Luigia Petre and Emil Sekerinski

From Action System to Distributed Systems: The Refinement Approach

Contents

Chapter 1

Action Systems for Pharmacokinetic Modeling

M.M. Bonsangue

LIACS, Leiden University, The Netherlands

M. Helvensteijn

LIACS, Leiden University, The Netherlands

J.N. Kok

LIACS, Leiden University, The Netherlands

N. Kokash

LIACS, Leiden University, The Netherlands

Kaisa, in this paper we use "your" action systems in a field you would have never predicted: pharmacokinetics. We miss you!

1.1 Introduction

Pharmacokinetics is a branch of pharmacology that studies processes caused by substances (such as drugs, hormones, nutrients or toxins) administered to a living organism, from the moment of substance administration

to the point of its complete elimination from the body. Pharmacokinetics describes how the body affects a specific drug after administration through the mechanisms of absorption and distribution, as well as the chemical changes of the substance in the body, and the effects and routes of its excretion.

Compartment models are typically used as mathematical models to describe level changes of the drug concentration in our bodies [18]. Compartments are used to model drug concentration in body tissues by using differential equations for absorption, distribution and elimination rates. Compartment models exist for many types of drugs, absorption methods (e.g. oral, intramuscular, intravenous), and elimination methods.

Compartment models are continuous-value models, based on systems of ordinary differential equations that can be analyzed by examining a graph with compartments as node and in-flow/out-flow rates labeling the arcs. However, pharmacokinetic models are not purely continuous. They also use a finite state logic to decide, for example, when absorption stops, when elimination starts, or when a process is triggered by cell death or division. As such they involve continuous states and dynamics, as well as some discrete logic corresponding to discrete states and dynamics. Because existing pharmacokinetic models often represent a tradeoff among accurately describing the data, having confidence in the results, and mathematical tractability, the discrete logic of the system is often ignored.

In this paper we use hybrid action systems as a model of pharmacokinetics. Action systems were originally proposed as a formalism for developing parallel and distributed systems [2]. They are based on a predicate transformer semantics for discrete computation, where parallel composition is described by interleaving of atomic actions. In hybrid action systems, computations may also continuously evolve over time [16, 17], and parallel composition is defined by interleaving discrete actions and combining continuous ones.

While it appears useful to be able to describe and reason about properties of pharmacokinetic models using hybrid systems, we are not aware of any existing consistent attempts to do so. In the last decade, there has been an increasing interest in quantitative and hybrid models of computation based on systems of ordinary differential equations. However, all models that we are aware of are concentrating on biological, bio-chemical or bio-medical processes. Examples of successful approaches in these areas include several process algebraic languages [14, 12, 15, 8, 6, 7]. In this paper we take a novel alternative approach and use hybrid action systems for a compositional construction of pharmacokinetic models. We concentrate on the modelling aspects, but our framework allows properties of the systems to be proven formally within the refinement calculus [3, 17]. For example, in pharmacokinetics we deal with models for the study of a drug's pharmacological effect on the body. Understanding what happens between the administration of a drug and the body's response could help in recommending a proper dosing regimen for that drug (how much, how often, under what assumptions) with regard to a population, a subpopulation or an individual patient. Drug concentrations must be kept

high enough to produce a desirable response, but not so high that they produce toxic effects. The problem is that current models are so complex that it is difficult to extract useful predictions from them. However, since the magnitude of an effect is proportional to the concentration of the drug, our action system model could be used to prove that concentration always remain below the toxic level.

1.2 Actions and action systems

Let *Var* be a countable set of *variables* and assume that each variable in *Var* is associated with a nonempty set of *values*. A *state* is a function mapping variables to values in their respective associated set of values. We denote by $true$ the predicate on *Var* which holds for every state, and by $false$ the predicate on *Var* which holds for no state. Given a predicate $\mathcal{P}$ over *Var*, a list of variables x and a list of values v, we denote by $\mathcal{P}[x/v]$ the predicate that holds for those states s such that $\mathcal{P}$ holds for $s[x/v]$, where $s[x/v]$ is the state that maps the variables x to the values v, but otherwise behaves as s.

A *conjunctive predicate transformer* is a function π mapping predicates to predicates such that, for every nonempty index set I,

$$\pi\left(\bigwedge\{\mathcal{P}_i \mid i \in I\}\right) \quad = \quad \bigwedge\{\pi(\mathcal{P}_i) \mid i \in I\}\,.$$

Conjunctive predicate transformers form the semantic domains for a class of statements, called *actions*, interpreted by means of a *weakest precondition* semantics [11]. These should be considered specification statements rather than actual programs, as they need not be strict, disjunctive, or preserving of directed disjunctions [4].

