Modeling Biological Reactivity: Statecharts vs. Boolean Logic

Naaman Kam
Dept. of Computer Science & Applied Math. and Dept. of Immunology
The Weizmann Institute of Science
Rehovot 76100, Israel
kam@wisdom.weizmann.ac.il

David Harel
Dept. of Computer Science and Applied Math.
The Weizmann Institute of Science
Rehovot 76100, Israel
harel@wisdom.weizmann.ac.il

Irun R. Cohen
Dept. of Immunology
The Weizmann Institute of Science
Rehovot 76100, Israel
irun.cohen@weizmann.ac.il

ABSTRACT

Remarkable progress in various fields of biology is leading in the direction of a complete map of the building blocks of biological systems. There is broad agreement among researchers that 21st century biology will focus on attempting to understand how component parts collaborate to create a whole. It is also well agreed that this transition of biology from *identifying* the building blocks *(analysis)* to *integrating* the parts into a whole *(synthesis)* should rely on the language of mathematics. In a recent publication, we described the results of a first attempt at confronting the above challenge using the *visual formalism* of *statecharts*. We presented a detailed model for T cell activation using statecharts within the general framework of *object-oriented modeling*. In this work, we compare the statechart-based modeling approach to a *Boolean formalism* presented by Thomas & D'Ari. This comparison was done by taking a model for T cell activation and anergy, which was constructed by Kaufman et al. using such a Boolean formalism, and translating it into the language of statecharts. Comparing these two representations of the same phenomena allows us to assess the advantages and disadvantages of each modeling approach. We believe that the results of this work, together with the results of our previous modeling work on T cell activation, should encourage the use of visual formalisms such as statecharts for modeling complex biological systems.

1. INTRODUCTION

There is broad agreement among researchers that biological research must prepare for the transition from *analysis* (the reduction of observations to elemental building blocks) to *synthesis* (the integration of the parts into a whole) [1], and that this transition should rely heavily on the language of mathematics [2-4]. This is particularly needed in the field of immunology. Over the last several decades there has been an explosion of experimental data describing the cellular and molecular components that are involved in the activity of the immune system. "At the present time, however, there is an emerging need to understand the system as it functions as a whole" [5].

An illustrative example for the current situation in immunological research is the case of *cytokine networks*. Cytokines are small protein or glycoprotein messenger molecules that convey information from one cell to another. Various aspects of the immune response are regulated by cytokine networks. More than 200 cytokines have now been identified, and more than 12,000 papers on cytokines have been published in 1999 alone. Yet, although many details of particular cytokine interactions have been elucidated, "practically nothing is known about the behavior of the network as a whole" [6].

In search for an appropriate mathematical language for modeling biological systems, we examined formal modeling methods in computers that were originally developed for *system design*. Software and system engineers need formal models that faithfully represent the system under design. Instead of constructing models that represent only fragments of the system, which are useful for investigating a limited number of questions, formal modeling methods in computer science enable the construction of comprehensive models for complex *reactive systems*[1]. Such models are useful for investigating the behavior of the system under many given scenarios (e.g., what will happen if button A is pressed while signal X is still being processed, handle H is pulled down and the temperature is 25°?), as well as for checking the validity of various properties (e.g., verifying that under no circumstances will the doors of an elevator open while the elevator is between two floors).

In a previous paper [8], we presented a model for T cell activation, using the visual formalism of *statecharts* [9], as implemented in the general framework of object-oriented modeling [10-11]. We aimed at constructing a model that will include essentially all the relevant immunological data as presented in a standard textbook [12] and some additional updated reviews. The model included more than 20 distinct object classes, for which both static properties and dynamic behavior were described in a diagrammatic manner. Having such a formal model in hand, we were able to run simulations of the model, using the *Rhapsody* tool [13]. The model was executed for several different immunological scenarios [8]. By and large, the behavior we encountered for most executions followed our expectations, except of one case: we found that in our model, the T cells could not reach a stable memory state. It was found that this mismatch with experimental data was due to some biological piece of information that did not appear in the scientific literature on which the initial model was based. In this

[1] For the definition of reactive systems see [7]; its implications for biological systems is discussed in [8].

specific case, we found the missing part of the puzzle after an extensive search in the literature, and revised the model accordingly. However, it may very well be the case that other such mismatches will need new experimental work to be solved. This is one of the major purposes of modeling: "one of the beauties of mathematical modeling is that it raises questions that may not have been addressed before" [6].

In the present paper, we carry out a comparison between the statechart-based modeling approach and an alternative modeling approach, based on a Boolean formalism presented by Thomas & D'Ari [14]. In this context, we examined a recent model for T cell activation that was constructed by Kaufman et al. using such a Boolean formalism [15] and constructed an analogous model using the visual formalism of statecharts. In Sections 2 and 3 we describe the logical model and the statechart-based model, respectively. In Section 4 we compare the two modeling approaches. Section 5 provides some concluding remarks.

2. THE BOOLEAN LOGIC MODEL

2.1 Biological background

The engagement of the T cell antigen receptor (TCR) with its cognate ligand can lead either to T cell activation or to the induction of a state of unresponsiveness termed anergy. A large body of data indicates that several anergy induction pathways may exist [16]. One of these pathways involves a transient T cell activation that precedes the induction of unresponsiveness. In their work [15], Kaufman et al aimed at exploring this form of "activation-induced anergy", placing special emphasis on specifying the conditions leading to positive and negative signaling of T cells: "Using a simple Boolean formalism developed by Thomas and coworkers [14], we show how the timing of the signaling events may affect decision making at branching pathways and may determine the properties of receptor signaling and final state of the system" [15].

The model suggested by Kaufman et al is based on several experimental observations and assumptions (all quotations are taken from [15]):

- "An important feature of the early events of the TCR signaling cascade is the activation of protein tyrosine kinase (PTK) enzymes, endowed with catalytic activity… we assume here that after stimulation, the phosphorylative activity of the receptor associated PTK's increases and may remain above background level after the ligands have been removed".
- "In addition to its positive role in T cell activation, we postulate that tyrosine phosphorylation also mediates a supressive effect on the signaling events, which results in the inhibition of lymphokine secretion and cell proliferation. In particular, residual PTK activity after ligand dissociation may be responsible for a defective signal transduction capacity of the TCR system".
- Costimulation is required for proper T cell activation. "In the present model, costimulation does not act to suppress the negative signal but rather enhances and accelerates the positive signaling process".
- "Positive signaling (i.e., IL-2 secretion and cell proliferation) inhibits, through a yet undefined mechanism, the persistent activity of the PTKs, thus reestablishing the signaling capacities of the TCR system".

