



System Functional and Nonfunctional Requirements and Interaction Analysis by Using Aspects Modeling Techniques

Mr. Sagar Mohite, Prof. Dr. S. D. Joshi

Department of Computer Engineering

B. V. U. College of Engineering

Pune, India

Abstract— *AspectsJ Modeling Techniques design for categorizes different system concerns appropriately within system model using various aspects Techniques. Within a software engineering process, AspectsJ consider as a system concerns & these concerns are divided into various system modules. In user functional & Nonfunctional requirements modeling techniques by using these techniques easily understand system consistent or not & system interactions. Here we aim to use UML class design to analyze user or customer requirements in form of model based techniques. Within a software engineering process of SRS requirement system model difficulty is collect using UML class design. UML design consist of many UML use cases, joinpoints, activities, classes, method to generate system concerns. Within AGG tool create rule & then rule transformation, after rule transformation process of analyzing conflicts matrix and dependencies matrix of generated rule in AGG tool.*

Keywords— *AspectsJ Modeling Techniques (AMT), User Requirement, Pointcuts, AspectJ, Conflicts matrix, Dependencies matrix, Rule Graph Transformations, System concerns.*

I. INTRODUCTION

In AspectsJ Modeling Techniques is implementing at starting of any system software implementation but in earlier it possible to implement in system analysis & system design and system modeling. Within a software engineering process, AspectsJ consider as a system concerns & these concerns are divided into various system modules. AspectsJ Modeling Techniques characterize the AspectsJ at stage when system requirement process going on.

The main use of AspectsJ Modeling Techniques is provide authority to system designer or programmer to analyze effectively about not excepted & unnecessary modification in previously implemented system. But system designer or programmer still support to well design system modules & fully tested system modules are as reusable system modules.

Till now AspectsJ Modeling Techniques not provide any AspectsJ implementation tool & AspectsJ programming language also not provide knowledge about AspectsJ analysis & AspectsJ design Techniques, that's reason of that previous system developer not use AspectsJ Modeling Techniques in good & big system development.

AspectsJ Modeling Techniques apply in any phase in system development process like analysis phase, design phase, development phase. Here we focus on to implement AspectsJ in design phase. AspectsJ Modeling Techniques mostly use for dividing different system concerns, AspectsJ Modeling Techniques apply at any phase of system development, here we apply AspectsJ in design phase. In AspectsJ Modeling Techniques we focus on to analyze user Nonfunctional requirements, the Non-functional requirements are security, availability, performance, efficiency, response time etc. here we mainly focus on response time & security.

We apply Nonfunctional requirements to web design, web pages & try to apply on web services by using AspectsJ techniques for that propose we use AspectJ development tool kit in 8.6 eclipse or j-spring technologies, by applying AspectJ to non-functional rules that generated in AGG tool, then eclipse provide created AspectsJ-tree for AspectsJ is apply well or not. AspectJ use for cross verification purpose, we also see in Dos-box about aspect applied fully or not.

II. EXISTING SYSTEM

Tell about existing system of AspectsJ Modeling Techniques, Nonfunctional requirement like availability, security, performance etc. not yet applied to web design or web services so we try to apply it on web design to achieve more security, performance & response time & in past AspectsJ modeling techniques not provide the good tool for developing Aspects & also not provide the language information about how to use the AspectJ in development which programming language is more suitable & easy to implement AspectsJ to produce correct output, also not provide backup file or user manual or plug-ins for easy implementations of AspectJ to rules. & that the reason old programmer not use AspectsJ modeling techniques for big industry projects or software engineering project that require more accuracy or security or correctness, but now the AspectsJ modeling techniques provide a different tool like AspectJ development tool kit.

For implementation also provide user manual of how to use it & also provide the more secure language for implementation of AspectsJ like java or latest java spring technology for the implementation of rule.

But there is no method is so far implemented for accessing rule directly & applies AspectJ in eclipse java or spring java technology, for that purpose we need to use a activity graph grammar tool i.e. AGG tool to create & provide rules for apply to AspectJ in eclipse aspect development tool kit, Access the AGG rules by using GGX, GXL AGG tool files or by using manual method directly use in eclipse 8.6. Or java spring technology.

III. IMPLIMENTATION

Implementation Techniques:

We use UML class design to analyze user, customer and system requirements in form of model based techniques, we analyze class design & various requirements and create rule, but we need a tool for creating rule, AspectsJ Modeling Techniques provide tool for creating rule known as activity graph grammar tool (AGG tool)

In AGG tool create rule after that rule transformation process, after rule graph transformation next process is analyzing conflicts matrix and dependencies matrix of generated rule in AGG. The entire think discus above see in detail in implementation detail

Implementation Detail:

We see implementations in Section wise one by one similarly the flow of implementation use in AGG tool.

