

# Verification of Medical Guidelines Using Background Knowledge in Task Networks

Arjen Hommersom, Perry Groot, Peter J.F. Lucas, Michael Balseer, and Jonathan Schmitt

**Abstract**—The application of a medical guideline to the treatment of a patient's disease can be seen as the execution of tasks, sequentially or in parallel, in the face of patient data. It has been shown that many of such guidelines can be represented as a “network of tasks,” that is, as a sequence of steps that have a specific function or goal. In this paper, a novel methodology for verifying the quality of such guidelines is introduced. To investigate the quality of such guidelines, we propose to include medical background knowledge to task networks and to formalize criteria for good medical practice that a guideline should comply with. This framework was successfully applied to a guideline dealing with the management of diabetes mellitus type 2 by using KIV.

**Index Terms**—Medical guidelines, background knowledge, formal verification, temporal logic.

## 1 INTRODUCTION

COMPUTER-BASED decision support in healthcare is a field with a long-standing tradition, dealing with complex problems in medicine, such as diagnosing a disease and assisting in the prescription of the appropriate treatment. The trend of the last decades has been to base clinical decision making more and more on sound scientific evidence, which has been called *evidence-based medicine* [1], [2]. In practice, this has led organizations of medical specialists in particular areas to develop medical guidelines, that is, structured documents suggesting the detailed steps that should be taken by healthcare professionals in managing the disease of a patient to promote standards of medical care. Ethical concerns about evidence-based medicine have been raised [3], and there is a potential risk that medical guidelines can cause harm when improperly developed [4]. However, guidelines have also shown to improve healthcare outcomes [4] and may even reduce the costs of care up to 25 percent [5].

Researchers in artificial intelligence have picked up on the increasing use of medical guidelines and are working toward offering computer-based support in the development and deployment of guidelines by using computer-oriented languages and tools [6], [7]. This has given rise to the emergence of a new paradigm for the modeling of complex clinical processes as a “network of tasks,” where a task consists of a number of steps, each step having a specific function or goal [8], [9]. Examples of languages that support task models, and which have been evolving since the 1990s, include PROforma [10], [11], Asbru [12], [13], EON

[14], [15], and GLIF3 [9]. In this work, medical guidelines are considered as real-world examples of structured documents, which can benefit from formalization although experience has shown that looking upon medical guidelines as formal objects is a nontrivial task [16].

Medical guidelines should not be considered static objects, as they are changed on a regular basis as new scientific evidence becomes available. Rapidly changing and evolving evidence makes it difficult to adjust guidelines in such a way as to keep them up to date. As a consequence, a computer-based support of guideline development should also be concerned with the updating of guidelines, that is, indicating where guidelines should be updated in the light of new evidence.

In this paper, we approach this problem by applying formal methods of checking the quality of medical guidelines. Here, we are mainly concerned with checking the *general* quality criteria of good medical practice that a guideline should comply to. This has been called the metalevel approach to quality checking of medical guidelines [17]. For example, a guideline should preclude the prescription of redundant drugs or advise against a prescription of a treatment that is less effective than some alternative. Newly obtained evidence may invalidate the properties of a guideline, because, for example, new patient management options have arisen, or financial costs have decreased through new developments in drug or equipment manufacturing.

A solid foundation for the application of formal methods for the quality checking of medical guidelines can already be found in literature. In [8] and [18], logical methods have been used to analyze properties of guidelines. We have shown in [17] and [19] that the theory of abductive diagnosis can be taken as a foundation for the formalization of quality requirements of a medical guideline in temporal logic, which has been used in verifying the quality requirements of good medical practice of alternative treatments.

The scientific contribution of the present paper is twofold. First, we formalize quality requirements of medical guidelines that include, besides separate treatments, also the temporal relations between separate treatments, by which we mean the order in which they are prescribed. Second,

• A. Hommersom, P. Groot, and P.J.F. Lucas are with the Institute for Computing and Information Sciences, Radboud University Nijmegen, PO Box 9010, 6500GL Nijmegen, The Netherlands.

E-mail: {arjenh, perry, peterl}@cs.ru.nl.

• M. Balseer and J. Schmitt are with the Institute für Informatik, Universität Augsburg, D-86135 Augsburg, Germany.

E-mail: {michael.balseer, jonathan.schmitt}@informatik.uni-augsburg.de.

Manuscript received 18 May 2006; revised 22 Nov. 2006; accepted 5 Feb. 2007; published online 16 Feb. 2007.

For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0262-0506.

Digital Object Identifier no. 10.1109/TKDE.2007.1030.

- Step 1: diet.
  - Step 2: if Quetelet index (QI)  $\leq 27$ , prescribe a sulfonylurea drug; otherwise, prescribe a biguanide drug.
  - Step 3: combine a sulfonylurea drug and biguanide (replace one of these by a  $\alpha$ -glucosidase inhibitor if side-effects occur).
  - Step 4: one of the following:
    - oral antidiabetic and insulin
    - only insulin

Fig. 1. Tiny fragment of a clinical guideline on the management of diabetes mellitus type 2. If one of the steps  $k = 1, 2, 3$  is ineffective, then the management moves to step  $k + 1$ .

using our quality requirements and medical background knowledge, we interactively verify a guideline dealing with the management of diabetes mellitus type 2. More specifically, we model the guideline as a “network of tasks” by using the language Asbru and, additionally, verify metalevel properties for this model by using KIV, an interactive theorem prover [20]. To the best of our knowledge, verification of a fully formalized guideline as a network of tasks by using medical background knowledge has not been done before. The presented framework provides a sound formal foundation for further research in quality checking of medical guidelines and the temporal relations among different treatments involved.

The remainder of this paper is structured as follows: Section 2 gives a short introduction to medical guidelines and deals with the formalization of some of its elements. Section 3 describes the three types of object and metaknowledge involved in the verification of medical guidelines: background knowledge, the medical guideline, and quality requirements. Sections 4, 5, and 6 describe these knowledge types in more detail and discuss their formalization. Section 7 describes the specification in KIV. Section 8 describes the verification process in detail, and Section 9 concludes our work. Formal notations used throughout this paper are summarized in the Appendix.

## 2 MEDICAL GUIDELINES

### 2.1 Introduction to Medical Guidelines

Guidelines, medical guidelines, or practice guidelines are all commonly used abbreviations for the full term “clinical practice guideline.” An often-cited definition of guidelines is the one by Field and Lohr [21]:

“Clinical practice guidelines are systematically developed statements to assist practitioner and patient decisions about appropriate healthcare for specific clinical circumstances.”

Though “protocol” is often synonymously used for “guideline,” a protocol gives detailed statements about *how* one should act in daily practice, whereas a guideline gives more general scientifically founded statements about *what* should be done. Protocols are often seen as more detailed practice-oriented versions of a guideline [22]. In this work, the focus is on medical guidelines.

An example of a fragment of a guideline is shown in Fig. 1. It is part of the guideline for general practitioners about the treatment of diabetes mellitus type 2 [23], which is used as a running example in this paper. General practitioners’ guidelines are normally quite compact. Guidelines

for medical specialists are often large—they can be as large as 100 pages—but even then, they consist of sections similar to our example. Translating a guideline into a clear and structured fragment such as in Fig. 1 can take a lot of effort; however, the formalization of a guideline is not the main focus of this paper, which is about the verification of a once formalized guideline.

The diabetes mellitus type 2 guideline provides practitioners with a clear structure of recommended actions to be taken for the control of the glucose level. This kind of information is typically found in medical guidelines in the sense that medical knowledge is combined with information about the order and time of treatment (for example, sulfonylurea (SU) in step 2), about the patients and their environment (for example, a Quetelet index (QI) lower than or equal to 27), and, finally, about which drugs are to be administered to the patient (for example, an SU drug). In the next section, we discuss these elements in more detail.

### 2.2 Requirements for the Formal Verification of Guidelines

To be able to verify quality criteria of medical guidelines by using formal methods, we need to have a language that can be used to express quality criteria that can be related to the key elements in a guideline. Above, we stated that the key elements in medical guidelines are (at least) order in time, patients, and interventions. Here, we discuss our choices for a language for the formal representation of those key elements, used in the remainder of the paper.

#### 2.2.1 Time

As medical management is a time-oriented process, diagnostic and treatment actions described in guidelines are performed in a temporal setting. It has been shown previously that the stepwise, possibly iterative, execution of a guideline can be described by means of temporal logic [18]. This is a modal logic [24], where relationships between worlds in the usual possible-world semantics are modal logic are understood as time order. In this paper, we will use a variant of this logic based on future-time linear temporal logic. The language of this logic is a first-order logic augmented with the temporal operators listed in Table 1. The semantics of this language is given by a set  $D$  representing the universe of discourse, a set of interpretations  $I_t$  for interpreting statements from the first-order logic, and a function  $succ$ , where  $succ(t)$  is the set of zero or one successors of the time points of  $t$ . First-order expressions  $\varphi$  at time  $t$  are interpreted using  $I_t$  in the domain  $D$ . For example,  $t \models \varphi$  means that  $\varphi$  is satisfied at time  $t$  with respect to  $I_t$  and  $D$  [24].

Note that the **last** modality can only hold in models where at some point following the successor function, no successor exists. In all other models, **last** will never hold. Also, note that some operators can be defined in terms of other operators, for example,  $\Box \varphi \equiv \neg \diamond \neg \varphi$  and **last**  $\equiv \bullet \perp$ . A more expressive logic can be gained by including, for example, the **until** operator, where  $\varphi$  **until**  $\psi$  denotes that eventually,  $\psi$  holds, and before that,  $\varphi$  holds. However, as such operators are not used in this paper, they have been omitted.

TABLE 1  
Used Temporal Operators ( $t$  Stands for a Time Instance)

Notation	Interpretation	Formal semantics
$\square \varphi$	$\varphi$ will always be true	$t \models \square \varphi \Leftrightarrow \forall t' \geq t : t' \models \varphi$
$\diamond \varphi$	$\varphi$ will eventually be true	$t \models \diamond \varphi \Leftrightarrow \exists t' \geq t : t' \models \varphi$
$\circ \varphi$	execution does not terminate and the next state satisfies $\varphi$	$t \models \circ \varphi \Leftrightarrow \exists t' \in \text{succ}(t) : t' \models \varphi$
$\bullet \varphi$	either execution terminates or the next state satisfies $\varphi$	$t \models \bullet \varphi \Leftrightarrow \forall t' \in \text{succ}(t) : t' \models \varphi$
<b>last</b>	the current state is the last	$t \models \text{last} \Leftrightarrow \text{succ}(t) = \emptyset$

This logic allows one to look at guidelines formally at a particular abstraction level. In Section 8, we show this logic to be suitable for quality checking of medical guidelines; however, it is possible to add more fine-grained temporal operators if they are needed.

