International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

Development Task Based Navigation for Software Documentation

Swapnil Gangadhar Thaware[1], Dinesh Bhagwan Hanchate[2]

[1]ME Second year Student Computer Engineering, Vidya Pratishthan's College of Engineering, Baramati, Pune, India

[2] Professor, Department of Computer Engineering, Vidya Pratishthan's College of Engineering, Baramati, Pune, India

Abstract: ***In requirement gathering and analysis phase of software model, knowledge is needed by software developers and other stakeholder is captured in different forms of documentation written by different stakeholder. Developer works on project or any software, we face problem to find the right information in the right form at the right time. This paper not only helps developers but also for other stakeholder like project admin, software developer, project designer, project manager, software tester and technical support to navigate documentation. In this paper, we have proposed Development Task Extraction Navigator (DTE Navigator). Job of this tool extracts essential information needed for stakeholder and annotates it automatically.***

Keywords: software documentation; development tasks; data extraction; retrieved task; bookmark; classification.

1. Introduction

A software developer or other any stakeholder who joins an existing software development team must come up to speed on a large for varied amount of information before becoming productive. Today's all types of documentation suffers from a number of potential problems such as documentation written by people who can't write. They are unavailable, developer can't find it when they need it [9]. Today's search engines are not sufficient for enabling effective navigation of software documentation because they require stakeholder to use search terms that match the vocabulary used by the documentation writers and documentation may be in the form of softcopy of project or hardcopy. There is still a gap between the information needs for software developers and the structure of today's documentation [4]. Today's structure comes with sections and subsections, it can only be enabled effective navigation if the section headers are adequate for the information needs of developers. To overcome these issues, we introduced DTE Navigator, a task based search engine for software documentation. DTE Navigator automatically analyzes a documentation corpus (typically an online tutorial) and detects every passage that describes how to accomplish some programming task. Here document extraction is the client (in our case developer) server application where client interacts with server through the web browser. Web server (in our case apache tomcat) receives client request for document search. To send request and get response back, we designed a user interface with HTML, JSP, and SERVLET. We divided our main project with different module such as Pre-processing, Task Extraction and Task Navigation [7].

2. Literature Survey

Information extraction provides services to user who retrieves the information by firing query on the internet. But, this approach is not so effective to produce accurate results because of human involvement and poor quality of data extraction output. In requirements engineering, the state of the practice. Colin J. Neill refers to prevalent, dominant and techniques used in the software development industry [1]. These authors did literature survey on web based technology. In paper [2], Vivek D. Mohod and J.V. Megha did survey on different HTML structure based technique to scrap data from web pages in data extraction of web pages using tag tree structure. Hubert F. Hofmann and Franz Lehner discussed how deficient requirements can be single biggest cause of software project failure in Requirements Engineering (RE) as a success factor [4]. In the paper [6], TaskNav: Task-based Navigation of Software Documentation bridges the gap between documentation structure and the information needs of software developers according to Christophe Treude. Christoph proposed TaskNav, a tool that automatically discovers and indexes task descriptions in software documentation [11] for helping developers to navigate documentation.

3. Proposed System

Proposed Development Task Extraction Navigator is a user interface for search queries that suggests tasks extracted with proposed technique in an auto complete list along with code elements, concepts and section headers [5]. We have given a google access for DTE Navigator which gives an advantage of no need to open separate tab for google without exiting current tool. DTE Navigator can automatically analyze and index any documentation corpus based on a starting URL such as ignoring HTML tags. Documentation stakeholders can benefit from this tool by taking advantage of the task based navigation offered by the auto complete search interface.

4. Proposed System Architecture

In document extraction process, client is stakeholder and developer who request for results and tasks through DTE Navigator and server application where stakeholder interact with WEB server through the web browser [2]. At web server in this case apache tomcat will receive client request for document search. To send request and get response back, we designed user interface with HTML, JSP and SERVLET. We

divided our DTE Navigator with different phases such as preprocessing, Task Extraction and Task Navigation [3]. The stakeholder asks a query for document extraction, server checks keyword and phrases by extracting concepts from search query.

