

Storage and Retrieval of Software Component using Hadoop and MapReduce

Shekhar Singh¹, Dr. Vikram Bali²

Department of Computer Science & Engineering,
Panipat Institute of Engineering & Technology, Samalkha, Haryana, India

¹shekhar.cse@piet.co.in

²vikram.cse@piet.co.in

Abstract-The meta-data information acquired during the development of software component is very precious which has to be stored and reused for the forthcoming software projects. Effective & efficient storage and retrieval system of software components is much essential in components reuse process in the forthcoming software projects. This paper presents a meta-data model and Hadoop environment for software components reuse that considers semantic information based on ontologies and taxonomies'. The proposed model makes possible the recommendation of interrelated components, as ontology and taxonomies characteristics were incorporated in Hadoop environment. A proposed method helps the software developer for easy storage and retrieval of the needed software artifacts. The storage of software component is managed by the Hadoop Distributed File System (HDFS) and retrieval of the relevant software component is managed by the Map / Reduce concept. The ontology and meta-data helps to extract the knowledge from the legacy system and it helps in understanding the semantic meaning of the user requirements and reusable software component. The results show that the new software component reuse method can evidently improve the software component retrieval precision and take care of the full-scale of the searching results.

Keywords-Software component reuse, Metadata repository, Ontology, Hadoop Distributed File System, Hadoop Environment

I. INTRODUCTION

Software component reuse is the use of existing software component, meta-data and knowledge or artifacts to build new software. Software component reuse is one of the major goals of software engineering research in recent years and component reusability has become an important factor in the process of software development. It reduces software development effort, time and cost and increase reliability, efficiency and flexibility. For software component reuse successfully, it is fundamental to choose appropriated reusable components from a collection of available components. In fact, the availability of software reusable assets in software development phases provides valuable support to design and implementation with software architectures by improving productivity, quality, reliability, speed of delivery, and in long-term decreases in costs for software development and maintenance of software. Thus, it is desirable to have a Hadoop Distributed File System for repository that supports the storage, query and retrieval of components and makes reuse possible. A technique to reusable software component repositories is needed that provides the retrieval of semantically interrelated software components. This paper presents ontology and Hadoop Distributed File System based meta-data repository and component repository for storage and retrieval of software components. The metadata repository integrates expert knowledge of correlative domains, reusable software component and generalizes crucial concepts and relations among concepts in these domains and software component [3] [10]. These query terms which are formed in virtue of metadata knowledge of software component can improve the software component retrieval precision. The function of a software component reuse (SCR) system is that construct the model of software component retrieval, in the model, functions, applied domains, work environments, working , static and dynamic behaviors of a software component can be accurately expressed, the software component can be store, searched and reused [3] [19].

II. SOFTWARE COMPONENT REUSE (SCR) SYSTEM

The proposed SCR system uses a multi-agent and meta-data approach to implement several reasoning mechanisms, searching mechanism like: knowledge access, process workflow automation or automated code generation and generation of software component knowledge. They use three different ontologies: Meta-data Ontology, Component Ontology and Describing Ontology. The SCR system is three tier architecture with data layer (Model), Application layer (Controller), View layer (View) where Model-View-Controller system is well established for software component code reuse.

