Proceedings of the 2012 IEEE 12th International Conference on Bioinformatics
& Bioengineering (BIBE), Larnaca, Cyprus, 11-13 November 2012

Medical Guidelines Reconciling Medical Software and Electronic Devices: Imatinib Case-study

Alena Simalatsar
École Polytechnique Fédérale de Lausanne, Switzerland
Email: alena.simalatsar@epfl.ch

Giovanni De Micheli
École Polytechnique Fédérale de Lausanne, Switzerland
Email: giovanni.demicheli@epfl.ch

***Abstract*—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 that includes both software and electronic devices. However, often medical GLs suffer from structural problems, such as incompleteness, inconsistencies, ambiguity and redundancy. Formal representation of GLs would enable the validation of GLs structural, real-time and life-cycle properties. 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 a novel approach based on Timed Automata extended with Tasks (TAT) for the medical protocol formal representation using the TIMES toolbox. We discuss the verification issues with the help of the Imatinib case study.**

***Index Terms*—Medical Guidelines, Formalization, Timed Automaton, Verification.**

I. 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 [18]. 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 behaviors 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 act as an intersection point between medical software and electronic devices, playing a role of a medical system specification.

Until now, most of the GLs are represented in a textual format and therefore often suffer from such structural problems as incompleteness, inconsistency, ambiguity and redundancy. Therefore, it is essential to find a proper formal representation for the GLs that would enable the validation of the GLs structural and life-cycle properties. Several frameworks for the computer-based interpretation of GLs have been presented in the past three decades [11], [17], [24], [26], [27]. These tools adapt flow-charts as a core formalism to represent a sequence of actions. However, they enable validation of the protocol structure only by means of their formal representation and have no support for the automatic verification of the protocol formal properties. Such frameworks as GLARE [25] and Asbru [14], [21] 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 novel approach of [23] for the medical GLs formal representation can bridge the gap between medical software and electronic devices. The approach is based on GLs formal representation with Timed Automaton extended with Tasks (TAT) [7] and exploits the TIMES toolbox [3] aimed to support the modeling and verification of real-time systems. TIMES includes not only a Graphical User Interface (GUI) for system modeling but also a model-checker engine that supports system verification. Therefore, no additional translation path and tracing back the result is needed when performing the model verification. We present a case study of modeling the Imatinib dose adjustment part of the protocol for adult patients with newly diagnosed Philadelphia positive (Ph+) Chronic Myeloid Leukemia (CML) complemented. The model is complemented with the TAT-based model of response definition to the treatment. We perform the validation of the protocol structure as well as discuss issues of the protocol real-time and life-cycle properties verification.

The paper is organized as follows. Section II gives an overview of the existing formalisms. In Section III 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 also discusse the advantages and disadvantages of TAT-based approach Section IV presents the Imatinib case study and discusses the verification issues. Section V concludes the paper.

978-1-4673-4358-9/12/$31.00 ©2012 IEEE

II. Related Work

In [18] 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 behavior is then verify 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. In this paper we use the TA modeling for a more general purpose to formally represent the step by step procedures of the medical treatment described in medical GLs. The GLs can be viewed as a medical system behavior specification, where actions are distributed between medical doctors, server machines and embedded medical devices.

A number of specific languages and tools [4], [11], [12], [14], [16], [17], [21], [22], [24]–[28] aimed to perform formalization of medical guidelines (GL) has been developed in the past decades. PRODIGY [17] 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*. PRODIGY did not allow the description of alternative scenarios or parallel actions. Its development has been already discontinued, however, it has created a fruitful base for other knowledge-based decision support tools. EON [27] is a component-based suite for GL 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. The Guideline Interchange Format (GLIF) [11] was developed to support guideline modeling as a flow-chart. 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. GUIDE [12] is focused on providing an integrated medical knowledge management through a unique central system and is composed 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. However, it is more focused on data centralization and distribution and thus goes further with formalizing the data structure, using GEM [22], 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. Non of the mentioned above tools support GLs formal verification.

