Journal of Theoretical and Applied Information Technology
20th June 2013. Vol. 52 No.2
© 2005 - 2013 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

FORMAL METHODS OF NON-FUNCTIONAL CONFLICT RESOLUTION FOR NETWORKED SOFTWARE TO SOLVE TRAFFIC TRAVEL COSTS

[1] LIU SHAO-HAI, [2] WU JIN- ZHAO, [3] AN NA

[1] Chengdu Inst. of Computer Application, Chinese Academy of Sci Chengdu ,610041 , China Command Colleges of Armed Police Forces in Shenyang, Shenyang 110113 , P . R. China
[2]Guangxi Key Laboratory of Hybrid Computational and IC Design Analysis, Nanning 53006, China
[3] The Academy of Equipment Command& Technology Beijing 101416, P . R. China
E-mail: [1]liushaohai@163.com, [2]himrwujz@yahoo.com , [3]anan983@163.com

ABSTRACT

The products of Networked Software must meet customers' business needs in addition to features other than the functional requirements, non-functional characteristics. In order to ensure the system performance, reliability, maintainability and scalability, it is necessary that non-functional conflict digestion method to be optimized so that the higher the efficiency of the implementation of Networked Software. Based on the Non-functional Requirement Conflict Management Meta-model, takes formal description of the modeling elements, and gives the definition of requirement semantic conflict as the foundation of checking requirement conflict. Provide a resolution method for conflict elimination. As the example of calculation traffic travel costs service nonfunctional requirements in tourism travel field, it verifies the effectiveness of this method.

Keywords: *Networked Software[1], requirement engineering, non-functional requirement[2], Formal Methods, conflict detection, conflict elimination*

1. INTRODUCTION

With the trend that people more and more attention to network and services of software system, needs conflict analysis research began to focus on the problem of non-functional requirements conflict. It is an urgent need to solve the problem how to analyze non-functional requirements of Networked Software requirements engineering. Mainly in:

(1)User-centered design thinking, user experience as an important indicator of the user experience is not only concerned about the system can do, and pay more attention to the system to do the "how".

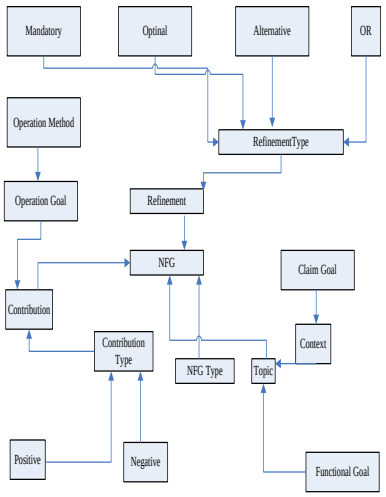
(2)The quality of service is an important constraint for service discovery and composition method, Non-functional requirements of Networked Software requirements [3]will directly affect the quality of service description. This paper explores a conflict resolution method of non-functional requirements, First of all, in order to achieve the automatic detection and resolution of conflicts demand on the basis of non-functional requirements conflict management element model, given the formal description of the modeling elements corresponding demand conflict defined as the basis for conflict detection needs. Secondly, according to the needs of different types and degrees of conflict, it is proposed to exclude conflict and conflict resolution methods reduce conflicting demands. Finally, to calculate travel billing services to a traffic travel demand "should have high security and high accuracy", for example, illustrates a complete process to exclude non-functional requirements conflict[4].

2. FORMAL REPRESENTATION OF THE NON-FUNCTIONAL REQUIREMENTS

2.1Non-functional Requirement Conflict Management Meta-model [5]

To support the formal modeling of the non-functional requirements, extracted the Meta of the non-functional requirements framework and expanded the Meta, proposed Non-functional Requirement Conflict Management Meta-model (NFRCMM), as illustrated in Figure 1.

