

Motivation

Software development is a complex process involving many developers working together to deliver a product. Researchers and practitioners have long recognized this collaborative aspect of software production, which emphasizes the importance of communication and coordination in successful software development efforts.

One of the reasons why software development is difficult is the large number of dependencies that exist in the code. These dependencies create communication and coordination needs among software developers.

ARIADNE aims to unpack dependencies between software developers' tasks that arise due to dependencies in the code they are writing. It is a plug-in for the popular Eclipse Integrated Development Environment that provides visualizations of relationships between software engineers, from the code they write. By integrating ARIADNE into Eclipse, we hope to facilitate its adoption among professional software developers

Configuring and Launching

Before invoking ARIADNE, developers need to specify their preferences regarding exactly which Java packages should be included in the analysis. Options exist to save textual data in addition to the graph the tool generates.

From inside the Eclipse workbench, developers can run ARIADNE to learn more about the social structure of the project they are working on.

After right-clicking on a project, ARIADNE presents developers with three visualization options: code dependencies, code dependencies annotated with social dependencies and social dependencies.

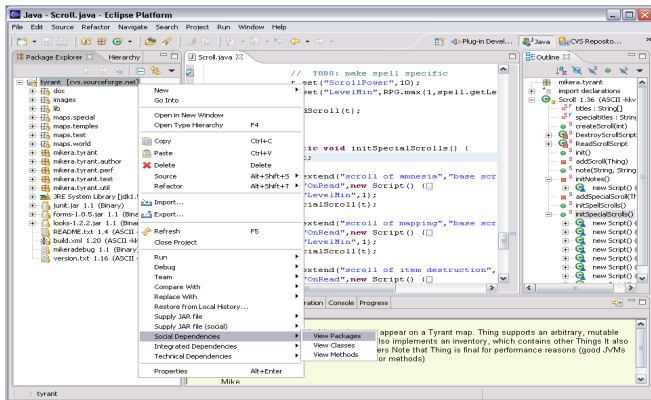


Figure 1: Launching ARIADNE

Understanding the Graphs

A directed edge from node A to B signifies that author A depends on author B. A thick arrow from A to B indicates that author A depends strongly on author B. Only the strongest dependencies are included in these graphs, so a node with no incoming/outgoing edges is an author who neither depends on another developer strongly, nor is strongly dependent upon by another developer. Because of the large number of methods and classes that might exist in a medium-sized Java program, and consequently, of the large size of the code dependencies graph, dependency information is aggregated to facilitate its visualization. Thus, information from the dependencies between methods is aggregated to generate information about dependencies between classes, and, similarly, information about classes is aggregated to generate information about packages.

Visualizing Dependencies

After ARIADNE has generated code dependencies and annotated them with authorship data, it displays a social call graph.