### 2.2.2 Patient Groups

Although, in practice, a guideline is used for the management of a particular patient, recommendations in guidelines are always written with a certain *patient group* in mind, not just a single patient. Patient groups are groups of patients that share common characteristics about their current state or previous states. One can abstract from the actual situation of a patient by providing a logical language that refers to one or more situations, including the necessary common characteristics, without fixing all the details. Typical elements for describing the state of patients are symptoms, signs, and test outcomes. Here, we have chosen to use predicate logic with equality and unique names assumption [25]. For example, the literal “Condition(*hyperglycemia*)” is used to represent the patient group of all patients that currently have the condition of hyperglycemia. Subgroups of patient groups can be specified by using a conjunction with additional literals; for example, “Condition(*hyperglycemia*)  $\wedge$  QI  $\leq$  27” specifies the patient group of patients who have hyperglycemia and also have a QI less than or equal to 27. We sometimes represent the conjunction also in set form; for example, the latter conjunction becomes “{Condition(*hyperglycemia*), QI  $\leq$  27}.”

### 2.2.3 Interventions and Treatments

An intervention is the act of intervening, interfering, or interceding with the intent of modifying the outcome. In medicine, interventions include all medical actions that influence the state of a patient or his environment. A treatment is usually restricted to methods that provide a cure for an illness or disability; however, the terms intervention and treatment are often used synonymously. We have chosen to represent the domain of interventions by a countable set. Subsets of this set are interpreted as *treatments* in which each intervention of the set is applied. Interventions that are not elements of the treatment are assumed not to be applied. We abstract from medical management details such as changing drug dosages.

## 3 VERIFICATION OF MEDICAL GUIDELINES

Medical guidelines give recommendations based on the best available evidence. Although diabetes mellitus type 2 is a complicated disease, the guideline fragment shown in

Fig. 1 is not. This indicates that much knowledge concerning diabetes mellitus type 2 is missing from the guideline. Verifying whether a guideline fulfills some property therefore additionally needs the specification of *background knowledge*.

The ideas that we use here to verify the quality requirements for medical guidelines are inspired by previous work, where a distinction was made between the different types of knowledge that are involved in defining quality requirements [19]. We assume that there are at least three types of knowledge involved in detecting the violation of good medical practice:

1. knowledge concerning the (patho)physiological mechanisms underlying the disease and the way the treatment influences these mechanisms (the knowledge involved could be, for example, causal or empirical in nature and is an example of *object knowledge*),
2. knowledge concerning the recommended treatment in every step of the guideline and how the choice for each treatment is affected by the state of the patient, that is, the order information from the medical guideline (this is also an example of *object knowledge*), and
3. knowledge concerning good practice in treatment selection (this is *metaknowledge*).

The first type of object knowledge will be called *background knowledge*. The second type of object knowledge is the order information from the medical guideline, which can be considered a network of tasks or a hierarchical plan. The plan prescribes the treatment that influences the (patho)physiological mechanisms, which results in information about patient groups that can be used by the plan to make the best possible decision in the subsequent step of the protocol. Incompleteness of background knowledge may lead to insufficient knowledge about a patient, which may result in a plan making a nondeterministic choice. Of course, the guideline should recommend the collection of data when possible if this data is crucial for decision making.

The third type of knowledge, the metaknowledge, includes general knowledge about good medical practice, for example, preferring a treatment over another if it uses a smaller number of drugs and has an equal effect on the patient. This knowledge will be formalized by *quality requirements*, that is, (reasoning) patterns that specify the behavior of treatment selection, given certain patient data. These quality requirements can be used as proof obligations in the verification of medical guidelines.

In Sections 4, 5, and 6, the three types of knowledge involved (background knowledge, medical guideline, and quality requirements) are described in more detail in the context of diabetes mellitus type 2, and a formalization in terms of temporal logic, as discussed in Section 2.2, is given. In Sections 7 and 8, the quality requirements are verified with an interactive theorem prover.

## 4 BACKGROUND KNOWLEDGE

In this section, we discuss how temporal logic can be used for the representation of medical background knowledge. We use this to give a formalization of drug treatments used in the management of diabetes mellitus type 2.

### 4.1 Informal Description of the Background Knowledge

In diabetes mellitus type 2, various metabolic control mechanisms are deranged, and many different organ systems may be affected by the disorder. Glucose level control, however, is the most important mechanism. The medical knowledge described below has been acquired with the help of clinical diabetes experts, but most of it is also publicly available in general medical handbooks or Web pages, for example, <http://www.diabetes.org>.

The protein hormone insulin, which is produced by the *B cells* in the Langerhans islets of the *pancreas*, has the following major effects:

- It increases the uptake of glucose by the liver, where it is stored as glycogen, and inhibits the release of glucose from the liver.
- It increases the uptake of glucose by insulin-dependent tissues such as muscle and adipose tissues.

At some stage in the natural history of diabetes mellitus type 2, the level of glucose in the blood is too high (*hyperglycemia*) due to the decreased production of insulin by the *B cells*. A popular hypothesis explaining this phenomenon is that target cells have become insulin resistant, which, with a delay, causes the production of insulin by the *B cells* to rise. After some time, the *B cells* become exhausted, and they are no longer capable of meeting the demands for insulin. As a consequence, hyperglycemia develops.

The treatment of diabetes mellitus type 2 consists of the following:

- Use of SU drugs such as tolbutamide. These drugs stimulate the *B cells* in producing more insulin, and if the cells are not completely exhausted, then the hyperglycemia can thus be reverted to normoglycemia (normal blood glucose levels).
- Use of *biguanides* (BGs) such as metformin. These drugs inhibit the release of glucose from the liver.
- Use of *α-glucosidase inhibitors*. These drugs inhibit (or delay) the absorption of glucose from the intestines.
- Injection of *insulin*, for example, by an infusion pump or pen. This is the ultimate, causal treatment.

As insulin can only be administered by injection, in contrast to the other drugs that are normally taken orally,

doctors prefer to delay prescribing insulin as long as possible. Thus, the treatment part of the diabetes mellitus type 2 guideline mentions that one should start with prescribing an oral antidiabetic (SU or BG). Two of these can also be combined if taking only one has an insufficient glucose-level lowering effect. If the treatment is still unsatisfactory, then the guideline suggests to 1) either add insulin or 2) stop the oral antidiabetic entirely and start with insulin.

From a medical point of view, advice 1) above is somewhat curious. If the oral antidiabetics are no longer effective enough, then the *B cells* could be completely exhausted. Under these circumstances, it does not make a lot of sense to prescribe an SU drug. The guideline here assumes that the *B cells* are always somewhat active, which may limit the amount of insulin that has to be prescribed. Similarly, the prescription of a BG (or an *α-glucosidase inhibitor*) is justified, as by adding such an oral antidiabetic to insulin, the number of necessary injections can be reduced from twice a day to once a day. It should be noted that, when on insulin treatment, patients run the risk of getting hypoglycemia, which can be seen as a side effect of insulin not mentioned explicitly in the guideline.

### 4.2 Formalization of the Background Knowledge

The formalization of the background knowledge of diabetes mellitus type 2 described here is based on previous work [17], [19], which formalized the background knowledge for the purpose of verifying the quality requirements of single treatments (see Section 5.1). An example formula in [17] and [19] is the following:

$$\begin{aligned} & (\Box \text{uptake}(\text{liver}, \text{glucose}) = \text{up} \\ & \wedge \Box(\text{uptake}(\text{peripheral-tissues}, \text{glucose}) = \text{up} \\ & \quad \wedge \text{capacity}(\text{b-cells}, \text{insulin}) = \text{exhausted}) \\ & \wedge \Box^{-1} \text{Condition}(\text{hyperglycemia}) \\ & ) \rightarrow \Box(\text{Condition}(\text{normoglycemia}) \\ & \quad \vee \text{Condition}(\text{hypoglycemia})), \end{aligned}$$

where  $\Box^{-1}$  is the converse modality of  $\Box$ . That is,  $\Box^{-1}$  (informally) means *always in the past*. This formalization, in combination with a fully formalized guideline, is unsuitable for verifying the quality requirements as the time frame in which we verify that the quality requirements have changed. In previous work [17], [19], the time frame was restricted to the start and end of the application of a *single treatment*, whereas now, we are interested in verifying the quality requirements in the time frame of a *guideline* in which *many treatments* may be started and ended at any time. Hence, the assumption made in [17] and [19] that a certain drug should *always* be applied to deduce its effects no longer holds when verifying the quality requirements in the time frame of a guideline.

