Mixed measurement and model-based dose recommending control flow with fixed sample library size

Alena Simalatsar*[2‡], Wenqi You[1‡], Dechao Sun[1], Verena Gotta[3], Nicolas Widmer[4] and Giovanni De Micheli[1]

[1]Lab of Integrated Systems, Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne, Switzerland
[2]Rigorous System Design Laboratory, Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne, Switzerland
[3]Leiden Academic Centre of Drug Research, Leiden University
[4]Division of Clinical Pharmacology, University Hospital Center of Lausanne (CHUV), Lausanne, Switzerland
‡ Equal contributors

Email: Alena Simalatsar*- Alena.Simalatsar@epfl.ch; Wenqi You - Wenqi.You@epfl.ch; Dechao Sun - Dechao.Sun@epfl.ch; Verena Gotta - v.gotta@lacdr.leidenuniv.nl; Nicolas Widmer - Nicolas.Widmer@chuv.ch; Giovanni De Micheli - Giovanni.Demicheli@epfl.ch;

*Corresponding author

Abstract

Background: The drug administration is a routine action regularly performed in every day medical practice that requires a lot of time and attention of medical personal. Therefore, it can benefit from the process automatisation using electronic devices, which will save time and thus reduce the treatment cost. The number of human factor errors will be also reduced, however, often at the price of possibly introducing new type of errors due to failures or unpredictable system behaviour. Even though such errors are rare, one small error may harm or even kill a patient. Hence, due to high risk associated with the usage of such devices, they belong to the class of safety-critical systems. In embedded system design domain safety-assured development of the software running on a device is addressed by employing formal methods, e.g. theorem proving, verification and/or correct-by-construction software development. In addition, there exist drugs (medicines) with a narrow therapeutic (effective and non toxic) range, such as those used to treat cancer, HIV, diabetes, shock, asthma, sepsis, and cardiovascular failures, as well as anaesthetics, analgesics, muscle relaxants, immunosuppressants, anti inflammatory drugs, which may be easily under- or overdosed. This resctrics the requirements to software running on drug delivery devices (e.g., infusion pumps). The introduction of a feedback-loop able to support drug dose and delivery rate adaptation based on patient current state is the approach presented in this article. The control flow is modelled with a formal model of computation that provides means for automatic verification of the design properties.

Methods: Various clinical studies show that responsiveness to treatment with a drug depends on the concentration of the medication in patient's blood, correlated with patients features, drug dose and delivery rate. Pharmacokinetics (PK) is a branch of pharmacology focused on studying the drug disposition in the human body. For lots of drugs the concentration in patient's blood is related to its effects. Pharmacodynamics (PD) is the study of the biochemical and physiological effects of drugs on the body. The concentration in the patient's blood is closely related to the drug effect (PK-PD relationship). The approach for safe closed loop automatic delivery of each of these drugs varies depending on drug PK and PD properties of the drug. Faster PK of the drug, meaning faster drug absorption, distribution, metabolism and elimination rate, requires more frequent drug delivery in order to maintain the plasma drug concentration within the therapeutic range, which implies harder real-time constrains on decision-making process.

Currently, the patient state is usually evaluated by either performing a real measurement of specific biomarkers, rarely drug concentration, as it is done within the Therapeutic Drug Monitoring (TDM) procedure; or, otherwise, by using models able to predict drug concentration in blood. Drug concentration mea-

surement with TDM is quite slow (takes about a day), expensive and requires an invasive measurement of individual drug concentration. Therefore, a frequently performed algorithm that is able to individually compute the drug concentration in patients blood may complement a more slow and expensive TDM. Recently, some advanced personalised methods based on Support Vector Machine(SVM) algorithms for drug concentration prediction have been introduced. They use the library of real measurements acquired with TDM. Therefore, one patient may benefit from every new measurement made for this patient added to the library.

Results: We present a generalised approach to the design of a control flow with a feedback loop for personalised drug dose recommending device. We support this approach with a prototype examples for delivery of *imatinib*, a drug used to treat Chronic Myeloid Leukaemia (CML). For this case study the feedback loop is implemented as a mix of occasional TDM measurements and model-based prediction of the drug concentration implemented as a parametrised Support Vector Machine (SVM) algorithm. This algorithm uses the population sample library updated and thus personalised with each new TDM measurement for currently treated patient. The long term treatment usually generates a large number of samples/measurements, which implies that the sample library will constantly grow. This may become a critical problem due to the memory limitation, especially if one imagines a small embedded device for automatic drug administration. Therefore, in our example each measurement preempts the place of the least relevant sample in the library to keep the library data size constant. The control flow makes decisions about the drug dose adaptation based on the outputs of the concentration prediction. The decision is further checked for conformity with the dose level rules of the medical Guidelines (GLs) for treatment with *imatinib*. The control flow is modelled using Timed Automata extended with Tasks (TAT) formal model that can be further synthesised using newly developed TIMES tool code generator for the *icycom* platform.

Conclusion: The provided examples of how to implement particular parts of the closed-loop drug delivery system are not unique and often depend of the drug for which the system is developed, mainly due to the fact that medical guidelines (GLs) for each drug are specific. The parametrised SVM algorithm used to adjust the dose and rate of drug delivery is not unique. It may be replaced by any other algorithm able to provide drug dose and rate adaptation based on occasional TDM measurements. However, we argue that introduction of a formal feedback loop to monitor the patient state helps to increase reliability and safety of the overall medical system.

Keywords: Medical cyber-physical systems, feedback loop, drug concentration prediction, medical guidelines.

1 Background

When treating patients, medical doctors are often following medical Guidelines (GLs) that are also called clinical protocols. They contain recommendations for practitioners needed to set the patient management procedures, perform the diagnosis and/or guide them through the treatment procedures. The treatment description includes not only the action-based step-by-step procedures but also recommendations to the action changes upon the definition-based response to the treatment, which is usually also specified by GLs. Therefore, they represent an informal control flow that synchronises the processes of data acquisition, decision-making and treatment provision, which should create the logical base for medical software of electronic devices providing treatment. This way medical GLs can be viewed as a specification of a medical system.

