

TAT-based Formal Representation of Medical Guidelines: Imatinib Case-study*

Alena Simalatsar and Giovanni De Micheli¹

Abstract—Computer-based interpretation of medical guidelines (GLs) has drawn lots of attention in the past three decades. It is essential to use a formalism for GLs representation that would enable the validation of GLs structural properties, be able to map medical actions into the time scale and support the automatic formal verification of GLs without additional translation paths. In this paper we present 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.

I. INTRODUCTION

Computer-based interpretation of GLs or medical protocols aimed to assist practitioners in their daily routine can be classified as tools providing the recommendation for the structural representation of the medical GL in text form, the text markup tools, and, finally, decision-support tools that form the central and the most essential class of the existing frameworks. These tools help to improve the quality of GLs, since the initial textual representation of the protocol often suffers from such structural problems as incompleteness, inconsistency, ambiguity and redundancy. Moreover, various decision-support tools are aimed to improve the quality of medical service by reducing the human factor errors when providing automatic healthcare recommendations or reminding about some procedures that could have been overlooked.

There have been many frameworks and languages [1]–[5] developed in the past years in order to assist medical doctors as well as patients. Many of them such as PRODIGY [2], EON [1], GLIF3 [3], PROforma [4], SAGE [5] represent a class of tools used to build complex decision-support systems. They adapt flow-charts as a core formalism to represent a sequence of actions that are supported by ontology based medical terminology interpretation modules. Most of these tools also provide links to patients' databases. However, they enable validation of GLs structure only by means of their formal representation and have no support for the automatic verification of their formal properties. There exist frameworks such as GLARE [6] and Asbru [7] that provide translation links to model checking environments such as SPIN [8] and SMV [9]. 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 and association of

time periods with respect to the patient condition evaluation, which enables only the validation of GLs structure. While, it is important to be able to map medical actions into the time scale in order to verify timing properties of a GL and thus close the gap of medical software and hardware interoperability.

In this paper we first evaluate the requirements for the protocol modeling formalism. As a key contribution, we present a novel approach for the medical protocol formal representation based on Timed Automaton extended with Tasks (TAT). TAT is an extension of the Timed Automaton (TA) [10], an automaton extended with clocks. For protocols formal representation with TAT we exploit the TIMES toolbox [11], which is a successor of the UPPAAL [12] toolbox, 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 where properties are described by means of Computational Tree Logic (CTL). It is possible to connect TAT models with other tools by means of tasks that can play a role of interfaces, function calls, to some external environments. The TAT model can also be turned to a fully synthesizable deterministic model [13] which is essential when we would like to generate executable code. We present a case study of modeling the Imatinib dose adjustment part of the protocol for adult patients with newly diagnosed Ph+ Chronic Myeloid Leukemia (CML) for whom bone marrow transplant is not considered as the first line of treatment. We perform the validation of the protocol structure as well as discuss issues of the protocol life-cycle properties verification. In Section II we define the requirements to the modeling methodology and introduce the TA and TAT models as our approach. Section III presents the Imatinib case study and discusses the verification issues.

II. FORMALIZATION APPROACH

Our view to the transformation steps applied to a textual representation of GLs in the automation process are presented in Figure 1. The flow starts from a textual document summarizing sequence of actions of a GL based on the experience and medical evidence. In order to choose an appropriate formalism one should first evaluate all its potential problems and requirements. We distinguish several key requirements. First of all it is important to choose a formalism that has a notion of time and allows modeling of the choice of sequential actions based on specific conditions and level of evidence (probabilities) mapped to the time scale. It should allow the modeling of several plans with pe-

*This work was supported by ISyPeM Project with a grant from the Swiss Nano-Tera initiative, evaluated by the Swiss National Science Foundation.

