Hybrid-System Modeling of Human Blood Clotting

Joseph Makin and Srini Narayanan

TR-08-002

February, 2008

Abstract

The process of human blood clotting involves a complex interaction of continuous-time/continuous-state processes and discrete-event/discrete-state phenomena, where the former comprise the various chemical rate equations (which can be written as differential equations) and the latter comprise both threshold-limited behaviors and qualitative facets of the coagulation cascade. We model this process as a hybrid dynamical system, consisting of both discrete and continuous dynamics. Previous blood clotting models used only continuous dynamics and perforce addressed only portions of the coagulation cascade. The model was implemented as a hybrid Petri net, a graphical modeling language that extends ordinary Petri nets to cover continuous quantities and continuous-time flows. The primary focus is simulation; specifically, we are interested in (1) fidelity of simulation to the actual clotting process in terms of clotting factor concentrations and elapsed time; (2) simulating deficiencies, surfeits, and other perturbations of initial values of blood proteins, and their consequences for clotting factor concentrations and for clotting time, especially in the context of known clotting pathologies; and (3) providing fine-grained predictions which may be used to refine clinical understanding of blood clotting.

Hybrid-System Modeling of Human Blood Clotting

Joseph Makin and Srini Narayanan
International Computer Science Institute
1947 Center Street
Berkeley, CA 94704
Email: makin@eecs.berkeley.edu, snarayan@icsi.berkeley.edu

***Abstract*—The process of human blood clotting involves a complex interaction of continuous-time/continuous-state processes and discrete-event/discrete-state phenomena, where the former comprise the various chemical rate equations (which can be written as differential equations) and the latter comprise both threshold-limited behaviors and qualitative facets of the coagulation cascade. We model this process as a hybrid dynamical system, consisting of both discrete and continuous dynamics. Previous blood clotting models used only continuous dynamics and perforce addressed only portions of the coagulation cascade. The model was implemented as a hybrid Petri net, a graphical modeling language that extends ordinary Petri nets to cover continuous quantities and continuous-time flows. The primary focus is simulation; specifically, we are interested in (1) fidelity of simulation to the actual clotting process in terms of clotting factor concentrations and elapsed time; (2) simulating deficiencies, surfeits, and other perturbations of initial values of blood proteins, and their consequences for clotting factor concentrations and for clotting time, especially in the context of known clotting pathologies; and (3) providing fine-grained predictions which may be used to refine clinical understanding of blood clotting.**

I. Introduction

The process of blood coagulation in mammals is complicated, and involves the interaction of more than a dozen coagulation factors as well as a number of proteins from the kinin-kallikrein system and protein inhibitors. Attempts to model coagulation mathematically therefore usually focus on a smaller subset of interactions, perhaps one of the pathways or just a portion of one of them (cf. [1], [2], [3], [4], [5], [6] and [7]). Such models generally consist of a set of coupled, usually nonlinear, differential equations governing the time evolution of protein concentrations, derived ultimately from the law of mass action. Although on one level of analysis all of the processes in blood clotting comprise discrete events, like cleavage of chemical bonds, formation of bonds, and the like; nevertheless, at the scale of interest it is concentrations that matter, and these exhibit continuous dynamics.

There are, however, at least two reasons why this methodology is inadequate for modeling the entire coagulation cascade. First, there is reason to believe [8] that certain events in the cascade are better modeled by punctuated phase changes, rather than as evolving continuously. Second, the problem with modeling coagulation as a purely continuous-time phenomenon is that the process is too complicated (with thresholds and discontinous phase changes) to permit a precise description of the various parameters and their interactions in terms of differential equations.

The alternative pursued here is to model the coagulation cascade as a "hybrid system" (HS), i.e. one consisting of interacting continuous and discrete dynamics. Hybrid-system theory is a fairly new area of research at the intersection of control theory and computer science which has made considerable progress in the last fifteen years along a number of different but related frontiers. Chief among these are analysis, in the form of verification and decidability; controller synthesis; and modeling and simulation. In the present case, our aim was to construct a robust and faithful model of the coagulation process: faithful in the sense of accurately modeling human (or generally, mammalian) blood clotting, and robust in the sense of doing so over a wide range of parameter settings.

We additionally required that our model be perspicuous (with the biologist in mind), and easily modifiable. In light of these constraints, the model was implemented using hybrid Petri nets (HPNs), a graphical modeling formalism for modeling hybrid systems. Classical Petri nets are a well known computational formalism for the modeling and simulation of discrete-event dynamical systems, with constructs for sequential and concurrent process execution, for resource consumption and production, and for inhibition. HPNs extend classical Petri nets from the domain of purely discrete phenomena to the domain of hybrid dynamics. HPNs are thus able to incorporate continuous-time and continuous-state phenomena by supplementing the traditional discrete-event architecture with continuously varying events and states.*

The model serves both a specific and a more general purpose. Specifically, by accurately simulating the blood clotting process, the model serves as a basis for predictions: the effect of alterations in coagulation-factor concentrations, on both the overall clotting time and on the concentration of other factors, can be simulated effectively. These simulations can serve as the basis for predictions about the effect of pharmacological intervention; for understanding the nature of certain blood-clotting disease pathologies (e.g. the various hæmophilias, factor V Leiden, etc.); and for refinement of our understanding of blood clotting in general. More generally, the model demonstrates the utility of the design methodology, *viz.* using hybrid systems to model cascade-like biological processes where both

*Our current implementation is based on the *Visual Object Net++* platform, a dedicated HPN modeling and simulation environment [9], which includes a graphical language that offers a suite of object-oriented programming (OOP) features: hierarchical organization, inheritance and object reuse.

discrete and continuous dynamics play a role. In virtue of its ability to incorporate both types of dynamics, the model is able to support robust analysis and prediction in cases where parts of a complex process may be known precisely (with differential equations) while other aspects may have qualitative descriptions only (through punctuated phase changes, discrete transitions, and threshold behaviors). This ability to reason effectively with representations of multiple granularities addresses a central requirement in modeling complex biological processes.

The model was validated by simulating normal blood clotting, as well as various blood clotting disorders. The resulting time to clotting and the time course of blood protein concentrations were compared against the clinical literature, and gave consistent results.

II. Modeling Blood Clotting: Overview

Mammalian blood clotting is a complicated process which unfolds largely through a protein activation cascade. A blood vessel breakage precipitates the modification of certain proteins called clotting or coagulation factors, transforming them from their unactivated to activated forms. As concentrations of the activated forms increases, the proteins trigger the activation of other clotting factors, and so on through the coagulation cascade. Ultimately, a fiber-like protein (aptly named "fibrin") is produced, and binds to the site of injury along with platelets and another clotting factor (XIIIa), producing a clot and sealing the damaged vessel. Along the way, other proteins serve to inhibit the activation of clotting factors and still others to dilate the blood vessel ("vasodilators").

Previous attempts to model the coagulation cascade mathematically have focused on the continuous-time, (usually) nonlinear differential equations which describe the evolution of protein concentrations, equations deriving from the law of mass action, the Michaelis-Menten equations, or other chemical considerations. Due to the extremely complicated nature of blood clotting, these attempts usually focus on small subsets of the entire process. So, for example, [6] models only the interactions of coagulation factors II, IX, and X; [1] and [10] model the extrinsic pathway; [5] models a portion of the common pathway and a small part of the intrinsic pathway; and [11] and [4] model some of the interactions of factors II, V, VII, VIII, IX, and X; (see §IV-B for an explanation of the different clotting pathways). The largest and most ambitious continuous-time model is [3], which uses a system of 73 coupled nonlinear differential equations to describe a large portion of the entire clotting process, including all of the extrinsic pathway, a large portion of the (ulterior) intrinsic pathway, and the common pathway up through thrombin production.

Whereas all of these models involve continuous dynamics, there is reason to believe that certain aspects of the cascade are better modeled as discrete events. First, certain enzymes exhibit threshold effects in activation or inhibition of other proteins [12], [13], [14]. For example, antithrombin III and α_2-macroglobulin are only able to inhibit thrombin activation below a certain threshold of thrombin [12]. Similarly, concentrations of free zinc ions are thought to toggle the activation of several of the proteins of the contact activation portion of the clotting cascade [13], [14]. Second, some of the more complex aspects of the coagulation process are poorly understood, and certainly there is no closed-form set of differential equations for the system as a whole. On the other hand, more coarse-grained information is available—e.g., whether a certain protein factor must be present in some quantity in order for a reaction to take place—and this can be encoded in the form of discrete states.

Even in cases where this type of information is not available, coarse representations can nevertheless be incorporated into a model, after which comparisons can be made between the simulation and clinical observations, on the basis of which adjustments may be made to the original representations. The process can then be iterated. It should be noted that model refinement is much more feasible within this methodology than via the alternative of experimenting with various differential equations, especially in the face of exiguous theoretical or experimental knowledge. And, finally, incorporation of so-called coarse representations into the model need not be seen merely as a stepping stone to an ideally complete model: a model which contains such representations can provide bounds on the behavior of the entire system, which are informative *per se*.

Thus, in order to capture both these discrete states and the continuous dynamics of the chemical rate equations, the coagulation process is best modeled as a hybrid composition of the two. Before explaining this model, we take a brief detour through hybrid system theory.

III. Hybrid Systems: Overview

A. An introduction to hybrid systems

Classical control theory and system modeling have focused on systems with purely continuous dynamics and those with purely discrete dynamics. However, many real-world systems necessarily involve both continuous and discrete components, or are best modeled as interacting continuous and discrete subsystems. These systems are called *hybrid systems*, and have motivated a great deal of research in the last two decades. What follows is a somewhat pedantic commentary for readers completely unfamiliar with hybrid systems; initiates should skip to §IV.

A system may be considered hybrid either because the state space consists of both continuous and discrete components, or because the dynamics manifest both continuous-time and discrete-time behaviors. Consider, for example, the by now well known example of a thermostat. (The following treatment of the classic thermostat model was adapted with little change from an example in [15].) Suppose the change in temperature is governed by one differential equation when the heat is on,

$$\dot{x} = K(85 - x); \tag{1}$$

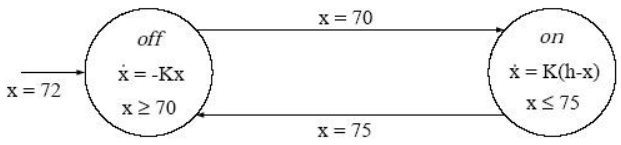

Fig. 1. A hybrid system model of a thermostat. The variable h should be greater than 75^o, say 85^o, in order to force a transition out of the on state.

but when the state (temperature) crosses a certain threshold, say 75^o F, the heat is turned off and a new differential equation applies:

$$\dot{x} = -Kx. \tag{2}$$

When the temperature falls below 70^o F, the heat goes back on and Eq.(1) again applies. Thus the temperature increases exponentially toward 85^o F until it hits a boundary in the state space, at which point the heat is turned off and the temperature declines, again exponentially, toward zero degrees. The various regions where different sets of continuous-time dynamics obtain may be modeled as different discrete states of the system; hence, the entire network may be thought of as the first type of hybrid system, where the state is jointly defined by a continuous variable (the temperature) and a discrete state (indicating whether or not the heat is on).

The model is illustrated in Fig. (1). The system is very much like a finite state machine, but additionally a set of continuous dynamics (a differential equation) is associated with each of the discrete states. The arrow labeled $x = 72$ indicates that the initial state is $(x = 72, q = off)$, where the pair (x, q) defines the state, with $x \in X$, the continuous state space, and $q \in Q$, the discrete state space. The arrows connecting discrete states are, as in finite state machines, called "edges," and are written (q, q'), where the edge starts at q and ends at q'. The edges are labeled with the so-called guard conditions ($x = 70$, $x = 75$); given the state (x, q), if x belongs to the set specified by the guard associated with the edge (q, q') then the system may transition to the discrete state q'. The continuous variable may in general also be reset at a discrete transition, but all of the reset maps of the thermostat model are the identity map.