1.1 Medical guidelines formalisation

There have been many frameworks and languages [1–7] developed in the past years for medical guideline representation in order to assist medical doctors as well as patients. Many of them such as PRODIGY [1], EON [2], GLIF3 [3], PROforma [4], SAGE [5], GLARE [6] and Asbru [7] 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.

However, automation of drug administration using electronic devices requires medical GLs representation that can allow the action management with precise time characteristics, real-time properties, while the above mentioned 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. To address this problem in [8, 9] we have proposed to used the Timed Automata extended with Tasks (TAT) models of computation, a well-known approach for modelling the behaviour of real-time systems. *Timed Automaton* (TA) [10] is a formal model of computation used in embedded system design domain 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, $a \subseteq Act$.

TAT [11] 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. The timed model checker UPPAAL [12], 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. TIMES tool [13] is the successor of UPPAAL that implements TA extended with tasks. The main advantages of using TA or TAT is that UPPAAL and TIMES are model checking environments that allow automatic verification of the models they implement.

As a case study, we present a TAT model of a small part of the dose adjustment medical guideline for an adult patient of the anticancer drug *imatinib*, marketed by Novartis as Gleevec® or Glivec®. It is a drug used to treat Philadelphia positive (Ph+) Chronic Myeloid Leukaemia (CML), Gastrointestinal Stromal Tumors (GISTs) and some other malignancies. A complete text of the *imatinib* drug administration protocol can be found here [14]. The model

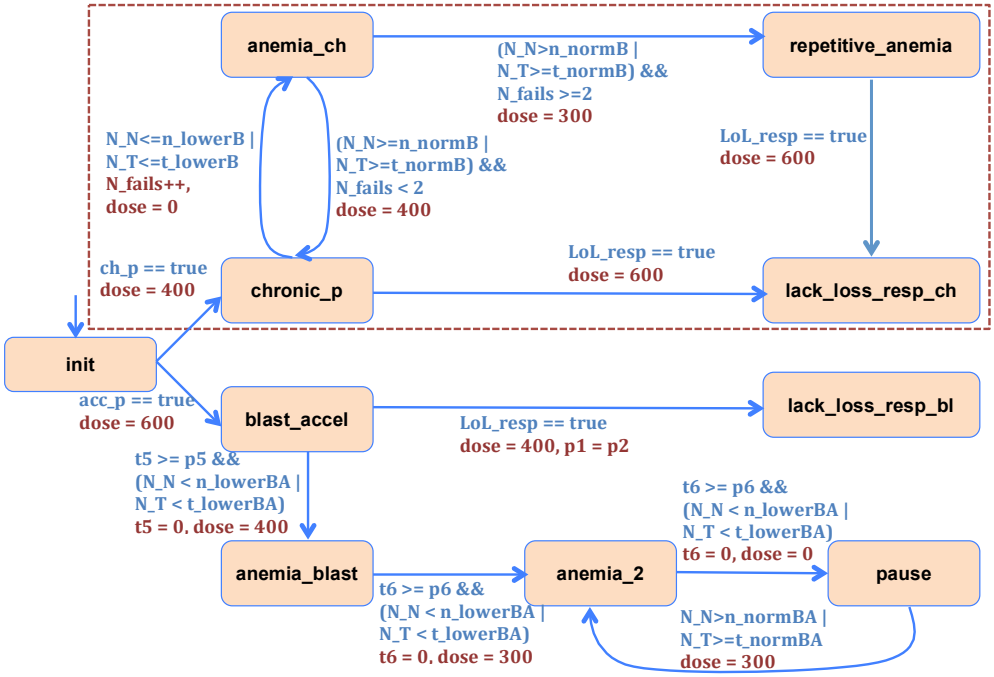

Figure 1: *Imatinib* dose adjustment model

depicted on Figure 1 includes only a small part of the *imatinib* GL describing the dose adjustment model for an adult patient with newly diagnosed Ph+ CML for whom bone marrow transplantation is not considered as the first line of treatment. A more complete model can be found in [15].

The dose is usually administered once a day. Medical tests of the level of neutrophils (N_N) and platelets (N_T) are usually performed every 2 weeks. The dose adjustment model uses the latest data of the medical tests and adjust the dose and intake interval of the drug. The recommended dosage of *imatinib* is 400 mg/day for patients in the *chronic* phase (ch_p) of CML (transition from the *init* to $chronic\_p$ location) and 600 mg/day for patients in the *accelerated* phase (acc_p) 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 model presented above 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 [16].

In the *chronic* phase of CML, marked with a rectangle in Figure 1 (starting dose 400 mg) the dose adjustments are performed as follows:

1. If the level of neutrophils goes below 1.0 x 10^9/l ($N\_N <= n\_lowerB$) and/or level of platelets goes below 50 x 10^9/l ($N\_T <= t\_lowerB$): stop the treatment with *imatinib* until ANC >1.5 x 10^9/l ($N\_N >= n\_normB$) and platelets >75 x 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 neutrophils <1.0 x 10^9/l and/or platelets <50 x 10^9/l, while $N\_fails >= 2$, repeat step 1 and resume *imatinib* at reduced dose of 300 mg (transition $anemia\_ch$ to $repetitive\_anemia$);

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. The complete modelling of *imatinib* medical GL can be found in [9].

Thanks to the GL representation with TAT we can perform its structural validation as was presented in [15] and fix certain problem. In Section 3.1 we present a control-flow prototype that combines the rules derived from the model above with the closed-loop drug delivery logic. The main advantage of TAT is that this model is deterministic and, therefore, the automatic platform oriented code synthesiser can be developed. An example of such code synthesis is presented in Section 3.2.

1.2 Closed loop of drug administration

From the system design point of view a patient is a reactive system that responds to any treatment events. This way we need to approach the task of medical systems design as of a cyber-physical system where the actions of the cyber part are changing with respect to the state of a physical system. The continuous adjustment of the drug dosage and delivery rate can be done by feedback loop that either performes real measurements of specific biomarkers or even drug concentration, as it is done within the Therapeutic Drug Monitoring (TDM) procedure; or, otherwise, by using models able to predict drug concentration in blood.