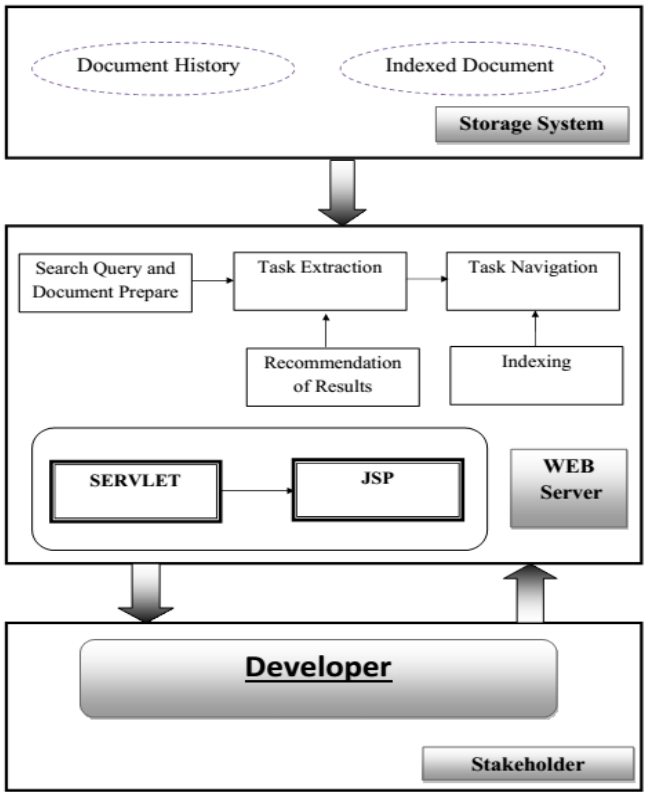


Figure 4.1: Proposed System architecture

In task extraction process, search process is followed by Bigram, Trigram and N Gram which used. Bigram checks pattern with two keyword or phrases e.g. "add link". Trigram checks the pattern with the help of three keywords or phrases. For example "what is synchronization?" N-gram checks pattern for multiple keywords as per given in [6]. Search code element considers all text as code from original HTML document. Index generation for document is preprocessed as per given in [8]. It creates tag by removing header, footer and summary from web page and converts that page into text file. Indexer will look for reference file from page history in document repository. If the reference exists in database then page is returned to the recommendation for client. In reverse engineering phase, server will extract necessary document if index exists. If it is not there then it preprocesses the document and results are achieved from google links. Task navigation phase executes algorithm for the search document with and index number. It navigates through documentation for expected page. Extraction phase will extract document from repository list as per given in [2]. It returns response with index of that page to the same server. Recommendation phase fetches document according to the index and show intelligence to navigate through documentation. Search Interface shows auto complete search for document in User Interface.

5. Development Task Extraction

In this section, we described DTE Navigator as well as the interactive auto-complete interface. There are total six menu in which project is executed.

Home : *After registring user profile and their account loging in, DTE Navigator tool will open. Then it will go for user query, developer can search using search box. It will show all related pages from google with bookmark and user can save bookmark pages.*

Task Extraction : *In second module, user have to develope task from bookmarks and code elements too.*

Upload Document : *Here, user can upload HTML files for task extraction and preprocessing will be done. All tags will be removed.*

Task Search From File : *Results are obtained in three forms i.e. Tasks, Code elements and concept. It is shown in column wise.*

Task Navigation : *DTE Navigator suggests the extracted documentation elements and the section headers from the original documentation and associates them with documents, sections, and paragraphs of the documentation.*

Search Classification and Report : *We have shown pie chart and graph chart for classification of result and report.*

i) Data set :
Data is added into database which is MYSQL to extract task to retrieve from database. We use google links and HTML pages so the developed task will be extracted.

ii) Which method is used for data collection?
We didn't use predefined dataset. Instead of that, we collect document from Google. Whenever user needs information about development task, He will enter query and collect data from Google for related query. It works not only offline but also online we can add HTML pages into tool also.

iii) Flow of project:
First user need to create a user profile. User collects task details from internet as input for searching query. User extracts task detail like task name, task result, and task concept. User store extracted task detail for future use in the database. Now, user search for existing task by entering search query. So user can collect the search details about task document from database and navigate search task from database to user of system [10].

