge engineer (ideally trained in Asbru, and with some medical background) and a formal methods expert trained in KIV.

11.2. Future challenges

In this section we discuss some future challenges. We first discuss some challenges concerning applying our proposed approach of protocol improvement. We continue with some current problems in medical protocol development, which leads to the challenge of living guidelines.

In this paper we have shown that our approach of applying formal methods for protocol improvement is in principal possible: we completed the entire process of modelling and formalisation of both protocol and properties, and completed some proofs.

As we have seen, several techniques (modelling, formalising and verifying) enable us to find anomalies in the protocols. Open questions that remain are: when do we need which type of techniques; what kind of anomalies can be found by using these different techniques; would it be possible to use verification only in the critical parts of the protocol, and how can we identify these critical parts.

We have used Asbru as the language for modelling protocols. Other less formal guideline representation language like EON, Glif, GUIDE or others [24] could be used as well. We expect that modelling in these languages can also detect anomalies. Although, we expect that this would yield fewer anomalies because of the informal nature of these languages.

Other techniques which could contribute to our formal methods approach is to use model checking for proving properties, or to use an Asbru interpreter such that patient data can be checked against the protocol.

An entirely different set of challenges arises from considering the two major problems with clinical guidelines from the medical perspective. Currently, the two problems are:

1. At this moment, a guideline is a static document which cannot be modified easily. Consequently, recommendations can be outdated or no longer applicable in practice, because most guidelines are only revised every 5 years, while scientific and pragmatic knowledge is growing faster every year.
2. Practitioners' adherence to guidelines is often insufficient.

These problems have led to future challenges, sometimes referred to as ''living guidelines'':

1. Update of the guidelines on a more continuous basis: clinical guidelines have to become flexible, adaptable documents. The aim is to develop guidelines which present up-to-date and state-of-the-art knowledge to practitioners. To make this possible, guidelines have to be modular in structure, so that only part of a guideline can be adjusted and not the whole document needs revision.
2. Enhancing the adherence to guidelines by supporting them with computerised tools aiming at integrating the guidelines more in the daily work-processes of practitioners.

To make the approach of living guidelines possible, there must be some major changes in the guideline development process. Most guidelines are authored in an unstructured narrative form. Computer-based support for the authoring process depends on a more formal, structured representation such as those investigated in this paper. To incorporate guidelines in computerised systems, they must be developed in a more structured way. Formal methods can be of help here. This will be an important first step to enhance further computer-based support of guidelines and protocols.

Acknowledgements

This work has been partially supported by the EC IST Programme, under contract number IST-2001-33049—Protocure. We want to thank Hugo Roomans, Geert Berger, and Frea Kruizinga for their contribution to the Asbru modelling tasks; Tibor Bosse, for his work on the Asbru-Light interpreter; Marije Geldof, for her analysis of the links between the original protocols and their Asbru counterparts; Peter Votruba, for his support with the GMT tool; and the medical experts Fred Storm, Christian Popow, and Margot van de Bor, who evaluated our work; our project evaluators Ahmed Bouajjani, J. Woodcock, and Mark Musen for their constructive and insighttful comments. Finally, we would also like to thank Yuval Shahar for all the fruitful discussions on verification of protocol properties.

References

[1] Field MJ, Lohr KN, editors. Clinical practice guidelines: directions for a new program. Washington, DC, USA: National Academy Press; 1990.
[2] Clayton PD, Hripsak G. Decision support in healthcare. Int J Bio-Med Comput 1995;39:59—66.
[3] Musen MA, Rohn JA, Fragan LM, Shortliffe EH. Knowledge engineering for a clinical advice system: uncovering errors in protocol specification. Bull Cancer 1987;74:291—6.
[4] Balser M, Reif W, Schellhorn G, Stenzel K, Thums A. Formal system development with KIV. In: Maibaum, T, editor, Fundamental approaches to software engineering, number 1783 in LNCS. New York: Springer; 2000. p. 363—6.
[5] Shahar Y, Miksch S, Johnson P. The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. Artif Intell Med 1998;14: 29—51.
[6] Elkin PL, Peleg M, Lacson R, Bernstam E, Tu S, Boxwala A, et al. Toward standardization of electronic guidelines. MD Comput 2000;17(6):39—44.
[7] American Academy of Pediatrics, Provisional Committee for Quality Improvement and Subcommittee on Hyperbilirubinemia. Practice parameter: management of hyperbilirubinemia in the healthy term newborn. Pediatrics 1994:94: 558—65.
[8] Rutten GEHM, Verhoeven S, Heine RJ, de Grauw WJC, Cromme PVM, Reenders K, et al. NHG-standaard diabetes mellitus type 2 (eerste herziening). Huisarts en Wetenschap 1999;42(2):67—84 (first revision).
[9] Miksch S. Plan management in the medical domain. AI Commun 1999;12(4):209—35.
[10] Fox J, Johns N, Lyons C, Rahmanzadeh A, Thomson R, Wilson P. PROforma: a general technology for clinical decision support systems. Comput Meth Program Biomed 1997;54: 59—67.
[11] Seyfang A, Kosara R, Miksch S. Asbru's reference manual, Asbru version 7.3. Technical report Asgaard-TR-2002—1,