An example of drug delivery control based on measuring specific biomarkers is presented in [17], where authors introduce a safety-assured approach to the development of a Generic Patient Controlled Analgesic (GPCA) infusion pump using TA model. The feedback loop is based on measuring such indirect marker as SpO_2 concentration in patient's blood to evaluate oxygenation. The low level of SpO_2 indicate respiratory insufficiency and thus the drug delivery must be stopped. In [18] authors represent examples of the control software, the feedback loop of which is represented by the offline Pharmacokinetic (PK) based model simulating changes of drug concentration in blood.

The continues closed-loop drug administration with TDM would be the best solution, since it provides the direct correspondence between the dose delivered and the actually value that need to me controlled, e.g. drug concentration. However, it is difficult to implement, mainly due to the fact that one measurements of the actually drug concentration is quite slow (takes about a day), expensive and requires an invasive measurement of individual drug concentration. Therefore, it is often performed only in case of adverse events or suboptimal response. This way, a frequently performed algorithm that is able to individually compute the drug concentration in patients blood may complement a more slow and expensive TDM. Included in a control-flow it would represent a surrogate of body reaction and thus can close the loop of the control flow.

There have been several models developed in support of PK studies that are able to predict the drug concentration in the blood. These models can be classified as analytical [19] and statistical [20]. The analytical models are able to

account only for the variables with real values, while binary-valued variables, such as gender, create strong discontinuities in the models and are in general not taken into account by the methods. Moreover, the analytical models are based on differential equations that are hard to modify if we were to add new parameters. The main drawback of the statistical approach, including Bayesian approach [21], is that it requires the knowledge of the data distributions, such as mean and/or deviation values. For newly-developed drugs which do not have a sufficient study on the patients, it is difficult to give a proper mean or deviation value to compute the concentrations for the following patients. Moreover, in order to provide a mixed measurement and model-based dose recommendation, the model in use should be able to incorporate each new TDM measurement and provide means for self-calibration.

Several personalised drug concentration prediction method based on *Support Vector Machine* (SVM) algorithm where presented in [22–24]. However, all these methods are able to perform only a point-wise drug concentration prediction and have no knowledge about the analytical dependency among all predicted values for one patient with fixed parameters. Therefore, it is impossible to calibrate in personalised manner the prediction every time when a new measured concentration value is available for the patient under treatment. These measurements will contribute only to the initial population studies, since they must be added to the common database and the whole relearning and predicting process must be reapplied. Therefore, each concrete patient cannot fully profit from his personal seldom TDM measurements. Moreover, for the closed loop drug delivery we need to solve the inverse problem, e.g. based on the drug concentration provide the recommendation of the dose and delivery rate. In Section 2.2 the Drug Concentration over Time (DCT) curve prediction approach, Parameterized SVM, which combines the SVM and analytical models is presented. This method allows to introduce a mixed approach, in which we unify an offline SVM-based drug concentration prediction model (different from PK modelling of [18]) and occasional direct TDM measurements of the real drug concentration in the blood. However, long term treatment usually generates a large number of samples/measurements. The potential embedded electronic device on which we can deploy the closed loop algorithm may have memory limitations. Therefore, it is essential to keep the size of sample library constant. This problem is address in Section 2.2.2.

2 Methods

This section is divided into two subsections, where the first one 2.1 presents the details of the feedback loop realization, while the second one 2.2 describes the parametrized SVM (ParaSVM) algorithm used in the feedback loop as a body reaction model.

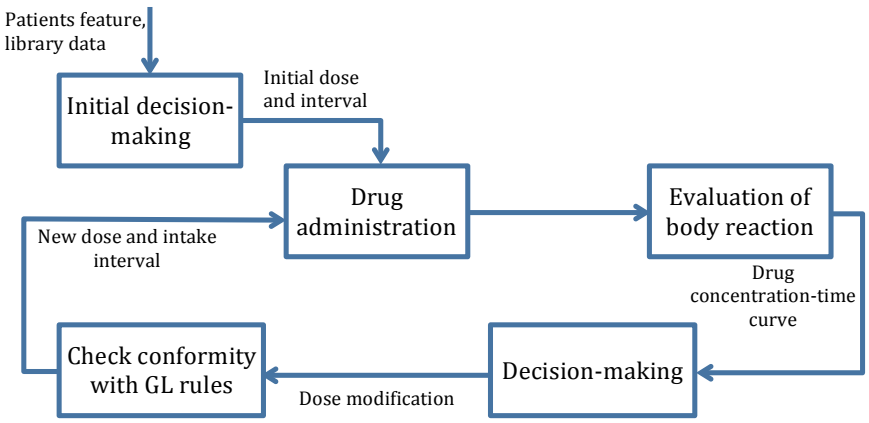

Figure 2: General control flow with a feedback loop

2.1 Feedback loop

Figure 2 depicts the abstract representation of a general control flow with the feedback loop for an autonomous drug administration support device. The detailed control flow is presented in Section 3.1.

It starts with an initial decision-making process that provides the initial dose and intake interval. The *drug administration* block can either perform the actual drug delivery or remind a patient or a caregiver when the drug needs to be administrated and with what dose for oral drugs such as *imatinib* [14]. Once the drug is administered, it is essential to evaluate the effect of the treatment reflected in the *evaluation of body reaction* block of the diagram. The decision regarding the next dose is then taken based on the body reaction (the *decision-making* block) and must conform with the medical GL for this specific drug administration (*check conformity with GL rules*). This way we have a cyclic procedure with a feedback loop.

The realization of a feedback loop for a medical system requires a patient body reaction model, which can provide us with the information about the response to the treatment required by the medical GL. The GL model in turn should be able to react according to the new data. Unfortunately, the GL model presented in Section 1 requires a body model that can return the level of White Blood Cells (WBC), e.g., neutrophils and platelets, which is hard to build. Moreover, even if such patients model exists, this GL model does not account for other patient features and thus will not provide personalization.

