RCES REVIEW OF COMPUTER ENGINEERING STUDIES

ISSN: 2369-0755 (Print), 2369-0763 (Online)
Vol. 4, No. 1, March 2017, pp. 1-4
DOI: 10.18280/rces.040101

A publication of IIETA

IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering
http://www.iieta.org/Journals/RCES

Traceability and association between models in aspect oriented programming

Asim Ismail*, Ambreen Yousuf

SZABIST Dubai, CIIT Pakistan

Email: asimismail7@gmail.com

ABSTRACT

Aspect Oriented Programming (AOP) is evolving as a better modular programming paradigm compared to the traditional Object Oriented Programming (OOP). Due to the increasing popularity, there is a need of techniques for the quantifiable measurement of AOP-based system design models. The main idea therefore is to develop such a technique, which can evaluate AOP-based software models for the association between them, and to improve traceability of system requirements. Software metric could map design models with each other, and could provide a mechanism to trace requirements from early phase to their implementation. There are some metrics already available to evaluate some specific property of an aspect-oriented model, such as cohesion, coupling, size, etc. However, there was a chance to design metrics for evaluating Aspect Oriented and UML models comprehensively, for example for evaluating mapping of models, association between models and inter and intra-dependencies of models. In this research traceability technique has been designed to check the validation of aspect oriented and UML models. Models are mapped with the help of traceability matrix and metadata keys. Each design model has a unique key, which has three parts: module number, phase ID and diagram number. With the help of this key, 3D traceability matrix and mapping diagrams have been designed, which shows the association between the models and as a result make the overall system design more traceable. The evaluation of the technique has revealed that the resultant technique has improved mapping between the design models and as a result have enhanced effective traceability of the requirements of the system.

Keywords: AOP, UML, Metric, Traceability, Metadata Key.

1. INTRODUCTION

Quantifiable measurement of a software product and the process that is directly observed, calculated, or anticipated is done by metric. In an overall view, the main aim of software measurement is to provide the developers with well-situated, data-driven results; to evaluate the response of process changes and to chase the advancement of an organization towards the desired destination. Software metric is also helpful for project manager in the different terms [1].

In Aspect Oriented Programming (AOP), concern is a feature like metrics is called a crosscutting concern. AOP is a programming paradigm, which compliments Object Oriented Programming (OOP) by separating concerns of a software application to enhance modularization. The separation of concerns aims for making software easier to handle by grouping features and behavior into manageable parts which all have a specific purpose and importance. AOP allow the altering to security (or any other) aspect into its own package and leaves the other objects with responsibilities, probably not enforcing any security themselves. For modularizing and composing concerns, which are not easily handled by using traditional techniques, AOP provides a gentle way [2] so it needs to be measured.

Proposed method designed new metrics for the evaluation of aspect-oriented and UML models. This research has designed metadata key for models. Each design model has a unique key, which has three parts: module number, phase ID and diagram number. With the help of this key, 3D traceability matrix and mapping diagrams has been designed. There are some metrics previously proposed for evaluation of some specific properties of software design models i.e. coupling, size, cohesion etc. However, there is no such comprehensive metrics available to evaluate software design models that can fully map the models and check their association. In the proposed metrics aspect- oriented and UML models are evaluated to find their correctness and mapping with each other. It also helped in forward and backward traceability of models in the system.

2. METHODOLOGY

This research aims to develop design metrics to evaluate software design models. For this purpose, following is the sequence of activities that are executed in this research:

A. Traceability matrix

During the validation process traceability, matrix (TM) is used as a table that links the demands required [3]. Traceability matrix guarantees the authenticity of all requirements being tested under standard protocols. Both the validation team and the auditors to make sure that during validation project all requirements are preserved and to review the validation documentation respectively use traceability matrix. In this research traceability matrix has been designed to check the validation of aspect oriented models and UML models. By using designed traceability matrix, mapping diagrams has been designed with the help of keys generated.

B. Mapping

The process used for creating data element mappings between two different data models is mapping. Mapping models identifies relationships of data. In this research mapping models are designed with the help of traceability matrix and keys generated with the help of three levels of designs i.e. Analysis level, Early Design level and Design level. This key table is shown in chapter 3. Key table is generated with the help of two case studies; one is of aspect-oriented models and the second case study is of UML models.

C. Metadata key

Each design model has a unique key, which has three parts: module number, phase ID and diagram number. With the help of this key, 3D traceability matrix and mapping diagrams has been designed.

D. Tool

A tool is developed to check the traceability of matrix; tool takes inputs from user for each level and keys for matrix. Then matrix shows the traceability of each level. This research work has designed the metadata keys to identify the levels uniquely. It has designed keys on the basis of three levels of design i.e. analysis level, early design level, and design level and then created tables of each correlated level. With the help of these correlated tables, this research has designed 3-D traceability matrix and with the help of key logs mapping diagrams are designed too. Each design model has a unique key, which has three parts: module number, phase ID and diagram number. A tool is developed to create traceability matrix, user will input the keys for matrix.

