An Open Source Multi Agent System for Data Preprocessing of Online Software Bug Repositories

ABSTRACT

Software bug repositories contain lot of useful information related to software development, software design and software's common error patterns. Most of the projects use bug tracking system to manage the bugs associated with the software. These bug tracking system works as an online bug repositories, which can be accessed by all of the project members situated at different locations. Researches can also access these online bug repositories for exploring knowledge from them. In order to extract knowledge from software repositories some preprocessing mechanism is required to extract, parse and save the data locally from these online repository. To address this problem an open source multi agent system is proposed in this paper for the preprocessing of online software defect repositories. The proposed system is also implemented using the open source technologies. Software agents are independent software units and works intelligently and also getting very popular for current and future research, and hence the concepts of agents are included for preprocessing task and multi agent system is implemented. The implementation is done using open source application programming interfaces (API's) and also performance is evaluated for the implementation in terms of bug data fetch and parse timings.

Keywords

Software bug repositories, Multi agent system, Fetching online bug repositories, Parsing software bugs, Preprocessing of online bug repositories.

INTRODUCTION

Data mining is a process of knowledge discovery in the databases. It is used discover useful and interesting patterns from a database by analyzing the history of the data. Online software repositories contain lot of knowledge regarding the software projects. It includes the software design related facts, common error patterns, software standards etc. Data mining can be applied on these software repositories to discover this knowledge. In order to apply data mining on databases, some preprocessing is required to keep the data in some standard and uniform manners. Online software repositories contains the data online, through web interface user can retrieve some fraction of the data kept in online repositories. So data mining can not be applied directly to these online software repositories, since all the data is not available at a time.

The multi agent systems (MAS) are getting very popular over the years. The elementary unit of MAS is the software agents, which are independent software and can, interact with the other agents. Generally for the complex and distributed problems MAS systems are proven best. MAS can be used in scenarios where the bigger task can be divided into the smaller independent task and some communication is also required for the smaller tasks. Data preprocessing of online software repositories includes lot of tasks which can be invoked individually and sometime some communication is also required between these tasks. Some of the useful term which are included in this paper are:

1.1 Agent

An agent is an autonomous software unit that can exist independently of other similar units in the software system. An agent performs some functions for other agents or external actors. Agents communicate with each other via messages via an agent communication language. As per the Foundation for Intelligent Physical Agents (FIPA), the agent definition is: "An agent is the fundamental actor in a domain. It combines one or more service capabilities into a unified and integrated execution model which can include access to external software, human users and communication facilities". [4]

1.2 Multi Agent Systems (MAS)

Multi-agent systems (MASs) are systems consisting of more than one autonomous agent that are able to interact with one another. These agents may have a global goal to solve together, or they may individually have their own goals to pursue. The particular characteristic here is that in order to achieve their goal(s), these agents must coordinate their actions [4].

1.3 Online Software Defect Repositories

For larger applications where the developers are working across different locations it is very difficult to manage the project related data. In a distributed development environment also it is very difficult to verify, integrate and manage changes in source code repository. To handle this situations there are number of tools available. These tools provide access to repositories through internet, which is called as online repositories. Using these tools project members can easily integrate and verify the individual changes. Managers can also verify and manages the individual's changes done in the repositories. To handle to software bug related data, there are tools like JIRA [12], Perforce [14], and BugZilla [8] etc. These tools provide the online way to deal with the software defect repositories. Developers, quality engineers and managers can log in to these repositories to fix the bugs, manages the bugs, comments on bugs and to verify their status.

1.4 Data Preprocessing

Data preprocessing refers to get the data and make it ready for the data mining operations. In order to apply data mining on online software defect repositories lot of data preprocessing task is required. The first main task includes retrieving the software defects on from online repositories. Every repository maintains the software bugs in some predefined format, most of the time it is either HTML (Hyper Text Markup Language) or XML (eXtensible Markup Language). The next important task in data preprocessing is the parsing the actual data out of the various formats and saving the data into some defined database schema.

Data preprocessing of online software repositories requires some special mechanism e.g. network programming is used to communicate with the online repositories.

The advantages of the software agents are taken to make the data preprocessing task more effective. The structure of this paper is as follows. In section 2 related works done in the similar area is discussed. Section 3 presents the proposed open source multi agent system. Section 4 describes the implementation of the proposed multi agent system and also mentioned the various open source API's. Section 5 describes performance evaluation of the implementation. Section 6 summarizes the contribution and future scope of the research.

