RESEARCH ARTICLE OPEN ACCESS

A Review on Re-documentation Approaches and their Comparative Study

Yamini Pathania[1], Gaurav Bathla[2]
Department of Computer Science and Engineering
Chandigarh University, Mohali, 140413
Punjab-India

ABSTRACT
Redocumentaion covers the understanding of the software and record the software comprehension for easier understanding in future. So, redocumentation is the key to software maintainability. This paper describes the comparison among two approaches of redocumentation by comparing their working to regenerate the document of existing system. In Incremental Redocumentation, software comprehension is recorded in hypertext in the style of World Wide Web by using PAS tool where in Model Oriented Redocumentation; generate the documentation based on the models to bridge the gap of a legacy system and evolved system.

Keywords: - Software Re-documentation, PAS Tool, MDE, MOREDOC.

I. INTRODUCTION

Software redocumentation is one of the approaches for aiding in program understanding to support the maintenance and evolution. Redocumentation is the simplest and oldest form of reverse engineering, and can be considered to be an un-intrusive, weak form of restructuring. So, as we define that "Redocumentation is a creation or revision of a semantically equivalent representation within the same relative abstraction level" [2]. Another definition of this "redocumenting code is a transformation from code (and other documents and stakeholder knowledge) into new or updated documentation about code" [1].

Main purpose of redocumentation

- ***a.*** To make sure that the software teams understand the legacy system.
- ***b.*** To generate documentation for newly modified program.
- ***c.*** To improve the current documentation because it is hard to update the system changes when we produce the document during development of the system.
- ***d.*** To create the alternative views (e.g. dataflow, data structures, and control flow) of the system to rise understanding for human audience [2].

Incremental redocumentation approach is used to rebuild the documentation incrementally after the changes are done by the programmer. PAS tool is used in incremental redocumentation which helps to achieve the program comprehension during the change request is implemented is recorded as a hypertext in the style of World Wide Web. Partitioned annotations of software (PAS) serve as a notebook of the maintenance programmer, in which the programmer can record all of his understanding [5].

On the other hand Model Oriented REDOCumentation (MOREDOC) is used to produce models from existing systems (including program code) and to generate the documentation based on these models. Main objective of the approach is redocumentation without human interaction [4].

In the next section, we describe the incremental Redoumentation process using PAS tool. Section 3 contains the Model Oriented Redocumentation using MOREDOC framework. Section 4 describes the comparison between these two approaches on the bases of redocumentation. Section 5 contains the conclusions.

II. INCREMENTAL REDOUMENTATION

Redocumentation can be done a whole-scale effort, or it can be done incrementally as a part of the maintenance process. Incremental redocumentation approach is used to rebuild the documentation incrementally after the changes are done by the programmer. This approach is used to take out the common issues in maintaining the system is to record the changes requested by the customer or the user to occur in the source code [5].

Incremental re documentation is done as a part of the maintenance miniprocess. In first step, request for change should be collected from the customers. Now, change request should be understand by the programmer and determine the place where the customers want to make the change [1]. Programmer implements the change according to his an understanding of a particular aspect of the system and executes the change according to his understanding and also confirms the correctness of system. At the end of redocumentation, by using PAS tool the program

comprehension achieved during the change request implementation he records it in hypertext so that it is not forgotten and used in later process [5].

As the software is repeatedly changed and as additional constructs are visited, more and more documentation is increased. There is no need for up-front investment in redocumentation, it is relatively inexpensive. Hence, the incremental redocumentation naturally directs the attention of the redocumentation effort to the parts with highest wages [7].

A). PAS Tool

PAS tool is used in incremental redocumentation with hypertext. The partitioned annotations of software (PAS) serve as a notebook of the maintenance programmer, in which the programmer can record all of his understanding. PAS helps to store the information either top-down or bottom up, complete or partial, or even whether it is a conformed or tentative. PAS tools divided into three main categories: PAS browsers, PAS editors, and PAS generators. They can be combined into one tool, or either implemented separately [5].

***a. PAS browsers*:** are the tools which support browsing (i.e. World Wide Web browsers like Netscape, Explorer etc.) through PAS documentation and no necessarily need to develop specific PAS browsing tools.

b. PAS editors: (i.e. existing special World Wide Web editor such as HTML editors) support editing and updating PAS documentation. And,

c. PAS generators: analysis existing code and prepares the skeleton files for PAS documentation. Universal PAS tools combine the functions of generation, editing, and browsing in one tool.

The main benefit of PAS is using hypertext in the style of World Wide Web in which there is no need to limit the number of partitions or their contents. Each program runs in a sealed domain of application. And each application domain has an ontology which consisting the general idea and substance of that domain. So, domain partition is the main partition which is important for the understanding the program.

III. MODEL ORIENTED REDOCUMENTATION

Without documentation, the only reliable and objective information is source code. One approach in this condition is to redocument the legacy system from the source code. Model Oriented REDOCumentation (MOREDOC) is to generate models from existing systems (i.e. legacy systems) that were previously produced somehow and also used to generate the documentation based on these models, using Model Driven Engineering (MDE) technique. Since the software models can bridge the gap of a legacy system and an evolved system, also enhance the abstraction level in final documentation. So, it increases the automation in program development [2].

The main benefit of MOREDOC is to yield documentation covers all the information of system and principle of MDE used to generate the document of the legacy system from the source code.

