Draft

Programming Life - Requirements Analysis and Design

Group 5/E:
Felix Akkermans
Niels Doekemeijer
Thomas van Helden
Albert ten Napel
Jan Pieter Waagmeester

March 2, 2012

Contents

1 Introduction

The purpose of biotechnology is to use micro-organisms like moulds and bacteria for the production of chemical substances. These chemical substances are used as energy (biofuels), in products (bioplastics), in food and in medicine. Not only can these micro-organisms produce, they can also be used to break down chemical substances, for example in the purification of water.[1]

Using molecular biology, these micro-organisms can be altered to add, change or remove functionality. This is done by mutating the DNA of the organism. Synthetic biology is the science in which biology and engineering are combined to create new biological functionality[2]. The engineering part is made easier by using biobricks[3]. These "bricks" are simple genetic circuits which provide a basic functionality (for example the imitation of an OR or AND gate) and share a common interface. By combining these biobricks, a new biological systems can be engineered.

In its abstract form, this system can be visualized as a logical circuit, but in the organism the circuit corresponds with a group of molecules (proteins, genes, RNA) that react with each other. How (and if) they will react depends on many different elements, like the concentration and reaction speed of a protein. This reaction can be simulated using a (heavily) simplified model.[1]

A logical circuit of biobricks can be defined using the System Biology Markup Language (SBML). SBML is a free and open interchange format for computer models of biological processes. [4] It is a widely supported format that continues to evolve and expand.

In this report, we will describe the design of a visual modeling environment for synthetic biology, in which biotechnologists can design, simulate and validate a logical circuit built using biobricks. We will clarify our proposed system using a list of requirements (2.2-2.4), a few analysis models (2.5), a business object model (2.6), a few dynamic models (2.7) and a preliminary drawing of the interface (2.8). In chapter 3 we will give a planning of the following of the project.

2 Proposed system

2.1 Overview

In this document, we will provide an analysis and design proposal for a visual modeling environment for synthetic biology, in which biotechnologists can design, simulate and validate a logical circuit built using biobricks.

We have made up a list of requirements which answer the following questions. What does the application have to do? What doesn't belong in the basic functionality? What kind of programming language will we use to develop this program and when is it due? First we will go into detail on functional requirements (2.2), so what does the program do? Secondly, we will discuss non-functional requirements (2.3). These answer questions like: What programming language will we use, but also how we will increase the usability of the application. Finally we will have a look at constraints (2.4).

Following the requirements we will specify a few use case models (2.5). These models describe specific scenarios of the program, what steps the user has to take to reach a goal and how the system should react to these steps. The main scenarios we will specify are loading/saving, modeling and simulating. Dynamic models such as sequence diagrams and activity diagrams will visualize the steps and interaction of the user and system in chapter 2.7.

Our business object model (2.5) will clarify the key concepts and their roles of our application. It will give a simplified overview of how proteins, biobricks and the System Biology Markup Language (SBML) relate to each other.

Last but not least, we will show a preliminary drawing of our interface (2.8) and explain why we have chosen for this interface and how it will work. Chapter 3 contains a planning for the rest of the project. This schedule is built up around the deadlines for the other documents and has a preliminary planning for the implementation fase.

[1] Programming Life – Contextproject assignment (Dick de Ridder, Marcel Reinders)
[2] Programming Life – Contextproject kickoff (Dick de Ridder, Marcel Reinders)
[3] http://biobricks.org/
[4] http://sbml.org/Main_Page

2.2 Functional requirements

The following are use cases the application must be able of completing in whatever way.

1. **Circuit abstraction** The application must be able to abstract a designed circuit into a BioBrick, and then treat this as a gate, so it can be used in building of subsequent circuits/BioBricks.
2. **Protein specification** The user must be able to manually specify which protein is used to represent which signal in a circuit.
3. **Available proteins** The application must be able to present the user with an overview of available proteins to assign to signals. The available proteins must be predefined by a user.
4. **Export XML** The application must be able to export a built circuit as a BioBrick in XML (in the SBML schema).
5. **Import XML** The application must be able to import XML (in the SBML schema) into the application as a BioBrick.
6. **Interfering/invalid signals** The application must be able to notify the user that the current circuit design has protein(s) assigned to multiple signals, or still has signals which do not have a protein assigned.

