Weekly Report

Team Dec 15-12: @PaniniJ Weeks 10 and 11: Mar. 30th - April 12th

Advisor Dr. Rajan

Client Dr. Rajan

Team Members Dalton Mills David Johnston Kristin Clemens Trey Erenberger

Webmaster Team Lead Communication Lead Key Concept Holder

Weekly Summary

We made some important design decisions with respect to syntax which will have a big impact on the annotation processor and end-user. We've also finished implementing a prototype that can generate a *working capsule system* which David will present to the Panini Lab on Wednesday.

Technical Progress

We decided to follow the the encapsulated "Capsule Template" design instead of the "split planes" design and the package-manipulation design. We presented all ideas with their pros and cons to Dr. Rajan who made the final decision.

A working prototype capsule generation prototype was created, though it has a number of limitations. These include:

- It has only been tested with a single example capsule system, helloworld.
- It has almost non-existent checks to see that a capsule template is valid and very poor runtime error reporting.
- It does not handled array capsule children which are arrays.

Individual technical contributions included.

- David worked on code to use information from the various capsule template declarations (i.e. init, design, and run) in order to influence the artifact generation process. This involved some debugging work with Dalton. Of particular note,
 - O The run declaration influences the construction of a thread capsule artifact's run method.
 - O The capsule's @Wired field annotations influence the generation of the panini\$wire() in the capsule interface and thread-capsule artifacts.
 - O The capsule's @Child field annotations influence the generation of the panini\$initChildren() method in thread-capsule artifacts.

David also prepared and delivered a presentation on @PaniniJ as part of Tuesday's the senior design lecture.

- Dalton did research on the template parsing solution to the reference problem, which was quickly abandoned because the tools for parsing the template and finding the target piece of (untested/untrusted) code won't be robust enough.
- Trey worked on adding source code generation helper methods that would allow us to refactor certain areas of capsule generation to a cleaner, more consistent code style. This work was triggered as the same problem, aligning multiple lines of code in one format wildcard with Source.format(), came up in multiple places in capsule generation. One specific example is the message constants that are generated for the capsule, which previously used a workaround to make the generated source look normal.

Meetings

Weekly (10) Administrative Meeting

Members Present: All

Additional Participants:

Date & Location: Tuesday Mar. 31st; Coover TLA

Minutes:

- Kristin testing roadmap
- Project plan v2
- Reference problem
 - O David 'newer' method user code uses generated artifacts directly
 - proof of concept works where compile smartly generates
 - O Trey/Dalton method separation of user and generated code
 - need change in api capabilities to access method contents of user source code to transform references
 - O Kristin/Dalton method namespacing to replace references in user code
 whiteboarding kristin/dalton method
 - O discussion on advantages/disadvantages to adding suffix to user class names
 - User may experience confusion when different system elements are named the same.
 - User may not like adding arbitrary syntax to their code
 - O Discussion of Split-plane philosophy
 - requires new tech for api
 - brings potential advantage of clear user written java code as a source
 - may open many potential issues with checking user behavior
 - may cause problems with breakpoints/debugging
 - O some consensus on suffix augmented encapsulated delegated philosophy

Bi-Weekly (10) Advisor Meeting

Members Present: Dalton, David, Trey Additional Participants: Dr. Rajan Date & Location: Friday April 3rd; Atanasoff 101 Minutes:

- Solutions to the reference problem (see weekly (10) administrative meeting above) were presented to Dr. Rajan.
- Drj. Rajan made the decision to go for the Template Design
- After much discussion everyone in the group decided it was the right solution.

• The pros and cons of each method will be included in the next revision of the Project Design documentation.

Weekly (10) Collaboration Meeting

Date & Location: Sunday April 5th; Google Hangouts Members Present: Dalton, David, Trey Additional Participants: Minutes:

- Discussed how design() works, and defined what capsule children and capsule requirements are.
- Dalton and David figured out the signatures for some utility methods for gathering capsule requirements and children.

Weekly (11) Administrative Meeting

Members Present: All

Additional Participants:

Date & Location: Tuesday April 6th; Molecular Biology 1414 **Minutes:**

- David presented @PaniniJ to the class
- Meeting was kept brief because Trey and Dalton had other work
- Trey will work on Source.formatAligned();
- Dalton and David will pair program on Thursday (8th) over Google Hangouts to figure out how to gather capsule children and required capsules.

Weekly (11) Collaboration Meeting

Date & Location: Sunday April 12th; Google Hangouts Members Present: Dalton, David, Trey Additional Participants:

Minutes:

- Made plans for the week
 - O David will work on removing arguments from design()
 - O Dalton and Trey will generate some examples using the new working prototype

Individual Hourly Contributions (Across Two Weeks)

Trey Erenberger	11 Hours
David Johnston	15 Hours
Kristin Clemens	?? Hours
Dalton Mills	20 Hours

Cumulative Time Contribution

Trey Erenberger	102 Hours
David Johnston	125 Hours
Kristin Clemens	54.4 + ?? Hours
Dalton Mills	102.5 Hours

Tentative Plans for Week 9

- David: David is going to look into renaming some elements of the system from PaniniPress to @PaniniJ, and to help with some of the bugs which arise from Trey and Dalton's work creating new example capsule systems.
- Trey: Will be porting the example code that is packaged with paninij-panc to @paninij to test our implementation and highlight what features we are missing and what bugs may occur with more complicated capsule code.
- Dalton: Will be testing out the Capsule Templates through making example programs. These capsule template examples will be based on examples from the PaniniJ project. The goal is to feel out potential future problem areas (and current problem areas) with our design.
- Kristin: TODO