

RBCrawler — a Visual Navigation System for Smalltalk's Refactoring Browser

[Adrian Kuhn](#)

Software Composition Group
University of Bern, Switzerland

Abstract

RBCRAWLER is a software visualization tool for the Refactory Browser, implement in Cincom Smalltalk. It extends both the editor pane and the navigation pane with software visualization views. RBCRAWLER features static, dynamic, and lexical analysis. Currently in implementation are: Lanza's *System Complexity View* and *Class Blueprint*, customizable *Polymetric Views* and *Distribution Maps*, as well as *Traceplots* of dynamic execution traces. *Semantic Clustering*, *polysemy-aware search* and *Wordclouds* are planned as well, but might not be ready in time for presentation at ESUG 2007. RBCRAWLER improves software understanding and reduces the time for software navigation. An empirical study based on the present implementation is planned for this fall's Smalltalk lecture at University of Bern.

Software Analysis for the Developer

For 10 years now Moose-based analysis has been around, but it never reached developer acceptance because it was trapped away in a analysis environment rather than being featured in their development environment (IDE).

Now, RBCRAWLER eventually bridges the gap between software analysis and development tools. RBCRAWLER augments the IDE with the best and latest from software visualization and reverse engineering. Just as today's automobiles feature a satellite navigation system to support car drivers, RBCRAWLER features a visual navigation system to support developers in their daily tasks.

Despite common belief, developers are spending more time for navigating and trying to understand source code, rather than actually developing new functionality. Software visualizations has the power to improve software understanding and reduce the time for software navigation. In recent years, many visualization techniques have been proposed that ease understanding and navigation.

RBCRAWLER integrates a selection of the best of these techniques in Smalltalk's Refactoring Browser. The Refactoring Browser has already been ahead of its time back in the nineties, when it was the first IDE to feature automatic code refactorings. Today, automatic code refactorings are an industry standard. Now again, Smalltalk is taking the lead in software development as RBCRAWLER enhances the Refactoring Browser with a visual navigation system.

RBCRAWLER extends both the editor pane and the navigation pane with the best and latest in software visualization:

System Complexity View uses polymetric figures to show structure and metrics of an entire system at one glance. Developed at the SCG by Michele Lanza, this view is undoubtedly among the most popular software visualization ever[5], and was maybe *the* path-leading breakthrough of software visualization in the early 2000s.

Class Blueprint uses polymetric figures to shows internal structure and metrics of single components. Another classic by Lanza, this view is the natural complement to the System Complexity View, as it shows the internals of object-oriented software[2].

Custom Polymetric Views are for all those of you that wanted Code-Crawler[6] there and back again! Based on Lanza's polymetric figures, this highly customizable views allow developers to build any visualization they might ever need in order to understand software.

Distribution Map shows how properties and metrics distribute over the system's structure. Based on on-going research at the SCG, this view brings a first spark of spatiality into software visualization[1]: our vision is that software shall eventually become as easy to navigate as real space using a geographical map.

Dynamic Traceplot View shows the executional behavior of selected methods, depicting up to millions of methods calls on one screen. Also on-going research at the SCG, the idea here is to apply signal processing techniques to analyze and visualize execution traces[4].

Semantic Clustering and **Polysemy-Aware Search** complement software analysis with another dimension: search engine technology, *aka* information retrieval (IR). The application of IR is a recent yet promising subfield of software analysis. All too often neglected, the vocabulary of source code bears the meaning of a software's domain and is thus indispensable for understanding any software system. Semantic Clustering is a recent development of SCG, it groups software components based on their domain vocabulary rather than structural dependencies[3].

Word Cloud View shows important terms in the vocabulary of selected software components. Popularized by the recent Web 2.0 as tag clouds, weighted lists are most useful to detect important vocabulary of – not only text documents and web-sites, but also – software components.

Currently, RBCRAWLER is at the stage of a prototype, details given in this document may be subject to change. However, a first stable release is planned for the *ESUG Technology Award 2007* in Lugano, featuring at least the first five views given above. The IR-based views are also planned, but might not be ready for presentation this sommer.

Please refer to RBCrawler’s website for more information

- <http://smallwiki.unibe.ch/RBCrawler>

References

- [1] Stéphane Ducasse, Tudor Gîrba, and Adrian Kuhn. Distribution map. In *Proceedings of 22nd IEEE International Conference on Software Maintenance (ICSM '06)*, pages 203–212, Los Alamitos CA, 2006. IEEE Computer Society.
- [2] Stéphane Ducasse and Michele Lanza. The class blueprint: Visually supporting the understanding of classes. *Transactions on Software Engineering (TSE)*, 31(1):75–90, January 2005.
- [3] Adrian Kuhn, Stéphane Ducasse, and Tudor Gîrba. Semantic clustering: Identifying topics in source code. *Information and Software Technology*, 49(3):230–243, March 2007.
- [4] Adrian Kuhn and Orla Greevy. Exploiting the analogy between traces and signal processing. In *Proceedings IEEE International Conference on Software Maintenance (ICSM 2006)*, Los Alamitos CA, September 2006. IEEE Computer Society Press.
- [5] Michele Lanza and Stéphane Ducasse. Polymetric views—a lightweight visual approach to reverse engineering. *Transactions on Software Engineering (TSE)*, 29(9):782–795, September 2003.
- [6] Michele Lanza and Stéphane Ducasse. Codecrawler—an extensible and language independent 2d and 3d software visualization tool. In *Tools for Software Maintenance and Reengineering*, RCOST / Software Technology Series, pages 74–94. Franco Angeli, Milano, 2005.