International Journal on Artificial Intelligence Tools
Vol. 23, No. 3 (2014) 1460003 (24 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0218213014600033

Representation of Medical Guidelines with a Computer Interpretable Model

Alena Simalatsar[*,§] Wenqi You[*,¶], Verena Gotta[†,‡], Nicolas Widmer[†,‖]
and Giovanni De Micheli[*,**]

[*]*École Polytechnique Fédérale de Lausanne, Switzerland*
[†]*Division of Clinical Pharmacology,*
Centre Hospitalier Universitaire Vaudois and
University of Lausanne, Switzerland
[‡]*School of Pharmaceutical Sciences,*
University of Lausanne and Geneva, Switzerland
[§]*alena.simalatsar@epfl.ch*
[¶]*wenqi.you@epfl.ch*
[‡]*v.gotta@lacdr.leidenuniv.nl*
[‖]*nicolas.widmer@chuv.ch*
[**]*giovanni.demicheli@epfl.ch*

Received 17 April 2013
Accepted 3 February 2014
Published 28 May 2014

Nowadays medical software is tightly coupled with medical devices that perform patient state monitoring and lately even some basic treatment procedures. Medical guidelines (GLs) can be seen as specification of a medical system which requires their computer-interpretable representation of medical GLs. Until now most of the medical GLs are often represented in a textual format and therefore often suffer from such structural problems as *incompleteness*, *inconsistencies*, *ambiguity* and *redundancy*, which makes the translation process to the machine-interpretable language more complicated. Computer-based interpretation of GLs can improve the quality of protocols as well as the quality of medical service. Several GLs formal representation methods have been presented recently. Only some of them enable automatic formal verification by introducing an additional translation path to the existing model checking environments. However, if a verified property fails it is difficult to trace back the result needed to change the model. Moreover, these formalisms provide the notion of time mostly in terms of actions order. In this paper we preset the application of a well-know formal behaviour representation approach of embedded systems design domain to medical GLs interpretation. We use Timed Automata extended with Tasks (TAT) and TIMES toolbox to represent medical GLs as a system behaviour in a computer interpretable form. We discuss the verification issues with the help of the anticancer drug *imatinib* case study.

Keywords: Medical guidelines; formalization; timed automaton; verification.

[§]Corresponding author

1. Introduction

Recently, the set of electronic medical devices has a tendency to be extended with a new class of closed-loop/autonomous devices able to not only acquire the vital information but also perform some basic patient treatment.[21] Involvement of such devices in the health care process reduces human factor errors, unfortunately, often with the price of introduction of new errors due to failures or unpredictable behaviours of the electronic systems. Medical Guidelines (GLs) contain step-by-step recommendations for practitioners about how to treat a patient. Therefore, they represent an informal control flow that synchronizes the processes of data acquisition, decision-making and treatment provision and thus can act as an intersection point between medical software and electronic devices, thus playing a role of a medical system specification.

Until now, most of the GLs are often represented in a textual format and often suffer from such structural problems as incompleteness, inconsistency, ambiguity and redundancy. Therefore, it is essential to find a proper representation for the GLs that would enable the validation of the GLs formal properties. Several frameworks for the computer-based interpretation of GLs have been presented in the past three decades.[15,20,29,32,33] These tools adapt flow-charts as a core formalism to represent a sequence of actions. However, they have no support for the automatic verification of the protocol formal properties. Such frameworks as GLARE[31] and Asbru[17,24] provide translation links to model checking environments such as SPIN[1] and SMV.[2] However, if a verified property fails it is difficult to trace back the result needed to change the initial protocol model. Moreover, these formalisms provide the notion of time only in terms of actions order (in a flow-chart). However, it is essential to map medical actions into the time scale with respect to both medical software, when over-time response definition to a treatment needs to be given, and electronic device, when parts of the protocol are performed by this device, and thus system real-time properties need to be verified. We believe that cooperation of the two domains of medical informatics and medical cyber-physical systems is an important step in practitioners assistance that is aimed to help them taking decisions and automate some routine actions thus reducing the number of mistakes due to the human factor.

In this paper we show that a well-known formal behaviour representation approach of embedded systems design domain that employs Timed Automaton extended with Tasks (TAT)[9] can be used to represent medical GLs. Medical GLs, in turn, can serve as a formally represented behaviour of a medical system. We show how TAT is used to formalize not only action-based step-by-step procedures[28] of a medical GL but also a response to the treatment definition by means of response level observers. TIMES toolbox[3] is a model-checker that implements TAT and is aimed to support the modeling and verification of real-time systems. TIMES provides not only a model-checker engine that supports automatic verification but also a Graphical User Interface (GUI) that facilitates model modifications, verification and also simplifies the dialog between medical and embedded system domains. We present a case study of modeling the anticancer drug *imatinib* dose adjustment part of the protocol for adult patients with newly diagnosed Philadelphia positive (Ph+) Chronic Myeloid Leukemia (CML). The model is complemented with TAT-based models of TAT observers guarding the defined response levels. We perform the validation of the protocol structure.

The paper is organized as follows. Section 2 gives an overview of the existing formalisms. In Section 3 we define the requirements to the modeling methodology and introduce the TA and TAT models as our approach while showcasing it with a small example. We discuss the advantages and disadvantages of TAT-based approach. Section 4 presents the *imatinib* case study, concluding it with verification issues are discussed in Section 4.2. Section 5 presents the extension of the original *imatinib* protocol with "rescue TDM". Section 6 concludes the paper.

2. Related Work

In Ref. 21 authors present a formal approach to the development of a Generic Patient Controlled Analgesic (GPCA) infusion pump. Similar to our idea they approach the problem of the safety-assured development of the pump software by using TA model. The behaviour is then verified with respect to a set of generic safety requirements. After the verification process they automatically generate a platform independent code using the TIMES tool that is then adapted for a specific platform. This work presents an example of using standard Model-driven development (MDD) approach to a new class of embedded systems, medical devices. In this paper we use the TA modeling to formally represent the step-by-step procedures of the medical GLs that can be viewed as a medical system behaviour specification.

A number of specific languages and tools[4,15–18,20,24,27,29,31–34] aimed to perform formalization of medical guidelines (GL) has been developed in the past decades. Some of these tools provide the recommendation for the structural representation of the GL in textual format such as AGREE[4] and GME.[27] Others, such as GMT,[18,34] play the role of the text markup tools. However, these tools only assist designers in representing medical protocols in one of the flow-charts supported by executable engines[29,20,33,15,32,16,31] and Refs. 17 and 24 representing a big class of decision-support tools. PRODIGY[20] introduced in 1996 was the first knowledge-based decision-support system. Its model is organized as a network of patient scenarios, management decisions and action steps, which produce further scenarios called a *disease-state map*. Its development has been already discontinued, however, it has created a fruitful base for other knowledge-based decision-support tools. EON[33] is a component-based suite for GLs modeling and creation of guideline-based applications. The EON architecture is composed of Dharma and RESUME problem-solving methods as well as a temporal query system called Chronus. The Dharma model is divided into two parts the first of which determines the eligibility of a patient for a treatment procedure (diagnosis phase), while the second one, called *therapy planner*, represents the treatment procedure. Similar to PRODIGY, disease-state map approach of the *therapy planner* is based on an abstract *skeletal-plan.* It is then gradually refined using patient condition specific details provided by RESUME that are then assigned to the skeletal plan elements as attributes. Chronus is a temporal query system that provides patient's data stored in electronic medical-record systems when the history of the disease progression is important. The Guideline Interchange Format (GLIF)[15] was developed to support guideline modeling as a flow-chart, showing the steps as boxes of various kinds, and their order by connecting them with arrows. However, GLIF2 flowcharts attributes were represented in plain text, which introduces a problem in translation of GLIF models into computable formalisms. The current version of GLIF (GLIF3) is similar to GLIF2, however, a formal structure for the class attributes is also provided. It also introduces a hybrid approach by combining ontology classes that provide parameters, such as medication name, dose and administration frequency, with a structured description of the medical actions.