In this article we rely on the fact that responsiveness to the treatment with drugs depends on the concentration of the drug in patient's blood (PK-PD relationship) correlated with patients features, drug dose and intake interval. In Section 2.2 we introduce the algorithm that is able to individually compute the drug concentration in patients blood aimed to complement a more slow and expensive TDM. This algorithm is trained with data (a library) that include real

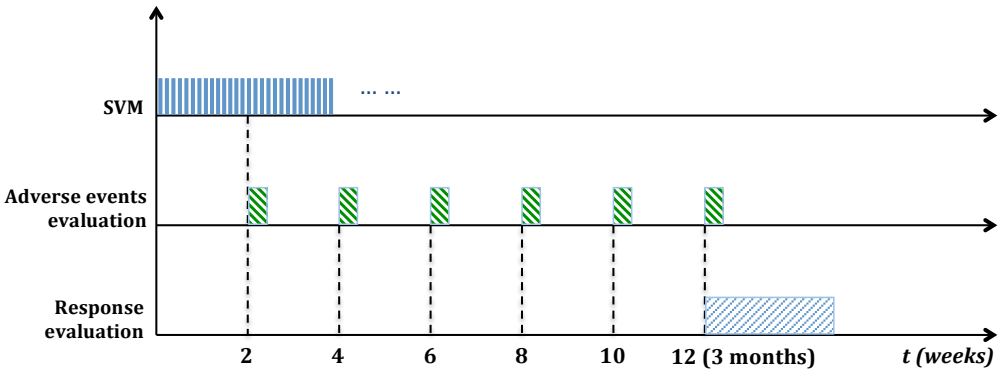

Figure 3: Time correlation of the procedures

measured values of *imatinib* concentration obtained at time of PK studies with corresponding patients parameters [21]. The collection of such library data is time consuming and thus expensive. However, once the data are available, they can be used by the SVM algorithm with no additional cost. This way it opens new possibility in patient treatment procedures. However, in case of CML the TDM procedure can not be used as a standalone approach and should not contradict with the existing approved GL presented in Section 1.

When applying the SVM algorithm each following drug concentration must account for the previously calculated residual concentration in the blood (e.g. at 24 hours). Therefore, we need to keep track of the concentration values after each dose intake and thus perform SVM computation every day even though the drug concentration may not undergo drastic changes in one day. The more frequent application of the TDM or its replacement with an algorithm, can improve the quality of treatment by preventing adverse events and lack of response caused by the suboptimal dosage. It is also possible to appropriately adjust the dose when the previous intake was missed out. The dose adjustment can be finer (e.g. 50 mg) compared to the one of the approved *imatinib* GL (100 mg). We only need to make sure that the dose and intake interval values adapted according to the algorithmic computation corrected from time-to-time with a real measurement lays within the ranges allowed by the approved GL.

This way we can extract the following rules that need to be checked in the *check conformity with GL rules* block of Figure 2:

1. $dose >= 300$ - the minimum dose assigned should be not less than 300 mg;
2. $dose <= 800$ - the maximal dose assigned should be not more than 800 mg;
3. $dose >= 800 \rightarrow p:=p/2, dose:=dose/2$ - when the dose is equal (or greater) than 800 mg it should be administrated in two shots.

According to [16] the response evaluation is performed every 3 months. The formal model of response evaluation

is presented in [15]. The check for quantitative adverse events is usually performed every 2 weeks, especially at the beginning of the treatment. The time correlation of the response evaluation, adverse event check up and drug concentration prediction using the SVM algorithm are presented on Figure 3. On this figure, we show that the SVM algorithm is performed every day so the model follows each step of drug administration.

2.2 Parametrized SVM (ParaSVM) algorithm

In this section we present a patient body (physical) model that is able to predict the drug concentration values over time required by the TDM approach to modify the behaviour of the cyber part of our medical system. It is an analytical model that captures the shape information of the curve that is essential for curve calibration needed in the *a posteriori* drug dose adaptation. Newly measured concentration values or the residual concentration of the drug in the blood after previous drug intakes (e.g. at 24 hours) can be used for curve calibration and accounting for consequent multiple doses between 2 consequent real measurement.

The transformation flow of the new Parametrised SVM (ParaSVM) model is shown on the lower row of Figure 4. Unlike applying the SVM algorithm to predict the concentration values directly after removing the outliers using the RANSAC algorithm as shown in the first row, here the RANSAC algorithm is utilised to compute the parameter database, or parameter library, used by ParaSVM to construct the DCT curve for each patient based on their personal features. To remove the outliers and keep enough data samples to build the subset at the same time for each patient we take into account all his/her samples in addition to the randomly-selected samples from the rest of the patients to build the subset. We use the common basis functions $\beta^j = \{t^{-2}, \log(t), 1-e^{-t}\}$, respecting the shape of DCT curve obtained from the PK method [21], where t stands for time [24]. Therefore, the target is to obtain the parameters y for the weights of β:

$$f_{concentration} = y \cdot \beta = \begin{bmatrix} y^1 & y^2 & y^3 \end{bmatrix} \begin{bmatrix} \beta^1 \\ \beta^2 \\ \beta^3 \end{bmatrix}. \tag{1}$$

These parameters $\{y^1, y^2, y^3\}$ together with patients' features form the Parameter Library being used as the training data. With one given measured concentration value, in case of *a posteriori* adaption, or a value of the residual concentration after the previous dose intake, for multiple-dose cases, the curve parameters then can be adjusted, this way allowing to build the personalised DCT curve. Here, we consider all patients' data are obtained during the steady state (after several doses).