2.2 Logical Description

The first stage of constructing the logical model was establishing an interaction diagram, consisted of "a series of events, each requiring a characteristic time to be realized". The model shown in Fig. 1 uses three graphical notations: rectangles represent events, solid arrows marked with a plus sign represent interaction between events (i.e., one event favors or activates the other) and dashed arrows marked with a minus sign represent negative interactions between events (i.e., one event suppresses the other). According to this interaction diagram, the event of *Free ligand* (f) favors the event of *Bound TCRs* (b) which by its own activates the PTK. *PTK activity* (k) accelerates itself, activates the *Inhibitory pathway* (x) and gets suppressed by *Positive signaling* (s). *Positive signaling* is favored by *PTK activity* and *Costimulation*, and suppressed by the *Inhibitory pathway*.

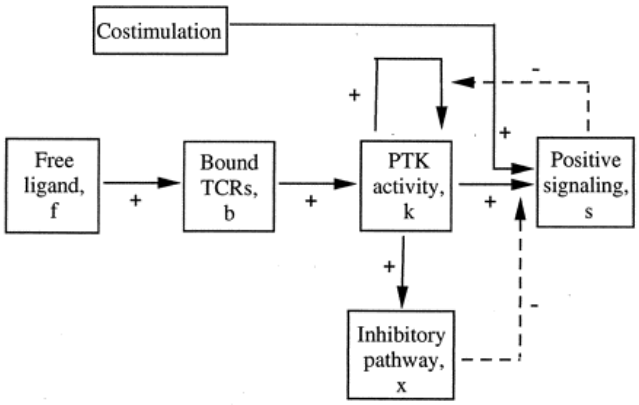

Figure 1: Schematic interaction diagram (adopted from [15])

Logical variables and logical equations:
Four logical variables, which represent four of the above described events, define the state of the system. The variables are **b**, **k**, **x** and **s**, representing *Bound TCRs, PTK activity, Inhibitory pathway and Positive signaling*, respectively (*Free ligand* and *Costimulation* are not accounted for explicitly). Assigning the value of 1 to a variable means that the event that it represents is taking place. Thus b=1 means that the TCR is bound to a ligand, otherwise b=0. Similarly, x=1 means the *Inhibitory pathway* is active, otherwise x=0.

The dynamics of the system is governed by four logical functions, B, K, X and S, attached to each of the four logical variables b, k, x and s, respectively. These functions describe the transition rules that reflect the "evolution of the state variables under the influence of the signaling interactions". A logical function gives, at any time, the future value toward which its associated variable will tend as a function of the present state of the system. For example, the function B=0 implies that no matter what is the current value of the variable b, at the next step it will receive the value 0. Similarly, the function X=k implies that the future value of the variable x is determined by the current value of the variable k.

The following logical functions were defined for the model ($\bar{x}$ denotes NOT x, $z.y$ denotes z AND y, and $z + y$ denotes z OR y):

[1] B = 0

[2] K = b + k.$\bar{s}$

[3] X = k

[4] S = b.k.$\bar{x}$

State table:

Table 1. State table for Eqs. **1–4**

b	k	x	s	B	K	X	S
(0	0	0	0)	0	0	0	0
0	0	0	$\bar{1}$	0	0	0	0
0	0	$\bar{1}$	$\bar{1}$	0	0	0	0
0	0	$\bar{1}$	0	0	0	0	0
(0	1	1	0)	0	1	1	0
0	$\bar{1}$	1	$\bar{1}$	0	0	1	0
0	$\bar{1}$	$\dot{0}$	$\bar{1}$	0	0	1	0
0	1	$\dot{0}$	0	0	1	1	0
$\bar{1}$	1	$\dot{0}$	$\dot{0}$	0	1	1	1
$\bar{1}$	1	$\dot{0}$	1	0	1	1	1
$\bar{1}$	1	1	$\bar{1}$	0	1	1	0
$\bar{1}$	1	1	0	0	1	1	0
$\bar{1}$	$\dot{0}$	$\bar{1}$	0	0	1	0	0
$\bar{1}$	$\dot{0}$	$\bar{1}$	$\bar{1}$	0	1	0	0
$\bar{1}$	$\dot{0}$	0	$\bar{1}$	0	1	0	0
$\bar{1}$	$\dot{0}$	0	0	0	1	0	0

Equations 1-4 make it possible to compute a state table (Table 1)[2] that provides the values of the logical functions (right half of the table) for each of the 2^4 possible combinations of the state variables (left half of the table). For example, at the last line of the table b=1, and k=x=s=0. The logical functions' values corresponding to this state of the system are K=1, B=X=S=0. This implies that at the next step two logical variables would be commanded to change there values: b should shift from 1 to 0 (with time delay $t_{\bar{b}}$), and k should shift from 0 to 1 (after a time delay of t_k).

Transition diagram:

From the state table one can derive all the possible temporal sequences of logical states, starting from any initial state (Fig. 2)[3]. Continuing with the last line of the table (marked in Fig. 2 as **'time 0'**), the asynchronous updating strategy adopted for this model implies that b and k will change their values one at a time. Therefore, after these two variables are commanded to shift their values, the system can transform in to one of two possible conformations: if $t_{\bar{b}} < t_k$ (i.e., the time delay for ligand dissociation is shorter than the time delay for PTK activation), the system will adopt the steady state b=k=x=s=0 (path 1), since b will change its value before k gets the chance of doing so. If $t_{\bar{b}} > t_k$, the next state of the system will be 1100, from which the dynamic behavior can drive the system through three different branches. Following this kind of analysis, one can derive all possible scenarios that may evolve in the system, starting from any initial conditions. These scenarios are defined in terms of relations between time delays that characterize the various transitions the system can take.

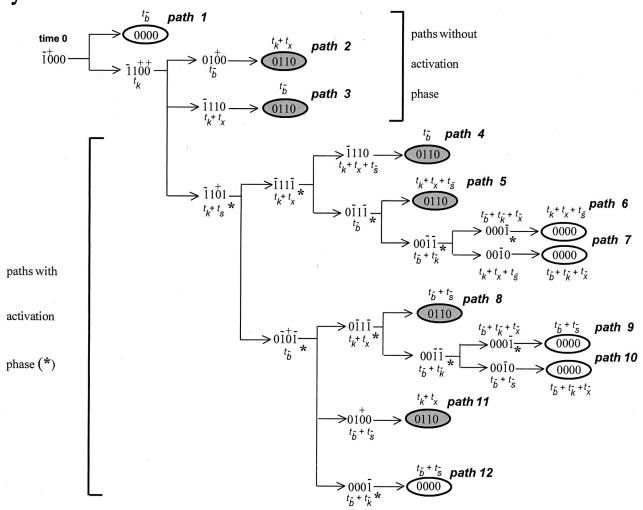

Figure 2: Transition diagram (adopted from [15])

3. THE STATECHART-BASED MODEL[4]

The model presented in this section is based on the same assumptions that were presented by Kaufman et al [15]. Therefore, no re-binding of ligands is assumed, and co-stimulation is not dealt with explicitly.