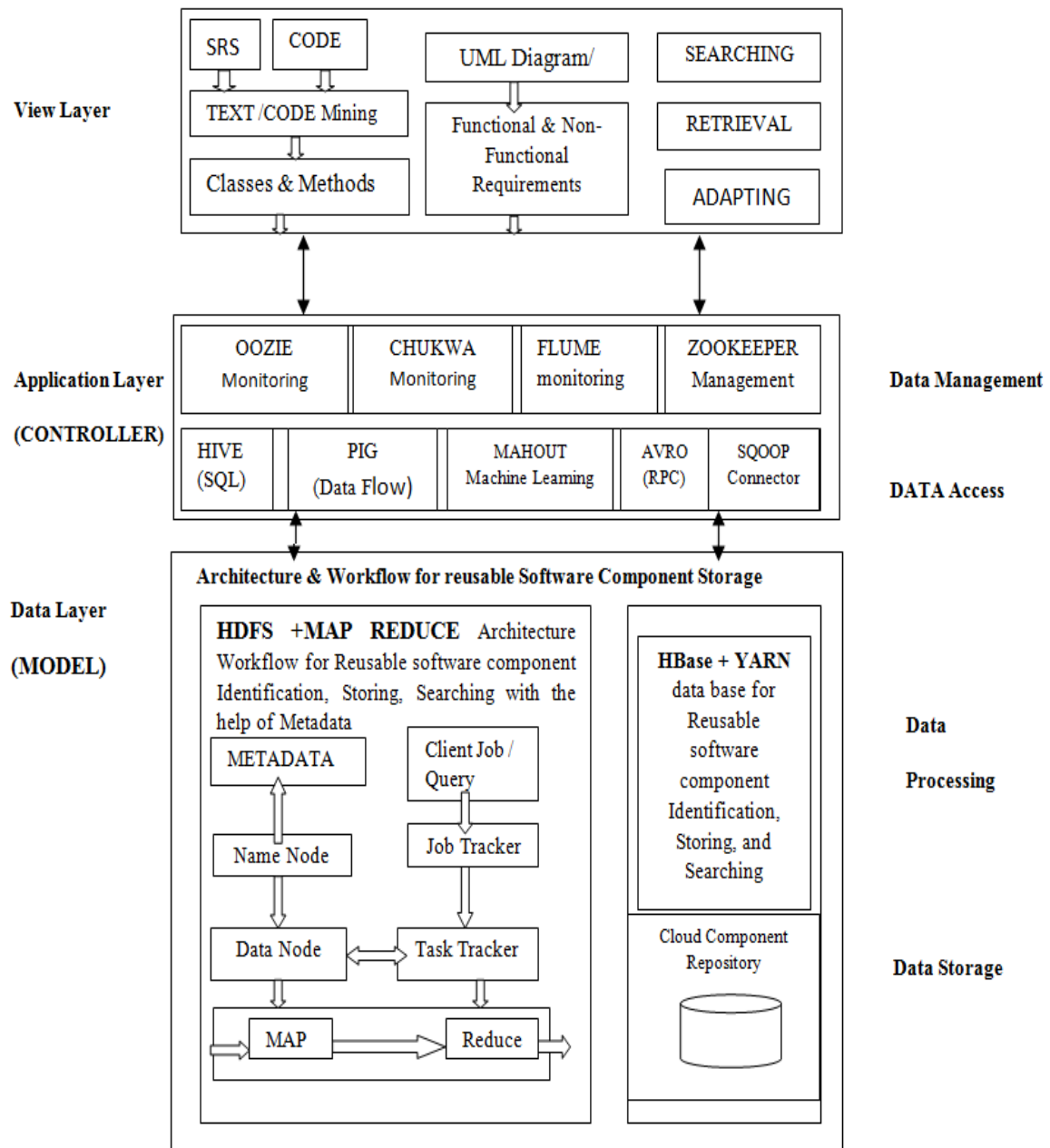


Fig 1: Software Component Storing and Retrieval System

This system works as a platform that provides a way to store the Software development and component reuse knowledge in the Ontology based Meta-data Repository in the form of Name nodes. Reusable software component being stored in the HDFS file system and accessible through an interface framework, the different reusable elements should be described using a software artifact Ontology in describing & Meta-Data repository and classified through Meta-data & Describing Ontology. Meta-Data & Describing knowledge representation structures will be used to empower the search mechanisms and make easier the reuse of stored component.

In figure 1, we present the software component reuse system's architecture. The system can be structured in three different logical layers: Data (Model), View and Application (Controller). The Data layer corresponds to the knowledge base and Meta-Data; it stores all the reusable software components and knowledge needed for the reuse system reasoning, including the knowledge representation structures and the components. The Controller layer implements the searching mechanisms, including the search, suggestion and browsing facilities. The View layer provides interface for software component retrieval and comprises the semantic web service and the applications that use the system's functionalities through this semantic web service. The proposed system uses a multi-agent and meta-Data approach with Hadoop architecture to implement several reasoning mechanisms for reusable software component storing and retrieval, like: knowledge access, process workflow automation or automated code generation and effectively reusing software component. Hadoop model does not include the applications and end-user presentation of reusable software components. The representation of

component is broken down into the Hadoop use cases from above Compute, Storage, retrieval and Database workloads Each workload has specific characteristics for operations, deployment, architecture and management.