RELATED WORK

There is several number of research problems associated with the online software bug repositories. Some of the studied research work is discussed here. The problem of duplicate bug detection in online bug repositories is addressed by many researchers which are studied in [7]. Similarity and duplicity detection method of graphical user interface (GUI) bugs is proposed by Nagwani and Singh [5]. Knowledge discovery techniques in online software repositories are given in [1-2].

There is several numbers of frameworks and other tools also exist, but they are addressing the online software code retrieving, and preprocessing. An Open Framework for CVS Repository Querying, Analysis and Visualization is proposed in [3]. A tool named Sourcerer is developed to support the infrastructure for the automated crawling, parsing, and database storage of open source software [6]. So none of them support the functionalities associated with the online software defect repositories, hence an approach is required to deal with the software defect repositories.

A new approach is proposed towards an integrated multi agent system for fetching, parsing, traversing and schema generation for software bug repositories. The goal is to provide users to data preprocessing of online software bug repositories. The proposed multi agent system is implemented using open source technologies only and experiments are also done for performance analysis.

PROPOSED MULTI AAGENT SYSTEM

The proposed multi agent system (MAS) is represented in figure-1. It consists of four software agents. All the agents are developed using the open source technologies under JADE (Java Agent Development framework) environment. The four agents and their responsibilities are as follows:

1. Fetch Bugs Agent - The fetch agent is created to fetch the bug stored in on line bug repositories. Each online bug repositories have some standard uniform resource locator (URL) pattern, by specifying the bug id the software bug can be retrieved locally. For example Mozilla online repository have the URL pattern as "https://bugzilla.mozilla.org/show_bug.cgi?id=" by appending a particular id (an integer), a Mozilla bug can be fetched locally. If the connection requires some proxy address settings, then user can also specifies the proxy address and port number using the fetch agent properties. User can also specify the location where the bugs can be stored locally.

2. Parser Bugs Agent - Once the software bug is retrieved at local machine, the retrieved software bug format is either html or xml. Since online repositories supports either the html or xml formats only. Parser module is responsible for parsing the html or xml bugs and save the bug information in local database.

3. Schema Generator Agent - To store the bug information after parsing, schema of the tables should be defined in local database. Using schema generator agent user can specify the table schema at the local database.

4. Database Traversal Agent - The purpose of traversal agent is to traverse through with the local stored database. Traversal agent is created in generic way, which supports all the types of schemas defined for the database. The graphical user interface (GUI) fields for the columns are generated at runtime based on the schema defined for the local database.

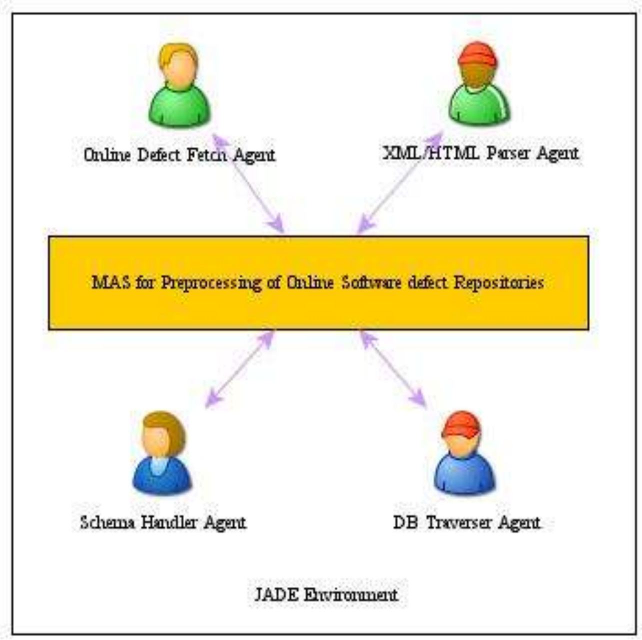

Online Defect Fetch Agent
XML/HTML Parser Agent
MAS for Preprocessing of Online Software defect Repositories
Schema Handler Agent
DB Traverser Agent
JADE Environment

Figure 1. Proposed Multi Agent System for Data Preprocessing of Online Bug Repositories

IMPLEMENTATION

Implementation is done using various open sources API's (Application Programming Interfaces). Implementation is done using generic interfaces and can be extended to accommodate more functionality and future changes. Some of the open source API's which are taken for the implementation is:

JADE

Jade (Java Agent DEvelopment Framework) provides the platform for creating multi agent system in java [9]. It is open source API and designed as per the FIPA (Foundation for

Intelligent Physical Agent) standards. It is fully implemented in Java language. It simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications and through a set of graphical tools that supports the debugging and deployment phases. All the major functionalities like agent co-ordination, agent management, and agent communication is supported by JADE API.