2.3 Pseudo-requirements

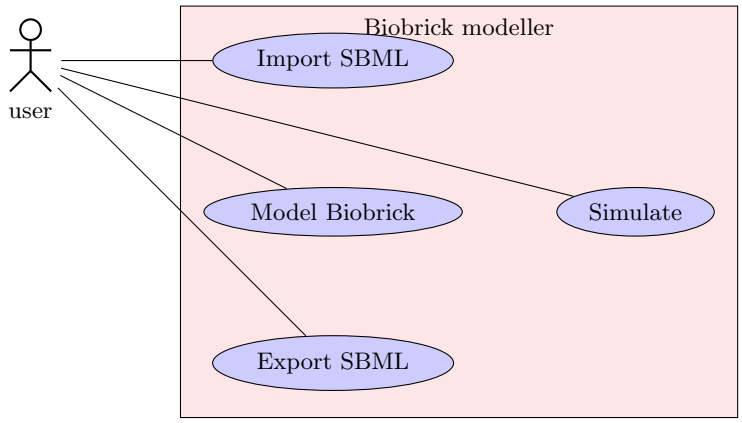
Non-functional requirements are requirements about how a the application behaves and operates (usability, accessibility, availability, performance). In this section we will also cover constraints on the source-code and development (programming language, documentation, tools/frameworks, compatibility, extensibility, maintainability, testability, deadlines).

1. **Documentation** For documentation we will use LaTeXto finalize the reports and will be written in English. GIT will be used for hosting and collaborating on documentation.
2. **Tools/frameworks** The use of other software development tools has not been decided upon yet.
3. **Testability** We want to be able to test every detail. We were discussion test drive development, but we have not fully decided upon that either. In any case, there should be a lot of tests, in the form of checks and sort of unit-tests, but also for user friendliness by real people.
4. **Usability** Because the program is not made for information technology students so we will spend some time increasing the usability. However, the target group consists of scientists, and to be more specific, mainly bio-informatics experts. Because of this we will not create step-by-step tutorials. Our aim is to make the application basic enough so that a regular scientist can work with the application.
5. **Extensibility** We want to create our application in such a way that BioBricks can be exported and reused later as part of even greater BioBricks. As for the application, it is always best practice to make the application extendable, and so we shall aim for that.
6. **Maintainability** Maintainability is actually not one of our focuses. Because the application works on its own, and is only required to be developed until the end of this semester, this is not a major concern.
7. **Accessibility** Because our application frontend will be written for the web, we will host it somewhere and it should be accessible from anywhere, if there is internet. From the time of launch of our first version until the end of the project, there should always be a testable version hosted somewhere. It's also likely that we will use modern frameworks such as Bootstrap CSS to provide a high start-level of usability, although this has not been decided yet. The use of such well developed frameworks guarantees a high accessibility of the website.
8. **Deadline** The first version is up for evaluation on the fourth of May (04-05-212). The final version of the application is due on the fifteenth of June (15-06-2012).
9. **Biological plausibility** The application does *not* have to give an biologically accurate or plausible simulation.
10. **Frontend** The GUI frontend of the application will be a website written in HTML+CSS+JavaScript. This means our application frontend will be compatible with all the big operating systems that support web browsers.
11. **Backend** The backend will be written in Java Server Pages.

