www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 6 June, 2014 Page No. 6729-6734

Fault Removal Effectiveness On Software Requirements Specification Using Non-Preemptive Relocation Technique

R.Saranya
Rensearch Scholar
Madurai Kamaraj University Madurai-21.
sharushara@gmail.com

ABSTRACT: ***Software specification is defined as the requirements that facilitates to achieve the specified organizational objective without any faults. Existing Model-based Oracle Software Generation (MOG) method capture the types of faults and define an automatically generated partial, passive oracle from the agent design models. MOG method not yet developed the framework for removing (i.e.,) preventing the fault at the initial stage of the software requirement specification. Present systematic review of requirements specifications from software engineering model fails to integrate the business requirements with the software functional model. To prevent the fault on the initial stage of the software requirement specification, Prior Fault Removal method using the Non-preemptive Relocation (PFR-NR) technique is developed. Prior Fault removal assures the definite software requirement specification without any faults. The functional model in PFR-NR technique refers to the set of software functions which offers a reliable fault free solution while performing the execution process. Non-preemptive Relocation performs the monitoring using the Reactive loop manage mechanism. The looping mechanism performs the integration between the business requirements and software functional model in the non-preemptive relocation technique. Non-preemptive relocation provides 12 % improved result to the customers. PFR-NR technique is measured in terms of Software fault prevention level, Customers satisfaction point, software specification overhead rate, true positive business requirements and software function integration time.***

Keywords: Software Requirements, A-Prior, Fault Removal, Non-preemptive Relocation

1. INTRODUCTION

The software development process is still an undevelopedaction with the several critical issues that compromise its victory. Its main cause derives from the non acceptance of a systematic approach founded on the best practices proclaimed by the software engineering community.Knowledge Discovery in Databases (KDD) as illustrated in [4]classifiesmashupcandidates to handle all the critical process.Lightweight syntactical approaches and larger semantic analysismixture is not developed on the business requirement specification.

Requirements Specification engineering is software process which classify the software requirements for the business development model. Controlled Normal language for requirements specification in [7] search for a higher point of inflexibility and quality of business requirements specifications. The concepts demonstrate its respective tool support but business language's extensibility and reuse mechanisms is not carried out effectively.

Design Meta-Model govern thetransformation during the source process in [5] to identify the faults. The process finally determines the appropriaterule to be applied for each concept andfail to generate a compositestate in the target model.

Model-driven engineering with the majorobject in [6] developed an evolvingapproach to address system complexity by thestrong use of models.The further relationship between the systems is not investigated well. Fault prediction research area and machine learning algorithms as demonstrated in [8] performs the effective prediction of faults.

Model-based Oracle Software Generation (MOG) methods as described in [1] captureall types of faults and define an automatically generated system.MOG method not yet developed the framework for preventing the fault at the initial stage of the software requirement specification.

In this work, focus is made on developing an effective integration system with the business requirement and the software functional model.PFR-NR initially covers the Non-preemptive Relocation work for locating and relocating the business requirements through the resource manager. Reactive loop manage mechanism plays a vital role while integrating the business requirements and software functional model in the non-preemptive relocation technique.

The structure of this paper is as follows. In Section 1, describes the basic problems insoftware requirement specification. In Section 2, present an overall view of the Prior Fault Removal method using the PFR-NR technique.Section 3 and 4 outline experiment results with parametric factors and present the result graph for research on software functional model. Finally, Section 5 demonstrates the related work and Section 6 concludes the work.

2. PFR-NR TECHNIQUE

The main objective of the prior fault removal method is to integrate the business requirement and the software function model without any fault occurrence on the software requirement specification. The architecture diagram of the Prior Fault Removal method using the PFR-NR technique is shown in Fig 1.

