# COSME: A NetBeans IDE plugin as a team-centric alternative for search driven software development

Juan M. Fernández-Luna, Juan F. Huete
Departamento de Ciencias de la Computación
e Inteligencia Artificial.
Universidad de Granada,18071.
Granada, Spain
{jmfluna,jhg}@decsai.ugr.es

Julio C. Rodríguez-Cano
Centro de Desarrollo de Software de Holguín.
Universidad de las Ciencias Informáticas, 80100.
Holguín, Cuba
jcrcano@uci.cu

Ramiro Pérez-Vázquez
Departamento de Ciencias de la Computación.
Universidad Central de Las Villas, 50300.
Las Villas, Cuba
rperez@uclv.edu.cu

Chirag Shah
School of Communication & Information, The
State University of New Jersey, 08901.
New Jersey, USA
chirags@rutgers.edu

## ABSTRACT

In the last few years, software developers habits of searching good source-code to reuse or rewrite, has increased the community's interest to improve it and some researchers are staring to refer to as search-driven development (SDD). In this position paper we examine SDD as a collaborative and commonplace task. However, current SDD research and tools are generally considered an individual activity and do not include support for explicit collaboration among developers with shared technical information need. We then introduce COSME, a NetBeans IDE plugin that enables teams of remote developers to collaborate in real time during source-code search sessions. COSME improves the SDD by supporting several collaborative information retrieval techniques such as search sessions persistence, division of labor, sharing of knowledge and awareness.

## Categories and Subject Descriptors

H.5.3 [**Information Interfaces and presentation (e.g., HCI)**]: Group and Organization Interfaces; H.3.3 [**Information Storage and Retrieval**]: Search Process.

## General Terms

Design, Human Factors.

## Keywords

Search Driven Development, Collaborative Information Retrieval, Collaborative Search, Source Code Search, Integrated Development Environment Plugins.



## 1. INTRODUCTION

*"Provided the development coordinator has a communications medium at least as good as the Internet, and knows how to lead without coercion, many heads are inevitably better than one."*.
– E. S. Reymond
*The Cathedral and the Bazaar*

Recently results from some research show that frequently software developers spend time in searching for source-code [1]. That is the fundamental key by there has been considerable effort from both academia and industry in building specialized information retrieval (IR) systems for software developers. Examples include Google Code Search[1], Krugle[2], CodeFetch[3] and Koders[4]. Some of then such as Koders can be integrated with IDEs such as Eclipse and Visual Studio.NET. In addition, software development can be considered as a collaborative activity in which business analysts, customers, system engineers, architects, and developers interact. The concurrent edition of models and processes requires synchronous collaboration between architects and developers who cannot be physically present at a common location.

However, this IR systems do not have support for explicit collaboration among developers with shared technical information needs, which frequently look for additional documentation, read newsgroups for people having the same problem, search the company's site for help, and search for source code examples where other people successfully used [6]. To address this issues is convenient fix SDD with collaborative information retrieval (CIR) techniques. CIR is another emerging research field in charge to establish techniques to satisfy the shared information needs of group members, starting from the extension of the IR process with the knowledge about the queries, the context, and the explicit collaboration habits among group members. We believe that

[1]http://google.com/codesearch
[2]http://krugle.com
[3]http://www.codefetch.com
[4]http://www.koders.com

