**Reverse Engineering Goal Models from Legacy Code**

A goal model is a graph structure representing stakeholder goals and their inter-dependencies. This paper describes a methodology for obtaining a goal model from legacy code. The process is defined by the following steps:

1. Refactor the source code into a more abstract form - they do this by recursively applying the Extract Method refactoring operation as described in Fowler's "Refactoring: Improve the design of existing code". They determine segments of code for extraction based on comments in the code (and the authors do recognize the weakness in relying on that strategy).
2. Extracting states - each method is treated as a Hammock graph (single entry, single exit) and is used as such to construct a statechart with the goal of obtaining an abstract view of system behavior. The initial statechart obtained from the refactored code is recursively refactored until a more abstract statechart is obtained. It is not specified when exactly that occurs so it is assumed to be at the practitioner's discretion.
3. The high-level statechart is now converted to a FORTRAN program - each state with more than one entry is is associated with a label and each transition for an additional exit is associated with a GOTO statement. This program is then run through the FPT compiler which implements a technique that eliminates GOTO's through hammock graph construction. This process essentially creates a structured code version of the statechart being operated on.
4. From here, obtaining a goal model is simply a matter of annotating the abstract syntax tree of the structured code produced by step 3.

At the time of this summary, the main focus of the ISVis reengineering process is on abstracting the display code so the process described in this paper is not necessary since the code to be refactored has already been identified. However, these methods could potentially be useful in future refactoring efforts on ISVis as they would help to categorize the code to help identify all the candidates for refactoring.