Analyzing the Evolution of User-Visible Features: A Case Study with Eclipse

Yuejiao Wang, Daqing Hou

Clarkson University
Potsdam, NY USA
motivation

- To better understand what’s happened in large, team-based, multi-year projects.
- study evolution at level finer than enhancement, adaptation, and correction.
- focus on user-visible features.
- Lessons learned can be applied to similar products in the same domain.
Eclipse as study subject

- 8 releases in 6.5 years (1.0, 2.0, 2.1, 3.0, 3.1-3.4)

- 645 “What Is New” release note entries [2.0, 3.4].

**Toggle Comment command** (from 3.0)

The old **Source > Comment** and **Source > Uncomment** commands in the Java editor have been replaced by the **Source > Toggle Comment** (Ctrl+/) command that uncomments the currently selected source lines if all of them are commented and comments them otherwise. (You can bind keyboard shortcuts to the old commands, which are still available,
Methodology

- Define a study interest/purpose.
- Identify a model composed of categories that reflect the interest.
- Code the raw data into the categories.
- Aggregate the categorization result and report trends and patterns.
- Two models used in this study.
Analysis based on IDE functional model

- model consists of project set up, code manipulation (read/edit), compilation and running, debugging and testing (+usability).

- Eclipse experienced gradual evolution.
  - 20 out of 26 views were added in 1.0 or 2.0.
  - 10 most commonly used commands 2.0/2.1.
  - only 1/7 features added in 3.4 considered new.

- Motivated to investigate usability model.
working more efficiently with structures

• tighter integration of information and actions
  “bread crumb”

• greater automation
  “code paste”

• information pushing
  “quick assists or fixes”
broader applicability

- features are applied in more contexts e.g., undo/redo, search, and text filter
- features are generalized, e.g., from Java editor to the more general text editor. e.g., line number, navigation annotation
standard view operations

• **sorting, grouping, filtering, item navigation, navigation history, and link with editor** are standard view operations.

• These operations were not introduced to each view systematically.
**Conclusion**

- Our methodology can be generalized for feature evolution study.
- This sort of study can potentially be relevant to practice. For example, our usability results seem to be applicable to workpiece software applications in general.