Finally, a domain $D(q)$ ($x \geq 70$, $x \leq 75$) is associated with each discrete state. When the continuous state reaches the boundary of the domain, a discrete transition is forced—unless no guard condition is satisfied, in which case the system is "blocked" and stops. Alternatively, the guard may be enabled *before* the trajectory reaches the boundary of the domain, in which case the discrete transition may but need not occur, and the model becomes non-deterministic. This point is made for the sake of generality; in the thermostat example, the guard conditions and domain have been written so as to preclude both blocking and non-determinism. The same is true of the blood clotting model presented in this paper.

The thermostat may be modeled as a hybrid system because, again, it has discrete states as well as continous states and evolves in continuous time. However it should be noted that hybrid systems may also arise in the interaction of discrete-*time* systems with continuous dynamics. So, for example, we may wish to model the interaction of a purely continuous system like a chemical batch process with a digital controller which has an essentially continuous state space but evolves in discrete time. The blood clotting model of this paper is of the first type.

B. Types of hybrid system investigation

Hybrid systems arise in numerous other contexts, among them chemical batch processes [16]; road-traffic controllers [17], [18]; air traffic control [19]; robotic control [20], [21]; automotive applications [22], [23]; embedded systems [24]; and biological systems [25], [26], [27], [28], [29]. In all of these examples, there is a variety of questions we may be interested in asking, and which recent research has attempted to find ways of answering. We provide a brief overview here in order to give the reader a flavor for the field.

1) Modeling and simulation: Perhaps the most obvious of these is modeling and simulation. For hybrid systems in particular, formal analysis is restricted to the simplest of systems or to highly circumscribed cases, so a powerful alternative is to construct a model of the system and simulate its behavior rather than perform an exhaustive analysis. However, whereas a great variety of modeling and simulation tools exists for purely continuous or purely discrete dynamics, only recently have such tools been developed specifically for hybrid systems. The extent of the novel modeling and simulation issues associated with hybrid systems are beyond the scope of this paper, but three should be mentioned because of their relevance and their ubiquity.

The first is choice of representation. Whereas purely discrete and purely continuous dynamics have fairly well-established representations from the computer science and control theory disciplines (respectively), a standard framework for modeling hybrid systems has not yet arisen. Hybrid systems also come in a host of flavors, and ideally a simulation or modeling program should be able to accommodate all of these. [15] has emphasized that a hybrid-system modeling language should be *descriptive*, in the sense of being able to model a wide range of continuous and discrete dynamics and their interactions, and to accommodate stochasticity in a variety of contexts; *composable* from smaller units into larger networks; and *abstractable*, in the sense of being able to cash out composite model specifications in terms of component specifications as well as determine composite-level behavior via knowledge of component behavior.

The second simulation concern is the development of accurate and efficient numerical integration techniques. In a familiar problem from the hybrid systems literature, imprecise numerical integration triggers a discrete event—and hence, perhaps, a new set of differential (state) equations—in the simulation, where no such event occurs in the actual system. Putative solutions which simply shrink the step size, however,

can greatly increase simulation time, and moreover are not *per se* a guarantee of eliminating this type of simulation error.

The third and final simulation issue to be discussed here is the problem of so-called "Zeno" systems, in which trajectories of the system take an infinite number of discrete transitions in finite time. In such cases, simulation time "stops," and the simulation terminates only when the system hangs. The well-known bouncing-ball hybrid system exhibits this type of behavior: the "moving up" and "moving down" state alternate increasingly faster, according to a geometric series that converges in the limit but will never converge in simulation. Even in non-Zeno systems, existence and uniqueness of solutions is not in general guaranteed for hybrid systems, and special care must be taken in their simulation.

2) Verification and Decidability: A second type of question we may be interested in asking about a particular hybrid system is known as *verification*. Verification is the formal proving of certain (interesting) system properties, given the system and a range of inputs. There are two variations: algorithmic verification ("model checking") and deductive verification, where the former uses search techniques and the latter involves the construction of a formal proof [30]. In both cases, the property of interest is usually reachability. In general, however, reachability analyses on hybrid systems are prohibitively difficult. Results have been confined to small classes of highly circumscribed systems (e.g., timed automata and rectangular automata [31]). Verification is also sometimes performed *vis-à-vis* the stability of the system; we may want to ask, say, if the closed-loop system is asymptotically stable [15].

It has also proven useful for analysis techniques, particularly reachability, to perform *abstractions* on systems; i.e., to replace a complicated hybrid system model with a less complicated one in which properties previously difficult or impossible to prove are rendered tractable. The general procedure is to construct a simplified model which contains the behavior of the original system as well as some new behavior, an artifact of the abstraction. This system is then tested for some appropriately abstracted version of the original property to be tested (e.g. reachability); for example, if it can be shown that some state is unreachable in the abstracted system, then it has been shown that it is also unreachable in the original system [30], [23].

A related question is that of *decidability*: whether a problem can be answered, affirmatively or negatively, by some algorithm in finite time. Specifically, in the present case, the question is whether or not a system can be verified in finite time; and more specifically, whether or not the reachability of some state(s) or region in the state space can be computed in finite time. It should be noted that reachability analyses can be fruitfully pursued even for undecidable systems, since the particular (say) state about which reachability is to be determined may be decidable, even though reachability in general is not [30], [31].

3) Controller synthesis: A third type of question that can be asked about hybrid systems concerns controller synthesis. Controller specifications in hybrid systems are often given in terms of temporal logic. For instance, we may require that all trajectories of a system remain within a set of states $F \subseteq X \times Q$, where X and Q are the continuous and discrete state spaces, respectively. The condition is written as

$$\Box((q, x) \in F) \tag{3}$$

where $q \in Q$ and $x \in X$ are the discrete and continuous state, respectively. Or we may insist that the trajectory eventually reach some set of states F, written

$$\Diamond((q, x) \in F) \tag{4}$$

Designing a controller then amounts to picking a set of inputs from the input space for each state of the system such that the specification of interest is satisfied. There are various techniques for performing this task. For example, both the theory of optimal control and game-theoretic approaches can be used to derive the Hamilton-Jacobi partial differential equations whose solutions are the boundaries of the reachable sets. These can then be solved approximately, providing (real-time) feedback control laws which provably satisfy the specifications [32].

IV. Model Implementation

A. Modeling language: Hybrid Petri Nets

There are numerous hybrid system modeling languages (see, for example, [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [9], and [43]) and in choosing among them the following constraints were considered. First and foremost, of the different types of hybrid system investigations canvassed above, our present focus is primarily simulation. (We are currently, however, exploring algorithms for verification and controller design, both of which would be useful in the context of pharmacological intervention; see §VI for details.)

A second major design focus was ensuring that the model be easily and intuitively modifiable, not just in parameter settings like concentrations of proteins, but in structure as well, so that new (say) proteins can be incorporated painlessly into it. An ideal model will also provide a perspicuous representation of the system of interest; this was especially significant in the present case since we intended the model for use by biologists who may have little or no familiarity with programming languages. This constraint obviously militates in favor of a graphical modeling framework.

A third and final consideration is that the clotting cascade consists in large part of numerous similar reactions among different clotting factors. The reactions often involve reactants which play no roles in any other reactions or events. These two facts suggest an object-oriented approach to modeling, since they make use of the usual features of object-oriented programming (OOP): abstraction, reuse, information hiding, and inheritance. (For another object-oriented approach to blood clotting, see [44]).

A modeling language which meets all of these constraints is the formalism of hybrid Petri nets. Petri nets are a graphical

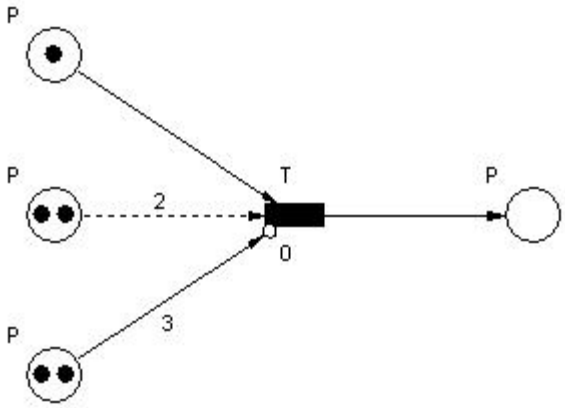

Fig. 2. A very simple Discrete Petri Net (DPN) with purely discrete components.

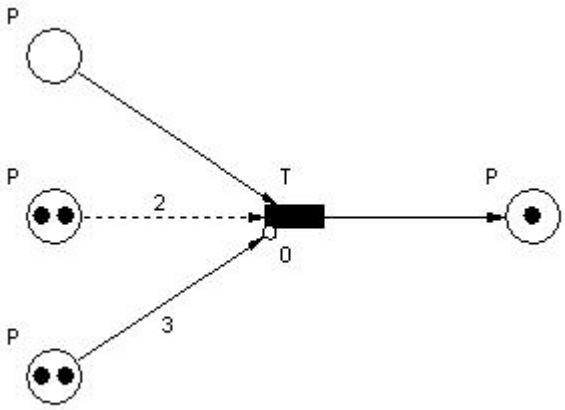

Fig. 3. The simple Discrete Petri Net (DPN) of Fig. (2), advanced by one step.

modeling formalism for distributed discrete-event systems which generalize automata theory to allow notions of concurrency and resource consumption. What follows is an an informal description, followed by a formal exposition of the network semantics.

An ordinary Petri net comprises three kinds of components: transitions, places, and directed arcs (see Fig. [2]). Places (drawn as circles) represent discrete quantities in virtue of the number of "tokens" (n) they carry ($n \in \mathbb{N}$), and events are modeled as firings of transitions (drawn as black rectangles). In the figure, tokens are drawns as little black circles which live in places. The distribution of tokens over all the places in the net is called the *marking* of the net. The marking changes when transitions fire and tokens are consumed from the input places and produced at the output places.

A transition is enabled if and only if each place connected via an input arc contains as many tokens as the "resource requirement" of its corresponding arc. If an arc is unlabeled, its resource requirement is assumed to be unity. When a transition is enabled, it "fires," meaning that tokens are consumed from input places according to the resource requirement, and produced in the output place(s) according to the weight(s) associated with the outgoing arc(s); since the outgoing arc is unlabeled in the figure, the output is assumed to be a single token.

Petri Nets are often outfitted additionally with "test" input arcs, which function exactly like normal arcs except that tokens are not consumed when a transition fires; and "inhibitory" input arcs, which enable transitions just in the case that the input place has *fewer* tokens than the resource requirements (again with no corresponding token consumption). These also appear in Fig. (2). The test arc is drawn as a dashed arrow; since the preceding place contains two tokens, it satisfies the resource requirement given by the test arc's weight. Similarly, the input place to the inhibitory arc, drawn as an arrow ending in an open circle, contains fewer than three tokens, so it does not inhibit the transition. The Petri net in Fig. (2) will thus transition to the one shown in Fig. (3).

Definition IV.1. Discrete Petri Net (DPN). *A* DPN *is a tuple* $(\mathcal{P}, \mathcal{S}, \mathcal{A}, \mathcal{W}, \mathcal{M}_0)$*, where:*

- $\mathcal{P}$ *is a set of* discrete places*.*
- $\mathcal{S}$ *is a set of* discrete transitions*.*
- $\mathcal{A}$ *is a set of weighted directed* arcs *which connect places to transitions and transitions to places; i.e.* $\mathcal{A} \subseteq (\mathcal{P} \times \mathcal{S}) \cup (\mathcal{S} \times \mathcal{P})$*.*
- $\mathcal{W}$*:* $\mathcal{A} \mapsto \mathbb{N}^+$ *maps each arc* $a \in \mathcal{A}$ *to a* weight w *from the positive integers.*
- $\mathcal{M}_0$*:* $\mathcal{P} \mapsto \mathbb{N}$ *is the* initial marking *of the network, which gives the original token distribution in the network via a map from each place* $p \in \mathcal{P}$ *in the net to the natural numbers.*

The meaning of an arc differs according to whether it connects places to transitions or vice versa, and on which of three flavors a member of the former set comes in: test arcs $\mathcal{E}$, inhibitory arcs $\mathcal{I}$, or resource arcs $\mathcal{R}$.

