Weekly Report

Team Dec 15-12: PaniniJ Eclipse Plugin
Week 2: *Jan. 26th - Feb. 2nd*

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

Weekly Summary

Our focus this week was to find topics and technologies to investigate and research. Because of our experience talking to Sambhav last week, we had felt that XText would likely suit all of our needs. However, after speaking with Dr. Rajan on Friday, we are no longer sure. In this conversation, Dr. Rajan presented a variety of pros and cons of a number of development strategies.

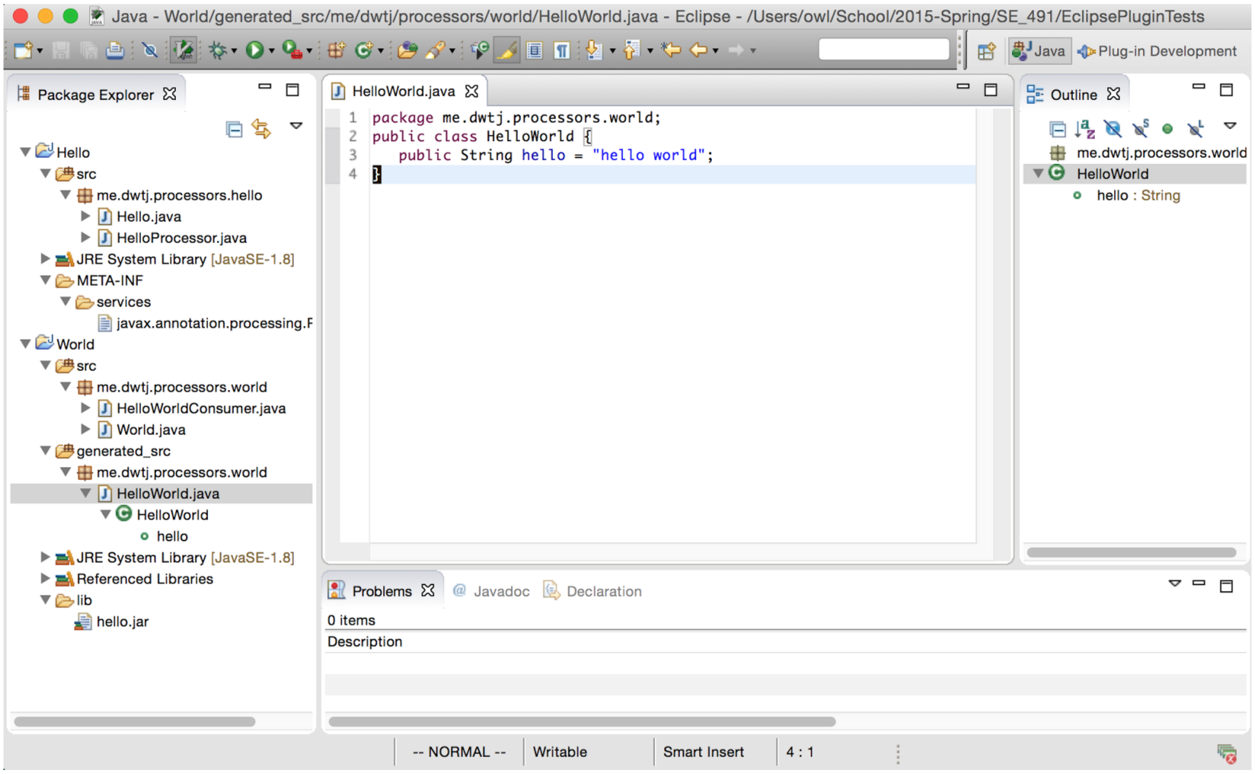
These strategies included one that we had not considered before: automatic code generation and typechecking of capsules via Java annotations. We are very interested in this possible strategy and have begun looking into it.

Each member of the team now has a specific area in which they will be investigating over the next week or two. In particular,

- David is investigating the architecture of the Eclipse framework, code generation via Java annotations, and Java 8 pluggable type checkers.
- Trey is investigating the possible limitations of implementing the project with Xtext. This mainly consist of reading API and documentation regarding Xtext, reviewing existing implementations of java-like languages, and reading available research papers on Xtext.
- Dalton is investigating Java annotations, more specifically, if they can achieve what we're looking for.
- Kristin is investigating Java 1.8 type annotations, in particular how other institutions have used them to effect in their projects.

