ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 7, July 2016

Service Identification in Cloud Using Non Functional Requirements

Rashmi Phalnikar
Dept. of Information Technology,
MIT College of Engineering
Pune, India
rashmi.phalnikar@gmail.com

Abstract— **Cloud computing that depends particularly on the Infrastructure as a Services model, allows the system administrators to identify resources or services to deploy their applications. This scenario however requires different expertise to identify the appropriate resources that the model requires to run in the cloud as the specifications are often vague.**

The approximation on identification of the resources can vary depending on the cloud provider characteristics. Selection of the most appropriate provider for a particular application is also a laborious task because there are a huge number of services offered by considerable number of providers that are not directly comparable and have similar functionality. In order to assist system administrators in this, a new framework is suggested that supports the process of deploying applications in cloud providers, using a comparison of high level requirements predominantly non-functional requirements (NFR). Experiments have already been conducted successfully with a small scale prototype where matching of user constraints during service selection process is done. The experiment uses Aspect Oriented Paradigm (AOP), XML and graph transformation. The relevance of the framework is illustrated using the Remote Patient Monitoring (RPM) scenario. The results demonstrate that the model considerably improves the process by reducing conflicting non functional requirements. Based on this work done, the next logical step is to extend it to Cloud Computing.

Keywords—**Cloud Computing, User constraints, NFR Conflicts.**

I. INTRODUCTION

Cloud computing is a remarkable technology that changed the way the services are acquired. It is no longer necessary to make investment in infrastructure and resources management, as there are significant numbers of cloud providers that can offer a solution for specific requirements. On the other hand, administrators working in cloud computing require new skills to perform tasks that includes judgment of application resources, the selection of service providers, and later the acquisition of resources in the selected clouds. The tasks mentioned before are complex due to the large number of variables that have to be considered to achieve the right balance between the performance requirements and non functional requirements (NFR).

The increasing demand for user satisfaction and the growing number of applications providing similar functionality are driving researchers to investigate ways to understand and accurately specify the NFRs and incorporate them into the design of software systems. NFRs impose user constraints that are a boundary to comply with and are addressed side by side with its linked functional scope. Incorporating the constraints into the design process is not a simple task. Dependency among NFR is one of the major issues to handle for delivering quality software. Conflicts can arise due to different views and priorities of the stakeholder on the system.

A particular complicated phase during the development of an application in Cloud Environment is the requirement engineering phase and it is proven that most failures in software applications are due to errors in the requirements and design phases. If the NFR are not properly taken into account, they are likely to conflict and generate the most complex of errors that are difficult to correct once the system is completed.

Various methods have been suggested to provide methods to deal with these aspects concerning NFR in the requirements engineering practices for Web applications. However, what is required is a complete solution that considers NFRs from the beginning of development of the Web application, in order to assure that the user constraints are satisfied. The above

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 7, July 2016

discussion sets the stage for the framework which has been already suggested in earlier work [1][2] and is proposed to be extended to cloud application development.

II. CLOUD COMPUTING AND SOFTWARE ENGINEERING ISSUES:

A cloud is a dynamic provision of computing services or resource in a coordinated fashion. The main concern found in current cloud computing framework is that it is primarily resource-centric and hence does not understand in depth the non-technological service-centric characteristic. This overlooks the aspects of system design that plays an significant role in user acceptance. It is a firm belief that cloud computing reference architecture should include the key players responsible for the cloud evolution, and more importantly, identify within their roles the non technological service-centric aspects as well.

The current needs for control of data location and handling of data in clouds have resulted in interoperability, privacy and security issues that impact on data migration. Current systems based on traditional software engineering principles and service-oriented architecture (SOA) requires suitable adaptations due to multi-tenancy concept of the cloud as compared to single ownership of existing systems. Programming models need to be reframed to include non technological user centric issues. Certain automatic capabilities for system development should be introduced to achieve efficiency in improved cloud adoption. These attribute pose challenges in adapting traditional software engineering design, implementation and testing methodologies into the cloud paradigm.

All participants in cloud architecture should be responsible in applying the software engineering principles during the system analysis and design in addressing issues that would cover dynamic location handling of the cloud, data protection and governance policies. Currently, the cloud architecture lacks appropriate software engineering approach to address the nonfunctional process and management issues that are user centric. Existing methods do not provide for other non-technological aspects or are not given the deserving and explicit importance. The proposed method discussed below address this issue.

