

Developmental Simulations with Cellerator

Bruce E. Shapiro
Jet Propulsion Laboratory
M/S 126-347, 4800 Oak Grove Drive
Pasadena, CA 91109
bshapiro@jpl.nasa.gov

Eric D. Mjolsness
Jet Propulsion Laboratory
M/S 126-347, 4800 Oak Grove Drive
Pasadena, CA 91109
Eric.D.Mjolsness@jpl.nasa.gov

ABSTRACT

We will describe how Cellerator can be used to perform developmental simulations. Biochemical reactions are specified in Cellerator using a compact arrow-based notation, which are then automatically translated into the appropriate ordinary differential equations. These reactions can be combined into modules, leading to a natural graph-based hierarchical implementation. We demonstrate how the paradigm of organisms-as-graphs can represent the basic features of developing tissue, and propose a variable-structure graph-based algorithm to describe simple developmental processes. In particular, we show how such a variable-structure system (VSS) can be implemented using a pre-packaged fixed-structure differential equation solver.

Cellerator is a Mathematica package designed to facilitate biological modeling via automated equation generation. The implementation is based on the concept of a hierarchy of *canonical forms* that describe biological processes at various levels of detail. At each level of hierarchy there are two classes of canonical forms: the *input canonical form*, which is used to supply information to the program, and the *output canonical form*, which is automatically generated by the simulator.

To understand canonical forms, consider the usual method of describing biological systems in terms of signal transduction networks (STN). Such networks are by their very nature hierarchical; the individual nodes of an STN may represent anything from single molecules (e.g., a particular enzyme or receptor molecule), through simple modules composed of a set of ubiquitous processes (e.g., a MAPK cascade or a particular transcription complex), to extremely complex STNs. The links between the nodes represent their (mutual) interactions including input and output signals. At the highest level of abstraction, a cartoon description of the network itself would be the input canonical form, and a complete set of differential equations describing the network would be the corresponding output canonical form.

The Cellerator paradigm is based on the proposition that there is a one-to-one relationship between each class of interaction and/or node in an STN and a formal (e.g., mathematical) description of that process. An input canonical form for an enzymatic reaction, for example, might be written as $E(S \Rightarrow P)$ meaning that enzyme E facilitates the conversion of S (the "source") into P (the product). One corresponding output canonical form would be the set of differential equations determined by the law of mass action. A slightly different arrow notation, such as $E(S \rightarrow P)$ could instead indicate that the desired output canonical form should be the steady state (Michaelis-Menten) equations. Cellerator represents STN nodes by *variables* (e.g., concentration of a chemical species) and links by *arrows* (e.g., for enzymatic reactions).

A wide variety of different arrow notations have been implemented in Cellerator to describe the input canonical forms for biological systems. The output canonical forms produced by Cellerator take two forms: reaction networks (standard biochemical equations), and systems of ordinary differential equations (ODEs) for the concentrations of chemical species. While no cartoon-based GUI interface has yet been implemented, the Cellerator paradigm allows for the addition of a graphical front-end that could produce the necessary input canonical forms.

The Cellerator implementation allows explicit output description at each level so that "power-users" can modify the equations at any stage desired. They can thus manually modify the equations or add additional constraints in the form of differential, algebraic, or chemical equations. The output canonical forms are produced in a variety of formats: as Mathematica differential equations, in C, FORTRAN, SBML, MATHML, or HTML. If desired, the user can also solve the equations numerically (using Mathematica's NDSolve).

Multi-cellular developmental processes have (at least) one additional layer of complexity. In this case it is possible to represent a tissue, such as the shoot meristem of a plant, as a mega-STN where the nodes represent cells and the links represent inter-cellular interactions. At many stages of development it should be possible to describe the essential processes with only a small number of cells, so the number of nodes does not present a computational obstacle. Complexity arises from the interactions. There may be many different signal transduction networks that need to be represented within any given cell, each of which may be mutually interacting, both intra- and inter-cellularly. In particular there will probably be many instantiations of essentially the same network (e.g., mitotic oscillators) in each cell. Even worse, there could be multiple instantiations of the same network (e.g. MAP-Kinase cascades or transcription complex formation) within the same cell. An even more difficult issue is how to deal with birth and death processes. Since either the birth or death of a cell would cause the total number of cells to change during a simulation, the corresponding number of differential equations necessary to describe the overall STN changes with time: we have a variable structure system.

We postulate that the paradigm of organisms-as-graphs can represent many of the basic features of developing tissue, and propose a variable-structure graph-based algorithm to describe simple developmental processes in this paper. In particular, we show how such a variable-structure system can be implemented using a pre-packaged fixed-structure differential equation solver. We then proceed to show how this has been done in Cellerator, and finally illustrate the overall process with the design of a minimal system for developmental simulation.