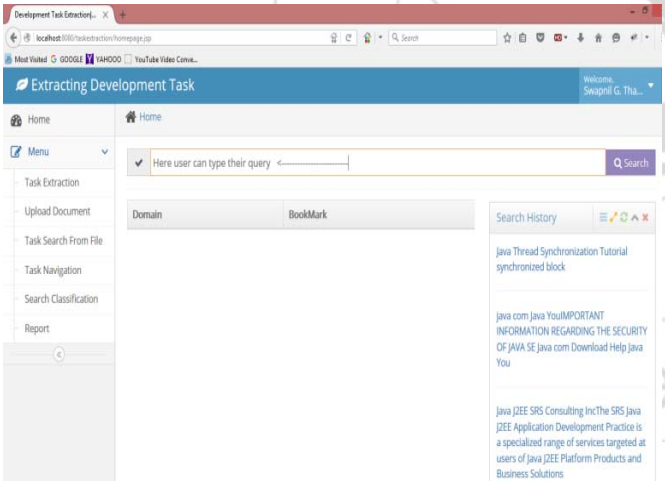
iv) Coding Flow
1) User profile creation according his role in the software development.(Index.jsp)
2) Login to system (Index.jsp)
3) Extracting new task detail from internet (Homepage.jsp)
4) Controller goes to SearchController.java
5) Algorithm for collecting data for ReverseSearching().
6) Cluster document by NaiveBayes Classification.
7) Extract task from ExtractionController.java
8) Display result from database to page.

6. Implementation

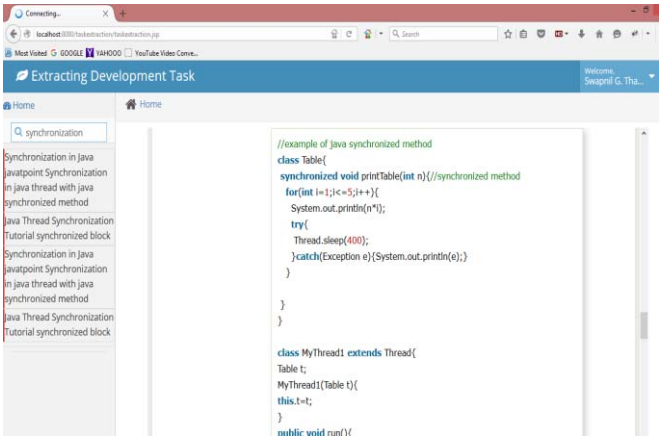
Input of this tool is a either html files or google links and output is in the form of concepts, code element and section header from DTE navigator. Task based navigation for software documentation is having three modules such as pre-processing, task extraction and task navigation. In pre-processing module, document is pre-processed first by transforming HTML files into normal text files and in this step of transformation most of the HTML mark-up is removed and stored in one list for indexing. In the next step, redundant information that is repeated on each page of the documentation, such as summaries, headers, and footers, are removed from the documents [8]. The only metadata is kept during pre-processing. In task extraction, we also use of grammatical dependencies to detect tasks in document. We need to recover the relationships between some common verbs, objects and prepositions. Use of the order of words or sentences is also insufficient as the order can be reversed, e.g., both "add module" and "module is added" refer to the same task. In third module of task navigation, we navigate document and suggest word after typing bigrams, trigrams from stakeholder. We use indexed file for navigation.

7. Results

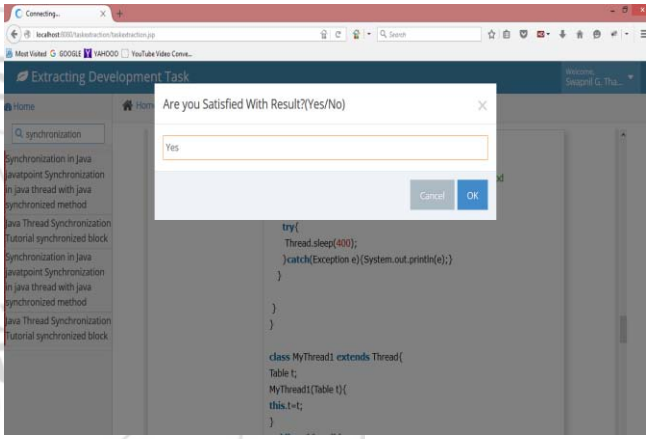
7.1 Following home page shows DTE Navigator's first page after stakeholder sign up.