Section I: Generation of TypeGraph using UML class design in AGG tool:

Section first we see about TypeGraph, in these section we generate TypeGraph with the help of UML class design. Create TypeGraph similar to UML class design. Here we consider type graph is input for AGG tool. In TypeGraph we need to add similar attribute, method, operation as show in UML class design. In TypeGraph classes in UML design can show as node & link in UML design can show as edge with bidirectional or unidirectional. In TypeGraph on edge show multiplicity & constrain apply on that edge. Multiplicity shows like that 0...N.

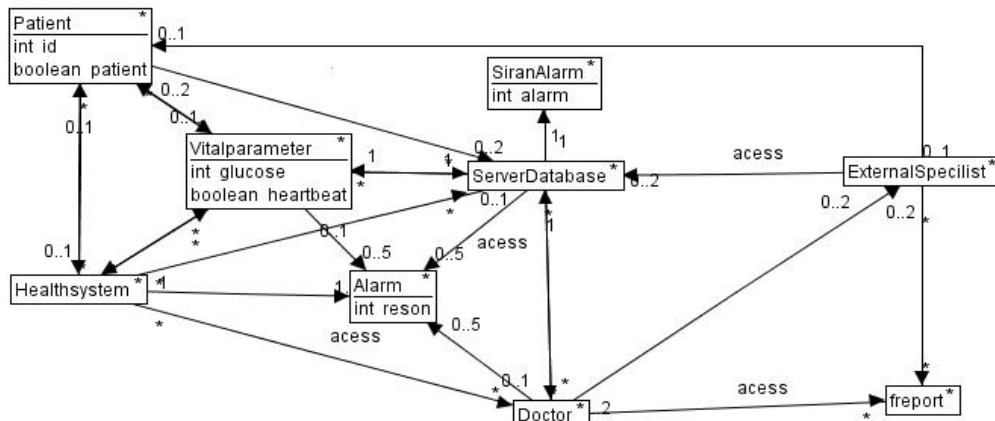


Fig. 1 Image AGG TypeGraph of Remote patient treatment System

After creation of TypeGraph next process is to generate many functional & Nonfunctional LHS & RHS rules in AGG tool. Next Process is generation Graph from rule & TypeGraph. That graph is transformed using created rule & TypeGraph in AGG.

Creation of AGG Rules:

After TypeGraph creation process, next process is analyze & create different functional & Nonfunctional LHS & RHS rules in AGG tool, in AGG tool we create rule first we create LHS rule condition after that create changes in the LHS rule finally known as RHS rule condition, we see many functional & Nonfunctional LHS & RHS rules one by one as follows.

Rule I: Authentication or System Login:

Here the rule state that if any system user ID or patient ID is valid then & then only this patient permitted to enter into Remote patient treatment System or not then show link of new user registration form. By using that rule system provide user non-functional requirement like security.

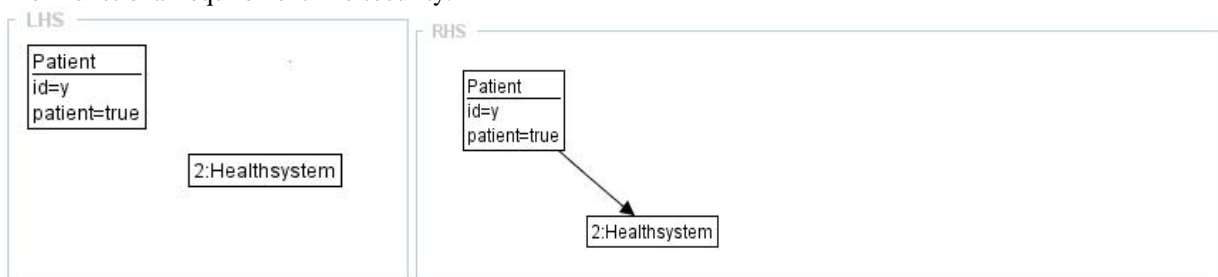


Fig.2 User Authentication

Rule II: Patient in health system notification to doctor:

Second rule state that when patient enter into system by authentication process after that system analyze patient condition by test then notification message to system doctor about patient in system & patient assign to your id & this rule use by remote patient treatment system for improving the Nonfunctional requirement response time.