3. PROBLEM VERBALIZATION

Some metrics are available to evaluate some specific property of an aspect-oriented model, such as cohesion, coupling, size, etc [4]. For evaluating Aspect Oriented and UML models comprehensively, mapping of models, association between models and inter and intra-dependencies of models metrics are designed. Since AOP is a new paradigm so there is immaturity as far as evaluation of models in the concerns [5]. Proposed method designed new metrics to measure the software design models comprehensively so far so it has motivated us to do work in this domain. This method designed new metrics for the evaluation of software design models. And then tested these metrics on more than one real life case studies, and developed a tool that can design the 3D traceability matrix by the given keys by the user.

4. ASPECT ORINTED PROFRAMMING

4.1 Software metrics for the evaluation of aspect oriented models

The purpose of this research is to show the association between the software design models and to show inter and intra dependencies of models. This study aims to achieve the following research objectives: To design metadata keys and formal explanation of the key by drawing Module State Diagram. To design mapping models for traceability matrix. To design traceability matrix to show the association between models and formal explanation of the key by drawing Module State Diagram. To apply on two case studies; one is of aspect oriented and second one is of UML and developing a tool to design traceability matrix. This research work has designed the metadata keys to identify the levels uniquely. It has designed keys on the basis of three levels of design i.e. analysis level, early design level, and design level. With the help of these correlated tables, this research has designed 3-D traceability matrix and with the help of key logs mapping diagrams are designed too. Each design model has a unique key, which has three parts: module number, phase ID and diagram number. A tool is developed to create traceability matrix, user will input the keys for matrix. This system has been developed only for aspect-oriented systems so far but with a few modifications it can be extended to be applied on object-oriented systems as well.

5. KEY

Each design model has a unique key, which has three parts: module number, phase ID, and diagram number. These keys are used to create relationships between models.

M1_DP_DN

In the above expression M1 is module that in which module the given model is used, DP is the diagram phase or model phase, there is three phases first one is analysis phase, second one is early design phase and the third and final phase is design phase and shows the diagram phase that in which phase the given diagram lies. DN shows the diagram number in the relative module. In this research, gathering the data of module number, diagram phase and diagram number generates the metadata key. With these three things, metadata key is designed. Metadata key can be generated by the following sample:

Module no_Diagram phase_Diagram no

5.1 Equations

Each design model has a unique key, which has three parts: module number, phase ID and diagram number. Example:

Module 1_E_1.1

Where:

Module 1 is module number

E is early design

And 1.1 is Diagram number

Based on module ID diagrams are mapped.

Given a set of module M having a mapped set of keys K, a unique/distinct k is assigned to each diagram such that:

M1_DP_DN

DP= {A, E, D} Where; A=Analysis E=Early Design D=Design

M= set of modules m ϵ M

k=set of keys k ϵ K

Here K= {ki, kj, kk} Here, i ϵ A, j ϵ E and k ϵ D

Each module correlates with each level like M1 is related to A, E, D, similarly M2 is also related to A, E, D and so on. With each design level one-to- one and one-to-many function can be imposed.

For all modules from i=1 to n is related to A/E/D and each design level is associated with diagram number DN that is j=1 to k. its mathematical expression is given below:

$$\forall \overset{n}{\underset{i=1}{M}} \rightarrow (A/E/D) \rightarrow \overset{k}{\underset{j=1}{F}}$$

where:

I Module number

K It is in between one to many

J Diagram number

$$\forall \overset{n}{\underset{i=1}{M}} \rightarrow A \rightarrow \overset{k}{\underset{j=1}{F}}_{\rightarrow 1}$$

$$\forall \overset{n}{\underset{i=1}{M}} \rightarrow E \rightarrow \overset{k}{\underset{j=1}{F}}_{\rightarrow 2}$$

$$\forall \overset{n}{\underset{i=1}{M}} \rightarrow D \rightarrow \overset{k}{\underset{j=1}{F}}_{\rightarrow 3}$$

For module i=1 is related to A and each design level is associated with diagram number

DN that is j=1.

For module i=2 is related to E and each design level is associated with diagram number

DN that is j=2.

For module i=3 is related to D and each design level is associated with diagram number

DN that is j=3.

6. TABLES AND FIGURES

6.1 General

This research describes how the diagrams are mapped to each other and how those mapped diagrams generate traceability matrix. During the validation process Traceability Matrix (TM) is used as a table that links the demands required [3]. It has three levels: Analysis, Early design, and Design level. In these three levels, names of diagrams of each level are shown in Table 1. By using these three levels, diagrams are mapped. Keys show their uniqueness and correlation of each level.

Table 1. Diagram's levels

Level	UML	AODL
Analysis Mode	•Use case Diagrams	• Joinpoint Identification diagram • Join point Behavioural diagram
Early Design	• Sequence Diagrams • Communication diagrams • Domain Model • Object Model	• Aspect Design Model • Pointcut Model
Design Level	• Class Diagram •Composition Diagram • Deployment Diagram	•Pointcut Composition model

During the validation process TM is used as a table that links the demands required [3]. Traceability matrix guarantees the authenticity of all requirements being tested under standard protocols. In this research, the TM has been designed to check the validation of aspect oriented models and UML models. Designing the TM and mapping diagrams generate keys.

