The Immune System as a Reactive System: Modeling T Cell Activation With Statecharts
(Extended Abstract)

Na'aman Kam[1,2], Irun R. Cohen[1] and David Harel[2]
[1]Department of Immunology,
[2]Department of Computer Science and Applied Mathematics
Weizmann Institute of Science, Rehovot, Israel
E-mail: harel@wisdom.weizmann.ac.il

Abstract

The construction of reliable reactive systems is considered to be one of the most challenging goals in the fields of software and system engineering. The definition of a reactive system suits biological systems at different levels, ranging from gene networks, developing embryos and the immune system. We report here the application of a tool developed for constructing computerized systems to the modeling and analysis of a biological system, the immune system. We use the language of statecharts within the framework of object-oriented modeling. The results described here indicate that this modeling strategy can contribute to the transition of biology from the phase of analysis to the phase of synthesis.

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 requires the language of mathematics [2-5]. 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 comprise the immune system. "At the present time, however, there is an emerging need to understand the system as it functions as a whole" [6].

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" [7].

In this paper we model a biological system using the visual formalism of statecharts [8], as it has been adapted to the framework of *object-oriented modeling* [9-11]. We modeled aspects of T-cell activation. With this model, we are able to run simulations, using the Rhapsody tool [12]. These simulations allow us to compare the dynamic behavior of the model to actual experimental data. Our results show that the use of statecharts can help confront open questions in immunology and probably in other fields of biology, which, because of their complexity, cannot be addressed by standard laboratory techniques alone. In addition, a visual formalism provides a clarity lacking in other approaches to the mathematical modeling of biological systems.

2. The biological system considered

A living cell can be viewed as a system defined by combinations of inner states, reflecting the interactions of the cell with its environment. Transitions between these inner states occur as a consequence of signals that are introduced to the cell by the environment (some of these signals might be produced by the cell itself). *Naive T cells*, for example, can live for many years without dividing. These naive cells circulate continuously from the blood stream to the lymphoid organs and back to the blood, serving as patrol units that sense changes that might occur in the body. This sensing process is carried out by interactions that the T cells make with antigen-presenting cells. Naive T cells that recognize their specific antigen on the surface of a professional antigen

presenting cell cease to migrate, and are activated to proliferate and to differentiate into armed effector cells.

The signaling process that drives cell proliferation can be divided into two stages:

1. "The initial encounter with specific antigen in the presence of the required co-stimulatory signal triggers the entry of the T cell into the G1 phase of the cell cycle and, at the same time, induces the synthesis of IL-2 along with the α chain of the IL-2 receptor" [13].
2. "Binding of IL-2 to the high-affinity receptor then triggers progression through the rest of the cell cycle" [13].

This short description of T cell activation illustrates some of the challenges one encounters while trying to model such a behavior. Describing this behavior raises several questions: Where is the cell? What other entities does it encounter? Which kinds of signals does it receive? What kind of behavioral outputs does it produce? Therefore, we would like to have a modeling language that is able to capture all these various aspects of behavior *simultaneously*. The need for this parallelism is due to the fact that these various components of behavior are in a sense orthogonal to each other: T cells at the lymph node can be either in an active or in a naive state, and an active T cell can be in any one of the four cell cycle stages.

Another important issue is the ability to distinguish between signals generated outside the cell and signals or events that are produced by the T cell itself. In the above example, the first stage towards cell proliferation – the transition from the G0 stage into G1 – is driven by a combination of two signals that originate from an extracellular event: the interaction of the T cell with a professional antigen-presenting cell. On the other hand, the progression through the rest of the cell cycle is a consequence of two intracellular events: the production of the α unit of the IL-2 receptor and the secretion of IL-2 itself by the T cell.

This traditional description treats the cell as a black box, in the sense that it specifies only the outputs produced by the cell in response to given inputs. Nothing is said about *how* the input signals are translated into the outputs; the signal transduction process is totally ignored. Yet, in order to gain a better understanding of such lower level behavior, and to investigate the influence it has on higher level behavior, one would like to have the ability to "zoom" into the system, and describe its inner behavior in a more careful and detailed manner. An adequate modeling approach should support such abilities in a smooth and intuitive, yet formal and rigorous fashion.