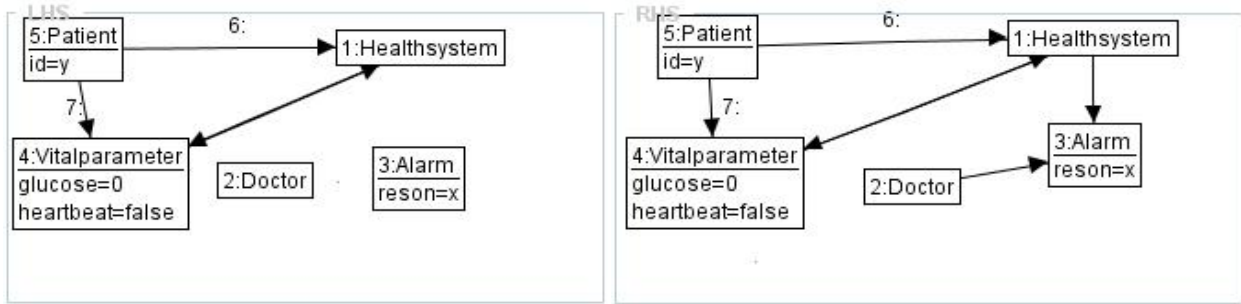


Fig. 3 Patient in health system notification to doctor

Rule III: Storage of patients report in different database:

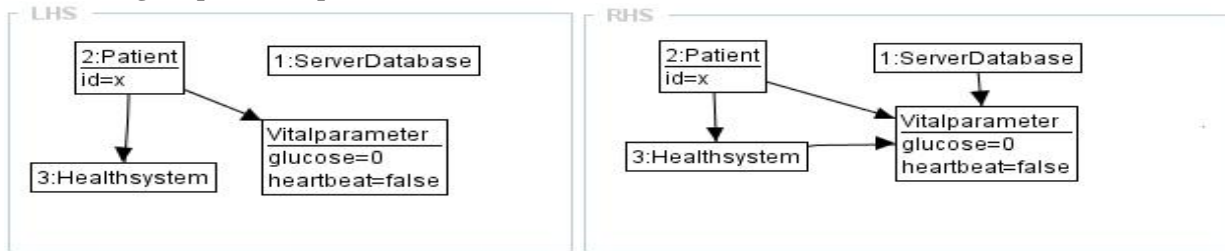


Fig. 4 storage of patients report in different database

Rule third Storage of patients report in different database state that system check patient & generate the test report & system analyze that test report & store on different server. & this rule use by system for provides the Nonfunctional requirement security.

Rule IV: Creation of alarm notification & storage of alarm reason:

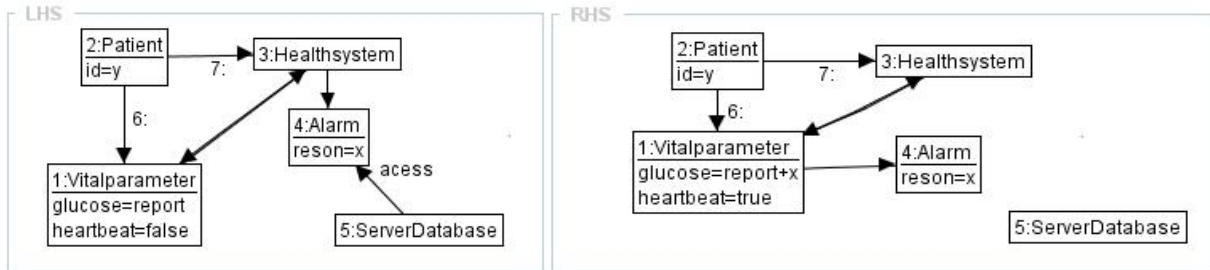


Fig. 5 Creation of alarm notification & storage of alarm reason

Rule fourth Creation of alarm notification & storage of alarm reason state that system check the patient & create patient report analyze by system & then creation of alarm message with reason & reason store in database. & this rule use by remote patient treatment System for implements the Nonfunctional requirement security & response time.