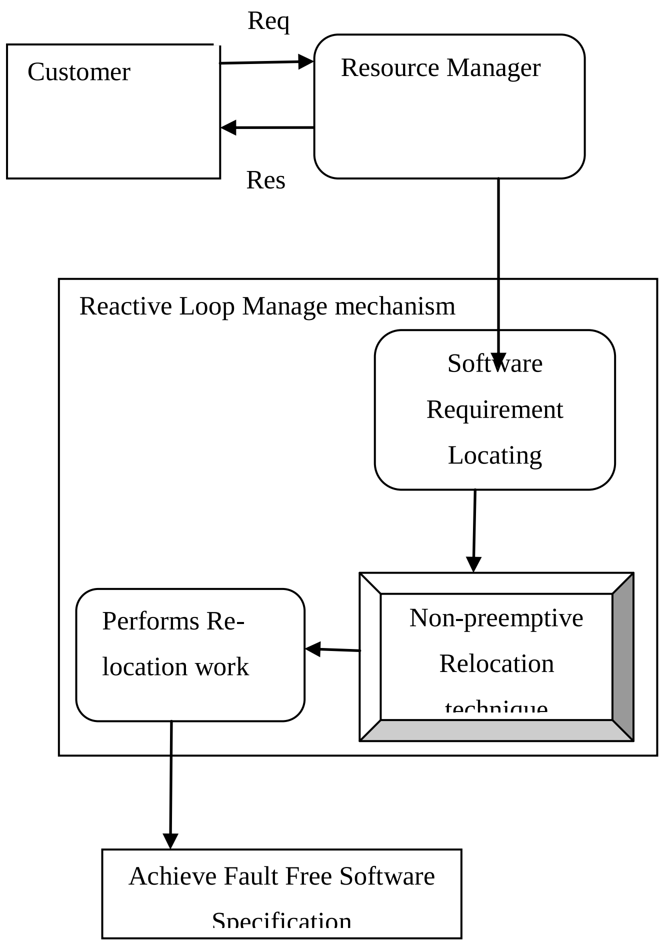
Req
Customer
Resource Manager
Res
Reactive Loop Manage mechanism
Software Requirement Locating
Performs Re-location work
Non-preemptive Relocation technique
Achieve Fault Free Software Specification

Fig 1 Architecture Diagram of PFR-NR Technique

As illustrated in Fig 1, overall framework is described to have a fault free software requirement specification. The customer specifies the list of requirements (i.e.) request send to the administrator. Prior Fault Removal method avoids the knowledgeable failure through protective measures. The preventive measure is carried out using the Reactive Loop Manage mechanism.

The Reactive Loop Manage mechanism performs anticipatoryreconfiguration of the software requirements based on the business person requirement and possibly graceful degradation is avoided. Reactive Loop Manage mechanism contains the reliable system without any fault. The reactive loop is formed by the continuous execution of the system to achieve the fault free software specification system. Relocation evicts in PFR-NR technique provide an effective system with the minimal integration time. Application relocation is finished before the expected customer result is produced to the customer side from the administrator. The quality of software service is attained in thereactive loop of PFR-NR technique.

The looping mechanism performs the integration between the business requirements and software functional model in the non-preemptive relocation technique. Non-preemptive relocation moves the business requirements to the software model framework and executes the customer requirements from the stopped execution points by providing the effective result to the customers.

2.1Non-preemptive Relocation technique

In PFR-NR technique, relocation is carried out based on the software propertiesto avoid the faults with the specified business requirements.The minimal overhead in the PFR-NR technique is very easy work with the business requirements and the software functional model without any faults.

Non-preemptive relocates all the customer requirements depending on the resource paths with effective non-preemptive (i.e.,) non-interrupted process for each customer and satisfy their needs by removing the fault using the Prior Fault Removal method.The flow process of allocation and reallocation procedure in PFR-NR technique is shown in Fig 2.

