

ARTISAn

Objective:

In situations in which developers are not familiar with a system or its documentation is inadequate, the system's source code becomes the only reliable source of information. Unfortunately, source code has much more detail than is needed to understand the system, and it disperses or obscures high-level constructs that would ease the system's understanding. Automated tools can aid system understanding by identifying recurring program features, classifying the system modules based on their purpose and usage patterns, and analyzing dependencies across the modules. Various representations can be used to describe successive levels of system's abstractions. Incited by Perry and Wolf's observation that the key architectural elements of a software system are (1) processing, (2) data, and (3) connecting, we have developed ARTISAn, a tool-supported, pluggable framework intended to aid program understanding and, ultimately, architectural recovery.

Description:

We have developed a prototype of ARTISAn targeted at Java systems. The tool is integrated with IBM Rational Rose®. The demonstration will consist in using the ARTISAn tool to analyze a software system. First, we will show a high-level functionality view of a system by categorizing system elements into the three aforementioned categories, based on a set of architecturally relevant clues and rules. We will then show how the system elements are organized into regions that are identified as a result of relationships among elements. Specifically, we will illustrate how these two views complement each other. Finally, we will show how ARTISAn could be used in identifying possible inconsistencies in the obtain views, as a result of applying various "what if" scenarios.

Contact:

Vladimir Jakobac
USC, SAL 327
jakobac@usc.edu