Proposed Hadoop architecture for reusable component retrieval has a variety of node types within each Hadoop cluster; these include DataNodes, NameNodes, and EdgeNodes. Hadoop's architecture for software component storage and retrieval is modular, allowing individual components to be scaled up and down as the needs of the environment change. The NameNode in Hadoop's architecture is the central location for information about the file system and contacted by clients of the Hadoop HDFS to locate information within the file system and provide updates for data they have added, moved, manipulated, or deleted. The DataNode serves two functions: It obtains portion of the reusable component data in the HDFS and it acts as a compute platform for storing and retrieval of component, some of which will utilize the local data within the HDFS. The EdgeNode provide interface and access point for the external applications, tools, and users that need to utilize the Hadoop environment for storing and retrieval of component. Controller work as a search engine and controller acts as a middleware between the model and view where the searching techniques for components is implemented for reuse software component.

III. SOFTWARE COMPONENT SEARCHING & RETRIEVAL PROCESS

Component retrieval is implemented based on the architecture of the SCR system that is shown in Figure 1. The SCR system searching is based on ontology-based metadata. Ontology characteristics are suitable for software component retrieval, as they allow capturing domain semantics and recommending interrelated software components. Meta data elements permit retrieving and recommending components based on the analysis of semantic information. Ontology with metadata based search uses the domain ontology for identification of the domain knowledge and enrich the query to get more relevant component from the repository. The ontology repository can be used to extract the code concept in the domain and the sub domain of a particular context and Ontology Repository contains the concept Metadata. After receiving the patterns in the Ontology repository we apply concept with semantic similarity based search mechanisms to extract the correct component related to user requirement. All the retrieved software components from the repository are listed and the most relevant is chosen. The relevant component can be used fully, partially or adapted according to the current need. Adapting is the concept of enhancing or expanding the component according to the current need of the project. First layer of searching system define as a user interface layer were the user interacts with the system by uploading the Software requirement Document for searching the relevant code and design from the repository. Text mining with meta-data is the concept of mining the text document and identifying the most needed information according to the context & similarity. In this model text mining with meta-data is implemented in the SRS document where the where the whole text are mined. The noun part specifies the need class or entity in the project domain and the verb part shows the needed service or method in the system. The list of classes and methods are identified from the SRS document using the text mining concept. This given as search query to retrieve the relevant component from the cloud based repository. The class and method in the SRS helps to identify the need functionality of the current project.

IV. RESULTS ANALYSIS

Precision: Precision is defined as the number of relevant components retrieved divided by the total number of components retrieved.

Recall: Recall is defined as the number of relevant components retrieved divided by the total number of relevant components in the index.

TABLE I Comparison of Results

Sr. No	Method Name	Total Relevant Components in data base	Total Components Retrieved	Total Relevant Components Retrieved	Precision Value	Recall Value
1	Proposed Method	300	290	275	95	94
2	MDL File (UML Diagram) Based search	300	240	210	87	70
3	Keywords Based Search	300	252	180	71	72

V. CONCLUSION

The benefits of Hadoop environment for reusable software component are best realized when it is teamed with a good archival solution that is cost effective, scalable and secure storing & searching of reusable software components, allowing for speedy data retrieval and analysis. By using Hadoop as an archival platform, organizations can benefit from multiple source connectivity, scalability, cost benefits, and high accessibility of archived data for reuse & analytics.

In this research work we described the CSR system, which intends to explore new ways to store and reuse software component & knowledge. In this system cloud based storage and Semantic based retrieval is used. The CSR platform provides several ways to reuse software component and share knowledge, including a hands-on way of suggesting relevant component & knowledge to the software developer using the user context. Semantic web technologies enable the association of semantics to software artifacts, which can be used by inference engines to provide new functionalities such as semantic retrieval, or suggestion of relevant knowledge. SCR provides way to integration with client applications through the semantic web service. Thus, a tool for Software Engineering and MVC (Model View Controller Architecture) that use metadata can be integrated in SCR has knowledge providers. Combining software reuse Cloud and metadata is a new emerging trend in software development process. Combining these technologies helps the software development process by locating pre-existing components at the design & implementation time due to which the total effort of software development is decreased.