Further, instead of predicting the drug concentration values as in [22, 25], we apply the SVM algorithm to learn the mathematic relationship between the parameters of the basis functions and then predict the parameter values of the DCT curve for a new patient in the testing dataset. Therefore, the main difference is the inclusion of minimising the

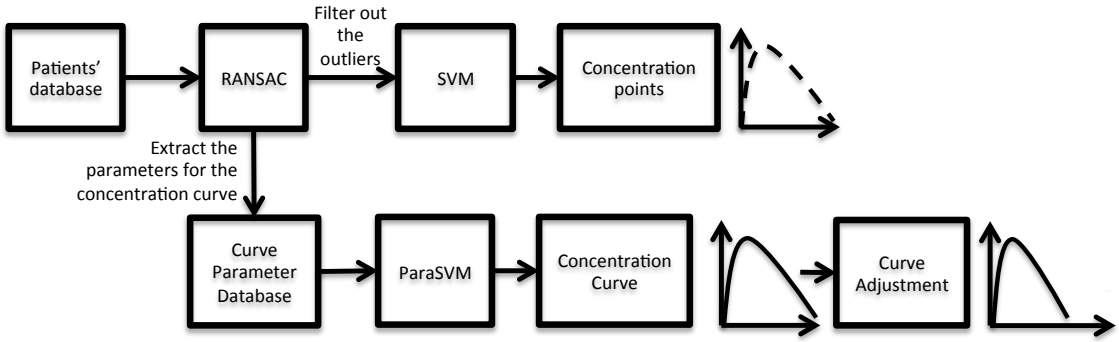

Figure 4: ParaSVM algorithms flow for building DCT Curves

objective function which considers a combined difference between the predicted parameter values and the ones in the parameter library (training dataset).

In case of modelling N patient samples, the form of patient samples becomes $(x_i, y_i^1, \cdots, y_i^j, \cdots, y_i^{NP})$, where i is the ID of a sample $i \in \{1, 2 \cdots, N\}$, y_i^j denotes the j-th parameter value of this patient, and NP is the number of parameters. The goal is to find NP linear functions $f^j(x) = w^j \cdot \phi^j(x) + b_j$ to describe the relationship between the dataset points and estimate the parameter value y according to a new input dataset. For that we need to minimize the following modified objective function:

$$\min_{w,b} \frac{1}{2}||w||^2 + C_0 \sum_{j=1}^{NP} \sum_{i=1}^{N} [y_i^j - w^j \cdot \phi^j(x_i) - b^j]^2, \tag{2}$$

where H takes into account the combined difference of all three predicted values plus the ones in the parameter library. Note that this objective function has Root of Sum of Square (RSS) fitting error and a regularization term, which is also a standard procedure for the training of Multi Layer Perceptron (MLP) and is related to ridge regression [26, 27]. Applying Lagrangian analysis to solve the optimization problem of objective function, we obtain w as:

$$w^j = \sum_{i=1}^{N} \alpha_i^j \phi^j(x_i). \tag{3}$$

Combining Equ. (2) and (3), we can obtain a linear system to:

$$\begin{bmatrix} \mathbf{K}^j + \frac{1}{C_0} I & 1 \\ 1^T & 0 \end{bmatrix} \begin{bmatrix} \alpha^j \\ b^j \end{bmatrix} = \begin{bmatrix} y^j \\ 0 \end{bmatrix}, \tag{4}$$

where each entry of the kernel matrix $\mathbf{K}^j$ is defined to be $K_{ab}^j = \phi^j(x_a)^T \phi^j(x_b)$. A Gaussian Kernel is applied in a similar way as in [22]. Therefore, the prediction function becomes: $f^j(x) = \sum_{i=1}^{N} \alpha_i \mathbf{K}^j(x_i, x) + b^j$.

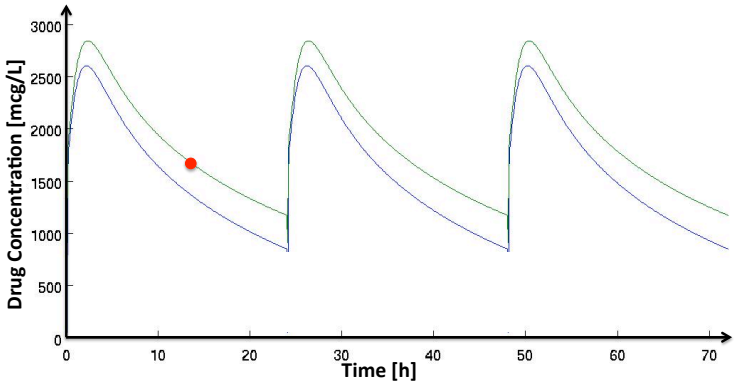

Figure 5: An Example of Parametrically Refined DCT curves; the lower DCT curve is computed by ParaSVM approach; the upper DCT curve is adjusted with one measurement (red dot)

2.2.1 The a posteriori adaptation

In the *a posteriori* dose adaptation, we refine the predicted drug concentration curve computed by ParaSVM after obtaining one real measured concentration value of a patient following these constraints:

- The modified DCT curve has to pass through the given measured concentration value;
- After a dose administration, the concentration value should start monotonically growing: $\frac{\partial g_{concentration}}{\partial t}|_{t=T_{bp}} > 0$, where T_{bp} is any time point before the peak value.
- After several hours, it reaches the peak value and starts to decrease: $\frac{\partial g_{concentration}}{\partial t}|_{t=T_{ap}} < 0$, where T_{ap} is any time point after the peak value, i.e. we set it as $T_{ap} = 24h$.
- Taking into consideration the residual concentration value, the difference between the estimated starting concentration value ($t = 0$) and the ending one (i.e. here $t = 24h$, when *imatinib* needs to be administered once a day) should be within a certain threshold, i.e. $< 50mcg/L$. $|g^{t=0}_{concentration} - g^{t=24}_{concentration}| < \text{TH}$.
- The concentration curve whose shape is the most similar compared with the one predicted from ParaSVM will be chosen: $\min_{g_r} \sum_{j=0,\cdots,N_s} (g_r^{t=j} - g^{t=j})^2$, where $g^{t=j}$ stands for the concentration value at time j estimated using the predicted parameters and $g_r^{t=j}$ denotes the one in the refined curve. The set of parameters y corresponding to the best g_r are selected.

Figure 5 shows an example of parametrically refined drug concentration curves satisfying the above constraints.