3. The Rhapsody tool

Just as programming languages require compilers and debugging tools, so do modeling languages require not only pretty graphics, but also means for executing models and for synthesizing code. In this work, we use the Rhapsody tool (developed by I-Logix), defined as an "object oriented fully integrated Visual Programming Environment (VPE)" (www.ilogix.com), as the working environment. In addition to providing a computerized visual design environment for performing object-oriented modeling, Rhapsody is capable of automatically translating any syntactically legal model into executable code, currently, in C, C++ or Java. Once the model is constructed and translated into executable code, Rhapsody can animate the execution of the code so that one can observe its progress in animated versions of the model's statecharts. Animation shows the current state and transitions by coloring them uniquely. In Rhapsody, the transitions between graphical diagrams and executable code are bi-directional: during the compilation stage, the object model diagrams and the statecharts are translated into C++ code lines, and while animating the application, the execution of the C++ code is translated into color changes within the statechart diagrams.

4. Model Construction

Due to space limitations, we omitted in this extended abstract many immunological details that were used for constructing the model. For the reader interested in such details, an extended version of this paper has been prepared [16].

4.1 Object Model Diagrams (OMD's)

The first object model diagram (see Fig. 1) describes the relationships between 3 major object classes: *ThelperCell*, *Receptor* and *Ligand*. The *ThelperCell* is a composite object class that contains other kinds of object classes, all of which are *Receptors*. Through these *Receptors*, the T cell can communicate with other objects that belong to the *Ligand* super-class. This is denoted by the 'Bind' relationship, which symmetrically associates *Receptor* and *Ligand*. Another relationship by which the *ThelperCell* class is associated with other objects is the 'Secrete' dependency, which connects the *ThelperCell* to the *Cytokines* class.