To sum it, two main issues arise in adopting cloud, one - the lack of functional migration of systems and the lack of non-functional user-centric architecture.

III. THE FRAMEWORK

In order to get a complete understanding of the way user constraints in a user driven application can interfere with the system, our work proposes a framework that enables the designer to select the best possible solution for the user needs. Based on literature survey, major steps have been identified in the framework. The input to the system is the user's requirements in natural language. The output of the framework provides the implementation of aspects along with a detailed, formal and declarative analysis of NFR conflicts [7]. The goal is to satisfy the user's personalized requirements so as to derive best performance from the application in an SOA environment. Also, to derive the best performance in these critical but pervasive systems, the compatibility of NFR has to be considered at design time itself.

The block diagram of the framework is given below in Fig.1:

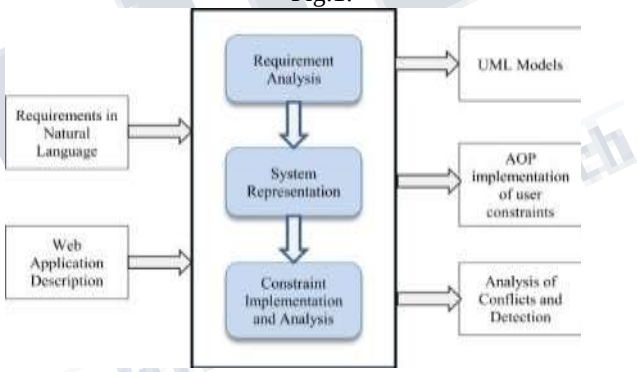

Fig. 1: Block Diagram of Proposed Framework

The framework includes three modules:

- Requirement Analysis
- System Representation
- Constraint Implementation and Analysis

The three modules are described briefly:

3.1 Requirement Analysis:

A simple method is used to analyze natural language application specification and to extract the required information from the text. This part is inspired by work by Ibrahim [3]. After analysis and extraction of associated information, the method draws UML use case diagrams and class diagram. The main reason to include this in framework is to understand the functional aspect of the application.

3.2 System Representation:

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 7, July 2016

After the specifications are illustrated using UML diagram, a method is devised that can analyze the effect of user constraints on the functioning of the application. Use of Graph transformation; which is a formal method for model transformation is suggested. Once an appropriate method was selected, it is needed to represent the system application in a graph readable format. The behavior of the application is modeled using a variant of the UML diagram - a type graph from graph transformation theory. A class diagram can thus be represented by a type graph called GXL (Graph Exchange Language) [4]. However what is required is to convert the system description in XML into GXL (Graph exchange Language) to make it suitable for Graph Transformation.

3.3 Constraint Implementation and Analysis

Use of Aspect Oriented (AO) methodology [5] is explored for the design of constraints and realized it can handle interleaved concerns to a certain level. AOP [6] entails breaking down program logic into distinct parts called concerns. Further requirement is the analysis of rules for interdependencies using Graph Transformation in Algebraic Graph Grammar (AGG) tool for Graph Transformation.

The Remote Patient Monitoring (RPM) is the case study for the framework. RPM refers to a wide variety of technologies designed to manage and monitor a range of health conditions of a patient monitored remotely. The emphasis is on user's roles and expectations in terms of care personalized to their needs, which are nonintrusive and operate seamlessly within the environment and thus classified as NFR. These requirements are crucial for a superior quality of service. The suggested approach should deliver a validated conceptual model of the user environment, identifying stakeholders, their interests, responsibilities and relationships within the relevant care-giving and care-receiving processes.

IV. EVALUATION RESULTS

The Algebraic Graph Grammar – AGG [9] environment is designed as a tool to edit directed, typed and attributed graphs and to define a graph grammar (i.e. a start (host) graph plus a set of transformation rules) as input for the graph transformation engine of the system. After graph grammar is specified the following steps are performed in AGG as given in Fig.2:

a. To define and validate a set of transformation rules.
b. To perform transformations of the host graph.
c. To analyze the specified graph transformation system.