A). MOREDOC Framework

The Model Driven Architecture (MDA) approach includes many other Technological Spaces (TS) in consistent way during software development. MDA generate from the concept of Model Driven Engineering. In initial phase, to transform the legacy system into stack of formal models, these

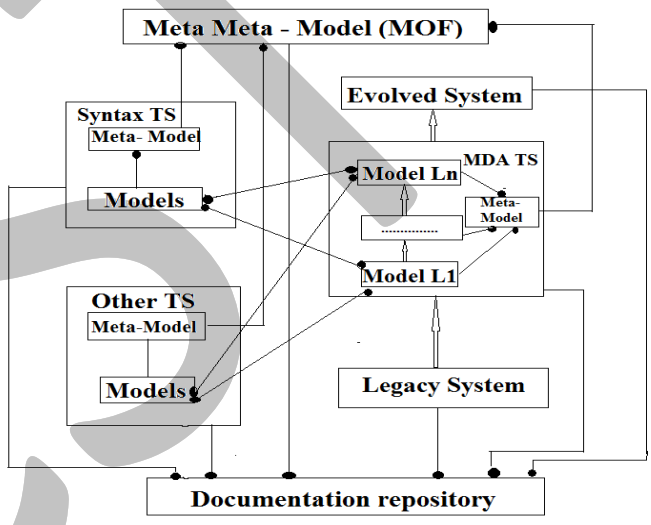

Fig.1 MOREDOC Framework

are written in formal language and again transformed into other TSs. The model information is well defined store in documentation repository which is containing from different TSs, in given Fig1. This helps to generate the structured documentation [4].

- Cross Reference of Documentation helps to understand the software structure of system as well as data and control flow and execution of events by people.
- Redocumentation on MDA TS is the process of abstraction, the high level of views of a system can be achieved without any knowledge about the all low-level detail
- To provide the multiple levels (i.e. high and low) of abstraction for every type of components WSL is used. It is the backbone of the proposed approach in which WSL as an intermediate language while legacy system's source code is translated. So, WSL support multiple language translation.

IV. THE COMPARATIVE STUDY

In MOREDOC, we generate the document by using their code and on the bases of code we generate the model. Model oriented approach is useful for system evolutionary because the exact characteristics of the original system are shown by this tool. It also shows the highest maturity level for the granularity criteria.

On the other hand Incremental approach is used to generate the document by using PAS tool. It is used to rebuild the documentation incrementally after changes are done by the programmer.

The comparisons among these two approaches are given below in Table 1*:*

	Incremental Redocumentation	**Model Oriented Redocumentation**
CASE classification:	Maintenance and redocumentation	Generate models and redoumentation.
Goal:	Improve software understanding and incrementally rebuild the document after changes are done by the programmer.	To redocument the legacy system from the source code.
Target problem domain:	Software maintenance process that involve absence or out-dated design document and to record the changes request by the user to occur in the source code.	Primarily for software redocumentation of new software system from existing source code beside redevelop the existing system again.
Tool and Technique:	Using PAS tool, helps to store the information either top-down or bottom-up, complete or partial	MOREDOC is to produce models from existing system and generate the documentation based on models which are produced by MDE technique.
Understanding aspect:	Easily understand the changes of newly modified system by its uniform structure and appearance across the whole system using PAS.	M-DOC provide the tree-like representation of structure of the system at various levels of refinement.
Re-documenting aspect:	Automatically record the incremental changes where the comprehension of the software is recorded in hypertext, in style of World Wide Web.	Automatically generate the model from source code and store the information in the repository, and then generate the final documentation, which is browser based and easy navigation for any software system.
Visualizing aspects:	Easily access the information which the programmer needs and trusts from the structure of the documentation.	Generate different levels of software abstraction and in different views of presentation but efficiency level is low, not able to search the information as needed.

Table1*:* The Summary of two Redoumentation approaches compares

V. CONCLUSION

The purpose of redocumentation to make sure the software teams understand the legacy system. Redocumentation only recovers the understanding of the software and records it, thus making future program understanding easier. The model oriented approach tries to solve the problem of abstraction; however, the efficiency level is low and not able to search the information as needed. The advantage of this approach is model based traceability, which cover system's transformation information. In incremental redocumentation, PAS proved to be a superior way to document software in style of World Wide Web. It has been used in several settings: as part of student projects in software engineering classes, as part of an ongoing research project. It also improved the maintainability of the software.

REFERENCES

[1]. Sugumaran, N., & Ibrahim, S. (2011). "An Evaluation on Software Redocumentation Approaches and Tools in Software Maintenance", Communications of the IBIMA, 2011.

[2]. Nallusamy, S., & Ibrahim, S. "A Review of Redocumentation Approaches".

[3]. Nallusamy, S., Ibrahim, S., & Mahrin, M. N. R. "A Software Redocumentation Process Using Ontology Based Approach in Software Maintenance."

[4]. Chen, F., & Yang, H. (2007, July). "Model oriented evolutionary redocumentation." In Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International (Vol. 1, pp. 543-548), IEEE.

[5]. Vaclav Rajlich, "Incremental Redocumentation with Hypertext", IEEE 1997.

[6]. S. Souza, N. Anquetil and K. Oliverira, *"A Study of the Documentation Essential to Software Maintenance",* 23rd Annual International Conference on Design of Communication (SIGDOC'O5), Sep. 2005, pp. 68-75.

[7]. Rajlich, V. (2000). "Incremental Redocumentation Using the Web," Software IEEE, 17(5), 102-106.

[8]. P. Brown, "Integrated Hypertext and Program Understanding Tools," IBM System Journal, Vol. 30, No. 3 1991.

[9]. N. Fletton and M. Munro, "*Redocumenting Software Systems Using Hypertext Technology",* 1988 Conferences on Software Maintenance (CSM'88), Oct. 1988, pp. 54-59.