Rule V: Creation of SirenAlarm notification:

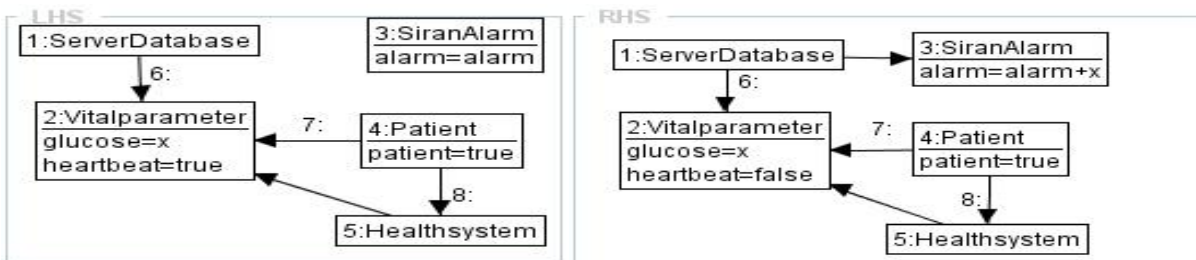


Fig. 6 Creation of SirenAlarm notification

Rule fifth Creation of SirenAlarm notification state that system checks the patient authentication & remote patient treatment System create patient report & analyze by system & then store in database. Final step is creating SirenAlarm notification & this rule use by remote patient treatment System for implements the Nonfunctional requirement security & response time.

Rule VI: External doctor call by internal doctor:

Rule sixth External doctor call by internal doctor state that internal doctor checks the patient report if patient test report is critical then remote patient treatment Systems internal doctor tack decision to call external doctor which is specialist in particular field to give treatment to patient for that purpose external need to collect old test report which is store in database. This rule used by treatment System for implementing the non-functional requirement such as security & response time.

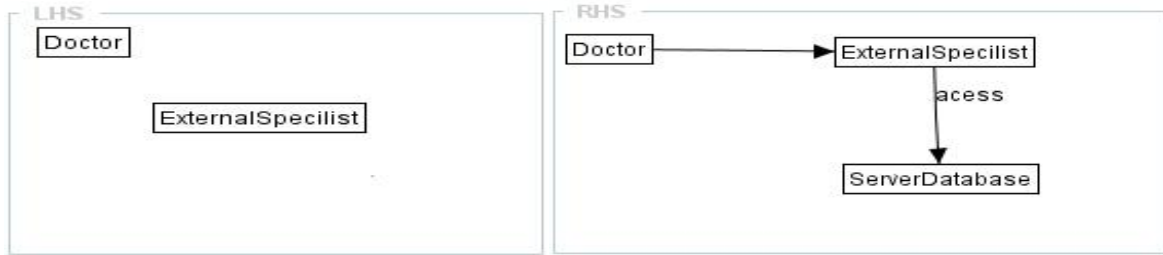


Fig. 7 External doctor call by internal doctor

Rule VII: External doctor final suggestion:

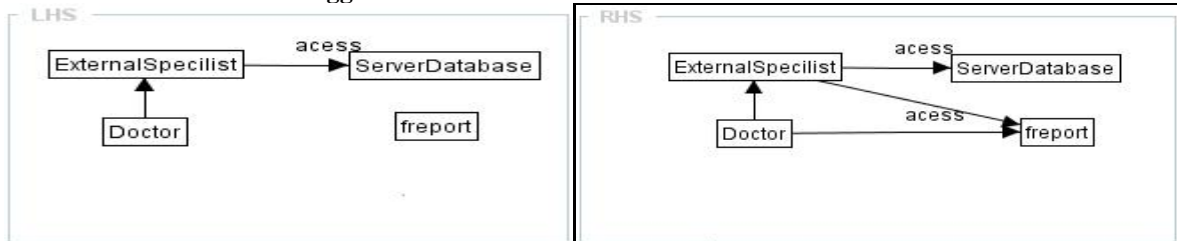


Fig. 8 External doctor final suggestion

Rule seven state that external doctor create final suggestion report of patient these report fetch by doctor for treatment.

AGG Before & After Transformation of Graph process:

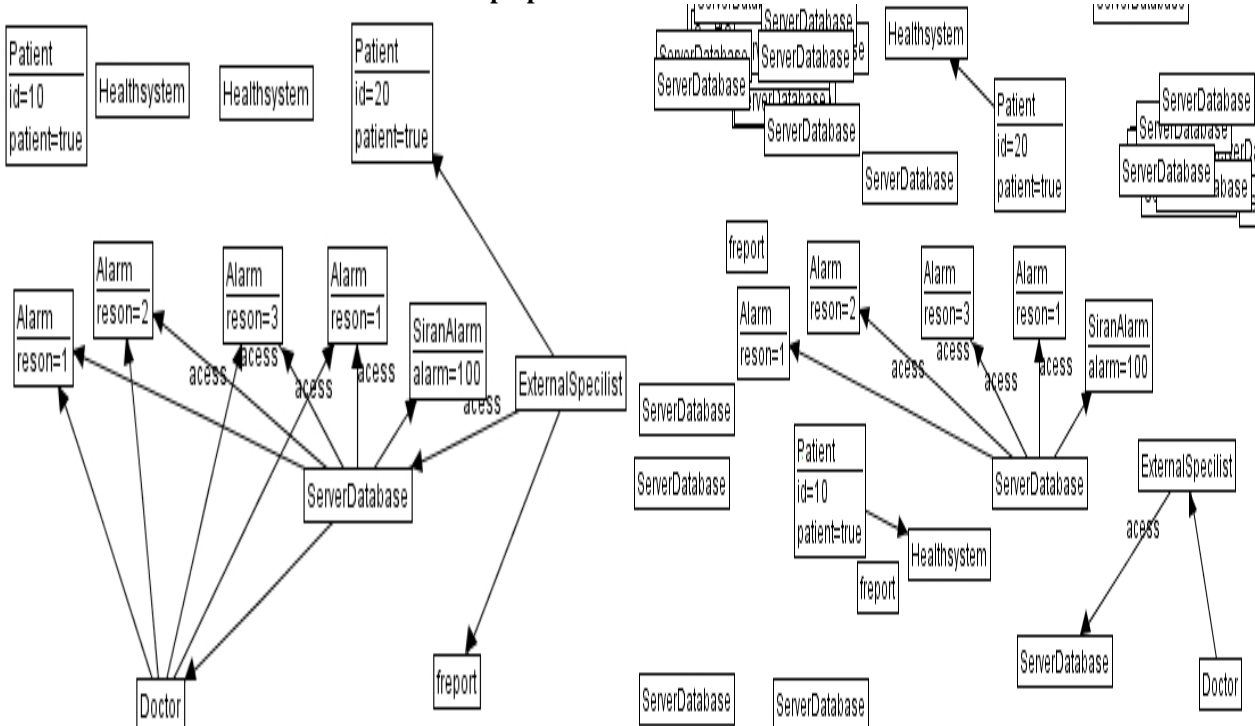


Fig. 9 before & after Transformation of Graph process

By using rules in AGG & TypeGraph possible to transform graph that show the changes in LHS & RHS rules condition.

Section II: Matrix of minimal Conflicts and dependency:

After transformation of graph, next is to analyze minimal conflicts & minimal dependencies. We see how matrix count number is show in diagram, if matrix shows 3 dependency that means the Row X Column [rule 1, rule 2] rule apply with each other then they shows dependency & conflict, dependency means system is dependent on each other,& conflict means by applying these create a confutation. By using matrix & AspectJ try to minimize that conflict matrix numbers.