GUIDE[16] is focused on providing an integrated medical knowledge management through a unique central system and consists of three independent modules: Guideline Management System (GlMS) (providing clinical decision support), Electronic Patient Record (EPR) and Workflow Management System (WfMS) or Careflow Management System (CfMS) (providing organizational support). The GUIDE graphical editor is a part of the whole environment used to formally represent a general GL as flow-charts that involve medical terms and concepts. This GL can then be instantiated by an end user for the management of an individual patient by annotating it with patient data. From the point of view of the GL representation GUIDE is similar to EON and GLIF by exploiting flow-charts to represent the sequence of actions and ontologies for medical terminology and concept representation. However, it is more focused on data centralization and distribution and thus goes further with formalizing the data structure, using GEM,[27] and data access representation. In our work we are more interested in the part of the protocol modeling equivalent to the *therapy planner* of EON. None of the mentioned above tools support GLs formal verification.

The GLARE[31] system is based on a modular architecture, which includes an acquisition and an execution tool. Similar to other formalisms, GLARE separates the concerns of the protocol representation (acquisition) and their execution or its application to a specific patient. The representation formalism of GLARE is based on the concept of an *action* that can be atomic or composite. Recently, GLARE was extended with a translation path into the PROMELA language accepted by the SPIN model checker. In Ref. 14 the authors provide a wide variety of the GL properties (examples) that can be verified. The idea of dividing the SPIN model into several agents is similar to the use of cooperating TA, where each TA plays a role of an agent. A GL described in Asbru[17,24] is called a *plan*, and it consists of a name, a set of arguments, and five components types: *preferences*, *intentions*, *conditions*, *effects*, and a *plan body*, which describes the actions to be executed. *Intentions*, *conditions* and world state are represented with temporal patterns. The temporal dimension of Asbru is a main advantage over other languages of this domain since it bridges the gap between the data delivered from monitoring devices (e.g. blood tests, manual examination, etc.) and the treatment plan. Plans of Asbru can be executed in parallel, sequentially, periodically or in a particular order. An Asbru plan is a hierarchical composition of nondecomposable subplans (*actions*) stored in a plan-specification library, which are executed by the user or by an external call to a computer program. AsbruView[5] is a data and plan visualization tool that has been developed specifically to support the understanding of Asbru guidelines. The formal verification of GLARE and Asbru protocols requires an additional translation into a formal model. For example, a translation path first from Asbru to the Karlsruhe Interactive Verifier (KIV), and further, to the SMV formal model checker was developed.[12] However, TIMES provides the GUI for TAT graphical representation as well as a CTL model checking engine. This excludes the necessity of developing a translator from the GLs representation formalism into a language accepted by a model checking and trace back the results of properties verification. TAT models introduce a very natural way of representing medical GL and it can also be turned to a fully synthesizable deterministic model.[9]

3. Formalization Approach

By definition, medical guidelines, also called clinical protocols, are the documents aimed to guide decisions of physicians or health personels regarding patient management (especially diagnosis and consequently medical treatment). In this paper we focus on formalizing the stage of patient medical treatment when the diagnosis has been already performed and a specific treatment protocol has been assigned to a patient. In this section we present our view to the transformation steps that need to be applied to a textual representation of the medical guideline in the automation process. The main elements of our methodological view are presented in Figure 1.

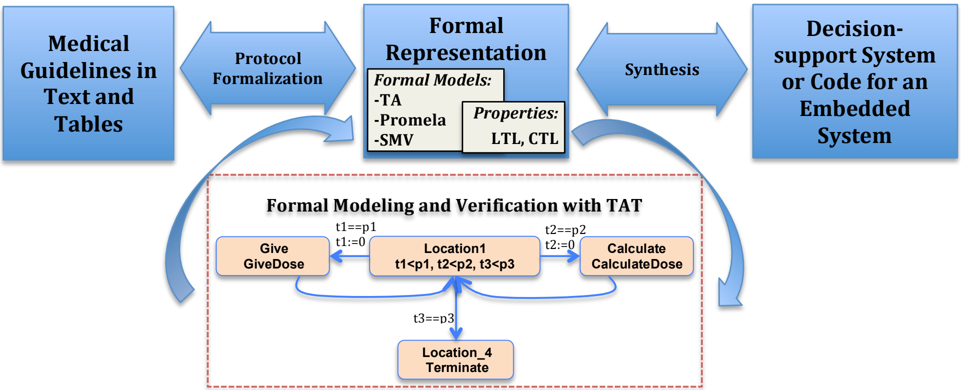


Fig. 1. General methodology.

As depicted in the picture, the transformation flow starts from a textual document summarizing sequence of actions of the medical protocol based on the experience and medical evidence. These actions as well as effects are often not instantaneous; actions may have some duration, while effects may be delayed. It is also common to see that medical protocols have some goals associated with the treatment, e.g. to achieve some effect or finish some procedure after a period of time. In order to choose an appropriate formalism one should first evaluate all its potential problems and requirements.

We distinguish several key requirements. For the protocol computer interpretation it is important to choose a formalism that first of all has a notion of time and allows modeling of the choice of sequential actions based on specific conditions mapped to the time scale. It should also allow the modeling of several plans with periodically repeating actions that can be executed in parallel with other actions. In other words, the formalism should be *expressive* enough to represent the step-by-step actions of medical protocols.

Very often the initial textual representation of the protocol suffers from: (i) *incompletenesses*, insufficient information or entirely missing pieces of information; (ii) *inconsistencies*, elements of the guidelines that can result in different/conflicting decision given the same input data, (iii) *ambiguity*, possible ambiguity of interpretation of some term used in the protocol (guideline), and (iv) *redundancies*, there might be parts of the protocol that do not change the resulting decision/prescription and thus can be skipped for the computer interpretation. Therefore, a chosen formalism should be able to assist a designer in correcting these problems. Basically, the first transformation step performs the initial structural validation of the protocol by representing it using one of the existing formalisms, since formal representation can also be considered a formal method by itself. When the protocol is formally represented it is possible to perform an automatic verification of the protocol structural properties, such as the reachability/non reachability of some states, or to be able to find a path that would avoid certain actions, e.g. surgery or chemotherapy. This introduces the requirements to the verification abilities of the methodology. A big variety of properties to be verified can be found in Refs. 14, 25 and 30. The central part of Figure 1 shows the verification process with a simple example described in details in Section 3.3.

Nowadays, medical software is tightly coupled with the medical devices that perform patient state monitoring and even some basic treatment procedures.[10] The main control flow of the electronic devices as well as of the complementary decision-support system should be based on the parts of the same medical protocol. The rightmost step of Figure 1 represents the synthesis step that may be performed in two different directions: (i) to produce a decision-support tool similar to the idea of Ref. 26 or (ii) to do the code synthesis for an embedded system as in Ref. 21. Patient treatment procedure often includes the combination of standard medical treatment GLs. Therefore, it is important to be able to synthesize a patient oriented personalized decision-support tool from a complex model of cooperating formal representations of the GLs. This way, the formal model should be *compositional* and *synthesizable*, which would allow one to generate the executable code of a personalized tool adapted for a patient's conditions.

When synthesizing a decision-support tool we could create a framework that would assist a doctor in suggesting the steps in time that need to be taken in general patient treatment procedure. Patient's conditions include not only the parameters of his/her state but also the combination of diseases and therefore combination of treatment procedures with their effects that may interfere, and thus create other complications, not related to the initial disease. The properties of the complex protocol should be verified taking into account all the elements of the model (all involved treatment procedures).

When developing a smart embedded medical device we would like to synthesis the code that would be executed on a specific embedded platform. The tasks executed on a chip could either perform the computation of specific values (dose or drug administration period), communicate with an external server that performs this computation in order to receive the updated values, trigger the operational code that reads and analyzes data coming from the sensor or activates the actuators of the chip.

In order to bridge the gap between embedded system design and medical domains we have to make these worlds speak the same language. The language of the framework chosen for formal representation should be simple enough to be understandable by someone not familiar with its syntax. Therefore, a graphical interpretation is a big advantage. We use TAT as a key formalism for protocols representation,[28] since the TIMES tool provides not only a GUI for system modeling but also a model-checker engine that supports verification of system properties that are expressed with CTL logic. A medical GL represented using TAT can be turned into a fully deterministic model[9] thus enabling further embedded or decision-support system code synthesis step.