¹A. Simalatsar alena.simalatsar@epfl.ch and G. De Micheli giovanni.demicheli@epfl.ch are with Integrated Systems Laboratory, EPFL, 1015 Lausanne, Switzerland

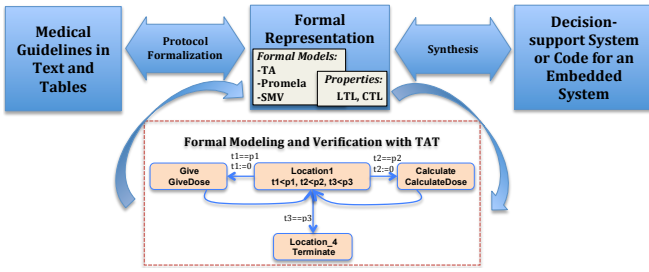


Fig. 1. General methodology

riodically repeating actions that can be execution in parallel with other actions. It is also important to be able to associate some goals with the treatment, e.g. to achieve some effect or finish some procedure after a period of time. Very often the initial textual representation of the protocol suffers from: (i) *incompletenesses*, (ii) *inconsistencies*, (iii) *ambiguity*, and (iiii) *redundancies*. 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. The central part of Figure 1 shows the verification process. Moreover, medical software is tightly coupled with the medical devices that perform the patient state monitoring and even some basic treatment procedures [14]. Medical protocols can play a role of an intersection point between the medical software and electronic devices. Therefore, it is essential to enable the verification of the timing and life-cycle properties of the 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 [15] or to do the code synthesis for an embedded system as in [16]. Further in the paper we present TAT as a key formalism for protocols representation.

1) *Timed Automata (TA)* [10]: is a formal model of computation often used to describe the execution semantics of a system and its progress in time. The classical Automaton is a finite state graph composed of the finite set of locations Loc and transition relations (edges) \hookrightarrow . Each location may contain a various number of incoming and outgoing \hookrightarrow . The location that does not have any incoming \hookrightarrow is called *initial* location. The locations that have no outgoing \hookrightarrow are either *final* or *deadlock* locations. TA extends the classical Automaton with the finite set of clocks C and a set of constraints on these 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 which can then enforce a transition to another location. Formally, *Timed Automaton* is defined as a tuple $TA = (Loc, Loc_0, Act, C, \hookrightarrow, I)$, where Loc is the finite set of locations, $Loc_0 \subseteq Loc$ is the set of initial locations, Act is a the set of action (transitions labels), C is the finite set of clocks, $\hookrightarrow \subseteq Loc \times ClockCons(C) \times Act \times 2^C \times Loc$ are its transitions, and $I : Loc \rightarrow ClockCons(C)$ is the map of invariants, which maps locations to clock constraints.

While the invariants of the location the TA currently resides in hold, time may progress, with all clocks increasing at the same speed. Transitions within a TA may either change the current location, or perform a self-loop (\odot) to the same location. Transitions are instantaneous and possible at any time, given that the source location of the transition is marked active and its guard evaluates to true. The timed model checker UPPAAL [12], however, 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. One TA can describe one instance of a medical protocol. A combination of medical GLs can be composed from a set of protocol instances, separate TA cooperating among each other.

2) *Cooperating TA*: 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 synchronized transitions.

a) *Cooperation via shared variables*: The values of the variables declared on the level of a network of TA, can be read and changed by individual TA.

b) *Rendez-vous mechanisms*: UPPAAL makes use of synchronization channels. TA may jointly execute their enabled edges if the involved edges carry the same (channel) identifier a followed by a question or exclamation, marked $a?$ for the receiver or $a!$ for a sender respectively.

3) *Timed Automata Extended with Tasks (TAT)*: TAT [13] is an extension of TA with tasks that represent pieces of code associated with locations of the model. The execution of a task is triggered by incoming transitions of the location with the task. Formally, TAT is defined as a tuple $TAT = (Loc, l_0, Act, X, \hookrightarrow, I, M)$, where Loc is the finite set of locations, $l_0 \subseteq Loc$ is the initial location, Act is a set of action (transitions labels), X is the disjoint union of the finite set of clocks and variables, $\hookrightarrow \subseteq Loc \times ClockCons(X) \times Act \times 2^X \times M \times Loc$ are its transitions, $I : Loc \rightarrow ClockCons(X)$ is the map of invariants which maps locations to clock and variable constraints, and $M : Act \rightarrow 2^P$ is a partial (some actions have no task) function assigning actions with set of tasks. The execution semantics of TAT is the one of TA extended with task queues, where tasks are executed upon a chosen scheduling policy.

III. IMATINIB CASE STUDY

In Figure 2 we present a model of the Imatinib dose adjustment protocol. It implements the drug administration part of [17] for adult patients whereby the dose should be administered once a day ($p1 = 1$ day).

