

Concept Location Based on Programmer's Activity

Pavol ZBELL*

*Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova, 842 16 Bratislava, Slovakia
pavol.zbell@gmail.com*

Search in source code is a necessary part of the daily work of most programmers. Programmers often search and explore source code to enrich their existing knowledge of the workings and functionality of a software system, or to get answers to questions about software evolution tasks they are currently working on, or they search for source code fragments that they could reuse. Research in this area focuses mainly on concept location – a process of identifying an initial location in the source code that implements functionality in a software system. Existing techniques were recently summarized by Dit, B. et al. [1].

Built-in search tools available in IDEs which programmers use usually return a list of functions relevant to specified queries and programmers try to locate desired concept by jumping through functions based on the function calls they see. This approach is generally considered ineffective as it is usually resource and time consuming.

In our work we focus on searching the source code in terms of concept location. We propose a method which takes changes over time to fine-grained elements (functions) of the source code into account. We assume that changes made at a particular time are related, and may represent a concept of the software system. Our project is part of the research project PerConIK (Personalized Conveying of Information and Knowledge, perconik.fiit.stuba.sk) [2].

Our approach to concept location is illustrated in Figure 1. First we watch how programmers modify the source code in IDE and how they contribute these modifications to the RCS system. Then the document creator takes activity logs containing sequences of changes over time made to the source code elements and maps them to corresponding commits obtained from the RCS system. The document creator then produces documents representing source code elements in particular time. The documents containing commit metadata, the original source code and the nearest comments are indexed by the search engine. As long as searching in the source code is

* Supervisor: Mária Bieliková, Adviser: Eduard Kuric, Institute of Informatics and Software Engineering

not the same as searching in plain text, proper indexing is achieved by extracting identifiers from the source code and tokenizing them using the INTT (Identifier Name Tokenization Tool) which is further described in [3]. Above mentioned steps are continually repeated during programming.

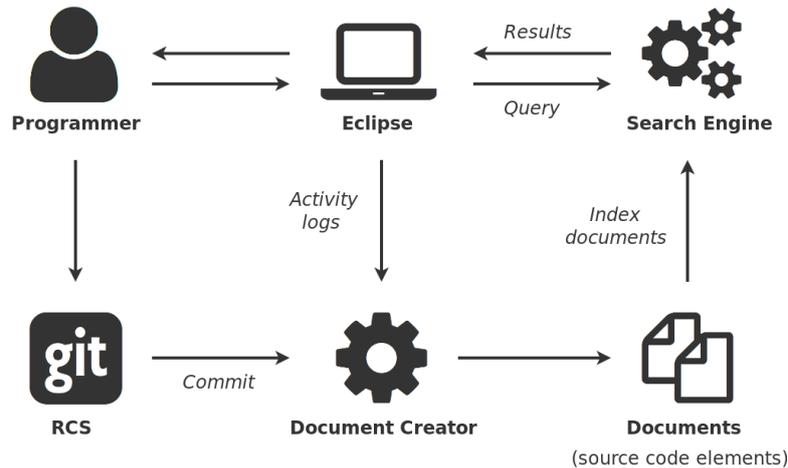


Figure 1. Proposed approach to concept location diagram.

Programmers are able to search in and explore the documents directly from the IDE's interface. The performed source code search is based on simple vector space model, TF-IDF weighting scheme and cosine similarity between terms from programmer's query and terms extracted from the documents – source code elements (identifiers and comments).

We believe that developers using our method will gain a better tool for exploring the source code, and a better overview of the source code evolution and concepts of the software system. They will not need to jump between function calls because they will be able to see the whole evolution of the desired concept.

Acknowledgement. This contribution is the partial result of the Research & Development Operational Programme for the project Research of methods for acquisition, analysis and personalized conveying of information and knowledge, ITMS 26240220039, co-funded by the ERDF.

References

- [1] Dit, B. et al.: Feature location in source code: a taxonomy and survey. *In Journal of Software Maintenance and Evolution: Research and Practice*, 2011.
- [2] Bieliková, M., Návrát, P., Chudá, D., Polášek, I., Barla, M., Tvarožek, J., Tvarožek, M.: Webification of Software Development: General Outline and the Case of Enterprise Application Development. *In Procedia Technology*, 3rd World Conference on Information Technology (2013).
- [3] Butler, S. et al.: Improving the Tokenisation of Identifier Names. *In Proceedings of the 25th European conf. on Object-oriented programming*, 2011, pp. 130–154.