2.4 Analysis models

2.4.1 Use case model, descriptions and scenarios

Figure 1: Use cases

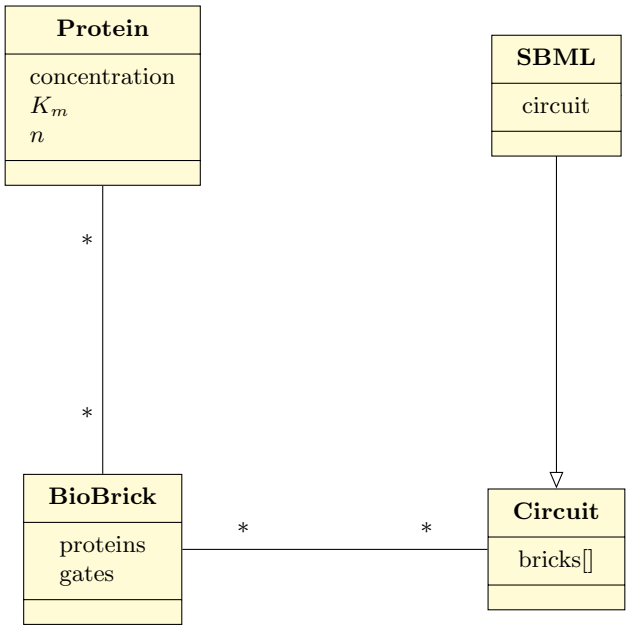


2.5 Business Object model

The objects of our model are the proteins, the biobricks, the circuits and the Systems Biology Markup Language.

The protein is the smallest building block of the model, with proteins and gates we can form Biobricks. These Biobricks can then be connected to each other to form a circuit. The Systems Biology Markup Language (SBML) can be used to represent and describe these circuits. So from the circuit the resulting SBML can be obtained and vice versa.

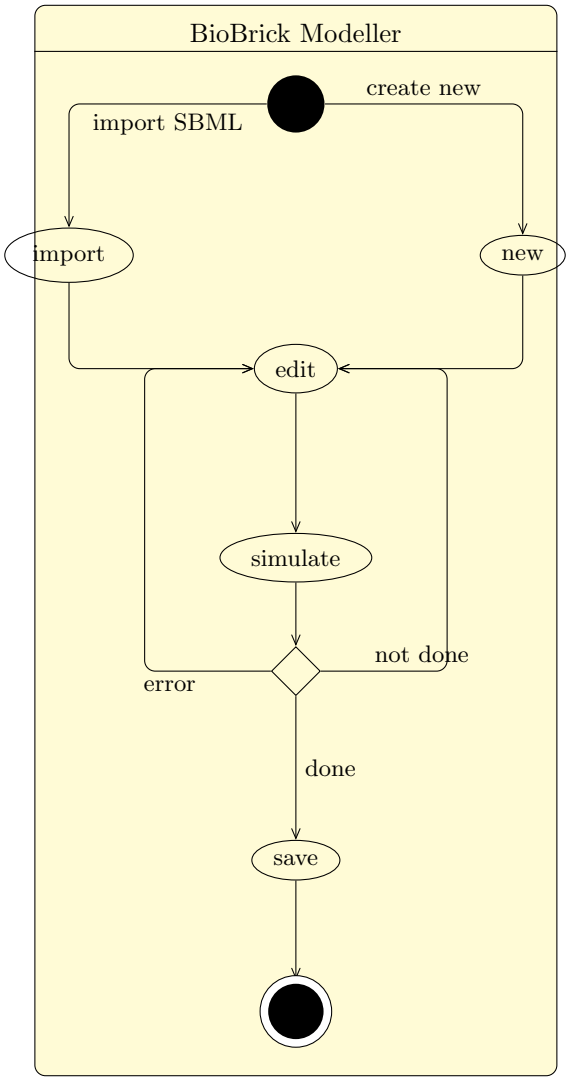
Figure 2: Business Object model



2.6 Dynamic models

When the user enters the program, two options are availible, he can import an existing circuit (SBML-file) or create a new one. The user then edits the circuit to suit his needs. After the editing the circuit can be simulated, if the user choses the proteins wrong an error will occur and more editing will be in order, if the user decides he is not done yet with editing he can continue after the simulation. When the user is done he can save the circuit as a SBML-file.