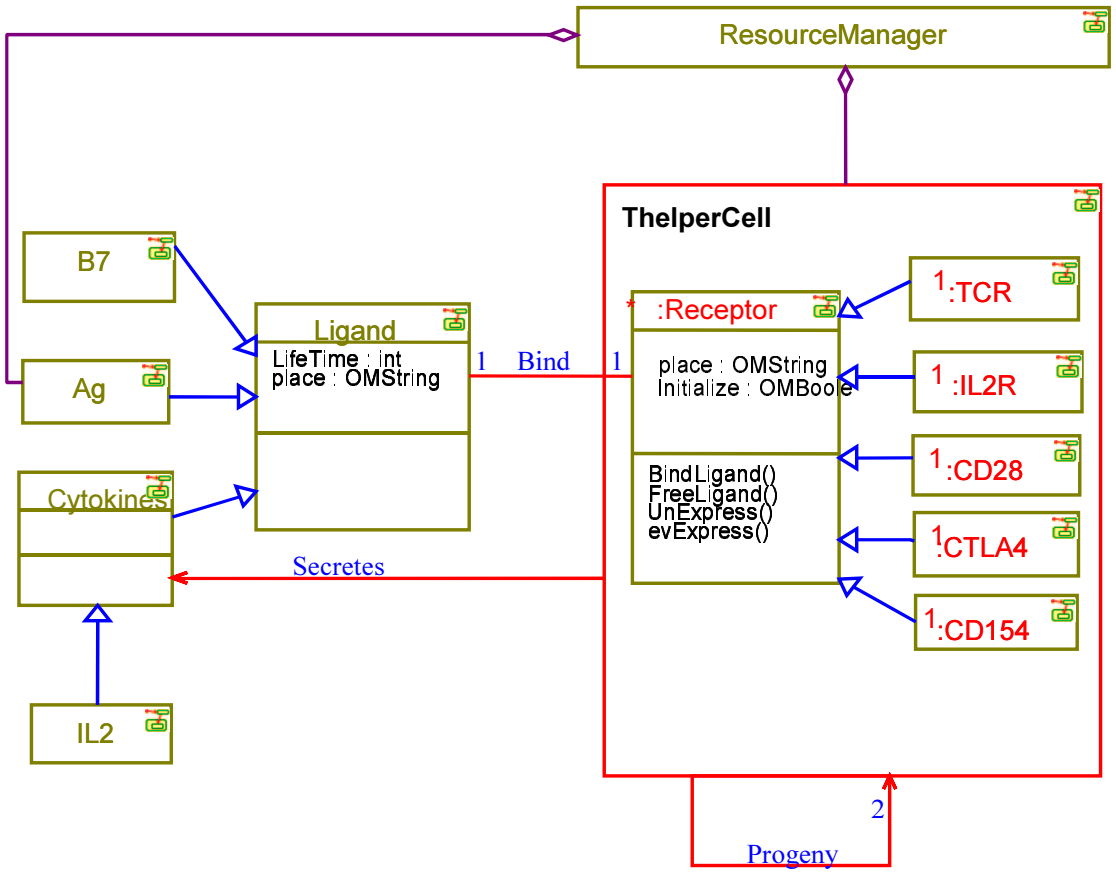

Figure 1

Another dependency arrow named 'Progeny' is drawn in Figure 1 from the *ThelperCell* class to itself, having a one-to-two multiplicity. This relationship implies that a single instance of a *ThelperCell* can refer to two other instances of *ThelperCells* as *its* progeny. This relationship is used during T cell proliferation, when one cell divides into two. We refer to this process as if the mother cell 'dies', and is replaced by two newly born daughter cells.

Another special object class that appears in the object model diagram of Fig. 1 is the *ResourceManager*. The *Resource Manager* carries out three basic functions, essentially 'taking over some tasks from nature', or fulfilling laws of nature:

1. Initiating the first 'generation' of cells (which then can proliferate and die by themselves).
2. Supplying the system with antigen molecules (see also [14]).
3. Manipulating the interactions between receptors and ligands.

Whenever a *Receptor* or a *Ligand* enters a **Free** state, it notifies the *ResourceManager*. The *ResourceManager* acts as a 'matchmaker', continuously searching for a suitable binding partner within the system. If such a match is found, the *ResourceManager* associates the *Receptor* with the corresponding *Ligand*, causing both of them to enter the **Bound** state. In this case, the resource manager carries out *Ligand* diffusion and *Receptor-Ligand* recognition.

It can be seen that all *Receptors* have an instance multiplicity of one in our model. In reality, however, each cell expresses multiple copies of the same receptor. We assume here that when a *Receptor* enters the **Bound** state, it reflects a situation in which the number of receptors bound to a ligand is sufficient to produce an observable response in optimal conditions. Since, on the average, each receptor copy contributes equally to global behavior, we assume that for a qualitative description of the signaling process it suffices to consider the behavior of a single receptor [15].

A second OMD (not shown) extends the system from one cell type to two major cell classes: the T helper cell and the Antigen Presenting Cell (*APC*). The super-class *APC* and the *ThelperCell* are associated by the '**Interaction'** relationship. The conditions under which such interactions can take place are defined later in the framework of the dynamic behavior.

A third OMD (not shown) is dedicated to the *ResourceManager* and relates to all the object classes in the system (since the *ResourceManager* has to be able to 'speak' to all *Receptors* and *Ligands* in order to handle the interactions between them).

4.2 Statecharts: describing dynamic behavior

Since there is a distinct statechart attached to each object class, the current model contains over 20 statecharts. In the interest of clarity, we present here only the statechart that describes the dynamic behavior of the *ThelperCell* class.

The statechart of the *ThelperCell* class describes three orthogonal components of the behavior of a T cell (see Fig. 2):

1. Its immunological state — whether the T cell is active or not.
2. Its phase in the cell cycle.
3. Its anatomical location.

4.2.1 Immunological state. The immunological state component contains two super-states: **Active** and **nonActive**.

The default transition is conditioned: if the number of cell cycles through which the T cell has gone is zero (i.e., the cell is naive), then the cell will enter the **nonActive** super-state. If, however, the number of cell cycles is greater than zero, implying that this newly born cell is the progeny of an active T cell, it enters the **Active** state.