Similar to other formalisms, GLARE [25] 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. 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 [14], [21] is called a *plan*. 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. From the semantical point of view 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. 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 [9]. This makes it difficult to trace back the results of properties verification and adjust the initial model if necessary. Moreover, non of the existing formalisms for GLs modeling provide real-time system analysis and platform oriented code synthesis tools as opposite to TIMES tool [7].

III. Formalization Approach

The control flow of the electronic devices as well as of the complementary decision-support system should be based on the parts of the same GL that act as an intersection point between medical software and electronic devices. We formalize a stage of patient medical treatment when the diagnosis has been already performed. This way some parts of the treatment protocol can be delegated to an electronic device.

In order to bridge the gap between medical software and electronic devices we have to make these two worlds speak the same language. Moreover, the language of the framework chosen for formal representation should be simple enough to be understood 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 [23], since TIMES tool includes 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 [7] thus enabling further embedded or decision-support system code synthesis step.

a) Timed Automaton extended with Tasks: *Timed Automaton* (TA) [6] 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 [10], a precursor of TIMES tool [3], implements TA extended with vac ljreriables. 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 [7] 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.

b) Modeling with TAT: An example of modeling a step-by-step medical action with TAT is presented in Section IV-A. 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 guideline 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-base GL 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 and invariants 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.

GLs actions and effects are often not instantaneous. Actions may have some duration, which can be modeled as the worst-case execution time of a task; while effects may be delayed, which we represent as a deadline and levels of treatment effectiveness (see Section IV-B) TAT introduces the possibility to mark transitions with probabilities that may represent the level of evidence or experience with respect to one choice or another. In other words, the TAT-based approach is *expressive* enough to represent the step-by-step actions of medical protocols as well as treatment response definition.

IV. Imatinib Case Study

Imatinib, marketed by 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. A complete drug administration protocol of Imatinib can be found here [5]. It includes over 50 pages.

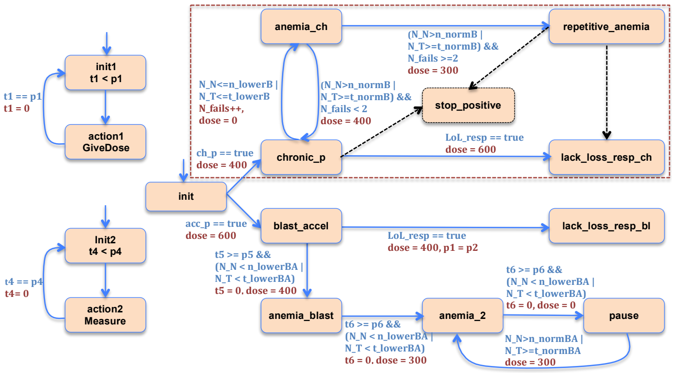

Figure 1. Drug delivery, measurements and Imatinib dose adjustment models

A. *Dose Adjustment and Delivery Models*

The dose should be administered once a day (*p1 = 1 day*). The model is a network of three cooperating TATs. The first TAT of the network consists of two locations (*init1* and *action1*) and is responsible for the periodic drug delivery. Thus every period *p1* (*t1 == p1*), TAT is transitioning to the *action1* location. On this transition the clock *t1* is reset to 0 and the *GiveDose* task is activated (added to the scheduling queue). The transition from the *action1* to the *init1* location is then taken. This model would either give a periodic reminder to a medical doctor to give a dose to a patient or send a command to a drug delivery device if this process is fully automated. The second TAT model composed of two locations represents a periodic action of performing medical 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 *N_N* and *N_T* variables of the third, Imatinib dose adjustment, model that is described below. The original model is composed of the elements with solid lines only.

The recommended dose of Imatinib is 400 mg/day for patients in the chronic phase (transition ($\hookrightarrow$) from *Init* to *chronic_p*) of CML and 600 mg/day for patients in the *accelerated* phase (*Init* $\hookrightarrow$ *blast_accel*) of CML. 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 mg to 600 mg in patients with the *chronic* phase of the disease (*chronic_p* $\hookrightarrow$ *LoL_resp*) or from 600 mg to a maximum of 800 mg given as 400 mg twice daily (*blast_accel* $\hookrightarrow$ *LoL_resp* and *p1* = *p2*, where *p2* = *half day*) in patients with accelerated phase or blast crisis in case of: *(i)* disease progression at any time; *(ii)* failure to achieve a satisfactory hematological or cytogenetic response; or *(iii)* loss of a previously achieved hematological and/or cytogenetic response. The complementary TAT model of the response definition [8] is presented in Section IV-B.