3.1. *Timed Automaton extended with Tasks*

Timed Automaton (TA)[8] is a formal model of computation used to describe a system behaviour and its progress in time. TA is an extension of the classical Automaton that is a finite state graph composed of the finite set of locations Loc and transition relations (edges) $\hookrightarrow$. TA extends the classical Automaton with the finite set of clocks C and a set of constraints over clocks $ClockCons(C)$, where constraints are conjunctions, disjunctions and negations of atomic expressions over clocks in the form $x \bowtie n$, $x \in C$, $n \in \mathbb{N}_0$ for $\bowtie \in \{<, \leq, >, \geq, =\}$. Each location is characterized by an *invariant* (I) that specifies a constraint on a clock under which TA can stay in this location and/ or enforce a transition to another location. An edge of TA $e = (l, g, a, r, l') \subseteq \hookrightarrow$ represents a transition from l to l' ($l, l' \subseteq Loc$), where g is a guard of e, which indicates when the transition can be executed, r is the set of clocks that is reset when the edge is taken, and a is the action of e, $a \subseteq Act$.

The timed model checker UPPAAL,[13] a precursor of TIMES tool,[3] implements TA extended with variables. Similarly to clocks, variables can be used within guards of edges and location invariants. Upon a transition, variables can be updated with values of a finite set, which, however, does not need to be known beforehand, i.e. it can be constructed on-the-fly upon the state space traversal. A set of cooperating TA is called a network of TA. The cooperation mechanism may either make use of shared (global) variables or be realized as joint execution of dedicated transitions, denoted as rendezvous synchronization.

TAT[9] is an extension of TA with tasks that represent pieces of code associated with locations of the model. The execution semantics of TAT is the one of TA extended with a task queue. Any time the task is triggered by a transition it is added to the task queue, after which it will be executed upon a chosen scheduling policy.

3.2. *Modeling with TAT*

A simple example of modeling a step-by-step medical action with TAT is presented in Section 3.3. One TAT can describe one instance of a medical GL. A combination of GLs applied to one patient can be composed from a set of protocol instances represented with a network of TAT cooperating with each other. GLs modeling with TAT is more general than any of the existing GLs representation formalisms and may create a larger number of design choices. However, it provides a high flexibility in choosing levels of abstraction. Various plans of the medical GLs can be represented with corresponding TAT models. Separate components hierarchically embedded in the plan can be represented either as a network of TAT or tasks of those TAT models. The synchronization among TATs as well as their parallel execution is then realized using the cooperation mechanism of TA. The sequential, cyclical and iterative steps of plans are explicitly modeled in TAT as a sequence of TAT conditional locations connected with guarded transition relations.

Each formalized medical protocol should have entry and exit points. The entry point in TAT is represented by an initial location and signifies the beginning of treatment procedure. The exit points are the locations that determine the end of the treatment that can be classified by positive and negative results. A positive result would mean that the *goal* of the treatment procedure was achieved and thus the treatment can be stopped. The negative exit point indicates that the treatment has failed and a change of protocol is required.

Actions, such as observations, medications, procedures, analytical computations or drug treatment modifications in TAT-based GLs, are easily represented as tasks of the model. The notion of time that performs the mapping of the action into the time scale is explicitly modeled with clock guards of the model. The choice of an action that depends on some specific conditions may change with time since the values of the parameters involved in these conditions can be updated at any transition. Decisions on choosing among all possible paths in the TAT-based GL models are taken upon guards of the transition relations.

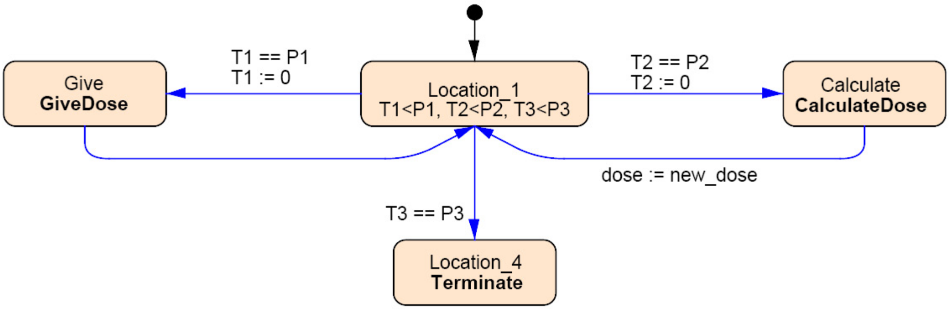


Fig. 2. An example of an abstract protocol represented by TAT.

3.3. *Simple protocol example*

We give a simple example of an abstract protocol in order to show the modeling steps and abilities of TAT to represent main features of an executable protocol. TAT-based model of this protocol is presented in Figure 2.

The model with the initial *location*_1 describes an action of periodic drug delivery. T1, T2 and T3 are the model clocks that are always compared with P1, P2 and P3 periods respectively. Periods can be annotated with any desirable values, for example 1 day, 3 days and 180 days (half a year) respectively. The model activates a drug dose delivery with period P1 (transition to the location *Give*). The transition from the *Give* location back to *location*_1 will be taken right after the *GiveDose* task is added to the task queue. The dosage value is adjusted every P2 period (transitions from *location*_1 to *Calculate* location and back) according to the current patient state. The *CalculateDose* task associated with *Calculate* location either calculates the drug dose according to the patient state or gets it delivered from another machine or a database. When the dose is calculated it is assigned to the model *dose* variable on the return transition from *Calculate* location to *location*_1. The treatment is finished after the period P3 (the transition from the *location*_1 to *location*_4). Transition to the final *location*_4 can also be guarded by a patient condition instead of a temporal guard. This example represents a part of controllable treatment procedure that can actually be performed by a medical electronic device.

4. Imatinib Case Study

In this section we first show the modeling of an action-based procedure of a part of the anticancer drug *imatinib* medical GL describing the step-by-step *a posteriori* dose adjustment (see Section 4.1) for an adult patient with newly diagnosed Ph+ chronic myeloid leukemia (CML) for whom bone marrow transplantation is not considered as the first line of treatment. Further, in Section 4.2 we discuss the model verification issues and fix several problems of initial model. In Section 4.3 we extend the model by enlarging the list of quantitative adverse events. In Section 4.4

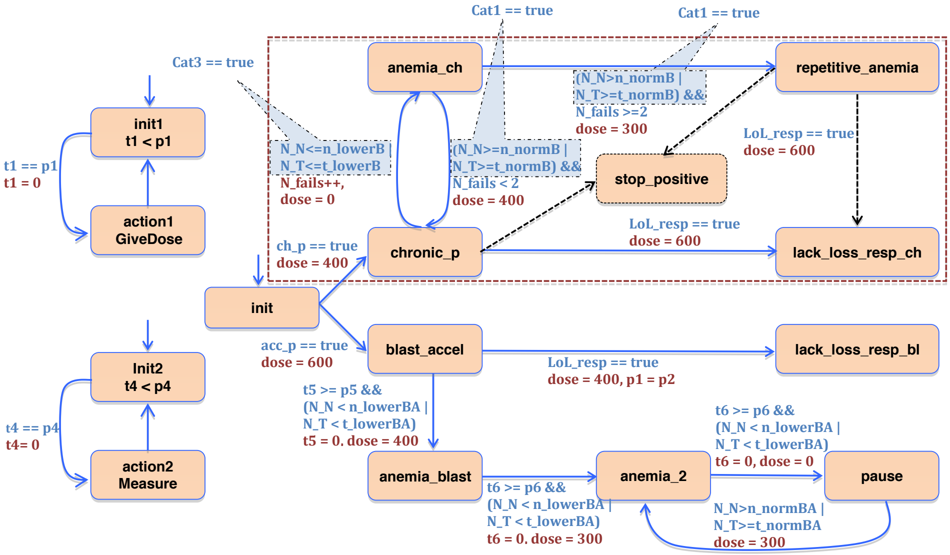


Fig. 3. *Imatinib* dose adjustment protocol.

we present the modeling of the controlling observers of the response levels built using the definition presented in Ref. 11.

4.1. *Imatinib dose adjustment and delivery models*

Imatinib, marketed by the drug manufacturer Novartis as Gleevec® or Glivec®, is a drug used to treat chronic myelogenous leukemia (CML), gastrointestinal stromal tumors (GISTs) and a number of other malignancies. The complete drug administration protocol of *imatinib* can be found in Ref. 6.

According to the protocol in Ref. 6, the prescribed dose should be administered orally once a day ($p1 = 1$ day), with a meal and a large glass of water while patients should keep normal eating habits. However, for some drugs that can be delivered intravenously the corresponding dose adjustment and drug administration could be performed by an automatic device similar to the infusion pump in Ref. 21.