More syntactically we consider *actions* denoting conjunctive predicate transformers. We define *ordinary actions* by the grammar

$$a \quad ::= \quad abort \mid skip \mid x := v \mid b \rightarrow a \mid p \mid a_1 \,;\, a_2 \mid [\!]_I a_i\,.$$

Here, x is a list of variables, v is a list of values (possibly resulting from the evaluation of a list of expressions), b is a predicate over *Var*, p is a procedure name, and I is an index set ranged over by i. Intuitively, '$abort$' is the action which models unwanted or disallowed behaviour, '$skip$' is a stuttering action (i.e. not changing the state), '$x{:=}v$' represents multiple assignment, '$b \rightarrow a$' is a guarded action that executes a only when the guard b holds, 'p' is a procedure call, '$a_1 \,;\, a_2$' is the sequential composition of two actions 'a_1' and 'a_2', and '$[\!]_I a_i$' is the nondeterministic choice among actions 'a_i' for $i \in I$.

A procedure declaration $p = P$ consists of a header p and an action P—the body of the procedure. Given a declaration for each procedure, we define the

weakest precondition semantics of the above language in a standard way [11, 4], as the *least* conjunctive predicate transformer wp such that, for any predicate $\mathcal{P}$,

$$\begin{array}{lcllcl} wp(abort,\mathcal{P}) & = & false & wp(skip,\mathcal{P}) & = & \mathcal{P} \\ wp(x := v,\mathcal{P}) & = & \mathcal{P}[x/v] & wp(b \rightarrow a,\mathcal{P}) & = & b \implies wp(a,\mathcal{P}) \\ wp(p,\mathcal{P}) & = & wp(P,\mathcal{P}) & wp(a_1\,;a_2,\mathcal{P}) & = & wp(a_1,wp(a_2,\mathcal{P})) \\ wp([\!]_I a_i,\mathcal{P}) & = & \forall i \in I.wp(a_i,\mathcal{P})\,, & & & \end{array}$$

where P is the action denoting the body of the procedure p. The *greatest* conjunctive predicate transformer satisfying the above is the *weakest liberal precondition* $wlp(a,\mathcal{P})$ of an action a with respect to the post-condition $\mathcal{P}$. Various form of iterations and other simple statements can be encoded in the above language. For example, assertions $\{b\}$ can be encoded as $b \rightarrow skip \,[\!]\, \neg b \rightarrow abort$: if the condition b does not hold in a given state, then the assertion aborts, otherwise has no effect. The details of the definition of the above functions is studied elsewhere [5].

Further on we will need the following notions. An action a is *enabled* in a given state if its *guard*

$$gd(a) \quad = \quad \neg\, wp(a,false)$$

holds in that state. For example, $gd(b \rightarrow x := v) = b$. An action a *terminates* in a given state if $t(a) = wp(a,true)$ holds in that state. An action a_1 *cannot enable* another action a_2 if

$$t(a_1) = true \quad \text{ and } \quad \neg\, gd(a_2) \implies wp(a_1, \neg\, gd(a_2))\,.$$

Moreover, a_1 *cannot disable* a_2 whenever

$$t(a_1) = true \quad \text{ and } \quad gd(a_2) \implies wp(a_1, gd(a_2))\,.$$

Actions that capture continuous-time dynamics are called *differential actions* [17]. They act not only on ordinary variables, but also on some *evolution variables* in $EVar$. Evolution variables describe the observation of an evolution but not its relation with respect to time. As such, they take values in the set of real numbers $\mathbb{R}$. A differential action, written as $e{:\rightarrow}\, d$, describes the evolution in time according to the predicate d of both evolution and ordinary variables from their initial value (as given by a state) to the values they reach when the guard e does not hold. The *evolution guard* e is a predicate over Var and a list $X = x_1 \cdots x_n$ of evolution variables. In this paper we consider the predicate d to be a partially defined system of differential equations of the form $\dot{X} = F(X)$, where $\dot{X}$ is a syntactic variant used to denote the component-wise first derivative of the variables in X.

A continuous function $f{:}\mathbb{R} \rightarrow \mathbb{R}^n$ with a continuous first derivative $\dot{f}$ is a *solution* for $e{:\rightarrow}\, d$, denoted by $SF(f, e{:\rightarrow}\, d)$, if $f(0) = X$ (the current value

of the variables) and for every positive $r \in \mathbb{R}$, if the guard e is *true* then it satisfies the system of differential equations d. More formally

$$SF(f, e{:\rightarrow}\, d) \;\equiv\; f(0) = X \wedge \forall r \in [0, \infty).\ (e \Rightarrow d)[f(r)/X, \dot{f}(r)/\dot{X}]$$

The first point in time, if any, when the guard e does not hold, give the *duration* $\Delta(f, e)$ of a differential action. In other words, $\Delta(f, e) = inf\{r \in [0, \infty) \mid \neg e[f(r)/X]\}$.