In this research, the TM has been designed of UML and Aspect oriented models, as shown below as generally,

In Table 2, generic data has been shown against each level. In the Analysis level we have A1, A2, A3 and A4, in Early design level E1, E2 are related to A1, E3 is related to A2, E4 is related to A3 and E4, E5 are related to A4, while in Design level there is D1, D2 related to A1 that are ultimately related to E1, E2 and D2, D3 are related to A2 that are ultimately related to E3, D1, D3 are related to A3 that are ultimately connected to E4 and D3, D4 are associated with A4 that is correlated with E4, E5.

In the above explanation, A shows the analysis level, E shows the early design level and D shows the design level.

For example, we have the following data:

Table 2. Data for generic traceability matrix

Analysis level	Early Design	Design level
A1	(A1) E1, E2	(A1) D1, D2
A2	(A2) E3	(A2) D2, D3
A3	(A3) E4	(A3) D1, D3
A4	(A4) E4, E5	(A4) D3, D4

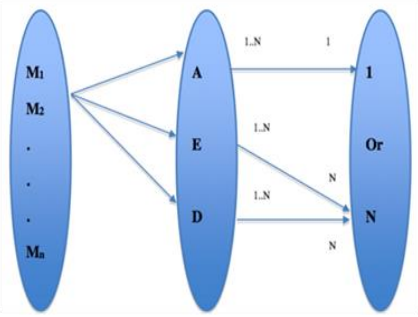
In Table 3, there are four analysis levels A1, A2, A3 and A4. There are four models in analysis level but in early design level there is five models E1, E2, E3, E4 and E5 that are associated to analysis level, in early design level there is E1 and E2 that correlates with A1, E3 correlates with A2, E4 correlates with A3 and E4 and E5 correlates with A4. Similarly, in design levels, there are four models D1, D2, D3 and D4. D1 and D2 correlates with A1, which is ultimately related to E1 and E2, D2 and D3 correlates with A2 that is ultimately related to E3, D1 and D3 correlates with A3 that is ultimately related to E4, D3 and D4 correlates with A4 that is ultimately related to E4 and E5Traceability matrix of this generic data is given below:

Table 3. Traceability matrix

	D1					D2					D3					D4				
	E1	E2	E3	E4	E5	E1	E2	E3	E4	E5	E1	E2	E3	E4	E5	E1	E2	E3	E4	E5
A1	√	√				√	√													
A2								√					√							
A3				√										√						
A4														√	√				√	√

6.2 Figure

Module state diagram in figure 1 showing that how a module is related to each level and what its relationship with that level.



Module no. Diagram phase Diagram no.

Figure 1. Module state diagram

7. CONCLUSIONS

Models can be mapped with the help of metadata key. Each design model has a unique key, which has three parts: module number, phase ID and diagram number. With the help of this key, 3D traceability matrix can be designed.

Following are the goals that are achieved during this research:

• Module state diagram has been showed that how a module is related to each level and what its relationship with that level is explained and the mathematical relationship also explains their association.

• Each design model has a unique key, which has three parts: module number, phase ID and diagram number. With the help of this key, 3D traceability matrix and mapping diagrams can be designed.

• The process used for creating data element mappings between two different data models is mapping. Mapping models identifies relationships of data. In this research mapping models are designed with the help of traceability matrix and keys generated with the help of three levels of designs i.e. Analysis level, Early Design level and Design level.

• Traceability matrix guarantees the authenticity of all requirements being tested under standard protocols. In this research traceability matrix has been designed to check the validation of software models.

• A tool is developed to check the traceability of matrix; tool takes inputs by user for each level and keys for matrix. Then matrix shows the traceability of each level.

• A tool can be developed that will automatically generate keys by inputting the models. The tool will automatically generate code of the software models. And the same technique can be applied on code for the evaluation of code. It can improve mapping between models and code that can enhance effective traceability of the requirements of the system.

REFERENCES

[1] Futrell R.T., Shafer L.I., D.F. (2002). Quality Software Project Management, Upper Saddle River, Prentice Hall PTR, NJ, USA.

[2] SantAnna A.C., Garcia C., Chavez C., Lucena A., Von Staa. (2003). On the reuse and maintenance of aspect-oriented software: an assessment framework, in *Proceedings of Brazilian Symposium on Software Engineering*, pp. 19-34.

[3] Z.F., Requirements traceability matrix, from *http://www.zf-laser.com*, accessed 25 Dec 2016.

[4] Schlageter G., Unland R., Wilkes W., Zieschang R., Maul G., Nagl M., Meyer R. (1988). Oops-an object-oriented programming system with integrated data management facility in Data Engineering, *Proceedings Fourth International Conference 1988 IEEE*, pp. 118-125. DOI: 10.1109/ICDE.1988.105453

[5] Bachmendo B., Hanenberg S., Herrmann S., Kniesel G. (2003). Aspect-oriented software development, *Proceedings of the Third German AOSD-Workshop of the SIG Object-Oriented Software Development*, German Informatics Society,2003.