Rigi

A Visual Software Understanding Tool

Rigi is an interactive, visual tool designed to help you better understand and redocument your software. Rigi includes parsers to read the source code of the subject software and produce a graph of extracted artifacts such as procedures, variables, calls, and data accesses. To manage the complexity of the graph, an editor allows you to automatically or manually collapse related artifacts into subsystems. These subsystems typically represent concepts such as abstract data types or personnel assignments. The created hierarchy can be navigated, analyzed, and presented using various automatic or user-guided graphical layouts.

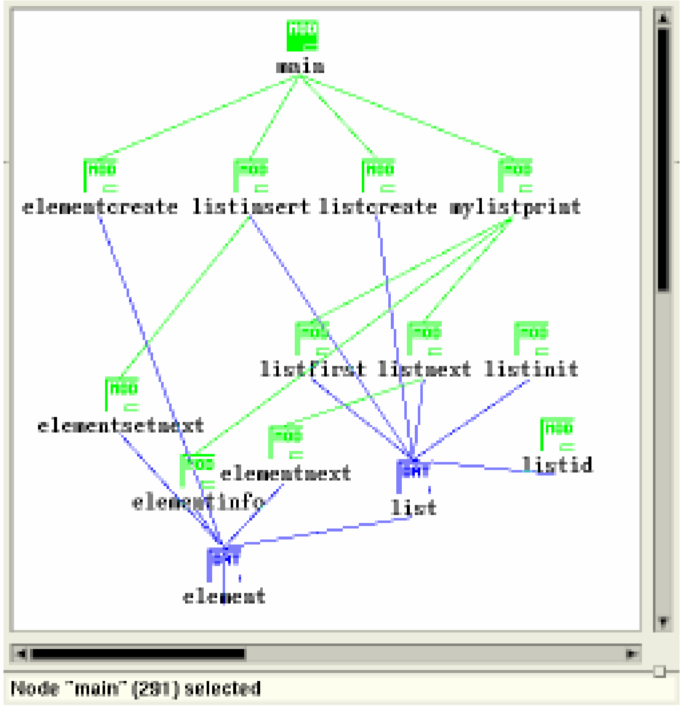
The discovered structural information is useful for making informed development and management decisions. The information serves as documentation that is up-to-date and accurate because it is derived from the actual source code. Thus, Rigi helps to understand legacy software systems where the existing documentation may be missing or lacking. Rigi aids reengineering tasks that need to discover design information in existing software.

Features

- easy-to-use visual interface
- parsers for C, C++, and COBOL
- selection, filtering, and editing operations
- dependency and change impact reports
- standard, overview, and projection perspectives
- metrics for cohesion and coupling
- views to capture interesting perspectives
- scripting language and command library
- adaptable to different languages and purposes
- customizable user interface
- simple file format to represent graphs

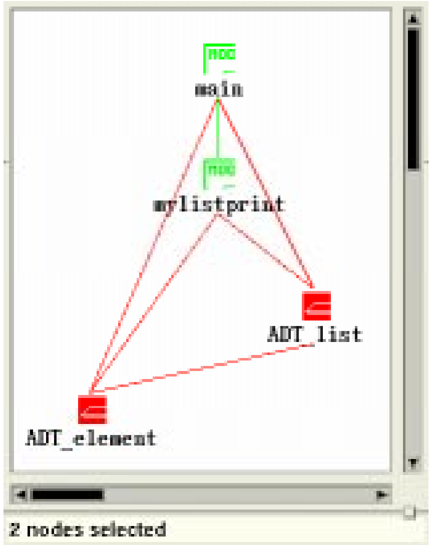
How it works

You run a parser that extracts artifacts from the subject software. A domain model specifies the entity types and relationships of interest. The input stream is fed into a graph editor to visualize the initial graph.



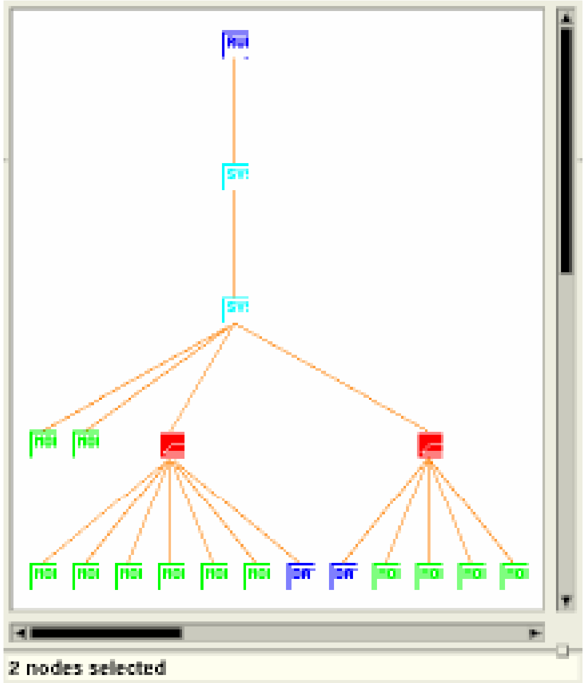
Node "main" (291) selected

You select, filter, layout, and edit the graph to identify pertinent subsystems (e.g., abstract data types).



2 nodes selected

These operations produce a simpler, hierarchical graph. You can save different perspectives of this hierarchy in a reloadable view.



2 nodes selected

Standard graphs such as the call graph, which shows procedures calling procedures, are quick to produce. You can easily report the dependencies of a subsystem on its neighboring artifacts. You can write scripts in Tcl / Tk using a Rigi command library to define and automate common operations on your graphs. You can even design your own menus and widgets to customize your working environment and visualization needs.

Rigi is flexible and can help you to understand the structure of software, technical documentation, and hypertext. Rigi is capable of presenting the structure of both small and large information spaces.

Rigi has been used successfully to view and navigate a number of real software systems, including IBM’s multi-million line SQL / DS, NASA’s CLIPS expert system shell, and a commercially-sold Doctor's Practice Management System. Rigi has been used to examine the dependencies among part assemblies in manufacturing processes.

Platforms

Rigi currently runs on Sun SPARCstations (SunOS), IBM RISC System 6000 (AIX) workstations, and PC-compatible (Windows 95, Windows NT) machines.

More information

Information about Rigi is available on the Web:

`http://tara.uvic.ca`

There are numerous conference and journal papers, also online, that describe our approach.

Contact:

Hausi A. Müller
Department of Computer Science
University of Victoria
P.O. Box 3055
Victoria, BC
Canada
V8W 3P6

Call:	(604) 721-7630
Fax:	(604) 721-7292
Internet:	`hausi@csr.uvic.ca`

Thanks

This work was supported in part by the British Columbia Advanced Systems Institute, the IBM Software Solutions Toronto Laboratory Centre for Advanced Studies, the IRIS Federal Centres of Excellence, the Natural Sciences and Engineering Research Council of Canada, the Science Council of B.C., and the University of Victoria.