Figure 3 depicts the model of drug dose adjustment and delivery based on test results. The model is represented by a network of three cooperating TAT models. The upper left TAT model in this figure consists of two locations ($init1$ and $action1$) and is responsible for the periodic drug delivery. It has location $init1$ set as the initial location. Thus every period $p1$ (when clock $t1$ is equal to $p1$), this TAT model is transiting to the $action1$ location. On this transition the clock $t1$ is set to 0 and the *GiveDose* task is activated (added to the common task queue). The transition from the $action1$ location to the $init1$ location is then taken. This model would either give a periodic reminder to a nurse to give a dose to a patient, or directly to a patient to take its drug, or send a command to a drug delivery device if this process is fully automated for intravenous administration. The second TAT model composed of two locations represents a periodic action of performing laboratory tests, that activates the task of measuring (*measure*) the level of neutrophils (N_N) and platelets (N_T) every period $p4 = 14$ days (2 weeks). The *measure* task of this model will update the values of the neutrophils (N_N) and platelets (N_T) variables of *imatinib* dose adjustment model that is described below. The original model is composed of elements with solid lines only. The dashed lines elements will be discussed below.

The dose adjustment model will use the latest data of the model performing the medical tests and adjust the dose for the model that delivers the drug. The recommended dosage of *imatinib* is 400 mg/day for patients in the chronic phase of CML (transition from the *init* to *chronic_p* location) and 600 mg/day for patients in the *accelerated* phase of CML (transition from *init* to *blast_accel* location). Therefore, the first two transitions of the model represent the choice of the treatment according to the patient condition.

The dose may be increased from 400 to 600 mg in patients with the *chronic* phase of the disease (*chronic_p* to *lack_loss_response_ch* transition relation) or from 600 mg to a maximum of 800 mg given as 400 mg twice a day (*blast_accel* to *lack_loss_response_bl* and $p1 = p2$, where $p2 = 1/2$ day) in patients with accelerated phase or blast crisis in the following circumstances:

- disease progression at any time;
- failure to achieve a satisfactory hematological or cytogenetic response;
- or loss of a previously achieved hematological and/or cytogenetic response;

In the third model all these conditions are combined into one Boolean variable $LoL\_resp == true$, meaning lack or loss of response. The definition of disease state and levels of response to the treatment is given in Ref. 11. The complementary TAT model of the response definition is presented in Section 4.4.

In the *chronic* phase of CML, marked with a rectangle in Figure 3 (starting dose 400 mg) the dose adjustments are performed as follows:

(1) If the level of neutrophils (ANC) goes below 1.0×10^9/l ($N\_N <= n\_lowerB$) and/or level of platelets goes below 50×10^9/l ($N\_T <= t\_lowerB$): stop the treatment with *imatinib* until ANC $>1.5 \times 10^9$/l ($N\_N >= n\_normB$) and platelets $> 75 \times 10^9$/l ($N\_T >= t\_normB$) (transition from *chronic_p* to *anemia_ch*);
(2) Resume the treatment with *imatinib* at previous dose, i.e. before severe adverse reaction (transition from *anemia_ch* to *chronic_p*, N_fails accounts to the number of anemia occurrences);
(3) In the event of recurrence of ANC $< 1.0 \times 10^9$/l and/or platelets $< 50 \times 10^9$/l ($N\_fails >= 2$), repeat step 1 and resume *imatinib* at reduced dose of 300 mg (transition *anemia_ch* to *repetitive_anemia*);

Treatment of a patient in the *accelerated* phase of CML or in the *blast crisis* (starting dose 600 mg) is represented in the lower part of Figure 3. If the level of neutrophils (ANC) goes below $0.5 \times 10^9/l$ ($N\_N <= n\_lowerBA$) and/or the level of platelets goes below 10×10^9/l ($N\_T <= t\_lowerBA$):

(1) If cytopenia is unrelated to leukemia, reduce dose of *imatinib* to 400 mg (transition from *blast_accel* to *anemia_blast* location);
(2) If cytopenia persists for 2 weeks ($p6$), reduce further to 300 mg (*anemia_blast* to *anemia_2*);
(3) If cytopenia persists for 4 weeks and is still unrelated to leukemia, stop *imatinib* until ANC 1×10^9/l and platelets 20×10^9/l, then resume treatment at 300 mg (*anemia_2* to *pause*).

4.2. *Model verification and correction*

The verification of a system with any model-checker requires the system be *closed*, which means that the behaviour (transitions from one state to another) of the system is completely determined by the states of itself. However, some of the transition guards of the *imatinib* protocol and response observer presented above depend on the information, such as the values of level of the *neutrophils* and *platelets* or counts of the Ph+ chromosomes coming from external patient body reaction models.

The *Module Checking* approach[22] suggests to compose an *open* system with the *maximal environment*, that enables all the external nondeterministic choices, makes the *guards* that depend on the environment always evaluated to **true**. This composition will be a closed system that contains all possible behaviours of that system combined with any other environments. Such a composition raises the level of non-determinism in the model behaviour. However, the model structure remains the same and thus can be verified.

The case study of Section 4.1 represents treatment of a chronic disease, which by definition may not be cured completely. However, there still should be a positive exit point (location) to describe even a highly improbable case of a complete cure. Therefore, we add this kind of location (*stop_positive*) together with an additional transitions marked with dashed lines in order to make our exit location reachable from the *chronic_p* and *repetitive_anemia*. The locations *lack_loss_resp_ch* and *lack_loss_resp_bl* play the roles of negative exit points associated with the lack or loss of response, meaning that the protocol should be changed or stopped. These locations should be reachable from any location of the model. Therefore, we have added a transition from *repetitive_anemia* to *lack_loss_resp_ch* in the GL chronic phase treatment, e.g. an alternative treatment should be chosen.

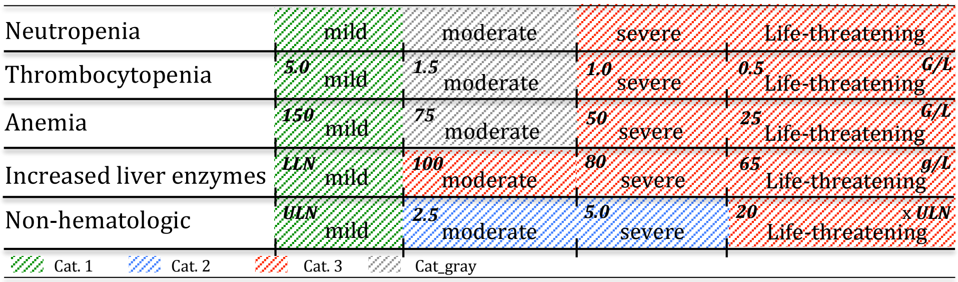
The *incompleteness* problem in the dose adjustment protocol exists in both the chronic and the acceleration phases, where once we go to the *repetitive_anemia* or *anemia_2* locations there is either no outgoing transition, as in the case of the *repetitive_anemia* location of the chronic phase or we enter a closed loop (*anemia_2* ↻ *pause*) in case of the accelerated phase or blast crisis. For instance, in the case of the chronic phase this problem can be found by verifying whether the *stop_positive* or *lack_loss_resp_ch* locations are always reachable from *chronic_p* location: *chronic_p*–>E<> (*stop_positive* or *lack_loss_resp_ch*) and finding the counterexample leading to the *repetitive_anemia* deadlock location. In order to avoid these problems we must modify the model such that there are no deadlock locations other than the exit points. The exit point should be reachable from every location of the model. To this end, in the chronic phase part of our case study, we add a transition from the *repetitive_anemia* to the *lack_loss_resp_ch* location.

4.3. *Adverse events classification*

Various clinical studies show that responsiveness to the treatment with a drug depends on the concentration of the drug in patient's blood, which depends on patients features, drug dose and intake interval. Pharmacokinetics (PK) is a branch of pharmacology focused on studying the drug disposition in the human body. Pharmacodynamics (PD) is the study of the biochemical and physiological effects of drugs on the body. Therapeutic Drug Monitoring (TDM)[23] is the approach that unifies the PK-PD knowledge. The concentration fo the drug in the patient's blood may be closely related to the drug its effect (PK-PD relationship). Drugs with clear PK-PD relationships and a narrow therapeutic ranges may be easily under- or overdosed. If the drug concentration in patients blood lays within the therapeutic range, the adverse events (AEs) or non-responsiveness to the treatment with high probability happen due to possible resistance to the drug (in case of lack of response) and other reasons not related to the treatment with *imatinib*.