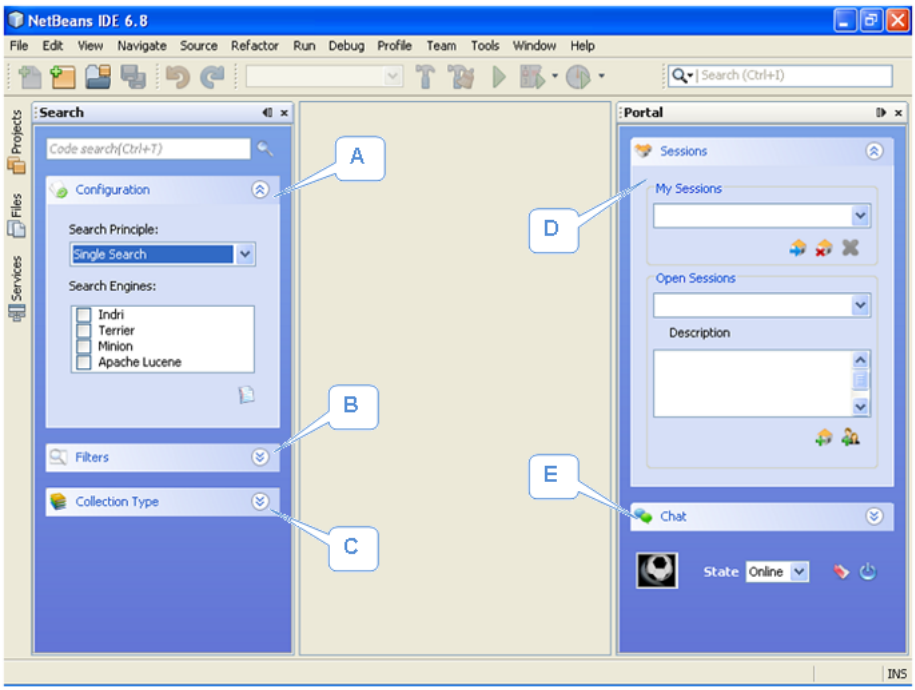


**Figure 1: COSME Client. A) Configuration of search principles and select the search engines, B) Filters specification, C) Collection type, D) Sessions management, E) Chat room**

CIR is going to become an important kind of social search, even though the tools to support collaborative information retrieval are not widespread and are limited by the systems that we currently use.

In this paper we introduce COSME (COde Search MEeting), a NetBeans IDE plugin that enables remote team developers to collaborate in real time during source-code search sessions and any other related technical information. To know the habits of the software developers during the source-code search, we carry out a survey to students and professors that are related with projects of software development in the University of Holguín, as well as to programmers of some companies, for a total of 117 interviewees. We carry out a similar, but smaller study in [3].

The five more commons collaborative search habits of software developers according to this survey are: a) the revision of the problem in the workstation of one of them, b) suggest addresses of Web pages that they have already visited previously, digital books stored in some FTP (File Transfer Protocol) or source files of some projects repository, c) send emails with algorithms or explanatory text, d) search tasks division of each member of the team for sharing the final results, and e) store relevant information in individual workstation. These patterns are caused primarily by the need to:

1. Learn a new programming language.
2. Optimize source-code.
3. Introduce himself new technologies.
4. Lack of documentation.
5. Fix Errors and software maintenance.

Based on survey respondents's descriptions, we identified some CIR techniques for supporting collaborative source-code search. *Search Session Persistence*: storing a search session in a persistent format is a key requirement to facilitate collaboration during the session, revising the search results at a later time, or sharing the results with others [7]. *Division of labor*: Morris's survey in [7] describes ad-hoc methods to avoid duplication of effort during a searching task, such as dividing up the space of potential keywords, searching engines, or sub-tasks among different team members. *Sharing of knowledge*: in any collaborative setting there will be a large and diverse knowledge base shared among team developers. Each one will bring their own experiences, expertise and topic knowledge to a particular searching task. What is needed is a way to enable the sharing of knowledge within the team [5]. Finally *Awareness*: awareness is an essential element in distributed collaborative environments. Over the last decade, a number of researchers have explored the role of awareness for supporting collaboration between distributed groups. Specifically in collaborative SDD, awareness is another important key requirement [10].

After this introduction, in the Section 2, we shall describe the main considerations in the design of COSME as a NetBeans IDE plugin, its implementation and an overview of a simple experimental evaluation based in the comparison of search results of two team of GUI (Graphic User interface) developers. After that, in Section 3 we present a brief description of related works with both CIR and SDD research areas. Finally, we conclude this paper in Section 4 by summarizing the exposed topics in this paper and our main research future direction.

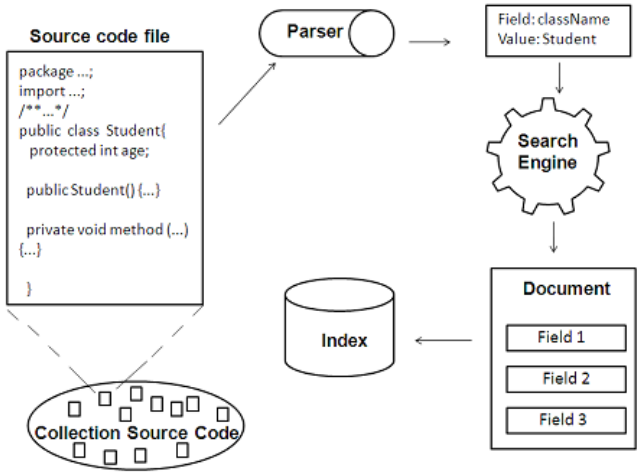


**Figure 2: Fields indexing.**

## 2. COSME

COSME is designed to enable either synchronous or asynchronous, but explicit remote collaboration among team developers with shared technical information needs. The Figure 1 shows the interface of the client side interface. The search panel (left tab) permits to specify the developers queries, division of labor principle (Figure 1: A, some possible combination using different search engines, ranking fusion, and split algorithms), searching field (Figure 1: B, comments, source-code, class or methods declaration), and the collection type (Figure 1: C, source-code files or digital documentation).