For this reason, we modified the background knowledge included in [17] and [19] such that it could be used for the purpose of verifying the quality requirements of a fully formalized guideline. Note that these changes do not contradict the previous formalization, but rather, they strengthen the formulas, so it can be used in this context. We employed the same logic used for formalizing guidelines, that is, future-time linear logic (see Section 2.2.1), with

- |   |
|---|
| (1) $\text{Drug}(\text{insulin}) \rightarrow \circ (\text{uptake}(\text{liver}, \text{glucose}) = \text{up} \wedge \text{uptake}(\text{peripheral-tissues}, \text{glucose}) = \text{up})$<br>(2) $\text{uptake}(\text{liver}, \text{glucose}) = \text{up} \rightarrow \text{release}(\text{liver}, \text{glucose}) = \text{down}$<br>(3) $(\text{Drug}(\text{SU}) \wedge \neg \text{capacity}(\text{b-cells}, \text{insulin}) = \text{exhausted}) \rightarrow \circ \text{secretion}(\text{b-cells}, \text{insulin}) = \text{up}$<br>(4) $\text{Drug}(\text{BG}) \rightarrow \circ \text{release}(\text{liver}, \text{glucose}) = \text{down}$<br>(5) $(\circ \text{secretion}(\text{b-cells}, \text{insulin}) = \text{up} \wedge \text{capacity}(\text{b-cells}, \text{insulin}) = \text{subnormal} \wedge \text{QI} \leq 27$<br>$\wedge \text{Condition}(\text{hyperglycemia})) \rightarrow \circ \text{Condition}(\text{normoglycemia})$<br>(6) $(\circ \text{release}(\text{liver}, \text{glucose}) = \text{down} \wedge \text{capacity}(\text{b-cells}, \text{insulin}) = \text{subnormal}$<br>$\wedge \text{QI} > 27 \wedge \text{Condition}(\text{hyperglycemia})) \rightarrow \circ \text{Condition}(\text{normoglycemia})$<br>(7) $(\circ \text{release}(\text{liver}, \text{glucose}) = \text{down} \vee \circ \text{uptake}(\text{peripheral-tissues}, \text{glucose}) = \text{up}) \wedge \text{capacity}(\text{b-cells}, \text{insulin}) = \text{nearly-exhausted}$<br>$\wedge \circ \text{secretion}(\text{b-cells}, \text{insulin}) = \text{up} \wedge \text{Condition}(\text{hyperglycemia})) \rightarrow \circ \text{Condition}(\text{normoglycemia})$<br>(8) $(\circ \text{uptake}(\text{liver}, \text{glucose}) = \text{up} \wedge \circ \text{uptake}(\text{peripheral-tissues}, \text{glucose}) = \text{up} \wedge \text{capacity}(\text{b-cells}, \text{insulin}) = \text{exhausted}$<br>$\wedge \text{Condition}(\text{hyperglycemia})) \rightarrow \circ (\text{Condition}(\text{normoglycemia}) \vee \text{Condition}(\text{hypoglycemia}))$<br>(9) $(\text{Condition}(\text{normoglycemia}) \oplus \text{Condition}(\text{hypoglycemia}) \oplus \text{Condition}(\text{hyperglycemia})) \wedge$<br>$\neg (\text{Condition}(\text{normoglycemia}) \wedge \text{Condition}(\text{hypoglycemia}) \wedge \text{Condition}(\text{hyperglycemia}))$ |
|---|

Fig. 2. Background knowledge  $\mathcal{B}_{\text{DM2}}$  of diabetes mellitus type 2.  $\text{Drug}(x)$  holds iff drug  $x$  is being administered at that moment in time. The  $\oplus$  operator denotes the exclusive OR operator.

the time frame starting at the beginning of the guideline and continuing to either infinity or the end of the guideline. It is assumed that any additional knowledge needed about the patient history is available at the start of the guideline, allowing us to omit the converse modality  $\square^{-1}$  from [17] and [19]. Furthermore, we interpreted the time period between causes (for example, drug administration) and effects (for example, drug effects) as the time period between the current state and the next state. In summary, the  $\square$  operator used in [17] and [19] to model global qualitative behavior is replaced by the  $\circ$  operator to model cause-effect relationships. Hence, the above formula is replaced by the following formula:

$$\begin{aligned}
& (\circ \text{uptake}(\text{liver}, \text{glucose}) = \text{up} \\
& \wedge \circ (\text{uptake}(\text{peripheral-tissues}, \text{glucose}) = \text{up} \\
& \quad \wedge \text{capacity}(\text{b-cells}, \text{insulin}) = \text{exhausted}) \\
& \wedge \text{Condition}(\text{hyperglycemia}) \\
& ) \rightarrow \circ (\text{Condition}(\text{normoglycemia}) \\
& \quad \vee \text{Condition}(\text{hypoglycemia})).
\end{aligned}$$

The final specification is denoted by  $\mathcal{B}_{\text{DM2}}$  and is shown in Fig. 2. As an illustration of how we can read these formulas, consider (5). This formula denotes that in case the B cells are being stimulated to secrete more insulin (that is, the secretion is *up* after some unspecified time period), the capacity of these B cells is subnormal, and if the QI is less than or equal to 27, then we expect that the condition will change from *hyperglycemia* to *normoglycemia*.

Other formalizations of this knowledge is possible, for example, using compartmental models that have been successfully used to measure and predict physiological variables and parameters in given individuals [26]. Such mathematical models are usually described using differential equations. In the context of clinical reasoning, compartmental models are not very useful, as they essentially concern determining dosages in time and, thus, do not offer the right level of abstraction for what we have investigated. In our work, we have used qualitative temporal reasoning compatible with reasoning as done in the guideline.

## 5 QUALITY REQUIREMENTS

We first summarize the quality requirements of treatment choice, which were developed previously [17], [19], and

extend these to the quality requirements of medical guidelines.

### 5.1 Quality Requirements of Treatments

The treatment of patients in accordance with good medical practice can be formalized as follows: Let  $\mathcal{B}$  be the medical background knowledge (for example, as in Fig. 2),  $T$  be a treatment,  $P$  be a patient group, and  $N$  be a collection of intentions, which the physician has to achieve.

The background knowledge consists of causal relations between treatments and the patient. Finding an acceptable treatment, given such knowledge, amounts to finding an explanation, in terms of a treatment, that the intention will be observed. Finding the best possible explanation for a number of findings is called *abductive* reasoning. Therefore, we say that a treatment  $T$  is called a *proper* treatment according to the theory of abductive reasoning; that is,  $T \in \text{Pr}_P$ , if [17]

- (T1):  $\mathcal{B} \cup T \cup P \not\models \perp$  (the treatment does not have contradictory effects) and
- (T2):  $\mathcal{B} \cup T \cup P \models N$  (the treatment handles all the patient problems intended to be managed).

Furthermore, let  $\preceq_\varphi$  denote a preference relation on treatments, assumed to be reflexive and transitive, with  $T \preceq_\varphi T'$ , meaning that  $T'$  is *at least as preferred* to  $T$ , given criterion  $\varphi$ . With  $\prec_\varphi$  we denote the order such that  $T \prec_\varphi T'$  if and only if  $T \preceq_\varphi T'$  and  $T' \not\preceq_\varphi T$ . When both  $T \preceq_\varphi T'$  and  $T' \preceq_\varphi T$  hold or when  $T$  and  $T'$  are incomparable with respect to  $\preceq_\varphi$ , we say that  $T$  and  $T'$  are *indifferent*, which is denoted as  $T \sim T'$ . For example, in diabetes mellitus type 2, a preference order would be to minimize 1) the number of insulin injections and 2) the number of drugs involved, which results, among others, in the preferences  $\{\text{insulin}\} \preceq_\varphi \{\text{BG drug}, \text{SU drug}\}$  and  $\{\text{SU drug}\} \preceq_\varphi \{\text{diet}\}$ .

If, in addition to conditions (T1) and (T2), the following condition holds:

- (T3):  $O_\varphi(T)$ , where  $O_\varphi$  is a metapredicate standing for an optimality criterion or a combination of optimality criteria  $\varphi$ , which is defined as  $O_\varphi(T) \equiv \forall T' \in \text{Pr}_P : \neg(T \prec_\varphi T')$ ,

then the treatment is said to be *in accordance with good medical practice*, abbreviated to *good* and denoted as  $\text{Good}_\varphi(T, P)$ .

An example of a preference relation is subset minimality  $\preceq_{\subseteq}$ : If we assume  $T, T' \subseteq \{d_1, \dots, d_n\}$ , to be a set of drugs, then

$$\forall T, T' : T \preceq_{\subseteq} T' \equiv T' \subseteq T,$$

that is, the minimum number of effective drugs are being prescribed. For example, if  $\{d_1, d_2, d_3\}$  is a treatment that satisfies (T3) in addition to (T1) and (T2), then the subsets  $\{d_1, d_2\}$ ,  $\{d_2, d_3\}$ ,  $\{d_1\}$ , and so on, do not satisfy (T1) and (T2). In the context of abductive reasoning, subset minimality is often used in order to distinguish between various solutions. It is also referred to in literature as *Occam's razor*. Another definition of the metapredicate  $\preceq_{\varphi}$  is in terms of minimal cost  $\preceq_c$ :

$$\forall T, T' : T \preceq_c T' \equiv c(T') \leq c(T),$$

where  $c(T) = \sum_{d \in T} \text{cost}(d)$ . Combining the two definitions also makes sense. For example, one could come up with a definition of  $O_{\subseteq, c}$ : Between two subset-minimal treatments, select the one that is the cheapest in financial or ethical sense.

## 5.2 Quality Requirements of Medical Guidelines

Medical guidelines consist of, besides a description of treatments, a control structure that uses patient information to decide on a particular treatment plan, that is, the order (sequentially or in parallel) of treatments. Quality requirements for guidelines should extend the quality requirements of treatments, as shown in the previous section, to include the requirements on the control structure.

Analogous to the previous section, good medical practice of medical guidelines can be formalized as follows: Let  $\mathcal{B}$  be the background knowledge,  $T$  be a treatment,  $P$  be a patient group,  $N$  be a collection of intentions, and  $M$  be a medical guideline in the form of a plan hierarchy. The structure  $M$  is composite and entails at certain points in time a treatment  $T$  that corresponds to a particular stage in the treatment process, as described by the guideline. At all of the other time points,  $M$  entails the empty treatment, that is,  $T = \emptyset$ . A medical guideline  $M$  is called a *proper* guideline according to the theory of abductive reasoning; that is,  $M \in \text{Pr}_P$ , if

- (M1):  $\mathcal{B} \cup M \cup P \not\models \perp$  (the guideline does not have contradictory effects) and
- (M2):  $\mathcal{B} \cup M \cup P \models \diamond N$  (at some future state, the guideline satisfies all intentions).

In contrast to meta-axioms (T1) and (T2), we focus here on treatment plans  $M$  rather than separate treatments  $T$ . Note that in contrast to (T2), in (M2), we state  $\diamond N$ , that is, eventually  $N$ , as the guideline may consist of multiple interventions and diagnoses that may not directly result in reaching each intention. Note that we here assume a broad notion of intention in the sense that intentions may hold for a longer period of time. For example, after two weeks, the intention may be satisfied that a certain drug is administered within the first week. This persistence of satisfied intentions explains why we require that at some point, all the intentions should all be satisfied at once.