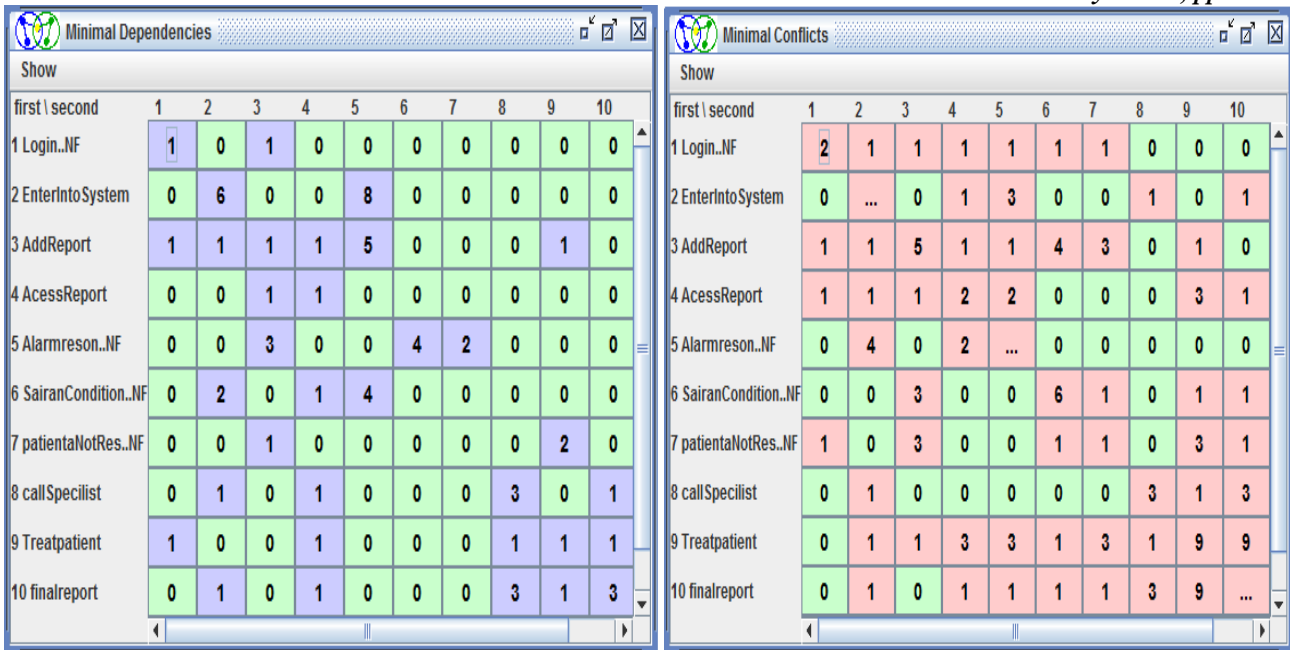


Fig. 10 Matrix of minimal Conflict and dependency matrix

Second fig. matrix of conflict 8X8 rule show 3 conflict reason of doctor call with specialist there are too many patient is present so who is call by system doctor not understand by system & that why show three 3conflict number, by using AspectJ development toolkit we try to minimize conflicts as well as dependencies.

Section III: Apply AspectsJ to AGG rule:

We aim to apply AspectsJ of created AGG rule by using AspectsJ development toolkit. We analyze the rules created in AGG tool, after that develop system similar to AGG rule using java eclipse 8.6 or java spring, create a system one by one as AGG rule & apply the AspectsJ to each developed rule design system, AspectsJ means java Aspect & AspectsJ use for cross checking the operation done successfully or not, in our system we use AspectsJ Modeling Techniques programming & their method to show AspectsJ technique, we use before method, after method, around method to generate AspectsJ message & AspectJ Tree, we see the syntax of different AspectsJ method.

//before (): next execution (*class name ()) show what done by calling class method before operation. //after (): next execution (*class name ()) show what done by calling class method after operation.

