

Evaluation of Component Adaptation Techniques

by George T. Heineman and Helgo M. Ohlenbusch

<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.54.5952>

In this paper the authors discuss different ways software can be adapted for use in a third-party environment. The authors make a distinction between evolution and adaptation stating that "evolution occurs when a software component is modified by the original component design team or by programmers hired to maintain and extend the component ... In contrast, adaptation occurs when an application builder acquires a third-party component and creates a new component to use within the target application." ISVis has both of these scenarios - the instrumentation tools being the third-party components that have been used in the program. Our current focus is on the evolution of the application to bring it up to date with today's standards. In that regard the rest of the paper applies to the future work that will need to be done with ISVis to adapt updated third-party components to get the instrumentation working.

The paper lists several "requirements" that are needed for the adaptation of a component - they are really more like guidelines and are not all necessary (or possible) depending on the adaptation being performed. The paper then discusses five techniques that can be used in the adaptation of software:

- Active interfaces - a component that allows for the execution of user-defined callback methods; in contrast to the other techniques discussed this one relies on the component being adapted to have this functionality
- Binary Component Adaptation (BCA) - a technique that applies adaptations to component binaries (no source code access)
- Inheritance - built into OO languages
- Wrapping - this technique wraps the component being adapted providing an interface that hides the component's usage
- In-place - occurs when changes are made directly to the source code of a component (think open source)

There is a comparison chart on page 7 showing how each of these techniques match up against the aforementioned requirements. The authors then describe the case study they did in which they performed the same adaptation on a sample application using each of the five techniques and give their conclusions on the strong/weak points of each technique.

Reference:

```
@INPROCEEDINGS{
  Heineman:Adaptation,
  author = {George T. Heineman and Helgo M. Ohlenbusch},
  title = {An Evaluation of Component Adaptation Techniques},
  booktitle = {2nd ICSE Workshop on Component-Based Software Engineering},
  year = {1999}
}
```