Searching and Exploring Software Repositories in Virtual Reality

Juraj Vincur
Slovak University of Technology in Bratislava
juraj.vincur@stuba.sk

Ivan Polasek
Slovak University of Technology in Bratislava
ivan.polasek@stuba.sk

Pavol Navrat
Slovak University of Technology in Bratislava
pavol.navrat@stuba.sk

ABSTRACT

In this paper, we propose a new approach to visualization of software repositories that allow users to search and explore software projects in virtual reality. In provided environment, respective information structures are mapped to interactive 3D objects. We assume that such transformation of information space to its visible representation will enable the users to gain problem domain knowledge subliminally during the explorations, and that the acquired knowledge will help them to fulfill future tasks more effectively.

CCS CONCEPTS

•**Human-centered computing** → **Visualization;** *Virtual reality;*
•**Information systems** → Information retrieval;

KEYWORDS

virtual reality, software visualization, software repositories, search results visualization, feature location

ACM Reference format:
Juraj Vincur, Ivan Polasek, and Pavol Navrat. 2017. Searching and Exploring Software Repositories in Virtual Reality. In *Proceedings of VRST '17, Gothenburg, Sweden, November 8–10, 2017,* 2 pages.
DOI: 10.1145/3139131.3141209

1 INTRODUCTION AND RELATED WORK

Software developers often reuse source code of existing open-source projects to develop a new software product [Dubinsky et al. 2013]. Source code of these projects is usually available in public repositories. In order to reuse it, developers have to first localize relevant fragments and understand how they work. Thus, feature location and code comprehension are required and necessary.

Feature location [Beck et al. 2015] across multiple repositories is usually realized by the direct search method. Efficiency of the direct search depends on quality of user defined query. To increase the quality, multiple approaches try to assist the user during query formulation and refinement or try to expand the query automatically [Beck et al. 2015]. However, none of these approaches try to explain the search results or problem domain to the user. He will not learn how to define more efficient queries in the future, only the supportive tools might become smarter. Appropriate search result visualization can both make searching more efficient and help users to understand the search results [Beck et al. 2015]. Moreover, visual representations are perceived in parallel, while traditional textual representations are processed sequentially (by reading).

Another problem related to the direct search is that the user does not always know what information he is looking for, or he is not able to express his need by a simple query. In these cases, exploratory search is required. Although there are several approaches that try to reduce the efforts required during the exploratory search [Rozenberg et al. 2016], [Greene et al. 2017], a huge amount of extra effort is still required to understand the related problem domain.

2 OUR APPROACH: VIRTUAL BUT VISIBLE REPOSITORIES

We propose a new approach[1] to visualization of software engineering data that supports exploration of search results and source code repositories in interactive virtual environment. The key idea behind our approach is to transform abstract information space to its virtual but visible representation. In such environment, users search and explore source code fragments and repositories via direct interactions with corresponding virtual objects. We assume that due to required interactions with objects that encode relevant information structures, the users will subliminally gain problem domain knowledge that will enable them to formulate more efficient queries in the future or even support them in program comprehension.

For these purposes, we propose and utilize the new real-life metaphor in which set of repositories is represented by a planetary system. Visualized planets represent repositories relevant to the submitted query (see Figure 1-1) or otherwise related repositories (e.g. repositories with common author or topic). Each planet provide semantic view of a corresponding repository (see Figure 1-2). In addition, selected characteristics of a repository are mapped to visual properties of the planet to support parallel perceiving during the exploration. To observe the structure of the underlying software system, static and dynamic aspects or evolution, users can switch to the city view, where parts of the system selected in planetary view are visualized using the city metaphor (see Figure 1-3). Detailed description of the city metaphor is provided in [Vincur et al. 2017].

The semantic view consists of interactive world clouds that form the surface of the planet. These world clouds visualize semantic clusters obtained by the method proposed by Vincur and Polasek [Vincúr and Polášek 2017]. Clusters identified by this method provide three layer hierarchy, where the first layer consists of classes extracted from the source code, the second provides groups of related classes with common medoid approximation and the third represents the resulted clusters. Each layer is represented by multiple word clouds that are positioned using treemap algorithm (tile size is proportional to the sum of weights of all corresponding terms, see Figure 1-2). Word clouds are linked between layers and

[1]Video footage of our approach. https://youtu.be/UnLD10BfKDg

VRST '17, Gothenburg, Sweden
 978-1-4503-5548-3/17/11…$15.00
DOI: 10.1145/3139131.3141209