Journal of Theoretical and Applied Information Technology
20th June 2013. Vol. 52 No.2
© 2005 - 2013 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



NFG is a representation that it organic blends the refinement of non-functional requirements of the non-functional requirements and other impact factors (such as functional requirements and situations). Non-functional objectives include to specify the type of the specific functional requirements (NFG Type) and topics (Topic). Topics include functional requirements and contexts (Context). Context as the theme, with the functional requirements of different situations can not be used alone as the theme only and functional requirements as the main theme. Operational Goal [6]is the solution of the non-functional goals, not only limited functionality, and is capable of data and system operation and constraint corresponding. The contribution (positive contribution or negative contribution) can be established by operational goals and non-functional goals .For example, "user-friendly" and "graphical user interfaces" are the positive contribution relationships.

Claim Goal is a Scene to explain the non-functional goals or dependencies. The topic of non-functional goals can be automatically generated using the content of claim goal.

The refinement is static dependencies between the upper goal and lower goal. The refined association is divided into several types according to the elements or ingredients, Mandatory and Optional. Divided according to the properties of the constraint refinement: Alternative and OR.

Contribution is the relationship that operational goals of the upper non-functional goals are "positive (Positive)" or "negative (Negative)" role. Its positive type comprising Help, Make and Some(+).Its negative type comprising Break, Hurt and Some(-).

Abstract the non-functional requirements for the goal is the core idea of the NFR framework, NFRCMM Meta model follows this core idea. Compared with the NFR framework enriched decomposition relationship of non-functional goals and objectives refined mode.

2.2 Formal Representation

Formal description of the modeling elements, can be given on the basis of the NFRCMM. Model constitute the main elements, including non-functional goals and operational goals.

2.2.1 Non-functional goals

A non-functional goals expressed as a tuple g_{NF}=<NFGType, topic>:

①NFGType ∈ {T1, T2, …, Tn}. It is non-functional attributes collection elements, classification structure of the set of non-functional properties according to the non-functional attributes.

②topic∈GF∪(GF×AContext).It is the topic of the non-functional. GF is a collection of functional goals, AContext situational properties collection. The topic of non-functional goals include at least one functional goals, and whether to include situational attributes claim goals related.

2.2.2 Operational goals

An operational goals for the tuple gOPER = <operationMethod, topic>:

①cType∈Positive∪Negative is the contribution associated type between the target. Positive={Make, Help, Some(+)}is the Positive contribution associated type, called a Positive contribution to the association . Respectively given to the contribution of the associated value of {3, 2, 1}

Negative={Break, Hurt, Some(–)} is the Negative contribution associated type, called a Negative contribution to the association. Respectively given to the contribution of the associated value of {-1,-2, -3}.

②cAss (GOPER×GOPER)∪(GOPER×GNF) is contribution the binary relationship between the contribution source and contribution goals. The GNF are non-functional goal collection. GOPER is operational goal collection. Recorded as cAss (source, target).

2.2.3 Goal refinement associated

Refined association between the goal is defined as a tuple refineDEP = <rType, rAss> to:

①rType∈{Mandatory, Optional, OR, Alternative} is the type of refined association between the goal.

© 2005 - 2013 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

②rAss GNF×GNF is a binary relation between the upper non-functional goals and lower non-functional goals. Recorded as rAss (upper, lower). The upper and lower, respectively, to act as a non-functional goals

3 NON-FUNCTIONAL REQUIREMENTS CONFLICT DETECTION

The definition of the non-functional requirements conflict is built on the basis of non-functional requirements conflict management Meta model was given semantics, semantic conflicts in the definition of non-functional requirements before first given domain ontology related semantic definitions, including the semantics of the domain concepts are equal and semantic inconsistencies.

Definition 1 (equal to the semantics of the domain concepts).For two domain concepts C_i and C_j, if C_i and C_j "owl: sameClasaAs", then the semantic concept C_i and C_j equal, denoted by $C_i \equiv C_j$.

Definition 2 (semantics of domain concepts inconsistent)Domain ontology for two concepts C_i and C_j if C_i and C_j "owl: disjointWith", then concept C_i and the concept C_j semantic inconsistencies, denoted as $C_i \cap C_j = \varnothing$.