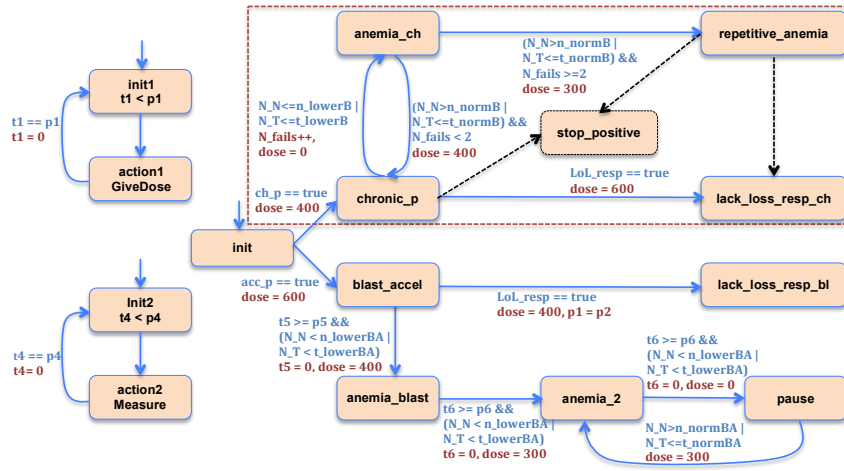


Fig. 2. Imatinib dose adjustment protocol

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 (\leftrightarrow) from *Init* to *chronic_p*) of CML and 600 mg/day for patients in the accelerated phase (*Init* \leftrightarrow *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* \leftrightarrow *lack_loss_response_ch*) or from 600 mg to a maximum of 800 mg given as 400 mg twice daily (*blast_accel* \leftrightarrow *lack_loss_response_bl* and $p1 = p2$, where $p2 = \text{half day}$) in patients with accelerated phase or blast crisis in case of: (i) disease progression at any time ($D_p == \text{true}$); (ii) failure to achieve a satisfactory hematological or cytogenetic response ($lack_r == \text{true}$); or (iii) loss of a previously achieved hematological and/or cytogenetic response ($loss_r == \text{true}$). The definition of disease state and response to the treatment is presented in [18] where levels of the response can be also represented with a separate network of TAT models.

In the *chronic* phase of CML, marked with a rectangle in Figure 2, if the level of neutrophils (ANC) goes below $1.0 \times$

$10^9/l$ ($N_N \leq n_{lowerB}$) and/or level of platelets goes below $50 \times 10^9/l$ ($N_T \leq t_{lowerB}$):

- 1) Stop Imatinib until ANC $> 1.5 \times 10^9/l$ and platelets $> 75 \times 10^9/l$ (*chronic_p* \leftrightarrow *anemia_ch*);
- 2) Resume treatment with Imatinib at previous dose, (*anemia_ch* \leftrightarrow *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 $< 50 \times 10^9/l$, repeat step 1 and resume Imatinib at reduced dose of 300 mg (*anemia_ch* \leftrightarrow *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 2.

A. Verification Issues

The model checking verification of a system requires that the system is *closed*, when its behavior is completely determined by the states of itself. However, some of the transitions guards of the Imatinib model depend on the information coming from external models. For example, the values of level of the *neutrophils* and *platelets* is provided by a model of the patient body reaction to the treatment. As another example we have the Boolean variables that determine the disease progression or lack/loss of the response. This particular model can be build upon the response definition presented in [18]. Therefore, the presented model is an *open* system, the properties of which should be verified with respect to any environment, which in turn may be very uncertain as in case of patients body reaction.

The *Module Checking* approach presented in [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. Therefore, we evaluate to true the guards related to the levels

of neutrophils and platelets as well as those related to the disease state and verify the structural properties of the model.

The entry point in TAT is represented by an initial location and signifies the beginning of the treatment procedure. The exit points are the locations that determine the end of the treatment that can be classified as by having positive, when the *goal* of the treatment procedure was achieved, and negative, when the treatment has failed, results, and thus the change of the protocol is required. The model above represents a treatment of a chronic disease, with no positive result, however, there still should be a positive exit point (location) to describe even a highly nonprobable case of a miracle. We artificially added the *stop_positive* location in the protocol chronic phase reachable from the *chronic_p* and *repetitive_anemia* locations. The locations *lack_loss_resp_ch* and *lack_loss_resp_bl* play the roles of negative exit points.