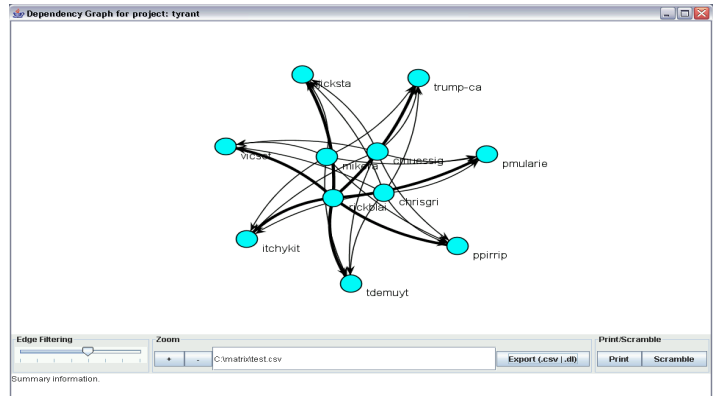


Figure 2a: Social dependencies - dependencies between authors

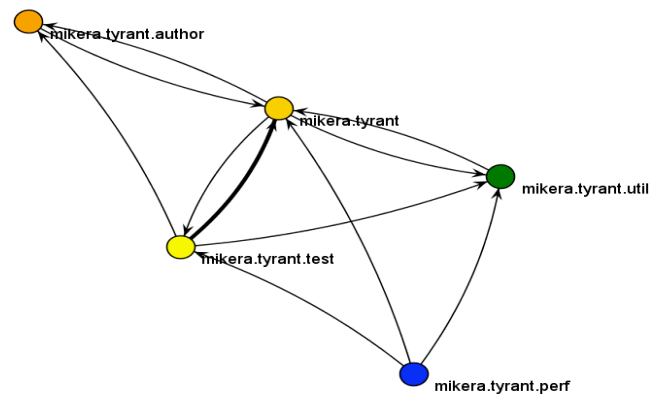


Figure 2b: Technical dependencies – dependencies between code modules

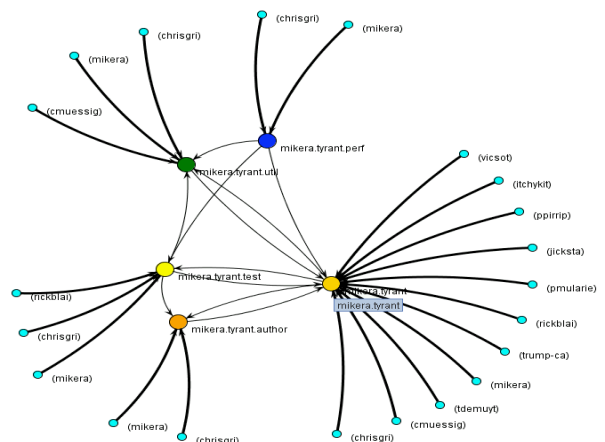
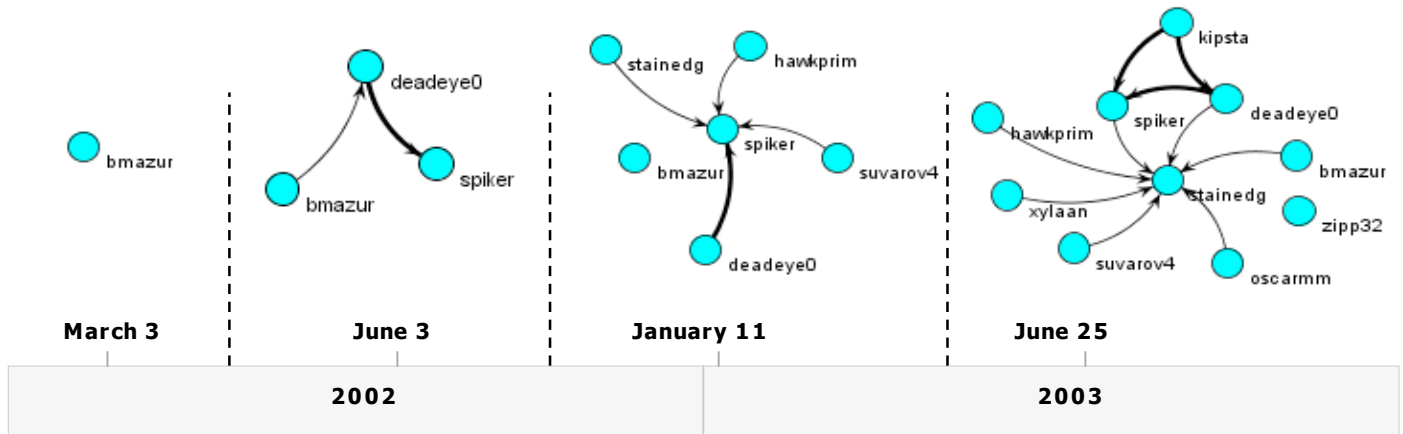


Figure 2c: Integrated code dependencies and authorship information

Evolution of Social Dependencies

One of ARIADNE's features is the ability to download specific releases of a software project from an arbitrary point in time. By using this feature, we created the following four graphs from four distinct releases of the Megamek project. Megamek is an open-source project freely available in the Internet. The graphs convey an idea of how developers' roles in Megamek have changed over time.

As the first graph illustrates, early in the development of Megamek, author "spiker" was the most depended upon. However, the latest release tells us that while author "spiker" remains central to the project, more and more developers are making calls to modules of code that author "stainedg" has implemented. As a result, we expect that the remaining authors are frequently coordinating their work with "spiker" and "stainedg," whose interfaces are heavily depended upon.



Benefits of Ariadne

We envision two types of users for Ariadne:

- **Software developers** who would use it to identify colleagues with whom they need to interact, that need to be informed about changes that are going to impact them, or with similar interests; and
- **Project managers or researchers** interested in understanding the interplay between the changes in the architecture of the software and its social impact. For example, by analyzing the density of a social network or by identifying bridges in this network one can understand the key role played by some software developers in the coordination process or better understand their coordination and communication needs.

The internal architecture and characteristics of ARIADNE offer the following benefits:

- Support for temporal analysis of technical and "social" dependencies in software projects over time;
- Use dependencies across time to identify code that is constantly being fixed, suggesting a need for redesign;
- Allows for appraising the efficiency of software projects by comparing the social relationships that exist among software developers versus those that should exist because of interdependencies in the source-code;
- Extensibility of the dependency extraction module to support different source code languages; and
- Extensibility of the visualization module to support different visualizations. For instance, we plan to use TreeMaps to indicate change impact.

Ariadne's Architecture

Ariadne's architecture consists of four components:

- 1. Core module:** coordinates the generation of the code dependencies and authorship information with the visualization module;
- 2. Code dependency generation module:** generates code dependencies from whatever source language it is defined to accept;
- 3. Authorship dependency generation module:** connects and extracts authorship information from a configuration management system and combines this data with the code dependency data previously generated; and
- 4. Visualization module:** creates a visual representation of the different types dependency information.

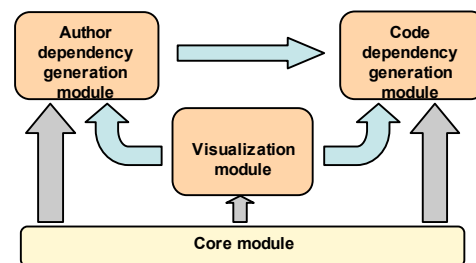


Figure 4: Modular structure of ARIADNE

Contact Information

Professor David F. Redmiles
 Cleidson R. B. de Souza
 Stephen Quirk
 Erik Trainer
 Department of Informatics
 Donal Bren School of Information and Computer
 Sciences
 University of California
 Irvine, California 92697-3425
 {redmiles, cdesouza, squirk, etrainer}@ics.uci.edu

To learn more about ARIADNE and have access to a prototype of the system and documentation, please visit the website:

<http://isr.uci.edu/projects/ariadne>

Effort sponsored by the UROP/UCI, IBM, CAPES (grant BEX 1312/99-5) and NSF (Grant Numbers 0205724 & 0326105).

The content of the information does not necessarily reflect the position or the policy of these institutions and no official endorsement should be inferred.