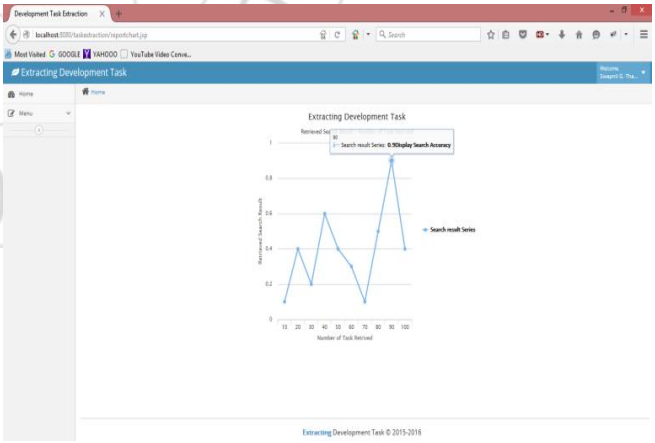
7.2 Following diagram shows extraction module for synchronization keyword and results are displayed with four tasks.



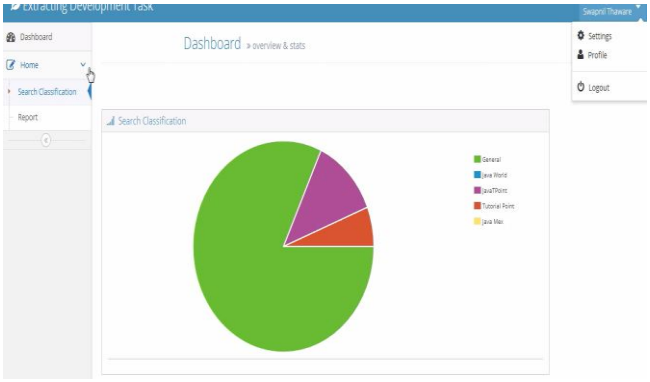
7.3 Following diagram shows pop up window after navigation of synchronization keyword and results are stored in the form of yes and no.



7.4 Following chart shows for task extraction per number of letters typed in search box.



7.5 Following pie chart shows results from domain such as general, java points, java points and social sites.



8. Accuracy of The task extraction

To evaluate the accuracy of the task extraction algorithm, we entered ten tasks in to a DTE Navigator then we determined the average precision and recall the paragraphs returned by DTE Navigator. For example, after typing synchronization characters of a task, DTE Navigator returned paragraphs with a precision of 0.40 and recall of 0.90 on an average. We decided to evaluate precision and recall after each typed character instead of each typed word. figure. 8.1 shows precision and recall after each typed character [10].

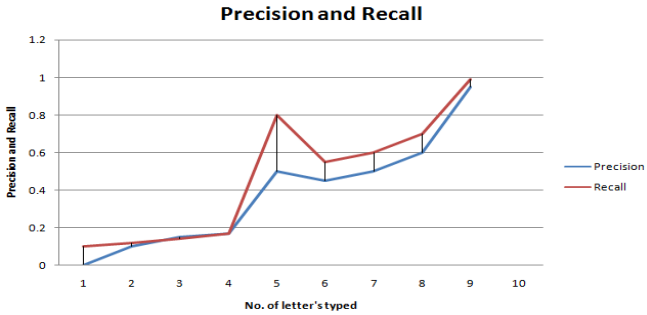


Figure 8.1: Precision and Recall Graph

8.1Methodology

We studied data which is obtained during the field study for common patterns from all developers as a participants[5]. After getting such search result, we asked "Are you satisfied with result ??" through a pop-up window, giving "yes" and "no" as answer options[3]. The pilot study of participants remarked that it would be useful to also have section titles appear as auto-complete suggestions. We added this feature for the field study and accuracy of the tool [10].

9. Conclusion

Proposed DTE Navigator is a bridge in between today's documentation structure. Using this DTE Navigator not only developer but also other stakeholder's like project admin, software developer, project designer, project manager, software tester and technical support can find the right information in the right form at the right time. In comparison with existing system, this tool is easy to use and not time consuming.