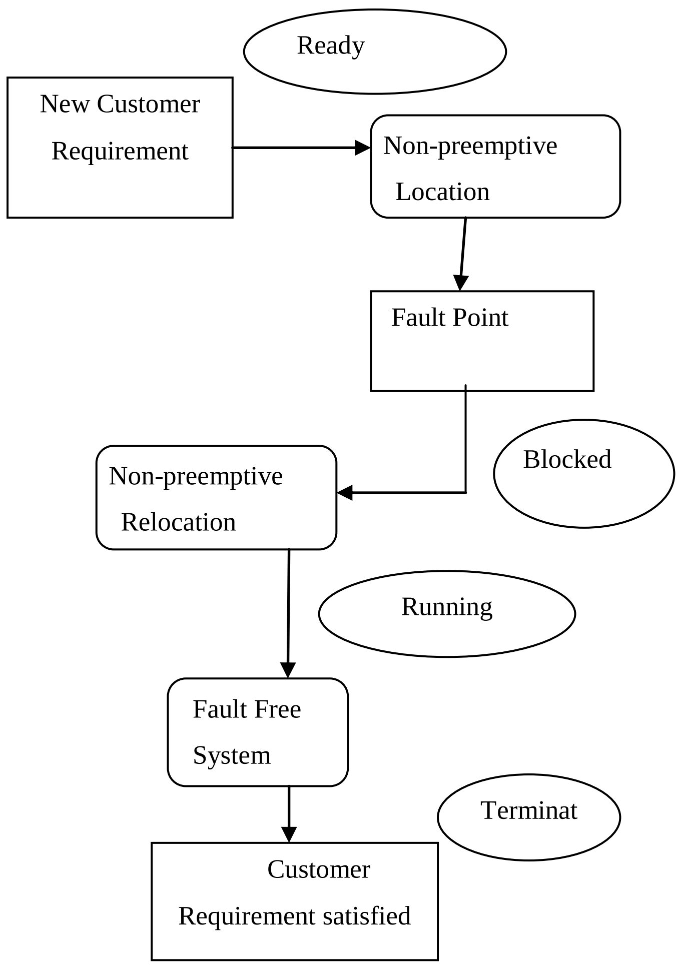

Fig 2 Non-preemptive Allocation and Relocation Procedure

Non-preemptive locating and relocating is statically controlled to offer the fault free system to the customer requirement specification.The non-preemptive work is carried out with the ready, blocked, running and terminated work. The ready block initially gets the customer requirement specifications. The specification is then allocated in the non-preemptive location block. The non-preemptive with the periodic automata identify the fault point and bock operation is carried out. PFR-NR technique then blocked the fault point and re-locates the customer requirement to the software functional model.

2.2Reactive Loop Manage Mechanism

In reactive loop,PFR-NR technique managesthe software functional statically andscheduled the task with an effective periodic system without any interruption. A software functional model in the PFR-NR technique describes the collection of periodic automata. The periodic automata perform the input, output and the intermediate action set. The software functional is carried out without any delay in PFR-NR technique. PFR-NR technique with the periodic automata denotes the variables globally to share between all the software functional systems.The integration time for customer requirement $R_i$is denoted as,

$$R_i = max_{\forall i,\pi i}(P_i) \quad \text{……. Eqn (1)}$$

P_jDenotesthe performance percentage level on ‘i’. $\forall i$denote all (i.e.) overall customer requirement specification in the PFR-NR technique. PFR-NR technique specifies the effectiveπisystem which denotes the integration work. The maximal type of integrating the software requirements produce the minimal integration time.Non-preemptive relocation is expressed as,

$$Nonpreemptive\ Relocation = R_i + (q.P_i)$$
………. Eqn (2)

The non-preemptive relocation achieves the best quality of services to the customers using the Eqn (2) outcome.$R_i$is the software requirement of the customers and ‘q’ denotes the quality of service with higher percentage level.Reactive loop manages mechanism is demonstrated in Fig 3

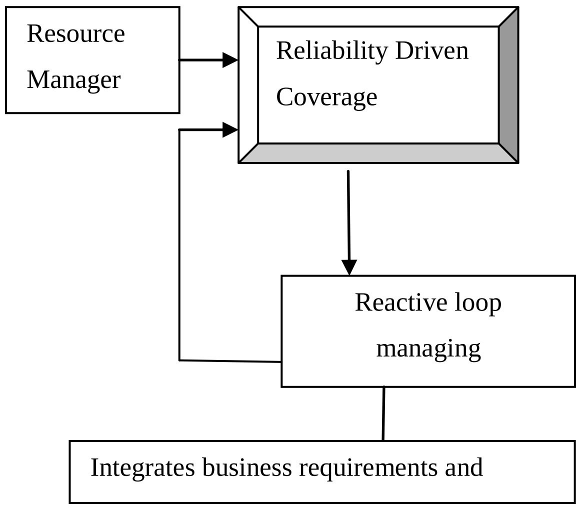
Resource Manager
Reliability Driven Coverage
Reactive loop managing
Integrates business requirements and

Fig 3 Reactive Loop Managing Steps

The Reactive loop control administers the customer requirement specification using the resource manager. Resource manager deeply identifies the false positive and negative rate to improve theperformance level in PFR-NR technique while performing the software requirement specification. In PFR-NR technique performs the reliability driven coverage in reactive loop manages mechanism.

In PFR-NR technique reactive loop, continues execution of the non-preemptive location for achieving the fault free software specification system. The relocation is carried out to provide an effective system in PFR-NR technique with the minimal integration time and effective software quality service. The looping mechanism in PFR-NR technique performs the integration between the business requirements and software functional model.