Definition 3 (operational goal conflict)g_{OPERi} and g_{OPERj} is the topic semantically equivalent of the operating goal. operationMethod$_i$= <modifierWord$_i$, headWord$_i$>， operationMethod$_j$=<modifierWord$_j$, headWordj>. ModifierWord$_i$ ∩ modifierWord$_j$ = and headWord$_i$ ≡ headWord$_j$ claimed that g_{OPERi} and g_{OPERj} generate semantic conflict, the denoted OConflict (g_{OPERi} ,g_{OPERj}).

Definition 4 (non-functional goals absolutely conflict)For the two contribution associated conTBDEP$_i$ = <cType$_i$, cAss$_i$> conTBDEPj = <cType$_j$, cAss$_j$> ,have cAss$_i$ (g_{OPERi}, g_{NFi}) and cAss$_j$ (g_{OPERj} g_{NFj}), if { cType$_i$, cType$_j$ } ∈ Positive and OConflict (g_{OPERi}, g_{OPERj}), g_{NFi} and g_{NFj} absolute conflict, Hutchison-for AConflict (g_{NFi} g_{NFj}).

Definition5 (non-functional goals hinder)There are 2 contribution associated conTBD-EP$_i$ = <cType$_i$, cAss$_i$> and conTBDEP$_j$ = <cType$_j$, cAss$_j$>, cAss$_i$(g_{OPERk}, g_{NFi}) and cAss$_j$(g_{OPERk}, g_{NFj}), If cType$_i$ ∈ Positive ∧ cType$_j$ ∈ Negative ,then g_{NFj} is the obstacle of $g_{NFi,}$ denoted makeObstacle (g_{NFj} g_{NFi}).

Definition 6 (non-functional goals relatively conflict) ,If makeObstacle (g_{NFj}, g_{NFj}), makeObstacle (g_{NFj}, g_{NFj}), claimed that g_{NFj}, and g_{NFj} generate the relative conflicts, denoted by Conflict (g_{NFj} ,g_{NFj}).

4. NON-FUNCTIONAL REQUIREMENTS CONFLICT RESOLUTION

The traditional demand conflict resolution method focuses on the needs of the conflict, mainly processing functions include WinWin [7] and the multi-view framework [8]. The WinWin use demand conflict strategy to help the needs of the digestion process, but can not support the machine automatically digestion. Method based on the multi-view framework to automatically discover the needs of conflict, but this method can only provide transfer conflict can not fully support automatic digestion.

Conflict between the high incidence of non-functional requirements conflict than functional requirements, the extent of the conflict does not directly affect the user's satisfaction, to discuss conflict resolution in the non-functional requirements conflict detection method based on the solution: exclude conflict and reduce conflict.

4.1 Conflict Exclude

Changing the party that needs of conflict is a method more thorough and effective to exclude the demand conflict. We Can find alternative goals for the non-functional goals conflict.

Definition 7:For the two refinement associated refineDEP$_i$ = <rType, rAss$_i$> and refineDEP$_j$ = <rType, rAss$_j$> the rAssi (g_{NFk}, g_{NFi}), and rAss$_j$ (g_{NFk},g_{NFj}) and g_{NFi} ≠ g_{NFj}, if rType {OR, Alternative}, then g_{NFi} and g_{NFj} each other to replace the non-functional goals, denoted by Sub (g_{NFi}) = g_{NFj} or Sub (g_{NFj}) = g_{Nfi} .

To replace the non-functional objectives defined based on the discussion of absolute conflict Remedy. The role of the algorithm is found in the non-functional requirements conflict management model in the conflict (g_{NFi} ,g_{NFj}).