For a postcondition $\mathcal{P}$ over X and *Var*, the *weakest precondition* of a differential action is the smallest set of states (assignments to X and *Var*) for which the evolution of $e{:\rightarrow}\, d$ is guaranteed to terminate in a state satisfying $\mathcal{P}$. More formally,

$$\begin{aligned} wp(e{:\rightarrow}\, d, \mathcal{P}) \equiv \forall f.(\ & SF(f, e{:\rightarrow}\, d) \wedge \Delta(f, e) > 0 \ \Rightarrow \\ & \Delta(f, e) < \infty \wedge \mathcal{P}[f(\Delta(f, e))/X]\ ) \end{aligned}$$

As for ordinary action, we say that a differential action $e{:\rightarrow}\, d$ is *enabled* if $gd(e{:\rightarrow}\, d) = \neg\, wp(e{:\rightarrow}\, d, false)$ holds, and that it *terminates* if $t(e{:\rightarrow}\, d) = wp(e{:\rightarrow}\, d, true)$ holds. Enabledness holds in states where there is evolution (i.e. $gd(e{:\rightarrow}\, d) \Rightarrow e$) and termination holds exactly when all evolutions terminate. Since $wp(e{:\rightarrow}\, d, \mathcal{P}) = wp(gd(e{:\rightarrow}\, d){:\rightarrow}\, d, \mathcal{P})$ [17], we will only consider *stutter free* differential actions [16], that is, differential actions $e{:\rightarrow}\, d$ that are enabled exactly when there is evolution (i.e. $gd(e{:\rightarrow}\, d) = e$). Finally, the weakest precondition of a differential action is a conjunctive predicate transformer. These and other interesting results on differential actions are studied in, e.g. [17].

1.2.1 Action systems

Action systems are a formalism for developing parallel and distributed systems originally proposed in [2]. We will consider here *hybrid action system* [17]. They consist of a set of ordinary and differential actions operating on local and global (evolution) variables. First, the variables are created and initialized. Then, repeatedly, enabled actions are chosen and executed. Actions operating on disjoint sets of variables can be executed in any order. The execution terminates if no action is enabled, otherwise it continues infinitely.

Syntactically, an action system $\mathcal{A}$ is a statement of the form

$$\begin{array}{lcll} \mathcal{A} & = & |[& \textbf{var} \quad Y^* := V\ ;\ X := U \\ & & & \textbf{proc} \quad p_1 = P_1\ ;\ \ldots\ ;\ p_m = P_m \\ & & & \textbf{do}\ A_1 \,[\!]\, \ldots \,[\!]\, A_n\ \textbf{od} \\ & &]| & : Z \end{array}$$

An action system provides a declaration section for variables and one for procedures. Here Y is a list of *global (evolution) variables*, marked with an asterisk $*$, that can be used locally by $\mathcal{A}$ and also by other action systems when put in parallel with $\mathcal{A}$. The (evolution) variables in the list X are *local*

to $\mathcal{A}$. The global variables Y get initial values componentwise from the list V and the local variables x get initial values componentwise from the list U. Finally, Z is the list of imported variables, i.e. global variables declared in action systems that can be put in parallel with $\mathcal{A}$. We assume that X, Y and Z are disjoint.

A *procedure* declared as $p_i = P_i$ is *local* and can be called only by the ordinary actions of $\mathcal{A}$ which can thus enable or disable the body P_i. *Actions* of the action system $\mathcal{A}$ are the actions $A_1, \ldots A_n$ and the bodies of all procedures declared in $\mathcal{A}$. Each action A_i can be either an ordinary action a as defined above, or a differential action $e{:\rightarrow}\, d$. All actions of $\mathcal{A}$ can refer to (evolving) variables which are in X, Y or Z. Actions are atomic, meaning that if an action A_k is chosen for execution, then it is executed to completion.

Action systems are models of reactive systems. The *behaviour* of an action system is described by the set of all its computations. A *computation* here is a finite or infinite sequence of states (i.e. maps of *global* variables to values), without finite repetition (i.e. no finite stuttering), possibly terminating with a special symbol to denote abortion [1].

To model the dynamics of a system consisting of several reactive components, action systems are equipped with a parallel composition operation: global variables are merged together, local variables are kept separate, and the imported variables will consist of all variables imported by one component that are not declared as global in the other. Procedure are local, and thus kept separate, whereas actions are combined non-deterministically, thus modelling parallelism by interleaving [1].