Let us discuss the *chronic* phase of CML marked with a rectangle in Figure 3. We can see that the initial dose of *imatinib* can be changed during the treatment, either increased or decreased. It will be increased at the transition $chronic\_p \hookrightarrow LoL\_resp$, which corresponds to a suboptimal response to the treatment. The treatment can be stopped for some time and the dose with which the treatment is further restarted may also be decreased when the level of neutrophils goes below 1.0×10^9/l ($N\_N <= n\_lowerB$) and/or level of platelets goes below 50×10^9/l ($N\_T <= t\_lowerB$)), which represent AEs during the treatment.

Assuming that the drug is chosen correctly, AEs or non-responsiveness to the treatment can happen only due to the suboptimal dose chosen for a patient with specific features, which would correspond to over- and under-dosing the patient respectively. In case of over-dosing a patient several AEs can be observed. We can classify them as quantitative (laboratory) and non-quantitative (non-laboratory). Among the quantitative or hematologic parameters of AEs classification we can name the levels of neutrophils, platelets (thrombocytes), erythrocytes and liver enzymes. The list of possible non-quantitative adverse events is larger and contains headache, dizziness, nausea, vomiting, dyspepsia, diarrhea, abdominal cramps, skin rash, pruritus, muscle cramps/pain/weakness, edema, fatigue and insomnia. The protocol presented above takes into account only two laboratory AEs: low levels of neutrophils and platelets. In this section we augment the protocol model with the extended list of AEs.



Neutropenia	🟩 mild	▨ moderate	🟥 severe	🟥 Life-threatening
Thrombocytopenia	🟩 5.0 mild	▨ 1.5 moderate	🟥 1.0 severe	🟥 0.5 Life-threatening G/L
Anemia	🟩 150 mild	▨ 75 moderate	🟥 50 severe	🟥 25 Life-threatening G/L
Increased liver enzymes	🟩 LLN mild	🟥 100 moderate	🟥 80 severe	🟥 65 Life-threatening g/L
Non-hematologic	🟩 ULN mild	🟦 2.5 moderate	🟦 5.0 severe	🟥 20 Life-threatening x ULN
🟩 Cat. 1	🟦 Cat. 2	🟥 Cat. 3	▨ Cat_gray	

Fig. 4. Adverse events classification table (according to: Common Terminology Criteria for Adverse Events (CTCAE) version 4.0).[7]

Let us first classify the AEs. Each event can be evaluated as mild, moderate, severe and life-threatening. With laboratory AEs we can associate *neutropenia*, *thrombocytopenia*, *anemia* and *increase of liver enzymes*. Since non-laboratory AEs are not quantitative we do not distinguish among them. Figure 4 presents a table that has four rows corresponding to quantitative (hematologic) and one row that unifies all non-quantitative (non-hematologic) parameters. Numbers under row separating lines correspond to the reference classification values of the rows above them. LLN and ULN stand for Lower Limit Normal and Upper Limit Normal respectively.

The table classifies evaluation of severity of AEs into four categories. For example, the first category (Cat.1) corresponds to the case when all AEs are classified as *mild*. Category 3 (Cat. 3) is the most severe one, when the levels of neutrophils, platelets and erythrocytes are at least severely low, while the level of liver enzymes is at least moderately high. The levels of neutrophils and platelets at which the drug administration on Figure 3 should be suspended belong to the category 3. The drug administration is then restarted only when the neutrophils and platelets levels raise up to the mild level of severity (Cat.1).

This way, the protocol model of Figure 3 can be modified using the AEs classification. In particular the guards of levels of neutrophils and platelets at the transition from *chronic_p* to *anemia_ch* can be replaced with an abstract guard checking the correspondence to category 3 ($Cat3 == true$). As well as guards of transitions from *anemia_ch* to *repetitive_anemia* or back to *chronic_p* can be replaced with $Cat1 == true$ guard, as shown on the figure. This way, the categories 1 and 3 are those that can influence the state transitions of the protocol model. When implementing the model the abstract guard can be either explicitly replaced with a set of five guards, one for each evaluated AEs or computed by an additional TAT model, shown in Figure 5, composed in parallel with the model of Figure 3.

4.4. *Response definitions*

Table 1 summarises the response definition to the treatment with *imatinib* as it was given in Ref. 11. In this section we present our approach to the data interpretation

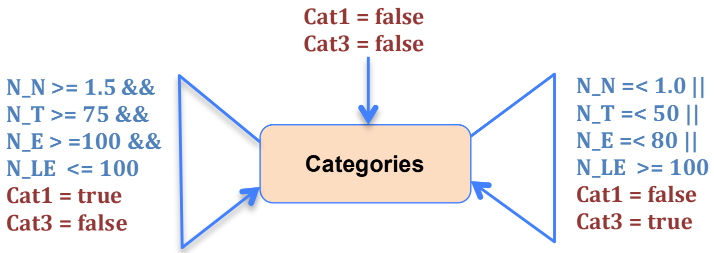


Fig. 5. Adverse events classification in categories model.

of this table. We need to be able to build an action-based model of patient's state evaluation that can be complemented with the model of Figure 3.

The response to *imatinib* treatment is measured based on hematological, cytogenetic and molecular tests. The hematological test measures the levels of various White Blood Cells (WBL). The cytogenetic test measures the percentage of Ph+ chromosome mutations, that is a specific chromosomal abnormality associated with CML. Molecular test measures the level of the specific BCR-ABL transcripts of the Ph+ chromosome. Based on the results of tests performed in specific time intervals (3, 6, 12 and 18 months) the response to a treatment can be classified as optimal, suboptimal and failure.

Optimal response means that based on current knowledge, the patient is projected to have a normal survival. *Failure* here means that the patient still may do well also for years, but will never become an optimal responder. Thus, a change of treatment should be considered. *Suboptimal* response is a grey area between failure and optimal response. The condition of a suboptimal responder is transitory by nature, so it is not yet clear if a suboptimal responder would benefit more from an early change of therapy or from the continuation of the same treatment.

In Figure 6 we have divided the initial Table 1 into three identical tables each corresponding to a different level of response: *optimal*, *suboptimal* and *failure*. Each table has three rows corresponding to a separate type of response tests: hematologic, cytogenetic and molecular. Even though the three tests should be evaluated in parallel, after analyzing the classification of response levels we can see that:

- The hematological response is expected to be observed first and is classified either as a Complete Hematologic Response (CHR) or as no CHR (NoCHR);
- The cytogenetic response has five gradations that are observed in projection to already achieved CHR;
- Similarly, the molecular response is evaluated after a Complete Cytogenetic Response (CCgR) and, consequently, CHR are achieved.

The three rows of the tables are shifted with respect to each other summarizing the above. From this representation we can see that the three mentioned types of response can be aggregated into one accumulated response achievement. Thus, each

Table 1. Table with the definition of the response to the treatment with IM.[11]

	Warnings	Failure	Suboptimal Response	Optimal Response
BASELINE	High risk CCA/Ph+	/	/	/
3 months	/	Non CHR	No CgR (Ph+ $>95\%$)	At least minor CgR (Ph+ $\leq 65\%$)
6 months	/	No CgR(Ph+ $>95\%$)	Less than PCgR (Ph+ $>35\%$)	At least PCgR (Ph+ $\leq 35\%$)
12 months	Less than MMolR	Less than PCgR (Ph+ $>35\%$)	PCgR (Ph+ 1–35%)	CCgR
18 months	/	Less than CCgR	Less than MMolR	MMolR
Any Time, during treatment	Rise in transcript levels CCA/Ph-	Loss of CHR Loss of CCgR Mutations CCA/Ph+	Loss of MMolR Mutations	Stable or improving MMolR