Algorithm 1 exclude absolute collision algorithm

EliminateAConflict(g_{NFi}, g_{NFj})
Input :g_{NFi}, and g_{NFj} is conflicting non-functional goals
Output :True or False
Steps:
Initialization Substituted， SubstitutingG=Null， int i=0，G[]；
If(value(cTypei)≤value(cTypej))
    Then G[2]={ g_{NFi}, g_{NFj} }；
Else G[2]={ g_{NFj}, g_{NFi} }；
EndIf
While(i<2)
    According to the definition 7,call of SubstitutingG=G[i]；
If Sub(SubstitutedG) return Null，Then i=i+1；

© 2005 - 2013 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Else out of the loop;
EndWhile
If(SubstitutingG=Null)
Then exclude demand fails and returns False;
Else use SubstitutingG to replace current SubstitutedG, return True;
EndIf.

Description:
(1) Substituted G a target object, substituted non-functional goals; SubstitutingG is also a target object, replace the non-functional goals.
(2) G [] array is a target type.
(3) value () method to obtain the contribution of the type of value.
(4) Sub (g_{NF}) method is achieved by by definition 7, returns one can replace the non-functional objectives for g_{NF}, and the non-functional goals and another non-functional goals lead to conflict will not generate new conflicts.

Exclude conflict is the preferred method to eliminate conflict, but exclude the conflicting requirements more stringent requirements conflict must be non-functional objectives can replace the target. When this premise is unable to meet the demand conflict can not be taken to exclude digestion. In addition, it is not necessary to replace the non-functional target approach, for generating the relatively non-functional goals conflict. For both cases, we provide a method of reducing demand for conflict.

4.2 Reduce Conflict

The optimizing operational goals to reduce demand for the conflict is not to change the non-functional objectives premise partial digestion needs conflict. Absolute conflict can not be ruled out, by replacing the smaller contribution of the associated value associated to reduce the conflict, both with non-functional goals have a positive contribution to the association, not conflict with the semantics of an operational objectives that add operational target, to weaken the cause indirect contribution to the conflicts associated. In the non-functional requirements conflict management model, AConflict (g_{NFi}, g_{NFj}) is uesed.

Algorithm 2: discovery of operational goals algorithm to reduce the absolute conflict findOperGforAConflict(g_{NFi}, g_{NFj}, g_{OPERi}, g_{OPERj})

Input: g_{NFi} and g_{NFj} two non-functional goals conflict with each other, g_{OPERi} and g_{OPERj} are the operational goals that lead to an absolute conflict
Output : operational goals that will be added
Steps:
Initialization OtherOperG, TargetG, AddingOperG, ALofG, int i=0;
If(value($cType_i$)≤value($cType_j$))
Then TargetG= g_{NFi}; OtherOperG=g_{OPERj};
Else TargetG= g_{NFj}; OtherOperG=g_{OPERi};
EndIf
Call choosePOperG(TargetG), and assigned to the queue ALofG;
While(i<ALofG.size())
If OConflict(OtherOperG, ALofG.get (i))
Then i=i+1;
Else AddingOperG=ALofG.get (i);
EndIf
EndWhile
Return AddingOperG.

Corresponding operational objectives for the relatively conflict hindered by substitution of non-functional goals, that adds a both positive contribution associated with non-functional objectives, without negative association actionable goals with a non-functional objectives, to weaken the relative contribution associated with conflict. The role of the algorithm 3 in non-functional requirements conflict management model is makeObstacle (g_{NFj} ,g_{NFi}).

Algorithm 3: reduce obstacles operational goal discovery algorithm findAddingOperGforObstacle(g_{NFi}, g_{NFj}, g_{OPERi}, g_{OPERj})

Input: g_{NFi} and g_{NFj} are two non-functional objectives, makeObstacle (g_{NFj} ,g_{NFi}) g_{OPERi} and g_{OPERj} are operational goals that lead to an absolute conflict
Output：operational goals to be added
Steps:
Initialization OperG, TargetG, TargetG′, AddingOperG, ALofG,
int i=0;
Assignment OperG=g_{OPERi}, TargetG=g_{NFj}, TargetG′ =g_{NFi};
Call choosePOperG(TargetG), and assigned to the queue ALofG;
While(i<ALofG.size())
If (OConflict(OperG, ALofG.get (i)) ∨ checkNConDEP(TargetG′, ALofG.get (i)))
Then i=i+1;
Else AddingOperG=ALofG.get (i);
EndIf
EndWhile
Return AddingOperG.