This cumbersome way of charting T cell history is due to the fact that the current version of the Rhapsody tool lacks the capability of *replicating* objects. Currently, it is only possible to create 'fresh' new instances, i.e., ones that start from scratch, in that they enter the statechart's default states. Replication, however, implies that when an existing object instance creates a new instance of the same object class, the statechart of the newly created instance should start in the exact state configuration of the original object at the moment of replication. Such a replication mechanism would enable the creation of instances that carry with them the history of their spawning parents. In

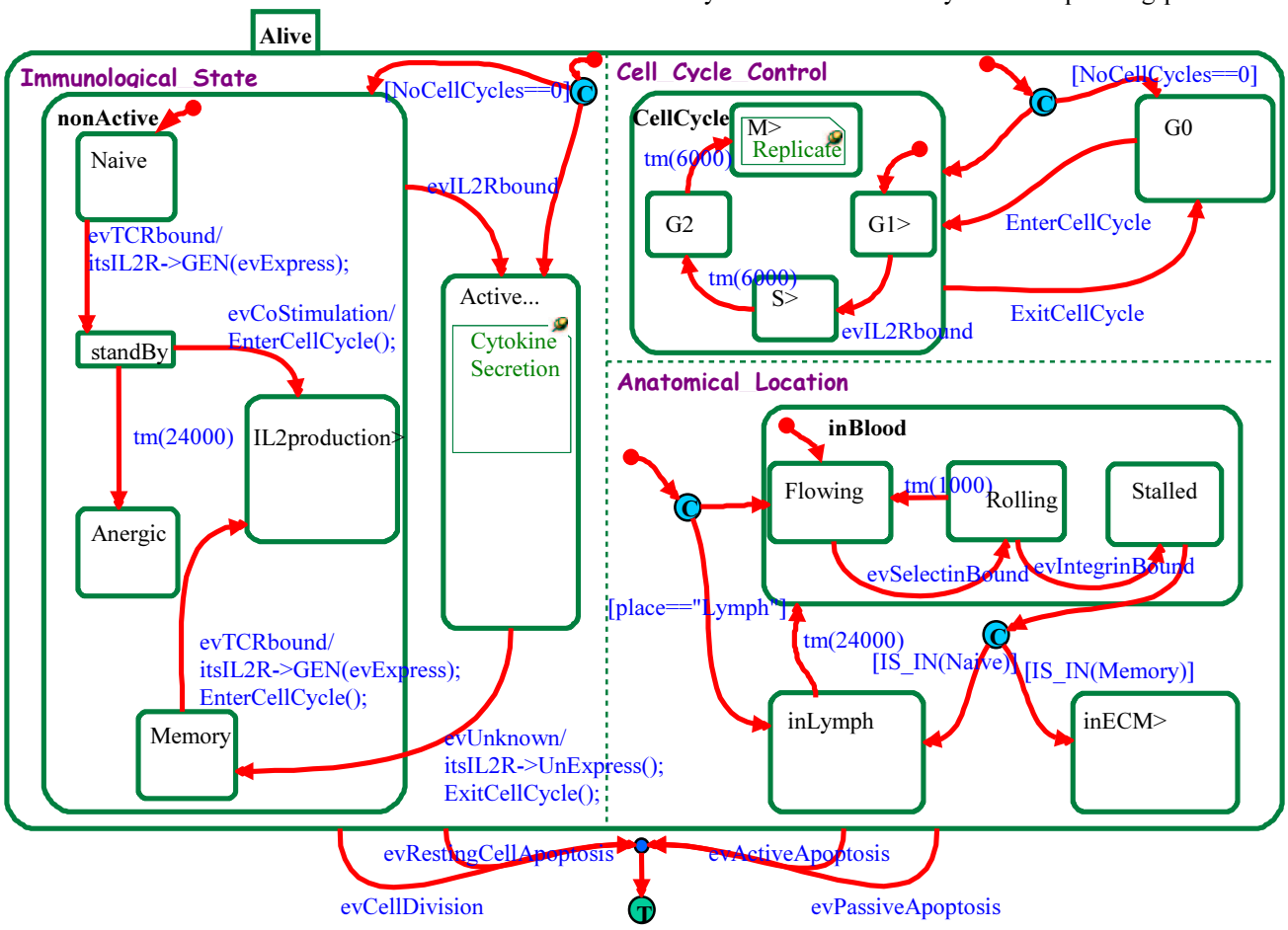

Figure 2

order to overcome this current deficiency in the language, we took advantage of the fact that when one object creates a new one (either of its own class, or of any other related object class), the original object can control the initial values of the new instance's attributes. In the case of the T-helper cell object class, these attributes can be made to reflect the relevant history of the T cell, and can be transferred to the daughter cell during proliferation. Therefore, instead of using simple default transitions, we use conditioned default transitions that are dependent on the values of these attributes. These attributes are as follows (in Fig. 2 they appear on the arrows emanating from the condition connectors):

- *NoCellCycles* — the number of cell cycles, which equals zero for naive cells and is greater than zero if the instance was generated during T cell proliferation.
- *Place* — designates the current anatomical location of the T cell. For naive cells, the default location is the blood vessels, while for newly born active cells it is lymphoid tissues.
- *CellType* — indicates whether the cell carries the Th0, Th1 or Th2 phenotype.

Within the **nonActive** super-state, the T cell can be in one of the following states:

1. **Naive** – a cell that was not previously activated by an antigen.