Definition IV.2. DPN Arcs. $\mathcal{A} = *\mathcal{A} \cup \mathcal{A}*$*, where*

- $*\mathcal{A} \subseteq (\mathcal{P} \times \mathcal{S})$ *are* input arcs*, and*
- $\mathcal{A}* \subseteq (\mathcal{S} \times \mathcal{P})$ *are* output arcs*.*

Furthmore, $*\mathcal{A} = \mathcal{E} \cup \mathcal{I} \cup \mathcal{R}$

DPNs have a well specified real-time execution semantics where the *next state* function is specified by the *firing rule*. In order to simulate the dynamic behavior of a system, a *marking* of the DPN is changed according to the following firing rule:

Definition IV.3. DPN Execution Semantics.

- *A transition* $s \in \mathcal{S}$ *is said to be* enabled *if:*
  1) *the source place* p *of each inhibitory arc* $i \in \mathcal{I}$ *of* s *has (strictly) fewer tokens than* w_{ps}*, and*
  2) *the source place of each test and resource arc* ($e \in \mathcal{E}$ *and* $r \in \mathcal{R}$*, respectively) contains at least* w_{ps}

tokens;

where in each case w_{ps} is the weight of the input arc from each source place p to s.

- *The* firing *of an enabled transition, s, removes w_{ps} tokens from the source, p, of each* resource arc*, and places w_{sp} tokens in each output place p.*

Notice that the semantics associated with the enable condition make reference only to the input arcs $*\mathcal{A}$, whereas the firing semantics invoke both input and output arcs. It should also be pointed out that the definition implies that transitions which have no input arcs are always enabled.

Transition firings take place as soon as their resource requirements are satisfied; i.e., time does not pass, even through successive transitions. Of course, we may want to model not simply the sequencing of events but the time it takes for the events to transpire. In this case we can assign delay times to the transitions: a transition with a delay of τ seconds will fire exactly τ seconds after its resource requirements have been met. If in the meantime the resources are depleted and the requirement is no longer fulfilled, then the transition will not fire. Time flows whenever none of the transitions are firing, which means that each transition is either not enabled or enabled but experience a delay. More formally, we need to augment the execution semantics as follows:

Definition IV.4. Time. *A DPN may be augmented with a* time *concept, where time $t \in \mathbb{R}$. Time stops running (i.e. increasing from $t_0 = 0$) whenever a (discrete) transition fires.*

Definition IV.5. Transition delay. *Associate to each transition $s_j \in \mathcal{S}$ a* delay *$\tau_j \in \mathbb{R}$. The firing of an enabled transition takes place at time $t^* + \tau$, where the transition was enabled at time t^* and remained enabled throughout the interval $[t^*, t^* + \tau]$.*

Assigning a delay of zero seconds to a transition restores the original execution semantics, i.e. firing takes place as soon as the transition is enabled.

In the present study, the Petri net language was additionally required to represent continuous-time events and a continuous state space. Hybrid Petri nets (HPNs) meet these requirements by providing, respectively, continuous transitions and continuous places with real-valued "token fluid."

The definition deserves some preparatory remarks: (1) In contrast to a DPN, an HPN marks its continuous places with real numbers, in addition to assigning integers to the discrete places. (2) No weights are assigned to arcs which link continuous places to continuous transitions, or which link continuous transitions to continuous places. Furthermore, the weight maps assign real or natural numbers where appropriate. (3) Neither resource arcs nor output arcs can join discrete places with continuous transitions. In contrast, continuous places and transitions can *only* be joined by resource or output arcs. Fig. (5) summarizes these constraints. (5) The meanings of the arc starring convention and of $\mathcal{E}, \mathcal{I}$, and $\mathcal{R}$ are the same as above. Notice that the continuous input arcs of $*\mathcal{A}^c$ are not members of these sets.

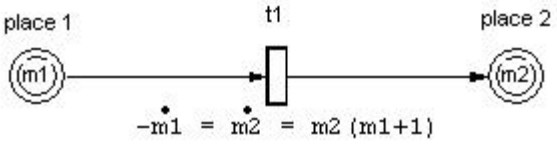

Fig. 4. A very simple Petri net with purely continuous components (CPN). The rate at which "token fluid" leaves m_1 is the rate at which it accumulates in m_2, which in this example is $m_2(m_1 + 1)$

Formally:

Definition IV.6. Hybrid Petri Net (HPN). *An* HPN *is a tuple $(\mathcal{P}, \mathcal{S}, \mathcal{A}, \mathcal{W}_c, \mathcal{W}_d, \mathcal{M}_{c0}, \mathcal{M}_{d0}, \mathrm{T}, F)$, where:*

- *$\mathcal{P} = \mathcal{P}_c \cup \mathcal{P}_d$ is a set of* continuous places *and* discrete places*.*
- *$\mathcal{S} = \mathcal{S}_c \cup \mathcal{S}_d$ is a set of* continuous transitions *and* discrete transitions*.*
- *$\mathcal{A} \subseteq (\mathcal{P} \times \mathcal{S}) \cup (\mathcal{S} \times \mathcal{P})$ is a set of weighted directed* arcs*. Furthermore,*
  - *$\mathcal{A}^d = \{a \in \mathcal{A} | a \in (\mathcal{P}_d \times \mathcal{S}_d) \cup (\mathcal{S}_d \times \mathcal{P}_d)\} \subseteq (\mathcal{E} \cup \mathcal{I} \cup \mathcal{R} \cup \mathcal{A}^d*)$;*
  - *$\mathcal{A}^{dc} = \{a \in \mathcal{A} | a \in (\mathcal{P}_d \times \mathcal{S}_c) \cup (\mathcal{S}_c \times \mathcal{P}_d)\} \subseteq (\mathcal{E} \cup \mathcal{I})$;*
  - *$\mathcal{A}^{cd} = \{a \in \mathcal{A} | a \in (\mathcal{P}_c \times \mathcal{S}_d) \cup (\mathcal{S}_d \times \mathcal{P}_c)\} \subseteq (\mathcal{E} \cup \mathcal{I} \cup \mathcal{R} \cup \mathcal{A}^{cd}*)$;*
  - *$\mathcal{A}^c = \{a \in \mathcal{A} | a \in (\mathcal{P}_c \times \mathcal{S}_c) \cup (\mathcal{S}_c \times \mathcal{P}_c)\}$; and*
  - *$\mathcal{A} = \mathcal{A}^d \cup \mathcal{A}^{dc} \cup \mathcal{A}^{cd} \cup \mathcal{A}^c$.*
- *$\mathcal{W}_c : \mathcal{A}^{cd} \mapsto \mathbb{R}^+$ is the continuous* weight map*.*
- *$\mathcal{W}_d : (\mathcal{A}^d \cup \mathcal{A}^{dc}) \mapsto \mathbb{N}^+$ is the discrete* weight map*.*
- *$\mathcal{M}_{c0}$: $\mathcal{P}_c \mapsto \mathbb{R}$ is the* initial token-fluid marking *of the network.*
- *$\mathcal{M}_{d0}$: $\mathcal{P}_d \mapsto \mathbb{N}$ is the* initial token marking *of the network.*
- *$\mathrm{T} : \mathcal{S}_d \mapsto (\mathbb{R}^+ \cup \{0\})$ maps each discrete transition $s \in \mathcal{S}_d$ to a non-negative real-valued* delay *τ.*
- *$F : \mathcal{S}_c \mapsto \mathcal{C}^0$ maps each continuous transition $s \in \mathcal{S}_c$ to a function $f(m_1, ..., m_n)$ of the network markings, from the space of continuous functions.*

The fundamental addition to the execution semantics is continuous token-fluid flow. As shown in Fig. (4), a continuous transition fires continuously according to a "firing speed" (i.e. a differential equation) which defines the speed of consumption and production of fluid from the various input and output places, respectively, associated with it. Token fluid leaves the input places at this rate and enters the output places at the same rate.

Definition IV.7. HPN Execution Semantics.

- *The* enabling *of an HPN is identical to that of a DPN, given in Def. (IV.3.)*
- *The* firing *for all arcs $a_{sp} \in (\mathcal{A}^d \cup \mathcal{A}^{dc} \cup \mathcal{A}^{cd})$ is identical to that of a DPN, given in Def. (IV.3.)*
- *The* firing *for all continuous* input *arcs $*a_{ps} \in *\mathcal{A}^c$ from*

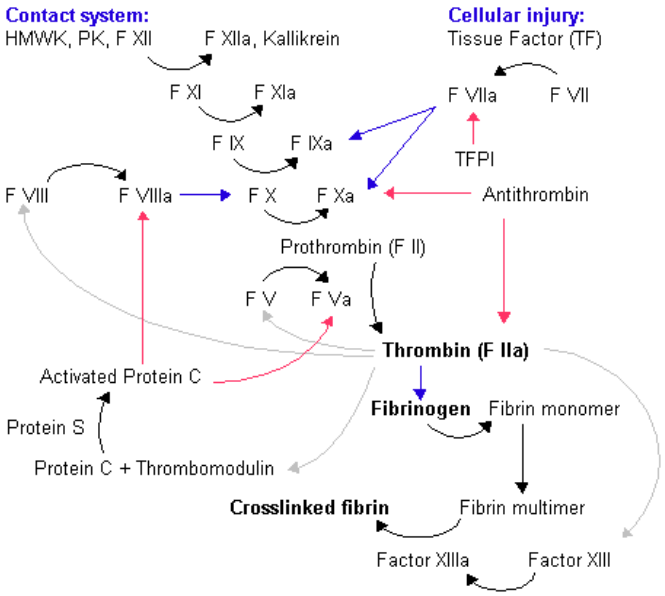

Fig. 6. The coagulation cascade comprises multiple interacting pathways. This figure is taken from http://en.wikipedia.org/wiki/Coagulation. Legend: HWMK = High molecular weight kininogen, PK = Prekallikrein, TFPI = Tissue factor pathway inhibitor. Black arrow = conversion/activation of factor. Red arrows = action of inhibitors. Blue arrows = reactions catalysed by activated factor. Gray arrow = various functions of thrombin.

place p to transition s is given by the equation

$$\frac{dm_p(t)}{dt} = -f_s(m_1(t), ..., m_n(t)), \tag{5}$$

where $m_p(t) \in \mathcal{M}_c$ is the marking associated with the input place p, and f_s is the function associated with the transition s.

- *The* firing *for all continuous* output *arcs $a*_{sp} \in \mathcal{A}_c*$ is given by the equation*

$$\frac{dm_p(t)}{dt} = f_s(m_1(t), ..., m_n(t)), \tag{6}$$

where $m_p(t) \in \mathcal{M}_c$ is the marking associated with the output place p, and f_s is the function associated with the transition s.

The reader should take care to note that the enable semantics from Def. (IV.3) apply only to arcs $*a \in \mathcal{E} \cup \mathcal{I} \cup \mathcal{R} = (*\mathcal{A}^d \cup *\mathcal{A}^{dc} \cup *\mathcal{A}^{cd})$, and that as a consequence, HPN transitions which are fed only by continuous input arcs $*a \in *\mathcal{A}^c$ or by no arcs at all are *always* enabled. Additionally, Def. (IV.3) refers only to "tokens," but in Def. (IV.7) it is assumed that this be taken as either tokens or token fluid, as the case may be.

B. The Coagulation Cascade

Fig. (6) outlines the coagulation cascade. The reader should refer to this figure *vis-a-vis* the description of its HPN counterpart in the sequel, which models just this cascade, albeit in more detail.

The cascade is simultaneously initiated by two different mechanisms whose resulting "pathways" meet at the activation of factor X into factor Xa (thrombokinase). The intrinsic pathway is triggered when vascular cell damage exposes blood plasma to a negatively charged surface (phospholipids)—hence the term "contact system." It is now also believed that vascular injury precipitates a rise in plasma zinc concentration, which in turn enables certain initiating reactions of the intrinsic pathway (see below for more details) [13], [14].

The extrinsic pathway is initiated when the ruptured blood vessel releases tissue factor (sometimes called factor III or tissue thromboplastin) into the plasma, which subsequently binds with unactivated factor VII (proconvertin). The details of this pathway are discussed in detail below in tandem with a description of the model.