The sessions panel (right tab) wraps the principal options to management the collaborative search sessions (Figure 1: D), which consists of a team of developers working together to satisfy their shared technical information needs. For example, a developer can use the embedded chat room (Figure 1: E), to negotiate the creation of a collaborative search session, show comments of the current and historical search results.

### 2.1 Plugin Design and Implementation

For the implementation of COSME we use CIRLab (Collaborative Information Retrieval Laboratory), a groupware framework for CIR research and experimentation [4], Java as programming language, NetBeans IDE Platform as plugin base, and AMENITIES (A MEthodology for aNalysis and desIgn of cooperaTIve systEmS) as software engineering methodology.

CIRLab has been designed applying design patterns and an object-oriented middleware platform to maximize its reusability and adaptability in new contexts with a minimum of programming efforts. The distribution and communication facilities of CIRLab are ICE[5] (Internet Communications Engine) conforming. ICE applications are suitable for using them in heterogeneous environments: client and server can be written in different programming languages, run on different operating systems and hardware architectures, and communicate using a variety of networking technologies.

[5]http://www.zeroc.com

| | $A_{T_1} \Longleftrightarrow A_{T_2}$ |
|---|---|
| $P_s$ | $0,024^{\dagger}(A_{T_1} = 0,0714, A_{T_2} = 0,1440)$ |
| $R_s$ | $0,024^{\dagger}(A_{T_1} = 0,4669, A_{T_2} = 0,6107)$ |
| SIGNIFICANCE (S) | |
| †: significant difference $(0,01 \leq S < 0,05)$ | |
| ‡: highly significant difference $(S < 0,01)$ | |

**Table 1: Wilcoxon Test Results.**

CIRLab also wraps some open-source three party APIs (e.g. search engines and a database engine). To do searches in different parts of the source-code (e.g. comments, class and function definitions) we extend CIRlab with parsers that allow indexing fields (parts of the source-code) when combined with search engines (e.g. Apache Lucene) as you can see in Figure 2.

### 2.2 Experimental evaluation

To evaluate our proposal we compare two different search scenarios on SDD, $T_1$ and $T_2$. $T_1$ represent a team of 5 developers that use conventional IR techniques, and $T_2$ represent a team of 5 developers that use CIR techniques, to satisfy their shared information needs. In both search scenarios, we install COSME in the NetBeans IDE with different configurations. In $T_1$ the developers can use differed search engines and ranking fusion techniques, but COSME was devoid of features such as automatic division of labor, search sessions management, explicit recommendations, among others, while at $T_2$, will be enabled all the CIR features. Our main hypothesis ($H_1$) is that the collaborative work should help to improve the retrieval performance: $A_{T_1} < A_{T_2}$.

We shall considered each search scenario when a common task is proposed to a group of developers without Java background: select the most relevant Java class to management windows visual component using the Java programming language without time limit. In order to measure the effectiveness of the described $T_1$ and $T_2$ we shall consider as a base the metric proposed by Pickens et al. in [9], i.e. selected precision ($P_s$, the fraction of documents - judged relevant by the developer - that were marked relevant in the ground truth), and selected recall ($R_s$) as their dependent measures.

In this experiment we use as test collection four Java APIs: Jidesoft, OpenSwing, SwingX and Swing. We have focussed on these APIs because they are directly related to the context of the experiment. To compute $P_s$ and $R_s$, we have to assess the relevance judgements over the set of source-code files in the test collection. This process have been done in the following way: first, generate a predefined set of queries strongly related with the objective of the experimentation, and second, determine a set of relevant judgements for each predefined query.

For statistical analysis we used the nonparametric Wilcoxon Test, because there not exist a normal distribution of data and we counted with few values of $P_s$, $R_s$. For the Wilcoxon Test we used the Monte Carlo method with confidence intervals at 99% and a number of samples exceeding 10 000. The Table 1 shows the average $P_s$ and $R_s$ values for $T_1$ and $T_2$ obtained during the experiment. The columns represent each team of developers.

Our experiment shows that the best average $P_s$ and $R_s$ results have been obtained with $T_2$, i.e., using CIR techniques

such as search sessions persistence, division of labor, sharing of knowledge and awareness in the SDD context. In the rows of Table 1 you can distinguish that $H_1$ is true according to the Wilcoxon Test result.

