

A Review on Mining for Web Engineering Business Intelligence

V. Sumalatha¹, T.Nishitha²,

¹Assistant Professor, Malla Reddy Engineering College, MRITS, CSE Dept, Hyderabad, TS, India.

²Assistant Professor, CVR College of Engineering, CSE Dept, Hyderabad, TS, India.

Abstract: Over the last few years, there has been a remarkable increase in use of World Wide Web for a wide and variety of applications. The web plays a central role in e-commerce applications. As a consequence, there is a need to improve Intelligence of Web Engineering Applications in the context of Business and IT industry. To achieve this objective web engineering tasks must be able to identify some useful insights for business intelligence. The proposed research work attempts to initiate Business Intelligence from Web Based Applications. The issues of research work are uniformly accommodated in five steps. The First step identification of fundamental tasks of Web Engineering, Mining for Software Engineering and its extension to Mining for Web Engineering, Web Mining and Web Based Business Intelligence Applications will be done.. The Second step recent Literature Review will be carried out. The Third Step case studies on present's business intelligence need of next generation business applications, development of Next Generation Business Intelligence Application Development Technologies (NGBIADT). The NGBIADT covers Web 2.0, Agile Modeling and Web Services. The Fourth step the idea of Mining approaches for Web Engineering and NGBIADT. The Mining approach of Web Engineering consists of 3 phases. In Phase 1, Web Engineering data such as web design, structure of Web, web logs and error reports are collected using a comprehensive algorithm. Phase 2 integrates a various Web Engineering data from phase 1 to develop web mining approach using association rules, classification and clustering techniques. In Phase 3, information generated from web mining approach can be considered as base for developing a web based application that satisfies the need of end user requirements and also satisfying needs of business industry. Thus, phase 3 produces tasks related to web application structure, web services, web architectures, web configuration management data, web classification, web communities, website Navigation and web security. These tasks are very much essential in developing NGBIADT applications. The mining approach of NGBIADT consists of technologies such as Web 2.0 for business intelligence, Agile modeling for business intelligence and Web Services for business intelligence with suitable exemplars. The Fifth Step demonstrates various insights to improve Web Engineering applications for the purpose of Business Intelligence using Secure Stock Market (SSM) web application as an example. Applying Mining strategies to Web Services will provide valuable insights in terms of Service discovery, Service dependency, Service composition etc. Hence this proposed research work investigates the following various areas of research like:

- The study on Web engineering tasks.
- Mining techniques related to web engineering
- Obtain and analyze Mining information in business intelligence
- Some insights that improve intelligence of web based application.

Keywords: web mining, web engineering, agile modeling, patterns, classification, clustering.

I. Scope Of The Proposed Problem Definition

The Research problems addressed here are: Mining Web engineering data pertaining to Program source code mining implementation tools which improves software debugging and its related challenges[1]. Strategies for debugging mining includes: Text Mining, Sequence Mining and Graph Mining.

Key research questions addressed are:

1. What types of Web Engineering Data are available to be mined?
2. Which Web Engineering Tasks can be helped using Data Mining?
3. How Data Mining techniques are used in Web Engineering?
4. What are the Challenges in applying Data Mining techniques to Web Engineering data?
5. Which Data Mining techniques are most suitable for specific types of Web Engineering data?

The importance of software is increasing in scientific research and our daily life [2]. Meanwhile, the cost and consequences of software failure caused by software bugs become more and more serious. This research emphasizes a standard process for data mining based software debugging [3]. This proposed process provides guidelines for software testing engineers and researchers on how to apply data mining techniques and software testing theory on real life software testing projects. Data mining based software debugging projects is a

five step process[4]: Establish the software testing project; data collecting, cleaning and transformation; select, train and verify the data mining models; classify, locate and describe the software bug found in previous steps; and deploy the knowledge gained into real life software testing project.

II. Objectives Of The Proposed Research Work

The objective of the research work to propose strategic Data Mining tools for program source code debugging which improves Software Reliability & Quality. The mining algorithms works on Web engineering data like text, sequences, graphs : Which improves software engineering tasks like Programming; Maintenance; Bug

Software engineers can start with either a problem driven approach, but in practice they commonly adopt a mixture of the first two steps: collecting/investigating data to mine and determining the Web Engineering tasks to assist[5]. The three remaining steps are, inorder, preprocessing data, adopting/adapting/developing a mining algorithm, and postprocessing applying mining results.

Processing data involved first extracting relevant data from the raw Web Engineering data – for example, static method call sequences or call graphs from source code, dynamic method call sequences or call graphs from execution traces, or word sequences from bug report summaries. This data is further processed by cleaning and properly formatting it for the mining algorithm[6][7]. For example, the input format for sequence data can be a sequence database where each sequence is a series of events.

The next step produces a mining algorithm and its supporting tool, based on the mining requirements derived in the first two steps. In general, mining algorithms fall into four main categories:

Frequent Pattern Mining: Finding commonly occurring patterns. Pattern Matching: Finding data instances for given patterns. Clustering – Grouping data into clusters[8] Classification- Predicting labels of data based on the already labeled data.

The final step transforms the mining algorithm results in to an appropriate format required to assist the Web Engineering task. For example, in the preprocessing step, a software engineer replaces each distinct method call with a unique symbol in the sequence data base being fed on to the mining algorithm. The mining algorithm then characterizes a frequent pattern with these symbols. In post processing, the engineer changes each symbol back to the corresponding method call[9]. When applying frequent pattern mining, this step also includes finding locations that match a mined pattern – for example, to assist in programming or maintenance – and finding locations that violate a mined pattern – for example, to assist in bug detection.

