ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

Survey on High Performance Java Script Dependency Analyzer

Neha G. Chauksey, Dr. A. S. Ghotkar

M.E Student, Department of Computer Engineering, PICT, Pune, India

Professor, Department of Computer Engineering, PICT, Pune, India

ABSTRACT: In various software development tools, it is necessary to know the dependencies amongst classes and other objects, so that the developers may get the information as to where are these dependencies currently in use, and which modules will be affected due to change in codes.
This paper aims to build a dependency analyzer that takes multiple XML documents as the input, each representing a huge repository with an ever growing list of artifacts produced by large teams in the organization. This tool is designed such that it analyzes simple and recursive dependencies by utilizing the power of database to efficiently store and search thedependencies so as to represent the result in a tree format. This tool will act as an important decision making point for approving changes including the mitigation strategy for reducing the risk of change.

KEYWORDS: Java Script recursive dependency, REST-API, XML, XML-Search

I. INTRODUCTION

In the complex world of developing software, dependencies plays a key role. As the software keeps changing, each new change to one of its elements could cause regression in its dependents. Hence it becomes very crucial from a risk mitigation perspective to not only understand the dependencies between layers but also within those layers that makes up that software. Therefore a tool for extracting these dependencies is required.

Performance is one of the biggest challenges while performing recursive dependency analysis on a large dependency tree (XML). The results of a dependency analyzer are usually cryptic and hence would not be easy to understand. Aggregating the dependencies of multiple projects and performing analysis is cumbersome. The large dependency trees are usually time consuming to parse. The recursive analysis are complicated and time consuming on large trees and requires a high performing analyzer. Presenting the results with a high performing user interface is a challenge. This dependency analyzer is designed to not only scale in terms of searching the dependencies across an ever growing list of repositories but also to display huge results with optimal performance using pagination techniques. A rich user interface that provides an easy to view dependency tree which accommodates on demand dependency information will provide a good user experience.

II. RELATED WORK

Prior research has shown that customer reported software faults are often the result of violated dependencies that are not recognized by developers implementing software. Many types of dependencies and corresponding measures have been proposed to help address this problem. The objective of this paper is to compare the relative performance ofseveral of these dependency measures as they relate to customer reported defects.

Many direct attempts to address the issues related to the paper is not found, but rather a very few research papers that tangentially address these issues are available and discussed here.

Matthias Keil and Peter Thiemann [1] used formalization and abstract interpretation to perform a type-based dependency analysis for JavaScript. Results from this approach can be used to ensure confidentiality and integrity of the data. Magnus Madsen, Benjamin Livshits and Michael Fanning [2] combined the pointer analysis and use analysis in order to obtain the analysis of Windows 8 JavaScript applications. This is performed so as to get the partial or full inference.Milos Savic, GordanaRakic, Zoran Budimac, Mirjana Ivanovic [3] showed that more precise software

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

networks can be extracted with a tool in comparison with dependency finder that provides even language-independent extraction. They managed to do it using Dependency finder to extract software networks that belongs to different programming paradigms and fuzzy parsing mechanism.

Liang Huai Yang, Mong Li Lee, Wynne Hsu, Decai Huang, Limsoon Wong [4] created a 2PXMiner. This tool was able to perform efficient and scalable query evaluation for frequent query patterns in an XML document. Kajal T. Claypool [5] used Label match Algorithm and Needleman Wunsch Algorithm to do approximate keyword search and context-specific searching providing a better precision and recall than exact keyword search. Binh Viet, Eric Pardede [6] proposed to extend the power and capability of XML with web service technologies and P2P architectures. Active XML (AXML) is extension of distributed XML databases. This was possible with the help of AXML- XML data exchange and AXML data query processing.

Tag extraction using DOM Structure, Knowledge base matching and classification when done simultaneously, average accuracy of 96.99% classification was obtained with less error rate by Krishna Murthy. A, Suresha [7]. Katalin Tunde, Janosi-Ranez, VioricaVarga and Timea Nagy [8] used conceptual lattice, XFD Mining to mine functional dependencies through formal concept analysis which analyzes XML documents. Compare and rank was another method that provides speedy and accurate search in an XML document that is for E-Commerce Applications to locate document schemata. This method was proposed by Eric Jiu-Lin Lu and Yu-Ming Jung [9].

Describing XML Stream processing problems related to text processing and tree pattern matching using process XPath expressions with automata and pushdown automata was suggested by Dan Suciu [10]. Susumu Nishimura, Keisuke Nakano [11] introduced the idea of attribute grammar composition and transformation and altSAX to develop stream transformation by giving the specifications for tree transformation. There were many limitations for memory and boundness seen while adapting this approach. Multi-query evaluation and query compaction was the technique used by Jun-Ki Min, Myung-Jae Park and Chin-Wan Chung [12] that supports wide class of XPath queries for tree shaped expressions, order-based and nested predicates.