10. Acknowledgment

This paper would not have been written without the valuable advices and encouragement of Prof. D.B. Hanchate, guide of ME Dissertation work. Author's special thanks go to Prof. P. M. Patil and Prof. S. S. Nandgaonkar, Head Of Computer Department and Hon'ble principal Prof. V. U. Deshmukh, for their support and for giving an opportunity to work on software task extraction and navigation.

References

[1] C. Neill and P. Laplante, "Requirements Engineering :The State of the Practice," IEEE SOFTWARE Published by the IEEE Computer Society, pp. 40–45, Dec 1955.
[2] V. Mohod and Mrs. J. Megha, "A Survey on Data Extraction of Web Pages Using Tag Tree Structure," IJCSIT International Journal of Computer Science and Information Technologies, pp.4361-4363, Jan 2005.
[3] J. Guyette and W. Huen, "Task-List Manager – A CS2 Lab on Advanced Graphical User Interface and Data Structures," 38th ASEE/IEEE Frontiers in Education Conference, pp.11-16, Oct 2008.
[4] H. Hofmann and F. Lehner, "Requirements Engineering as a Success Factor in Software Projects,"IEEE SOFTWARE Published by the IEEE Computer Society, pp. 58–66, July 2001.
[5] G. Palshikar and R. Srivastava, "Information Extraction for Effective Knowledge Management," TCS White paper, pp. 1-14, Nov 2015.
[6] C. Treude and M. Sicard, "TaskNav: Task-based Navigation of Software Documentation," IEEE/ACM 37th IEEE International Conference on Software Engineering, pp. 649-652, June 2015.
[7] B. Lawrence and K. Wiegers, "The Top Risks of Requirements Engineering," IEEE SOFTWARE Published by the IEEE Computer Society, pp. 62–63, Dec 2001.
[8] S. Abebe and P. Tonella, "Natural Language Parsing of Program Element Names for Concept Extraction," 18th IEEE International Conference on Program Comprehension, pp.156-159 Feb. 2010.
[9] J. Bhogal and A. Macfarlane, "A review of ontology based query expansion", Elsevier Information Processing and Management, pp. 866-886 Oct 2006.
[10] B. Sumit and M. Debapriyo, "Query Suggestions in the Absence of Query Logs," ACM SIGIR, pp. 795-804 July 2011.
[11] H. Zhong and L. Zhang, "Inferring resource specifications from natural language API documentation," in Proc. 24th IEEE/ACM Int. Conf. Automated Soft. Eng., 2009, pp. 307–318.
[12] N. Wilde and M. C. Scully, "Software reconnaissance: Mapping program features to code," J. Softw. Maintenance, vol. 7, no. 1, pp. 49–62, 1995.
[13] S. Thummalapenta and T. Xie, "PARSEWeb: A programmer assistant for reusing open source code on the web," in Proc. 42nd IEEE/ACM Int. Conf. Automated Soft. Eng., 2016, pp. 204–213.
[14] J.-R. Falleri, M. Huchard, M. Lafourcade, C. Nebut, V. Prince, and M. Dao, "Automatic extraction of a WordNet-like identifier network from software," in

Proc. 67th IEEE Int. Conf. Program Comprehension, pp. 4–13, Jan 2016.

Author Profile

Swapnil G. Thaware Received B.Tech in Information Technology from Dr. B.A.T. University, Lonere in 2014. Now pursuing Master of Engineering in Computer Engineering at Vidya Pratishthans college of engineering, Baramati, Savitribai Phule Pune University.

Dinesh B. Hanchate received BE in Computer Engineering from Walchand College of Engineering, Sangli (1995), Lecturer in Gangamai College Of Engineering, Dhule (1995-96), Lecturer in S.S.V.P.S. B.S.D. College Of Engineering, Dhule In Computer IT dept. (1996-2005), M.Tech. Computer from Dr.Babasaheb Ambedkar Technological University, Lonere (2002-05), Currently Asst. Prof. Computer Engineering, in Vidya pratishthans College Of Engineering, Baramati, currently doing research at SGGS Institute of Technology and Engg, Nanded affiliated to SRTMU, under the guidance of Dr. Bichkar R.S. ,G.H. Raisoni College of Engineering and Management,Wagholi,,Pune.