## 3. RELATED WORK

COSME include several areas of research, highlights of which CIR and SDD. On the one hand, some researchers have identified different search scenarios where is necessary to extend the IR systems with collaborative capabilities. For example, in the Web context, SearchTogether [8] is a system which enables remote users to collaborate when searching the Web. It supports collaboration with several mechanisms of group awareness, division of labor, and persistence. On the other hand, the SDD community present different prototypes and systems. For example, Sourcerer [2] is an infrastructure for large-scale indexing and analysis of open source code. Sourcerer crawls Internet looking for Java source-code from a variety of locations, such as open source repositories, public web sites, and version control systems.

In contrast to these approaches, COSME[6] makes a contribution in current SDD providing explicit support for teams of developers, enabling developers to collaborate on both the process and results of a search. It provides collaborative search functions for exploring and managing source-code repositories and documents about technical information in software development context. In order to support such CIR techniques, COSME provides some collaborative services. The embedded chat room enables direct communication among different developers. Also relevant search results can be shared with the explicit recommender mechanisms. Another important feature enabling improvement is the automatic division of labor. Through awareness mechanisms all developers are always informed about the team activities to avoid the unnecessary duplication of effort. Awareness is a valuable learning mechanism that help the less experienced developers to view the syntax used by their teammates, and then be inspired to reformulate their queries. All search results can be annotated, either for personal use, like a summary, or in the team context, for discussion threads and ratings.

## 4. CONCLUSION AND FUTURE WORK

Many CIR techniques such as session persistence, division of labor, knowledge sharing and awareness can be applied in several domains. For example, in Web context, interactive multimedia, education and medical environment. We identified SDD as another applicable field, given both the collaborative nature and the interest in having specialized source-code search tools in the area of software development. In this sense we present COSME, a NetBeans IDE plugin designed to enable either synchronous or asynchronous, but explicit remote collaboration among team developers with shared technical information needs. In this sense we carry up an experimentation that show how CIR techniques are useful in the field of SSD, as they help to improve the retrieval effectiveness, since exist significant differences in the retrieval results between traditional IR systems and collaborative IR systems according to the Wilcoxon Test result. In the other hand, and taking into consideration that developers use their workstation as an important dynamic collection of relevant information, we will add to COSME a P2P (peer-2-peer) model with the capability of indexing local collections.

[6]Before COSME, we develop the PosseSrc tool, but this was not integrated to any IDE [3]

## 5. ACKNOWLEDGEMENTS

This work has been supported by the Spanish Ministerio de Ciencia e Inovación (TIN2008-06566.C04.01).

## 6. REFERENCES

[1] S. Bajracharya, A. Kuhn, and Y. Ye. Suite 2010: 2nd international workshop on search-driven development - users, infrastructure,tools & evaluation. In *ICSE '10*, Cape Town, South Africa, May 2010.

[2] S. Bajracharya, J. Ossher, and C. Lopes. Sourcerer: An internet-scale software repository. In *SUITE '09: Proceedings of the 2009 ICSE Workshop on Search-Driven Development-Users, Infrastructure, Tools and Evaluation*, pages 1–4, Washington, DC, USA, 2009. IEEE Computer Society.

[3] J. M. Fernández-Luna, J. F. Huete, R. Pérez-Vázquez, and J. C. Rodríguez-Cano. Improving search-driven development with collaborative information retrieval techniques. In *HCIR '09: Proceedings of the Third Workshop on Human-Computer Interaction and Information Retrieval*, The Catholic University of America, Washington DC, USA, October 2009.

[4] J. M. Fernández-Luna, J. F. Huete, R. Pérez-Vázquez, and J. C. Rodríguez-Cano. Cirlab: A groupware framework for collaborative information retrieval research. *Information Processing & Management*, 46(6):749–761, 2010.

[5] C. Foley, A. F. Smeaton, and H. Lee. Synchronous collaborative information retrieval with relevance feedback. In *CollaborateCom 2006 - 2nd International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 1–4, Atlanta, Georgia, 2006.

[6] K. Krugler and J. D. Mitchell. Search-driven development: Five reasons why search is your most powerful tool, 2007.

[7] M. R. Morris. A survey of collaborative web search practices. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, Florence, Italy, 2008. ACM.

[8] M. R. Morris and E. Horvitz. Searchtogether: an interface for collaborative web search. In *UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 3–12, Newport, Rhode Island, USA, 2007. ACM.

[9] J. Pickens, G. Golovchinsky, C. Shah, P. Qvarfordt, and M. Back. Algorithmic mediation for collaborative exploratory search. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 315–322, New York, USA, 2008.

[10] C. Shah. *A Framework for Supporting User-Centric Collaborative Information Seeking*. PhD thesis, University of North Carolina, School of Information & Library Science, Chapel Hill, USA, June 2010.