2. **StandBy** – a cell that encountered an antigen presented on the surface of an antigen-presenting cell. The interaction between the T cell and the antigen-presenting cell resultst in a signal transmitted by the T cell antigen-binding receptor (*TCR*), which drives the transition from the **Naive** state to the **standBy** state. Yet, in order to become fully activated co-stimulatory signals are needed. Such signals are generated by *professional* antigen-presenting cells. There is a certain 'window of opportunity', during which the T cell is capable of responding to the co-stimualtion. If this signal is transmitted to the cell while the cell is still in its **Naive** state, or when the cell is already in its **Anergic** state (see below), the cell will not be able to respond to co-stimulation.
3. **Anergic** – "Naive T cells that recognize antigen on cells that cannot provide co-stimulation become anergic, as when a T cell recognizes a self antigen expressed by an uninfected epithelial cell. This T cell does not differentiate into an armed effector cell, and cannot be stimulated further by professional antigen-presenting cells" [13].
4. **Memory** – memory T cells are long-lived lymphocytes that remain in a resting state after the antigen has been cleared. The events that trigger the differentiation of an **Active** effector T cell into a **Memory** cell are still unknown [17].
5. **IL2production**: a T cell that received the appropriate set of signals starts to secrete a cytokine molecule called IL-2. The binding of IL-2 by the IL-2 receptor leads to full activation.

The **Active** state: The process of T cell activation confronts two basic challenges:

1. *Proliferation*—making more of the same;
2. *Differentiation*—assuming a function.

Control of proliferation is carried out within the cell cycle component of the statechart. In the current model, we do not get too deeply into the differentiation aspect of T cell activation. However, a basic sketch appears in the **Active** sub-statechart (see Fig. 3).

As helper T cells differentiate, they are thought to go through an intermediate stage known as Th0. Therefore,

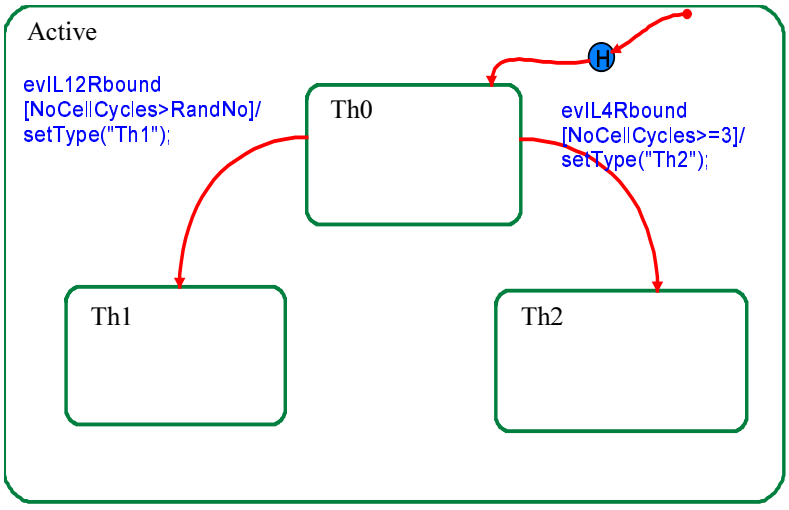

Figure 3

the default arrow in the **Active** sub-statechart leads to the **Th0** state. Further differentiation into Th1 or Th2 armed effector T cells is dependent on two basic factors:

1. The cytokine pattern sensed by the T cells. Two key cytokines that control this process are IL-12 and IL-4, which drive the cells towards the Th1 and Th2 pathways, respectively.
2. The number of cell cycles these cells have gone through. Th0 cells fail to respond to IL-4 unless they have traversed at least 3 cell cycles, while the responsiveness of Th0 cells to IL-12 was found to increase as cell proliferation proceeds: "A naive cell can become a Th1 cell with increasing likelihood after each successive cell cycle but can only become a Th2 cell when eight siblings have arisen" [18]. This dependency on cell cycle progression was found to be related to cell cycle dependent changes in chromatin structure, which gradually 'expose' these cytokine genes to transcription.

In the model, the combination of these two factors is taken care of by the guarded transitions emanating from the **Th0** state (see Fig. 3). The **Th0**-to-**Th2** transition will take place when the event of binding IL-4 is transmitted by the IL-4 receptor ('evIL4bound'), only if the condition *'NoCellCycles'* ≥ *3* is fulfilled. As for the **Th0**-to-**Th1** transition, the conditional guard here adds a probability factor to the transition: A random number between 1 and 5 is chosen; if this number is smaller than the number of cell cycles, the transition will take place once the event of IL-12 binding is generated by the IL-12 receptor. This mechanism ensures that the probability of the IL-12-binding event to promote the Th1 state will increase with the number of cell cycles.

4.2.2 Cell cycle control. As mentioned above, naive T cells can live for many years without replicating, implying that during this time period they stay in the **G0** state. The initial encounter with a specific antigen in the presence of the required co-stimulatory signal triggers the transition from **G0** to the **G1** phase of the cell cycle. In our model, the transition from the **G0** state to the **G1** state is driven by the 'EnterCellCycle' triggered operation, which is initiated during the transition from the **standBy** state to the **IL2production** state. Further progression through the cell cycle is dependent on IL-2 binding by the IL-2 receptor, an event that triggers both the transtion from the **G1** state to the **S** state in the *Cell_Cycle_Control* component and the transition from the **nonActive** state to the **Active** state in the *Immunological_State* component. If no death signal is induced while being in the **S** phase of the cell cycle, the cell will go through the rest of the cell cycle, depending only on timeout events. Once entering the **M** state, a series of actions is carried out, resulting in the generation of two new instances of the ThelperCell class, representing the two newly born daughter cells.

4.2.3 Anatomical location. Here, we have not provided a detailed description of T cell migration. The main role of the anatomical location component is to provide the spatial context for receptor-ligand interactions: in order for a chemical interaction to take place, the two interacting entities must be in proximity. Whenever a T cell changes its location, it informs the resource manager about this transition. The *ResourceManager* then searches among the relevant free *Ligand* objects for such that share with the T cell the same anatomical location.

5. Results

5.1 Model execution

In order to check whether the formal representation of the model fulfills our requirements (based on the immunological data presented in previous sections), we executed the model using the Rhapsody tool. Due to space limitations, we omitted this section from this version of the paper; this section will be included in a more comprehensive version, which is in preparation.