3.1 Object Model Diagram

The system, as described by Kaufman et al, consists of four object classes (Fig. 3): *Tcell*[5], *TCR*, *PTK* and *Ligand*.

Three different relationships are defined in the model:

1. *TCR* and *PTK* are components of the *Tcell* composite object.
2. *Ligand* and *TCR* are associated by the 'Bind' relationship.
3. *TCR* and *PTK* are associated by the 'Interact' relationship.

The attributes of each of the simple classes are also shown. These attributes represent the characteristic time delays of each class: binding and association delays in the case of Ligand and TCR, and activation and inactivation delays in the case of PTK. Similarly, the Tcell composite class carries with it attributes that represent time delays having to do with the activation and suppression of the

[2] A plus or minus sign over a logical value indicates that that variable is being driven upward (0 to 1) or downward (1 to 0), respectively. Parentheses indicate a stable state.

[3] An asterisk indicates stages corresponding to T cell activation (s=1).

[4] A detailed description of the modeling approach we used will appear in [8]. Meanwhile, such a description can be found at [17].

[5] Actually, the T cell does not appear explicitly either in the model diagram (Fig. 1) or in the logical analysis. Yet, it is definitely part of the system described by the model.

positive and inhibitory signaling pathways (see also in the relevant statecharts described below). Instance multiplicities, as well as relationship multiplicities, are in accordance with original model (see also parallel discussion in [8]).

3.2 Statecharts

Each of the four objects has an attached statechart describing its dynamic behavior. We will describe each of these statecharts separately.

3.2.1 Statechart of Ligand

In the model described by Kaufman et al, a free ligand will eventually bind to a free TCR. This is represented in the *Ligand's* statechart (Fig. 4) by a timeout transition that takes place after a time delay defined by the attribute 'BindingDelay'. When taking the transition from **Free** to **Bound**, the *Ligand* generates an event ('evBindLigand') on *itsTCR*. After remaining in the **Bound** state for a time period defined by the 'DissociationDelay' attribute, the *Ligand* object will terminate (since no re-binding is allowed in the original model).

3.2.2 Statechart of TCR

Receiving the event 'evBindLigand' from its *Ligand* while being in the **Free** state causes the *TCR* to take the transition from **Free** to **Bound** (Fig. 5). While doing so, the *TCR* also informs *itsPTK* about entering the **Bound** state. After a time delay defined by the 'Dissociation delay' attribute, the *TCR* will return to its **Free** state, sending the appropriate messages to *itsPTK* and to *itsTcell.*

3.2.3 Statechart of PTK

Binding of the *TCR* by its *Ligand* triggers the activation of the *PTK* (Fig. 6). However, in order to become fully activated, the 'ActivationDelay' of the *PTK* should not exceed the 'DissociationDelay' of its *TCR*. Therefore, only if the timeout defined by 'ActivationDelay' expires while the *TCR* is still bound, will the *PTK* enter the **Active** state; otherwise, it will return to **Inactive**.

Once in the **Active** state, the *PTK* will remain there as long as the *TCR* is still bound. However, even when the *TCR* dissociates from its *Ligand,* the *PTK* can remain active (due to its autophosphorylative activity), as long as the positive signaling was not evoked. Therefore, the transition from **Active** to **becomingInactive** takes place when the 'evTCRfree' event is generated, only if the *Tcell* is in its **PositiveSignaling_On**[6] state. Yet, the inactivation process will not run into completion unless the timeout defined by 'InactivationDelay' will expire before the positive signaling is turned off.

3.2.4 Statechart of Tcell

The Tcell's statechart (Fig. 7) is divided into two *orthogonal components*[7]: **ActivationState** and **SignalTrunsduction**. The **SignalTransduction** component is further divided into two orthogonal components – **InhibitoryPathway** and **PositiveSignaling**, each representing a distinct signaling pathway. The event 'evPTKactive' is responsible for driving 3 different transitions within the *Tcell's* statechart: from **Resting** to **standBy** in the **ActivationState** component, from **Off** to **On** in the **InhibitoryPathway** component, and from **Off** to **On** in the **PositiveSignaling** component (only if *itsTCR* is in the **Bound** state).

The SignalTransduction component:

Both signaling pathways have time delays. When a signaling pathway enters its **On_Full** state, a timeout event is triggered within this state (using the 'reaction in state' mechanism, as shown in Fig. 8 for the **InhibitoryPathway_On_Full** state). Only after this timeout expires, an appropriate event is generated ('evPositiveSignal' in the **PositiveSignaling_On_Full** state, 'evInhibitorySignal' in the **InhibitoryPathway_On_Full** state). This implies that if the **On_Full** state of a signaling pathway is left before the relevant timeout expires, no signal is produced.

Signal decay: once the *PTK* becomes inactive, both signal pathways – the inhibitory as well as the positive – receive messages that trigger their decay (in the case of the **PositiveSignaling** components, there are two additional events that may trigger this transition). According to Kaufman et al, such a decay signal is a "one way ticket" (for both signaling pathways): since the model assumes no rebinding between the *TCR* and its *Ligand*, the *PTK* cannot become re-activated after returning to its **Inactive** state. Yet, we included transitions leading from the two **Decay** states back to the **Full** states, by this allowing the *Tcell* to respond also to events triggered by *TCR* rebinding (see further discussion in section 4).

The ActivationState component:

The transitions within the **ActivationState** component are dependent on the timing of the signals that are produced by the **SignalTransduction** component. From the **standBy** state the *Tcell* can go either to the **Anergic** state or to the **Active** state, depending on the received signal. From the **Active** state, it can either return to **Resting** or go the **Anergic**, depending on the activity state of *itsPTK*. The **Anergic** state is a 'dead end', while returning to **Resting** enables the *Tcell* to become re-activated in the future.

[6] This notation means that within the **PositiveSignaling** super-state, the *PTK* enters the **On** sub-state.

[7] Orthogonal components represent independent substates that an object can occupy simultaneously (see [7] or [8]). Thus, a given state of *Tcell* is actually a combination of three substates: one substate from the **ActivationState** component, one substate from the **InhibitoryPathway** component and one substate from the **PositveSignaling** component.