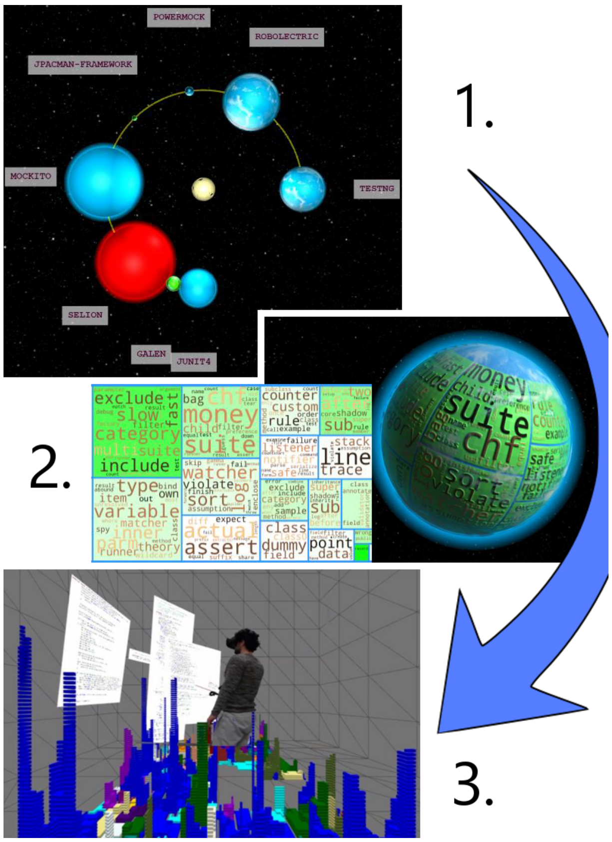

Figure 1: (1) Overview of repositories relevant to query “testing framework java”; (2) semantic view of repository JUnit4 and sample of texture for query “test time out”; (3) exploration of source code fragments directly in VR.

so any word cloud can be selected and expanded into corresponding set of word clouds from the previous layer.

During search sessions, selections made in semantic view limit the search space to the corresponding group of classes (search results filtering). Similar behaviour is achieved by selecting whole planets as well (search support across repositories). Each tile of a word cloud is colored according to the similarity of contained classes to the submitted query (see Figure 1-2). Colors vary from green to white. Green tiles are the most similar, while white tiles are not similar at all. The color of words represents their similarity to query based on Wordnet’s path similarity (the darker - the more similar, see Figure 1-2). During explorations, semantic views provide effective overview of repository since semantic clusters enable users to understand a system’s domain upon first contact.

Visual properties of the planet reflect chosen characteristics of a software repository that might be useful during explorations (see Figure 1-1). The size of the planet (radius) is proportional to the physical size of a repository. Clouds in atmosphere of a planet signalize that these repositories does not have their own project sites. The background color of a planet encodes the date of the last update of the repository (from red to blue, blue planets are most maintained and red planets are dead - least maintained).

The layout of a planetary system varies depending on the use case scenario. During the exploration of search results, constant number of the most relevant repositories are sorted by their score and corresponding planets are positioned in a circle. In the resulted layout, not only the order of the search results is preserved, but also the relative similarity, i.e. how similar are other repositories to the same query (see Figure 1-1). During the exploration of repositories related to the common GitHub topic, all repositories are visualized and their position is calculated in the same way. During the exploration of user’s repositories, score of a repository is substituted by its other characteristic (e.g. physical size, date of creation).

Resulted virtual environment is presented to the user using HMD HTC Vive. Interactions are realized using Vive’s controllers (similarly as in [Vincur et al. 2017]). Navigation in scene is realized by the motion cues as in real environment (e.g. walking). Additional interactions options and informations are available via UI attached to the controller (text inputs, settings, modes, transformations, search results details). The user can also manipulate the objects presented in virtual environment, grab them, change their initial rotation or position and so organize them as he wishes during the exploration.

3 CONCLUSION AND FUTURE WORK

We have proposed a new idea to transform information space to its virtual but visible representation in order to support the user during various tasks. We focused on feature location and exploratory search and we demonstrate our vision on described early prototype.

The open questions are: if provided features of our visualization could provide more efficient way of exploring software repositories, better support parallel perceiving of search result attributes, and maximize comprehension of software knowledge for reusing. We have an opportunity to answer these questions as a part of a new big industrial insurance software project for Australia and Europe, proposed to our university by international software company.

ACKNOWLEDGMENTS

This work is the partial result of grants No. APVV-15-0508 and VG 1/0752/14 and the projects ITMS 26240220084 and ITMS 26240220039, co-funded by the ERDF.

REFERENCES

Fabian Beck, Bogdan Dit, Jaleo Velasco-Madden, Daniel Weiskopf, and Denys Poshyvanyk. 2015. Rethinking User Interfaces for Feature Location. In *Proceedings of the 2015 IEEE 23rd International Conference on Program Comprehension (ICPC ’15)*. IEEE Press, Piscataway, NJ, USA, 151–162.

Yael Dubinsky, Julia Rubin, Thorsten Berger, Slawomir Duszynski, Martin Becker, and Krzysztof Czarnecki. 2013. An Exploratory Study of Cloning in Industrial Software Product Lines. In *Proceedings of the 2013 17th European Conference on Software Maintenance and Reengineering (CSMR ’13)*. IEEE Computer Society, Washington, DC, USA, 25–34.

Gillian J. Greene, Marvin Esterhuizen, and Bernd Fischer. 2017. Visualizing and Exploring Software Version Control Repositories Using Interactive Tag Clouds over Formal Concept Lattices. *Inf. Softw. Technol.* 87, C (July 2017), 223–241.

Daniel Rozenberg, Ivan Beschastnikh, Fabian Kosmale, Valerie Poser, Heiko Becker, Marc Palyart, and Gail C. Murphy. 2016. Comparing Repositories Visually with Repograms. In *Proceedings of the 13th International Conference on Mining Software Repositories (MSR ’16)*. ACM, New York, NY, USA, 109–120.

J. Vincur, P. Navrat, and I. Polasek. 2017. VR City: Software Analysis in Virtual Reality Environment. In *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. 509–516.

Juraj Vincúr and Ivan Polášek. 2017. *An Incremental Approach to Semantic Clustering Designed for Software Visualization.* Springer International Publishing, Cham, 348–361.