If, in addition to (M1) and (M2), the following condition holds:

- (M3):  $O_{\varphi}(M)$ , where  $O_{\varphi}$  is a metapredicate standing for an optimality criterion or a combination of optimality criteria  $\varphi$ , which is defined as  $O_{\varphi}(M) \equiv \forall M' \in \text{Pr}_P : \neg(M \prec_{\varphi} M')$ ,

then the guideline is said to be *in accordance with good medical practice*, again denoted as  $\text{Good}_{\varphi}(M, P)$ .

A particular instance of condition (M3) and the preference relation can be constructed using a predicate. This predicate divides the possible guideline models into two groups, namely, one group in which all models satisfy the predicate and one in which all models do not satisfy the predicate. The preference relation is then the preference over one of these two groups. For example, some predicates that should hold for a proper guideline are the completion of treatments as soon as the patient is cured or the consistency of the order of the prescribed treatments, with the preference order between treatments.

Formally, one can represent this by specifying, in addition to (M1) and (M2), additional proof obligations. For example,

- The plan never prescribes treatments that are less preferred than the treatment that is in accordance with good medical practice:

$$\begin{aligned} \mathcal{B} \cup M \cup P \models \square (\forall T' : \text{Good}_{\varphi}(T', P) \\ \wedge T \rightarrow \neg(T \prec_{\varphi} T')). \end{aligned}$$

- If the patient is successfully treated, that is, assuming no new conditions or comorbidities, then the plan should not start new treatments, but unboundedly long treatments (like insulin) are allowed:

$$\mathcal{B} \cup M \cup P \models \square (N \wedge T \rightarrow \circ(T \vee \text{last})).$$

- The order of prescribed treatments is consistent with the preference relation:

$$\mathcal{B} \cup M \cup P \models \square (T \wedge \diamond T' \rightarrow \neg(T \preceq_{\varphi} T')).$$

These quality requirements are examples that we consider realistic in the case study. We believe that these examples have sufficient generality and can be used to illustrate the techniques that we employ. Nonetheless, further research will have to decide which quality requirements are most suitable for which guidelines, as quality requirements should always be considered in the context of a specific guideline. In Section 8, we describe the actual verification process of these proof obligations for the guideline for the management of diabetes mellitus type 2.

## 6 MEDICAL GUIDELINES IN ASBRU

Much research has already been devoted to the development of representation languages for medical guidelines. Most of them look at guidelines consisting of a composition of actions whose execution is controlled by conditions [22]. However, most of them are not formal enough for the

purpose of our research, as they often incorporate free-text elements, which do not have a clear semantics. Exceptions to these are PROforma [10], [11] and Asbru [12], [13]. The latter has been chosen in our research as a basis to implement the medical guideline  $M$  mentioned in the previous section.

### 6.1 Introduction to Asbru

A medical guideline is considered in Asbru as a hierarchical *plan*. The main components of an Asbru plan are intentions, conditions, plan body, and time annotations. Furthermore, a plan can have arguments and can return a value.

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 to can be intermediate or final (overall). In total, there are 12 possible forms of intentions built up by combining elements from the sets {achieve, maintain, avoid}, {intermediate, overall}, and {state, action}.

*Conditions* can be associated to a plan to define different aspects of its execution. The most important types of conditions are 1) filter and setup conditions,<sup>1</sup> which must be true before a plan can start, 2) abort conditions, which define when a plan must abort, and 3) complete conditions, which define when a started plan finishes successfully. Conditions can be “overridable” (that is, health personnel can manually satisfy the condition) or “require confirmation” (that is, conditions must be explicitly confirmed before they are satisfied).

The *plan body* contains the actions, subplans, or both to be executed as part of the plan. The main types of the plan body are listed as follows:

1. User performed: An action has to be performed by a user, which requires interaction, which is not further modeled.
2. Single step: An action that can be an activation of a subplan, an assignment of a variable, a request for an input value, or an if-then-else statement.
3. Subplans: A set of plans to be performed in a given order: sequentially, in parallel, in any order, or unordered.
4. Cyclical plans: A repetition of actions over a time period.

In the case of subplans, it is also required to specify a waiting strategy to describe which of the subplans must be completed for the superplan to complete. For example, all subplans should be executed (**wait for all**).

*Time annotations* can be associated to various Asbru elements, for example, intentions, conditions, and plan activations. A time annotation specifies the following:

1. the interval in which things must start,
2. the interval in which things must end,
3. their minimal and maximal duration, and
4. a reference time point.

1. Filter conditions are conditions about values that cannot change value, for example,  $sex = male$ , whereas setup conditions are conditions about values that may change, for example, glucose level.

### 6.2 The Semantics of Asbru

To help in the understanding of Asbru, we review here the semantics of Asbru in a semiformal statechart notation [27]. In Asbru, plans are organized in a hierarchy, where a plan may include a number of subplans. The semantics of Asbru is defined in [28] by flattening the hierarchy of plans and using one top-level control to execute all plans synchronously. Within each top-level step, a step of every plan is executed. Whether a plan is able to progress depends on its conditions. The plan state model, as shown in Fig. 3, defines the semantics of the main plan hierarchy. The “Plan\_Control” is divided into a selection phase, an execution phase, and a termination phase. Each plan goes into the “Considered” state when it receives a *consider* signal. In this state, its *filter condition* is checked. If it evaluates to true, then the control advances to the “Possible” state. Then, the setup condition is checked. If it is passed, then the control advances to the execution phase. If the filter condition is not satisfied, or the setup condition is not satisfiable anymore (that is, it is not possible to satisfy the condition in the future because a deadline has passed), then the plan is rejected. The same happens if the superplan terminates. In the execution phase, the plan waits for an external signal *activate* to be sent by its superplan.

In the “Activated” state, the subplans are executed, which can be sequentially, in parallel, unordered, or in any order, and each order determines a different controlling statechart [28]. A plan can synchronize its subplans by using the signals *consider* and *activate*. Additional control to propagate the execution states of a subplan to its parent, and vice versa, is also present; for example, the abortion of a mandatory subplan enforces the parent-plan also to abort. Subplans can be either completed successfully or aborted, for example, in the case of emergency patient readings.

The complete technical definitions, in addition to the semantics of the other constructs not shown here, can be found in [27].

### 6.3 Asbru Model of the Diabetes Mellitus Type 2 Guideline

The overall structure of the Asbru model of the guideline fragment (Fig. 1) of the treatment of diabetes mellitus type 2 is shown in Fig. 4. The Asbru model consists of a hierarchy of seven plans. The top-level plan “Treatments\_and\_Control” sequentially executes the four subplans “Diet,” “SU\_or\_BG,” “SU\_and\_BG,” and “Insulin\_Treatments,” which correspond to the four steps of the guideline fragment in Fig. 1. The fourth subplan “Insulin\_Treatments” is further refined by the two subplans “Insulin\_and\_Antidiabetics” and “Insulin,” which can be executed in any order.

The Asbru specifications of two plans in the hierarchy, namely, “SU\_or\_BG” and “Insulin\_Treatments,” are shown in Fig. 5. Independent of the intentions, which are high-level goals, one can describe the expected behavior of plans by using the **effects** attribute. In the case of “SU\_or\_BG,” there is a relationship between the QI and the drug administered. If the QI is less than or equal to 27, then SU is administered; else, BG is administered. The “SU\_or\_BG” plan corresponds to Step 2 in the guideline fragment of Fig. 1, which completes if the patient condition improves (that is, the patient no longer has hyperglycemia). In Fig. 5,

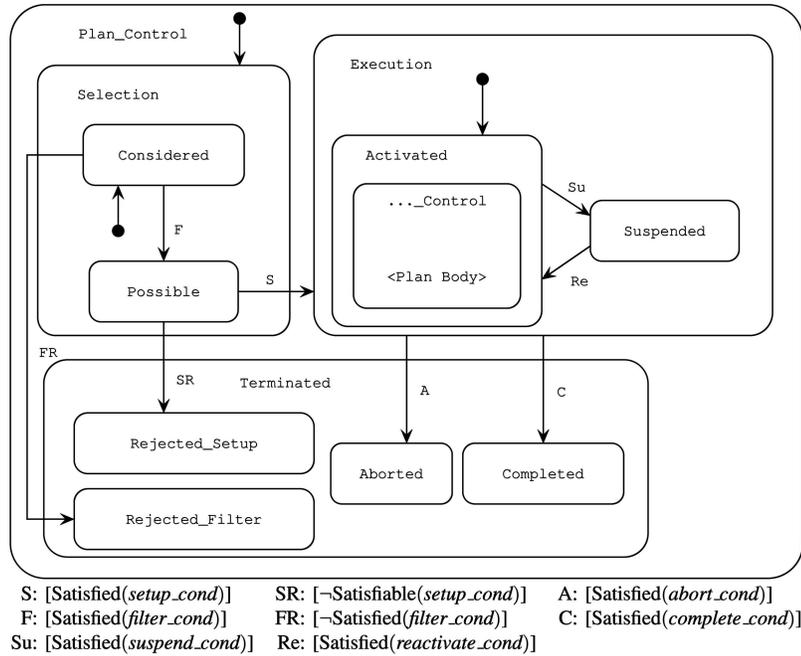


Fig. 3. The plan state model, where *Satisfied(cond)* denotes that the environment satisfied the condition *cond*, whereas *Satisfiable(cond)* denotes that, theoretically, the environment could still satisfy the condition *cond*; that is, that no deadline has passed in the case of time constraints.

this is represented by the **complete condition**. The “SU\_or\_BG” plan aborts when the condition of the patient does not improve. In Fig. 5, this is represented by the **abort condition**, which requires a manual confirmation to ensure that some time passes for the drugs to have an impact on the patient condition.

The “Insulin\_Treatments” plan consists of two subplans, which correspond to the two options of Step 4 in the guideline fragment of Fig. 1 (that is, either insulin is administered or insulin and antidiabetics are administered). Subplans are represented using the **body** attribute (Fig. 5). In this case, the subplans are executed in any order, and execution completes if one of the two subplans successfully completes (**wait for one**).

## 7 SPECIFICATION IN KIV

Sections 4, 5, and 6 have given the temporal logic formalization of the background knowledge of diabetes mellitus type 2, the quality requirements, and the Asbru model of the medical guideline for diabetes mellitus type 2, respectively. In this section, we discuss how these elements

can be translated into KIV representations so that they become amendable to verification.