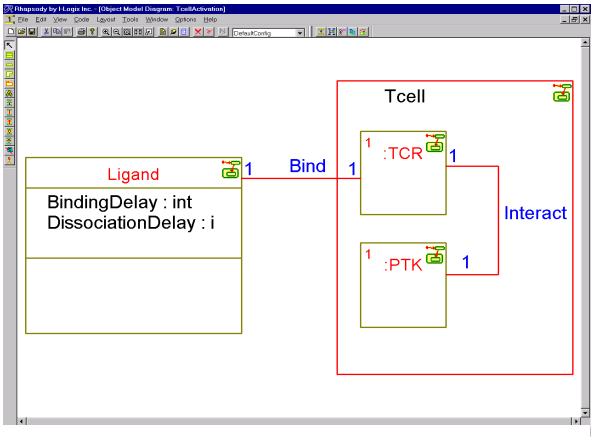

Figure 3: Object Model Diagram

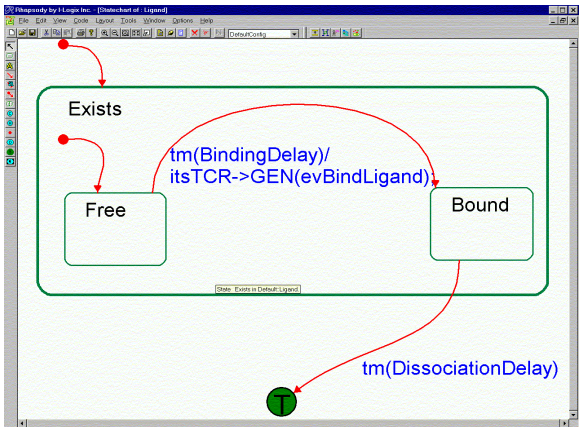

Figure 4: Statechart of *Ligand*

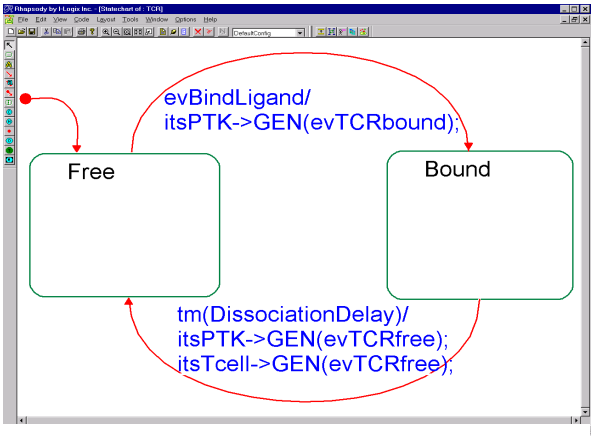

Figure 5: Statechart of *TCR*

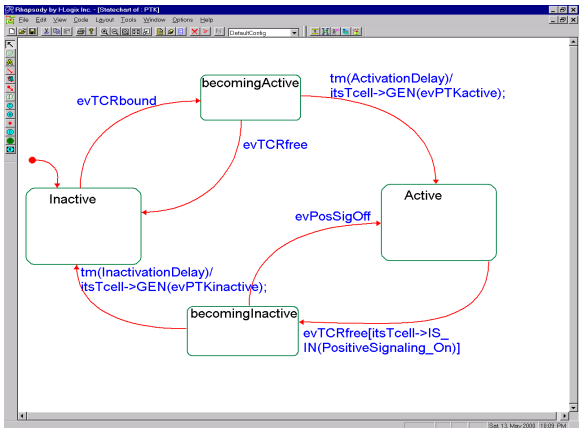

Figure 6: Statechart of *PTK*

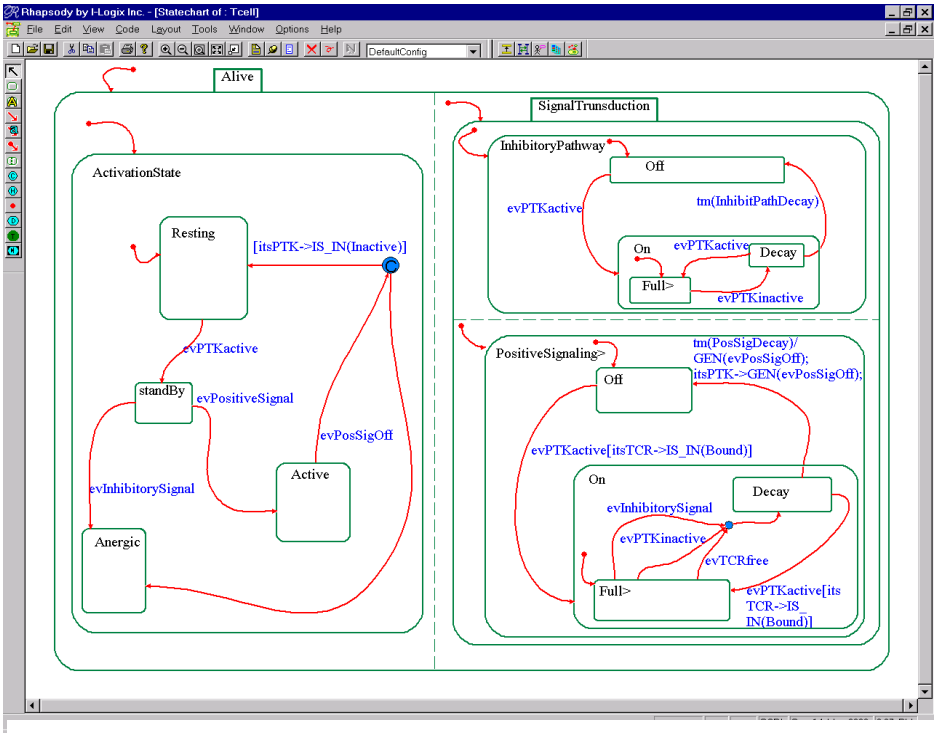

Figure 7: Statechart of *Tcell*

3.3 Model Execution

Figures 9 and 10 include snapshots from a real-time animation session that was executed using the Rhapsody tool [3]. In both cases, the *Tcell's* statechart went through the **Active** state. Figure 9 (a-b) corresponds to a situation in which the positive signal is turned off before the *PTK* returns to its **Inactive** state. In the described scenario (Fig. 9-a), this is due to the fact that even though the time delay for the onset of the positive signal was shorter than the corresponding time delay of the inhibitory signal (as reflected by the fact that from the **standBy** state the *Tcell* went to the **Active** state), eventually the inhibitory signal was generated, driving the transition of the **PositiveSignaling** component from **Full** to **Decay** (as could be seen from the olive coloring on this transition). After another time delay (defined by 'PosSigDecay') the **PositiveSignaling** component took the transition from **Decay** to **Off** (Fig. 9-b). Since during all this time period the *PTK* did not become inactive (Fig. 9-c), as reflected by the fact that the **InhibitoryPathway** remained in the **On** state, the generation of the 'evPosSigOff' event drove the *Tcell* to the **Anergic** state. In Figure 10, the 'evPosSigOff' event was generated when the *PTK* was already in its **Inactive** state. Therefore, when reaching the condition connector within the **ActivationState**, the left branch was activated (notice the olive color), leading back to the **Resting** state. This figure reflects *positive signaling with recovery of responsiveness* – the *PTK* is inactive, and the *Tcell* is **Resting**, both its signal transduction pathways being turned off after being activated.

