CAD/CAM Software for Synthetic Biology

Frank T. Bergmann[1,2]*, Deepak Chandran[1], Herbert M. Sauro[1]
1. Department of Bioengineering, University of Washington, Seattle, USA
2. Computational & Systems Biology, Keck Graduate Institute, Claremont, USA
*email: fbergman@u.washington.edu

Abstract

This poster describes a novel software application that we are developing that will help researchers in Synthetic Biology to design, simulate and fabricate biological systems.

The software system is designed from the ground up to be extensible. It features a rich graphical platform, integration of a python script engine, an application plug-in architecture and is compliant with the Systems Biology Workbench.

A new modular model definition language is under development that will provide the default file format for this tool. The new format will be compatible with the de-facto standard SBML. Thus many software tools will be able to consume models created by our tool.

Introduction

The BioBricks parts registry is a repository of standardized, reliable biological parts enabling synthetic biologists to design and build biological systems. The process of designing, testing and fabricating new systems is still however arduous and error prone.

The software application that we are developing is designed to make the process of design and construction simpler. We focus our attention on two areas: Standard formats and an intuitive extensible software system based on standards.

Standard formats are especially important in this field. Looking back at the success of digital and analog electronics, we find that this success is partly due to the standardization of electrical components. This standardization allowed the creation of computer aided design (CAD) tools that allowed engineers to focus on designing new devices, rather than issues of compatibility between parts. With languages such as VHDL, it became easy to characterize and fully test new devices using CAD tools, before building them in hardware. BioBricks is a first step in this direction for synthetic biology.

Our software system will accompany synthetic biologists through all stages of design, testing and fabrication using the following approach:

- First a generic model implementing the desired functionality is built and simulated.
- Next constraints on the design are made, such as qualifying the host organism, product outputs and input types.

- Based on those constraints on the generic model, the software will deliver a set of concrete designs from a predefined library of parts.
- In a last step one of the concrete designs can then be built by the biologists, or sent to a biological hardware fabrication company that will be contracted to build the system.

Approach

The software system we developed is based on the .NET framework 2.0. Creating a flexible plug-in architecture has been a primary focus during the design. Apart from plug-in interfaces we also integrated the IronPython runtime, allowing expert users to extend the capability of the software by Python scripts.

This plug-in system was used in the development process for all components except for the core-graphics system, thus it is easy to exchange development modules by more readily accessible components that we envision at a later stage of the project.

A growing library of parts, initially derived from BioBricks sequences and characterized by initial parameters is included with the project. Biologists can make use of this database, alter parts on their local copies, or submit parts for review.

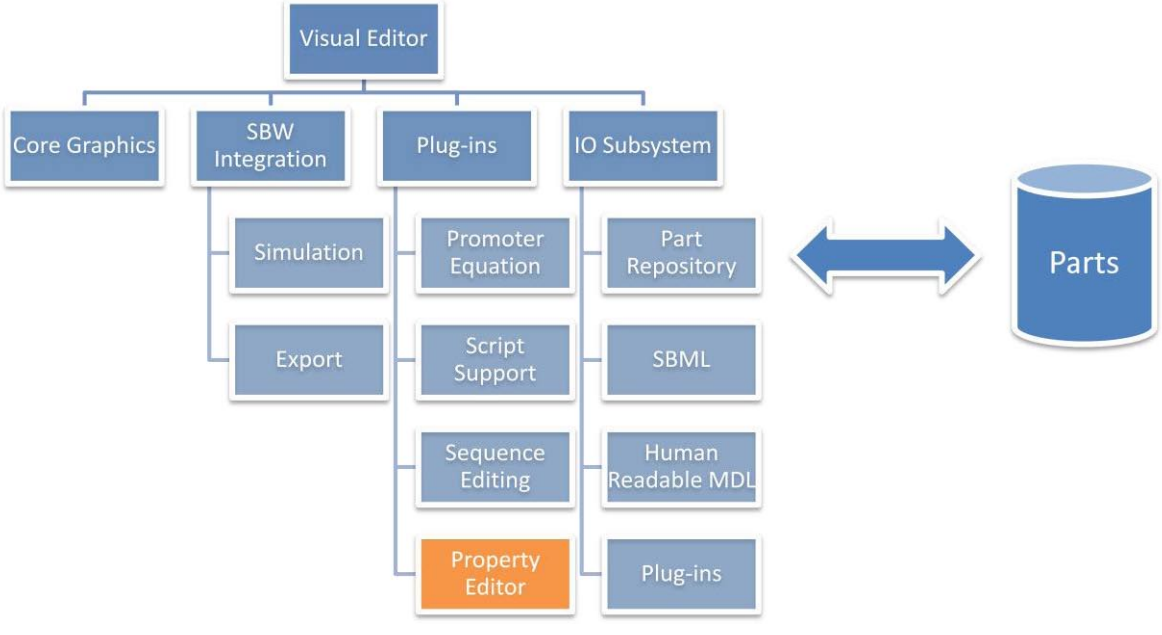

Figure 1: Architecture of the Prototype

Results

We have created a working prototype of the application that allows a user to build biological systems from parts and to combine them into more complex devices. Models created, if kinetically characterized, can be simulated within the application. Models can also be exported into the de-facto standard model definition language SBML, and thus be analyzed by more than one hundred third-party software applications. We have also integrated our prototype with the Systems Biology Workbench and thus all analysis

facilities of SBW are available to the user.

