

Eclipse Ide for Education Creation Review

Presented by Wayne Beaton and Dwight Deugo, The Eclipse Foundation

Review date: February 11/2009

Comments: [news://news.eclipse.org/eclipse.ide4edu](http://news.eclipse.org/eclipse.ide4edu)

The Eclipse IDE for Education is a version of Eclipse streamlined specifically for use by post-secondary (undergraduate university and college) students. The environment provides support for programming languages that are commonly used in university courses, including Java, Scheme, and Prolog.

1 Background

Students arriving at university have many new things to contend with. They are very often living away from their parents for the first time, and are new to the post-secondary environment. They must contend with a new instruction style, new ideas, programming languages, and concepts. They must also contend with new tools and development environments. We can't help students make healthy meals for themselves, but we can help by providing a development environment that is streamlined specifically for their needs.

A common criticism of Eclipse-based IDEs is that the initial learning curve is steep. Indeed, if you take an objective look just at the process of creating a new Java class, it can be daunting. First, you must create a new Java Project. In that Java project, you can create a Java class. But it needs a package, and the package needs to be defined in a `sources` folder. The wizards present options that can seem overwhelming. What JRE will be used? What projects are referenced? Where in the file system should the project be created? Mercifully, the default values tend to be good choices and ultimately the flexibility is good for the seasoned developer. But we think we can do better for the first time user.

Ultimately, with this project, we intend to explore different paradigms for interaction between the developer and the IDE. More immediately, we are focused on providing an environment with reduced clutter that allows students to focus on their immediate requirement of getting classroom work done. In essence, the current work provides a streamlined environment featuring *stripped down* versions of wizards for creating common elements. We anticipate that students will use this environment in their first months of learning and then progress to a more complete Eclipse configuration.

2 Create and Move

With this review, we intend to both create the Eclipse IDE for Education (IDE4EDU) and move the

existing IDE4EDU component from the SOC¹ project. All existing code in the IDE4EDU component is EPL-licensed and exists in the `org.eclipse.soc.ide4edu` namespace. As part of the move, the namespace changes to `org.eclipse.ide4edu`.

The current implementation includes our initial work on the streamlined Java development tools, along with some initial work on a Scheme-language development environment. Our initial contribution is contained in CQ 2502². CQ 2903³, which aims to include the JScheme interpreter, is currently in a triage state.

Along with code, the existing component contains a releng project that is used to build the component, with scripts that leverage the Eclipse Packaging Project⁴ to produce a build. These scripts are used by an existing and mature automated build process that creates bundles that are both signed and compressed using pack200. This build process will be migrated along with the source code.

A wiki page⁵, created for the existing SOC component, provides both downloads and information about the existing work.

3 Goals/Scope

There is one overriding goal for this project: to make Eclipse adoption easier for students.

To that end, we have several immediate goals:

- To streamline installation and configuration;
- To streamline Java development; and
- To provide support for Academic programming languages.

More longterm, we envision this project providing more value to students and faculty by providing

- Assignment workflow management;
- Teacher, student, teach-assistance collaboration;
- Education material (slides and videos); and
- More!

Each of these points is described in more detail below.

We constrain the scope of this project to be concerned with the development of an Education IDE

1 <http://www.eclipse.org/soc>

2 https://dev.eclipse.org/ipzilla/show_bug.cgi?id=2502

3 https://dev.eclipse.org/ipzilla/show_bug.cgi?id=2903

4 <http://www.eclipse.org/epp>

5 http://wiki.eclipse.org/Eclipse_IDE_for_Education

based on Eclipse technology.

We further envision this project as an ideal focal point for other open source projects engaged in the development of education based offerings. We have, for example, been in contact with developers from the CoFFEE⁶ project and O'Reilly's Ellipse⁷ Project. We intend to continue to work with these, and other like organizations to ensure compatibility and interoperability between our offerings.