.

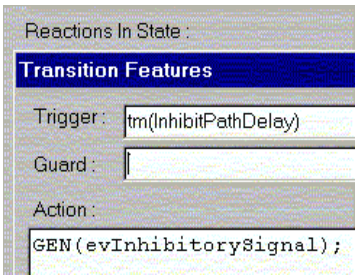

Figure 8: Reaction in state

4. DISCUSSION

Since the models presented in the two previous sections rely on the same assumptions and concepts, a comparison between them should be helpful in pointing out some differences between the two modeling approaches. Indeed, it seems that several such points can be indicated:

Model construction. In the work presented by Kaufman et al, the model was constructed in three phases: (i) drawing a schematic diagram, (ii) computing a state table, (iii) constructing a transition diagram. This construction process includes a transition from a relatively clear graphical diagram into a highly nested collection of binary strings. In the statecharts-based model, all the modeling of states in their relationsips is done using diagrammatics.

Model analysis. All the analysis work presented by Kaufman et al was carried out manually (this includes calculating the state table, constructing the transition diagram and figuring out the various conditions leading to activation vs. anergy). With the statecharts-based approach, once the model was constructed all the analysis was carried out by running simulations using the Rhapsody tool. Furthermore, the Rhapsody tool was used not only to analyze the behavior of the valid model, but also to check its validity (in principle, a proper computational tool could be designed to support the logical analysis approach as well).

Model extension. The logical analysis method developed by Thomas & D'Ari [14] implies that the number of entries in the state table, as well as the number of branches in the transition diagram, grows exponentially with the number of parameters. For example, the model described by Kaufman et al is based on four logical variables. Yet, the schematic diagram presenting the basic concepts of the model (Fig. 1) includes two extra parameters – Free ligand and Costimulation. Adding these two parameters to the logical model would result in a state table that is four times larger than the current one (64 entries instead of 16). Furthermore, not only would the model become far more complex, but the whole construction process (i.e., computing the state table and the transition table) would have to start from scratch. Actually, not only the addition of parameters, but even slight changes in the model's assumptions might require reconstruction of the Boolean analysis. For example, it was already mentioned that one of the assumptions in the original model [15] was that no ligand rebinding is allowed. If, however, one wants the model to include rebinding, this could be achieved by replacing the original logical equation $B=0$ with the new equation $B = \bar{b}.f$, where f is an input variable, representing the presence (f=1) or absence (f=0) of free ligand. Yet, replacing the equation is not enough. The authors state that with this new equation replacing the original one, "one can *again* compute the temporal state transitions", implying that the most laborious part of the modeling would have to begin from scratch.

With statecharts, however, the picture is completely different. Relating to the last example (allowing rebinding of the *TCR* to its *Ligand*), the only change that need be introduced into the model is in the *Ligand's* statechart (Fig. 11): instead of leading to a termination connector, the transition triggered by 'tm(DissociationDelay)' should lead back to the **Free** state (another possibility is to add continuously new *Ligand* instances into the system). The same holds for adding costimulation explicitly to the model, as was shown in the model described in [8]. In the static view of the model (Fig. 3), it only requires adding the relevant *Receptor* object as a component of the *Tcell* (*CD28* in the case of costimulation) and defining the appropriate *Ligand* object (in this case, *B7*). Since the *Receptor* and *Ligand* super-classes are already defined, the addition of new sub-classes, which inherit the structural and behavioral properties of their parent classes, is relatively simple. As for the dynamic behavior of the system, here things might become more complicated: complicated behavior requires complicated models. Yet, due to the special features of statecharts (zooming-in/zooming-out capabilities, clustering of states, the ability to divide a statechart into orthogonal components), much of this complexity can be reduced. Thus, for example, adding a new signal transduction pathway to a *Tcell* model can be achieved using orthagonality – the new pathway can be added as a new orthogonal component into the existing **SignalTransduction** component.