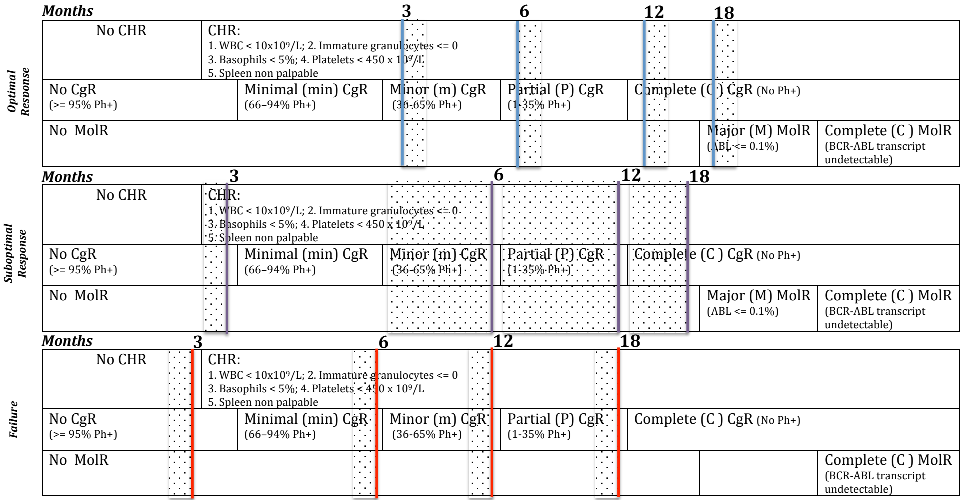

Optimal Response
Months
3
6
12
18
No CHR
CHR:
1. WBC < 10x10⁹/L; 2. Immature granulocytes <= 0
3. Basophils < 5%; 4. Platelets < 450 x 10⁹/L
5. Spleen non palpable
No CgR (>= 95% Ph+)
Minimal (min) CgR (66–94% Ph+)
Minor (m) CgR (36-65% Ph+)
Partial (P) CgR (1-35% Ph+)
Complete (C) CgR (No Ph+)
No MolR
Major (M) MolR (ABL <= 0.1%)
Complete (C) MolR (BCR-ABL transcript undetectable)
Suboptimal Response
Months
3
6
12
18
No CHR
CHR:
1. WBC < 10x10⁹/L; 2. Immature granulocytes <= 0
3. Basophils < 5%; 4. Platelets < 450 x 10⁹/L
5. Spleen non palpable
No CgR (>= 95% Ph+)
Minimal (min) CgR (66–94% Ph+)
Minor (m) CgR (36-65% Ph+)
Partial (P) CgR (1-35% Ph+)
Complete (C) CgR (No Ph+)
No MolR
Major (M) MolR (ABL <= 0.1%)
Complete (C) MolR (BCR-ABL transcript undetectable)
Failure
Months
3
6
12
18
No CHR
CHR:
1. WBC < 10x10⁹/L; 2. Immature granulocytes <= 0
3. Basophils < 5%; 4. Platelets < 450 x 10⁹/L
5. Spleen non palpable
No CgR (>= 95% Ph+)
Minimal (min) CgR (66–94% Ph+)
Minor (m) CgR (36-65% Ph+)
Partial (P) CgR (1-35% Ph+)
Complete (C) CgR (No Ph+)
No MolR
Complete (C) MolR (BCR-ABL transcript undetectable)

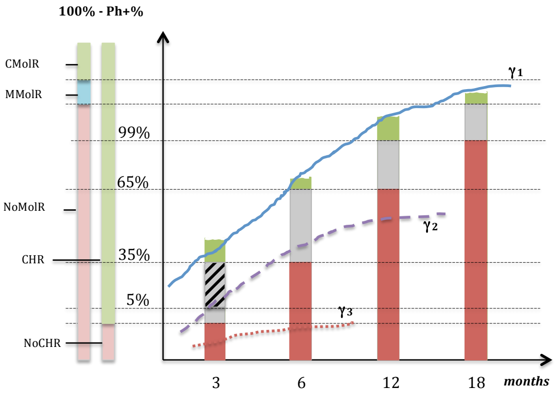
Fig. 6. Gradual graphical representation of the response data.

table is crossed by four vertical lines corresponding to four response thresholds at particular milestones (3, 6, 12 and 18 months from left to right respectively) that need to be considered during the treatment.

From the quantitative point of view, if the value of accumulated response achievement is above the optimal threshold the result will still be evaluated as optimal. Therefore, the values of the dotted regions covering the area on the right side of the thresholds of the "optimal response" (upper table), are still considered as the optimal response. On the other hand, if the accumulated response value is below the thresholds of the "failure response" it is still considered to be a *failure*. The dotted regions on the left side of the the failure thresholds represent the failure values at coresponding milestones.

The region of the values of a suboptimal response at a particular milestone is defined as the area between the "failure" and "optimal" response thresholds. From Figure 6 we can see that part of the suboptimal accumulated response belonging to the first milestone (at 3 months) as described in Table 1 is not complete. This represents the *incompleteness* problem of the protocol description. However, taking into account that the suboptimal response is by definition the response between *optimal* and *failure* we can conclude that the definition of the suboptimal response is formally *redundant*. Furthermore, this *redundancy* introduces an additional information *incompleteness*.

The attempt to formalize the data of Table 1 has allowed us to fix an incompleteness problem inclosed into a redundant part of the data. However, this representation still cannot be interpreted by a machine. The response levels presented in Figure 6 can be further transformed into a graph as depicted in Figure 7. The right part of Figure 7 lists the definitions of different levels of hematological, cytogenetic



Complete Hematologic Response (CHR)	1. WBC < 10x10⁹/L; 2. Immature granulocytes <= 0 3. Basophils < 5%; 4. Platelets < 450 x 10⁹/L 5. Spleen non palpable
Complete Cytogenetic Response (CCgR)	No Ph+ metaphases
Partial Cytogenetic Response (PCgR)	1-35% Ph+ metaphases (65-99% in 100% - Ph+%)
Minor Cytogenetic Response (mCgR)	36-65% Ph+ metaphases (35-64% in 100% - Ph+%)
Minimal Cytogenetic Response (minCgR)	66-94% Ph+ metaphases (5-35% in 100% - Ph+%)
No Cytogenetic Response (NoCgR)	>=95% Ph+ metaphases (<=5% in 100% - Ph+%)
Major Molecular Response (MMolR)	BCL-ABL: ABL <= 0.1% on the Internal Scale
Complete Molecular Response (CMolR)	BCR-ABL transcript undetectable

Fig. 7. (Color online) Response definitions to the treatment with *imatinib*.

and molecular responses in a quantitative way, which are summarized into two bars on the left side of the graph. The graph shows four sets of bars for each defined test milestone (3, 6, 12 and 18 months). The highest (green) bar corresponds to the minimum level of the *optimal* response, the lowest (red) bar represents failure. The middle (gray) bar is associated with a suboptimal response. The gray area with stripes, that is present only in the first set, represents the area that was not described in Table 1, incompleteness of the information.

The three cumulative curves γ_1, γ_2 and γ_3 in the graph represent three different scenarios of progressive patient reaction to the treatment. The first curve (γ_1) corresponds to a situation when an optimal response is achieved. The second curve (γ_2) represents a failure of the treatment after one year. The third curve (γ_3) represents the lack of response to the treatment within 6 months. Based on the response definition we can build two observer TAT that would control the failure and/or suboptimal level of the accumulated response achievements. The least restrictive observer would be the one built to guard the level of failure bars. This observer will ensure that the progressive patient reaction to the treatment will always remain at least above the failure level, at the level of *suboptimal* response and higher.

The observer TAT controlling the failur level is depicted in Figure 8 and represent a complementary model to the one presented in Section 4.1. The two models synchronize by means of shared variable *LoL_resp*. The periods $p1$ and $p2$ of the model presented in Figure 8 are assigned to 3 and 6 months respectively. We have only two periods since time will also be accumulated, accounted always with respect to the time when the treatment was initiated.

The execution of the model starts from the initial location *INIT* in which the model will stay for 3 months after the treatment has been started. Afterwards it will either transit to the *FAILURE* location and thus set the shared Boolean variable *LoL_resp* to **true** or, if the response level is at least at the border level of *failure*, go to the next location in which it will stay for another 3 months (6 months in total

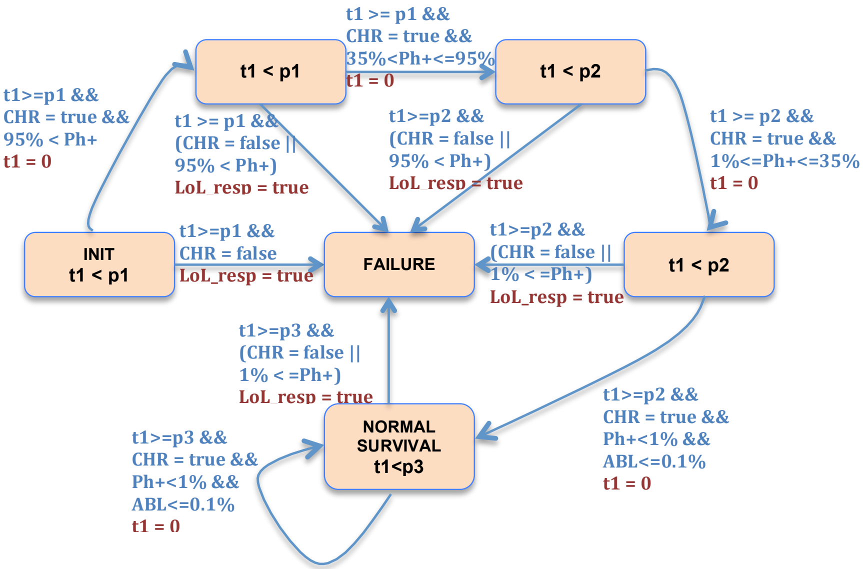


Fig. 8. Observer TA.

from the beginning of treatment). Thus, the model may proceed to the *NORMAL SURVIVAL* location.

