Spontaneous Source Code Recommendation Tool based on Text Search Techniques

東京大学
THE UNIVERSITY OF TOKYO

Contact Information

Hidehiko Masuhara and Takuya Watanabe (University of Tokyo)

◆ email: masuhara@acm.org, sodium@graco.c.u-tokyo.ac.jp

◆ URL: http://www.graco.c.u-tokyo.ac.jp/ppp/

Goal: code recommendation tool

◆ Because code examples are useful

– to understand complicated APIs, and idioms

– to learn for novice programmers

◆ but keyword-based search engines are not so useful

– as you have to stop to think about appropriate keywords, and choose from results

```
...
panel = new JPanel();
panel.setLayout(new BorderLayout());
button1 = new JButton(■
```

```
...
super(new BorderLayout());
button = new JButton("Click Me");
button.setPreferredSize(new Dimension(200, 80));
add(button, BorderLayout.CENTER);
button.addActionListener(this);
...
```

suggests next actions

your code   example code

Selene: IDE integrated association-based text search

➢ Automatically searches **similar** programs

◆ uses an association-based text search engine GETA [Takano et al.]

➢ From a large code repository

◆ containing a collection of open source programs

➢ Shows relevant lines

recommended examples

editing code
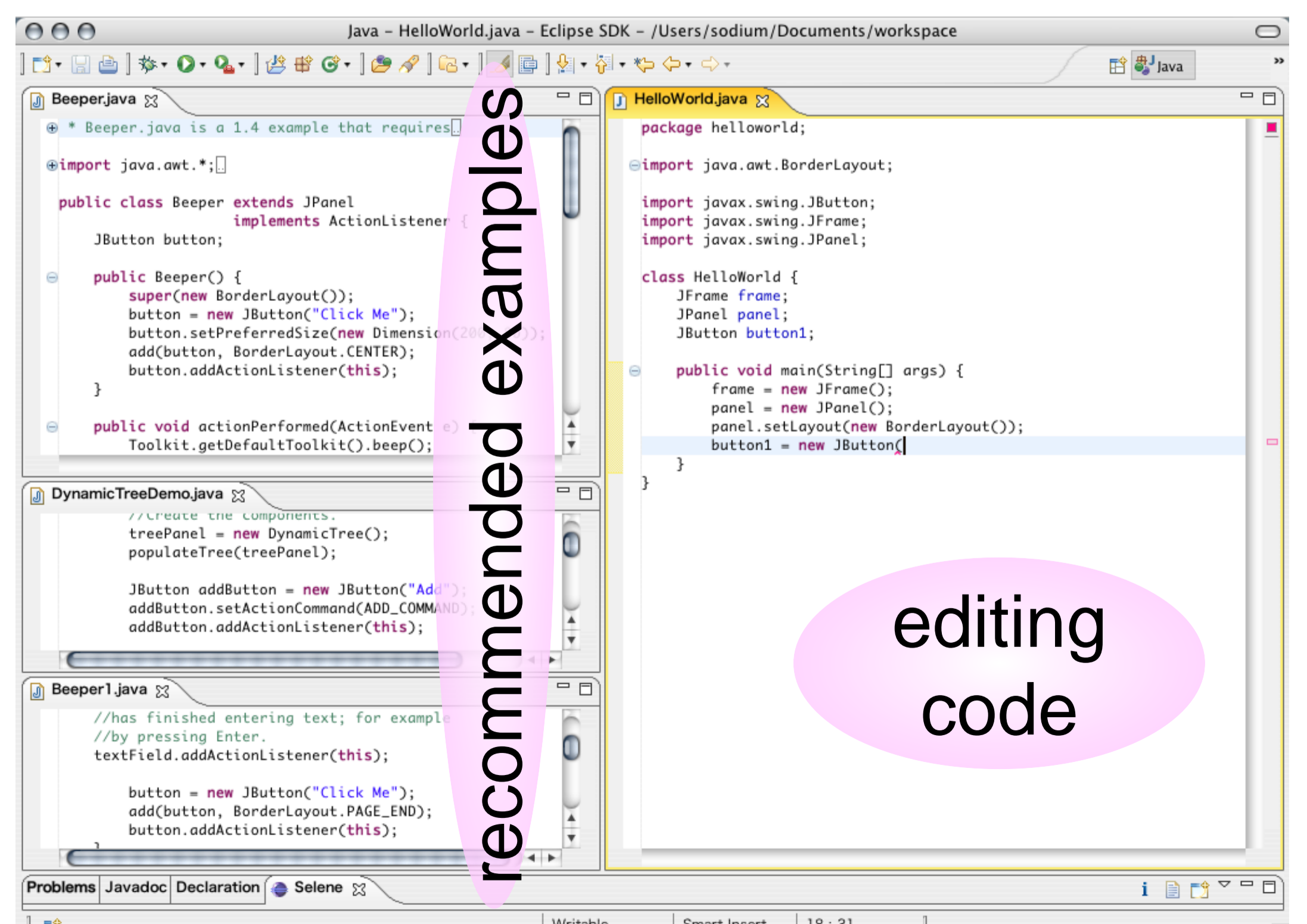
screenshot of Selene prototype