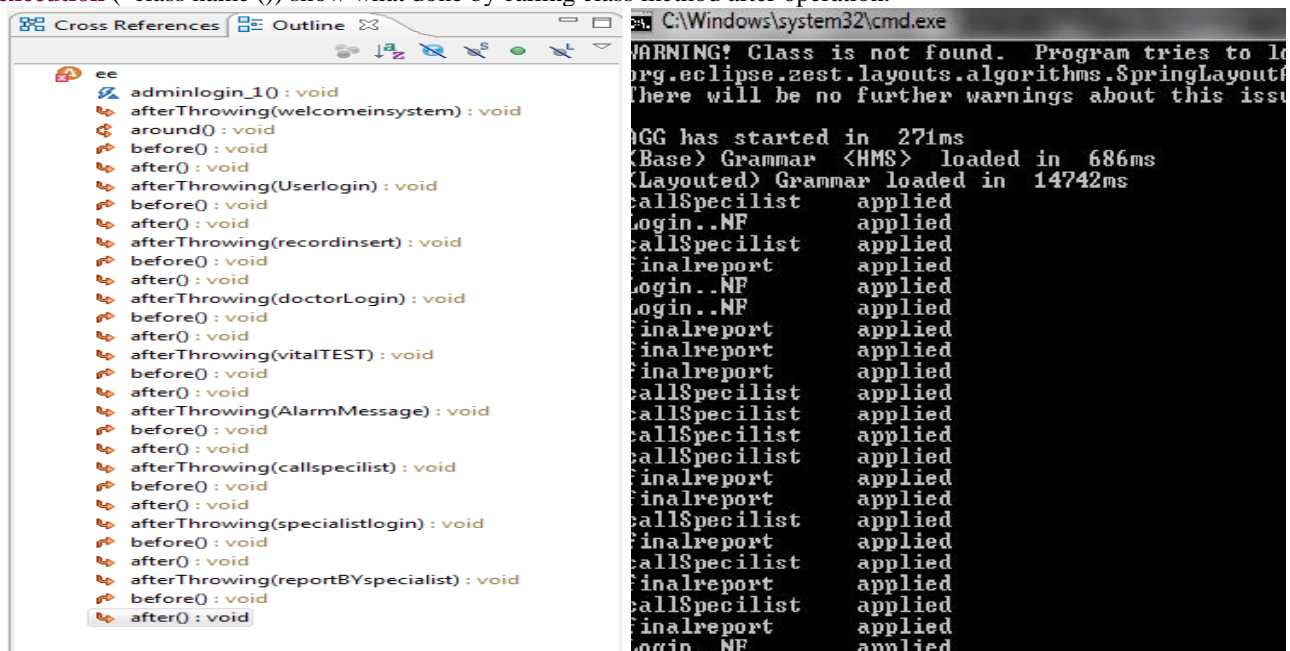


Fig. 11 the tree of aspects & Dos-box output aspects is applied

By using the AspectJ method after & before, around we apply the AspectsJ to AGG rules in eclipse 8.6 than eclipse shows as output of AspectJ, AspectJ is for cross-verification process,& that shows by eclipse in Aspect tree format, if AspectsJ is applied successfully then & then only aspect tree is generated. We also see the aspect applied verification in Dos-box also see how much time is required for applying to all rules.

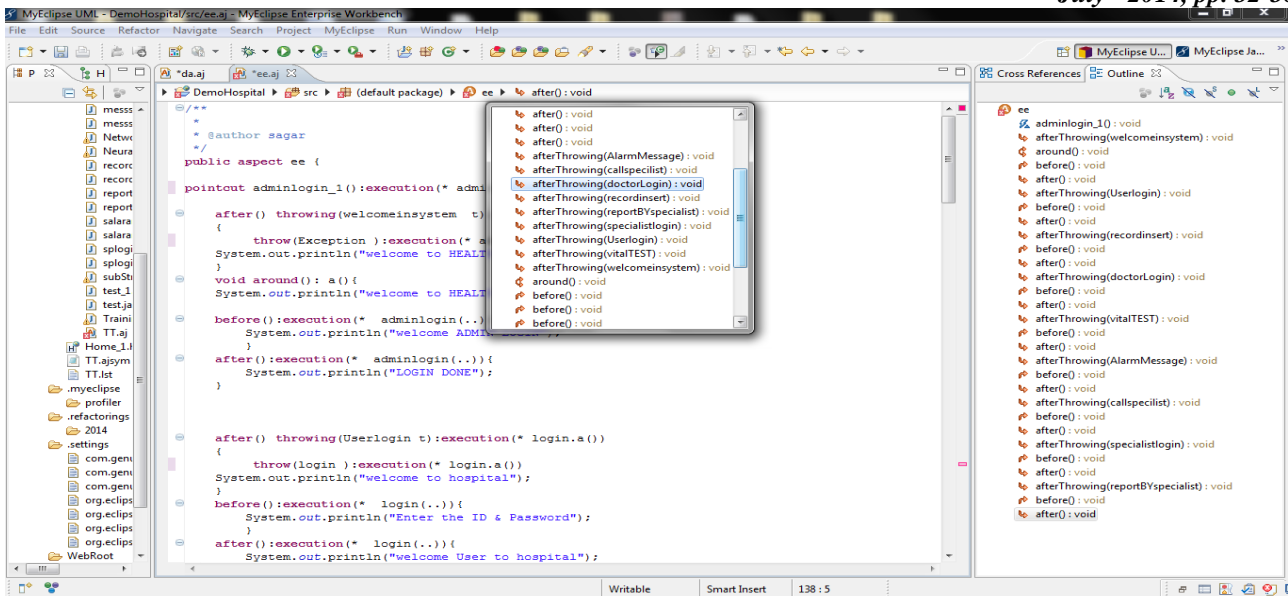


Fig. 12 AspectsJ coding, AspectsJ is applied & AspectsJ tree is generated.

IV. CONCLUSION

In AspectsJ modeling techniques mostly use for dividing different system concerns. Dividing different concerns means dividing different system concerns into many small one. using AspectsJ modeling techniques we analyze user Nonfunctional requirement & that requirement are the performance, availability, security, response time, here we concentrate on Nonfunctional requirement response time & security, for that purpose we create rules but there is no method to directly access the rule & that's the reason we use AGG tool for creation of non functional rules for remote patient treatment system. In creation of rule in AGG by analyzing the user Nonfunctional requirement & then making LHS rule condition & RHS rule condition & after that transformation of that rule & then we analyze the dependency & conflict in AGG rule, after that access the these AGG file by GXL or GXJ file or manual analyzing in eclipse 8.6 java AspectsJ development toolkit or access in java spring technology, after getting file from AGG Tool eclipse 8.6 java tool implement remote patient treatment system web application & apply AspectsJ rule to each user require stages for that purpose we need to install all AspectsJ plug-in. after successfully applying the AspectsJ to web design i.e. remote patient treatment system the eclipse 8.6 show the aspect applied or not through AspectsJ tree, that means the AspectsJ applied successfully then & then only aspect tree is generated show otherwise not show AspectsJ tree, we also cross verify the aspect applied or not By using DOS box ,then we generate a web design of remote patient treatment system that provide response time & security etc Nonfunctional requirement.