1.2.2 Hybrid Action Systems

While parallel composition of action systems works fine with ordinary actions that have no duration in time, interleaving of differential actions does not model true concurrent evolution. Parallel composition of differential actions can be defined as the linear composition '$\oplus$' of two partially defined functions [16]: their values are added together on their common domain, and remain unchanged on the remaining domains.

$$
\begin{array}{lcl}
e_1{:\rightarrow}\, \dot{X} = F(X) \;\oplus\; e_2{:\rightarrow}\, \dot{X} = G(X) & = & \quad e_1 \wedge \;\; e_2{:\rightarrow}\, \dot{X} = F(X) + G(X) \\
& [\!] & \quad e_1 \wedge \neg e_2{:\rightarrow}\, \dot{X} = F(X) \\
& [\!] & \neg e_1 \wedge \;\; e_2{:\rightarrow}\, \dot{X} = G(X)
\end{array}
$$

We want the dynamics of a hybrid action system to be an alternation between ordinary actions and differential ones, so that in a parallel composition we can linearly compose the continuous-time differential actions and interleave the discrete-time ordinary actions. A hybrid action system $\mathcal{A}$ is a statement

of the form

$$\begin{array}{lcll} \mathcal{A} & = & |[& \textbf{var} \quad Y^* := V \ ; X := U \\ & & & \textbf{proc} \quad p_1 = P_1 \ ; \cdots ; \ p_n = P_n \\ & & & \textbf{alt } A \textbf{ with } D \\ & &]| & : Z \end{array}$$

where A is a non-deterministic composition of ordinary actions $a_1 [\!] \ldots [\!] a_n$, D is the non-deterministic composition of differential actions $e_1 :\!\rightarrow d_1 [\!] \ldots [\!] e_n :\!\rightarrow d_n$ and **alt** A **with** D is their alternation, defined as

$$\textbf{do } A \ [\!] \ \neg gd(A) \rightarrow D \textbf{ od}$$

When $gd(D)$ is *false* we just write '**alt** A', and when $gd(A)$ is *false* we just write '**with** A'. A computation of a hybrid action system is, a finite or infinite sequences of evolution states (mapping *evolution* variables to $\mathbb{R}$), interleaved with ordinary states (mapping of *evolution* variables to values).

We conclude by recalling the definition of parallel composition of hybrid action systems [16, 17]. Consider the action systems $\mathcal{A}_i$ for $i = 1, 2$:

$$\begin{array}{lcll} \mathcal{A}_i & = & |[& \textbf{var} \quad Y_i^* := V_i \ ; X_i := U_i \\ & & & \textbf{proc} \quad p_{i,1} = P_{i,1} \ ; \ldots ; \ p_{i,n_i} = P_{i,n_i} \\ & & & \textbf{alt } A_i \textbf{ with } D_i \\ & &]| & : Z_i \end{array}$$

where the global variables, local variables and the local procedure headers declared in each action system $\mathcal{A}_i$ are required to be distinct. The *parallel composition* $\mathcal{A}_1 \parallel \mathcal{A}_2$ is defined as the action system

$$\begin{array}{ll} |[& \textbf{var} \quad (Y_1^* := V_1 \ ; Y_2^* := V_2 \ ; X_1 := U_1 \ ; X_2 := U_2) \\ & \textbf{proc} \quad p_{1,1} = P_{1,1} \ ; \ldots ; \ p_{1,n_1} = P_{1,n_1} \ ; p_{2,1} = P_{2,1} \ ; \ldots ; \ p_{2,n_2} = P_{2,n_2} \\ & \textbf{alt } A_1 [\!] A_2 \textbf{ with } D_1 \oplus D_2 \\ ]| & : (Z_1 \cup Z_2) \setminus (Y_1 \cup Y_2) \end{array}$$

1.3 Pharmacokinetic Modeling

The pharmacokinetics of a drug is a very complex biological process and there are no exact mathematical models for describing the concentration of a drug at any time in any part of the body. Pharmacometricians approximate the physiological process of the interactions between an organism and a drug by a compartmental model. Compartments represent the fluids and tissues of the human body, and each of them is assigned with input and output rates modeling the process of absorption and removal of the drug with regard to the tissues. This method is known as the ADME modeling scheme [13]:

- *Absorption* - the process of a substance movement to the blood stream.
- *Distribution* - the dispersion or dissemination of substances throughout the fluids and tissues of the body.
- *Metabolization* - the recognition by the organism that a foreign substance is present and the irreversible transformation of parent compounds into daughter metabolites.
- *Excretion* (or elimination) - the removal of the drug from the body. In rare cases, some drugs irreversibly accumulate in body tissue.