In the *chronic* phase of CML, marked with a rectangle in Figure 1, if the level of neutrophils (ANC) goes below 1.0 x 10^9/l (*N_N<=n_lowerB*) and/or level of platelets goes below 50 x 10^9/l (*N_T<=t_lowerB*):

1) Stop Imatinib until ANC >1.5 x 10^9/l and platelets >75 x 10^9/l (*chronic_p* $\hookrightarrow$ *anemia_ch*);
2) Resume treatment with Imatinib at previous dose, (*anemia_ch* $\hookrightarrow$ *chronic_p*, *N_fails* accounts to the number of anemia occurrences);
3) In the event of recurrence of ANC <1.0 x 10^9/l and/or platelets <50 x 10^9/l, repeat step 1 and resume Imatinib at reduced dose of 300 mg (*anemia_ch* $\hookrightarrow$ *repetitive_anemia*);

The 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 1.

B. Response Definitions

The response to Imatinib treatment [8] 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 value of Ph+ chromosome mutations, that is a specific chromosomal abnormality associated with CML. Molecular test measures the level of a specific BCR-ABL transcripts of the Philadelphia 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.

Complete Hematologic Response (CHR)	1. WBC < 10x10^9/L; 2. Immature granulocytes <= 0 3. Basophils < 5%; 4. Platelets < 450 x 10^9/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

Figure 2. Response definitions to the treatment with Imatinib

Figure 2 lists the definitions of three different levels of responses in a quantitative way. Even though the three tests should be evaluated in parallel, 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 and, consequently, CHR are achieved. Therefore, the three mentioned types of response can be aggregated into one accumulated response achievement.

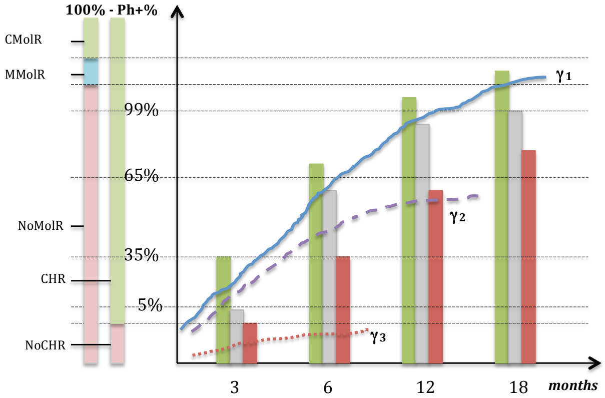

Figure 3. Response definition graph

Figure 3 depicts a graph with three sets of bars for each defined time when tests should be performed (3, 6, 12 and18 months). Each set contains three bars. The leftmost bar corresponds to the minimum level of the *optimal* response, while the rightmost represents the maximum level still evaluated as failure. The middle (gray) bar is associated with a suboptimal response. *Optimal* response means that based on current knowledge, the patient is projected to have a normal survival. From the quantitative point of view, if the value of measured cumulative response achievement is above the optimal response bar the result will still be evaluated as optimal. The concept of *failure* here means that the patient still may do well also for years, but will never become an optimal responder, this way, a change of treatment should be considered. Therefore, if the accumulated response value is below the rightmost bar it is considered to be a *failure*. *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.

The three cumulative curves γ_1, γ_2 and γ_3 crossing the graph represent three different scenarios of progressive patient reaction to the treatment. The first (γ_1) curve corresponds to a situation when an optimal response is achieved. The second (γ_2) curve represents a failure of the treatment after one year. The third (γ_3) curve represents lack of response to the treatment within 6 month. Based on the response definition we can build an observer TA that would control the level of accumulated response achievements. The least restrictive way would be to build it based on the level of failure bars. This way we would like to insure 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.