REFERENCES

- [1] Lina and Z. Shijie, "Progress and prospects of expert system", *Application Research of Computers*, vol. 24, no. 12, pp.1-5, 2007.
- [2] Z. Zipeng and L. Longshu, "Construct the expert system knowledge base with XML", *Computer Technology and Development*, vol. 17, no. 7, pp.31-34, 2007.
- [3] Jayasudha Subburaj, S.Subramanian, "Cloud and Ontology Based Software Reuse (COSR)", *journal of nano science Nano technology*, ISSN 2279 – 0381, Vol 2 , Issue 3 Spring Edition, Pp 357-363, February 2014.
- [4] Y. Wensheng, T. Pinghui and C. Xiuguo, "Problem Oriented Analysis and Decision Expert System with Large Capacity Knowledge-base", *Proceedings of 2008 International Conference on Intelligent System and Knowledge Engineering*, China, pp.32-37, 2008.
- [5] F.Yuku, "Dynamic Behavior Specification of Web Component Based on Logic Programming", *Proceeding of the First International Multi-Symposiums on Computer and Computational Sciences*, pp.480-482, 2006.
- [6] L. Xiaobo, M. Huaikou and L. Jing, "Components Matching Based on Formal Specifications", *Computer Applications and Software*, vol. 23, no. 10, pp.10-12, 2006.
- [7] L.Yuhua, Z. Bandar and D. Mclean, "An approach for measuring semantic similarity between words using multiple information sources", *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp.871-882, 2003.
- [8] Haining Yao, Letha Etzkorn; *Towards A Semantic based Approach for Software Reusable Component Classification and Retrieval*; ACM Software Engineering, 2004.
- [9] Vitharana, Padmal, Zahedi, Fatemeh M., Jain, Hemant; *Knowledge-based repository scheme for storing and retrieving business components: a theoretical design and an empirical analysis*; *IEEE Transactions on Software Engineering*, 2003, vol. 29, no. 7, pp. 649-664.
- [10] Rajender Nath, Harish Kumar; *Building Software Reuse Library*; 3rd International Conference on Advanced Computing and Communication Technology- ICACCT-08; Asia Pacific Institute of Information Technology, Panipat , India; November 08-09, 2008, pp. 585-587.
- [11] Rajesh K Bhatia, Mayank Dave, R.C Joshi, "Retrieval of most relevant reusable Component using genetic algorithms", *Software Engineering Research and Practice* 2006, 151-155.
- [12] Rajesh K Bhatia, Mayank Dave, R.C Joshi, "A Hybrid Technique for Searching a Reusable Component from Software Libraries", *DESIDOC Bulletin of Information Technology*, Vol.27, No.5, September 2007, pp. 27-34.
- [13] James manyika, michael chui, brad brown, jacques bughin , richard dobbs, charles roxburgh,,"big data: the next frontier for innovation, competition, and productivity" may 2011.
- [14] Marc wilhelm k'uster christoph ludwig "software reuse through resource registries position paper", 19th australian conference on software engineering, doi 10.1109/aswec.2008.87
- [15] Michael t. helmick, james d. kiper, janet burge, valerie cross, and gerald gannod,,"incorporating wikis into software repository mining", 1530-0803/08 \$25.00 © 2008 IEEE.
- [16] Panos constantopoulos, martin doerr and yannis vassiliou "repositories for software reuse: the software information base", *international journal of emerging technology and advanced engineering website: www.ijetae.com (issn 2250-2459, volume 2, issue 4, april 2012)* 578
- [17] Ronaldo c. veras and silvio r. l. meira, Adriano l. i. oliveira and bruno j. m. melo "comparative study of clustering techniques for the organization of software repositories"1082-3409/07 \$25.00 © 2007 IEEE.
- [18] S. thaddeus, s.v.k.r.: a semantic web tool for knowledge-based software engineering. in: 2nd international workshop on semantic web enabled software engineering (swese 2006), athens, g.a., usa, springer (2006)
- [19] younas wahab, muhammad imran babar, shahbaz ahmed,"single repository for software component selection (srscs): a reusable software component selection technique", *journal of theoretical and applied information technology*, april 2011 vol 2619th IEEE international conference on tools with artificial intelligence.