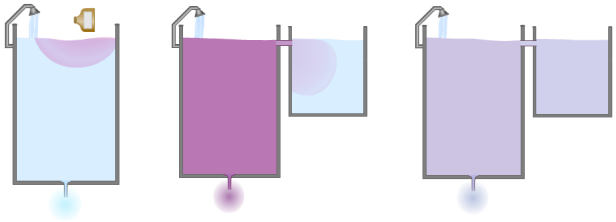
A drug can be given to a patient in several ways, e.g., intravenously, orally, subcutaneously, intramuscularly, with a skin patch or by inhalation. Intravenous administration does not involve absorption and, hence, there is no loss of the drug. On the other hand, with oral administration, several factors affect the absorption of a drug: solubility and permeability, gastrointestinal pH, gastric emptying time, small intestinal transit time, etc. [19]

The fraction of an administered dose of unchanged drug that reaches the systemic circulation is known as *bioavailability* and is one of the principal pharmacokinetic properties of a drug. After absorption, most of the drug is distributed via the blood to the body tissues. Distribution describes the reversable transfer of the drug between blood, tissues and organs. The drug is more easily distributed in highly perfused organs such as heart, brain, lungs, liver, and kidney than in poorly perfused tissues such as fat, skin and muscles.

Metabolization refers to the bio-transformation of the drug to other molecules—called *metabolites*—inside the body. Metabolites are normally pharmacologically inactive, but they can be active or toxic. Some drugs remain inactive until metabolized. The principal organ involved in excretion of a drug is the kidney, which eliminates water soluble drugs with urine. Bio-flow from the liver is also an important route for the elimination of a drug. Drugs also leave the body via other natural routes: breath, tears, sweat, and saliva.

About half of the human body weight is water. Water is eliminated from the body in urine, with about 6 liters of water passing through the kidney filter per hour. Thus, we can model this process as a filled up tank with a tap and an open plug where the flow through the tap equals the flow through the plug hole. We can think of water flowing in the blood-stream through the kidneys as water in the tap. Now imagine adding some drug to the water. The speed with which the drug distributes through the tank can represent the process of drug absorption (Figure 1.1(a)). How the drug penetrates to the peripheral tanks can represent the distribution process (Figure 1.1(b)). Finally, the elimination process can be represented by the flow out of the tank through the plug hole (Figure 1.1(c)).

The pharmacokinetics compartment model assumes that the drug concentration is perfectly homogenous in each compartment of the body at all time. Thus, we also assume that the drug concentration is homogenous at each



(a) Absorption (b) Distribution (c) Elimination

FIGURE 1.1: ADME scheme

time in each tank. This is a strong assumption but it allows a quantitative description of the pharmacokinetics processes. The quantitative pharmacokinetic model describes how the drug amount varies in each compartment, or equivalently, how the drug amount varies in each water tank. Such a description mainly reduces to 3 main components:

- *Rate in* describes how the drug moves to the blood stream (or how the drug moves to the main tank)
- *Rate of distribution* describes the movement of a drug between the compartments (or the distribution of the drug between tanks)
- *Rate out* describes how the drug is eliminated from the body (or how the drug flows out of the main tank)

Absorption, distribution and elimination are continuous processes limited by the physiological limitation of the tank, and the administered dose of the drug. Using the chemical balance law

$$\text{rate of change} = \text{input rates - output rates}$$

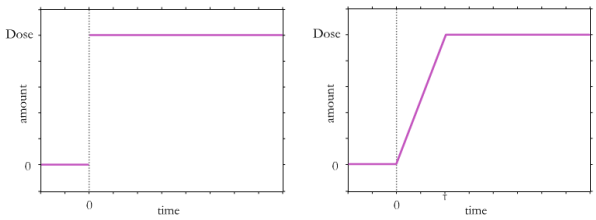
distribution can be described in terms of the other two components of each compartment. The pharmakinetics model of the system combines all these rates for each compartment into a single large system of differential equations. Next we describe in more detail the absorption and elimination processes, and give an example of their models as action systems.

1.3.1 Absorption

Absorption starts right from the beginning and stops when the amount of drug in the entire system is a bioavailability fraction $0 \leq f \leq 1$ of the administered dose (denoted *dose*). We use the local Boolean variable *abs* to records whether absorption take place or not, and the global variable *dose* to record the amount of administered dose of the drug. The global evolution

variable x describes the amount of drug present in the tissue modeled by a compartment (here, for example, blood), and it is dependent on time t.

Intravenous administration bypasses the absorption phase: the entire drug dose enters the general circulation (bioavailability fraction $f = 1$). We distinguish intravenous *bolus administration* and intravenous *infusion*. In the first case, the drug is administrated through the intravenous route in a negligible time and achieves instantaneous distribution throughout the central compartment. Thus, just before the administration, the amount of drug is 0, and just after administration the amount equals the administered dose (Figure 1.2(a)). In the case of intravenous infusion, the dose is administered with a constant rate during some time of infusion T (Figure 1.2(b)).



