Overcoming Decomposition in Legacy Code

The primary concern of the research described in this paper is to address the problems associated with identifying hidden concerns in legacy code. A hidden concern (HC) is any issue that has been grafted into existing code to extend it's functionality in some way that is different from the piece of code it is being added to. This decreases the code quality (by reducing it's coherence) because the code is now handling more than one specific issue. Hidden concerns are “'pressed' into the primary decomposition” of the existing system. More often than not, an HC suffers from two problems that make them hard to track down: they are usually scattered throughout the project and tangled with other code. Consequently the primary concern which the paper recognizes as being non-trivial, is how to identify and extract the code related to a hidden concern.

To do this, the paper describes a process that combines text-based and type-based mining. The paper illustrates why doing them separately doesn't prove to be as effective as doing them together and then introduces the Aspect Mining Tool, developed by the authors. AMT is an extensible framework which uses a modified version of the AspectJ compiler and provides type-based and text-based analysis techniques. As a test case, the authors analyzed Apache Tomcat, specifically looking for ways to improve the way session states were coded.

The techniques described in the paper should prove to be very useful in refactoring ISVis. The current work being planned for ISVis is to factor out all the code related to the display to provide an interface that can be implemented by more modern versions of the