The definition of responses presented in Ref. 11 does not say anything about the medical procedures after 18 months. This represents another *incompleteness* problem. Therefore, we add, for demonstration purpose, a self-loop to the location for the case when response stays at the same level and a transition to the *FAILURE* location, in case of loss of response. These structural problems are verified automatically. However, for small examples above they are still easily detectable by inspecting the graph. The input data to the model, such as the amount of Ph+ chromosomes or the BCR-ABL level should come from the measurement TAT similar to the one in Section 4.1.

5. "Rescue" Therapeutic Drug Monitoring

As was discribed in Section 4.3, the response to the treatment with various drugs is correlated with the drug concentration in the blood. The intervention of TDM for dose individualization can help keeping the drug concentration within the limited ranges provided by the PK-PD studies.[19] The real TDM measurement procedure is quite slow (takes about a day), expensive and requires an invasive measurement of individual drug concentration. When treating CML, TDM is performed only in case of adverse events or suboptimal response in order to figure out whether it is related to over- or under-dosing or not. Such TDM interventions can be called "rescue TDM" for tolerance (overdosing) and efficacy (under-dosing) concerns. In this section we present the modification of the chronic part of the protocol model

Table 2. Main abbreviations.

TDM	Therapeutic Drug Monitoring	**SR**	Suboptimal response
TDMT	TDM for “tolerance” concerns	**AE**	Adverse event
TDME	TDM for “efficacy” concerns	**CDSR**	Concentration dependent SR
N_fails	Number of AE of cat3 cases	**CISR**	Concentration independent SR
$\mathbf{C_{min}}$	Blood drug concentration	**D_Dint**	Drug-drug interaction
CDAE	Concentration dependent AE	**NC**	Non compliance
CIAE	Concentration independent AE	**AP**	Absorption problem

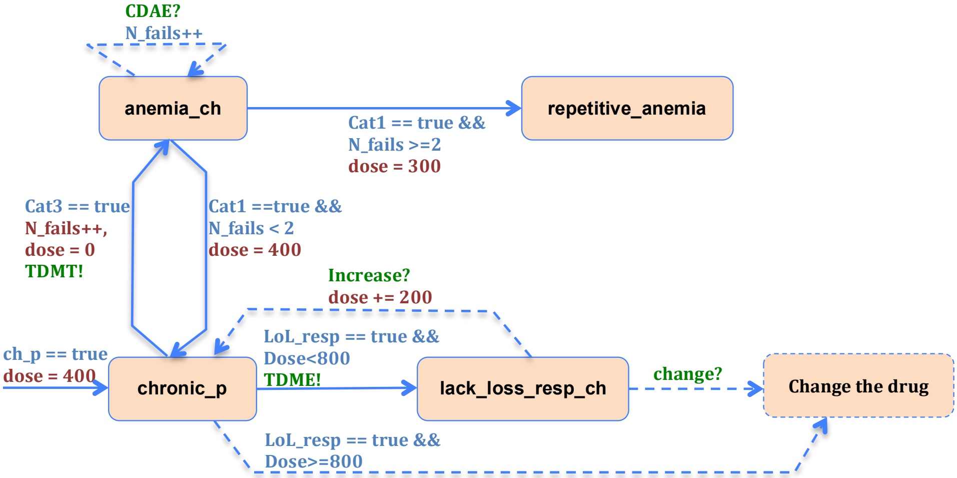


Fig. 9. Drug dose adjustment in the chronic phase adapted for the selective “rescue TDM” for efficacy and tolerance concerns.

presented earlier on Figure 3 synchronized with an additional TDM models evaluating the drug concentration level in blood. The main abbreviations used in further presented models are summarized in Table 2.

Figure 9 depicts a modified model of drug dose adjustment in the chronic phase adapted for the selective “rescue TDM” for efficacy and tolerance concerns. Transitions represented by solid line arrows are the original transitions of the initial model. Dashed arrows represent transitions added to conform with selective “rescue TDM” interventions.

Figure 10 presents two complementary models evaluating the level of plasma drug concentration. These models are used to decide whether the AE is concentration dependent or has some other, non-hematologic reasons. The model on the left side of Figure 10 represents the decision with tolerance concerns, meaning that the plasma drug concentration is expected to be higher than the upper bound of the therapeutic range. The transitions of this model will be triggered by the transition from *chronic_p* to *anemia_ch* location of *imatinib* model (see Figure 9) by means of TDMT synchronization channel. We assume that the therapeutic range for *imatinib*

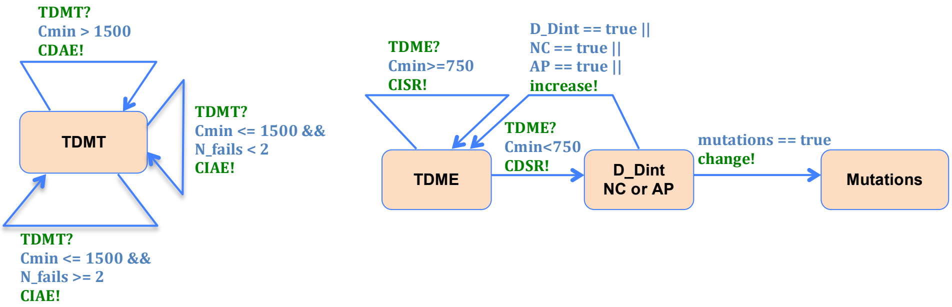


Fig. 10. Supportive models for selective "rescue TDM".

is [750–1500] μg/l.[19] Therefore, this model is checking whether the concentration value is below or above the upper bound of the therapeutic range (1500 μg/l). If the drug concentration is greater than 1500 μg/l, the CDAE! synchronization channel will be activated, which will trigger the dashed self-loop transition of the *anemia_ch* location. Upon this transition the N_fails value will be increased, which will force the further restart of the treatment with a reduced dose (300 mg) even after the first anemia case, since it was confirmed by TDM that the reason of the AE is the high drug concentration in blood. If TDM shows that the AE is not related to the drug concentration, the CIAE! synchronization channel will be activated. However, it will not produce any changes to the drug adjustment model.

The model on the right part of Figure 10 is checking whether the concentration value is below or above the lower bound of the therapeutic range 750 μg/l, efficacy concerns. If the concentration is lower then 750 μg/l it can be considered that we are facing a concentration dependent suboptimal response (CDSR). In this case it is important to figure out whether it is due to the drug-drug interaction (D_Dint), non compliance (NC), absorption problems (AP) or resistance to the drug ($mutations$). If there is evidence of mutations, the change of the drug should be considered. Otherwise, the dose may be increased up to maximum of 800 mg with 200 mg step after each response evaluation with a suboptimal result. If the level of the accumulated response does not change when the dose is already set to the maximum of 800 mg, the drug must be changed.

6. Conclusions

In this paper we have applied a methodology of the area of embedded systems design to a computer-based interpretation of medical protocols. The methodology exploits the Timed Automata extended with Tasks (TAT) model widely used for the real-time systems analysis. Using TAT for modeling medical GLs as well as a control flow is not a unique approach. However, TAT has several advantages. First of all it has a notion of time and allows modeling of choices of sequential actions based on specific conditions mapped to the time scale. It enables modeling of several plans with periodically repeating actions that can be executed in parallel with other actions. It can associate tasks with locations to model simple actions (e.g., computational blocks or medical tests) that need to be finished before a certain deadline. When the model is executed a task will be added to the scheduling queue whenever an associated location is reached. The tasks will be then executed in a scheduler defined order. We have presented the anticancer drug *imatinib* dose adjustment protocol case study complimented with a model of the response level control based on the levels of accumulated achievements. We also present the extension of the protocol model with the "rescue" TDM approach for *efficacy* and *tolerance* concerns that enhance the model without contradicting official protocol for patients treatment with *imatinib*. We where able to fix some incompleteness problems in these models.