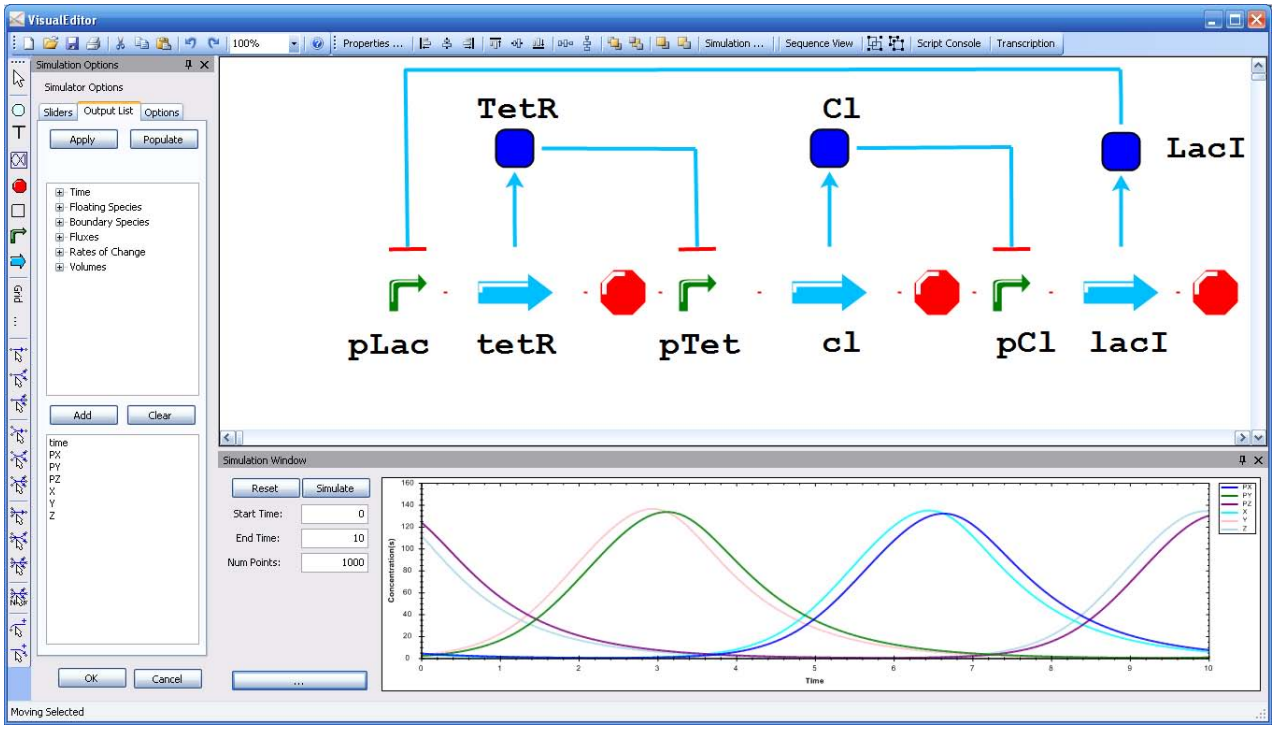

Figure 2: A Screenshot of the application in its current state, simulating a Repressilator model

Funding

This work is supported by a Microsoft Computational Challenges in Synthetic Biology grant.

References

Bergmann, F., & Sauro, H. (2006). SBW - a modular framework for systems biology. *Proceedings of the 38th conference on Winter simulation* (pp. 1637 - 1645). Monterey, California: Winter Simulation Conference .

BioBricks Parts List. (2002). Retrieved August 30, 2007, from MIT Registry of Standard Biological Parts: http://parts.mit.edu/

Hucka M, F. A. (2003). The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* , 19(4):524-31.

IronPython . (2007). *A fast Python implementation for .NET and ECMA CLI*. Retrieved August 30, 2007, from http://www.ironpython.com/

Knight, T., Rettberg, R., Chan, L., Endy, D., Shetty, R., & Che, A. (2003). *Idempotent Vector Design for the Standard Assembly of Biobricks.* Retrieved August 30, 2007, from http://people.csail.mit.edu/tk/sa3.pdf