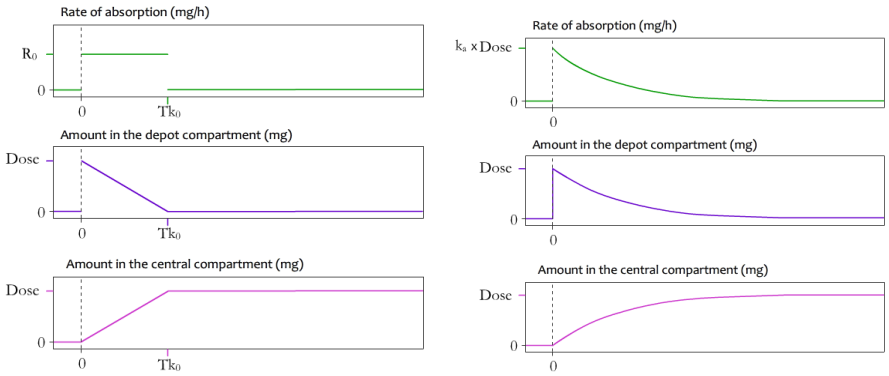
(a) Bolus (b) Infusion

FIGURE 1.2: Intravenous administration

With oral administration of a drug (e.g., in tablet, capsule or liquid form), the entire dose does not reach the systemic circulation ($f < 1$) due to factors such as breakdown in the intestine, poor absorption, and pre-systemic extraction. When swallowed, the drug enters the gastrointestinal tract and is then absorbed to the blood stream. The simple pharmacokinetic model considers the gut as a depot compartment that receives the dose. Describing the absorption process then reduces to describing how the dose is transferred from the depot compartment to the central blood compartment. In a zero-order absorption process (see Figure 1.3(a)), a drug is absorbed over time-period T with a constant rate $k_0 = dose/T$. In a first-order absorption process (see Figure 1.3(b)), the absorption rate is proportional to the amount of drug in the depot compartment. The proportionality constant is usually denoted k_a.

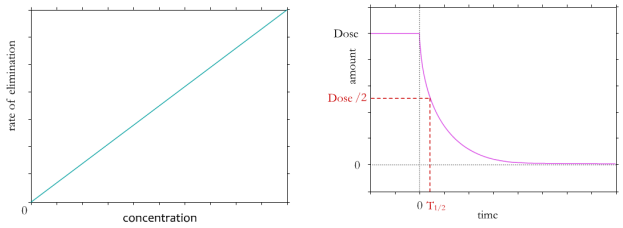
For example, let us consider oral administration of a drug, assuming the mono-exponential rate of transfer to the blood circulation given by $dx/dt = f \cdot dose \cdot k_a e^{-k_a \cdot t}$, where f is the fraction of bioavailability and k_a is the proportionality constant. Absorption into the blood can then be modeled as the following hybrid action system:

$$
\begin{array}{lcll}
Absorption & = & |[& \textbf{var} \quad abs^* := true \\
 & & & \textbf{alt}\ abs \wedge x \geq f \cdot dose \rightarrow abs := false \\
 & & & \textbf{with}\ abs \wedge x < f \cdot dose :\rightarrow \dot{x} = f \cdot dose \cdot k_a e^{-k_a \cdot t} \\
 & &]| & : dose, x
\end{array}
$$



(a) Zero-order absorption (b) First-order absorption

FIGURE 1.3: Oral administration



(a) Linear elimination rate (b) Drug concentration

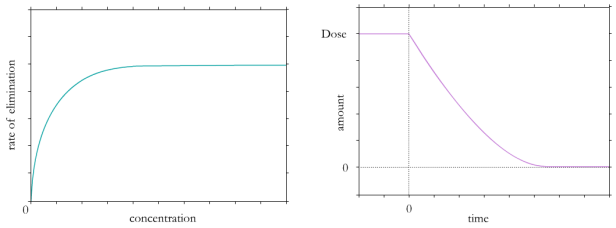
FIGURE 1.4: First-order elimination process

According to the above differential equation, the amount of drug into the blood stream at time t is given by $x(t) = e^{f \cdot dose \cdot k_a t}$. Here t ranges from 0 to the first point in time when $x(t)$ is not smaller than the bioavailability fraction of the administered dose.

1.3.2 Elimination

As soon as drug concentration is above a certain threshold, the elimination process starts. Elimination is modeled by opening the plug of a water tank. We assume that the concentration of the drug that comes out of the tank at any time equals the concentration of the drug in the tank. There are several mathematical models to describe the elimination process. With first-order (linear) elimination, the elimination rate is directly proportional to the plasma concentration (Figure 1.4(a)). Thus, the amount of the drug in the tank decreases with a decreasing rate (Figure 1.4(b)).

