Programming Life - Final report

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

June 11, 2012

Contents

1 Introduction

2 Key problems and solutions - highlights

During this project we have encountered some problems. In this section we will have a look at some of the highlights of these problems and how we fixed them.

Our program was developed test driven. This makes for very neat code, but it also comes with some problems. We mainly encountered problems with QUnit, a JQeury testing framework similar to Java's JUnit. Although the testing was done properly, the feedback on the results was not very helpful. Eventually we managed to tackle this problem with some extra manual testing.

A common issue using server-client structure is the actual connection between the two. In the beginning we had some trouble with this but this was mainly caused by out of sync code and trouble with Apache Tomcat. We decided on a standard version of Tomcat and wrote a setup page on the wiki. Afterwards, the problem was solved.

Another problem client side was the dragging and dropping of gates and connecting them with wires. We had anticipated this as being tough and found a solution in jsPlumb, a solution with its downsides. The framework jsPlumb makes it easy to draw wires between gates, but it didn't fully meet our needs so we had to customize it, leading to quite some hours of extra work. (NIELS MORE EXPLANATION)

At one point, we figured we wanted to select our proteins from dropdown menus on our wires. Regular dropdown menus from Javascript are not an issues, however combining this with jsPlumb and bootstrap proved to be a little more difficult. This took more time than we had anticipated but was eventually solved after trying several options.

Throughout this project we worked with GIT to share our code. We also used the ticket mechanism to keep track of what has to be done. GIT is exceptionally good at merging code but it can go wrong. If there is too little communication between two people who are working on the same code, merging can become difficult. However, with some reverts and recommitting these problems were resolved.

Towards the end of our project, we did some major restructuring. We shifted some functions around, changed some names and did some more testing and commenting. We did this based on the SIG evaluation. The biggest changes were made on the client side.

Our program is not only to model circuits but also to calculate and display the outcome. We displayed the outcome in a graph, but this showed some weird results. It turned out that the solver for our circuits had some issues (ALBERT EXPLAIN HOW IT WAS SOLVED).

3 Reflection on the teamwork

4 Individual reflection on the project

4.1 Felix Akkermans

4.2 Niels Doekemeijer

4.3 Thomas van Helden

4.4 Albert ten Napel

4.5 Jan Pieter Waagmeester

5 Lightweight SCRUM Plans

The following pages will include the relevent parts of our lightweight scrumplans for each iteration. The colored and numbers with a hashtag refer to GitHub issues on https://github.com/FelixAkk/synthbio/issues. For each iteration, a milestone is available providing a nice overview of the issues for each iteration.

	Task	Developer	Effort estimated	actual
#2	Server: Tomcat server that handles HTTP requests HELLO WORLD and class scaffolding	Niels/Albert	2*10	17
#3	Server/Client: Design and document JSON format for circuit (`.syn`) and protein list.	Jan Pieter	4	3
#4	Client: JavaScript scaffolding	Thomas	8	10
#5	Client: Create basic GUI	Felix	8	12
#6	Server/client: create and display connection state by ping/pong heartbeat	Niels/Albert	2	3
#7	Client: Show available basic gates	Jan Pieter	3	1 + 2
		Total	45	48
	Optional tasks[1]			
#8	Server: Parse and serve list of Proteins[2]	Jan Pieter[3]	4	16
#11	Cient: Display list of Proteins	Jan Pieter[4]	2	2
		Total	6	18
	Added during iteration[5]			
-	Ant build evironment	Jan Pieter	-	2
		Total	-	2
		Grand Total	65	68

3 Reflection on this iteration

In this section we will give a quick review on this iteration. First we will comment on our initial planning and second we will review the implementation phase and the workflow.

3.1 Planning

The estimated efforts were not that far off the actual time needed for implementation and we are confident that our estimations will improve as the project continues. We did, however, forget to plan a few obvious tasks. For example, code reviewing, tool selection (testing HTML responses and JSON-serialisation) and creating a building environment are tasks that could have been foreseen. Also the time needed for everyone to setup their working environment was underestimated.

A lot of these tasks were one-time efforts, but they can consume precious time. We think the number of these small tasks will reduce, as this was the very first iteration. We now have a starting point that can be built on. For next iterations, we really have to think things through, to make sure there will be as little unforeseen subtasks as possible.

[1] Things from next iterations that could be done if sufficient time is available
[2] Only after list is supplied by Alexey
[3] Original plan was *Niels/Albert*, but Jan Pieter finished his tasks...
[4] Initially unassigned
[5] Significant tasks not planned before.