5.2 Unexpected Behavior

By and large, the behavior we encountered in the executions of the model followed our expectations, except for one case: we found that the *Thelpercell* object could not reach a steady **Memory** state. The moment a T cell took the transition from **Active** to **Memory**, it returned right back to the **Active** state. This happened because IL-2 molecules were still in the system, while the *Thelpercells*, being in their **Active** state, expressed their IL-2 receptor (*IL2R)* component objects. As shown in Fig. 2, the event of IL-2 binding by the IL-2 receptor triggers the transition from the **nonActive** super-state to the **Active** state. Only after an extensive search in the literature did we find that the transition from **Active** to **Memory** is accompanied by downregulation of IL-2 receptor expression, thus causing the memory T cells to become insensitive to the presence of IL-2 [19]. Accordingly, we corrected the statechart by adding the action 'itsIL2R->GEN(evUnExpress)' to the transition leading from the **Active** state to the **Memory** state (see Fig. 2).

6. Discussion

In this work, we examined the feasibility of statecharts in bridging the gap between analysis and synthesis by

modeling a well-studied immunological process termed T cell activation using the language of statecharts, as implemented in the framework of the Object-Oriented Analysis and Design (OOAD) approach. This methodology supports the following attributes:

Concurrency: Biological systems in general, and the immune system in particular, are highly concurrent. Changes in the surrounding environment can trigger the execution of many parallel processes. At the cellular level, a given cell carries out different tasks in parallel, as shown in Fig. 4 (e.g., secreting IL-2 and progressing though the cell cycle). Dividing the T-cell's statechart into three orthogonal components enabled us to derive a description of T-cell behavior which is composed of three different aspects: the T-cell's immunological state, its cell-cycle phase and its anatomical location.

Multi-level description: Information processing in biological systems occurs at different levels, ranging from the molecular and cellular levels, up to the higher levels of organisms and populations. Computations in biological systems are carried out on each of these levels. At the molecular level, for example, the T-cell receptor could be simply regarded as **Free** or **Bound**. One can further zoom into the sub-molecular level, and divide the binding process into several events occurring at the sub-molecular level: successive additions of phosphate groups to six different positions on the zeta (ζ) chain of the antigen receptor (TCR) [20]. Only a full phosphorylation of all six sites will result in an activation signal transmitted by the TCR. Completion of the phosphorylation steps is dependent on the nature of the TCR ligand. Thus, the phosphorylation steps establish a threshold for T cell activation at the sub-molecular level.

As well as traveling between the molecular and sub-molecular description level, a multi-scale description is also needed at the cellular and sub-cellular levels. The model described in this paper, for example, treats the T cell as a 'black box': it describes the output behavior of the cell as a reaction to external signals, regardless of the signal transduction events that occur within the cell and lead to this behavior. Thus, the model simply states that proper signaling via the TCR, followed by co-stimulation, triggers the transition from **Naive** to **Active**. The present model says nothing about the molecular or sub-molecular meaning of "proper signaling" (the phosphorylation events), neither does it take care of the cascade of events that translate the signal transmitted by the bound TCR into a transition from **Naive** to **Active** (the signal transduction pathway). The model, however, can be modified to include signal transduction events as well (though we have not done so in this paper). This 'zooming in' process has two implications: (i) adding new components (e.g., *enzymes*) to the *ThelperCell* composite object, and (ii) adding several orthogonal components to the *ThelperCell's* statechart – a distinct component for each signal transduction pathway.

This ability to describe the system in a multi-scale manner answers one of the key challenges of theoretical biology: "understanding how detail at one scale makes its signature felt at other scales, and how to relate phenomena across scales" [4].

The model described in this work was constructed in the framework of the **object-oriented** modeling approach. This methodology was favored for the following reasons:

- Describing the system as a collection of objects that communicate between them highly correlates with the challenge of studying entire biological systems "by attempting to understand how component parts collaborate to create a whole" [3].
- Interestingly, most of the experimental data in biology accumulates in an object-oriented manner. The specialization through which biological research has progressed during the last few decades has lead to the current situation whereby most research has been reduced to questions about "the individual components of complex biological systems". As a consequence, much is known about individual proteins (their amino acid sequence, their 3D structure, the role they play in concrete biological reactions, the identity of other proteins with which they interact, etc.), but much less is known about networks of genes and proteins. Concerning the immune system, a significant amount of data exists about its cellular compartments (the receptors they express, the molecules they secrete, their activation requirements, etc.), but very little is known concerning the way these cells collaborate to function as a system. Object-oriented modeling provides a useful formalism for constructing models by organizing information gathered around individual objects.
- People actually think about biological systems in an object-oriented manner. The first section of the immunology textbook of Janeway & Travers is entitled "The components of the immune system" [13]. This reflects our tendency to base most of the discussions related to biological systems and processes on the concrete biological objects that constitute the system.