The common pathway begins at the activation of factor X and terminates ultimately in the production of a fibrin clot. The most important product of this pathway is thrombin (activated factor II), which is directly involved in the formation of a blood clot but also participates in feedback reactions in both the intrinsic and common pathways. To reiterate, both the intrinsic and extrinsic pathways feed the common pathway, but in fact it is now widely believed that the extrinsic pathway serves to "kick-start" the intrinsic pathway into action via feedback from thrombin (IIa) and thrombokinase (Xa) activating factors VIII and XI. Thus the extrinsic pathway directly achieves minimal thrombin production but does so on the order of seconds, whereas the intrinsic pathway generates large quantities of thrombin but on the order of minutes [45].

More recently, it has become clear that all three pathways interact a good deal during the clotting process. In fact, the pathway appellations are in some respects merely holdovers from early, more limited models of blood clotting; they are presently retained primarily to indicate initiation mechanism. The traditional nomenclature also prevails because it provides a way of bracketing parts of the cascade for ease of understanding, and because the distinctions still provide a valid and instructive way to conceptualize the clotting cascade. We chose to follow tradition in layout and labeling our model, but because of the interconnections between the pathways they are not modeled as separate objects in the OOP sense.

The HPN version of the coagulation cascade is depicted in Fig. (7). Places at the left hand side represent unactivated blood factors—zymogens of serine proteases, their cofactors, and various ancillary proteins, regulatory and otherwise—at pre-injury *in vivo* concentrations. The figure refers to the factors by their Roman numeral designations (if such they have); alternative names, along with initial concentrations and their sources in the literature, appear in Table I.

The boxes in the figure are obviously neither places nor transitions; they are rather objects which themselves contain Petri nets. The majority of these objects are in fact three continuous-time/state modules; the remaining two, the modules $INITintrinsic$ and $fibrin_1$, are hybrid Petri nets.

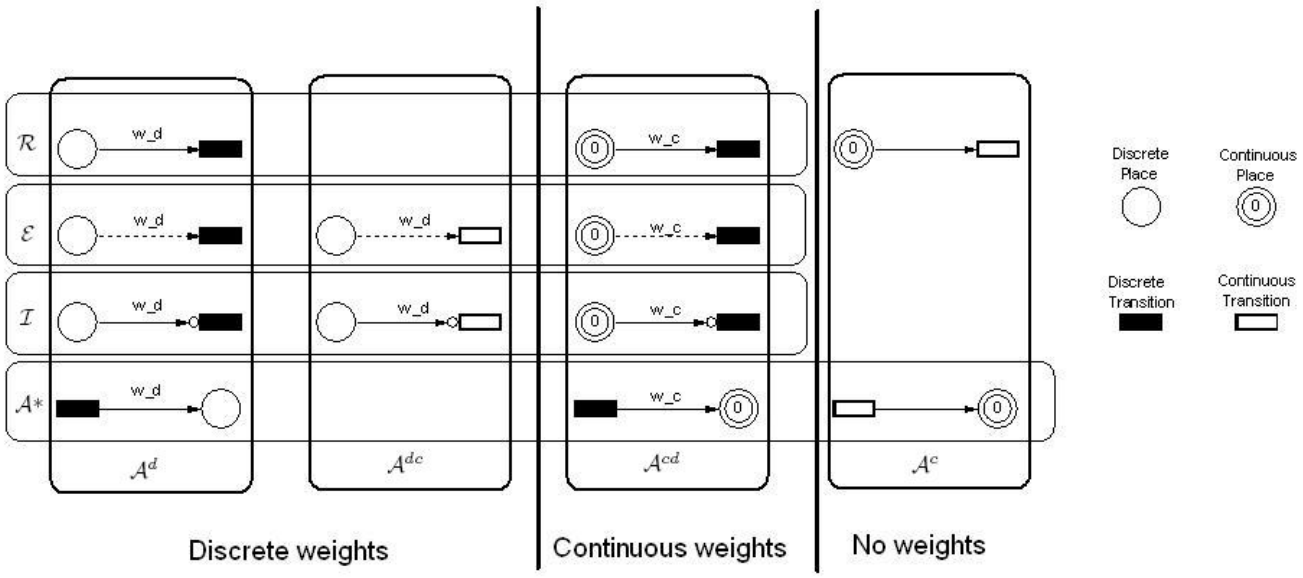

Fig. 5. A summary of the various types of arcs in an HPN; cf. Defs. (IV.6) and (IV.7). The nets in first three columns have discrete execution semantics, and those in the last column fire continuously; see Def. (IV.7) for details.

Factor (abbr.)	Trivial Name	Activated Form	Initial Conc. (nM)	Source
I	fibrinogen	fibrin (Ia)	6000-13000	[8]
II	prothrombin	thrombin (IIa)	1400	[3]
III	tissue factor	—	0.005†	[3]
IV	calcium ions	—	$1.2\text{x}10^6$	[46]
V	proaccelerin	accelerin (Va)	20	[3]
VII	proconvertin	convertin (VIIa)	10/0.1‡	[3]
VIII	antihæmophiliac factor A	VIIIa	0.7	[3]
IX	plasma thromboplastin component	IXa	90	[3]
X	Stuart-Prower factor	thrombokinase (Xa)	170	[3]
XI	plasma thromboplastin antecedent	XIa	30	[3]
XII	Hageman factor	XIIa	500	[8]
XIII	fibrin stabilizing factor	plasma transglutaminase (XIIIa)	70	[8]
prekallikrein (PK)	Fletcher factor	kallikrein	500	[8]
protein C (PC)	—	APC	60	[3]
protein S (PS)	—	—	300	[3]
antithrombin (AT)	—	—	3400	[3]
thrombomodulin (Tm)	—	—	1	[3]
tissue factor pathway inhibitor (TFPI)	aka LACI, EPI	—	2.5	[3]
high molecular-weight kininogen (HMWK)	Fitzgerald factor	—	1000	[8]

TABLE I
THE PRIMARY BLOOD COAGULATION FACTORS AND THEIR PRE-INJURY CONCENTRATIONS

C. Modules

Fig. (7) shows the entire clotting cascade. The overall model comprises multiple instances of five basic modules, with different parameters for different pathways, factors, and enzymes. There are three modules with purely continuous dynamics and two modules with hybrid dynamics. The three continuous modules involve (a) blood factor activation, (b) factor-factor binding, and (c) enzyme to lipid surface binding. The two hybrid modules involve (a) the initiation of blood clotting in the intrinsic pathway and (b) the formation of a fibrin clot. These modules are described below.

1) Blood Factor Activation: Fig. (8) depicts one of the basic aspects of the coagulation cascade, the enzyme-induced transformation of a blood factor (which may be either a serine protease or a glycoprotein) from its inactive (zymogen) form to its active configuration. In fact, the coagulation cascade consists largely of a series of such activations, where the

†Of course, no tissue factor is actually present pre-injury; this quantity represents the amount released immediately upon blood vessel rupture

‡These are the unactivated and activated concentrations, respectively, of factor VII

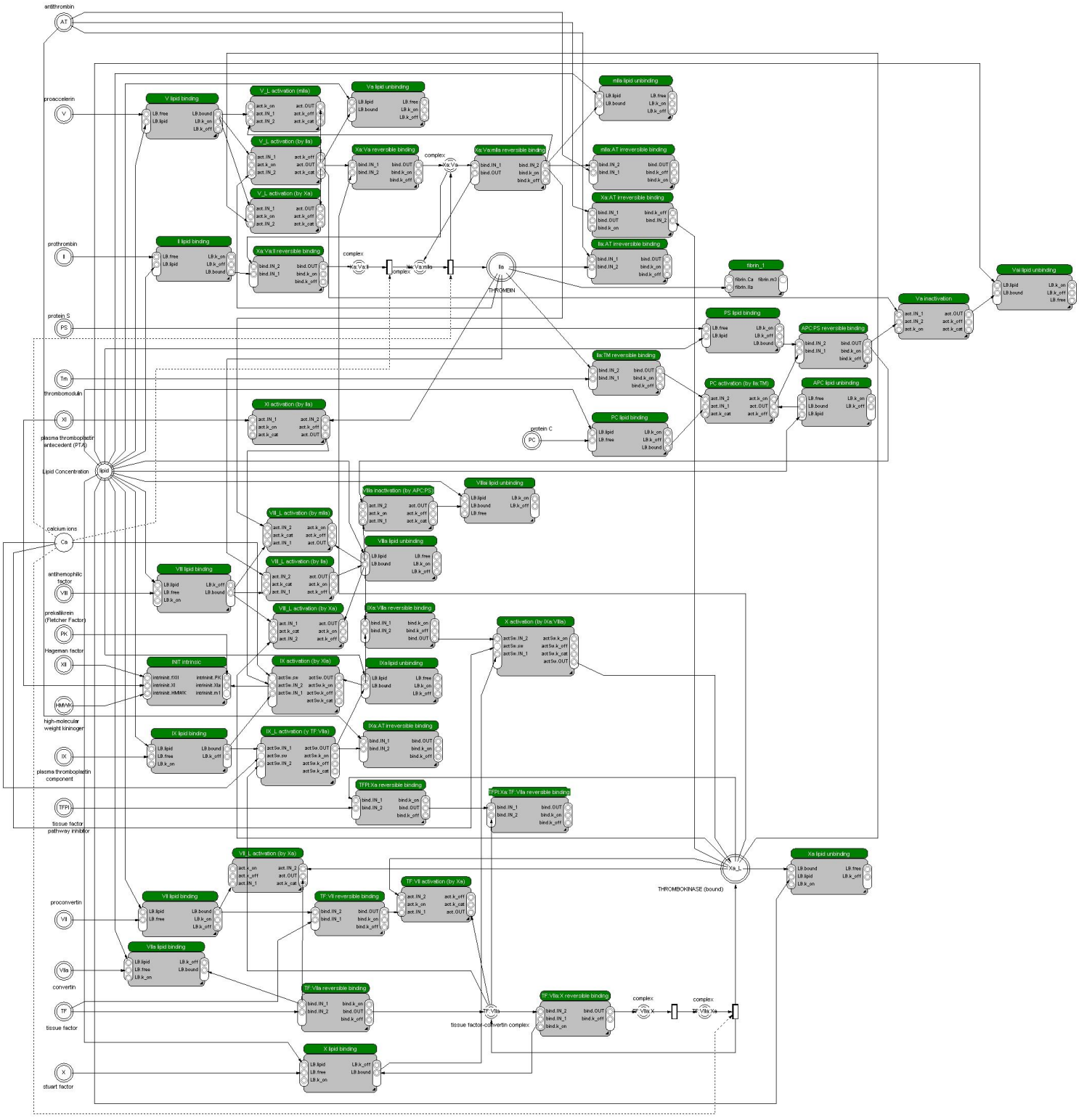

Fig. 7. The entire blood clotting process, modeled as a hybrid Petri net. The large boxes are modules.

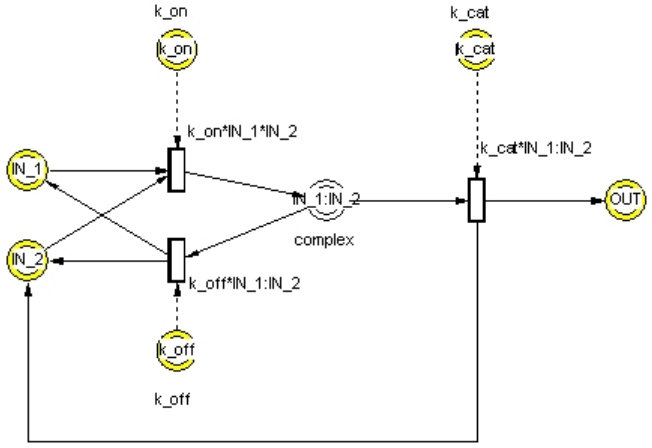

Fig. 8. The activation module, which models the activation of a zymogen (IN_1) into its active configuration (OUT) by a catalyst (IN_2).

