Variations on Active Aspect
Katherine Irvine
Summer 2005

Over the last four months, I have been working with Active Aspect, an Eclipse Plug-in developed by Wesley Coelho. Active Aspect is a tool for Java AspectJ developers that visually maps relationships to and from Aspects. Aspects, which encode cross-cutting elements of code, are not adequately represented by traditional views, such as tree- and hierarchy views. Active Aspect presents the code in a flowchart-type view, similar to those used by many programmers when they develop the code. Furthermore, Active Aspect attempts to only present the most important elements, allowing the user to expand the model as needed. Several types of relationships are only visible when requested; my job was to modify the program to provide several ways of displaying these relationships, and then develop a study to determine which of these methods was most effective or helpful to the user.

One of the main things about Active Aspect is that it is flexible and interactive. Method calls and field references can be incrementally added to the diagram by the developer. Advises, introduces, and uses-pointcut relationships are shown "On Demand" – they appear transiently as the developer interacts with the model. The three versions I worked on are distinguished by how these Show on Demand relationships are shown.

The first two variants work similarly, but display differently. The user hovers over an element to show all the Show on Demand relationships associated with that element. Clicking on the element will cause those relationships to remain visible, even when no longer hovering over them. The first variant, Arrows, shows the relationships as dashed, color coded arrows, while the second, Highlight, changes the background color of the elements. The source element is highlighted in brighter yellow, while the target elements are highlighted in the same color scheme as the arrows. The third variant, toggle, works slightly differently. The relationships are displayed as arrows, just as in Arrows mode, and hovering produces the same effect. However, instead of clicking on each element to make the relationships remain showing, the user clicks on buttons in the toolbar, which show and hide all the relationships of a particular type.

I added the tool palette on the left side of the screen and six buttons on the top bar. The tools palette was necessary so that all the functionality could be accomplished with either right- or left-clicking, rather than ctrl-clicking as Coelho's original program had. The buttons I added were in two groups of three. The first group was used in Toggle mode, the three Show on Demand relationship types each had an associated button, and clicking that button would show or hide all the relationships of that type. The other three buttons were to switch between the three versions of the program. I added an Aspect which showed or hid the Show on Demand tool and enabled or disabled the Toggle buttons, as appropriate. It also indicated to the Show on Demand tool whether it was to show arrows or highlighting.

Finally, I had to create a study to determine which of the three versions was the best. I decided to use a timing variable to determine the usefulness or efficiency of each version. The study participants would be given a series of questions about an AspectJ Aspect, and would be timed on how long it took them to answer each question using an AA model. One important consideration was that the study would be conducted remotely, as there are few people who are familiar with AspectJ, and for the most part they are spread across various universities. It was therefore decided that the study would be conducted on one of our computers, but that the study participants would log in remotely with Remote Desktop. I created a custom view for Eclipse which would present the questions and multiple choice answers in sequence. I added an Aspect which read the questions and answers from a file, timed the user, and output the question, answer, time blocks to another file. The program does not determine whether the answer given is correct, only indicates which answer was chosen. The person conducting the study will have to determine which questions each participant answered correctly and compute all relevant statistics. Several days before the study begins, each participant would also be given a how-to document and short video demonstration (tailored to the version they will be using).

Because this is a new area of research, there are not many previous publications of related research. My work was based in part on the concepts in the following paper:

Wesley Coelho and Gail Murphy, *Presenting Crosscutting Structure with Active Models*. Forthcoming.