### 7.1 Introduction to KIV

KIV is an integrated development environment to develop systems using formal methods [20]. It has been found suitable for verification of large software systems such as a Prolog compiler [29] or an electronic purse [30]. The specification language of KIV is based on higher order algebraic specifications. Reactive systems can be described in KIV by means of statecharts or parallel programs: here, we use parallel programs. Parallel programs are modeled as follows: Let *e* denote an arbitrary (first-order) expression, *v<sub>d</sub>* be a dynamic variable (see below), and the constructs for parallel programs include *v<sub>d</sub> := e* (*assignments*), **if** *ψ* **then** *φ*<sub>1</sub> **else** *φ*<sub>2</sub> (*conditionals*), **while** *ψ* **do** *φ* (*loops*), **var** *v<sub>d</sub> = e* **in** *φ* (*local variables*), **patom** *φ* **end** (*atomic execution*), *φ*<sub>1</sub>||*φ*<sub>2</sub> (*interleaved execution*), and [*p*##(*e*; *v<sub>d</sub>*)] (*call to procedure p, with*

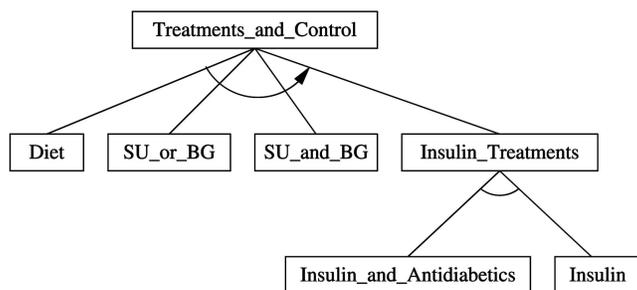


Fig. 4. Asbru plan hierarchy of the diabetes mellitus type 2 guideline.

```

plan SU_or_BG
  effects
    (QI ≤ 27 → SU ∈ Drugs) ∧
    (QI > 27 → BG ∈ Drugs)
  abort condition
    condition = hyperglycemia confirmation required
  complete condition
    condition = hypoglycemia ∨
    condition = normoglycemia

plan Insulin_Treatments
  body anyorder wait for one
    Insulin_and_Antidiabetics
    Insulin
    
```

Fig. 5. Asbru specifications of two treatments recommended in the diabetes mellitus type 2 guideline.

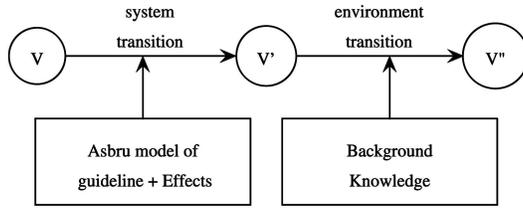


Fig. 6. The relation between unprimed and primed variables as two distinct transitions: the system transition (including the Asbru model and its effects) and the environment transition (including the background knowledge).

value parameters  $e$  and reference parameters  $v_d$ ). The semantics of this extended language is defined in [31].

The correctness of systems is ensured by reasoning about the parallel program using a future-time linear temporal logic [32]. In KIV, the logic in Table 1 is extended with static variables  $v_s$ , which are variables that are mapped to the same element in the universe of discourse at each time point. Dynamic variables  $v_d$  such as program variables may have different interpretations at different time points. In the upcoming sections, the use of static variables will be explicitly mentioned. A specialty of KIV is the use of primed and double-primed variables: A primed variable  $v'_d$  represents the value of this variable after a system transition, whereas the double-primed variable  $v''_d$  is interpreted as the value after an environment transition. System and environment transitions alternate, with  $v''_d$  being equal to  $v_d$  in the successive state (see Fig. 6 and Section 8.1).

## 7.2 Specification Methodology in KIV

The guideline and patient can be looked upon as a system (guideline) that interacts with the environment (patient). KIV allows a clear distinction between the system and environment transitions by using primed and double-primed variables. Therefore, the Asbru model is only allowed to map variables into primed variables, whereas the environment is only allowed to map primed variables into double-primed variables. The system and environment transitions alternate (Fig. 6).

However, the system transitions in Asbru may involve a large number of steps (for example, signals and plan state changes) before the model reaches a stable state, from which no further step can be made, unless time progresses, or the environment changes. Asbru is mainly a control-oriented language, and many control steps are not considered to take any real time at all. In an interactive theorem prover like KIV, this behavior can be modeled by the introduction of two transition types: *microsteps* and *macrosteps* [33]. Microsteps are technical Asbru steps where time and environment are not allowed to change. Macrosteps are temporal steps in which an interaction can occur with the environment (for example, plan activations) and are only executed when there are no microsteps possible. The variable "Tick," controlled by the symbolic execution of the Asbru semantics, holds when a macrostep occurs.

In KIV, system descriptions are represented by means of a set of algebraic specifications. These algebraic specifications can be enriched with additional algebraic structures, which form a dependency structure between the different

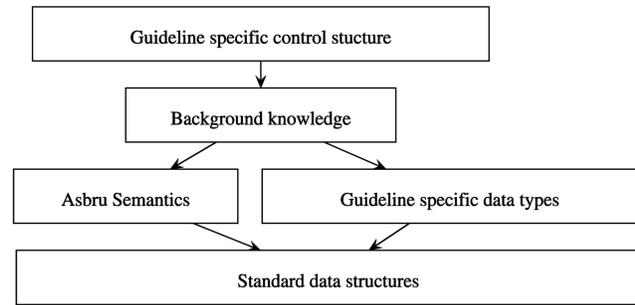


Fig. 7. Dependency structure of Asbru specifications, with  $A \rightarrow B$  denoting that  $A$  depends on  $B$ .

specifications. To maximize reusability, several layers are used for representing our framework in KIV. The lowest layer in this dependency structure consists of standard data structures like Booleans and sets, which are typically obtained from libraries in KIV. On top of that, all data structures required to obtain a full definition of the semantics of Asbru were provided. The remaining layers consist of the structures dependent on the specific guideline under study. On top of the standard data structures, additional data structures are represented. For the diabetes case study, the data types are modeled as enumeration types. On top of the Asbru semantics and data structures, the background knowledge is represented. The top layer consists of the control structure of the guideline, which is the structure of Fig. 4 in the diabetes case study (see Fig. 7).

## 7.3 Specification of Background Knowledge in KIV

The background knowledge, that is, (1), (2), (3), (4), (5), (6), (7), (8), and (9) in Fig. 2, has been reformulated in terms of preconditions and postconditions (for reasons that will become clear in Section 8.1). Every element that refers to the current point in time is interpreted as a precondition, and each element that refers to the next point in time is interpreted as a postcondition. The values of these elements are stored in a data structure, denoted by "Patient." The patient is modeled by a sequence of pairs  $[v, c]$ , where  $v$  is the name of a variable, and  $c$  is a constant denoting the value of that variable, depending on the point in time. Updates to the patient record are done by appending a pair to the end of the sequence. Moreover, the most recent value of a variable  $v$  in a sequence  $s$  is given by the term  $s[v]$ . An example of the final translation can be found in Fig. 8.

## 7.4 Specification of Asbru in KIV

As each Asbru plan has a strict format, an algebraic function "mk-asbru-def" has been defined for the translation of Asbru plans into KIV specifications. By calling "mk-asbru-def" with the parameters that constitute a plan, the translation of any guideline in Asbru becomes straightforward. The parameters consist of the various conditions that control the plan state changes, the control type of subplans, a list of subplans, a retry value (for aborted plans), a wait-for condition (for mandatory subplans), and an optional wait-for flag (whether to wait for subplans). As there are quite a number of parameters, default values are provided to ease specification.

```

predicates
  Knowledge : patient × patient;
axioms
BDM2-1:
  Knowledge(pre, post) → (insulin ∈ pre[treatment] →
    post[uptake(liver,glucose)] = up ∧
    post[uptake(peripheral-tissues, glucose)] = up)
BDM2-8:
  Knowledge(pre, post) → (post[uptake(liver,glucose)] = up
    ∧ post[uptake(peripheral-tissues,glucose)] = up)
    ∧ pre[capacity(b-cells,insulin)] = exhausted
    ∧ pre[condition] = hyperglycemia →
    post[condition] = normoglycemia)

```

Fig. 8. Background knowledge in KIV as a first-order predicate using pre and postconditions. That is, *pre* and *post* are shorthand notations for patient data structures, with *pre*[*v*] = *c* and *post*[*v*] = *c* referring to the condition *v* = *c* of the patient in the current and next states, respectively. The use of *pre* and *post* variables is necessary to parameterize the background knowledge for arbitrary patient data structures. In addition, two translated rules from the background formalization in Fig. 2 are shown with BDM2-*i* representing rule (*i*).

The Asbru semantics is implemented as a parallel program parameterized with a given Asbru model. Temporal properties are proven using symbolic execution and induction [31].

## 7.5 Specification of Quality Requirements in KIV

With the help of KIV, we have verified that the diabetes guideline is proper; that is, the guideline satisfies conditions (M1) and (M2), which are discussed in detail in Sections 8.1 and 8.2. The metalevel quality requirements are verified in KIV by using a sequent  $\Gamma \vdash \Sigma$ , where the succedent  $\Sigma$  is some instantiation of (M3), and the antecedent  $\Gamma$  is a fixed structure that consists of the initial state of the patient and of the Asbru model, the Asbru model, the effects of treatments, the background knowledge, and the environment assumptions. The sequent in Fig. 9 is the specification in KIV of the quality requirement mentioned in Section 5.2 (that is, each patient is eventually cured from hyperglycemia).