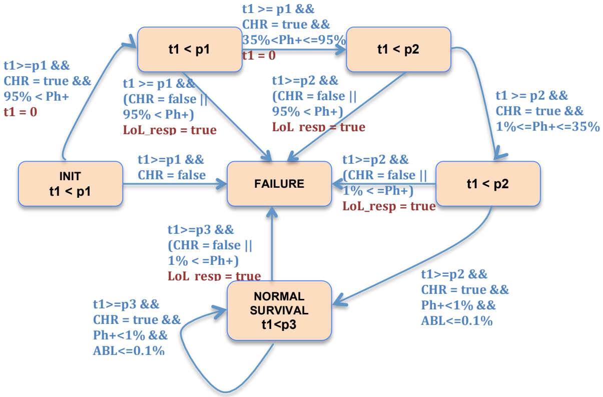

Figure 4. Observer TA

The controlling observer TAT is depicted on Figure 4 and represent a complementary model to the one presented in Section IV-A. The two models synchronize by means of a shared variable *LoL_resp*. The periods p1and p2 of the model are assigned to equal 3 and 6 months respectively. We have only two periods since time will also be accumulated. Execution of the model starts from the initial location *INIT* in which the model will stay for 3 month 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 from the beginning of treatment). The model may proceed up to the *NORMAL SURVIVAL* location. The input data to the model, such as the amount of Ph+ chromosomes or the ABL level should come from the measurement TAT similar to the one in Section IV-A.

C. Model Verification and Correction

The verification of a system with UPPAAL or TIMES requires that the system is *closed*, which means that the behavior (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 [19] suggests to compose an *open* system with the *maximal environment*, that enables all the external nondeterministic choices, make the *guards* that depend on the environment always evaluated to **true**. This composition will be a closed system that contains all possible behaviors of that system combined with any other environments. Such a composition raises the level of non-determinism in the model behavior, however, the model structure remains the same and thus can be verified.

1) Validation of structural properties: The case study above represents treatment of a chronic disease, which by definition may not have any positive result. However, there still should be a positive exit point (location) to describe even a highly nonprobable case of a miracle. Therefore, we artificially 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. An *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. 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. This way, in the chronic phase part of our case study we add a transition from the *repetitive_anemia* to the *lack_loss_resp_ch* location. Another incompleteness problem was detected in the response definition model presented in [8]. Its does not say anything about what will happen after 18 months, therefor we added, artificially, 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.

2) Real-time and Life-cycle Properties Verification: The verification of the real-time and life-cycle properties should be performed after the structural protocol verification, when there are no obvious structural defects. Since potentially an automatic decision-support system can be combined with an autonomous electronic device, it is essential to perform the verification of the real-time properties of the system, so that communication tokens and, therefore, notifications will never be delayed more then a deadline. This kind of verification is called *Schedulability Analysis* [15] and can be automatically performed in TIMES tool for the parts of the protocol that are planned to be automated, e.g. a periodic drug delivery task combined with some other computational tasks running on the same computational unit. Life-cycle properties are those that require that the modeled system contains all the components of the real system, including the patient model. However, the patient model is a very complicated task. There exist several models of the patient body reaction. However, they reflect only a specific aspect of drug concentration prediction in the blood. Among them we can name the pharmakokinetical models [13] and the SVM-based approach [29].

V. 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 methodologie exploits the Timed Automata extended with Tasks (TAT) model widely used for the real-time systems analysis. We have shown that TAT is expressive enough to represent a step-by-stem treatment procedures as well as formalize a non immediate effect of the treatment which is important for closing the patient care loop. We have presented the Imatinib dose adjustment protocol case study and a model of the response levels control based on the levels of accumulated achievements. We where able to fix some incompleteness problems in these models. The verification of life-cycle properties of medical GLs required a patient body reaction model. There exist several models that may predict the drug concentration in the patient blood, however, their use with respect to our case study implies the change of the protocol or use of another protocol, such as the therapeutic drug monitoring approach [20]. In the next step of our work we will focused on closing the design loop.

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 N. Widmer and V. Gotta from CHUV Hospital of Lausanne for the discussion regrading the treatment procedures with Imatinib.

References

[1] SPIN. http://spinroot.com/spin/whatispin.html.