In that system we mostly concentrate on Nonfunctional requirement response time & security to web design but in next we try to focus on more Nonfunctional requirement availability, performance, efficiency etc.

In next system we try to apply Nonfunctional requirement using AspectJ to web services using more secure technology java spring ADJ.

Reference

- [1] K. Mehner and G. Taentzer. Supporting Aspect- Oriented Modeling with Graph Transformations. In P. Clements et al., editor, AOSD 05Workshop on Early Aspects, 2005.
- [2] E. Baniassad and S. Clarke. Theme: An Approach for Aspect-Oriented Analysis and Design.
- [3] Robert France, Indrakshi Ray, Geri Georg, and Sudipto Ghosh. Aspect-oriented Approach to Early Design Modelling. IEEE Proceedings Software, 151(4):173– 185, August 2004
- [4] G. Kiczales, E. Hisdale, J. Hugunin, M. Kersten, and J. Palm. An overview of AspectJ.
- [5] T. Elrad, M. Aksits, G. Kiczales, K. Lieberherr, and H. Ossher: “Discussing Aspects of AOP”. Communications of the ACM 44(10), pp. 33–38, October
- [6] P. Tarr, H. Ossher, W. Harrison, and S. M. Sutton, Jr.: “N Degrees of Separation: Multi-Dimensional Separation of Concerns”. In Proceedings of the 1999 International Conference on Software Engineering.
- [7] H. Poor, an Introduction to Signal Detection and Estimation. New J. Rumbaugh, I. Jacobson, and G. Booch.
- [8] S. Herrmann, C. Hundt, and K. Mehner. Mapping Use Case Level Aspects to Object Teams/Java. In A. Moreira etc. al, editor, OOPSLA Workshop on Early Aspects, 2004.
- [9] Jon Whittle & Praveen K. Jayaraman (2007): MATA: A Tool for Aspect-Oriented Modeling Based on Graph Transformation. In: Holger Giese, editor: MoDELS Workshops, Lecture Notes in Computer Science 5002. Springer, pp. 16–27. Available at <http://dblp.uni-trier.de/db/conf/models/models2007w.html> WhittleJ07.
- [10] I. Jacobson and P.-W. Ng. Aspect-Oriented Software Development with Use Cases. Addison Wesley, 2005.
- [11] B.Nuseibeh, J. Kramer, and A. Finkelstein. AFramework for Expressing the Relationships betweenMultipleviews in Requirements Specifications.

- [12] Object Teams home page. <http://www.ObjectTeams.org>.
- [13] J. Araújo and P. Coutinho. Identifying aspectual use cases using a viewpoint-oriented requirements method. In *Early Aspects 2003: Aspect-Oriented Requirements Engineering and Architecture Design*, Boston, MA, USA, March 2003
- [14] A. Rashid, P. Sawyer, A. Moreira, and J. Araújo. Early aspects: A model for aspect-oriented requirements engineering. In *Proc.*
- [15] E. Baniassad and S. Clarke. Theme: An Approach for Aspect-Oriented Analysis and Design.
- [16] Jon Whittle & Praveen K. Jayaraman (2007): MATA: A Tool for Aspect-Oriented Modeling Based on Graph Transformation. In: Holger Giese, editor: *MoDELS Workshops, Lecture Notes in Computer Science 5002*. Springer, pp. 16–27. Available at <http://dblp.uni-trier.de/db/conf/models/models2007w.html> WhittleJ07.
- [17] G. Kiczales, E. Hisdale, J. Hugunin, M. Kersten, and J. Palm. An overview of AspectJ.
- [18] I. Jacobson and P.-W. Ng. *Aspect-Oriented Software Development with Use Cases*. Addison Wesley, 2005.
- [19] B.Nuseibeh, J. Kramer, and A. Finkelstein. *A Framework for Expressing the Relationships between Multiple Views in Requirements Specifications*.
- [20] Object Teams home page. <http://www.ObjectTeams.org>.
- [21] J. Araújo and P. Coutinho. Identifying aspectual use cases using a viewpoint-oriented requirements method. In *Early Aspects 2003: Aspect-Oriented Requirements Engineering and Architecture Design*, Boston, MA, USA, March 2003
- [22] A. Rashid, P. Sawyer, A. Moreira, and J. Araújo. Early aspects: A model for aspect-oriented requirements engineering. In *Proc.*