Description:
(1) choose POperG and AddingOperG are operational goal object; TargetG and TargetG 'non-target object function; ALofG ArrayList is an operational goal type.
(2) checkNConDEP (g_{NF}, g_{OPER}) to judge g_{NF} with g_{OPER} whether the negative contribution associated method, if it returns True, otherwise return False.

4.3 Algorithm Implementation

Analysis and design framework above 3 algorithms are built by the goal refinement and non-functional goals conflict resolution composition.

© 2005 - 2013 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In Networked Software, the user needs can be achieved by the SORL[9] language. Non-functional requirements of the framework user needs first input interface, under the guidance of the domain ontology, receiving SORL described the goals and the SORL language described in the user needs to be resolved. Target refinement, under the guidance of the list of non-functional type conflict between the goal of detecting non-functional requirements, whether there is a conflict in consultation with users, to eliminate or reduce the potential conflict and conflict resolution.

In order to achieve network-style software requirements process of the interaction of the human network built from the framework of a B / S (Browser / Server) program, the program is implemented in Java, is a typical application of the MVC pattern. The model will be based on formal description of NFRCMM modeling elements stored in the relational database persistence, loaded into memory, use the relational database Berkeley (Berkeley Software Distribution, BSD) open source database PostgreSQL. In order to facilitate the operation, the domain model is encapsulated by Javabean.

5. INSTANCE ANALYSIS

We apply this instance that the non-functional requirements to calculate travel costs of service in the field of travel and tour, to illustrate the non-functional requirements conflict detection and resolution process.

5.1 Formal Description

User demand usually can be abstracted as a top-level non-functional goal. It is an initial demand that the security and accuracy of trip expense service should be high. To establish the non-functional requirements conflict management model based on the NFRCMM framework.

(1) decomposition of refined

The functional goals travel costs (CalculateTripExpense) are refined into a mandatory sub-goals CalculateRoadExpense and CalculateHotelExpense, according to the refinement of the non-functional objectives will be non-functional goals broken down into two non-functional goals: Required goal -security calculate travel costs and optional security- accommodation cost. Functional goals "road costs" fine as the two mandatory sub-goal "get road pricing standards" and monitoring itinerary, according to a further refinement of the functional goals of non-target functional goals security [calculated road costs] is decomposed into two mandatory "security [get road pricing standards] and security [monitoring stroke]. "Refined decomposition accuracy [calculate travel costs], select the appropriate operational objectives to achieve the goal of non-functional and the establishment of the contribution associated security [for road pricing standards] "In order to achieve the non-functional goals and the accuracy of the fees and charges standard] ", respectively, specified the only operable objectives and establish a positive contribution to the association. According to the formal description of NFRCMM modeling Meta, we reach the formal representation in the travel field NFRCMM model .

(1) non-functional goals

g_{NF1}=<Security, CalculateTripExpense>

g_{NF5}=<Security, CaptureREStandard>

g_{NF6}=<Accuracy, CalculateTripExpense>

g_{NF11}=<Accuracy, QueryREStandard>

g_{NF12}=<Accuracy, QueryPERecord>

(2) Operational goals

g_{OPER1}=<AuthenticateRight, CaptureREStandard>

g_{OPER2}=<RealtimeValidation, QueryREStandard>

g_{OPER3}=<UnauthenticateRight, QueryREStandard>

(3)Refined association

g_{OPER1}=<AuthenticateRight, CaptureREStandard>

g_{OPER2}=<RealtimeValidation, QueryREStandard>

g_{OPER3}=<UnauthenticateRight, QueryREStandard>

(4)Contribution association

$conTBDEP_1$=<Help, cAss(g_{OPER1}, g_{NF5})>

$conTBDEP_2$=<Help, cAss(g_{OPER2}, g_{NF11})>

$conTBDEP_3$=<Help, cAss(g_{OPER3}, g_{OPER2})>

5.2 Collision Detection

First topic compatible should be considered by g_{OPER1} and g_{OPER3}, according to the defined judgment OConflict (g_{OPER1} g_{OPER3}). Second contribute association can be checked by $conTBDEP_2$ = <Help, cAss(g_{OPER2}, g_{NF11})> and $conTBDEP_3$ = <Help, cAss (g_{OPER3}, g_{OPER2})> .Last get $conTBDEP_4$ = <Help, cAss(g_{OPER3}, g_{NF11})>.