With saturable elimination, the elimination rate increases with the increase in the concentration until a certain concentration is reached, after that the



(a) Saturable elimination rate (b) Drug concentration

FIGURE 1.5: Mixed elimination process

elimination rate stays constant (Figure 1.5(a)). Such capacity-limited elimination is known as a mixed-order process. The corresponding drug amount over time is shown in Figure 1.5(b): the amount decreases with a rate which, at first, is almost constant and then slows down to resemble linear elimination.

For example, in a zero-order one-compartment model, let us assume that the drug elimination rate is $dx/dt = kx$, for some constant $k > 0$. An action system modeling the elimination processes is as follows:

$$\begin{array}{lcll} \mathit{Elimination} & = & |[& \textbf{var} \quad x^* := 0 \\ & & & \textbf{with}\ x > \mathit{min} :\rightarrow \dot{x} := -k \cdot x \\ & &]| & : \end{array}$$

where min is a threshold of minimal amount of drugs in the blood-stream before starting, so that the amount of drugs at time t is defined by the function $x(t) = x_0 e^{-kt}$ in terms of the initial amount x_0 at time 0.

1.3.3 One-compartment model

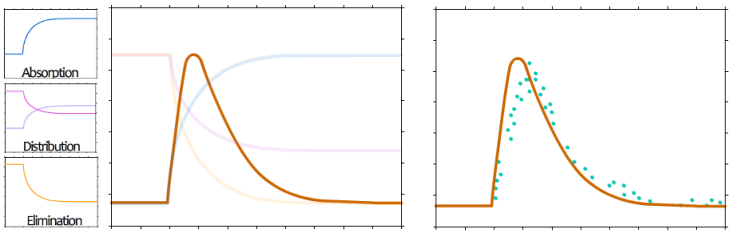
The overall rate of change of the amount of drugs in a single compartment over time is the combination of the rate *in* and the rate *out*. In this simplified model, distribution does not play a role.

The overall pharmacokinetic model is obtained through the parallel composition of the hybrid action systems modeling absorption and elimination of that compartment. Continuing our example we have

$$\mathit{BloodTank} = \mathit{Absorption} \parallel \mathit{Elimination}$$

Unfolding the definition we obtain the following one-compartment model of drug concentration in blood from an oral administration:

$$\begin{array}{l} \mathit{BloodTank} = |[\ \textbf{var} \quad \mathit{abs}^* := \mathit{true}, x^* := 0 \\ \qquad \textbf{alt}\ \mathit{abs} \wedge x \geq f \cdot \mathit{dose} \rightarrow \mathit{abs} := \mathit{false} \\ \qquad \textbf{with}\ \mathit{abs} \wedge \mathit{min} < x < f \cdot \mathit{dose} :\rightarrow \dot{x} = f \cdot \mathit{dose} \cdot k_a e^{-k_a \cdot t} - k \cdot x \\ \qquad [\!] \ \mathit{abs} \wedge x \leq \mathit{min} \wedge x < f \cdot \mathit{dose} :\rightarrow \dot{x} = f \cdot \mathit{dose} \cdot k_a e^{-k_a \cdot t} \\ \qquad [\!] \ (\neg \mathit{abs} \vee x \geq f \cdot \mathit{dose}) \wedge x > \mathit{min} :\rightarrow \dot{x} = -k \cdot x \\ \quad]| : \mathit{dose} \end{array}$$



(a) ADME model (b) Predictions vs. observations

FIGURE 1.6: pharmacokinetic modeling

The first differential equation gives the concentration of drug in the bloodstream when the drug is absorbed and eliminated at the same time. Until the minimal concentration is reached, only absorption takes place (as we see in the second differential equation). The third differential action represents elimination when absorption is terminated.

1.3.4 Distribution

The above standard pharmacokinetics equations for one-compartment models are rather simple. Many categories of drugs, however, cannot be accurately characterized by one-compartment models. For example, the effects of anesthetic drugs greatly depend on distribution of the active substance into and out of peripheral tissues. To reflect the differences in perfusion of tissues, several compartments are used in the pharmacokinetic model. In this case, the construction of a certain pharmacokinetic model consists of choosing a model for absorption (or distribution) and elimination for each compartment from the current models pharmacometricians have developed (Figure 1.6(a)), and which have already been validated against observations in clinical studies (Figure 1.6). For example, the DDMoRe model repository[1] is a place where pharmacokinetic models can be stored, retrieved and shared with the community.

A multi-compartment model can be translated into the parallel composition of hybrid action systems, each modeling a single compartment. Using one global evolution variable for each compartment, we can describe the amount of a drug present, assuming circular drug distribution among compartments, e.g., as i n the case when drug metabolites are reabsorbed from kidneys along with water before they are excreted in the urine.

The overall rate of change of the amount of a drug in the body over time is the combination of the rate in, *in*, the rate of *distribution* and the rate *out*.