The *initial states* of the *patient* and the *Asbru model* are represented using additional data structures [34]. The patient data is represented in a data structure “patient-data-history,” which, as shown in Fig. 9, is set to the patient group  $\{\text{Condition}(\text{hyperglycemia})\}$ . The initial state of the Asbru model is represented using a data structure “AS” of type “asbru state,” which keeps track of all plan states over time and in which, initially, each plan is set to inactive. The *Asbru model* of the guideline describes the control structure, and its specification in KIV has already been discussed in Section 7.4. The *effects of treatments* specify in KIV the behavior of plans in the Asbru model. This is a direct translation of the **effects** attribute used in the Asbru model (see Section 6.3). In our diabetes case study, the effects of plans are the administration of a certain drug as soon as the plan becomes activated, which may depend on the value of other variables like the QI (see Fig. 5). The *background knowledge* is represented in the sequent by using the first-order predicate “Knowledge” and has already been discussed in Section 7.3. The environment is, in principle, allowed to change every

```

/* Initial state of patient */
Patient[condition] = hyperglycemia,
/* Initial state of guideline */
AS[Treatments_and_Control] = inactive, ...,
/* Asbru model */
[asbru##(Treatments_and_Control; AS, P)],
/* Effects */
□ (AS[SU_or_BG] = activated ↔
  BG ∈ Patient'[treatment] ∧ ...),
/* Background knowledge */
□ Knowledge(Patient', Patient'')
/* Environment assumption */
□ (AS''[Treatments_and_Control] =
  AS'[Treatments_and_Control] ∧ ...)
⊢
/* Property */
◇ (Patient[condition] = hypoglycemia ∨
  Patient[condition] = normoglycemia)

```

Fig. 9. Specification in KIV of the quality requirement that each patient is eventually cured from hyperglycemia.

variable arbitrarily. The *environment assumptions* restrict the behavior of the environment. These restrictions 1) forbid the environment to change any variable, 2) force the environment to deterministically change a variable (for example, advancing a clock), and 3) guarantee certain variable assignments in a nondeterministic way (for example, the existence of a value when a signal is sent).

## 8 VERIFICATION USING KIV

This section discusses in detail the verification of the quality requirements that we have defined in Section 5.2 for the guideline for the management of diabetes mellitus type 2 by using KIV.

### 8.1 Consistency of the Formal Model

Property (M1) ensures that the formal model, including the Asbru guideline and the background knowledge, is consistent. Verifying property (M1) corresponds to verifying that the preconditions of the sequent in Fig. 9 are not contradictory. KIV is a theorem prover; as such, it can only derive theorems and cannot be used to directly show that a given set of formulas is consistent. Nonetheless, the KIV system can be used to derive a theorem that creates a strong argument to show that our model is consistent, which we will illustrate here.

The initial state is, in our case, described as a set of equations, and it has been trivial to see that they are consistent, as they do not contain any logical operators. The guideline is given as an Asbru plan. The semantics of any Asbru plan is defined in a programming language where every program construct ensures that the resulting reactive system is consistent: In every step, the program either terminates or calculates a consistent output for arbitrary input values. The Asbru plan thus defines a total function from unprimed to primed variables in every step (Fig. 6). The formula defining the effects maps the output variables of the guideline to the input variables of the patient model, which cannot violate the consistency of the resulting model.

The background knowledge defines our patient model. We consider the patient to be part of the environment,

$$post = pre[uptake(peripheral-tissues, glucose), up] \\ [uptake(liver, glucose), up] \\ [release(liver, glucose), down] \\ [secretion(b-cells, insulin), up] \\ [condition, normoglycaemia]$$

Fig. 10. Example patient adhering to the background knowledge, with *pre* and *post* denoting patient data structures. Using the algebraic sequence notation for patient data structures described in Section 7.3, *pre* denotes an arbitrary patient data structure, and *post* denotes the patient data structure equal to the patient data structure *pre*, in which certain variables are updated.

which is the relation between the primed and the double-primed variables in every step, that is, respectively, the states before and after an environment transition (see Fig. 6). If the patient model ensures that for an arbitrary primed state, there exists a double-primed state, then the overall system of the alternating guideline and environment transitions is consistent. Given an initial (unprimed) state, the guideline calculates an output (primed) state, the effects define a link between the variables of the guideline and the variables of the patient model, and the patient model reacts to the (primed) output state and yields a (double-primed) state, which acts again as the input to the Asbru guideline in the next step. The additional environment assumption in Fig. 9 does not destroy consistency, as the set of restricted variables of the environment assumption is disjoint to the set of the variables of the patient model.

It remains to ensure the consistency of the background knowledge, which is defined in terms of a predicate “Knowledge.” An additional property

$$\forall pre. \exists post. Knowledge(pre, post)$$

ensures that the relation is total. In order to verify that the properties in Fig. 2, together with the property above, are consistent, we have assumed a specific patient (see Fig. 10), for whom all possible physiological effects described by the background knowledge occur. The patient reacts to an arbitrary input state *pre*, with a raised uptake of glucose in the liver and peripheral tissues, a raised secretion of insulin from the B cells, and a lowered release of glucose from the liver. Furthermore, the condition of the patient always improves to normal. Verifying that the example patient satisfies all of the properties in Fig. 2 has been fully automatic.

## 8.2 Successful Treatment

In order to verify property (M2), that is, the guideline eventually manages all patient problems, a proof for the sequent in Fig. 9 must be constructed. The verification strategy in KIV is a symbolic execution with induction [31], [32]. The plan state model in Fig. 3 is implemented by a procedure called “asbru.” This procedure is symbolically executed. In the initial state, the top-level plan “Treatments\_and\_Control” is in the “inactive” state. After executing the first step, the plan is “considered.” Execution continues, as described by the plan state model in Fig. 3, and produces the proof tree in Fig. 11. Starting from the bottom, the proof consists of a number of step execution rules followed by simplification rules to simplify the first-order formulas describing the current state. Each proof step is represented

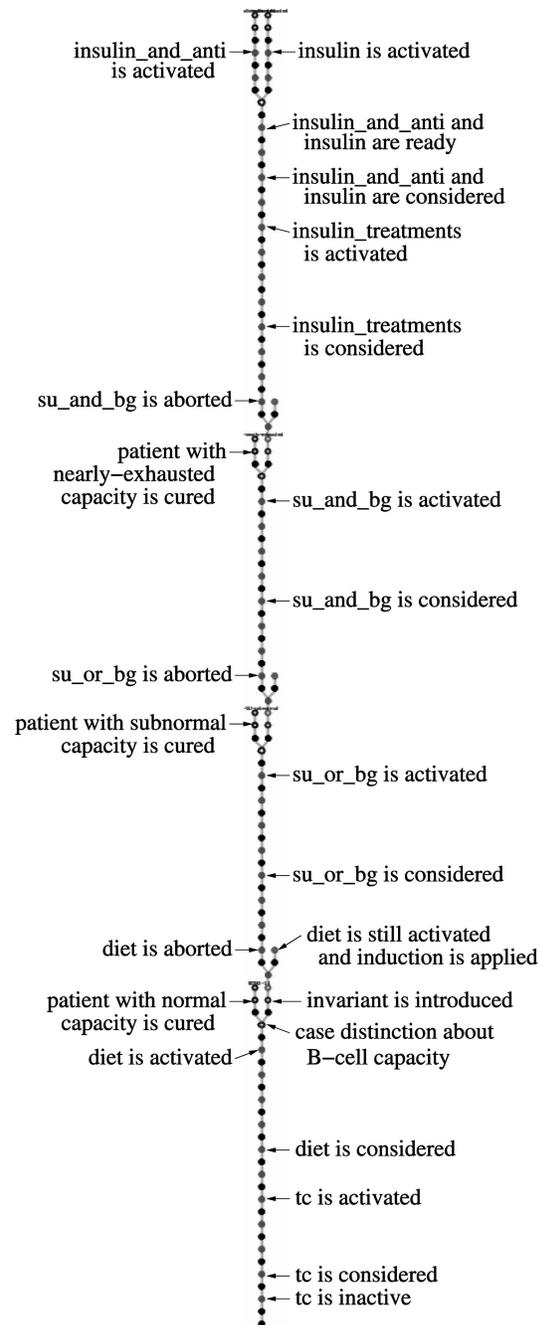


Fig. 11. Annotated proof tree for property (M2), that is, the guideline eventually manages all patient problems. Proofs are started from the bottom, with each subsequent bullet representing a proof step. Different branches represent case distinctions. The term “tc” stands for “Treatments\_and\_Control.”

graphically with a bullet. The final proof tree contains all of the possible execution paths of the guideline.

After the “Treatments\_and\_Control” plan is activated, the first subplan “diet” is considered. Execution continues until the “diet” plan is activated. The axioms in Fig. 2 do not contain any knowledge about how the diet effects a patient. We have therefore added the following axiom:

$$(10) \quad diet \wedge Patient[capacity(b-cells, insulin)] = normal \\ \rightarrow Patient[condition] = normoglycemia.$$

Patients whose capacity of the B cells is *normal* are cured with the diet (left branch of case distinction). For other patients, the diet may not be sufficient (right branch of case distinction). In this case, we assume that the doctor eventually aborts the diet treatment. We use induction to reason about the unspecified time period that a diet is followed by the patient. As an invariant

$$\text{Patient}[\text{capacity}(b\text{-cells}, \text{insulin})] \neq \text{normal}$$

is used. In the next step, the doctor has either aborted the “diet” (left branch), or the “diet” is still active (right branch). In the second case, induction can be applied. If the first treatment is “aborted,” then the second treatment “SU\_or\_BG” is “considered,” and after some steps, it is “activated.” In this case, either SU or BG is prescribed, depending on the QI. For a patient whose B cell capacity is subnormal, the background knowledge ensures that the condition of the patient improves (properties (3), (4), (5), and (6) in Fig. 2). Thus, for the rest of the proof, we can additionally assume that

$$\text{Patient}[\text{capacity}(b\text{-cells}, \text{insulin})] \neq \text{subnormal}.$$

The third treatment (“SU\_and\_BG”) is similarly executed, and because of properties (3), (4), and (7) of the background knowledge, patients with nearly exhausted B cell capacity are cured. Thus, the precondition concerning the capacity of the B cells can be strengthened to

$$\begin{aligned} & \text{Patient}[\text{capacity}(b\text{-cells}, \text{insulin})] \neq \text{normal} \\ & \wedge \text{Patient}[\text{capacity}(b\text{-cells}, \text{insulin})] \neq \text{subnormal} \\ & \wedge \text{Patient}[\text{capacity}(b\text{-cells}, \text{insulin})] \neq \text{nearly-exhausted}. \end{aligned}$$

Here, we require an additional axiom, which says that  $\text{capacity}(b\text{-cells}, \text{insulin})$  is a function and therefore can only obtain one of the values from the set

$$\{\text{normal}, \text{subnormal}, \text{nearly-exhausted}, \text{exhausted}\}$$

to conclude that

$$\text{Patient}[\text{capacity}(b\text{-cells}, \text{insulin})] = \text{exhausted}.$$

This axiom, together with properties (1) and (8) of the background knowledge, ensures that the prescription of insulin finally cures the patient.

