



Various Methods for Inconsistency Check in Distributed SRS- A Review

Shivangi Bhardwaj

CSE, Amity University, India
shivangibhardwaj28@gmail.com

Abstract- Distributed Software Requirement Specification is important in day to day world as data has to be taken from different places and not located in one place. So requirement gathering is an important aspect so as to maintain consistency in that environment. The paper aims to discuss various techniques used for checking consistency in a distributed software system.

Keywords— Software Requirement Specification (SRS), Inconsistency, Distributed System

I. INTRODUCTION

Software Requirement Specification is a document which requires all functionality for a system to be developed. It covers some of the aspects such as performance, design, quality, and many more. Due to globalization of market and effects of economy, we require Distributed Software Requirement Systems which is helpful for stakeholders. Different stakeholders will need different data from different storages. SRS should be correct, unambiguous, complete, consistent, verifiable, modifiable, traceable.



Fig. 1 Features of SRS

If the above mentioned qualities are not there that means a system is having faults due to inconsistency, incompleteness, ambiguity in the requirement stage are more expensive and difficult to correct than other faults that occur in later stages of system development.

Requirement Engineering is the most significant phase of software development. So that consistency of the software is maintained while doing the SRS.

Inconsistency is the issue that the paper intends to highlight, which means when there is disagreement between two aspects.

II. METHODS FOR INCONSISTENCY CHECKING

A. Heat Map Representation

Heat map representation is used for graphical representation of data based on frequency usage. Categorization is done on the basis of different shades of orange colour.

Higher Value \longrightarrow Darker colour of square

Secondly representation of models used as semi-formal specification for requirements. Your paper must be in single column format.

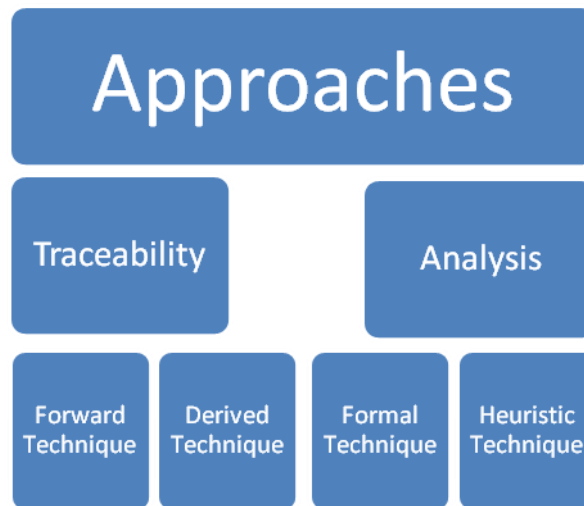


Fig. 2 Model for Requirement Specification

Partial consistency checking focuses either only on the consistency of the natural language requirement or the models, or consistency between the natural language requirement and the models [1].

B. Model Based Approach

A. Felfernig proposed a of model-based diagnosis effective consistency management in different requirements engineering scenarios, such as open source projects or large and geographically/organizationally distributed software projects [2]. From the given preferences, the minimal set of preferences is chosen for consistency.

It is helpful for distributed groups of stakeholders in the requirement specification.

C. Consistency Checker Tool:

To check requirement consistency CCT (i.e. Consistency Checker Tool) is designed [4]. It is tested on standalone system by taking requirement from different users. By changing colour of the requirements inconsistency can be detected.

D. GenSRS Tool

It generates the requirements automatically. It uses sequence diagrams and use case diagrams in it. This helps in removing inconsistency at early stages. So consistency between diagrams is maintained and the text is saved in TXT. File [12].

E. Web Based Technology

Requirement gathering is used for semantic web based technology for consistency management. Example of case tracking system for law enforcement agency is taken into account [5].

F. CAMLE Model

Lijun Shan investigates the consistency check in the modelling of multi-agent systems (MAS) [6]. CAMLE Models are used for automatic checking of consistency. (MAS) is based on three types of modes which are caste models, collaboration models, behaviour models. And it also detects which modelling environment suits best for what kind of consistency.

G. Hierarchical Tree Approach

This paper presents consistency analysis based on 3 types of consistencies and removing the identified inconsistencies for example distributed system. Hierarchical tree approach is used for C&C analysis which is performed based on the constructed tree and inter-relations among condition guards and associated actions [7].

H. Context Based

In this paper heterogeneous to single model i.e. stakeholders, consistency management is discussed. On the basis of context, inconsistency management can be done by requirement analyst [11]. As the requirement as a whole set in distributed environment is sometimes not taken into account generally.

III. CONCLUSION

Growth of businesses and organizations have established new paradigms of collaborative and distributed working environment. Distributed software requirement specification has become an area of interest for researchers in recent years. Different researchers have proposed different techniques for eliminating inconsistencies of software requirement specifications acquired from geographically distributed locations. This paper reviews all the proposed and efficient techniques for establishing consistency in software requirements specification in distributed scenario.

REFERENCES

- [1] Massila Kamalrudin and Safiah Sidek, "A Review on Software Requirements Validation and Consistency Management," *International Journal of Software Engineering and Its Applications*, Vol. 9, No. 10, pp. 39-58, 2015.
- [2] A. Felfernig, M.Schubert, M. Mandl and P. Ghirardini, "Diagnosing Inconsistent Requirements Preferences in Distributed Software Projects," *3rd International Workshop on Social Software Engineering*, pages 1–8, Paderborn, Germany, 2010.
- [3] Bashar Nuseibeh, Steve Easterbrook and Alessandra Russo, "Making Inconsistency Respectable in Software Development," *Journal of Systems and Software*, vol. 58, pp. 171-180, 2001.
- [4] Neha Bhateja1, Anshul Chugh2, "Consistency Handling In Distributed Requirement Engineering"; *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 4, Issue 12, December 2014.
- [5] Zerong Chen and Aditya Ghose, "Web Agents for Requirements Consistency Management," *Proceedings of IEEE/WIC International Conference on Web Intelligence*, 2003.
- [6] Lijun Shan and Hong Zhu, "Consistency Check in Modelling Multi-Agent Systems," *Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC'04)*, 2004.
- [7] Lian Yu1, Shuang Su, Shan Luo, Yu Su, "Completeness and Consistency Analysis on Requirements of Distributed Event-Driven Systems," *2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering*, 2008.
- [8] Yong Piao, Xiukun Wang and Zhendong Shang, "Integrated Consistency Constraints Checking in a Complicated Development Environment," *International Symposium on Computational Intelligence and Design*, 2008.
- [9] Ivo Krka and Nenad Medvidovic, "Distributing Refinements of a System-Level Partial Behavior Model," *IEEE Explore*, 2013.
- [10] Irit Hadar and Anna Zamansky, "Cognitive Factors in Inconsistency Management," *IEEE Explore*, 2015.
- [11] Gilles Perrouin, Erwan Brottier, Benoit Baudry and Yves Le Traon, "Composing Models for Detecting Inconsistencies: A Requirements Engineering Perspective," pp. 89103, *Springer*, 2009.

- [12] Mohammed Riyad Abdullah and Rosziati Ibrahim, "GenSRS: Generation of Software Requirements Specification From Use case diagram and Sequence Diagram," *Proceedings of the Second Intl. Conf. on Advances in Computer and Information Technology -- ACIT 2013*.
- [13] Kedian Mu_ and Weiru Liu, "Handling Inconsistency In Distributed Software Requirements Specifications Based On Prioritized Merging," pp. 631–670, *Fundamental Inform.*, 2009.