Figure 3: Activity diagram

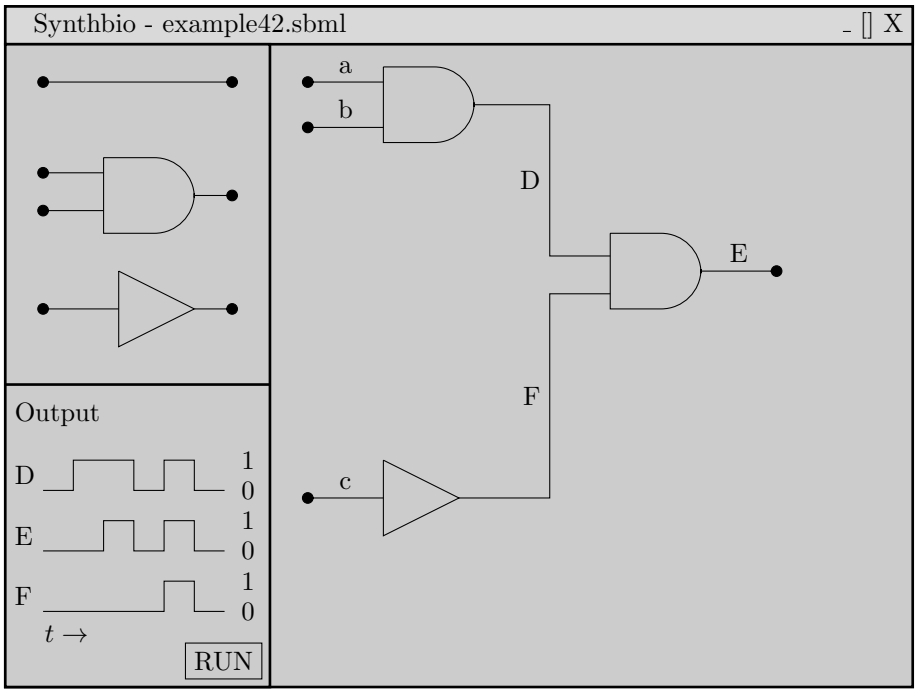


2.7 Interface

To model a network of circuit from the basic gates provided, a simple GUI is proposed in figure 2.7. Three different actions can be performed in the main working area: adding wires, AND- and NOT-gates. For each wire/signal a protein can be selected.

The simulation can be run, and the result will be displayed in a simple graph.

Figure 4: Quick mockup of the Graphical user interface



3 Planning

Date	Deliverable/Event
02.03.2012	Deadline **draft** RAD document
09.03.2012	Deadline RAD document
Q3 W5	First Peer-Review
10.03.2012	Deadline **draft** Architectural Design document
14.03.2012	Deadline **draft** Test and Implementation plan
16.03.2012	Deadline Architectural Design document
23.03.2012	Deadline Test and Implementation plan
04.05.2012	Deadline upload source code to SIG for first evaluation)
Q4 W4	Second Peer-Review
24.05.2012	Feedback SIG evaluation by Eric Bouwers (10:00-11:00, room TBA)
06.06.2012	Deadline **draft** Final Report
15.06.2012	Deadline Final Report
15.06.2012	Deadline upload source code to SIG for final evaluation
20.06.2012	(morning) student presentations within each context
20.06.2012	(afternoon) plenary presentations of the best projects per context

4 Glossary

BioBrick Isolated and documented cell function to be reused in future projects. For example, the production of a light emitting protein when some other protein is available.

Circuit What we think of as a circuit is actually just a cell wherein all signals and gates are proteins and parts of the DNA, mixed together without any separation.

DNA Deoxyribonucleic acid is a very long molecule containing the information needed to support the life in allmost all creatures.

Protein a complex molecule produces by certain processes in the cell. In turn it can activate other processes, or perform functions such as emission of light or change color.

RNA Working copy of the information in the DNA. This copy is then used to synthesize proteins.

SBML *Systems Biology Markup Language* is a XML-based format for storage of various computermodels of biological processes.

Simulation Execution of a set of differential equations in order to predict the output of the modelled circuit

Transcription Process of making the mRNA-copy of the DNA master.