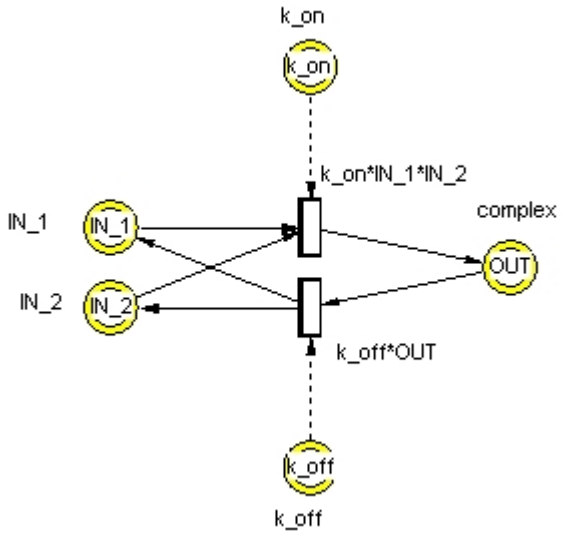

Fig. 9. The binding module, which models the (reversible) binding of two components, IN_1 and IN_2, into the macromolecule OUT.

newly activated coagulation factor proceeds to activate another factor (again see Fig. [6]). The places IN_1, IN_2, and OUT represent the concentration (in nM, though the units are not conceptually relevant) of various blood factors: IN_1 is the zymogen and OUT is its activated form; IN_2 is the catalyzing enzyme; IN_1:IN_2 is an intermediate macromolecule. The places labeled with k_i are the constants of classic enzyme kinetics: on-rates, off-rates, and catalytic rates. This reaction can also be written as a set of differential equations; square brackets are used to remind the reader that concentrations are being indicated:

$$\frac{d[IN_1]}{dt} = k_{off}[IN_1{:}IN_2] - k_{on}[IN_1][IN_2] \tag{7}$$

$$\begin{aligned}\frac{d[IN_2]}{dt} = \quad & k_{off}[IN_1{:}IN_2] - k_{on}[IN_1][IN_2] \\ & +k_{cat}[IN_1{:}IN_2]\end{aligned} \tag{8}$$

$$\begin{aligned}\frac{d[IN_1{:}IN_2]}{dt} = \quad & k_{on}[IN_1][IN_2] - k_{off}[IN_1{:}IN_2] \\ & -k_{cat}[IN_1{:}IN_2]\end{aligned} \tag{9}$$

$$\frac{d[OUT]}{dt} = k_{cat}[IN_1{:}IN_2] \tag{10}$$

Of course, since the variables IN_1, IN_2, and OUT participate in other reactions, Eqs. (7), (8), and (10) do not completely define the dynamics of any of these variables; the complete governing equations may contain additional additive terms from other reactions. That is why these three places are colored yellow in Fig. (8): it indicates that they are "published," i.e. available to interact with other objects and hence other reactions. (The rate constants are also published, but this is rather so that the user can easily change them without having to open up the object of interest.)

2) Factor-Factor Binding: The second recurring reaction is the binding of two blood factors, depicted in Fig. (9). Note that, as in the activation reaction (Fig. [8]), the rates constants k_i are connected to transitions via test arcs; this reflects the fact that these quantities are unchanged by the reaction. In the present case, the governing equations are simply:

$$\frac{d[IN_1]}{dt} = k_{off}[OUT] - k_{on}[IN_1][IN_2] \tag{11}$$

$$\frac{d[IN_2]}{dt} = k_{off}[OUT] - k_{on}[IN_1][IN_2] \tag{12}$$

$$\frac{d[OUT]}{dt} = k_{on}[IN_1][IN_2] - k_{off}[OUT] \tag{13}$$

3) Enzyme to Lipid Surface Binding: The third recurring continuous-time/state component is the binding of an enzyme to the surface of a lipid. Many of the reactions of the coagulation system require a negatively charged surface, which is provided *in vivo* by phospholipids. These molecules normally line the inner membrane of vascular walls but are exposed to blood plasma by the rupture of blood vessels. Note that the concentration of lipids is modeled as unperturbed by the reaction (wherefore the test arc); it is assumed that there is a surplus of phospholipids and hence they are not depleted by the lipid-binding reactions (Fig. [10]). For completeness, the corresponding differential equations are given as:

$$\frac{d[free]}{dt} = k_{off}[bound] - k_{on}[free][lipid] \tag{14}$$

$$\frac{d[bound]}{dt} = k_{on}[free][lipid] - k_{off}[bound] \tag{15}$$

Here $bound$ refers to the concentration of blood factor bound to a lipid surface, and $free$ refers to the concentration of unbound enzyme.

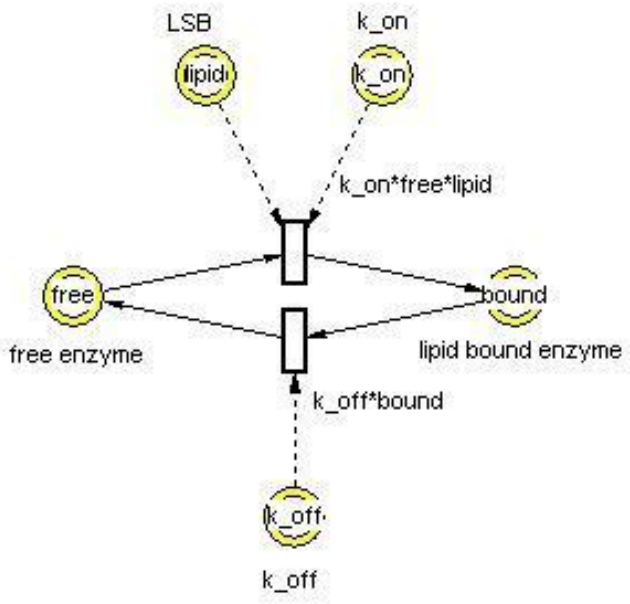

Fig. 10. The lipid binding module. Free proteins bind to the surface of an inexhaustible supply of lipids. The bound enzyme also dissociates from the lipid surface, hence the second transition.

Each of these three modules appears in multiple instantiations throughout the model, differing from each other only in their rate constants and their interconnections with the rest of the network. The differential equations that they model were drawn from [3].

4) The Intrinsic Pathway: The $INITintrinsic$ module, shown in Fig. (11), models the initiation of blood clotting via the intrinsic pathway. (Details of this process were drawn from [14] and [13].) The continuous transition labeled "zinc flow rate" models the increase in zinc concentration according to a simple first-order differential equation (exponential growth up to an asymptote). At the same time, exposure of a negatively charged surface (modeled as the binary variable m_1) allows solid-phase activation of factor XII (Hageman factor).

When [Zn] exceeds 0.3 μM, high-molecular-weight kininogen (HMWK, also known as Fitzgerald factor) binds with the plasma protein prekallikrein. The ambient zinc concententation continues to rise, meanwhile, and when it exceeds 5 μM the activation of prekallikrein to kallikrein is enabled. This process is greatly enhanced by the presence of activated factor XII, sufficient quantities of which enable activation via the "fast" transition.

Once activated, kallikrein participates in a feedback loop by enabling the fluid-phase activation of factor XII, which in turn activates more kallikrein. Notice that factor XIIa can activate kallikrein through either a "slow" or "fast" transition, where the former corresponds to activation of free prekallikrein and the latter to activation of prekallikrein bound to the surface of HMWK. The speeds of these reactions, fast and slow, are implemented by assigning appropriate time delays to the discrete transitions.

Eventually the concentration of factor XIIa crosses the threshold for the activation of factor XI (plasma thromboplastin antecedent, or PTA), generating a discrete quantity of factor XIa (given by the output arc weight). A sufficient quantity of factor XIa prevents further activation by factor XIIa, hence the inhibitor arc returning from the place XIa to the activation and binding transition. Factor XIIa production is itself inhibited by the serine protease inhibitor C1-inhibitor, which is activated by sufficient quantities of fXIIa.

5) The Fibrin Module: Fig. (12) shows the $fibrin_1$ module, which models the final portion of the blood clotting pathway: the activation of factors XIII and I, and the formation of a fibrin clot. (Data for this module were drawn from [47], [48], [49], [50], [51], and [52]; see Table II for details.) Factor XIII (fibrin stabilizing factor) normally circulates in plasma bound to fibrinogen (factor I). When thrombin concentrations exceed a threshold, the Arg37-Gly38 bond of factor XIII is cleaved, producing factor XIIIa$'$. There is again a "fast" and "slow" cleavage, modeled by transitions with identical delays but which are enabled by differently weighted test arcs from the place IIa. The fast cleavage requires lower levels of thrombin but additionally the presence of fibrin or fibrinogen. (The reader may convince himself that the boxed elements labeled "OR" in Fig. [12] do in fact enforce an or-gate of sorts.) Factor XIII can also be cleaved at the Lys513-Ser514 bond, which renders it useless with respect to clotting. This cleavege is however entirely inhibited by the presence of calcium ions (Ca^{2+}), hence the inhibitor arc from the calcium place to this transition. Meanwhile, thrombin levels rise, and when they exceed 2.48 nM, fibrinopeptide A is released from fibrinogen thereby activating it to fibrin.

In the presence either of calcium or fibrin (or both), the A$'$ and B subunits of factor XIIIa$'$ dissociate. Next, again only if calcium ions are present, the active site on the A$'$ subunits is unmasked, resulting in the transglutaminase FXIIIa*. Finally, FXIIIa and the fibrin polymers crosslink to form a clot.

The place labeled m_1 in the figure represents the percentage of cross-linking accomplished, where each token corresponds to 10%. Thus when m_1 acquires ten tokens, cross-linking is complete. Now, cross-linking is self-regulating in that it inhibits the upstream promoter effects of fibrin and fibrinogen on factor XIII activation once about 40% of the cross-linking has been accomplished [47]; hence the inhibitor arc from m1 to the "fast" transition.

Both of these HPNs made use of various thresholds, arcweights, timing delays, and binary dependencies (i.e. switches). Table II lists all of these parameters, and their sources in the literature, if such there be. Parameters which do not have literature sources were either interpolated from other relevant data or, in the limit, are informed guesses.

Finally, several of the activation reactions of the clotting cascade require free calcium ions. Fig. (13) reprises Fig. (8), except that the discrete variable sw switches the binding and catalytic reactions on and off; if the sw place holds a token (or more than one), then the reactions may take place, otherwise they may not.

