Software Difference Analyzation Tool

Rolf Biehn
Why Abstract Syntax Trees?

- Use an Abstract Syntax Tree (AST) and structurally compare code
  - Reduce noise (such as renamed variables, method declaration order, etc.)
  - Cross-language comparisons become possible
  - Better visualizations possible
  - Improved Navigation
AST & Analysis

- Created my own (partial) AST model
- AST model serializer/deserializer JAVAXML
- Redefined JAVAXML format slightly (MethodName, ClassName, VariableName attributes → name)
- Simple Comparison + manual intervention
AST & Analysis (Problems)

- JAVA XML scalability issues (5K Lines of code -> 800K of memory)
- Limited number of AST nodes supported
- Need to redo my AST object model
Method View
<methodCall name="mName">
    <meta name="diff" />
    <arguments>
        <var-ref name="var1" />
    </arguments>
</methodCall>

- In JAVA this would be "mName(var1);"
Adapt the AST Model

```xml
<methodCall name=" mName">
  <meta name="diff" />
  <arguments>
    <var-ref name=" var1" />
  </arguments>
</methodCall>
```

- We don’t want to Visualize the AST Model.
Flatten the Tree

```xml
<methodCall name="mName">
    <meta name="diff" />
    <arguments>
        <var-ref name="var1" />
    </arguments>
</methodCall>

- Therefore, flatten the list.

{ <DIFF>, "mName", </DIFF>, "(" , "var1" , ")" }
Mini-Map Demo

- Markers (Red, Blue, Purple)
- Collapsible (purple – updates both, efficient)
- Left click moves to spot
- ViewMode (anchor, length, current size)
- Page up / down buttons
- %Width
- Locked Mode
- Resize Mode
Mini-Map

- The Good
  - Can represent a large number of lines
  - Interaction is easy once learned

- The Bad
  - Not intuitive?
  - Resize mode needs help

- The Future
  - Further Optimizations possible
  - User Studies
  - Clean up Interfaces for library re-use
Detailed View Demo
Final Conclusions

- Mini-Map can be used anywhere with a pixel map + line concept (i.e. SeeSoft)
- Not really leveraging AST comparison at this point
- Piece of a puzzle – long road ahead including:
  - More support of JAVA & C# AST nodes
  - Better comparison algorithms
  - Able to compare and visualize classes, packages, etc..
  - More interaction {filters, go to, multiple views, etc..}
Questions?