Vienna University of Technology, Institute of Software Technology; 2002.
[12] Marcos M, Roomans H, ten Teije A, van Harmelen F. Improving medical protocols through formalisation: a case study. In: Ehrig H, Kraemer B, Ertas A, editors. Proceedings of the 6th international conference on integrated design and process technology (IDPT-02), Session on formal methods in healthcare. USA: Society for Design and Process Science; 2002. p. 59—64.
[13] Marcos M, Balser M, ten Teije A, van Harmelen F. From informal knowledge to formal logic: a realistic case study in medical protocols. In: Gómez-Pérez A, Richard Benjamins V, editors. Proceedings of 13th international conference on knowledge engineering and knowledge management (EKAW-02), number 2473 in LNAI. Spain: Sigüenza; 2002. p. 49—64.
[14] Acteto L, Fokkink W, Verhoef C. Handbook of process algebra. In: Chapter structural operational semantics. New York, USA: Elsevier; 2001.
[15] Harel D. On visual formalisms. Commun ACM 1988;31 (5):514—30.
[16] Balser M, Duelli C, Reif W. Formal semantics of Asbru — an overview. In: Ehrig H, Kraemer B, Ertas A, editors. Proceedings of the 6th biennial world conference on integrated design and process technology (IDPT-02). USA: Society for Design and Process Science; 2002. p. 1—8.
[17] Moszkowski B. A temporal logic for multilevel reasoning about hardware. IEEE Comput 1985;18(2):10—9.
[18] Balser M, Duelli C, Reif W, Schellhorn G. Verifying concurrent systems with symbolic execution. J Logic Comput (special issue) 2002;12(4):549—60.
[19] MAJIC. MAJIC Steering Committee meets. Making Advances Against Jaundice in Infant Care (MAJIC) Newsletter 1998;1 (2).
[20] Marcos M, Balser M, ten Teije A, van Harmelen F, Duelli C. Experiences in the formalisation and verification of medical protocols. In: Keravnou E, Dojat M, Barahona P, editors. Proceedings of the 9th European conference on artificial intelligence in medicine (AIME-2003), number 2780 in LNAI. New York: Springer-Verlag; 2003. p. 132—41.
[21] Deliverable D5: evaluation of results, Protocure Project, ISTD2001—33049, 2002. http://www.protocure.org/ (accessed: 1 June 2005).
[22] Fox J, Johns N, Rahmanzadeh AH. Disseminating medical knowledge: the proforma approach. Artif Intell Med 1998; 14(12):157—81.
[23] Musen MA, Tu SW, Das AK, Shahar Y. Eon: a component-based approach to automation of protocoldirected therapy. J Am Med Inform Assoc 1996;39(6):367—88.
[24] Peleg M, Tu S, Bury J, Ciccarese P, Fox J, Greenes RA, et al. Comparing computer-interpretable guideline models: a case-study approach. J Am Med Inform Assoc 2003;10(1): 52—68.
[25] Wang D, Peleg M, Tu S, Boxwala A, Greenes R, Patel V, et al. Representation primitives, process models and patient data in computer-interpretable clinical practice guidelines: a literature review of guideline representation models. Int J Med Inform 2002;68(1):59—70.
[26] Miller, PM. A critiquing approach to expert computer advice: ATTENDING. Research notes in artificial intelligence. London: Pitman Publishing; 1984.
[27] Marcos M, Berger G, van Harmelen F, ten Teije A, Roomans H, Miksch s. Using critiquing for improving medical protocols: harder than it seems. In: Quaglini, S, Barahona, P, Andreassen S, editors. Proceedings of the 8th European conference on artificial intelligence in medicine (AIME-2001), number 2101 in lecture notes in computer science. Portugal: Cascais; 2001. p. 431—41.
[28] Advani A, Lo K, Shahar Y. Intention-based critiquing of guideline-oriented medical care. In: Chute CG, editor. Proceedings of the annual fall symposium of the american medical informatics (AMIA), Orlando, 1998. Philadelphia: Hanley and Belfus; 1998. p. 483—7.
[29] Advani A, Goldstein MK, Shahar Y, Musen MA. Developing quality indicators and auditing protocols from formal guideline models: knowledge representation and transformations. In: Musen M, editor. Proceedings of the annual fall symposium of the american medical informatics, Washington, DC, USA, 2003. Philadelphia: Hanley and Belfus; 2003. p. 11—5.
[30] Miller DW, Frawley SJ, Miller PL. Using semantic constraints to help verify the completeness of a computer-based clinical guideline for childhood immunization. Comput Meth Program Biomed 1999;58:245—54.
[31] Hammond P, Sergot M, Wyatt J. Formalisation of safety reasoning in protocols and hazard regulations. In: Gardner RM, editor. Proceedings of the annual symposium on computer application in medical care (SCAMC-95), New Orleans, Louisiana, 1995. Philadelphia: Hanley and Belfus; 1995. p. 253—7.
[32] Duftschmid G, Miksch S. Knowledge-based verification of clinical guidelines by detection of anomalies. Artif Intell Med 2001;22:23—41.
[33] Fox J, Das S. Safe and sound: artificial intelligence in hazardous situations. Cambridge: MIT Press; 2000.
[34] Shiffman R, Greens R. Improving clinical guidelines with logic and decision-table techniques. Med Decis Making 1994;14:245—54.
[35] Shiffman R. Representation of clinical practice guidelines in conventional and augmented decision tables. J Am Med Inform Assoc 1997;4:382—93.
[36] Graham I, Calder L, Hébert P, Carter A, Tetroe J. A comparison of clinical practice guideline appraisal instruments. Int J Technol Assess Health Care 2000;16(4):1024—38.
[37] Appraisal of guidelines for research and evaluation (AGREE) instrument, 2001. Obtained in http://www.agreecollaboration.org/ (accessed: 1 June 2005).