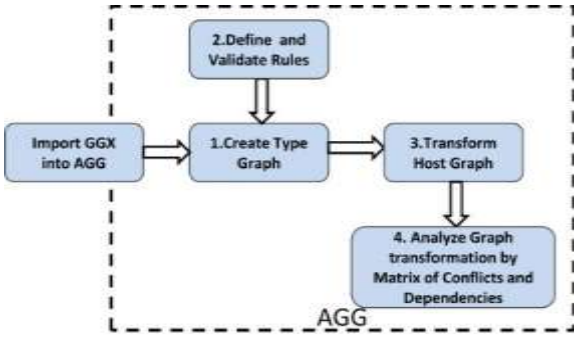

Fig.2: Steps During Graph Transformation In AGG

Create Type Graph: The type graph or the instance of class diagram defines all possible node and edge types, their possible interconnections and all attribute types. The type graph is an instance of the class diagram (System representation in GGX). Fig. 6 shows the type graph for RPM system.

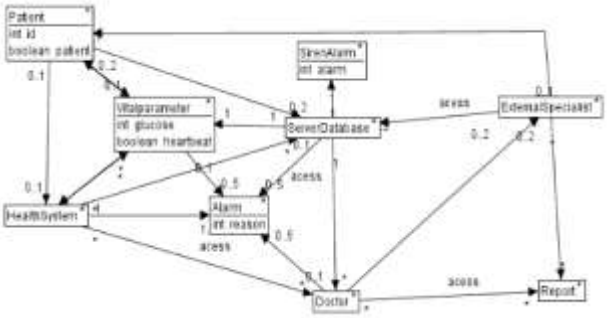

Fig.3: Type Graph of the System

Define Rules: Due to the limitation of space only some of the rules are implemented. Rules are defined that include both functional and user constraints as given below :

Rule for Patient Login: If patient is authorized then and then only data is entered into the health system as shown in Fig. 4.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 7, July 2016

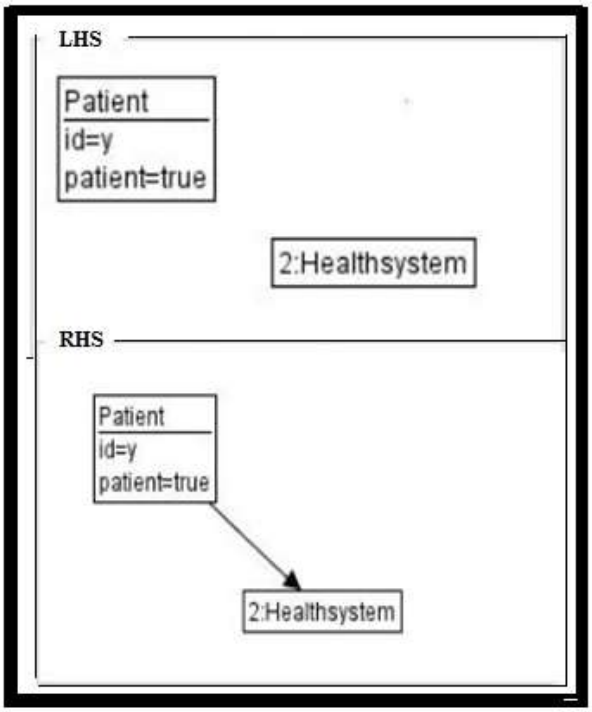

Fig. 4: Rule for Patient Login

Rule for accessing Report: The generated report is
Added to the server database as shown in fig. 5

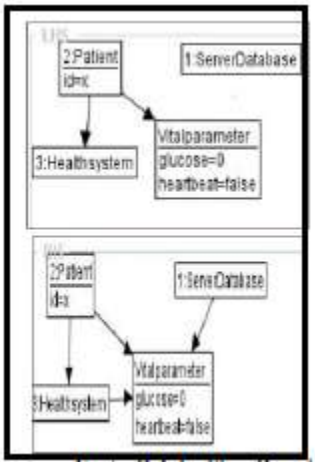

Fig. 5: Rule for Store Report

iii. Rule for adding report: Report generated as per
DOCTOR'S PRESCRIPTION IS ADDED.

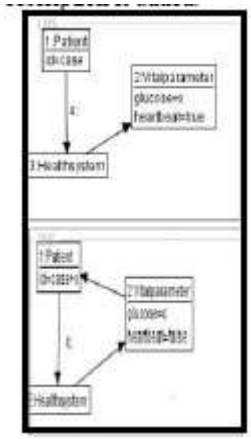

Fig. 6: rule for add report

On the application of all rules, the instance of the type graph is generated as shown in Fig. 7 below.