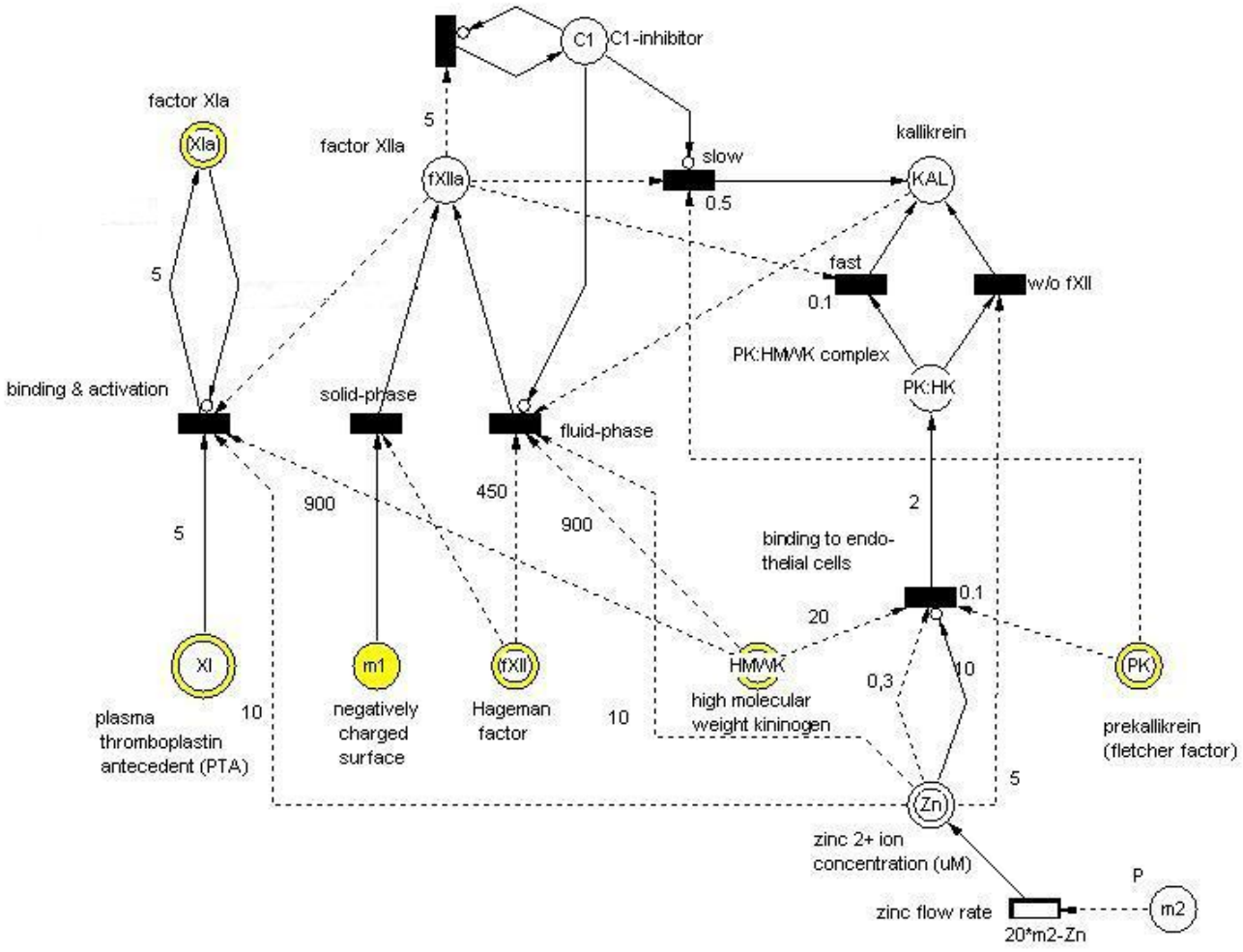

Fig. 11. A hybrid Petri net module modeling the initiation of the intrinsic pathway

Parameter	Type	Description	Value	Source
$[Zn^{2+}]$	arcweight	threshold for HMWK:PK binding	0.3μM	[14]
$[Zn^{2+}]$	arcweight	threshold for PK activation on cells	5μM	[13]
$[Zn^{2+}]$	arcweight	threshold inhibition of HMWK:PK binding	10μM	[14], [13]
$[Zn^{2+}]$	arcweight	threshold for FXII fluid-phase activtion	10μM	[14]
$[Zn^{2+}]$	arcweight	threshold for FXI:HMWK binding	10μM	[14]
Zinc flow rate	continuous transition (differential equation)	rate of zinc ion accumulation	$[\dot{Zn}] = 20 - [Zn]$	extrapolated from intrin. pathway time data
[IIa]	arcweight	threshold for fast cleavage of Arg37-Gly38 bond	1.56nM	interpolated from [51]
[IIa]	arcweight	threshold for slow cleavage of Arg37-Gly38 bond	90nM	interpolated from [51]
[IIa]	arcweight	threshold for release of fibrinopeptide A	2.48nM	interpolated from [51]
[IIa]	arcweight	threshold for release of fibrinopeptide B	3.28nM	interpolated from [51]
[XIII]	arcweight/delay	amount of fXIII cleaved per unit time:		
		fast cleavage	1nM/3s	interpolated from [51]
		slow cleavage	0.33nM/3s	interpolated from [51]
cross-linking	arcweight	percentage of cross-linking at which promoter effect of fibrin on Arg37-Gly38 cleavage is inhibited	40%	[47]

TABLE II
PARAMETER VALUES AND TYPES USED IN THE MODEL

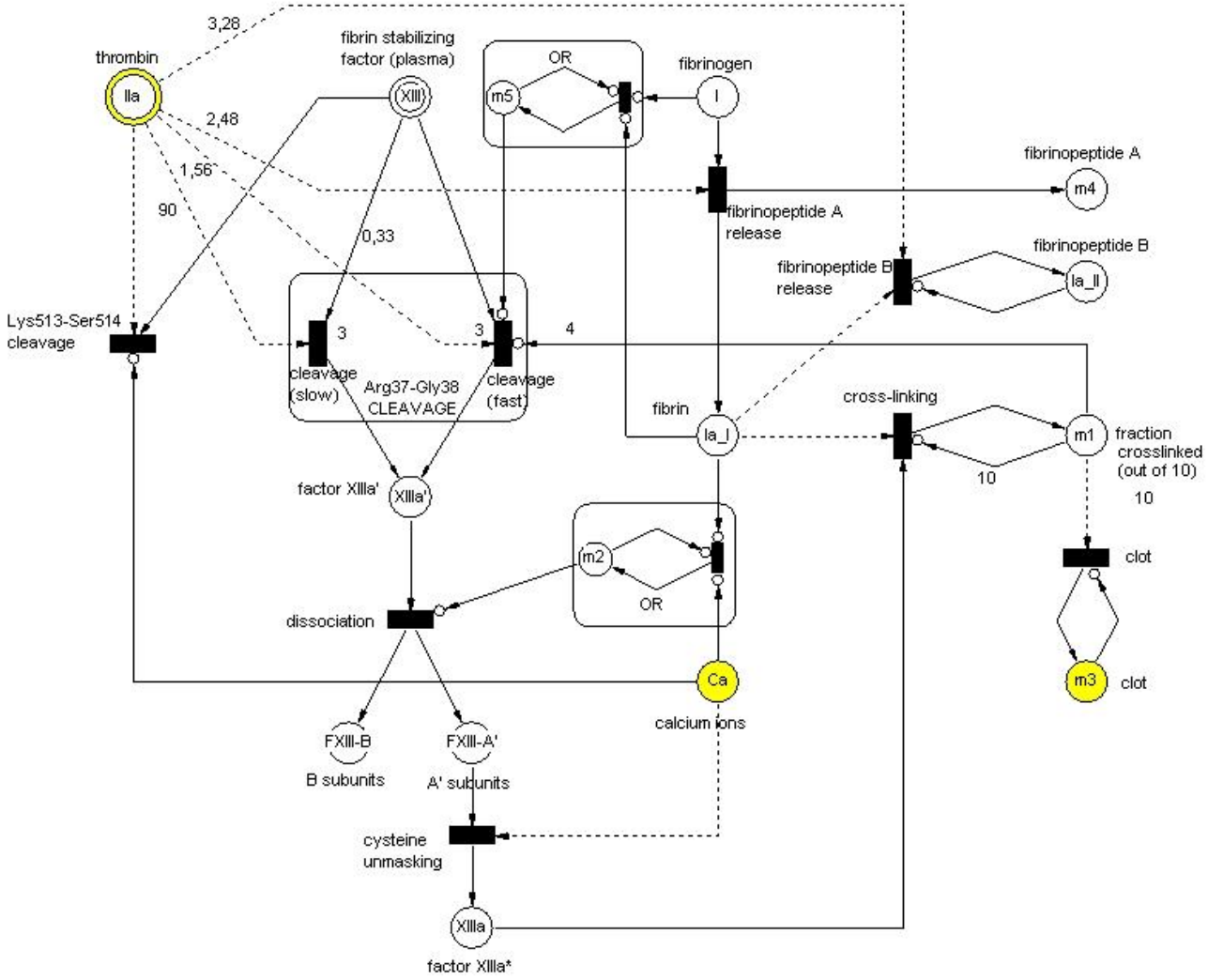

Fig. 12. A hybrid Petri net module depicting the final stages of blood clotting: the activation of factors I and XIII, and the formation of a clot

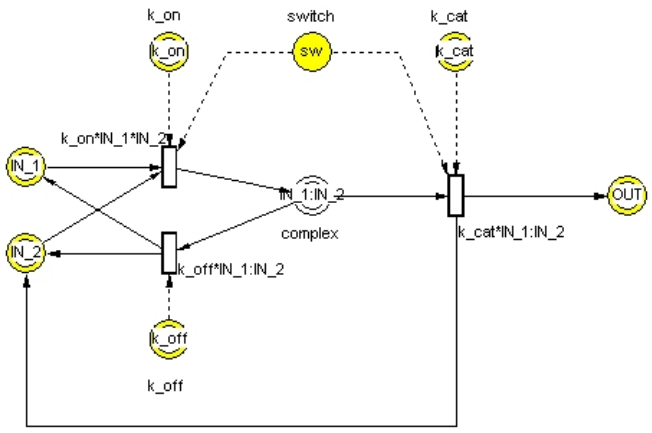

Fig. 13. The activation module of Fig. (8), but here two of the reactions are switched on (off) by the presence (absence) of a discrete variable.

6) The complete cascade: With the component modules explained, we can now turn to the entire coagulation cascade, referring to Figs. (7) and (6) throughout. To reiterate, the cascade is intiated through both the intrinsic and extrinsic pathways. The intiation of the former has been explained; the latter pathway (see the lower left corner of Fig. [7]) begins with the binding of tissue factor (TF) to lipid-bound factor VII, in both its activated and unactivated forms. (The lipid binding reactions will be taken to be self-explanatory throughout and will not be mentioned explicitly). Now, factor VII may be activated by factor Xa, but this requires completion of either the intrinsic or extrinsic pathways, both of which, as we shall see, require the activation of factor VII. Thus it is generally accepted [3] that some very small quantity of factor VIIa (0.1 nM) exists in the blood stream prior to vascular injury. This solution does not imply spontanteous activation of the coagulation system since without tissue factor the rest of the

pathway cannot proceed.

The TF:VIIa complex proceeds to bind with lipid-bound factor X (lower right corner of Fig. [7]), which complex in turn transforms (see the subsequent continuous transition) into TF:VIIa:Xa via cleavage of a bond in the factor X component. The complex then dissociates (the subsequent transition) into factor Xa (i.e., its activated form, thrombokinase) and the TF:VIIa complex. Factor Xa then feeds back to activate lipid-bound factor VII into VIIa as well as the TF:VII complex into TF:VIIa. Factor Xa also binds with a protein called tissue factor pathway inhibitor (TFPI) (middle bottom of Fig. [7]), which inhibits the extrinsic pathway by binding to free TF:VIIa complex and removing it from further reactions. Activation of factor X demarcates the traditional ending point of the extrinsic pathway.

As we have already seen in the $INITintrinsic$ module, the exposure of HMWK, kallikrein, and factor XII to an electronegative surface results ultimately in the activation of factor XI to XIa. Factor XIa can then activate factor IX to IXa (middle right of the figure), but the cascade can proceed no further until factor Xa resulting from the action of the extrinsic pathway activates lipid-bound factor VIII. This comports with the accepted theory that the extrinsic pathway serves to kick-start the intrinsic pathway into action (see above). The activated form of factor VIII (VIIIa) forms a complex with factor IXa, which in turn activates more (lipid-bound) factor X. This marks the end of the intrinsic pathway.

The common pathway of the coagulation cascade involves factors V, II, and X; plus the proteins antithrombin, thrombomodulin, protein S, and protein C (see the upper left corner of the figure). Lipid-bound factor V is activated by factor Xa. The activated form, Va, then forms a complex with Xa, which in turn binds to lipid-bound factor II. This Xa:Va:II complex spontaneously converts (via the continuous transition) to Xa:Va:mIIa, where mIIa is meizothrombin, an intermediate form of factor II. This complex then dissociates either into Xa:Va and the activated form of factor II, thrombin (via the continuous transition), or into Xa:Va and mIIa (via the Va:Xa:mIIa reversible binding module). Both meizothrombin and thrombin feed back to activate more factor V. Thrombin also feeds into the intrinsic pathway to activate more factor VIII, as does meizothrombin; and to activate factor XI.

The common pathway is inhibited by antithrombin, thrombomodulin, and proteins C and S. Thrombin binds to thrombomodulin (upper right corner), effectively removing it from further catalytic reactions. This complex has a further inhibitory role, however, in activating lipid-bound protein C. Activated protein C (APC) binds with lipid-bound protein S, and the APC:PS complex feeds back to inactivate both VIIIa in the intrinsic pathway and Va in the common pathway. Factors Xa, mIIa, and IIa (thrombin), meanwhile, are irreversibly bound by antithrombin, removing them completely from further reactions.