Java, JDBC and Swing

Java is the language used for the implementation of proposed MAS. JDBC refers to Java Data Base Connectivity, which provides a set of interfaces to communicate with back end database to the programming language. It works as a middle tier in application to fetch the data from the database and to store the data back into the database. Swing is java based open source technology which is used to create GUI’s for the java based project. It also provides some built in classed for parsing the HTM L documents. In the current implementation Swing is used for creating various agent properties GUI’s and is also used for parsing the HTML documents.

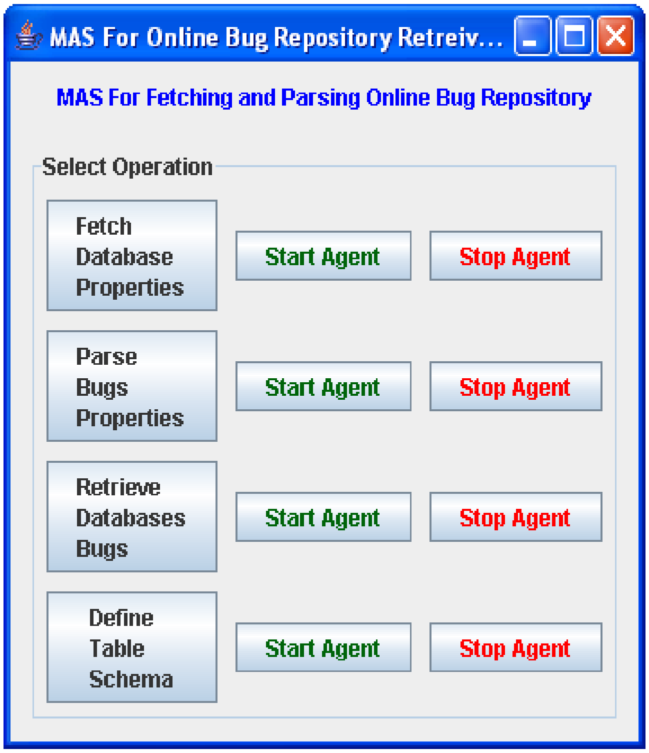


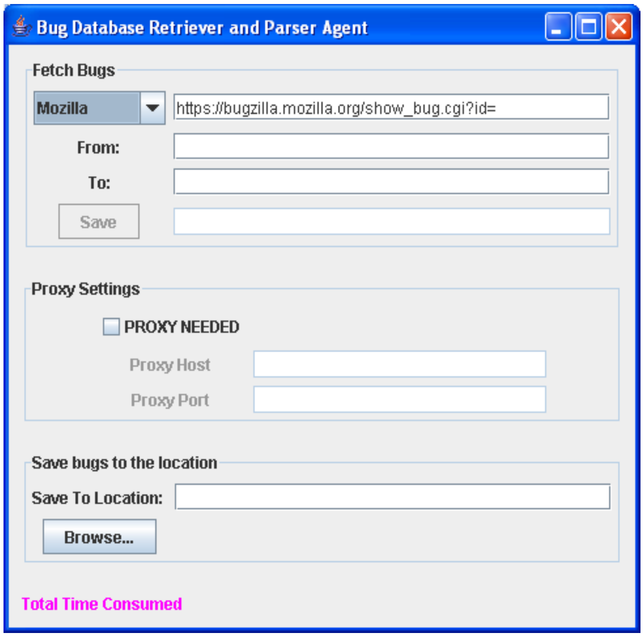
Figure 2. Interface for proposed MAS to control various Agents

MySql

Mysql is the open source free database [13]. It is very popular and good choice for the open source development work. It provides the database connectivity with the most of the programming languages available.

