## JQuery with Multiple Views

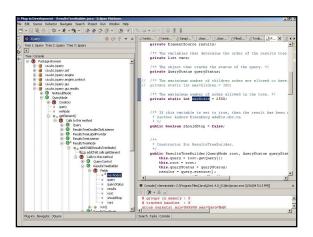
Adding Context to Navigation and Querying of Code

### JQuery in terms of AOP

- Scattering and tangling of concern specific code
- How can JQuery help?
  - Navigation path reflects the users thought process
  - Form a decomposition of the system of code by building a browser to fit the task.

### Current Design of JQuery

- Queries can be performed on any node at any time
  - Match and then Order
  - Top Level queries (choose a working set)
  - Intermediate queries ("who extends this class")
- Navigation organized as a tree hierarchy
  - Low use of screen space... well, relatively low
  - The current query is made in terms of the parent node
  - It is possible to relate two leaf nodes by where their paths diverged in the tree (adding some context)
- Multiple tabs allows for multiple browsers
- How does this relate to scattering and tangling?



# Potential Problems with this Visualization

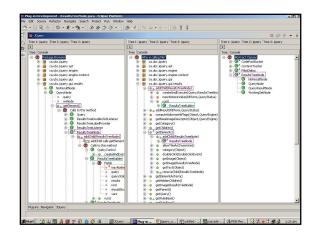
- Nodes relate to each other in a more complex way than the tree hierarchy is able to reflect.
- Hidden relationships: Method 'a' may be reside in a class 'A' from three queries back.
- A confused programmer means a confusing traversal.
  - Nonexistent relationships.
  - Circular (recursive) navigation.
  - Taking a longer route than necessary.

### Proposal

- Possibilities:
  - Provide some automatic traversal analysis/pruning.
    - Automatically Inferring Concern Code from Program Investigation Activities (Robillard, Murphy. 2003)
  - Provide a new Visualization
    - How can we provide context for the multi-dimensional traversals?
- Proposal:
  - Linked Views: Browsers created by the user are linked by color.
    - Reminiscent of Snap-together visualizations. (North, Shneiderman. 2000)
    - · Helps in forming interesting decompositions

## Design Details

- By default, the last five nodes in the dominant view are highlighted. Otherwise, last five active (clicked on) nodes.
- At this time, no support for "guaranteed visibility" (Munzner, 1997)
  - If a node is off the screen, the user must find it. (automatically expand parent nodes to reveal matches)
- A great deal of screen space will be used.
  - Code view window closed during navigation.
  - When leaf node is found, minimize 'multiview' window to replace context views with code view.



#### Deadlines

- Design nailed down/Learn Eclipse plug-in tools (week 1)
- View implementation:
  - MultiView button on toolbar when clicked the view expands to fill window.
  - Currently: implementing concurrent browsing capabilities and corresponding node identification. (Weeks 2,3...)
- Future Steps:
  - Link the views,
  - Revealing active nodes (automatic expansion)
  - Color-coding of active nodes Look into how Eclipse can help me with this. Perhaps an icon next to corresponding nodes will suffice.

questions? comments?