Feature Oriented Refactoring of Legacy Applications

Feature oriented refactoring is the process of decomposing a program into features, thus recovering a feature based design and giving it an important form of extensibility. A characteristic of feature-oriented refactoring (FOR) that makes it challenging is that a feature cannot always be implemented by a single module. To try and deal with this problem the paper describes a methodology for expressing programs and features as mathematical equations. Engaging in this practice gives the practitioner a reliable way of predicting the effects of adding or removing features from the base program because the derived equations communicates properties that decompositions and their underlying base modules and derivatives must have.

These properties are automatically checked by the tools the authors developed to aid in the FOR process. The authors developed an Eclipse plugin to be used with legacy Java applications. The plugin guides the user through a five-step process which defines the base modules of the program, optional modules, and then reconstitutes the program for the user.

Pertaining to ISVis, the concept presented in this paper would seem useful, but since the tool the authors developed only works for Java programs, the work involved in implementing their ideas by hand would probably not be worth the effort. Additionally, this method requires a substantial knowledge of the program being examined and that would make using it on ISVis difficult since ISVis suffers from the "embedded knowledge" symptom of an aged system.

Note: A prerequisite to fully understand this paper is a familiarity with Aspect oriented programming - specifically, AspectJ is made reference to quite a bit.