

Modularizing Your Software

with less time, cost, and pain

SOFTWARE-RELIANT SYSTEMS must evolve over time in response to new requirements and technologies. Often the structure of legacy software becomes too complicated to allow improvements to be made quickly and cost effectively. The SEI developed a tool that recommends how to restructure existing software to achieve greater modularity, in turn supporting strategic reuse, service-based deployment, and improved productivity.

If your team has tried to isolate a software capability from a large system—perhaps to migrate to the cloud, for reuse in another system or platform, or to decompose a monolith into more modular code—chances are your team faced significant challenges. Untangling a capability from its knot of dependencies can take weeks to months of effort, sometimes rendering the task impractical. This is a common predicament. Large-scale refactoring is typically a labor-intensive process in which developers must

- figure out where to make changes.
- identify which refactorings to use.
- implement refactoring by rewriting code.

We surveyed 107 industry practitioners to understand the state of large-scale refactoring practice and found that the mean effort required for large-scale refactoring was 1,500 staff days.

Our Solution

To remedy these issues, we developed a tool that aims to reduce the time required to isolate software to as little as 20% of the time required by manual approaches.

Using objectives that you identify, it analyzes software structure using an AI-based approach and generates recommendations to enable software isolation to solve project-specific problems. Our refactoring assistant automatically generates a collection of solutions that represent tradeoffs among competing objectives. The refactoring assistant recommends specific steps to modularize your code to achieve benefits such as the following:

- the ability to reuse well-tested pieces of code resulting in reduced software bugs, increased software quality, and improved time to market
- fewer dependencies among code elements, allowing teams to work and deploy more independently. This typically results in lower maintenance costs and introduction of fewer bugs during development
- improved scalability, and hence customer experience, by isolating highly used instances into stand-alone services that support replication
- increased ability to move to new platforms quickly, whether it is a cloud platform or another platform that provides improved performance or lower cost of total ownership

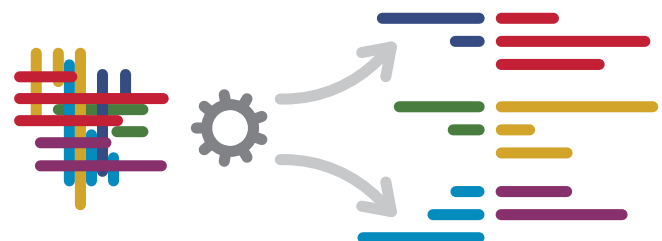


Figure 1: Our refactoring assistant recommends specific steps to untangle software.

How It Works

Our approach integrates automated code analysis and multi-objective optimization techniques to quickly analyze and solve software isolation problems. The steps are as follows:

1. You specify which software isolation problem to solve.
2. Our multi-objective genetic algorithm identifies several alternate recommendations for code refactoring that solve the specified problem. These recommendations represent different tradeoffs among objectives you select.
3. You then review and choose which recommendations best match your context.

When the above steps are complete, you can press a button and generate a file with directions for how to refactor your code.

Technical Requirements

Our refactoring assistant supports codebases in C#, C++, and Java. It has been used to analyze software up to 1.2M sources lines of code. Source code is required to use the refactoring assistant.

We Want to See if We Can Solve Your Problems

The SEI is looking for suitable code bases for analysis. If you have a code base that is problematic and are willing to share your code, we will use the refactoring assistant to generate software isolation recommendations.

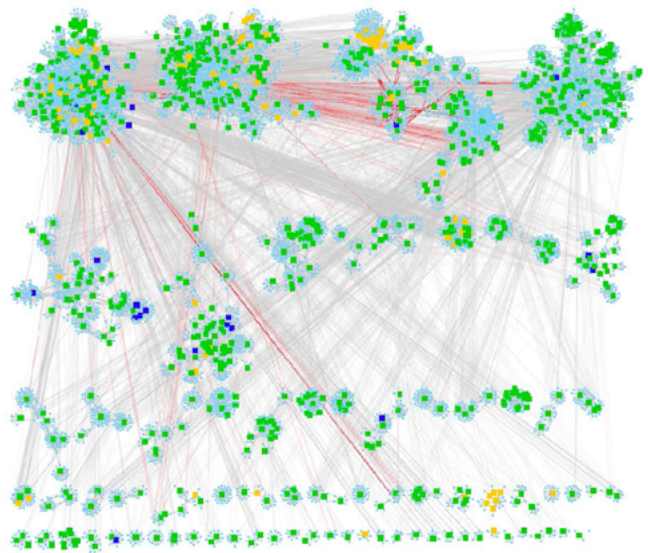
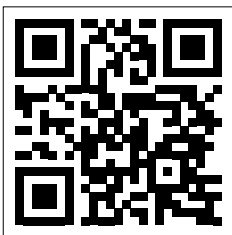


Figure 2: Entangled software contains many dependencies among code elements, but only some of those dependencies are relevant to each software isolation problem. Once those dependencies are identified, our optimization algorithm searches for solutions to refactor those problematic dependencies.



For more information about our work, follow the link or scan the QR code.

sei.cmu.edu/go/knot

About the SEI

Always focused on the future, the Software Engineering Institute (SEI) advances software as a strategic advantage for national security. We lead research and direct transition of software engineering, cybersecurity, and artificial intelligence technologies at the intersection of academia, industry, and government. We serve the nation as a federally funded research and development center (FFRDC) sponsored by the U.S. Department of Defense (DoD) and are based at Carnegie Mellon University, a global research university annually rated among the best for its programs in computer science and engineering

Contact Us

CARNEGIE MELLON UNIVERSITY
SOFTWARE ENGINEERING INSTITUTE
4500 FIFTH AVENUE; PITTSBURGH, PA 15213-2612

sei.cmu.edu
412.268.5800 | 888.201.4479
info@sei.cmu.edu