2.2.2 Library adaptation

Each new measured concentration value with patient's features is added to the training library, and, meanwhile, a least 'relevant' sample in the library is removed for the sake of resource. The 'relevance' here is measured using a vector storing the Curve Features: $CF = \{C_0, T_p, C_p, T_t, C_t\}$, where C_0 denotes the residual concentration value from the previous dose, C_p is the peak concentration value with T_p its corresponding time (within 24h), C_t is the trough concentration value with $T_t = 24h$ corresponding time. Those curve features are trivial to obtain with paraSVM. Thus, a *K*NN approach, based on Euclidean distance, is applied to measure the relevance between two curves and selects only K most relevant samples as the library.

$$\text{Distance}(CF_{\text{new}}, CF_{\text{lib}_j}) = \sqrt{\sum_{i=1}^{5} (cf^i_{\text{new}} - cf^i_{\text{lib}_j})^2}. \tag{5}$$

3 Results and discussions

In this section we introduce the actual control flow (Section 3.1) and discuss possible implementations of the control flow while introducing the highlights of implemented TIMES code generator for *icycom* platform (Section 3.2).

3.1 The control flow

In Section 1 we have briefly described an approach to formal representation of a medical GLs showcasing it with an example of modelling a small part of the *imatinib* dose adjustment guideline. In Section 2.1 we have described our approach to the feedback loop realization, where the body reaction model is represented by the Parametric SVM algorithm described in details in Section 2.2. In this section we present an executable control flow that refines the one presented in Section 2.1 combining body reaction model, GL rules and results of the PK-PD studies.

The control flow that can be executed on an embedded electronic device is presented in Figure 6. Following the general feedback loop control flow of Figure 2 it is divided into *initial decision-making*, *drug delivery*, *body reaction*, *decision making* and *check for conformity with GL rules* blocks. The flow has fixed values for the intake intervals: one day p_2 or half a day p_3; while the dose can be changed with more fine than in currently applied *imatinib* GL values defined by the parameter Δ_{dose} (e.g., 50 mg).

The flow starts with the initialisation of the peripheral of an embedded device. Here the daily dose is also initialized with a personalized values (*d_dose:=init_dose*) after being computed externally using the SVM-based algorithm taking into account all the patient features. The period p is first set to one day ($p:=p_2$) in case the initial dose value is less than 800 mg, while the actual dose to be administrated is set to the daily dose (*dose:=d_dose*).

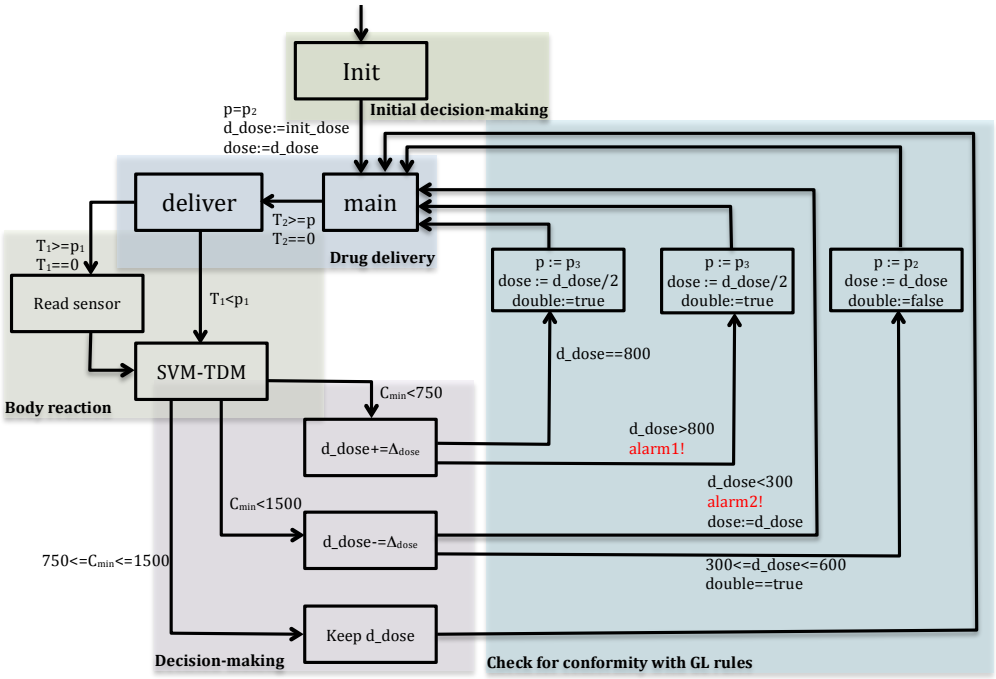

Figure 6: Synthesized control flow

The *drug delivery* block is composed of *main* and *deliver* blocks. Here we will stay in the *main* block until clock T_2 is less than intake period p. Once the condition $T_2 >= p$ holds we transit to the *deliver* block, where either the drug is automatically administrated, in case of an implantable device or a reminder is given to a user. Immediately after each drug administration the body reaction must be evaluated where SVM-based prediction of the drug concentration performed in the SVM-TDM block. The algorithmic computation of the body reaction is fired every time after the drug administration (every day or half a day), while the clinical measurement of the drug concentration is performed every period p_1 (e.g. every 2 weeks). The value of the real measurement is further used to correct the SVM-TDM algorithm.

In the *decision-making* block the decision about increasing, decreasing or keeping the dose is taken. In the present example we assume that the trough concentration value C_{min} (at 24 or 12 hours after the last intake) should lay within the 750:1500 $\mu g/l$ range ($750\ \mu g/l <= C_{min} <= 1500\ \mu g/l$) [28]. If $C_{min} < 750\ \mu g/l$ the daily dose will be increased ($d\_dose\ += \Delta_{dose}$) and decreased in case $C_{min} > 1500\ \mu g/l$.

In the next block we check if the value of the daily dose and intake interval are conformed with the rules defined in Section 2.1. For example, when $d\_dose > 800\ mg$ the period must be set to half day interval and the dose divided by 2 for each drug administration. The alarm (*alarm1!*) will be also generated since the maximal dose defined by *imatinib*

GL is exceeded. The period will be set back to one day next time only after the *d_dose* falls down to 600 mg. An alarm (*alarm2!*) will be generated when we reach the minimal dose value defined in *imatinib* GL (300 mg).

