

Project Zelula - SCRUM plan #2

Group 5/E:

Felix Akkermans

Niels Doekemeijer

Thomas van Helden

Albert ten Napel

Jan Pieter Waagmeester

May 7, 2012

1 Introduction

In this SCRUM plan we define the tasks we want to complete in this sprint. Every task is assigned to one or two developers and an estimate is provided for the effort required.

For each SCRUM run a milestone is created at GitHub, with issues for the tasks selected. The issues for this milestone can be found on: <https://github.com/FelixAkk/synthbio/issues?milestone=9>.

2 Selection and assignment of tasks

Because this is the second iteration, we think we will find less unexpected tasks. The available time is more than last time, so we want to do more work than in the last iteration. For this iteration, the goal is to end up with a GUI that supports basic circuit design and a back-end that supports basic circuit handling (loading/saving/validating).

Available time

The time available is five mornings in two weeks by five people. That's about twenty hours per person, however, some time should be reserved for the meetings. So the total effort available is about 90 man hours.

A list of tasks, assignments and effort estimations is included in a table.

Presentation

Results for this iteration will be presented Friday May 4, 12:15 in DW-PC 0.010.

Task	Developer	Effort	
		estimated	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	↓
#24 Client: Drag-and-drop gates to working area	Felix/Niels		↓/3
#25 Client: Move gates in the working area	Felix		Actual
#30 Client: Draw wires between gates	Niels	} 14	12
#31 Client: Draw input/output wires	Niels		8
#28 Reflection scrum plan 2	Thomas	3	1
#29 Scaffolding scrum plan 3	Thomas	1	Actual
#32 Code review	Everyone	5 * 4	Actual
Total		92	
Optional tasks¹			
- Server: Simulate circuit	-	0	0
- Client: Validate circuit	-	0	0
- Client: Load circuit	-	0	0
- Client: Save circuit	-	0	0
- Client: Specify proteins for wires	-	0	0
Total		-	-
Grand Total		92	-

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.

¹Things 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, but due to the help of JQuery we managed. 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.

We predicted that drawing wires would prove to be one of the major issues during this iteration. However, we discovered JSPlumb which made things a lot easier.

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.

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).