Thrombin's final role, of course, is to activate factors I and XIII, but this has already been detailed above in the description of the $fibrin_1$ object.

V. Results

We now present the results of various simulations. We start with the normal clotting pathway, and then simulate disorders of blood clotting as changes to either the relevant coagulation factors and associated protein complexes—i.e. the initial values of places in the network—or to the structure of the network. The results shown for the abnormal cases are for thrombin levels, but of course simulations produce time course results for all the places in the network. Thus the impact of specific diseases or combinations of diseases on the entire clotting cascade may be examined. This includes both qualitative and quantitative aspects, like coagulation with versus without therepeutic intervention, and like the effect of specific dosage levels of an intervention, respectively.

A. Normal Clotting

We choose to examine thrombin because it is the most important enzyme product of the coagulation cascade: it participates in far more reactions than any of the other factors, including both feedforward and feedback regulation, and is essential for normal blood clotting. Time to thrombin activation is consequently one of the major parameters measured in clinical tests. To evaluate the baseline performance of our model, we set the initial concentrations as shown in Table I and compared the time course of thrombin production in our computational simulation with results reported in the clinical literature [3], [8]. Fig. (14) shows the concentration of thrombin produced under normal conditions upon triggering of the clotting cascade. The value of thrombin is shown in nM, as a function of time, in seconds. Consistent with previously reported clinical studies, thrombin concentration using the model peaks at about 160 nM around 100 seconds (cf. [53]). A complete clot (i.e. 100% cross-linking) occurs after 105 seconds, which is again a reasonable figure.

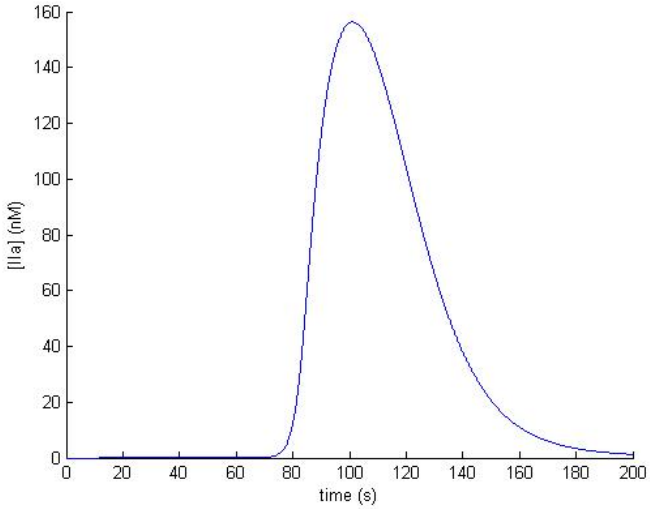

Fig. 14. A simulation of the concentration of thrombin (factor IIa) in nM as it varies with time (in seconds) in the blood clotting of a normal patient.

Condition	Cause	Result	Notes
Hæmophilia A	deficiency of fVIII	excessive bleeding	severity: $> 5\%$ (mild) $1 - 5\%$ (moderate) $< 1\%$ (severe)
Factor V Leiden	APCR in fV	hypercoag.	APC resistance $\approx 5\%$ of population

TABLE III
COAGULATION DISORDERS

B. Simulating Hæmophilia A

Now consider a simulation of hæmophilia A, a disease of the clotting system which results in excessive bleeding. The cause of the pathology is a deficiency of coagulation factor VIII, which can range from mild to severe (see Table III).

To simulate the clotting disorder, therefore, the continuous place VIII, representing the initial (pre-injury) plasma concentration of clotting factor VIII, is set to 0.035 nM, which corresponds to the borderline between mild and moderate hæmophila (cf. Tables I and III), and the simulation is re-run. The result is shown in Fig. (15): thrombin concentration peaks later (at 120 seconds) and much lower (at just over 6 nM), which is congruent with clinical observations. The maximum cross-linking acheived is 40%, which again is consistent with the disease pathology.

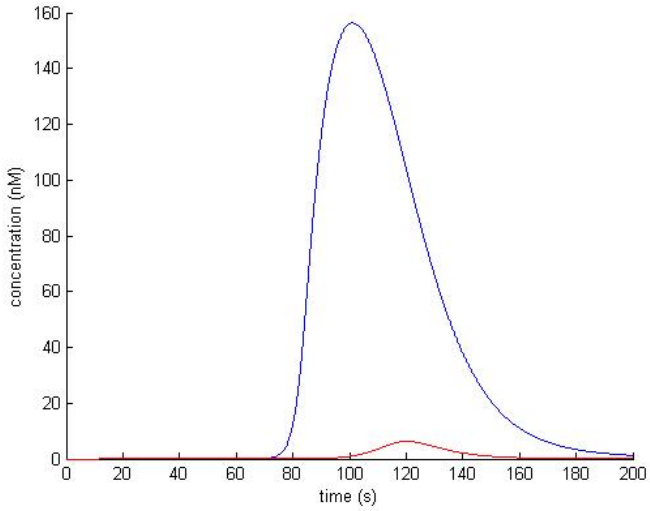

Fig. 15. Simulated results of the concentration (nM) of thrombin versus time (s) during the clotting process for a person with severe hæmophilia A (red) and for normal clotting (blue).

C. Simulating Factor V Leiden

Factor V Leiden is a coagulation disorder characterized by a condition called activated protein C resistance (APCR), in which a genetic mutation in the factor-V gene renders the resulting factor-V protein resistant to inactivation by activated protein C (APC). (Recall, following Fig. [6], that the function of APC is to inactivate factor Va and factor VIIIa.) Factor V is a procoagulant, so the consequence of its slower rate of inactivation is generally a thrombophilic (propensity to clot) state. The disorder has in fact only recently been documented: APCR was first described in 1993; factor V Leiden was subsequently discovered in 1994.

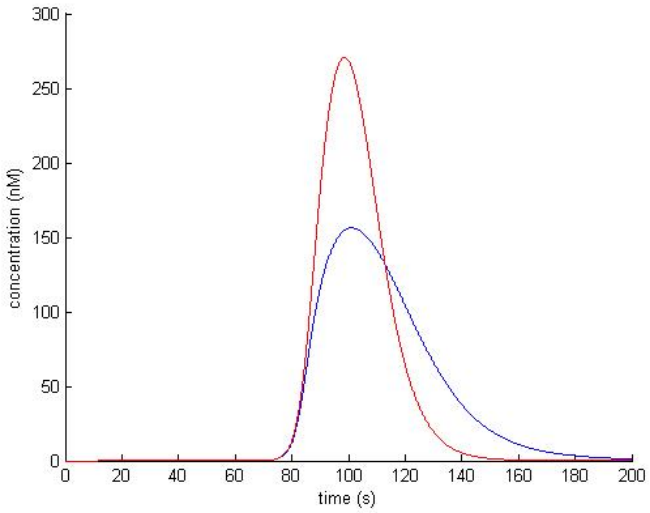

Fig. 16. Simulated results of the concentration (nM) of thrombin versus time (s) during the clotting process for a person with factor V Leiden (red) and a normal patient (blue).

More sophisticated ways of simulating factor V Leiden are available within the present model, but for simplicity we choose simply to remove the module by which APC inactivates factor V (see §IV-C6). As expected, the result (Fig. [16]) is an increase in the amount of thrombin produced, congruent with clinical observations (cf., e.g., [54]).

VI. DISCUSSION

The modeling framework and software program described provides a robust, faithful, interactive, and graphical computer simulation of the entire coagulation process. The framework is faithful in that it accurately models mammalian blood clotting; robust in the sense of doing so over a wide range of parameter settings; interactive in that the system operation and parameter settings can be interactively changed while the software program is executing, and hypothetical "what-if" simulations performed; and graphical in that the model is a formal graphical structure that supports visualization of the clotting process as well as exact quantitative analysis. Viewing the coagulation cascade as a hybrid system allows us to tap into the growing set of tools and analysis techniques available from this active area of current research. Although the present study focused on simulation, results in areas of reachability analysis and controller synthesis are relevant to future work (see below).

The simulations of factor V Leiden and hæmophilia A were consistent with clinical results, and to that extent vindicate both the present model and its parameters as well as the methodology, i.e. a HS approach. Of course, consistency is not tantamount to quantitative identity, but here the obstacle

lies not with the model so much as the state of the clinical data. The authors are unaware of any exact data for *in vivo* thrombin concentration time courses for any blood diseases. This is no doubt due in part to the very dearth of quantitative data which this model is intended to help clinicians redress.

Since the present model covers the entire clotting cascade, it is—unlike other models—theoretically capable of simulating both of the major clinical blood clotting tests, activated partial thromboplastin time (aPTT) and prothrombin time (PT). These tests are *in vitro* measures of clotting, and have slightly different dynamics, but in both cases the parameter of interest is the time to clotting, which is simulated in the model. We intend to explore simulations of these tests in a subsequent paper.

We are currently attempting to use the model to investigate the impact of potential therapeutic interventions to patients with hyper- and hypocoagulatory diseases. Some of the conditions that can be readily studied include the various manifestations of hæmophilia A, which results from deficiencies, moderate to severe, of factor VIII; hæmophilia B, which is a consequence of factor IX deficiency; and the hypercoagulation disorder antithrombin deficiency, which affects about 0.02% of the population [55]. Other blood-clotting diseases can be modeled as mutated forms of clotting factors; patients suffering from factor V Leiden, for example, are prone to thrombophilia (excessive coagulation) because their factor V proteins are resistant to inactivation by activated protein C. The framework is open and modifiable to incorporate new pathologies and study their impact on clotting processes.

The authors plan to release the present model of the blood clotting pathway as a software package with options for simulating a variety of blood clotting disease pathologies and associated therapeutic interventions already installed. [§] The blood clotting software will be usable off-the-shelf and users will also be able to add components and modules to the model in an "object-oriented" manner, in which components are designed for maximum reuse of existing structure and functionality. We expect such a framework to have multiple uses. Clinicians could use it to decide on treatment choices, including dosage levels for specific diseases and for specific patients, through computer simulation and analysis, potentially lessening the need for clinical trials. The model may also provide a tool for researchers in the field to refine their understanding of the complex blood-clotting process: parameters, equations, and model structure are easily modified, adding new reactions is straightforward, and simulations provide detailed information about (*inter alia*) the concentrations of clotting factors with time. In addition, the framework can support research, exchange, and dissemination of information (with common data and model formats) on the impact of disease pathologies (including combinations of diseases) that are caused by a deficiency or surfeit of blood factors.

[§]The current model uses the commercially available Visual Object Net++ software [9]. We have implemented an open-source HS framework in a JAVA-based environment and are migrating the blood clotting model to this environment.

Finally, the model poses questions whose answers lie in other areas of HS theory: verification and controller synthesis. In particular, blood factor concentrations may be thought of from a reachability perspective; one may ask, for instance, whether a certain (say, hyper- or hypocoagulatory) state may be reached given the initial concentrations and HPN marking. The present rather vague criteria for thrombophilia or hæmophilia might then be replaced with quantitative safety boundaries. Furthermore, pharmacological intervention may be conceived along the lines of controller verification: given a set of states, i.e. a set of safe ranges for blood factor concentrations, the appropriate dosage levels of the intervention (e.g. heparin, Warfarin, etc.) can be calculated as the inputs from a controller which will keep the system within the safe set. Such algorithms exist for HS.

REFERENCES

[1] M. Panteleev, V. Zarnitsina, and F. Ataullakhanov, "Tissue factor pathway inhibitor - a possible mechanism of action," *European Journal of Biochemistry*, vol. 269, pp. 2016–2031, 2002.

[2] A. Kogan, D. Kardakov, and M. Khanin, "Analysis of the activated partial thromboplastin time test using mathematical modeling," *Thrombosis Research*, vol. 101, pp. 299–310, 2001.

[3] S. Bungay, P. Gentry, and R. Gentry, "A mathematical model of lipid mediated thrombin generation," *Mathematical Medicine and Biology*, vol. 20, pp. 105–129, 2003.

[4] R. Leipold, T. Bozarth, A. Racanelli, and I. Dicker, "Mathematical model of serine protease inhibition in the tissue factor pathway to thrombin," *The Journal ob Biological Chemistry*, vol. 270, no. 43, pp. 25 383–25 387, 1995.