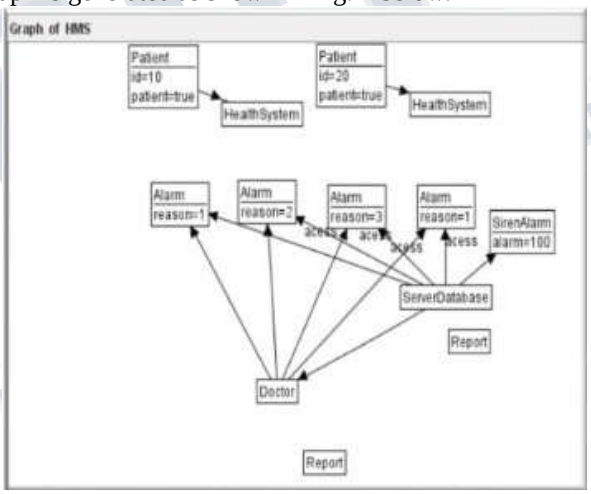

Fig.7: Graph for the type graph

Transform: This allows performing an in-place graph transformation. By default, the rules given by a graph grammar are applied non-deterministically. Rules to be applied and their matches are chosen randomly. atrix of Conflicts: Using this method in AGG, a matrix is derived to identify conflicts for the seven given

Constraints given below in fig. 8

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 7, July 2016

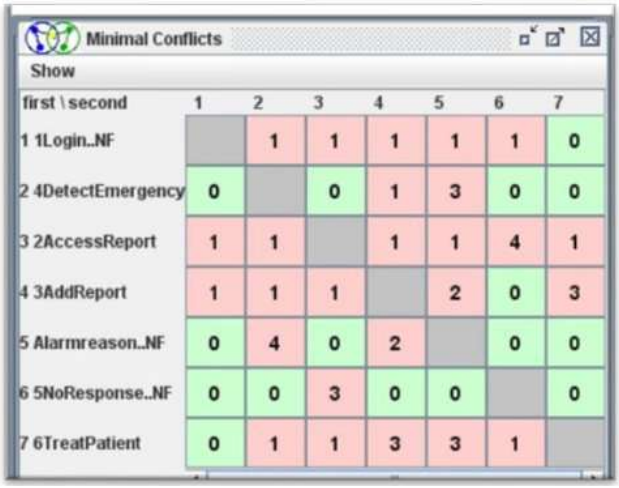

Fig. 8: Matrix of minimal conflicts

v. Matrix of Dependencies: Similarly the matrix of dependencies is as shown in Fig.9:

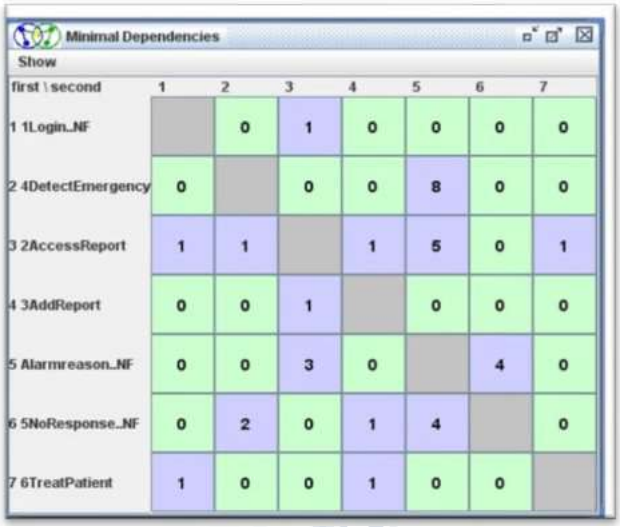

Fig. 9: Matrix of Dependencies

Critical Pair Analysis (CPA Graph): The computation of conflicts and dependencies is based on the idea of critical pair analysis which is known from term rewriting. Critical pairs formalize the idea of a minimal example of a conflicting situation. Fig. 10 shows the critical pairs for the case study.