JAXP

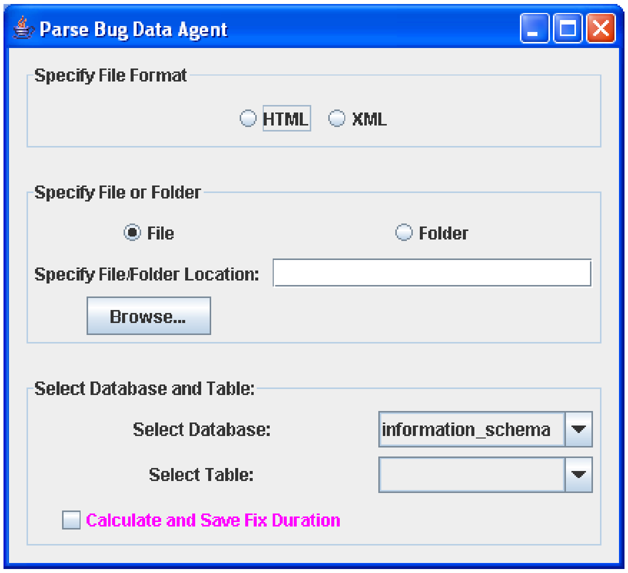
JAXP refers to Java API for XML Parsing [11]. The Java API for XML Processing (JAXP) enables applications to parse, transform, validate and query XML documents using an API that is independent of a particular XML processor implementation.



Mozilla https://bugzilla.mozilla.org/show_bug.cgi?id=

□ PROXY NEEDED

Figure 3. Software Bug Fetch Agent Properties GUI



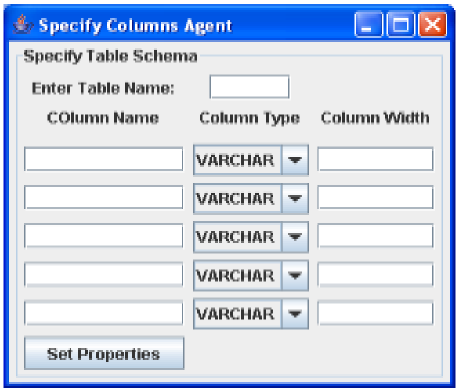
○ HTML ○ XML

● File ○ Folder

Select Database: information_schema

□ Calculate and Save Fix Duration

Figure 4. Parse Software Bug Agent Properties GUI



Specify Columns Agent

Specify Table Schema

Enter Table Name:

COlumn Name	Column Type	Column Width
	VARCHAR	
	VARCHAR	
	VARCHAR	
	VARCHAR	
	VARCHAR	

Figure 5. Schema Generator Agent Properties

Snapshot for some of the software agents and its properties are presented in this paper. Figure 2 depicts the main GUI to control all the four agents. Properties associated with the agents can be modified by clinking on buttons for agents. Once the properties of the agents are set, they runs automatically, they can also be controlled by start and stop buttons provided in the main GUI. Figure 3 depict the properties associated with the fetch agent. User can specified the properties like online repository URL, range of the bugs, proxy related settings etc on the property window of fetch agent. Figure 4 depicts the property GUI for parse agent, currently parsing is supported only for HTML and XML. User can specify the parse document format, and can also select the local database where the parsed values can be saved. And figure 5 represents the some of properties associated with schema generator. Using schema generator user can create, edit and drop the database schema and can bind this schema to save the parsed software defects.

PERFORMANCE EVALUATION

Retrieving timings from different defect repositories and Parsing timings are calculated for performance evaluation of the implemented framework. Data is taken from open source bug repositories of projects like Mysql, JBoss-Seam, Mozilla projects.

Table 1. Fetch timings in ms for various online repositories

No. of Bugs	100	200	500	700	1000
MySql	138516	336848	672628	857 723	1391885
Mozilla	98639	275122	827876	1227289	2004878
Jboss-Seam	221764	460078	1400084	1753747	2701744

Table 2. Parse timings in ms for various online repositories

No. of Bugs	100	200	500	700	1000
MySql	138712	242165	740782	1074279	1695034
Mozilla	156978	274612	858684	1268449	1947208
Jboss-Seam	240342	413128	1334143	1748400	2032180

Table-1 contains the fetching timings of fetch module from different bug repositories for different numbers of software bugs. Table-2 contains the parsing timings of fetched software bugs from different repositories in html format for different numbers of bugs.

