

Using UML to Model EAE and its Regulatory Network.

Mark Read¹, Jon Timmis^{1,2}, Paul S. Andrews¹, and Vipin Kumar³

¹ Department of Computer Science, University of York, UK.

{markread | jtimmis | psa}@cs.york.ac.uk

² Department of Electronics, University of York, UK.

³ Laboratory of Autoimmunity, Torrey Pines Institute for Molecular Studies.

Experimental Autoimmune Encephalomyelitis (EAE) is an autoimmune disease in mice which serves as a model for multiple sclerosis in humans [4, 5]. The disease constitutes the direction of immunity towards myelin, an insulatory material that covers neurons. The consequential damage to the central nervous system (CNS) can lead to paralysis and death [6].

EAE can be spontaneously induced by immunisation with myelin basic protein (MBP, a myelin derivative) and complete Freund's adjuvant. The immunisation prompts the expression of MBP peptides on MHC molecules by antigen presenting cells (APCs), and the consequent activation of MBP-reactive T cells. The activated T cells migrate to the CNS parenchyma where their secretion of type 1 cytokines promotes the destruction of myelin.

A network of immune cell interactions operates to counter EAE. This regulatory network consists of CD4⁺ and CD8⁺ regulatory T cells (Tregs). The natural lifecycle of MBP-reactive CD4Th1 cells leads to their physiological apoptosis and subsequent phagocytosis by APCs. The peptides derived from CD4Th1 cells, when presented on MHC, prompt the activation of CD4⁺ and CD8⁺ Tregs. The CD8Tregs, with prior help from CD4Tregs, can induce the apoptosis of activated MBP-reactive CD4Th1 cells. The resulting population reduction permits the expansion of CD4Th2 cells which do not promote debilitating destruction of the CNS.

It is essential that a coherent understanding of the biological domain is obtained to construct an accurate representation of the system [7]. In line with others, for example [2, 1, 3], we have selected the UML as the tool with which to develop our models, before we move to an agent based simulation. We have completed a first pass in modelling the biological domain and wish to highlight certain issues that we have found when employing the UML in this context.

We have found that the construction of our models has raised various questions of the system's operation; the immunological literature typically reports what *does* happen in a certain experimental setup for a particular event to manifest, it does not indicate all of what *can* happen under altered conditions. To correctly model and simulate the system we must know the latter as well as the former.

Capturing system wide behaviours with activity diagrams.

Since we intend for our simulations to capture the higher level behaviours of the biological system we believe the modelling of how these manifest (at an abstract level) from the interactions of lower level components to be an appropriate starting point. Activity diagrams have proven to be a very suitable medium through which to accomplish this. Any abstract concept can be expressed as an activity, and links between activities can span across multiple system entities; in our case cells. Furthermore, activity diagram semantics allow for the expression of concurrent activities; concurrency is a fundamental intrinsic quality of biological systems.

Representing static relationships with class diagrams.

We have found the construction of class diagrams to be effective at generating questions relating to the quantities of entities that may partake in an activity at a particular time. These are valid questions, because they pertain to the dynamics of the system. However, reasoning about the system in a static manner is not as informative to its operation as is examination from a dynamic viewpoint. *In vivo* the number of entities that can attempt to simultaneously interact with one another varies considerably, and this usually manifests in '0..*' cardinalities on class diagrams which are not particularly informative. Furthermore, biology is rich with entities that interact and influence large numbers of other entities, which leads to highly connected class diagrams that are difficult to interpret in a meaningful manner.

Sequence diagrams are misleading.

We have not used sequence diagrams in this stage of modelling as we consider them to be misleading; thinking about this biological domain in terms of entities that wait on other entities to complete some task is inappropriate. For example, a cell may require a series of signals to reach some state, but it does not lie in wait in between receipt of signals; it will continue to interact with its environment. Cells can be open to more than one path of events. The syntax of sequence diagrams does not communicate this well, and rather implies that an entity be temporarily 'locked' or suspended whilst activity proceeds elsewhere. A cell does not hold responsibility over sub-actions that result from its own.

Low level dynamics and state machine diagrams.

State machine diagrams of individual system elements that depict low level dynamics have proven to be very informative. Their provision of facilities to express orthogonality, concurrency, mutual exclusion, and containment of states renders them appropriate for expressing behaviour of cells. Though most behaviours in the system can be extrapolated through examination of state machine diagrams, higher level system dynamics that rely on interactions between several system components are difficult to comprehend through examination of state machine diagrams alone. This presents another use for activity diagrams; they tie low level dynamics of individual entities together into system wide be-

haviours.

Depicting feedback with the UML.

There are aspects of the biological system that we have not been able to satisfactorily express using the UML. The biological system of interest is heavily governed by the interactions of feedback mechanisms. Activity diagrams can demonstrate the order in which critical interactions and events must take place for a high level behaviour to manifest, however they incorrectly imply that one activity stops and another starts. In reality the entity responsible for a preceding activity does not hand off control to that which follows, it continues and can potentially perform the same activity again. This concurrency amongst system elements can manifest in feedback, where an increasing number of elements engage in some activity. Relative population dynamics play a significant role in this biological system (for example the interplay between CD4Th1 and CD4Th2 cells) and it is important to communicate this information in the model. Owing to their ability to express any abstract concept across any number of system elements, we believe that the modification of activity diagram syntax and semantics can yield an appropriate medium for the expression of feedback. This forms ongoing research.

In conclusion.

We have found UML to be a reasonably expressive medium in which to represent this biological system, however that are aspects of the system for which this is not the case. Future work entails the development of a simulation of the biological system with which we intend to integrate known biological data and perform in silico experimentation.

References

1. Hugues Bersini. Immune system modeling: The OO way. In Hugues Bersini and Jorge Carneiro, editors, *ICARIS*, volume 4163 of *Lecture Notes in Computer Science*, pages 150–163. Springer, 2006.
2. David Harel. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8(3):231–274, June 1987.
3. Colin G. Johnson, Jacki P. Goldman, and William J. Gullick. Simulating complex intracellular processes using object-oriented computational modelling. *Progress in Biophysics & Molecular Biology*, 86(3):379–406, 2004.
4. Vipin Kumar. Homeostatic control of immunity by TCR peptide-specific Tregs. *The Journal of Clinical Investigation*, 114(9):1222–1226, November 2004.
5. Vipin Kumar and Eli Sercarz. An integrative model of regulation centered on recognition of TCR peptide/MHC complexes. *Immunological Reviews*, 182:113–121, 2001.
6. Loui Thomas Madakamutil, Igor Maricic, Eli E. Sercarz, and Vipin Kumar. Immunodominance in the TCR repertoire of a TCR peptide-specific CD4+ Treg population that controls experimental autoimmune encephalomyelitis. *The Journal of Immunology*, 180(1):4577–4585, April 2008.
7. Fiona A. C. Polack, Paul S. Andrews, and Adam T. Sampson. The engineering of concurrent simulations of complex systems. To appear in CEC '09.