3.1 Streamlined Installation and Configuration

This goal initially sounds a little ambitious. In the context of this project, we view installation as a process, and so at a minimum we believe that we can do a better job of describing the installation process for students to make it easier for them to obtain and install Eclipse using traditional methods. If forces allow, we believe that it will be in the students' best interests to create at least a simple installer with few options (potentially using the EPP installer technology, p2, or some combination).

Certainly the creation of an EPP-built and hosted package is a goal of this project. The IDE for Education package will contain only the minimum set of bundles required to provide students with the functionality they need. This will include most (if not all) of the JDT, but will exclude value-added features like team-development tools, and Mylyn. It is our goal to keep the initial download size as small as possible. Additional features can be added using the update manager.

We further intend to minimize out-of-the-box configuration of the environment by ensuring that reasonable default for student values are provided for options.

3.2 Streamline Java development tools

The streamlined Java development tools leverage the Eclipse project's Java development tools⁸ (JDT) subproject. Our initial work is focused on providing a student-specific perspective, an Assignment Explorer view, and lean wizards that hide all options for which reasonable defaults can be determined.

The Assignment Explorer is similar to JDT's Package Explorer, but provides a simplified view by hiding many of the nodes that the Package Explorer exposes. The context menu is simplified, showing only the those options that we consider especially interesting for the intended audience. With our initial work, we have created a new view, but we intend to review our options and may attempt to instead leverage capabilities and other features provided by the platform.

We do not intend to make it especially difficult for students to make use of relatively-advanced technology such as code-completion and quick-assist/fix. Previous efforts, such as the GILD project (discussed later) have attempted to restrict the functionality exposed to students for pedagogical

6 <http://www.coffee-soft.org/home.aspx>

7 <http://www.oreillyschool.com/courses/ellipse.php>

8 <http://www.eclipse.org/jdt>

reasons. As part of the ongoing development of this project, we intend to review this concept and may pursue it if interest is high.

3.3 Support for Academic Programming Languages

We have started adding rudimentary support for Scheme development to the environment. Scheme is a popular language for education⁹, and as such, we believe that support for the language is crucial. Current development environment support for Scheme is effectively non-existent. Our current implementation is concerned exclusively with writing and executing Scheme code interactively. We intend to attempt to provide debugging support for Scheme as well.

We believe that Prolog is another important language for future consideration.

3.4 Assignment Workflow Management

Many universities and colleges have server-based systems that are used for making assignments available to students, collecting results, grading, and entering marks. In an ideal world, the student should never have to leave their IDE. Ideally, from within the Eclipse IDE for Education, the student can obtain their assignment (potentially including stub code that is automatically added into their workspace), and then submit it for grading directly from within the environment (the aforementioned O'Reilly Ellipse project does this). Instructors and their assistants can then, ideally, load student work directly into their own workstation, add comments, and assign a grade.

In past years, we have had student applications for the Google Summer of Code (GsoC) programme propose to build interactions with some of these systems.

3.5 Collaboration

As our world leverages concepts like telecommuting, so too do students. The ability to collaborate among peers, or even ask questions of a remotely-located teaching assistant has tremendous value. Collaboration can take many forms, including sharing of workspaces and editors, along with instant messaging.

Much of the work in this area has already been done for us (primarily by the ECF project) and so our efforts are concerned mostly with packaging the existing functionality.

We believe that the open source CoFFEE project, a groupware application for digital discussions in a live classroom situation, provides great value that should complement this project.

⁹ <http://www.schemers.com/schools.html>

4 Collaborations

There are several important collaborations with this project. Most of these have been discussed already in this document. We list them here:

The streamlined Java development tools makes clear (and obvious) use of the JDT.

ECF Collaborations; ECF committers are among the interested parties.

EPP Installer technology, packaging

Jscheme Scheme language interpreter

CoFFEE Groupware application for digital discussions in a live classroom situation.

Where possible, existing open source projects, technology, and communities will be leveraged.