As of now, there is diversified research in all the techniques and platform this paper suggests. Particularly for analyzing JavaScript files that too from an input holding relational database approach has not being executed.

III. PROPOSED SYSTEM

A. *DESIGN CONSIDERATIONS:*

This system aims to build a dependency analyzer that takes multiple XML documents as the input, each representing a huge repository with an ever growing list of artifacts produced by large teams in the organization.

This tool is designed such that it analyzes simple and recursive dependencies by utilizing the power of database to efficiently store and search the dependencies so as to represent the result in a tree format. It will help programmers, testers and managers to efficiently analyze the risk from each of their perspective so that effective mitigation strategies can be applied for each change depending upon their occurrence in the software development lifecycle.

Fig. 1 shows the architecture of the proposed Java Script Dependency Analyzer(JDA) tool.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

B. ***DESCRIPTION OF THE PROPOSED SYSTEM:***

The proposed system is modelled in two phases for getting a clear vision of the tasks the tool has to perform.

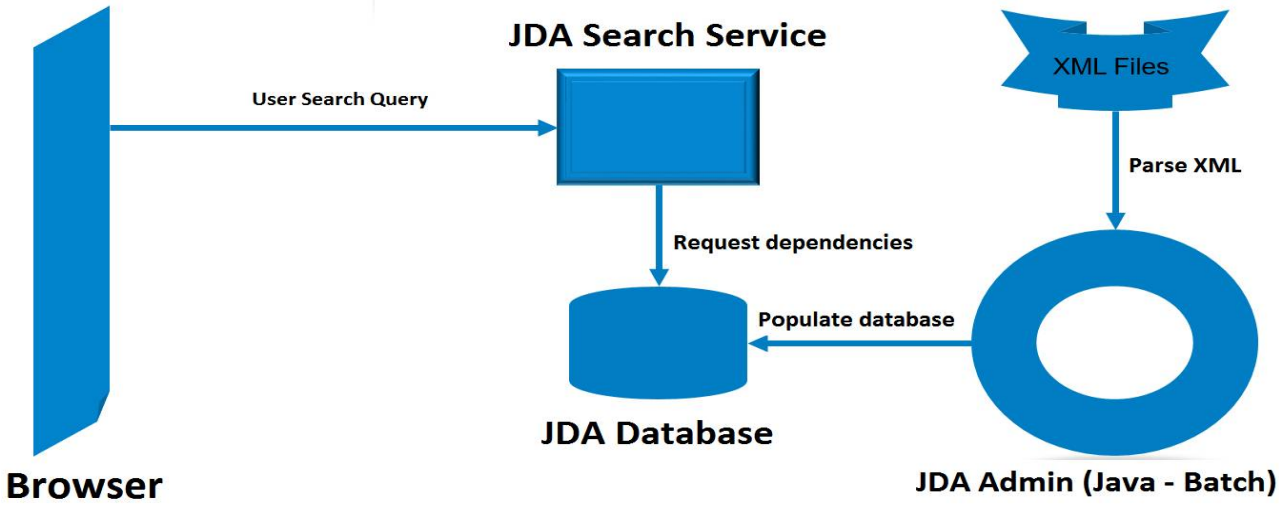

Fig:1 JDA Architecture

A. Phase I
•Evaluate different tools to handle huge XML
•Perform literature survey for gap identification
•Design and implement Rest based client for multiuser scenario
•Implement simple search through the database
•Design and implement audit trailing mechanismfor every search query

In this tool, the input is an XML file that containsfunction metadata of the Java Script files. This XMLhave details of functions i.e. where they are called from,declared and instantiated. The XML is parsed and allvalue from XML tags are stored in a JDA database.This database schema is predefined because the XMLschema is fixed. The JDA database is populated andmanaged by the following two ways:
•JDA Admin
•JDA Search Service

1) JDA Admin:
This module is a Java batch jobthat is designed to populate the JDA database. Theinput XML file is parsed with the help of StAX parser,in order to extract appropriate information of dependencies of various functions present in that particularrepository. The job of this admin module is to populatethe database not just once but in equal intervals so thatupdates in repository should be tracked.

2) JDA Search Service:
In the search service,queries are fired in terms of function name for whichthe dependencies are to be found. This module will beable to execute the queries successfully if and only ifthe JDA Admin module has performed its operation.