2.3PFR-NR technique Algorithmic Step

Non-preemptive relocation moves the required business specification to the software model framework by executing non-preemptive relocation technique with the effective result to the customers.PFR-NR technique is demonstrated with the algorithmic step as,

```
Step 1: Customer Requirement requested ‘Req’
Step 2: Resource Manager responseevery customer ‘Req’
Step 3: Locating the business requirement with matched software functional model
// Non-preemptive Technique
Step 4: Ready: Initially performs location work, identify fault point
Step 5: Blocked: Fault is identified and blocked
Step 6: Running: Non-preemptive Relocationin Prior Fault Removal method eliminates faults
Step 7: Terminated: Customer requirement satisfied and removed
//Reactive loop manage Mechanism
Step 8: Behavior of the software system is defined
Step 9: Periodic automata satisfy customer requirements without any interruption
Step 10: Computed $R_i = max_{\forall i,\pi i}(P_i)$ attain minimal integration time
Step 11: Integrate the business requirements and software functional model
Step 12: Resource manager response ‘Res’with customer satisfaction point.
```

Prior Fault removal executes the non-preemptive technique to locate and reallocate the software functional model without any fault points. Reactive loop accomplish the reliable coverage on each customer requirements without any interruption. The periodic automata arecarried out to satisfy customer requirements without any interruption while performing integration operation.

2 EXPERIMENTAL EVALUATION OF PFR-NR TECHNIQUE

PFR-NR technique performs the experiment in the JAVA platform using theWholesale customers Data Set.The business requirement data and the software functional model are integrated in PFR-NR technique.Wholesale customers Data Setinformation are used to perform the experiment on the factors such as Software fault prevention level, and software function integration time.

Non-preemption relocation prevents the fault occurrence while customizing the customer requirement. The software fault prevention level is measured in terms of percentage (%).Integration time in software function denotes the amount of time taken to integrate the business requirements and software functional model and measured in terms of seconds (sec).

$$Integration\ Time = I1 - I2$$

I1 – Integration Start Time

I2 – Integration Finish Time

3 RESULT ANALYSIS

In section 4, PFR-NR technique results are analyzed with the existing Model-based Oracle Software Generation (MOG) method and present systematic review of requirements specifications from software engineering model. PFR-NR technique is compared using the table and graph values on Wholesale customers Data Set.

Table 1 Tabulation of Software Fault Prevention Level

Fault Points	*Software Fault Prevention Level (%)*		
	MOG method	*S/W Engineering Model*	*PFR-NR*
3	*85*	*90*	*94*
6	*80*	*87*	*95*
9	*80*	*89*	*95*
12	*80*	*89*	*96*
15	*82*	*90*	*97*
18	*83*	*91*	*97*
21	*84*	*91*	*98*

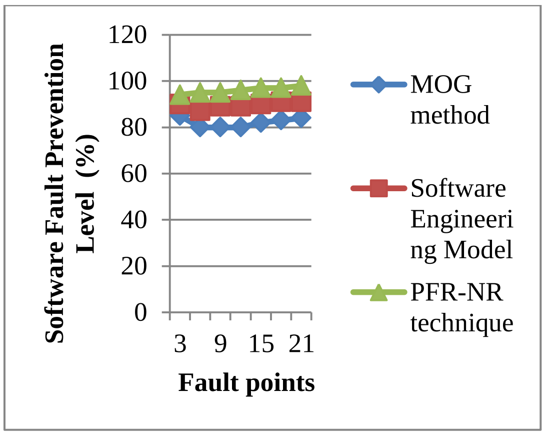

Fig 4 Performance of Software Fault Prevention Level

Table 1 and Fig 4 illustrate the software fault prevention level based on the fault points.Prior Fault removal executes the non-preemptive technique to locate

and reallocate the software functional model without any fault points. Non-preemptive technique
improves the fault prevention level from 10 – 20 % when compared with the MOG method [1] and 4 – 9 % when compared with the Software Engineering Model [2].The fault point is removed and then reactive looping is carried out for the effective management of customer specified business requirement.

Table 2 Tabulation of Software Function Integration Time

No. of Software Functions	*Software Function Integration Time (sec)*		
	MOG method	*Software Engineering Model*	*PFR-NR*
5	*192*	*179*	*160*
10	*210*	*204*	*192*
15	*239*	*221*	*211*
20	*237*	*227*	*216*
25	*256*	*249*	*234*
30	*279*	*264*	*251*
35	*321*	*302*	*287*

