October 2020

Static Source Code Navigation Tool

N/A

Static Source Code Navigation Tool

ABSTRACT

This disclosure describes a tool for users to navigate software source code. The tool enables users to navigate a sequence of function calls in a customized sequence, maintain a record of functions reviewed and functions still to be reviewed, and record notes associated with each function call. It enables users to maintain track of functions already visited and take notes associated with corresponding code sections. At code branch points and/or points in the code that fork to parallel code paths, users can select any of the available branches/paths to walk through. Based on user selection, a record is maintained of other branches/paths that are pending user review. The features and tools can be utilized for various tasks such as debugging, analyzing how a functionality or feature is implemented, and enhancing/optimizing code.

KEYWORDS

- Code review
- Source code
- Function call
- Call graph
- Code walkthrough
- Software engineering

BACKGROUND

Computer programs (code) commonly include subroutines and/or functions that are called during execution based on triggering rules or variables. Static code graphs and dynamic call graphs (e.g., execution traces) are utilized by programmers to visualize function calls and software program flow. During static analysis of previously written source code for purposes such as code review, debugging, etc., users commonly review the code and navigate a sequence of function calls to understand code functionality and how tasks of interest are implemented. Currently available tools enable users to navigate between functions with use of regular expressions, search tools, etc. Tools with additional features and functionality can further enhance user experience when analyzing code.

DESCRIPTION

This disclosure describes a tool and techniques for users to navigate through or conduct a walkthrough of software source code. Per techniques of this disclosure, a tool is provided that enables users to navigate a sequence of function calls in a user customizable sequence, maintain a record of functions reviewed and functions still to be reviewed, record notes associated with each function, etc. The tool with the described features can be utilized for various tasks such as debugging, analyzing how a functionality or feature is implemented, enhancing or optimizing code, etc.

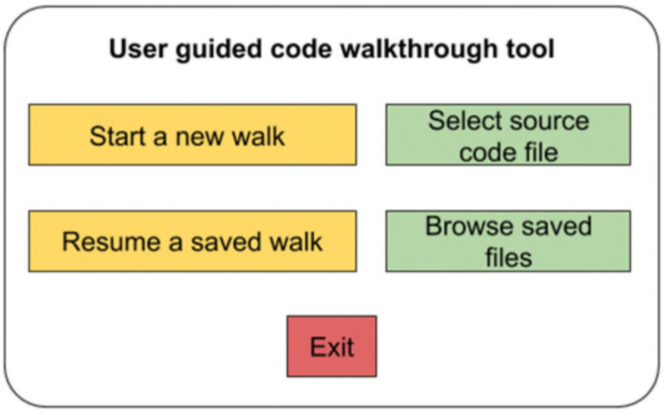

Fig. 1: Example user interface (UI) of code navigation tool

Fig. 1 depicts an example user interface of a user guided code walkthrough tool, per techniques of this disclosure. As depicted in Fig. 1, the tool enables a user to start a new walkthrough or resume a previously saved and suspended walkthrough. Icons are provided to enable the user to easily browse and select source code from different locations.

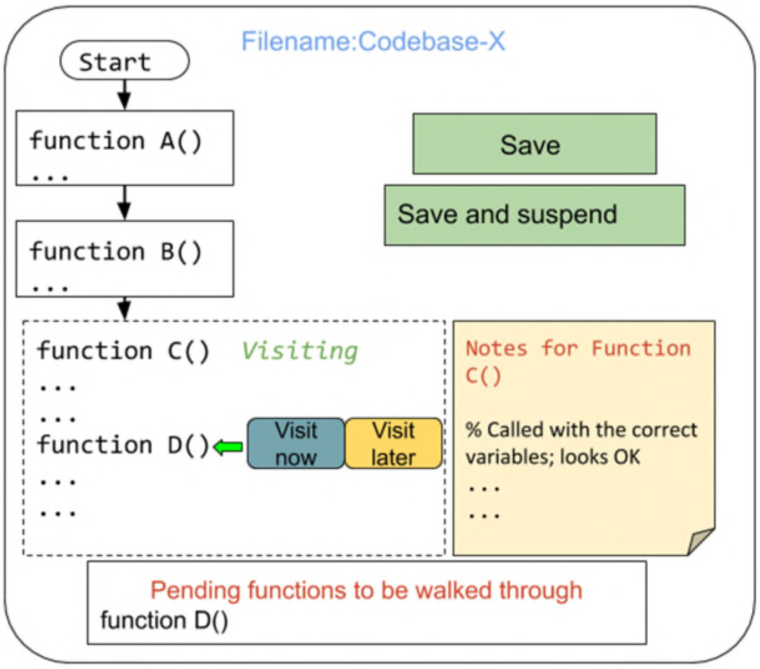

Fig. 2: User guided navigation of source code

Fig. 2 depicts example features of user guided navigation of source code, per techniques of this disclosure. As depicted in Fig. 2, the tool provides features that enable a user to perform a static code analysis. Using the tool, a user can perform code analysis by walking through sections of code and following a user guided sequence of function calls, e.g., to verify logic, accuracy, etc.

The tool enables the user to maintain track of functions already visited by a user in his walk through the code, e.g., by highlighting already visited sections of code. The tool also enables users to take notes associated with corresponding code sections during the walk-through.

As depicted in Fig. 2, at code branch points and/or points in the code that represent a fork to parallel code paths, the user can select any of the available branches/paths to walk through.

Based on user selection, a record is maintained of other branches/paths that are pending user review. The tool enables users to skip a function in the code and walk through the function at a later point of time. Similarly, users can visit a previously unvisited branch. The tool also enables the user to suspend a walk through for subsequent resumption.

Features of the tool provide an efficient way for users to navigate through code by enabling custom navigation of the code, capturing notes and comments within the tool rather than a separate document, and maintaining a record of functions and sections of code that have been visited, and those that still remain to be visited by the user.

CONCLUSION

This disclosure describes a tool for users to navigate software source code. The tool enables users to navigate a sequence of function calls in a customized sequence, maintain a record of functions reviewed and functions still to be reviewed, and record notes associated with each function call. It enables users to maintain track of functions already visited and take notes associated with corresponding code sections. At code branch points and/or points in the code that fork to parallel code paths, users can select any of the available branches/paths to walk through. Based on user selection, a record is maintained of other branches/paths that are pending user review. The features and tools can be utilized for various tasks such as debugging, analyzing how a functionality or feature is implemented, and enhancing/optimizing code.

REFERENCES

1. https://wiki.eclipse.org/Linux_Tools_Project/Callgraph/User_Guide#Watching_function_calls_in_chronological_order, accessed on 12 October 2020.