These three issues are related to each other. Yet, presenting them as three distinct points illustrates the great advantage of using the object-oriented modeling approach: it fits the way we think, it fits the way the experimental data are collected and it seems suitable for coping with the challenge of understanding how biological objects collaborate to establish a system.

As shown in this work, statecharts can be used to represent different aspects of biological system behavior,

at varying scales of description. Constructing statecharts-based models can be helpful in addressing various questions, such as:

- How can the order of events and the duration of time delays influence the behavior of the system?
- Can contradictory output behaviors be explained by reorganizing inner states into different clusters or orthogonal components? (see further discussion in [16]).

Besides predications and explanations, "one of the beauties of mathematical modeling is that it raises questions that may not have been addressed before" [6]. Merely organizing the existing information using a visual formalism can raise questions that were not thought of before.

Finally, modeling can influence not only the biological system, but also the computational methods used. Confronting tools developed for modeling and analyzing industrial systems with the challenge of modeling and analyzing biological systems may reveal limitations of the computational tools that were not appreciated before. During our modeling work, two such points became evident:

1. Currently, the Rhapsody tool lacks a mechanism for object replication. As a consequence, modeling the process of T cell proliferation resulted in a relatively cumbersome model (see Section 4.2.1).
2. Communication between objects in Rhapsody is handled as point-to-point communication, i.e., each message sent by a given object instance has only one destination (the address of the recipient object instance). Rhapsody does not allow *broadcasting*, i.e., sending messages to groups of objects, such as all the components of a composite object (limited broadcast) or all the objects in the system (full broadcast). The lack of a broadcast mechanism in Rhapsody is the main reason for heavily relaying, in the present model, on the *ResourceManager* for enabling *Ligand-Receptor* interactions (see Section 4.1).

These and other deficiencies of the modeling language/tools can be rectified. But even at this early stage of application, it is rewarding to see how well an object-oriented visual language fits the needs of biological understanding.

Acknowledgments

We would like to thank Prof. Lee Segel, Dr. Felix Mor and Francisco Quintana for fruitful discussions and for their useful comments.

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] Wooley, J. C., "Trends in computational biology: a summary based on a RECOMB plenary lecture", *J. Comput. Biol*, 1999, 6: 459-74.

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

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

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

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

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

[11]Rumbaugh, J.,Blaha, M., Premerlani, W., Eddy, F. and Lorensen, W, *Object-Oriented Modeling and Design*, Prentice Hall, Englewood Cliffs, NJ, 1991.

[12]www.ilogix.com.

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

[14]Brass, A., Grencis, R. K., and Else, K. J., "A cellular automata model for helper T cell subset polarization in chronic and acute infection", *J Theor Biol*, 1994, 166: 189-200.

[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]Kam, N., Cohen, I.R., and Harel, D., "The immune system as a reactive system: modeling T cell activation with statecharts", Technical Report MCS01-09, The Weizmann Institute of Science, Rehovot, Israel, June 2001.

[17]Metz, D. P., and Bottomly, K., "Function and regulation of memory CD4 T cells", *Immunol Res*, 1999, 19: 127-41.

[18]Bird, J. J., Brown, D. R., Mullen, A. C., Moskowitz, N. H., Mahowald, M. A., Sider, J. R., Gajewski, T. F., Wang, C. R., and Reiner, S. L., "Helper T cell differentiation is controlled by the cell cycle" *Immunity,* 1998, 9: 229-37.

[19]Carter, L. L., and Swain, S. L., "From naive to memory. Development and regulation of CD4+ T cell responses", *Immunol Res*, 1998, 18: 1-13.

[20]Kersh, E. N., Shaw, A. S., and Allen, P. M., "Fidelity of T cell activation through multistep T cell receptor zeta phosphorylation", *Science*, 1998, 281: 572-5.