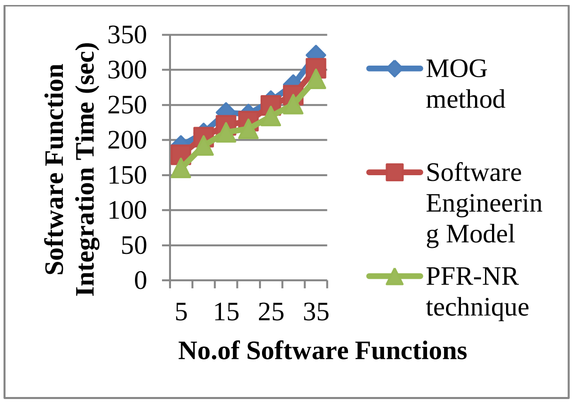

Fig 5Software Function Integration Time Measure

Fig 5 describes the software function integration time based on the software functions. The software functionin PFR-NR technique reactive loop continues execution of the non-preemptive location for achieving the minimal integration time. The integration time is reduced by 8 – 20 % when compared with the MOG method [1]. The looping mechanism in PFR-NR technique reduces the integration time by 4 – 10 % when compared with the Software Engineering Model [2].

Finally, Reactive loop manage mechanism plays a vital role while integrating the business requirements and software functional model in the non-preemptive relocation technique.

4 RELATED WORK

Present logical review of requirements specifications from software engineering model in [2] fails to integrate the business requirements with the software functional model. Systematic software requirement errors as demonstrated in [10] fail to identify the errors in the succeeding software lifecycle but do not develop an effective verification process.

Object oriented formal specifications in a collaborative expansion setting as shown in [3] consistently define the merging disagreement with precision and recall values. Model transformations obtain the architectural configuration specifications in [11] to correspondingly mark the language with iStar modeling language. Still an architectural devise solution needs an enhanced system to accomplish the NFRs present in i* SR model with low cost.Software requirements management as illustrated in [9] particularly takes the requirements specification and recognizes the major issues and concerns. The extra attributes at each dimension of business requirement fails to attain the customer need.

5 CONCLUSION

Software functional model for the business specification tackles the fault problem and reduce the overhead rate on the software specification. PFR-NR techniquepresents a simple prior fault free method for preventing failures and monitoring the resources continuously by the resource manager.The customer requirements perform experiment on theSoftware fault prevention level, and 11.349 % improved customer’s satisfaction point. Software specification overhead rate is reduced averagely up to 10 % and true positive business requirements are enhanced gradually.Reliability Driven Coverage in PFR-NR technique satisfies the different type of customer specification.

REFERENCES

[1] Lin Padgham., Zhiyong Zhang., John Thangarajah1 and Tim Miller., “Model-based Test Oracle Generation for Automated Unit Testing of Agent Systems,” IEEE TRANSACTIONS ON SOFTWARE ENGINEERING., 2013

[2] Joaquín Nicolas., AmbrosioToval., “On the generation of requirements specifications from software engineering models: A systematic literature review,” Information and Software Technology., Elsevier journal., 2009

[3] FathiTaibi., “Automatic Extraction and Integration of Changes in Shared Software Specifications,” International Journal of Software Engineering and Its Applications, Vol. 6, No. 1, January, 2012

[4] M. Brian Blake., and Michael F. Nowlan., "Knowledge Discovery in Services (KDS): Aggregating Software Services to Discover Enterprise Mashups," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 6, JUNE 2011

[5] Ahmed Harbouche., Mohammed Erradi., and AichaMokhtari., "A TRANSFORMATION APPROACH FOR COLLABORATION BASED REQUIREMENT MODELS," "International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.1, January 2012,"

[6] Michel dos Santos Soares., Jos Vrancken., "Model-Driven User Requirements Specification using SysML," JOURNAL OF SOFTWARE, VOL. 3, NO. 6, JUNE 2008

[7] David Ferreira., Alberto Rodrigues da Silva., "A Requirements Specification Case Study with Project IT-Studio/Requirements," ACM journal., 2008

[8] CagatayCatal., BanuDiri., "A systematic review of software fault prediction studies," Expert Systems with Applications., Elsevier journal., 2009

[9] Fernando Belfo., "People, Organizational and Technological Dimensions of Software Requirements Specification," Procedia Technology., Elsevier journal, 2012

[10] Gursimran Singh Walia., Jeffrey C. Carver., "A systematic literature review to identify and classify software requirement errors," Information and Software Technology., Elsevier journal., 2009

[11] JaelsonCastroa., Marcia Lucenab., Carla Silva., Fernanda Alencara., Emanuel Santosa., Joao Pimentela., "Changing attitudes towards the generation of architectural models," The Journal of Systems and Software., Elsevier journal., 2012