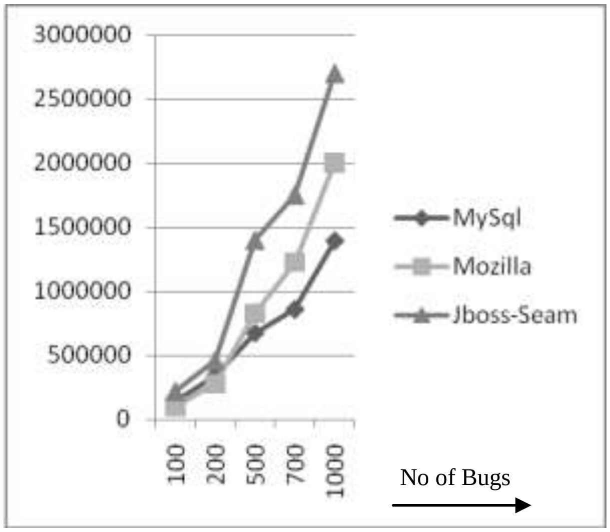


Figure 6. Graph displaying the fetch timings of bugs in ms.

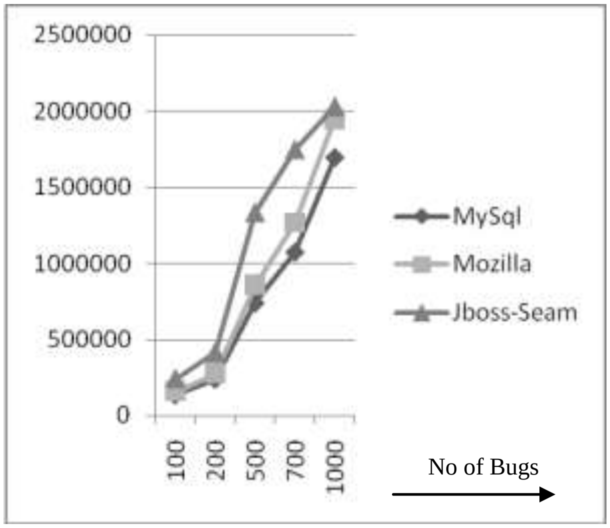


Figure 7. Graph displaying the parse timings in ms.

Figure 6 depicts the fetching timings from different online bug repositories in ms and figure 7 depicts the parsing timings for fetched bugs in ms for the different online bug repositories. Behaviors of both of the operations are almost same, time consumed increases linearly with increase the number of bugs to fetch and parse. Each bug repository consists of different size of bugs; therefore the overall time consumed by parsing and fetching is increasing almost in linearly order.

CONCLUSION AND FUTURE SCOPE

In this paper an open source multi agent system for data preprocessing of software bug repositories is proposed. This system will help the research community to experiment techniques in online software bug data retrieving, parsing and traversing with the help of software agent based system. As a future direction of research improvements can be done by adding more features and functionalities for the various agents and data cleaning task can also be implemented as software agents and can be added to the proposed multi agent system. Additionally the proposed system can be extended for fetching and parsing the online source code repositories also apart from the online software bug repositories.

REFERENCES

Erik Linstead, Paul Rigor, Sushil Bajracharya, Cristina Lopes and Pierre Baldi, “Mining Internet-Scale Software Repositories”, Advances in Neural Information Processing Systems (NIPS) , Vol. 21,2008.

Fatudimu I.T, Musa A.G, Ayo C.K, Sofoluwe A. B, “Knowledge Discovery in Online Repositories: A Text Mining Approach”, European Journal of Scientific Research, ISSN 1450-216X, Vol. 22 No. 2, pp. 241-250, 2008.

Lucian Voinea, Alexandru Telea, “An Open Framework for CVS Repository Querying, Analysis and Visualization”, Proceedings of the 2006 international workshop on Mining software repositories table of contents, Shanghai, China, pp. 33 - 39, 2006.

Margus Oja, Boris Tamm and Kuldar Taveter: Agent-Based Software Design, Proc. Estonian Acad. Sci. Eng., 7, 1, 5–21, 2001.

Nagwani, N.K. Singh, P., "Bug Mining Model Based on Event-Component Similarity to Discover Similar and Duplicate GUI Bugs", Advance Computing Conference, 2009. IACC 2009. IEEE International, pp.1388-1392, 2009.

Sushil Bajracharya, Joel Ossher, Cristina Lopes, "Sourcerer: An internet-scale software repository", Proceedings of the 2009 ICSE Workshop on Search-Driven Development-Users, Infrastructure, Tools and Evaluation table of contents, pp. 1-4 , 2009.

Xiaoyin Wang, Lu Zhang, Tao Xie, John Anvik and Jiasu Sun: An Approach to Detecting Duplicate Bug Reports using Natural Language and Execution Information, ACM, ICSE’08, May 10–18, 2008, Leipzig, Germany.