Incompleteness problems exists in both the chronic and the acceleration phases of the model, where once we go to the *repetitive_anemia* or *anemia_2* locations there is either no outgoing transition (*repetitive_anemia*) or we enter a closed loop (*anemia_2* \circlearrowright *pause*). This problem is found by verifying whether the *stop_positive* or *lack_loss_resp_ch* locations are always reachable from *chronic_p*: *chronic_p* $\rightarrow E \langle \langle \rangle \rangle$ (*stop_positive* or *lack_loss_resp_ch*) and finding the contrexample leading to the *repetitive_anemia* deadlock location. This way, in the chronic phase part of the model we added a transition *repetitive_anemia* \leftrightarrow *lack_loss_resp_ch*.

Life-cycle properties verification requires that the modeled system is complete, including the patient model and should be verified after the structural protocol verification. There exist several models of the patient body reaction reflecting only such a specific aspect as drug concentration in the blood among which we can name the pharmacokinetical models [20] or the SVM-based approach [21]. The Imatinib protocol presented above can not benefit from these models. However, a novel therapeutic monitoring approach [22] in the drug administration may be a potential candidate.

IV. CONCLUSIONS

In this paper we have presented a new approach for computer-based interpretation of the medical protocols that exploits the Timed Automata extended with Tasks (TAT) model widely used for the modeling of real-time systems. To prove the applicability of our approach we have presented the Imatinib dose adjustment protocol case study. The analysis of the protocol structure showed incompleteness problem of the protocol. Several structural problems have been fixed. The verification of the life-cycle properties of medical protocol remains an opened question for the moment.

ACKNOWLEDGMENT

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] S. W. Tu and M. A. Musen, "Modeling data and knowledge in the EON guideline architecture," *Medinfo*, vol. 10, no. 1, pp. 280–284, 2001.
- [2] 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*, pp. 389–93, PubMed Central PMCID, 2000.
- [3] 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*, vol. 37, pp. 147–161, June 2004.
- [4] 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*, vol. 6, pp. 20+, Apr. 2006.
- [5] 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*, vol. 14, pp. 589–598, June 2007.
- [6] 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*, vol. 101, 2004.
- [7] G. Duftschmid, S. Miksch, Y. Shahar, and P. Johnson, "Multi-level verification of clinical protocols," 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)*, pp. 4–1, 1998.
- [8] SPIN. <http://spinroot.com/spin/whatspin.html>.
- [9] SMV. <http://www.cs.cmu.edu/modelcheck/smv.html>.
- [10] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.
- [11] TIMES. <http://www.timestool.com/>.
- [12] 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*, pp. 200–236, Springer, 2004.
- [13] T. Amnell, E. Fersman, P. Pettersson, H. Sun, and W. Yi, "Code synthesis for timed automata," *Nord. J. Comput.*, vol. 9, no. 4, pp. 269–300, 2002.
- [14] D. Arney, M. Pajic, J. M. Goldman, I. Lee, R. Mangharam, and O. Sokolsky, "Toward patient safety in closed-loop medical device systems," in *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS '10*, pp. 139–148, 2010.
- [15] 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 *Engineering of Computer Based Systems, 2008. ECBS 2008. 15th Annual IEEE International Conference and Workshop on the*, pp. 210–218, 31 2008–april 4 2008.
- [16] 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 *Embedded Software (EMSOFT), 2011 Proceedings of the International Conference on*, pp. 155–164, oct. 2011.
- [17] Imatinib. <http://www.ema.europa.eu/ema/>.
- [18] M. Baccarani, F. Castagnetti, G. Gugliotta, F. Palandri, and S. Soverini, "Response definitions and european leukemianet management recommendations," *Best Pract Res Clin Haematol*, vol. 22, no. 3, pp. 331–41, 2009.
- [19] O. Kupferman and M. Y. Vardi, "Model checking revisited," in *Proceedings of the 9th International Conference on Computer Aided Verification, CAV '97*, (London, UK), pp. 36–47, 1997.
- [20] 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*, vol. 73, no. 1, pp. 20–30, 2003.
- [21] 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*, pp. 153–157, 30 2011–sept. 3 2011.
- [22] 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*, vol. 31, no. 1, pp. 21–24, 2010.