[1] `http://repository.ddmore.eu`

1.4 Conclusions and Future Work

In this paper, we propose action systems as precise models of the compositional semantics of pharmacokinetic processes. For simple compartmental models, we found no need to develop new extensions for action systems, as existing work on continuous and hybrid actions fit perfectly. However, we have not fully explored the potential of action systems for reasoning about pharmacokinetic (and pharmacodynamic) processes. In particular, it will be interesting to prove properties of such systems formally within the refinement calculus [3, 17].

We are planning to use hybrid action systems in the context of the ApiNATOMY project[2] to automatically combine and formally analyze quantitative descriptions of physiological processes across multiple scales. A method central to the ApiNATOMY effort is the visualization of ontology terms as tiles in a treemap, which allows us to automatically generate body plans from given ontologies, preserving spatial relations among selected components [9]. Metadata related to ontology terms is represented in the form of objects associated with tiles as well as visual connections between tiles and their associated objects. To be able to generate views which are of interest to a user, we need to employ compositional formal models of related processes. Action systems represent a promising framework for our goal. As future work, we will employ action systems to model and control heterogeneous sets of independently developed pharmacokinetic and pharmacodynamic models, e.g., as in the alcohol consumption process described by de Bono and Hunter [10]. Each of the individual models can be represented by a separate hybrid action system combining both discrete and continuous logic, which are then assembled into a complete process using sequential, parallel or guarded choice composition operators defined for action systems.

[2] `http://www.apinatomy.org`

Bibliography

[1] R.J.R. Back. Refinement calculus ii: parallel and reactive programs. In J.W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *Stepwise Refinement of Distributed Systems: Models, Formalisms, Correctness*, volume 430 of *Lecture Notes in Computer Science*, pages 67–93. Springer-Verlag, 1990.

[2] R.J.R. Back and R. Kurki-Suonio. Decentralization of process nets with centralized control. *Distributed Computing*, 3(2):73–87, 1983.

[3] R.J.R. Back and K. Sere. From action systems to modular systems. *Software - Concepts and Tools*, 17, 1996.

[4] R.J.R. Back and J. von Wright. Refinement calculus i: Sequential nondeterministic programs. In J.W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *Stepwise Refinement of Distributed Systems: Models, Formalisms, Correctness*, volume 430 of *LNCS*, pages 42–66. Springer-Verlag, 1990.

[5] M.M. Bonsangue and J.N. Kok. The weakest precondition calculus: recursion and duality. *Formal Aspects of Computing*, 6A:788–800, 1994.

[6] L. Cardelli. On process rate semantics. *Theoretical Computer Science*, 391(3):190–215, 2008.

[7] F. Ciocchetta and J. Hillston. Bio-pepa: A framework for the modelling and analysis of biological systems. *Theoretical Computer Science*, 410(33–34):3065–3084, 2009.

[8] V. Danos and C. Laneve. Formal molecular biology. *Theoretical Computer Science*, 325(1):69–110, 2004.

[9] B. de Bono, P. Grenon, M. Helvenstijn, J. Kok, and N. Kokash. ApiNATOMY: Towards multiscale views of human anatomy. In *Processings of the 13th International Symposium on Intelligent Data Analysis (IDA)*, volume 8819 of *LNCS*, pages 72–83. Springer-Verlag, 2014.

[10] B. de Bono and P. Hunter. Integrating knowledge representation and quantitative modelling in physiology. *Biotechnology Journal*, 7(8):958–972, 2012.

[11] E.W. Dijkstra. *A Discipline of Programming*. Prentice–Hall International, 1976.

[12] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.

[13] M. Lavielle. *Mixed Effects Models for the Population Approach: Models, Tasks, Methods and Tools*. Chapman and Hall/CRC, 2014.

[14] C. Priami. Stochastic pi-calculus. *Computer Journal*, 38(7):578–589, 1995.

[15] A. Regev and E. Shapiro. Cellular abstractions: Cells as computation. *Nature*, 419(6905), 2002.

[16] M. Rönkkö and A.P. Ravn. Action systems with continuous behavior. In P.J. Antsakilis, W. Kohn, M. Lemmon, A. Nerode, and S. Sastry, editors, *Hybrid System V*, volume 1567 of *Lecture Notes in Computer Science*, pages 304–323. Springer, 1999.

[17] M. Rönkkö, A.P. Ravn, and K. Sere. Hybrid action systems. *Theoretical Computer Science*, 290:937–973, 2003.

[18] L. Shargel, A. Yu, and S. Wu-Pong. *Applied Biopharmaceutics & Pharmacokinetics*. McGraw Hill Professional, 6th edition, 2012.

[19] N.N. Song, S.Y. Zhang, and C.X. Liu. Overview of factors affecting oral drug absorption. *Asian Journal of Drug Metabolism and Pharmacokinetics*, 4:167–176, 2004.