3.2 Implementation

The implementation phase for this iteration didn't go as smoothly as we hoped it would go, but it did gave us a few points to work on. One of these points is communication; there was little communication between members. This, combined with a low number of commits, made it hard for members to see what everyone was working on and what the overall progress was. Only when the deadline was approaching, the commits started to flow and the project advanced quickly. Late commits make code reviewing and integration testing harder. That is why we want to commit early and commit often in the next iteration.

The client-side scaffolding didn't go as planned either. One of the issues was the QUnit testing. Implementing tests works well, but the feedback once something goes wrong is often rather useless. So the debugging took more time than expected. The Javascript scaffolding was not completed within the scheduled time because of a lack of communication (as more people could have helped). This has been pointed out and will be improved. In the next iteration, this point has priority and will be finished as soon as possible.

We are happy with the delivered code and product. Test-driven development was applied correctly and we think we have built a solid base. For this base we needed a few (previously undocumented) tools:

Build environment

We decided to use Apache ant to build the sources and provide easy access to the JUnit test suites. Currently, four targets are present.

JSON serialisation

Selected the library from http://json.org/java since it is lightweight.

HTML/JSON Unit testing

We have selected HtmlUnit (http://htmlunit.sourceforge.net/) to help test the server applets. This unit acts as a browser (without the GUI) and fetches pages from the server. This helps testing the JSON responses. Unfortunately, we have found no easy way to compare JSON objects in Java. That is why we are comparing the objects as strings. JSON objects are unordered by definition, but we make sure the objects have a predictable layout (alphabetical order).

Project title

After the name confusion about BioBricks we wanted to clear this issue once and for all. In this scrum the term still floated around, for example in the GUI. So we decided on a name for the project to be used from here on everywhere in our work; **project Zelula** (meaning cell in a foreign language [6]). In Scrum 2 this should be apparent in the GUI, documentation, code comment headers.

[6] http://eu.wikipedia.org/wiki/Zelula

	Task	Developer	Effort estimated	Effort actual
#18	Server: Save circuit	Jan Pieter	12	16
#19	Server: Load circuit	Jan Pieter		
#20	Server: Validate circuit	Jan Pieter		
#26	Server: Serve saved circuits	Jan Pieter	3	2
#21	Server: Connection to simulator	Albert	12	15
#27	Server: Convert circuit to simulator input	Albert	3	-
#4	Client: Javascript scaffolding	Thomas	10	12
#22	Client: Gate scaffolding and rendering	Felix	14	- [1]
#24	Client: Drag-and-drop gates to working area	Felix+Niels		7+3
#25	Client: Move gates in the working area	Felix		- [2]
#30	Client: Draw wires between gates	Niels	14	12
#31	Client: Draw input/output wires	Niels		8
#28	Reflection scrum plan 2	Thomas	3	2
#29	Scaffolding scrum plan 3	Thomas	1	1
#32	Code review	Everyone	5 * 4	11
		Total	92	89

	Optional tasks[3]			
-	Server: Simulate circuit	-	-	-
-	Client: Validate circuit	-	-	-
-	Client: Load circuit	-	-	-
-	Client: Save circuit	-	-	-
-	Client: Specify proteins for wires	-	-	-
		Total	0	0

		Grand Total	92	87

3 Reflection on this iteration

In this section we will give a quick review on this iteration.

3.1 Planning

From the beginning of the iteration we spend a decent amount of time on planning. During the first meeting we mainly established our plan, dividing tasks and estimating how much time it would take us. From the previous iteration we learned that things will always take more time, especially if you take into account the extra hours you spend documenting and reading in on your work. So we added a small amount of hours on top of our expectations which brought us closer to our real time spent. We added the hours for code reviewing. So overall we improved on the planning part with respect to the first iteration.

[1] This issue is moved to scrum #3. It was not worked on due to issue #22 taking longer than expected.
[2] This issue was not implemented because it's included in the jsPlumb framework. See paragraph 3.2.
[3] Things from next iterations that could be done if sufficient time is available

3.2 Implementation

As for the implementation, things went well. We did run into some issues, but eventually they were fixed. Main problems were drag and drop for our gates and old documentation.

Dragging and dropping gates had some issues. This also had to do with jsPlumb implementation. We managed to drag and drop a gate, so the rendering part is finished. However, we cannot extract a JSON representation of our circuit yet. Old documentation was one of the reasons our work was getting out of sync. We established an HTTP API so we could work based on that.