[5] Y. Qiao, C. Xu, Y. Zeng, X. Xu, H. Zhao, and H. Xu, "The kinetic model and simulation of blood coagulation–the kinetic influence of activated protein c," *Medical Engineering and Physics*, vol. 26, pp. 341–347, 2004.

[6] S. Butenas, T. Orfeo, M. Gissel, K. Brummel, and K. Mann, "The significance of circulating factor ixa in blood," *The Journal of Biological Chemistry*, vol. 279, no. 22, pp. 22 875–22 882, 2004.

[7] B. Pohl, C. Beringer, M. Bomhard, and F. Keller, "The quick machine-a mathematical model for the extrinsic activation of coagulation," *Hæmostasis*, vol. 24, pp. 325–337, 1994.

[8] T. Halkier, *Mechanisms in blood coagulation, Fibrinolysis, and the Complement System*. Cambridge, England: Cambridge University Press, 1991, translation from Danish into English by Paul Woolley.

[9] R. Drath, "Description of hybrid systems by modified petri nets," in *Modelling, Analysis, and Design of Hybrid Systems*, ser. Lecture Notes in Control and Information Sciences (LNCIS). Springer-Verlag, July 2002, vol. 279, pp. 15–36.

[10] T. Adams, S. Everse, and K. Mann, "Predicting the pharmacology of thrombin inhibitors," *Journal of Thrombosis and Hæmostatis*, vol. 1, pp. 1024–1027, 2002.

[11] C. Xu, Y. Zeng, and H. Gregersen, "Dynamics model of the role of platelets in the blood coagulation system," *Medical Engineering and Physics*, vol. 24, pp. 587–593, 2002.

[12] F.Ataullakhanov, G. Guria, V. Sarbash, and R. Volkova, "Spatiotemporal dynamics of clotting and pattern formation in human blood," *Biochimica et Biophysica Acta*, vol. 1425, pp. 453–468, 1998.

[13] R. Røjkjær and A. Schmaier, "Activation of the plasma kallikrein/kinin system on endothelial cell membranes," *Immunopharmacology*, vol. 43, pp. 109–114, 1999.

[14] Z. Shariat-Madar, F. Mahdi, and A. Schmaier, "Assembly and activation of the plasma kallikrein system: a new interpretation," *International Immunopharmacology*, vol. 2, pp. 1841–1849, 2002.

[15] J. Lygeros, "Lecture notes on hybrid systems," Feb. 2004, for the Department of Electrical and Computer Engineering, University of Patras. [Online]. Available: http://robotics.eecs.berkeley.edu/~sastry/ee291e/lygeros.pdf

[16] S. Engell, S. Kowalewski, C. Schulz, and O. Stursberg, "Continuous-discrete interactions in chemical processing plants," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 1050–1068, 2000.

[17] O. Czogalla, R. Hoyer, and U. Jumar, "Modelling and simulation of controlled road traffic," in *Modelling, Analysis, and Design of Hybrid Systems*, ser. Lecture Notes in Control and Information Sciences, S. Engell, G. Frehse, and E. Schnieder, Eds. Springer-Verlag, 2002, vol. 279, pp. 419–435.

[18] R. Horowitz and P. Varaiya, "Control of an automated highway system," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 913–925, 2000.

[19] M. Oishi, I. Mitchell, A. Bayen, C. Tomlin, and A. Degani, "Hybrid verification of an interface for an automatic landing," in *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, NV, Dec. 2002.

[20] R. Fierro, A. Das, V. Kumar, and J. Ostrowski, "Hybrid control of formations of robots," in *Proceedings of the 2001 IEEE Conference on Robotics and Automation*, vol. 1, Seoul, Korea, May 2001, pp. 157–162.

[21] T. Schlegl, M. Buss, and G. Schmidt, "Hybrid control of multi-fingered dexterous robotic hands," in *Modelling, Analysis, and Design of Hybrid Systems*, ser. Lecture Notes in Control and Information Sciences, S. Engell, G. Frehse, and E. Schnieder, Eds. Springer-Verlag, 2002, vol. 279, pp. 437–465.

[22] S. Pettersson and B. Lennarton, "Stability analysis of hybrid systems a gear-box application," in *Nonlinear and Hybrid Systems in Automotive Control*, S. Engell, G. Frehse, and E. Schnieder, Eds. Springer-Verlag, 2003, ch. 17, pp. 373–389.

[23] P. Antsaklis and X. Koutsoukos, "Hybrid systems: review and recent progress," in *Software-Enable Control: Information Technologies for Dynamic Systems*, T. Samad and G. Balas, Eds. IEEE Press, 2003, pp. 273–298.

[24] S. Neuendorffer, "Modelling real-world control systems: beyond hybrid systems," in *Proceedings of the 2004 Winter Simulation Conference*, R. Ingalls, M. Rossetti, J. Smith, and B. Peters, Eds. Washington DC, USA: IEEE Press, Dec. 2004.

[25] R. Ghosh and C. Tomlin, "Lateral inhibition through delta-notch signaling: a piecewise affine hybrid model," in *Proceedings of the 4th International Workshop on Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, M. Di Benedetto and A. Sangliovanni-Vincentelli, Eds., vol. 2034, no. XIV. Rome, Italy: Springer-Verlag, Mar. 2001, pp. 232–246.

[26] M. Chen and R. Hofestädt, "Quantitative petri net model of gene regulated metabolic networks in the cell," *In Silico Biology*, vol. 3, no. 3, pp. 347–365, 2003.

[27] H. Matsuno, A. Doi, N. M, and S. Miyano, "Hybrid petri net representation of gene regulatory network," in *Proceedings of the Pacific Symposium on Biocomputing (PSB)*, ser. Lecture Notes in Computer Science, R. Altman, K. Dunker, L. Hunter, T. Klein, and K. Lauderdale, Eds., vol. 5. Honolulu, Hawaii: World Scientific Press, Jan. 2000, pp. 338–349.

[28] H. Matsuno, Y. Tanaka, H. Aoshima, A. Doi, M. Matsui, and S. Miyano, "Biopathways representation and simulation on hybrid functional petri net," *In Silico Biology*, vol. 3, no. 3, pp. 389–404, 2003.

[29] H. Jong, J. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselmann, "Hybrid modeling and simulation of genetic regulatory networks: a qualitative approach," in *Proceedings of the 6th International Workshop on Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, A. Pnueli and O. Maler, Eds., vol. 2623. Prague, Czech Republic: Springer-Verlag, Apr. 2003, pp. 267–282.

[30] S. Kowalewski, "Introduction to the analysis of hybrid systems," in *Modelling, Analysis, and Design of Hybrid Systems*, ser. Lecture Notes in Control and Information Sciences, S. Engell, G. Frehse, and E. Schnieder, Eds. Springer-Verlag, 2002, vol. 279, pp. 153–171.

[31] T. Henzinger, P. Kopke, A. Puri, and P. Varaiya, "What's decidable about hybrid automata?" *Journal of Computer and Systems Sciences*, vol. 57, no. 1, pp. 94–124, 1998.

[32] J. Lygeros, S. Sastry, and C. Tomlin, "The art of hybrid systems," July 2001, textbook on hybrid systems, forthcoming.

[33] R. Alur *et al.*, "CHARON," Department of Computer and Information Science, University of Pennsylvania. [Online]. Available: http://www.cis.upenn.edu/mobies/charon/

[34] A. Chutinan *et al.*, "Checkmate," Department of Electrical and Computer Engineering, Carnegie Mellon University. [Online]. Available: Homepage:http://www.ece.cmu.edu/~webk/checkmate

[35] T. Dang and O. Maler, "d/dt - reachability analysis of continuous and hybrid systems," VERIMAG. [Online]. Available: http://www-verimag.imag.fr/~tdang/ddt.html

[36] T. Henzinger, P. Ho, and H. Wong-Toi, "Hytech," Department of EECS, University of California, Berkeley. [Online]. Available: http://www-cad.eecs.berkeley.edu/~tah/HyTech/

[37] S. Miyano *et al.*, "Genomic object net," Human Genome Center, Miyano Laboratory. [Online]. Available: http://www.genomicobject.net/

[38] E. Lee *et al.*, "Hyvisual," Department of EECS, University of California, Berkeley. [Online]. Available: http://ptolemy.eecs.berkeley.edu/hyvisual/

[39] "MATLAB," The MathWorks, Inc. [Online]. Available: http://www.mathworks.com

[40] C. Daws *et al.*, "KRONOS," VERIMAG. [Online]. Available: http://www-verimag.imag.fr/TEMPORISE/kronos/

[41] L. de Moura *et al.*, "Symbolic analysis laboratory (sal)," SRI International. [Online]. Available: http://sal.csl.sri.com/

[42] A. Deshpande *et al.*, "SHIFT," California PATH. [Online]. Available: http://www-path.eecs.berkeley.edu/shift/

[43] K. Larsen *et al.*, "UPPAAL," Uppsala University, Sweden and Aalborg University, Denmark. [Online]. Available: http://www.uppaal.com/

[44] J. Signorini and P. Gruessay, "Object-oriented specification of complex bio-computing processes: a case study of a network of proteolytic enzymes," in *Biologically Inspired Approaches to Advanced Information Technology*, ser. Lecture Notes in Computer Science, A. Ijspeert, M. Murata, and N. Wakamiya, Eds., no. XIV. Lausanne, Switzerland: Springer-Verlag, Jan. 2004, pp. 1–12.

[45] I. H. Viera Štvrtinová, J. Jakubovský, *Inflammation and Fever*. Academic Electronic Press, 1995. [Online]. Available: http://www.savba.sk/logos/books/scientific/Inffever.html

[46] D. Goltzman, "Approach to hypercalcemia, hhm/pthrp, and treatment," in *Diseases of Bone and Calcium Metabolism*, A. Arnold, Ed. MDText.com, Inc, 2005, ch. 4. [Online]. Available: http://www.endotext.org/parathyroid/parathyroid4/parathyroidframe4.htm

[47] L. Muszbek, V. Yee, and Z. Hevessy, "Blood coagulation factor xiii: structure and function," *Thrombosis Research*, vol. 94, pp. 271–305, 1999.

[48] C. Greenberg, C. Miraglia, F. Rickels, and M. Shuman, "Cleavage of blood coagulation factor xiii and fibrinogen by thrombin during in vitro clotting," *The Journal of Clinical Investigation*, vol. 75, pp. 1463–1470, May 1985.

[49] K. Standeven, R. Ariëns, and P. Grant, "The molecular physiology and pathology of fibrin structure/function," *Blood Reviews*, 2005.

[50] K. Lewis, D. Teller, J. Fry, G. Lasser, and P. Bishop, "Crosslinking kinetics of the human transglutaminase, factor xiii[a2], acting on fibrin gels and γ-chain peptides," *Biochemistry*, vol. 36, pp. 995–1002, 1997.

[51] K. Brummel, S. Paradis, S. Butenas, and K. Mann, "Thrombin functions during tissue factor-induced blood coagulation," *Blood*, vol. 100, no. 1, pp. 148–152, 2002.

[52] M. Mosesson, "Fibrinogen functions and fibrin assembly," *Fibrinolysis & Proteolysis*, vol. 14, no. 2, pp. 182–186, 2000.

[53] A. Undas, K. Brummel, J. Musial, K. Mann, and A. Szczeklik, "Pl^{A2} polymorphism of β_3 integrins is associated with enchanced thrombin generation and impaired antithrombotic action of aspirin at the site of microvascular injury," *Circulation*, vol. 104, pp. 2666–2672, 2001.

[54] C. van't Veer, M. Kalafatis, R. Bertina, P. Simioni, and K. Mann, "Increased tissue factor-intiated prothrombin activitation as a result of the $\text{Arg}^{506} \rightarrow$ Gln mutation in factor V^{LEIDEN}," *The Journal of Biological Chemistry*, vol. 272, no. 33, pp. 20 721–20 729, 1997.

[55] N. Manchanda *et al.*, "Hemotology resource page," University of Illinois – Urbana/Champaign. [Online]. Available: http://www-admin.med.uiuc.edu/hematology/