

Weekly Report

Team Dec 15-12: @PaniniJ

Week 9: Mar. 23rd - Mar. 29th

Advisor	Dr. Rajan	
Client	Dr. Rajan	
Team Members	Dalton Mills	<i>Webmaster</i>
	David Johnston	<i>Team Lead</i>
	Kristin Clemens	<i>Communication Lead</i>
	Trey Erenberger	<i>Key Concept Holder</i>

Weekly Summary

In the past week, we have made a number of incremental improvements to the code base. We also continued researching specific details of the capsule system generation process.

Technical Progress

At the most recent Sunday meeting, we identified a fundamental problem with our capsule artifact generation strategy, one which will need to be addressed. We had planned on having an encapsulate template instance inside. This encapsulated template instance was essentially wrapped and delegated to by each capsule class.

We realized that in the particulars of our user input syntax and artifact generation process, the encapsulated template instance's procedures could not have references to other capsule instances; the Java type system could only have references to capsule template instances. This would not produce correct behavior, so we needed to reformulate a new strategy.

After performing some small proof-of-concept work to explore some specifics of the behavior of the annotation processing API, we identified three potential strategies of varying levels of definitiveness, certainty, and risk. In the coming weeks we will be considering and evaluating these alternative strategies and possibly others.

Individual technical contributions were as follows:

- Dalton's original plan for this week was to generate the `run()` method for capsules. During Friday's meeting with Dr. Rajan there were some moments of enlightenment which not only changed the ideas behind the run method and its generation, but also how capsules should get started up and wired together. The group meeting on Sunday further solidified how the approach that Dr Rajan currently uses will be transferred to our project.
- David performed some refactoring of some helper methods involved in building capsules and other artifacts on the `[capsule-generation/master]` branch. This work was also integrated into the `[capsule-generation/type-collector]` branch. The type collector is now also used the construction of other artifacts (e.g. the capsule interface maker). After this, he started working on a better understanding of the properties of capsule systems, and he wrote up some preliminary notes of his thoughts (currently under review by Dr. Rajan).
- Kristin was unavailable to contribute this week.
- Trey explored issues with the current implementation regarding the delegation of classes to the user written template code. Our Sunday meeting featured discussion of this issue at great length and led to three potential solutions. Documentation of this issue was started, but postponed due to time limitations.

Meetings

Weekly Administrative Meeting

Members Present: All

Additional Participants: Dr. Mani

Date & Location: Tuesday 24th of March; Molecular Biology 1414

Minutes:

- The group discussed syntax for the `design()` method of capsules
- We clarified terms and definitions related to the capsule graph, these definitions were further clarified in the meeting with Dr. Rajan on the following Friday.

Bi-Weekly Advisor Meeting

Members Present: Dalton, David, Trey

Additional Participants: Dr. Rajan

Date & Location: Friday 27th of March; Atanasoff 101

Minutes:

- Dr. Rajan explained how capsules are initialized and wired.

Weekly Collaboration Meeting

Date & Location: Sunday 29th of March; Google Hangouts

Members Present: Dalton, David, Trey

Additional Participants:

Minutes:

- Review of the issues regarding the implementation of the run method for active and passive capsules.
- Discussion of the delegation issue (see Trey weekly progress)
- Brainstorm of possible solutions to the delegation issue
 - Trey focused on separation of user written code and PaniniJ generated code.
 - Requires finer granularity when parsing source code to perform transformations of the source and regenerate the model into generated template classes.
 - David provided first solution that stuck with the current API with the stipulation that code can be incrementally generated.

- proof of concept work branched off of the original generator proof of concept. Showed that compiler can work out the order of the generated dependencies
 - David honed solution to relieve burden on User by altering paniniJ syntax
-

Individual Hourly Contributions (Across Two Weeks)

Trey Erenberger	8 Hours
David Johnston	12 Hours
Kristin Clemens	2 Hours
Dalton Mills	8 Hours

Cumulative Time Contribution

Trey Erenberger	91 Hours
David Johnston	110 Hours
Kristin Clemens	54.4 Hours
Dalton Mills	82.5 Hours

Tentative Plans for Week 9

- David is going to continue his recent work on formulating a new capsule instance inheritance strategy and on understanding the properties of capsule systems.
- Trey is going to research possibility of expanding our API capability for parsing the body of the source's methods. This work may improve the viability of completely separating the two 'planes', user code and PaniniJ generated code.
- Dalton is going to look work with Trey on the above-described effort.
- Kristin is going to touch base with David to determine what if any changes need to be made to the previously discussed capsule generation tests.