According to definition 4 , $conTBDEP_1$ = <Help, cAss(g_{OPER1}, g_{NF5})> and $conTBDEP_4$ = <Help, cAss(g_{OPER3}, g_{NF11})>, since OConflict (g_{OPER1}, g_{OPER3}), therefore AConflict (g_{NF5}, g_{NF11}).

5.3 Resolve Conflict

© 2005 - 2013 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

First， use the algorithm 1 to preclude the use of non-functional requirements conflict, perform the following steps:

(1)Since value(Help)=value(Help) ， G[2]={g_{NF5}, g_{NF11}}；

(2)When i=0 ， SubstitutedG=g_{NF5} ， SubstitutingG=Null，i=i+1；

(3)When i=1 ， SubstitutedG=g_{NF11} ， SubstitutingG=g_{NF12}；

(4) Use g_{NF12} replace g_{NF5}, conflict resolution.

This example illustrates the absolute conflict resolution process, finds the replacement g_{NF1} non-functional objectives is the key to conflict resolution. Can not exclude methods of conflict resolution to the conflict, can be used to reduce conflict. It is the key that successfully find the operational objectives which can replace g_{OPER1} in resolve conflict.

6. SUMMARY

This paper discusses the non-functional requirements conflict resolution method, to some extent, to fill the blank that needs of the lack of analysis of the problem of non-functional requirements analysis. In addition, with the traditional demand for conflict management method, the method is a demand for a conflict management method in the semantic, promoting progress based on semantic demand conflict[10] studies.

REFERENCES

[1]He K Q,Peng R,Liu J,et al.2006.Design methodology of networked software evolution growth based on software patterns[J].Journal of Systems Science and Complexity,19(2):157-181.

[2]Li X and Chen G.2003.A local-world evolving network model[J].Physica A,328(1/2):274-286.

[3]Valverde S and Sole R V.2005.Network motifs in computational graphs:a case study in software architecture[R].Working paper of Santa Fe Institute,SFI/05-04-008.

[4] Saraswat V. X10 Language Specification[EB/OL]. (2010-08-28).http://x10-lang.org.

[5] Raman R. Efficient Data Race Detection for Async-finish Parallelism[C]//Proc. of the 1st International Conference on Runtime Verification. Heidelberg, Germany: Springer-Verlag, 2010: 368-383.

[6] Boehm B, Kitapci H. The WinWin Approach: Using a Requirements Negotiation Tool for Rationale Capture and Use[EB/OL]. (2006-06-30). http://sunset.usc.edu/csse/TECHRPTS/2006/usccsse 2006-630.pdf.

[7] Vasudevan N. Compile-time Analysis and Specialization of Clocks in Concurrent Programs[C]//Proc. of CC’ 09. [S. l.]: IEEE Press,2009: 48-62.

[8] Agarwal S. May-happen-in-parallel Analysis of X10 Programs[C]//Proc. of the 12th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming. San Jose, USA: ACM Press,2007.

[9] Christiaens M, de Bosschere K. TRaDe: A Topological Approach to On-the-fly Race Detection in Java Programs[C]//Proc. of 2001 Symposium on Java TM Virtual Machine Research and Technology. Berkeley, USA: USENIX Association, 2001: 105-116.

[10] Yelick K. Productivity and Performance Using Partitioned Global Address Space Languages[C]//Proc. of 2007 International Workshop on Parallel Symbolic Computation. New York, USA:ACM Press, 2007: 27-28.