5 Overlap

Per section 3.2, there is obvious overlap with the JDT project. We intend to leverage JDT functionality where possible. In some cases, we will be developing alternative wizards and user interaction techniques. Where necessary, possible, and sensible, we will work to try and drive changes into the JDT.

We intend to apply some of the lessons learned by the University of Victoria's Groupware enabled Integrated Learning and Development (GILD) project¹⁰. The GILD work builds on the existing Eclipse IDE and its associated plug-ins, as well as borrows concepts from web-based learning tools and collaborative desktop technologies to enhance learning in both co-located and distributed settings. Unfortunately, the GILD project is no longer being developed. We believe that collaboration with the groups who developed GILD is possible, and will actively seek out that collaboration.

We have also mentioned the CoFFEE and Ellipse projects as sources of potential overlap, and opportunities for collaboration.

In short, while there may be some incidental overlap, we believe that this project is more complementary to existing work than it is competitive.

6 Frameworks and Exemplary Tools

Much of the work on this project is concerned with fit and finish for student consumption. However, there will be many opportunities for framework development, APIs, and extensibility. Specifically,

¹⁰ <http://gild.cs.uvic.ca/>

- The current Scheme development feature supports multiple Scheme runtime environments via an extension point.
- We intend to develop the `Assignment Explorer` as pluggable technology to support new languages and student-specific functionality.
- Packages produced by this project will support extensibility via the standard Eclipse update mechanism.

7 Committers

7.1 Wayne Beaton, Project Co-lead, Committer

Wayne is a committer on multiple Eclipse projects. He has considerable experience with the development of Eclipse-based applications.

Wayne co-contributed the initial work, currently part of the SOC project.

7.2 Dwight Deugo, Project Co-lead, Committer

Professor Deugo also has considerable experience with Eclipse. He is a university professor who teaches undergraduate courses using Eclipse. He is working closely with students and gathering their feedback on this effort.

Dwight co-contributed the initial work, currently part of the SOC project.

7.3 Ian Kennedy, Committer

Ian is a graduate student at Carleton University researching, with direction from Professor Deugo, the use of Eclipse in an educational context. The development of an education-focused IDE is part of his research

8 Mentors

Wayne Beaton

Boris Bokowski

9 Communities

The project proposal currently shows eleven interested parties from a diverse set of organizations and institutions

Other education-related projects have contacted us with interest in participating

- O'Reilly Ellipse (distance learning)
- CoFFEE Project (student collaboration)

10 Development Plan

The first release in August 2009 will feature streamlined JDT and Scheme support. The current implementation (developed and distributed via the SOC project) is currently in use by Professor Deugo's undergraduate classes; Professor Deugo has been monitoring student use and gathering feedback. That feedback will be used to tune the release in time for the following academic year.

Future releases to coincide with school year; one month before start of the northern hemisphere's fall and winter terms.

11 Maturity Plan

We expect this project to spend a minimum of 2-3 years in incubation. Community interest and interaction is key to success, and so we expect that several iterations over multiple school terms will be required.

It has been noted that incubation is a phase and not a place, and that IDE4EDU cannot stay in the Technology top-level project forever; the project needs an ultimate home. Given the current landscape of top-level projects at Eclipse, the Tools top-level project is probably the best fit. So we intend to proceed with this project with the assumption that the Tools top-level project will be its ultimate home, but do expect to revisit this decision as the project matures and the top-level project landscape at Eclipse evolves.

12 Conclusion

We believe that the IDE4EDU project will be a valuable addition to Eclipse. By catering specifically to the post-secondary student population, we expect to reach broader numbers than current Eclipse distributions are able to reach. In so doing, we expect to expose students to the richness of Eclipse, showcasing the many different sides of Eclipse: from Java IDE, frameworks, and APIs to community

and eco-system. The net result is that more students will be graduating with much needed skills and knowledge of the Eclipse landscape.