3.2 Code synthesis

The model presented above is directly implementable with TAT in TIMES toolbox and can be synthesised by newly developed code generator for *icycom* platform that we have developed as a proof of concept. In the synthesised code model tasks such as *delivery*, *read sensor*, *SVM-TDM* and time tick handling operations represent the class of drivers. There might be several variations of the implementation of the control flow mainly concerning small operations with intake interval and dose values when the dose must be administrated in 2 shots. They can be implemented either as functional tasks associated with corresponding locations, or as assignments of the corresponding transitions. The SVM-TDM algorithm is modelled as a location associated task, a driver. The *icycom* code generator is based on platform independent code generator of TIMES with several features added. Further we describe some details of the new code generator implementation.

Each embedded platform needs to be initialized. Therefore, we have added an obligatory *Init* location with an associated initialization task that can be modified and adapted for any target platform independently from the control flow. The *Init* function is present in our control flow depicted on Figure 6. It has to run before all other tasks and be executed only once. In this task, the libraries and the initializing procedures of the optional peripheries are included.

We have reworked the implementation of the variables passing through the network of automata in the new code generator, since we often use them in our case studies, e.g. when modifying the dose and intake interval values. The global variables defined in TAT can be evaluated and set in the guard and assignments of the automata network, as well as in the task body. When the task is added to the scheduling queue by the control flow, the value of the global variable is copied to a local variable, this way, variable modification within task execution does not affect the value of the global one until the task is completed.

If the maximum number of the identical tasks in the queue is greater than 1, then we implement a FIFO of local variables. The task body first added to the scheduling queue will take the first local variable. If the second one is added to the queue before the first task of the same type is finished, the current, not yet modified, value of the global variable will be stored at the second position of local variables FIFO. The third task will store the current value of the global variable at the third position and so on. When the first task finishes, all the values will be moved by one position up.

An example of time diagram for global variable passing is presented on Figure 7. Let us assume that we have a model with two tasks (except *init*) and a global variable *dose*. One task named *dose_adj* adjusts the value of the global variable, and the other task named *dose_deliver* delivers the drug based on the value of the global variable.

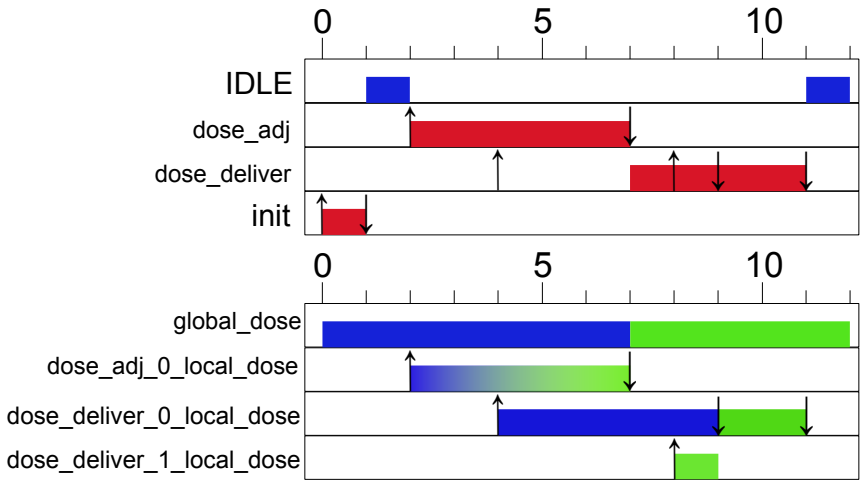

Figure 7: An example for the global variable handling

The maximum number of the *dose_deliver* task in the task queue is two and the maximum number of the *dose_adj* task is one. The behavior of tasks is shown in the upper graph while the changes of the global and local variables are shown in the lower one in the Figure 7. The *dose_adj* task is added to the queue at cycle 2 and keeps running for five cycles, until cycle 7. When it is added to the task queue, the value of the *global_dose* is copied to the local variable, *dose_adj_0_local_dose*. The local variable changes during the *dose_adj* task execution. After the *dose_adj* task finishes, the value of *dose_adj_0_local_dose* is copied to the global one, so the *global_dose* is changed at the 7th cycle. The *dose_deliver* task is added to the queue in the 4th cycle. Therefore, the global variable copied to the *dose_deliver_0_local_dose* variable still has the unchanged initial one. Thus, the first *dose_deliver* task running during the 8th and 9th cycle will use the initial value of *dose*. This way of handling global variables passing is valid only for the First Come First Serve scheduling policy. For other scheduling policy, since the order of tasks execution might change dynamically, the scheme will be more complicated. It is essential to be aware of the global variables passing mechanism when building new models.

TIMES tool is a model checking environment that allows the automatic verification of real-time properties of the control flows with associated tasks modelled with TAT. In terms of time constrains, we can introduce strict deadline for tasks execution, which is essential especially for delivery control flows of drug with fast PK characteristics. For example, anaesthetics are the drugs that need to be delivered every 10 seconds, which raises the requirements to the algorithm as to a realtime system. This implies that the whole processing of the data and decision-making should happen within the administration period (e.g. 10 seconds) and before the next dose delivery. TIMES provides schedulability analysis technique, which allows verification that all the tasks triggered by the control flow will finish they

computation within 10 seconds. The synthesiser that generates the code that preserve timing properties of the verified model is an important step towards a safety-assured development of the medical control-flow software.

4 Conclusion

In this paper we have presented the anticancer drug *imatinib* dose adjustment control flow modeled using TAT with an implemented *a posteriori* drug dose adaptation. The feedback loop of the control flow is based on the ParaSVM algorithm model of the body reaction that is able to build the DCT curves in a personalized manner complemented with occasional real TDM measurements for curve calibration. The drug concentration values are further used to make decisions regarding the dose modification. We also show that the control flow conforms with the currently approved *imatinib* protocol. The K-Nearest Neighbor (KNN) method is applied to keep the library size constant. We also presented a new TIMES code generator of the executable code for the *icycom* embedded platform.