### 8.3 Optimality of Treatment

With respect to property (M3), an optimality criterion of the guideline is that no treatments are prescribed that are not in accordance with good medical practice (Section 5.1); that is, some preference relation  $\preceq$  between treatments exists, and the guideline never prescribes a treatment  $T$ , with  $T \preceq T'$ , and  $T'$  being sufficient for reaching the physician’s intentions for the patient group in question.

In our case study, the preference for treatments is based on the minimization of 1) the number of insulin injections and 2) the number of drugs involved (see Section 5.1). We have defined this by using a reflexive transitive order  $\leq$  such that for all treatments  $T$ , it holds that  $\{\text{insulin}\} \leq T$  and  $T \leq \{\text{diet}\}$ . Furthermore, the treatments prescribing the oral antidiabetics SU and BG are incomparable. The proof obligation is then given as follows:

```

/* Initial state of guideline */
AS[Treatments_and_Control] = inactive, ...,
/* Asbru plan */
[asbru#(Treatments_and_Control, st; AS, Patient)],
/* Effects */
□ (AS[SU_or_BG] = activated ↔
  BG ∈ Patient'[treatment] ∧ ...),
/* Background knowledge */
□ Knowledge(Patient', Patient'')
/* Environment assumption */
□ (AS'[Treatments_and_Control] =
  AS'[Treatments_and_Control] ∧ ...)
⊢
/* Property */
□ (∀T : Good≤(T, Patient) → ¬ (Patient[treatment] < T)).

```

Furthermore, we needed to add the following axiom to our system:

$$\square \text{Patient}[\text{QI}] = \text{Patient}''[\text{QI}],$$

that is, the QI does not change during the execution of the guideline. This axiom is needed because the decision to prescribe a treatment is not exactly made at the same time as the actual application of the treatment and, therefore, the decision to prescribe this treatment could be based on a patient with a QI different from the patient that takes the drugs.

Proving this property in KIV was done in approximately one day by using particular heuristics for the straightforward parts, for example, propositional simplification and symbolic execution. The theorem was proven using two lemmas for two specific patient groups. In total, it took approximately 500 steps to verify this property, with a degree of automation of approximately 90 percent. The verification process yields an insight in the inferences needed to construct the proof, which provides the opportunity to construct case-specific heuristics. This improves the level of automation; however, the verification of other properties will not necessarily benefit from the additional heuristics.

### 8.4 No New Treatments

The previous property does not rule out that the guideline for diabetes mellitus type 2 prescribes additional treatments in case the patient is cured. This can be formalized as follows:

$$\begin{aligned} & \square \forall T_s (\text{Tick} \wedge T_s = \text{Patient}[\text{treatment}] \wedge \\ & \quad \circ \text{Patient}[\text{condition}] \neq \text{hyperglycemia} \rightarrow \\ & \quad \bullet \square ((\neg \text{last} \wedge \text{Tick}) \rightarrow \text{Patient}[\text{treatment}] = T_s)). \end{aligned}$$

To compare the current treatment with any future treatment, a static variable  $T_s$  is used to store the current treatment administered to patient “Patient,” as “Patient” may dynamically change. The variable  $T_s$  only needs to be compared with future treatments in case the patient is cured. Because curing the patient requires one time step in

the formalization of the background knowledge, to check if treatment  $T_s$  has cured the patient, we need to check whether the patient's condition is different from hyperglycemia in the *next* state by using the  $\circ$  modal operator (see Table 1). When both conditions hold, either the execution of the guideline should complete or, when it does not, only treatment  $T_s$  should be administered. The "Tick" variable is introduced to restrict the property to macrosteps (see Section 7.2). As we are only interested in the temporal behavior of plan activations, the property would trivially be violated when one would allow microsteps, as microsteps do not allow temporal behavior; that is, plans are never activated in microsteps.

The effort to prove this property was of the same order as proving that the treatment is successful. The difficult part was finding the right formalization, taking into account that the execution of the guideline completes and that internal (micro)steps can violate the proof obligation.

### 8.5 Order of Treatments

Finally, it was proven that the order of any two treatments in the guideline is consistent with the order relation, as we have defined in Section 8.3. The formalization of the property in KIV was done as follows:

$$\begin{aligned} & \Box \forall T_s (\text{Tick} \wedge T_s = \text{Patient}[treatment] \\ & \rightarrow \Box (\text{last} \vee (\text{Tick} \rightarrow \neg(T_s \leq \text{Patient}[treatment])))). \end{aligned}$$

At each time, the current treatment is bound to a static variable  $T_s$ , which can be used to compare against the subsequent steps in the guideline. For any future steps, we require that either the guideline completes (last holds) or activated treatments are not more preferred than  $T_s$ . The formalization represents the property introduced in Section 5 by using convenient KIV features. Again, the variable "Tick" is needed in the formalization to abstract from technical system steps (see Section 8.4).

This property also had a high degree of automation, with roughly 800 steps in total. The reason for this slightly higher number of steps is due to nested temporal operators. However, the steps were straightforward, as the different branches have a similar structure. The only knowledge used was the axiom that states that the QI is constant during the run of the guideline. Without this assumption, it is possible that the treatment switches from "SU" to "BG" or from "BG" to "SU" during the activation of the "SU\_or\_BG" plan. This unwanted behavior would lead to a counterexample, as these two drugs are assumed to be incomparable. This shows that additional assumptions about patients may be necessary, even for properties that only state something about the structure of the guideline.

## 9 CONCLUSIONS

This study began with the premise that medical guidelines are more and more based on sound scientific evidence for promoting standards of medical care and may improve the quality of healthcare and reduce costs. However, as new scientific knowledge becomes available on a continuous

basis, and the development and management of medical guidelines currently take much effort, medical guidelines are difficult to keep up to date. This study therefore focused on the use of formal methods for quality checking of medical guidelines to support the development and management of guidelines when newly obtained evidence invalidates the quality of the guideline.

In our study, we have set up a general framework for the verification of medical guidelines, consisting of a medical guideline, a medical background knowledge, and quality requirements. A model for the background knowledge of glucose level control in diabetes mellitus type 2 patients was developed based on a general temporal logic formalization of (patho)physiological mechanisms and treatment information. Furthermore, we developed a theory for quality requirements of good medical practice based on the theory of abductive diagnosis. This theory of quality requirements and the model of background knowledge were then used in a case study, in which we verified several quality criteria of the diabetes mellitus type 2 guideline used by the Dutch general practitioners. In the case study, we used Asbru to model the guideline as a network of tasks and KIV for the formal verification.

In the course of our study, we have shown that setting up a general framework for the formal verification of medical guidelines with medical background knowledge is feasible and that the actual verification of the quality criteria can be done with a high degree of automation. We believe that both the inclusion of medical background knowledge and semiautomatic decision support are essential elements for adequately supporting the development and management of medical guidelines. Our approach allows one to reason about the guideline in terms of the effects that treatments have on patients and, consequently, it is possible to specify general requirements in terms of the *outcome* of a guideline rather than in terms of the fairly arbitrary document structure of the guideline.

Although the presented case study is small compared to some other guidelines, the framework has been set up to be scalable for larger verification studies. First, a large number of algebraic specifications in KIV dealing with the Asbru semantics and data types are reusable. Second, the translation of Asbru models into KIV can be done automatically. Third, KIV has an integrated proof maintenance system that keeps track of invalidated proofs in case of changes to specifications and even tries to correct invalidated proofs automatically. However, KIV is an expressive tool, which may result in an additional overhead when verifying the quality criteria of medical guidelines. Additional research could focus on other techniques like model checking, which may be less expressive but require less overhead, for verifying the quality criteria. Such techniques could be part of a process of quality checking guidelines, in which our approach would be at the far end of the spectrum of possible techniques, whereas simpler techniques can be used as early as the modeling of the guideline itself to remove errors and ambiguities in the guideline. This would improve scalability even further.

## APPENDIX

## NOTATION

Temporal logic operators and statements (Sections 2, 3, 4, and 5):

- $\square \varphi, \diamond \varphi, \circ \varphi, \bullet \varphi$ , last: see Table 1,
- $B$ : background knowledge,
- $T$ : treatment,
- $P$ : patient group,
- $N$ : medical intentions,
- $M$ : medical guideline,
- $\text{Drug}(x)$ : holds if and only if drug  $x$  is administered at that point in time,
- $\text{SU}$ : sulfonylurea drug,
- $\text{BG}$ : biguanide drug,
- $\text{QI}$ : Quetelet index,
- $T \preceq_{\varphi} T'$ : treatment  $T'$  is at least as preferred as treatment  $T$ , and
- $\text{Good}_{\varphi}(T, P), \text{Good}_{\varphi}(M, P)$ : treatment  $T$ , respectively medical guideline  $M$ , is in accordance with good medical practice for patient  $P$  and criteria  $\varphi$ .

Specification in KIV (Sections 7 and 8):

- $v_s, v_d$ : A static, respectively, dynamic, variable, which has a constant, respectively, changing, interpretation on each time point.
- $v'_d, v''_d$ :  $v'_d$  is the value of  $v_d$  after a system transition, and  $v''_d$  is the value of  $v'_d$  after the environment transition, that is, the value of  $v_d$  in the next state.
- $\text{Knowledge}(pre, post)$ : For patient data structures  $pre$  and  $post$ , with  $pre$  denoting the current state and  $post$  the next state of the patient, the predicate  $\text{Knowledge}$  defines the relation that must hold between  $pre$  and  $post$ .
- $s[v]$  is the value of variable  $v$  in algebraic sequence  $s$ .
- $s[v, c]$  is algebraic sequence  $s$ , where  $v$  is updated with value  $c$ .
- $\text{AS}$  is the internal state of the Asbru program.
- Tick is a macrostep in the asbru execution.

Asbru (Section 6):

- considered, possible, activated: plan states,
- suspended, aborted, completed filter, setup, complete, abort: conditions controlling the execution, and
- consider, activate: synchronizing signals.