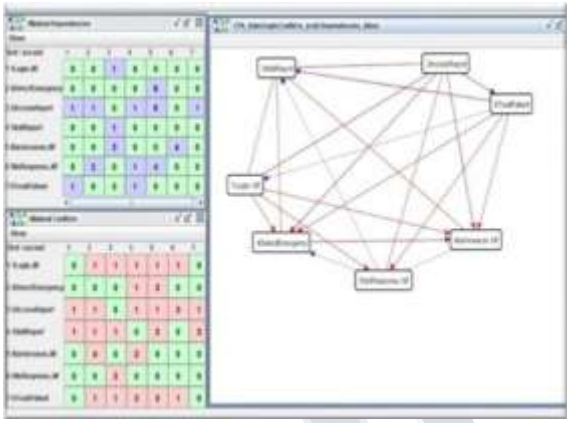

Fig. 10: Critical Pair analysis (CPA)

V. DISCUSSION AND FUTURE WORK

The goal of the research work was to develop a framework that could aid the software designer to understand the suitability of particular system w.r.t user constraints. The contribution lies in the realm of a noncompulsory alternative proposals as opposed to heavyweight ones. The point is that in addition to methods that are used for NFR in web service, an addition of such a framework will ensure that the system provides user satisfaction and also provide effective results. The results obtained from both this study for RPM are very positive and of great importance. It surely points out that the use of such a method leads to less effort in error correction and a lower cost during the maintenance phase.

In this paper, the issues of software engineering in cloud computing have been highlighted. Non-functional user requirement issues such as privacy, and system's functional migration bottlenecks due to interoperability and other migration issues play a major role in cloud adoption. It is argued that present cloud computing issues require prime attention from software engineering theory and architecture perspectives for a sustained cloud adoption.

Cloud user-roles and traditional software engineering principles and models require re-thinking of the system architecture, design, implementation, testing and maintenance in engineering the cloud services.

A framework is recommended to alleviate the current situation. Since it is already proven for a single service, tits application in Cloud environment would lead towards refining the roadmap further. The existing work will provide an initial framework and more research in this direction would evolve into an interesting solution.

REFERENCES

1) Rashmi Phalnikar, Devesh Jinwala,"Analysis of Conflicting User Requirements in Web Applications Using Graph Transformation", ACM Special Interest Group on , Software Engineering (SIGSOFT) January 2015

2) Rashmi Phalnikar and Devesh Jinwala "Framework For Analysis Of Complex User Requirements In RPM System Using Aspect Oriented Use Case Method And Graph Theory", Poster presentation at IRISS ACM Event 9th

3) Inter-Research-Institute Student Seminar in Computer Science Feb 5, 2015, Goa (co-located with the ACM India Annual event 2015), Goa , 5-7 February 2015.

4) Mohd Ibrahim, Rodina Ahmad "Class diagram extraction from textual requirements using Natural language processing (NLP) techniques", *In Proc. Of Second International Conference on Computer Research and Development, IEEE Journal*, 2010 , Pp. 200-209.

5) Richard C. Holt and Andreas Winter, "A Short Introduction to the GXL Software Exchange Format", In *Proc. of the Seventh Working Conference on Reverse Engineering – WCRE* , IEEE Computer Society, Washington, DC, USA,200 Pp. 299-. 306.

6) Clarke S., Baniassad E.: *Aspect-Oriented Analysis and Design: The Theme Approach*, Addison Wesley, Boston (2005).

7) Kiczales, Gregor, "Aspect-oriented programming *"Springer Berlin Heidelberg*, 1997

8) Chung, Lawrence, and Julio Cesar Sampaio do Prado Leite. "On non-functional requirements in software engineering." In *Conceptual modeling: Foundations and applications. Springer Berlin Heidelberg*, 2009. Pp. 363-379

9) Lu, C. and Song, I. 2008. A Comprehensive Aspect- Oriented Use Case Method for Modeling Complex Business Requirements. In *Proceedings of the ER 2008 Workshops (Cmlsa, Ecdm, Fp-Uml, M2as, Rigim, Secogis, Wism) on Advances in Conceptual Modeling: Challenges and Opportunities* (Barcelona, Spain, October 20 - 23, 2008).

10) Hartmut Ehrig, "Fundamentals of algebraic graph transformation" *Springer Berlin Heidelberg* , Vol. 373, 2006