[2] SMV. http://www.cs.cmu.edu/ modelcheck/smv.html.

[3] TIMES tool. http://www.timestool.com/.

[4] AGREE instrument. http://www.agreetrust.org/.

[5] Imatinib. http://www.ema.europa.eu/docs/en_GB/document_library.

[6] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[7] T. Amnell, E. Fersman, P. Pettersson, H. Sun, and W. Yi. Code synthesis for timed automata. *Nord. J. Comput.*, 9(4):269–300, 2002.

[8] 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):331–41, 2009.

[9] S. Bäumler, M. Balser, A. Dunets, W. Reif, and J. Schmitt. A verification of medical guidelines by model checking - a case study. In *In Proc. Model Checking Software SPIN*, 2006.

[10] 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*, pages 200–236. Springer, 2004.

[11] 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):147–161, June 2004.

[12] P. Ciccarese, E. Caffi, L. Boiocchi, S. Quaglini, and M. Stefanelli. A guideline management system. *Stud Health Technol Inform*, 107(Pt 1):28–32, 2004.

[13] C. Csajka, C. Marzolini, K. Fattinger, L. A. Dcosterd, J. Fellay, A. Telenti, J. Biollaz, and T. Buclin. Population pharmacokinetics and effects of efavirenz in patients with human immunodeficiency virus infection. *Clinical Pharmacology and Therapeutics*, 73(1):20–30, 2003.

[14] G. Duftschmid, S. Miksch, Y. Shahar, and P. Johnson. Multi-level verification of clinical protocols. In *in: Proceedings of the Workshop on Validation Verification of Knowledge-Based Systems (VV98), in conjunction with the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR98*, pages 4–1, 1998.

[15] E. Fersman, L. Mokrushin, P. Pettersson, and W. Yi. Schedulability analysis of fixed-priority systems using timed automata. *Theor. Comput. Sci.*, 354(2):301–317, Mar. 2006.

[16] M. Fieschi, P. Votruba, S. Miksch, and R. Kosara. Facilitating knowledge maintenance of clinical guidelines and protocols, 2004.

[17] P. D. Johnson, S. Tu, N. Booth, B. Sugden, and I. N. Purves. Using scenarios in chronic disease management guidelines for primary care. In *Proceedings of American Medical Informatics Association(AMIA) Symposium*, pages 389–93. PubMed Central PMCID, 2000.

[18] 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 *Proceedings of the 9th ACM international conference on Embedded software*, EMSOFT '11, pages 155–164, New York, USA, 2011. ACM.

[19] O. Kupferman and M. Y. Vardi. Model checking revisited. In *Proceedings of the 9th International Conference on Computer Aided Verification*, CAV '97, pages 36–47, London, UK, 1997. Springer-Verlag.

[20] 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):21–24, 2010.

[21] S. Miksch, Y. Shahar, and P. Johnson. Asbru: a task-specific, intention-based, and time-oriented language for representing skeletal plans. In *UK, OPEN UNIVERSITY*, pages 9–1, 1997.

[22] 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):418–26, Sep-Oct 2004.

[23] A. Simalatsar and G. De Micheli. TAT-based Formal Representation of Medical Guidelines: Imatinib Case-study. In *Engineering in Medicine and Biology Society, EMBC, 2012 Annual International Conference of the IEEE*, Aug. 28 - Sept. 1 2012, https://infoscience.epfl.ch/record/179883.

[24] 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*, 6:20+, Apr. 2006.

[25] 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.

[26] 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):589–598, June 2007.

[27] S. W. Tu and M. A. Musen. Modeling data and knowledge in the EON guideline architecture. *Medinfo*, 10(1):280–284, 2001.

[28] P. Votruba, S. Miksch, A. Seyfang, and R. Kosara. R.: Tracing the formalization steps of textual guidelines. In *in Computer-based Support for Clinical Guidelines and Protocols*, pages 172–176. Press, 2004.

[29] W. You, N. Widmer, and G. De Micheli. Example-based support vector machine for drug concentration analysis. In *Engineering in Medicine and Biology Society,EMBC, 2011 Annual International Conference of the IEEE*, pages 153 –157, 30 2011-sept. 3 2011.