Neglected conditions are an important but difficult-to-find class of software defects. This approach presents a novel approach for revealing neglected conditions that integrates static program analysis and advanced data mining techniques to discover implicit conditional rules in the novel approach for revealing neglected conditions that integrates static program analysis[10].

Most Software engineering (and Web Engineering) data mining studies rely on well-known, publicly available tools such as association rule mining and clustering. Such black-box reuse of mining tools may compromise the requirements unique to software engineering by fitting them to the tools undesirable features[12][13]. Further, many such tools are general purpose and should be adapted to assist the particular task at hand. However, Software engineering researchers may lack the expertise to adapt or develop mining algorithms or tools, while data mining researchers may lack the background to understand mining requirements in the software engineering domain[11]. One promise way to reduce this gap is to foster close collaborations between the software engineering community (requirement providers) and data mining community (solution providers). This research effort represents one such instance. Writing Requirements is a two way process, classified as Functional Requirements (FR) and Non-Functional Requirements (NFR) statements from Software Requirements Specification (SRS) documents. This is systematically transformed into state charts considering all relevant information. The test cases can be used for automated or manual software testing on system level[14]. A method for reduction of test suite by using mining methods there by facilitating the mining and knowledge extraction from test cases.

Plan of the Proposed Research Work

- Through extensive literature survey, related work pertaining to Mining for Software Engineering with its extension to Mining for Web Engineering will be studied.
- The present and future industry needs related to Mining for Web Engineering will be studied and documented. A universal design approach will be followed meeting most of the industry standards.
- Tools will be implemented for obtained research found outs for Mining for Web Engineering
- A case study on Web 2.0 Services Business Intelligence applications will be carried out to validate and justify research results.

References

- [1]. Tao Xie, Suresh Thummalapenta, David Io, Chao Liu, "Data Mining for Software Engineering", IEEE Computer, August 2009, pp. 55-62.
- [2]. Hamid Abdul A, Stan Jarzabek, "A Data Mining approach for detecting higher-level clones in Software", IEEE Transactions on Software Engineering, Vol. 35, No. 4, July/August 2009, pp. 497-514.
- [3]. Ivano Malavelta, Henry Muccini, Patrizio Pelliccion, Damien Andrew Tamburri, "Providing Architectural Languages and Tools Interoperability through Model Transformation Technologies", IEEE Transactions on Software Engineering, Vol. 36, No. 4, January/February 2010, pp. 119-140.
- [4]. Tao Xie, Jain Pei, Ahmed E Hassen, "Mining Software Engineering Data", IEEE 29 the International Conference on Software Engineering ICSE 07.
- [5]. Francisco P.Romero, Jose A.Olivas, MARcele Genero, Mario Piattini, "Automatic Extraction of the main terminology used in Empirical Software Engineering through Text Mining Techniques" ACM ESEM 08 pp. 357 – 358.
- [6]. Mohammed J Zaki, Christopher D Carothers, Boleslan K Szymaski, "VOGUE: A Variable Order hidden Markov Model with duration based on Frequent Sequence Mining", ACM Transactions on Knowledge Discovery from Data, Vol. 4 No.1, Article 5, January 2010.
- [7]. Francine Bermas, "Got Data? A guide to data preservation in the Information Age", Communications of the ACM, December 2008 Vol 51, No.12, pp. 50-56.
- [8]. Nizar R Mabroukeh, Christe I Zeize, "Using Domain Ontology for Semantic Web Usage Mining and Next Page Prediction", ACM CIKM 08 pp. 1677 – 1680.
- [9]. Tim Menzein, Gary D Boettiecher, "Smarter Software Engineering: Practical Data Mining Approaches", IEEE/NASA 27 th Annual Software Engineering
- [10]. Josh Eno, Craig W Thompson, "Generating Synthetic Data to Match Data Mining Patterns", IEEE Internet Computing May/June 2008 pp. 78 – 82.
- [11]. O.Maqbool, A Karim, H.A.Babri, Misarwar, "Reverse Engineering using Association Rules", IEEE INMIC 2004, pp. 389 -395.
- [12]. Gang Kou Yipeng, "A Standard for Data Mining based Software Debugging", IEEE 4 th International Conference on Networked Computing and advanced Information Management, pp. 149 – 152.
- [13]. Qi Wang, Bo yo, Jie Zhu, "Extract Rules from Software Engineering Quality Prediction Model based on Neural Networks", ICTAI 2004.
- [14]. Ngoavel Moha, Yann-Gael Gueheneu, Laurence Duchien, Anne-Fran Coisele
- [15]. Mew, "DÉCOR – A Method for the Specification and Detection of Code and Design
- [16]. Smells", IEEE Transactions on Software Engineering, Vol. 36, No. 4, January/February 2010, pp. 20-36.
- [17]. Ray-Yaung Chang, Andy Podgurski, Jiong Yang, "Discovering Neglected Condition in Software by Mining Dependency Graphs", IEEE Transactions on
- [18]. Software Engineering, Vol. 34, No. 5, September/October 2008, pp. 579- 596. Security is rapidly becoming a critical business concern for companies,
- [19]. Dr.D.Vasumathi, T.Murali Krishna, "A Study of Mining Software Engineering Data and Software Testing", Journal of Emerging Trends in Computing and Information Sciences, Vol 2 No 11 pp. 598 – 603 October 2011.
- [20]. Kavita Sharma, Gulshan Shrivastava, Vikas Kumar, "Web Mining: Today and Tomorrow", IEEE 971-1-4244-8679-3/11/ pp. 399 – 403 IEEE 2011.