Acknowledgment

This work was supported by ISyPeM project of the Swiss Nano-Tera initiative, evaluated by the Swiss National Science Foundation. The authors would like to thank Prof. T. Buclin from CHUV Hospital of Lausanne for the discussion regarding the project and Dr. Simon Bliudze from Rigorous System Design (RiSD) laboratory of EPFL for valuable comments regarding formal modeling.

References

1. SPIN. http://spinroot.com/spin/whatispin.html.
2. SMV. http://www.cs.cmu.edu/~modelcheck/smv.html.
3. TIMES. http://www.timestool.com/l.
4. AGREE instrument. http://www.agreetrust.org/.
5. AsbruView. http://www.openclinical.org/dld_asbruview.html.
6. Imatinib. http://www.ema.europa.eu/docs/en_GB/document_library.
7. http://evs.nci.nih.gov/ftp1/CTCAE/CTCAE_4.03_2010-06-14_QuickReference_5x7.pdf.
8. R. Alur and D. L. Dill, A theory of timed automata, *Theoretical Computer Science* **126**(2) (1994) 183–235.
9. T. Amnell, E. Fersman, P. Pettersson, H. Sun, and W. Yi, Code synthesis for timed automata, *Nord. J. Comput.* **9**(4) (2002) 269–300.
10. D. Arney, M. Pajic, J. M. Goldman, I. Lee, R. Mangharam, and O. Sokolsky, Toward patient safety in closed-loop medical device systems, in *Proc. of the 1st ACM/IEEE Int. Conf. on Cyber-Physical Systems* (*ICCPS '10*) (2010), pp. 139–148.
11. M. Baccarani, F. Castagnetti, G. Gugliotta, F. Palandri, and S. Soverini, Response definitions and european leukemianet management recommendations, *Best Pract Res Clin Haematol* **22**(3) (2009) 331–341.
12. S. Bäumler, M. Balser, A. Dunets, W. Reif, and J. Schmitt, A verification of medical guidelines by model checking — A case study, in *Proc. Model Checking Software SPIN* (2006).
13. G. Behrmann, R. David, and K. G. Larsen, A tutorial on Uppaal, in *Formal Methods for the Design of Real-Time Systems: 4th Intl. School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM-RT 2004* (Springer, 2004), pp. 200–236.

14. A. Bottrighi, L. Giordano, G. Molino, S. Montani, P. Terenziani, and M. Torchio, Adopting model checking techniques for clinical guidelines verification, *Artif. Intell. Med.* **48** (January 2010) 1–19.
15. A. Boxwala, M. Peleg, S. Tu, O. Ogunyemi, Q. Zeng, D. Wang, V. Patel, R. Greenes, and E. Shortliffe, GLIF3: A representation format for sharable computer-interpretable clinical practice guidelines, *Journal of Biomedical Informatics* **37**(3) (June 2004) 147–161.
16. P. Ciccarese, E. Caffi, L. Boiocchi, S. Quaglini, and M. Stefanelli, A guideline management system, *Stud Health Technol Inform* **107**(Pt 1) (2004) 28–32.
17. G. Duftschmid, S. Miksch, Y. Shahar, and P. Johnson, Multi-level verification of clinical protocols, in *Proc. of the Workshop on Validation Verification of Knowledge-Based Systems* (*VVÕ98), in conjunction with the Sixth Int. Conf. on Principles of Knowledge Representation and Reasoning (KRÕ98* (1998), pp. 4–1.
18. M. Fieschi, P. Votruba, S. Miksch, and R. Kosara, Facilitating knowledge maintenance of clinical guidelines and protocols, (2004).
19. V. Gotta, N. Widmer, L. Decosterd, C. Csajka, M. Duchosal, Y. Chalandon, D. Heim, M. Gregor, and T. Buclin, Therapeutic drug monitoring (TDM) of imatinib: Effectiveness of bayesian dose adjustment for CML patients enrolled in the imatinib concentration monitoring (I-COME) study. Swiss Medical Forum (2010), p. 285.
20. P. D. Johnson, S. Tu, N. Booth, B. Sugden, and I. N. Purves, Using scenarios in chronic disease management guidelines for primary care, in *Proc. of American Medical Informatics Association* (*AMIA Symposium*) (2000), pp. 389–393.
21. B. Kim, A. Ayoub, O. Sokolsky, I. Lee, P. Jones, Y. Zhang, and R. Jetley, Safety-assured development of the gpca infusion pump software, in *Proc. of the Ninth ACM Int. Conf. on Embedded Software* (*EMSOFT '11*) (ACM, New York, NY, USA, 2011), pp. 155–164.
22. O. Kupferman and M. Y. Vardi, Model checking revisited, in *Proc. of the 9th Int. Conf. on Computer Aided Verification* (*CAV '97*) (Springer-Verlag, London, UK, 1997), pp. 36–47.
23. J. H. Martin, R. Norris, M. Barras, J. Roberts, R. Morris, M. Doogue, and G. R. D. Jones, Therapeutic monitoring of vancomycin in adult patients: A consensus review of the American society of health-system pharmacists, the infectious diseases society of america, and the society of infectious diseases pharmacists, *American journal of health system pharmacy AJHP official journal of the American Society of HealthSystem-Pharmacists* **31**(1) (2010) 21–24.
24. S. Miksch, Y. Shahar, and P. Johnson, Asbru: A task-specific, intention-based, and time-oriented language for representing skeletal plans, in (Open University, UK, 1997), pp. 9–1.
25. B. Pérez and I. Porres, Authoring and verification of clinical guidelines: A model driven approach, *J. of Biomedical Informatics* **43** (August 2010) 520–536.
26. I. Porres, E. Domfnguez, B. Perez, A. Rodriguez, and M. Zapata, A model driven approach to automate the implementation of clinical guidelines in decision support systems, in *15th Annual IEEE Int. Conf. and Workshop on the Engineering of Computer Based Systems, 2008* (*ECBS 2008*) (31 March 2008–4 April 2008), pp. 210–218.
27. R. N. Shiffman, G. Michel, A. Essaihi, and E. Thornquist, Bridging the guideline implementation gap: A systematic, document-centered approach to guideline implementation, *J. Am. Med. Inform. Assoc.* **11**(5) (September–October 2004) 418–426.
28. A. Simalatsar and G. De Micheli, Tat-based formal representation of medical guidelines: Imatinib case-study, in *Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society* (*EMBC 2012*) (August 28–September 1, 2012), pp. 5078–5081.

29. D. R. Sutton, P. Taylor, and K. Earle, Evaluation of PROforma as a language for implementing medical guidelines in a practical context, *BMC Medical Informatics and Decision Making* **6** (April 2006) 20.
30. A. Teije, M. Marcos, M. Balser, J. van Croonenborg, C. Duelli, F. van Harmelen, P. Lucas, S. Miksch, W. Reif, K. Rosenbrand, and A. Seyfang, Improving medical protocols by formal methods, *Artif. Intell. Med.* **36** (March 2006) 193–209.
31. P. Terenziani, S. Montani, A. Bottrighi, M. Torchio, G. Molino, and G. Correndo, The glare approach to clinical guidelines: Main features, *Studies in Health Technology and Informatics* **101** (2004).
32. S. W. Tu, J. R. Campbell, J. Glasgow, M. A. Nyman, R. McClure, J. McClay, C. Parker, K. M. Hrabak, D. Berg, T. Weida, J. G. Mansfield, M. A. Musen, and R. M. Abarbanel, The SAGE Guideline Model: Achievements and Overview, *Journal of the American Medical Informatics Association* **14**(5) (June 2007) 589–598.
33. S. W. Tu and M. A. Musen, Modeling data and knowledge in the EON guideline architecture, *Medinfo* **10**(1) (2001) 280–284.
34. P. Votruba, S. Miksch, A. Seyfang, and R. Kosara, R.: Tracing the formalization steps of textual guidelines, in *Computer-based Support for Clinical Guidelines and Protocols* (Press, 2004) 172–176.