During the previous iteration we had already thought ahead about one specific problem. We thought that drawing wires would prove to be one of the major issues during this iteration. However, during this iteration we discovered jsPlumb which made things a lot easier and good looking. Using this framework also meant some features are implemented out-of-the-box, saving time and bringing focus more to high-level functionality.

Furthermore, we really noticed the benefits of code review during this iteration. Many small bugs were easily fixed by just having another person review your code.

Connecting to the simulator input took extra time as well, which lead a delayed implementation of the conversion of a circuit to simulator input. This task will be pushed on to the next sprint.

The test driven development for the JavaScript scaffolding didn't go according to plan, because we instantly apply functions to our Ajax results, which makes it hard to test the intermediate results. However, through manual testing we still managed to do some proper testing.

jsPlumb

We chose to use one extra tool which was not mentioned before. This is because we only recently discovered it. We used jsPlumb (`http://jsplumb.org/`) as framework for the (visual) connections in the GUI. This framework makes it easy to define anchors and draw wires between these anchors. Unfortunately, the framework did not match up to all our wishes. Small changes to the library were necessary to allow jsPlumb to automatically create anchors from which multiple wires can be drawn (these kind of anchors are used in the connector from which input signals originate).

	Task	Developer	Effort estimated	Effort actual
#42	Client: Specify protein for wires by drop-down menu	Thomas	3	12
#44	Client: Specify input signals	Jan-Pieter	16	20
#22	Client: Circuit bookkeeping	Felix+Niels	2*15	5+15
#45	Client: Save Circuit	Felix+Niels		5+0
#46	Client: Load Circuit	Felix+Niels		8+4
#47	Server: Simulate Circuit	Albert	2	2
#48	Server: Convert circuit to simulator input	Albert	14	16
#49	Wiki: Simulator	Albert	1	0
-	Code Review	All	5*4	2+3+4+1+1
#50	Final Scrumplan-3	Thomas	2	3
#51	Scaffolding Scrumplan-4	Thomas	1	1
#52	Final report: Key Problems/Solutions	Thomas	4	0
		Total	93	92
	Optional tasks[1]			
#43	Polishing modelling/grid styles and workflow	Felix+Niels	-	1+1
#53	Client: Output visualisation	-	-	-
#54	Client: Simulate Circuit	-	-	-
#55	Final report: Reflection on teamwork	-	-	-
		Total	-	2
	Added during iteration			
-	JSLint ant target and style review.[2]	Jan Pieter	-	2
-	Client: validate Circuit[3]	Jan Pieter	-	2
		Total	-	4
		Grand Total	93	98

[1] Things from next iterations that could be done if sufficient time is available
[2] To ensure some best practices in our JavaScript code, including coding style, we use JSLint/JSHint
[3] For some reason, this task was not in the initial scrum plan #3, but it was an optional task in #2.

3 Reflection on this iteration

During this iteration one of the main problems we encountered were in late commits and a lack of communication when someone was getting behind on schedule. This would lead to a lack of time to correct late work or to help someone catch up with the schedule. During the scrum-reflection we discussed this and noticed that as before this is still a problem. Some improvements were noticed this scrum iteration, and we further emphasized that this is still to be worked on and requires more attention during in our group-process. Everyone agreed on this and plans to show improvements in this aspect.

Extending on this issue we also noticed that at the end of the scrum iteration there were still quite some issues open that should have been closed. This requires more attention so the state of the project issues on GitHub more accurately reflects it's current state in the code.

Another thing we noticed is that our actual time expenditure on code review was below of that what was planned. We adjusted our expectations for the next scrum iterations so this would be more realistic.

3.1 Planning

We had planned to quite some work this iteration even though there was a holiday during this scrum run. We were requested to have a working program during the next sprint, so we looked at what was still missing and aimed to complete a great proportion of that during this sprint. We mainly shoved the Circuit Simulation on the client side to the next scrum run. The rest of the big chunks we tried to tackle this iteration. We also had a little bit of work left from the previous sprint so we planned that in as well. We acknowledged that our final report will be a big piece of work as well, so we decided to work a little bit in advance on that.

3.2 Implementation

During the implementation of all of our functions we encountered some difficulties. We had some difficulties implementing dropdown menus on wires. We didn't expect this but it seems jsPlumb combined with bootstrap gave quite some errors. We didn't manage to fully complete this part so we pushed the remainder on to the next sprint.

Other than this we mainly made a lot of small improvements to our program. We did not encounter great problems, instead we found a lot of room for minor improvements. We took these enhancements and set them as optional in our next sprint.