5 List of abbreviations

- TDM - Therapeutic Drug Monitoring
- SVM algorithm - Support Vector Machine algorithm
- GLs - Guidelines
- TA - Timed Automata
- TAT - Timed Automata extended with Tasks
- Ph+ CML - Philadelphia positive Chronic Myelogenous Leukemia
- GISTs - Gastrointestinal Stromal Tumors
- GPCA - Generic Patient Controlled Analgesic
- PK - Pharmacokinetics
- ParaSVM - parametrized SVM
- WBC - White Blood Cells
- PD - Pharmacodynamics
- RANSAC algorithm - RANdom SAmple Consensus algorithm

- DCT curve - Drug Concentration-Time curve
- RSS - Root of Sum of Square
- MLP - Multi Layer Perceptron
- TH - THreshold
- FIFO - First Input First Output

6 Competing interests

The author(s) declare that they have no competing interests.

7 Authors' contributions

AS is the author of the concept of modeling the medical system as control system with feedback loop who has also made a major advancement in formal medical guidelines modeling. She has made most of the paper writing, editing and formatting. Together with GdM she was also leading the work of WY on developing and advancing the SVM-based algorithm for in blood drug concentration prediction. She also leaded the work of DS on development of new TIMES code synthesizer. NW and VG has provided us with the sample library of real drug concentration measurements required by the SVM algorithm. They also critically revised the paper for important medical content. All of the authors have been involved in drafting the manuscript and have given final approval of the version to be published.

Acknowledgements

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 regrading the project.

References

1. Johnson PD, Tu S, Booth N, Sugden B, Purves IN: **Using scenarios in chronic disease management guidelines for primary care**. In *in Proceedings of American Medical Informatics Association(AMIA) Symposium*, PubMed Central PMCID 2000:389–93.

2. Tu SW, Musen MA: **Modeling data and knowledge in the EON guideline architecture**. *Medinfo* 2001, **10**:280–284.

3. Boxwala A, Peleg M, Tu S, Ogunyemi O, Zeng Q, Wang D, Patel V, Greenes R, Shortliffe E: **GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines**. *Journal of Biomedical Informatics* 2004, **37**(3):147–161.

4. Sutton DR, Taylor P, Earle K: **Evaluation of PROforma as a language for implementing medical guidelines in a practical context**. *BMC Medical Informatics and Decision Making* 2006, **6**:20.

5. Tu SW, Campbell JR, Glasgow J, Nyman MA, McClure R, McClay J, Parker C, Hrabak KM, Berg D, Weida T, Mansfield JG, Musen MA, Abarbanel RM: **The SAGE Guideline Model: Achievements and Overview**. *Journal of the American Medical Informatics Association* 2007, **14**(5):589–598.

6. Terenziani P, Montani S, Bottrighi A, Torchio M, Molino G, Correndo G: **The GLARE approach to clinical guidelines: main features**. *Studies in Health Technology and Informatics* 2004, **101**.

7. Duftschmid G, Miksch S, Shahar Y, Johnson P: **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)* 1998:4–1.

8. Simalatsar A, De Micheli G: **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* 2012:5078– 5081.

9. Simalatsar A, You W, Gotta V, Widmer N, Micheli GD: **Representation of Medical Guidelines with a Computer Interpretable Model**. *International Journal on Artificial Intelligence Tools, Special Issue* 2014.

10. Alur R, Dill DL: **A theory of timed automata**. *Theoretical Computer Science* 1994, **126**(2):183–235.

11. Amnell T, Fersman E, Pettersson P, Sun H, Yi W: **Code Synthesis for Timed Automata**. *Nord. J. Comput.* 2002, **9**(4):269–300.

12. Behrmann G, David R, Larsen KG: **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 2004:200–236.

13. [TIMES. http://www.timestool.com/l].

14. [Imatinib. http://www.glivec.com/files/Glivec-400mg-coated.pdf].

15. Simalatsar A, De Micheli G: **Medical Guidelines Reconciling Medical Software and Electronic Devices: Imatinib Case-study**. In *International Conference on BioInformatics and BioEngineering* 2012.

16. Baccarani M, Castagnetti F, Gugliotta G, Palandri F, Soverini S: **Response definitions and European Leukemianet Management recommendations**. *Best Pract Res Clin Haematol* 2009, **22**(3):331–41.

17. Kim B, Ayoub A, Sokolsky O, Lee I, Jones P, Zhang Y, Jetley R: **Safety-assured development of the GPCA infusion pump software**. In *Proceedings of the ninth ACM international conference on Embedded software*, EMSOFT '11, New York, NY, USA: ACM 2011:155–164.

18. Shafer S: **STANPUMP User's Manual**. In *Stanford University*, Stanford, CA 1996.

19. Brown RF: **Compartmental System Analysis: State of the Art**. *IEEE Transactions on Biomedical Engineering* 1980, **27**:1–11.

20. Blau G, Orcun S: **A Bayesian Pharmacometric Approach for Personalized Medicine - A Proof of Concept Study With Simulation Data**. In *Proceedings of the 2009 Winter Simulation Conference* 2009:1969–76.

21. Widmer N, Decosterd LA, Csajka C, Leyvraz S, Duchosal MA, Rosselet A, Rochat B, Eap CB, Henry H, Biollaz J, et al: **Population pharmacokinetics of imatinib and the role of alpha-acid glycoprotein.** *British Journal of Clinical Pharmacology* 2006, **62**:97–112.

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

23. You W, Simalatsar A, Widmer N, Micheli GD: **A drug administration decision support system,**. In *Bioinformatics and Biomedicine Workshops, 2012 IEEE International Conference* 2012:122–9.

24. You W, Simalatsar A, De Micheli G: **RANSAC-based enhancement in drug concentration prediction using support vector machine**. In *Proceedings of the International Workshop on Innovative Simulation for Healthcare (IWISH)* 2012:21–5.

25. Y W, Widmer N, De Micheli G: **Personalized modeling for drug concentration prediction using Support Vector Machine**. In *Biomedical Engineering and Informatics (BMEI), 2011 4th International Conference on*, *Volume 3* 2011:1505 –1509.

26. Golub G, Van Loan CF: *Matrix Computations*. John Hopkins University Press 1989.

27. Brabanter JD: **LS-SVM Regression Modelling and its Applications**. *PhD thesis* 2004, :25–28.

28. Gotta V, Widmer N, Decosterd L, Csajka C, Duchosal M, Chalandon Y, Heim D, Gregor M, Buclin T: **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 2010:285.