Active Code Completion: Brief Description

Cyrus Omar, YoungSeok Yoon, Thomas D. LaToza, Brad A. Myers
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA
{comar,youngseok,tlatoza,bam}@cs.cmu.edu
https://github.com/cyrus-/graphite

I. Presentation Method

We will present our active code completion technique in the context of object instantiation by introducing our current prototype, called Graphite. A 10 minute long video tutorial is going to be played (or conducted live) while a couple of laptops are provided so that the conference attendees can test Graphite on the Eclipse IDE. The details of the video are described in the following section.

II. Video Tutorial

The video tutorial demonstrates the basics of how to use Graphite tool in the Eclipse code editor, two example palettes, two different ways of associating a palette to a class, and a quick tutorial of writing a custom palette for a specific class. As being a tutorial, these elements are not strictly separated, but they are naturally mixed in order to make the video more interesting and understandable to people.

A. Basic usage of Graphite

The video begins with the basics of when and how to invoke Graphite palette in the Eclipse code editor. It is shown that how Graphite is incorporated in the Eclipse content assist system.

B. Palette Example 1. Color

The first example palette is the one for `Color` class as its benefits are quite intuitive. The palette looks similar to those of graphical editors. Users can specify the desired color in many different ways. For example, they can choose one of the colors predefined in the `Color` class from the palette, or they can input a '#' character followed by six hexadecimal digits as they normally do in the HTML and CSS files. Whenever they specify a color, they can directly see the resulting color from the color preview box.

C. Palette Example 2. Regular Expression

The regular expression palette shows even more benefits of the active code completion technique. Human mistakes can be significantly reduced by using the palette, better programming practices (e.g., writing test cases first) are encouraged to use, and some of the API usability problems can be solved.

D. Writing a new palette for a class

One of the most important points of Graphite is that anyone can write new palettes. The video will demonstrate how to build a very simple palette from scratch which inserts a simple string (e.g., "hello world") into the cursor position when the user clicks a button. Moreover, it is shown how to use a web browser to test and debug a palette as they do when writing web applications. This quick demonstration is short enough to follow, and convincing that writing a custom palette is not difficult.

E. External invocation model

The aforementioned two example palettes are associated with their corresponding classes via an external configuration file. They can be edited from the preferences page of Eclipse IDE, and the video will show how to do so.

F. Annotation-based invocation model

When showing how to write a new palette, it is also shown that how we associate a palette to a class using Java annotation.

III. Video

The video is available online now[1]. It is longer than the 4 minutes requested, so reviewers may stop after the Color palette has been demonstrated.

[1] http://www.youtube.com/watch?v=FfCoXQEHGAI