Advantages

- vs. keyword-based search engines
  - ◆ **Spontaneous**: i.e., search while you type
    by using the entire program text as a query
- vs. semantic-based recommendation tools
  - ◆ **Scalable**: search from a large code repository
    thanks to GETA, which uses in-core indices for association search
  - ◆ **Easier multi-language support**: no need for time-consuming program analysis
    (though you can apply after filtering)

Implementation
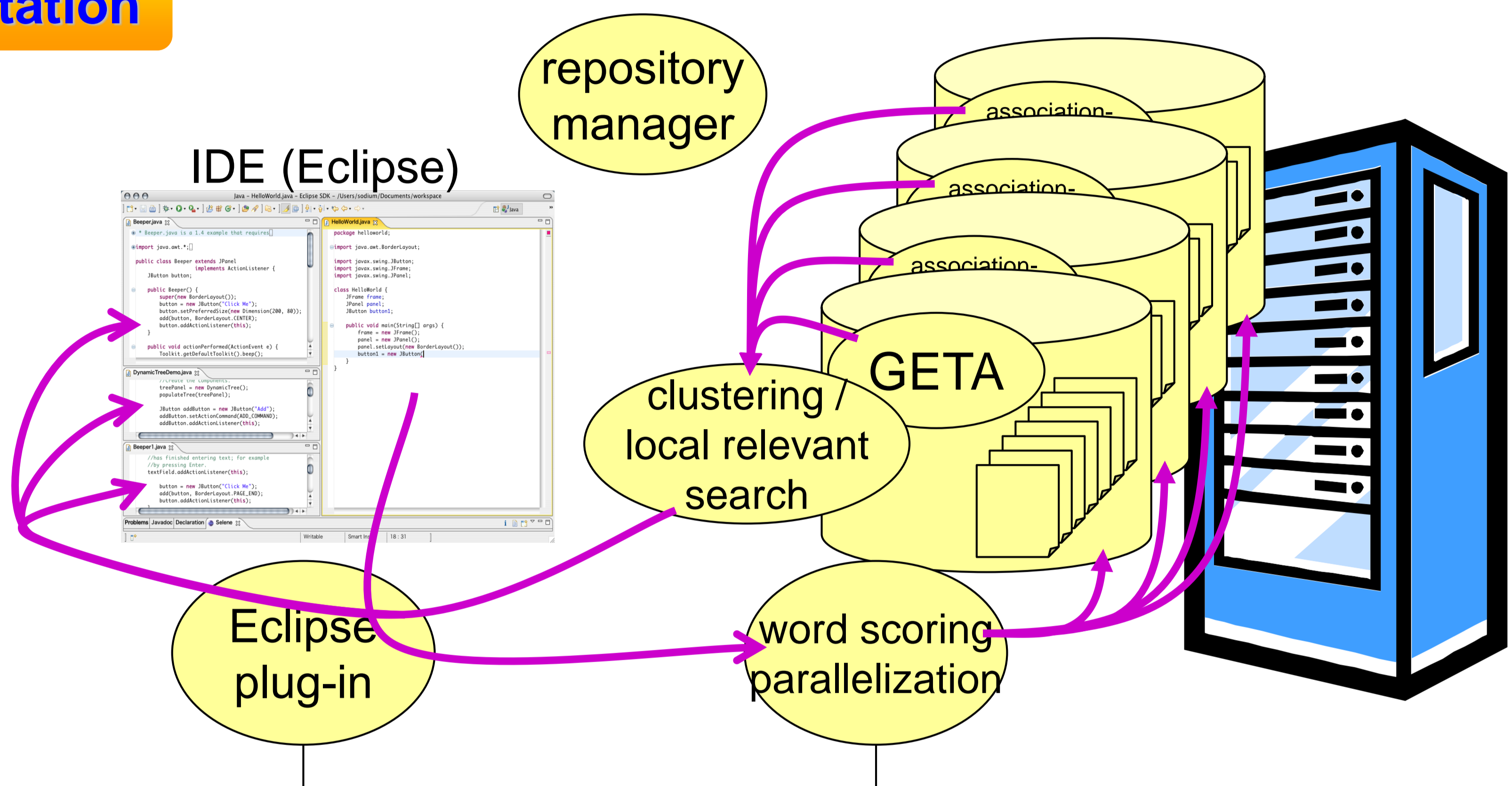
Research issues

- Improving response time
  - ◆ (current version) 1 sec. / 50,000 files
  - ◆ parallelization
- Improving user interface
  - ◆ When to start new search / display results
    - – won't update examples when being watched
  - ◆ Need a good user model
- Improving search quality **by applying NLP techniques**
  - ◆ Weight function
    - – (current version) simple tf-idf + distance to cursor line
  - ◆ Result clustering
  - ◆ Incorporating semantic information
    - – types, distinction between user-defined or library methods, etc.