Technical Progress

David spent time investigating the basics of automatic code generation via Java annotations. He was able to get a simple Hello World example working from within Eclipse. (Projects are available from https://github.com/dwtj/annotation-processor-hello-world.)



```
package me.dwtj.processors.world;
public class HelloWorld {
    public String hello = "hello world";
}
```

Here is the background about what is in this image:

- Hello implements a simple annotation.
- HelloProcessor extends an AbstractProcessor to perform code generation at compile time. This processor is triggered by any declaration with the @Hello annotation.
- World is a class annotated with @Hello.
- hello.jar contains an exported copy of the Hello project.
- HelloWorld is an automatically generated class produced at compile-time.
- HelloWorldConsumer is a class which was not automatically generated but its definition requires the generated HelloWorld class. It compiles successfully.

Meetings

Weekly Administrative Meeting

Members Present: All
Additional Participants: N/A
Date & Location: Tuesday 27 of January; Molecular Biology 1414
Minutes:

- Discussion of Milestones (very rough draft)
    - 0.0
        - Set up development environments, git repository
        - Commit base xText plugin
        - Solidify coding standards
    - 0.1
        - Compile and run PaniniJ program in eclipse
    - 0.2
        - Compilation errors and program output integrated into eclipse
        - Syntax highlighting for PaninJi program
    - 0.3
        - Compiler Errors and Jump to Line
- Discussion of coding style standard
    - 4 spaces per identation; no tabs; otherwise Java standard
- Assigning work to be done before Sunday Coding Session
    - Each person make xtext plugin project
- Talk of formal documentation
    - Proof of Concept then documentation?
        - Yes, learn API through implementing
    - When to start documenting
        - Weekly reports will show experimental progress
        - Formal documentation during product development
    - Expectations for the class
- Discussion of milestone vs versioning
    - Prototypes have more visibility in milestone system
- Proof of concept branch for requirements solicitation
    - Use for Validation with main stakeholder/client: Dr. Rajan
- Dalton dubbed ‘UX Czar’ to ensure UI/X design consistency.

Bi-Weekly Advisor Meeting

Members Present: All except for Kristin Clemens
Additional Participants: Dr. Rajan
Date & Location: Friday 30 of January; Atanasoff 101
Minutes:

- Review of administrative meeting for Dr. Rajan
- Discussion of possible technologies for use in project
  - Pros/Cons of XText, JDT
- Discussion of what key factors in deciding a technology platform
- Topics for research in the coming week:
  - Xtext capable of all intended features
  - Xtext java-like language existing implementations
    - Does added Xtext cause the developers to reinvent the wheel?
  - JDT entry points, what features will affect what modules
- Discussion of a possible alternative technology, Java8 Annotations
  - Would appeal to the java developer base more than a plugin
  - Alternative method of implementing the ideas/algorithms that panini implements
- Request for weekly meetings instead of bi-weekly
- Concluded

Weekly Collaboration Meeting

Date & Location: Sunday 1 of February; Atanasoff 216 & Google Hangouts
Members Present: All
Additional Participants:
Minutes:

- Cap meeting length at 45 minutes.
- Review research of various development strategies
  - Annotation method
  - Xtext
  - JDT bare metal
- Recap of rajan meeting for kristin
- Discussion of annotation method vs xtext in terms of maintainability
  - Would this cause a parallel code base for rajan's research team?
- What is our main first goal?
  - Allow developers to code panini without leaving eclipse-sphere
- Discussion of mix of two methods.

    - Using annotations to preprocess code into either java or panini code
- Source to source transform of annotated java to threadsafe java
- Research topics:
    - Transforming annotated java into panini code
    - Exploring the bounds of Xtext's implemented potential
    - JDT eclipse architecture
- 2:08 end

Individual Hourly Contributions

Trey Erenberger	5 Hours
David Johnston	8 Hours
Kristin Clemens	2.5 Hours
Dalton Mills	4 Hours

Tentative Plans for Week 3

As suggested by Dr. Rajan, the main objective for Week 3 is for each of us to continue investigating our own topics and technologies. (These were described in the summary.)