## REFERENCES

- [1] S.E. Strauss, W.S. Richardson, P. Glasziou, and R.B. Haynes, *Evidence-Based Medicine—How to Practice and Teach EBM*. Churchill Livingstone, 2005.
- [2] S.H. Woolf, "Evidence-Based Medicine and Practice Guidelines: An Overview," *Cancer Control*, vol. 7, pp. 362-367, 2000.
- [3] D. Dickenson and P. Vineis, "Evidence-Based Medicine and Quality of Care," *Health Care Analysis*, vol. 10, pp. 243-259, 2002.
- [4] S. Woolf, R. Grol, A. Hutchinson, M. Eccles, and J. Grimshaw, "Potential Benefits, Limitations, and Harms of Clinical Guidelines," *British Medical J.*, vol. 318, pp. 527-530, 1999.
- [5] P. Clayton and G. Hripsak, "Decision Support in Healthcare," *Int'l J. Biomedical Computing*, vol. 39, pp. 59-66, 1995.
- [6] P.A. de Clercq, J.A. Blom, H.H.M. Korsten, and A. Hasman, "Approaches for Creating Computer-Interpretable Guidelines That Facilitate Decision Support," *Artificial Intelligence in Medicine*, vol. 31, no. 1, pp. 1-27, 2004.
- [7] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R.A. Greenes, R. Hall, P.D. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E.H. Shortliffe, and M. Stefanelli, "Comparing Computer-Interpretable Guideline Models: A Case-Study Approach," *J. Am. Medical Informatics Assoc.*, vol. 10, no. 1, pp. 52-68, 2003.
- [8] J. Fox and S. Das, *Safe and Sound: Artificial Intelligence in Hazardous Applications*. AAAI Press, 2000.
- [9] M. Peleg, A. Boxwala, O. Ogunyemi, P. Zeng, S. Tu, R. Lacson, E. Begnstam, and N. Ash, "GLIF3: The Evolution of a Guideline Representation Format," *Proc. Am. Medical Informatics Assoc. Ann. Symp. (AMIA '00)*, pp. 645-649, 2000.
- [10] J. Fox, N. Johns, A. Rahmzadeh, and R. Thomson, "PROforma: A Method and Language for Specifying Clinical Guidelines and Protocols," *Proc. 13th Medical Informatics Europe Int'l Congress (MIE '96)*, J. Brender, J. Christensen, S. J.R., and P. McNair, eds., pp. 516-520, 1996.
- [11] J. Fox, N. Johns, A. Rahmzadeh, and R. Thomson, "PROforma: A General Technology for Clinical Decision Support Systems," *Computer Methods and Programs in Biomedicine*, vol. 54, pp. 59-67, 1997.
- [12] A. Seyfang, R. Kosara, and S. Misch, "Asbru's Reference Manual, Asbru Version 7.3," Technical Report Asgaard-TR-20002-1, Inst. of Software Technology, Vienna Univ. of Technology, 2002.
- [13] Y. Shahar, S. Miksch, and P. Johnson, "The Asgaard Project: A Task-Specific Framework for the Application and Critiquing of Time-Oriented Clinical Guidelines," *Artificial Intelligence in Medicine*, vol. 14, pp. 29-51, 1998.
- [14] S. Tu and M. Musen, "A Flexible Approach to Guideline Modeling," *Proc. Am. Medical Informatics Assoc. Ann. Symp. (AMIA '99)*, pp. 420-424, 1999.
- [15] S. Tu and M. Musen, "From Guideline Modeling to Guideline Execution: Defining Guideline-Based Decision-Support Services," *Proc. Am. Medical Informatics Assoc. Ann. Symp. (AMIA '99)*, pp. 863-867, 1999.
- [16] M. Peleg, L. Gutnik, V. Snow, and V.L. Patel, "Interpreting Procedures from Descriptive Guidelines," *J. Biomedical Informatics*, vol. 39, no. 2, pp. 184-195, 2006.
- [17] P.J.F. Lucas, "Quality Checking of Medical Guidelines through Logical Abduction," *Proc. 23rd SGAI Int'l Conf. Innovative Techniques and Applications of Artificial Intelligence (AI '03)*, F. Coenen, A. Preece, and A. Mackintosh, eds., pp. 309-321, 2003.
- [18] M. Marcos, M. Balsler, A. ten Teije, and F. van Harmelen, "From Informal Knowledge to Formal Logic: A Realistic Case Study in Medical Protocols," *Proc. 13th Int'l Conf. Knowledge Eng. and Knowledge Management (EKAW '02)*, pp. 49-64, 2002.
- [19] A.J. Hommersom, P.J.F. Lucas, and M. Balsler, "Meta-Level Verification of the Quality of Medical Guidelines Using Interactive Theorem Proving," *Proc. Ninth European Conf. Logics in Artificial Intelligence (JELIA '04)*, pp. 654-666, Sept. 2004.
- [20] M. Balsler, W. Reif, G. Schellhorn, K. Stenzel, and A. Thums, "Formal System Development with KIV," *Proc. Third Int'l Conf. Fundamental Approaches to Software Eng. (FASE '00)*, T. Maibaum, ed., 2000.
- [21] *Clinical Practice Guidelines: Directions for a New Program*, M. Field and K. Lohr, eds. Inst. of Medicine. Nat'l Academy Press, 1990.
- [22] S. Miksch, "Plan Management in the Medical Domain," *AI Comm.*, vol. 12, no. 4, pp. 209-235, 1999.
- [23] G.E.H.M. Rutten, S. Verhoeven, R.J. Heine, W.J.C. de Grauw, P.V.M. Cromme, and K. Reenders, "NHG-Standard Diabetes Mellitus Type 2 (eerste herziening)," *Huisarts Wet.*, vol. 42, pp. 67-84, 1999.
- [24] E.A. Emerson, "Temporal and Modal Logic," *Handbook of Theoretical Computer Science: Formal Models and Semantics (B)*, pp. 995-1072, 1990.
- [25] R. Reiter, "Equality and Domain Closure in First-Order Databases," *J. ACM*, vol. 27, pp. 235-249, 1980.
- [26] P. Magni, R. Bellazzi, G. Sparacino, and C. Cobelli, "Bayesian Identification of a Population Compartmental Model of C-Peptide Kinetics," *Annals of Biomedical Eng.*, vol. 28, pp. 812-823, 2000.
- [27] M. Balsler, C. Duelli, W. Reif, and J. Schmitt, "Formal Semantics of Asbru—v2.12," technical report, Univ. of Augsburg, June 2006, <http://www.informatik.uni-augsburg.de/lehrstuehle/swt/se/publications/>.
- [28] M. Balsler, C. Duelli, and W. Reif, "Formal Semantics of Asbru—An Overview," *Proc. Sixth Biennial World Conf. Integrated Design and Process Technology (IDPT '02)*, 2002.

- [29] G. Schellhorn and W. Ahrendt, "Reasoning about Abstract State Machines: The WAM Case Study," *J. Universal Computer Science*, vol. 3, no. 4, pp. 377-413, <http://hyperg.iicm.tu-graz.ac.at/jucs/>, 1997.
- [30] G. Schellhorn, H. Grandy, D. Haneberg, and W. Reif, "The Mondex Challenge: Machine-Checked Proofs for an Electronic Purse," *Proc. 14th Int'l Symp. Formal Methods (FM '06)*, J. Misra, T. Nipkow, and E. Sekerinski, eds., pp. 16-31, 2006.
- [31] M. Balsler, "Verifying Concurrent Systems with Symbolic Execution—Temporal Reasoning Is Symbolic Execution with a Little Induction," PhD dissertation, Univ. of Augsburg, Augsburg, Germany, 2005.
- [32] M. Balsler, C. Duelli, W. Reif, and G. Schellhorn, "Verifying Concurrent Systems with Symbolic Execution," *J. Logic and Computation*, vol. 12, no. 4, pp. 549-560, 2002.
- [33] J. Schmitt, M. Balsler, and W. Reif, "Support for Interactive Verification of Asbru in KIV," Technical Report 2006-16, Inst. of Informatics, Univ. of Augsburg, June 2006.
- [34] J. Schmitt, M. Balsler, and W. Reif, "Complementary Material to Deliverable D4.2b: Improved Verification System," *Protocure II—Integrating Formal Methods in the Development Process of Medical Guidelines and Protocols*, 2005.



**Arjen Hommersom** received the master's degree in computer science from the University of Utrecht in 2003. He is a PhD student under the supervision of Peter Lucas at the University of Nijmegen. From January 2004 to July 2006, he was involved in the Protocure II project, which dealt with the integration of formal methods in the life cycle of medical guidelines. His research interests include knowledge representation and reasoning, in particular, in the use of formal methods.



**Perry Groot** received the degrees in computer science (artificial intelligence) and mathematics (topology) in 1998 and the PhD degree from the Department of Artificial Intelligence, Vrije University Amsterdam in 2003. Currently, he is a researcher in the Institute of Computing and Information Sciences at Radboud University Nijmegen, the Netherlands. His research interests include knowledge representation and reasoning, and symbolic problem solving, with a current focus on applications in the medical domain.



**Peter J.F. Lucas** is an associate professor in the Institute of Computing and Information Sciences at Radboud University Nijmegen, the Netherlands. He has been involved in the research on artificial intelligence, in particular knowledge-based systems, since the beginning of the 1980s. He has contributed to this area by theoretical and applied research, with the latter (for the major part) focusing on the field of medicine. At the moment, his research interests include applied logic and theorem proving, knowledge representation, decision-support systems, model-based diagnosis, Bayesian networks, and statistical machine learning. He has extensively published in computing science and medical informatics journals and conferences (more than 100 journal and conference publications), written and edited several books, organized a number of workshops in the field, and edited six thematic issues of journals on the topics mentioned above.



**Michael Balsler** received the PhD degree from the University of Augsburg in 2005 for his work on quality assurance of concurrent systems with formal methods. He has carried over the idea of symbolic execution to the verification of temporal properties for concurrent systems. The resulting interactive proof method has been implemented in KIV, an interactive verification environment. Since 2005, he has been working as a freelancer, with a special interest in software engineering and quality improvement.



**Jonathan Schmitt** received the degree in computer science from the University of Augsburg, Germany, in 2004. In his diploma thesis, he worked on techniques for the verification of concurrent systems with temporal logic. Until June 2006, he worked as a research assistant on reasoning about correctness of medical protocols under the supervision of Professor Reif in the Department of Computer Science at the University of Augsburg, where he has been working as a teaching assistant since July 2006. He is currently working on his PhD dissertation, concerned with the verification of medical guidelines. His main research interests include formal methods and software engineering.

▷ **For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).**