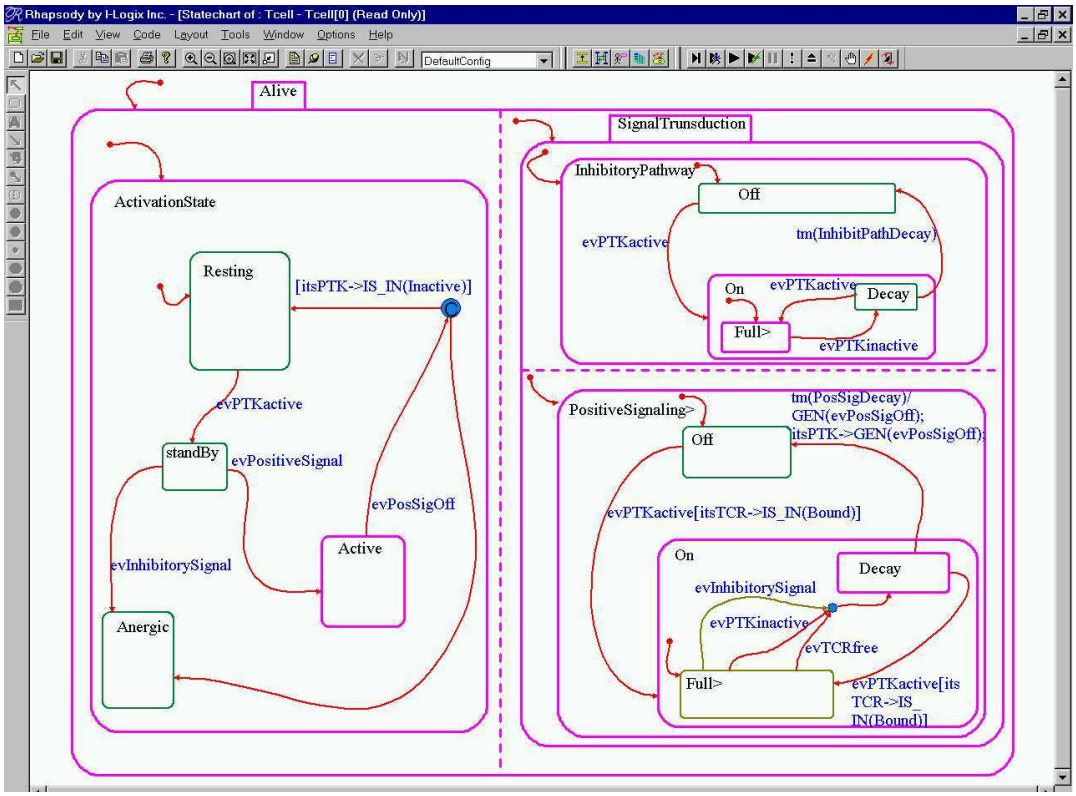

Figure 9-a: Animation snapshot of an active *Tcell*

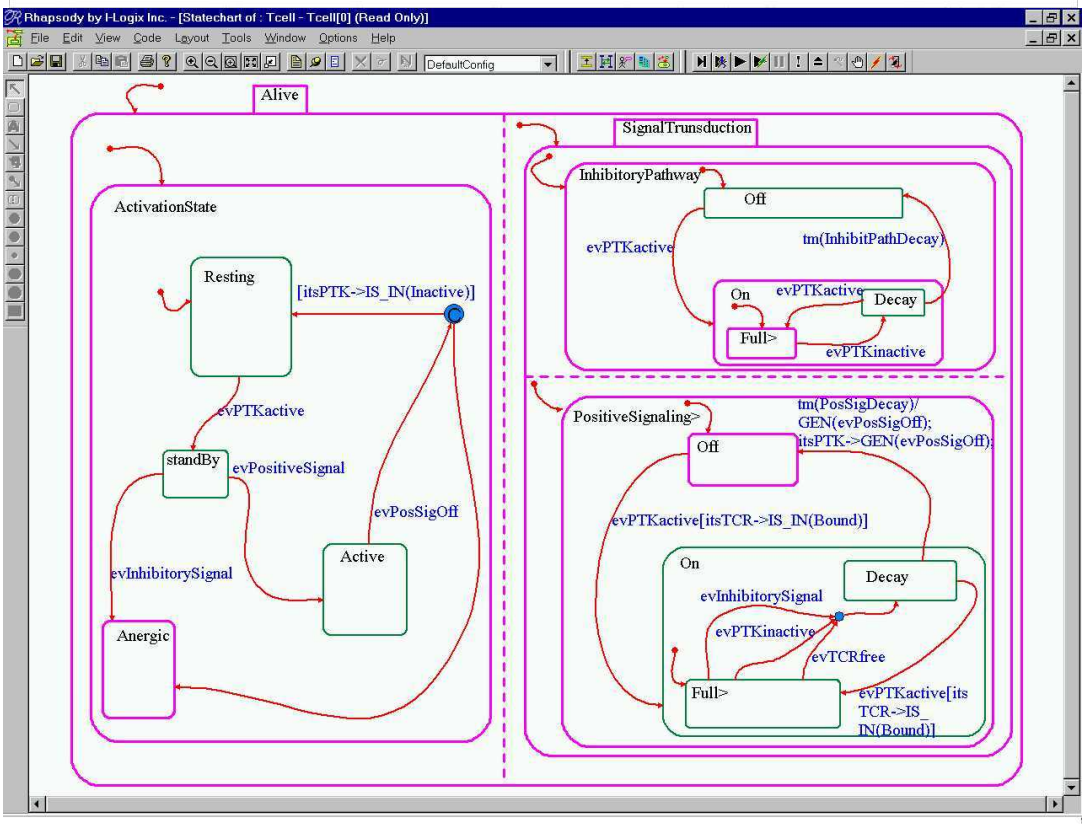

Figure 9-b: Animation snapshot of *Tcell* in its Anergic state

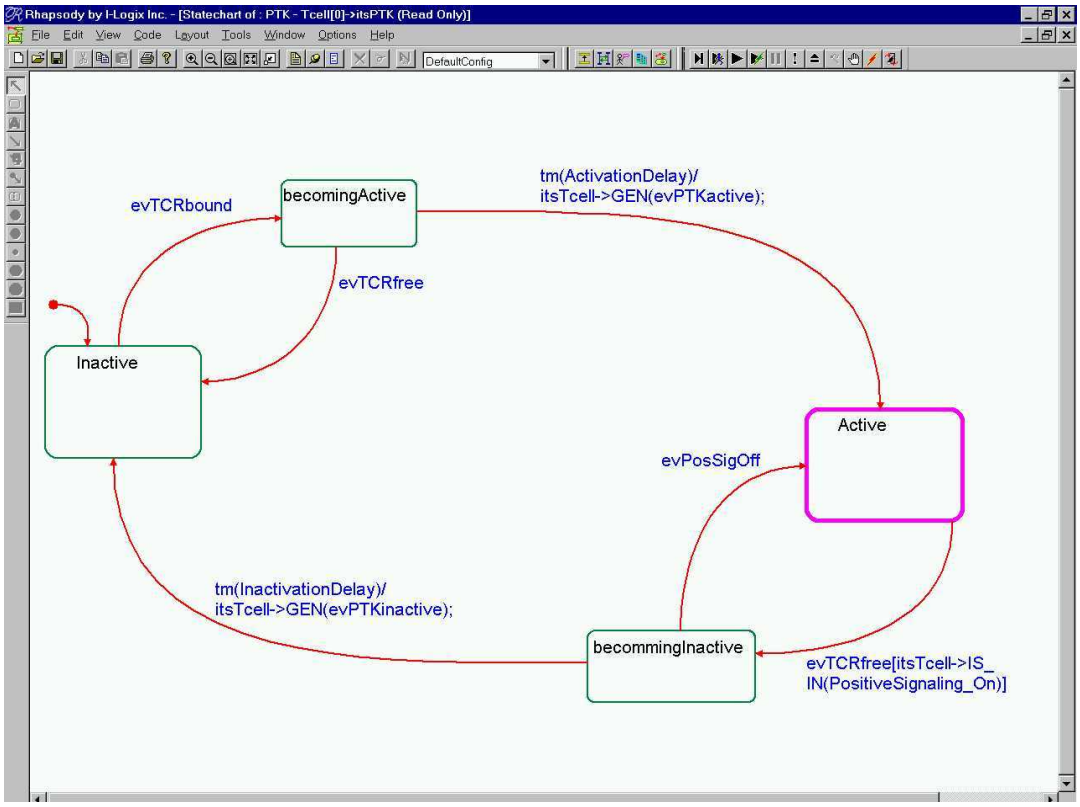

Figure 9-c: Animation snapshot of *PTK* in its Active state

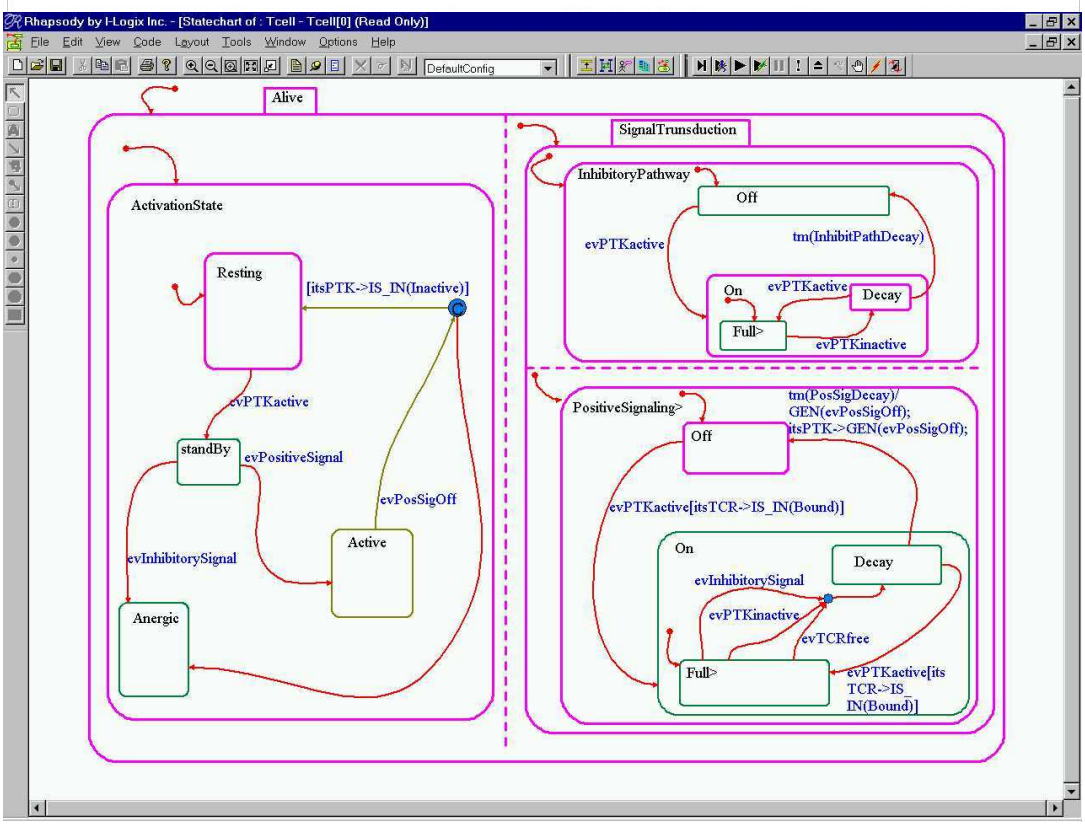

Figure 10: Animation snapshot of *Tcell* that returned to its Resting state

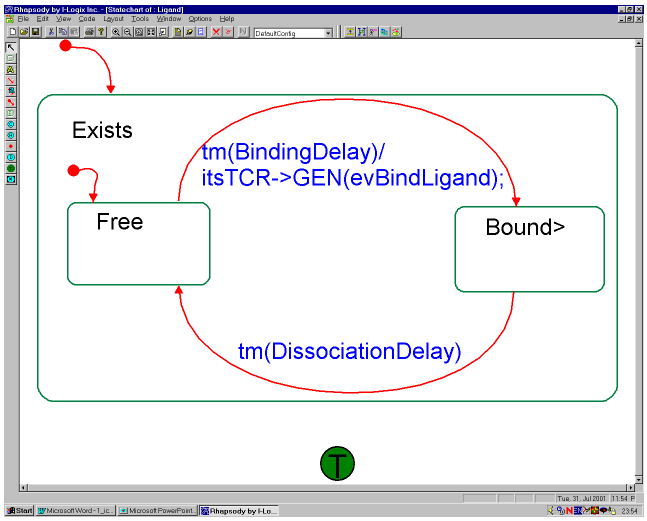

Figure 11: Statechart of *Ligand* where rebinding is enabled

No less important is the fact that in the object-oriented approach, complexity is added only where justified, i.e., within the newly added object, and perhaps within other objects that interact with it (this can result in a chain reaction that will influence the whole system, but again – if the real system is complicated the model cannot hide this complexity). The methodology used by Kaufman et al, however, doesn't support modularity: the whole system is modeled as one big entity; any small change requires re-evaluation of the entire model.

Model results. The Boolean formalism used by Kaufman et al seems to have some advantage over statecharts concerning the 'bottom line' of the model. Once the logical analysis is completed, one ends up with a Boolean formula that sums up all the possibilities for achieving a given scenario (e.g., positive signaling followed by recovery of responsiveness). With statecharts, however, there is no built-in mechanism that assures that all the possibilities have actually been checked. For example, it is reasonable to assume that one would check the model for a situation in which the time delay for the activation of the positive signaling pathway is shorter than the corresponding time delay of the inhibitory pathway. Yet, it is not trivial to assume that the modeler will check what happens when $t_{\bar{b}} < t_k + t_x + t_s - t_{\bar{k}}$. The issue of carrying out an extensive analysis of all possible play-out scenarios of the model can be addressed using temporal logic or live sequence charts (LSC's) [18], but this is beyond the scope of this work.

It should be mentioned, however, that the results received from the Boolean-formalism-based model relate only to one kind of analysis: how different relationships between time delays influence the behavior of the system. Yet, if one wants to examine, for example, how different orderings of events influence the system's behavior, the Boolean formalism does not seem to be unhelpful. This point is crucial for the question under investigation. If the model was designed to include co-stimulation (like in [8]), then the order of signals would have played a critical role: proper TCR binding followed by a co-stimulatory signal leads to full activation, while a TCR binding with no co-stimulation to follow will lead to anergy (co-stimulatory signal by its own has no effect on the T cell) [12]. With the statechart-based model, these kinds of things are quite simple: the same model could be run for a wide range of scenarios (see further discussion in Section 5; for more detailed examples see [8]).