Phase I of this JDA tool aims to provide a verticalslicing of back-end services by getting the resultsthrough rest APIs. These APIs are designed for showing simple search results that contain the informationof dependencies. Reports of audit i.e. tracking thesearch query so as to generate a report of what theusers have searched for.

B. Phase II
•Code re-factoring
•Enhance the tool for multiple XML documents

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

•Implement recursive search through the database
•Design and implement the User Interface usingOpenUI5 libraries and components.

In this phase, the structure and implementation offirst phase will be enhanced by re-factoring the classes and codes of the project. The tool should be ableto process multiple XML files, to get the results from different versions of the repository. A recursive search is to be designed in order to get the nested dependencies among the functions. And finally the user interface using OpenUI5 libraries and components will be created to make a smoother user experience. The results are planned to be displayed in a form of tree so as to make it visible in a user readable format.

IV. CONCLUSION

The proposed Java Script dependency analyzer tool represents an approach to get the dependencies of the functions to the programmer. It is capable of handling huge and multiple XML files with lesser complexity. The proposed design includes usage of PostgreSQL relational database, which in turn produces cost effective and faster results. This kind of tool will be beneficial for the programmers to make important decisions for approving changes including the mitigation strategy for reducing the risk of change.

V. ACKNOWLEDGEMENT

I take this opportunity to express my deep sense of gratitude towards my mentor Mr. Seetharaman Venkatesan, Manager, CDD, SAS R & D India Pvt. Ltd., Pune for giving me this splendid opportunity for my dissertation work. I would like to thank my dissertation guide Dr. Archana. S. Ghotkar for her inspiration, priceless suggestions and support for this project. I wish to express my thanks to our HOD Dr. Rajesh B. Ingle for encouragement and providing me with the best facilities for my dissertation work. I thank all the staff members, for their indispensable support and for most valuable time lent as and when required. I also thank my family for their continuous support.

REFERENCES

1. Matthias Keil and Peter Thiemann, “Type-based Dependency Analysis for JavaScript”, in ACM 978-1-4503-2144-0/13/06, June 2013.
2. Magnus Madsen, Benjamin Livshits and Michael Fanning, “Practical Static Analysis of JavaScript Applications in the Presence of Frameworks and Libraries”, in Microsoft Research Technical Report, MSR- TR-2012-66, 2012.
3. Milos Savic, GordanaRakic, Zoran Budimac, Mirjana Ivanovic, “A language-independent approach to extraction of dependencies between source code entities”, in Elsevier: Information and Software Technology, 2014.
4. Liang Huai Yang, Mong Li Lee, Wynne Hsu, Decai Huang, Limsoon Wong, “Efficient Mining of frequent XML query patterns with repeating-siblings”, in Elsevier: Information and Software Technology 50 375-389, 2008..
5. Kajal T. Claypool, “SUSAX: Context-specific searching in XML documents using sequence alignment techniques”, in Elsevier: Data and Knowledge Engineering 65 177-197, 2008.
6. Binh Viet, Eric Pardede, “Active XML (AXML) research: Survey on the representation, system architecture, data exchange mechanism and query evaluation” in Elsevier: Journal of Network and Computer Applications, 2013.
7. Krishna Murthy. A, Suresha, “XML URL Classification based on their semantic structure orientation for Web Mining Applications” in Elsevier: Procedia Computer Science 46 143-150, 2015.
8. Katalin Tunde, Janosi-Ranez, VioricaVarga and Timea Nagy, “Detecting XML Functional Dependencies through Formal Concept Analysis”, in IEEE transactions, Sept-20, 2010.
9. Eric Jiu-Lin Lu and Yu-Ming Jung, “XDSearch: an efficient search engine for XML document schemata”, in PERGAMO: Expert systems with Applications 24 213-224, 2003.
10. Dan Suciu, “From searching text to querying XML streams”, in Elsevier: Journal of Discrete Algorithms 2 17-32, 2004.
11. Susumu Nishimura, Keisuke Nakano, “XML Stream transformer generation through program composition and dependency analysis” in Elsevier: Science of Computer Programming 54 257-290, 2005.
12. Jun-Ki Min, Myung-Jae Park and Chin-Wan Chung, “XTREAM: an efficient multi-query evaluation on streaming XML data”, in Elsevier: Information Sciences 177 3519-3538, 2007.

BIOGRAPHY

Neha G. Chauksey received Bachelor’s degree in Computer Science and Engineering from G. H. Raisoni institute of Engineering and technology for Women, Nagpur in 2014(email:chauksey.neha@gmail.com). She is currently pursuing Master of Engineering from SP Pune University. She is also an Intern at SAS R & D India Pvt. Ltd., Pune. Her research of interest includes Knowledge and data engineering.