Model clarity. Note in Figure 1 that there are several drawbacks with the representation of the system:

Semantics: the authors of [15] present Figure 1 as describing "a series of events", but later on use the notation "state variables" to describe the logical variables attached to these "events". This point is not raised here as a criticism, but rather as an anecdote that reflects an inherent problem in their methodology: there is no clear distinction between states and events. *Bound TCR* is a state and not an event, while for *Costimulation* it is the other way around. The logical model indeed describes *transitions* between *states* as a consequence of a series of *events*, but the modeling methodology provides no clear distinction between these various terms.

Hierarchy: Figure 1 contains six rectangles that represent states or events. Three of these rectangles refer explicitly to precise objects (Ligand, TCR and PTK), but the other three seem "objectless". Object model diagrams provide such organization by defining relationships and hierarchy between objects. Thus, the relatively simple object model diagram associated with this model (Fig. 3) expresses quite clearly the hierarchy between the Tcell composite object and its two component objects, as well as the different relationships among the various objects. Hierarchy and organization are also essential features of statecharts, as expressed by the clustering of sub-states into super-states.

The statecharts-based object-oriented modeling approach deals with the issues of semantics and hierarchy in a relatively intuitive manner. This observation is relevant for biologists, who deal, as a matter of routine, with decomposing a biological system into its cellular and molecular compartments, examining the relationships between them and analyzing their dynamic behavior.

5. SUMMARY

We recently reported the application of visual formalisms, which were developed for constructing computerized reactive systems, to the modeling of biological systems [8]. In this work, we have illustrated some of the advantages of this modeling approach by using the language of statecharts to represent a logical model for T cell activation and anergy, which was originally constructed in [15] using a Boolean formalism. In closing, we would like to raise another issue regarding the kinds of models that can be constructed using these different approaches.

The model presented by Kaufman et al. involves three stages that characterize the traditional quantitative modeling of biological systems:

1. Identifying a biological problem that should be addressed using mathematical tools. In [15], the motivating question was: what are the conditions leading to alternative T cell behavior in response to a given signal? Additional examples for such problems are: how sensitive is the functioning of a given protein network to variations in its biochemical parameters [19]? Under which conditions will a given system adopt

oscillatory behavior and what is the purpose of the oscillations [20]?

2. Isolating a few key components in the system, which are believed to be responsible for the phenomenon under investigation.
3. Constructing a model that relates these key components to each other, attempting at exploring the dynamics of the system.

Such models are usually based on a relatively small number of parameters. Thus, in the three examples mentioned above ([19],[20] and [15]), the models take into account six, three and four parameters, respectively. Hence, the biological questions studied with these models tend to be limited to systems that are indeed governed by a very small number of parameters, or require neglecting (at least as a first approximation) most of the parameters that constitute the real biological system.

As further discussed in [8], when constructing a statechart-based model of a biological system we need not necessarily limit the study to some specific problem such as robustness or oscillations. Rather, in the first stage, we can organize (almost) everything we know about the biological system of interest into a formal, well-structured model.
In the second stage, we can test whether the formal representation of the model fulfills our requirements (based on the existing biological data). This is done both by model simulation (see Section 3.3 and [8]) and by formal verification methods (to be discussed in a future paper).

We would like to stress that the advantages of statechart-based modeling are not merely arguments for preferring one modeling *technique* over the other. Rather, we are dealing with a totally different conceptual modeling approach. As further discussed in [8], statechart-based object-oriented modeling appears to be a natural language for describing biological systems. Describing biological systems as a collection of objects that communicate between them fits the way we think about them, it fits the way experimental data are collected and it seems suitable for coping with the challenge of understanding how biological objects collaborate to establish a system.

6. ACKNOWLEDGMENTS

We would like to thank Prof. Lee Segel for fruitful discussions and useful comments.

7. REFERENCES

[1] Cohen, I. R., Tending Adam's Garden: Evolving the Cognitive Immune Self, Academic Press, 2000.

[2] Hartwell, L. H., Hopfield, J. J., Leibler, S., and Murray, A. W., From molecular to modular cell biology, Nature, 1999, 402: C47-52

[3] Lander, E. S., and Weinberg, R. A., Genomics: journey to the center of biology, Science, 2000, 287: 1777-82.

[4] Levin, S. A., Grenfell, B., Hastings, A., and Perelson, A. S., Mathematical and computational challenges in population biology and ecosystems science, Science, 1997, 275: 334-43.

[5] Morel, P. A., Mathematical modeling of immunological reactions, Front Biosci, 1998, 3: d338-7.

[6] Callard, R., George, A. J., and Stark, J., Cytokines, chaos, and complexity, Immunity, 1999, 11: 507-13.

[7] Harel, D. and Pnueli, A., On the Development of Reactive Systems, in Logics and Models of Concurrent Systems, K. R. Apt ed., Springer-Verlag, 1985, pp. 477-498.

[8] Kam, N., Cohen, I.R., and Harel, D., The Immune System as a reactive system: modeling T cell activation with statecharts, Bull. Math. Biol. (submitted).
An extended abstract appeared in Proc. IEEE Symposia on Human-Centric Computing, IEEE, September 2001, 15-22.

[9] Harel, D., Statecharts: A Visual Formalism for Complex Systems, Science of Computer Programming, 1987, 8: 231-274.

[10] Booch, G, Object-Oriented Analysis and Design, with Applications, Benjamin/Cummings, 1994.

[11] Harel, D., and Gery, E., Executable Object Modeling with Statecharts, Computer, 1997, 31-42.

[12] Janeway, C. A., and Travers, P., ImmunoBiology, Second Edition: Current Biology Ltd./Gerland Publishing Inc., 1996.

[13] http://www.ilogix.com

[14] Thomas, R. and D'Ari, R., Biological Feedbacks, CRC Press, Boca Raton, FL, 1990.

[15] Kaufman, M., Andris, F., and Leo, O., A logical analysis of T cell activation and anergy, Proc Natl Acad Sci U S A, 1999, 96: 3894-9

[16] Schwartz, R. H., T cell clonal anergy, Curr Opin Immunol, 1997, 9: 351-7.

[17] http://www.wisdom.weizmann.ac.il/~kam/ooad.pdf

[18] Harel, D., From Play-In Scenarios to Code: an Achievable Dream, Computer, IEEE, 2001, 34: 53-60.

[19] Alon, U., Surette, M. G., Barkai, N., and Leibler S., Robustness in bacterial chemotaxis, Nature, 1999, 397: 168-171.

[20] Bar-Or, R., Maya, R., Segel, L. A., Alon, U., Levine, A. J., and Oren, M., Generation of oscillations by the p53-mdm2 feedback loop: A theoretical and experimental study, Proc Natl Acad Sci U S A, 2000, 97: 11250-11255.