Measuring Cohesion and Coupling of Sequence Diagram Using Program Slicing

Daljeet Singh and Sunint Kaur Khalsa

Department of Computer Science and Engineering, Guru Nanak Dev Engineering College
Gill Road, Ludhiana, Punjab, India 141006
diljitsingh007@gmail.com, sunintkaur@rediffmail.com

Abstract. This paper proposes a technique for the measurement of Cohesion and Coupling of Sequence Diagram using the Program Slicing. The sequence diagram contains the dynamic information of the Object Oriented system. In this work Sequence Dependency Graph (SDG) is generated from the states and scenarios of Sequence diagram. The SDG is then dynamically sliced taking various aspects into consideration. These slices can then be used to measure Cohesion and Coupling. The novelty of this approach is the direct measurement of Cohesion and Coupling of an object oriented system from Sequence diagram.

Keywords: Sequence Diagram, Program Slicing, Cohesion, Coupling.

1 Introduction

The UML Sequence Diagram refers to time dependent sequences of interactions between objects. They show the sequence of the messages. A Sequence Diagram has two dimensions: the vertical dimension represents time, and the horizontal dimension represents various instances. In UML, a message is a request for a service from one UML actor to another. In fig. 1 the Sequence Diagram is showing the various transactions of messages from one object to another object. The figure shows three objects A, B and C which are having flow of messages in various directions. Each message is given a message number for ease of understanding which is discussed in the following sections.

2 Related Work

The slicing of the Sequence diagram has been proposed in [1] and coupling metrics are analyzed in [2]. Pam Green [3] in there paper proposes a report that provides an overview of slice-based software metrics.Philip Samuel [4]presents a new approach and a novel methodology for test case generation based on UML sequence diagrams using Sequence Dependence Graph.

V.V. Das, E. Ariwa, and S.B. Rahayu (Eds.): SPIT 2011, LNICST 62, pp. 84–88, 2012.

3 Recognition of Scenarios and States of Sequence Diagram and Sequence Dependency Graph

A sample Sequence Diagram (SD) is shown in figure 1. It is converted into Sequence Dependency Graph (SDG) using various states and scenarios as shown in table 1.

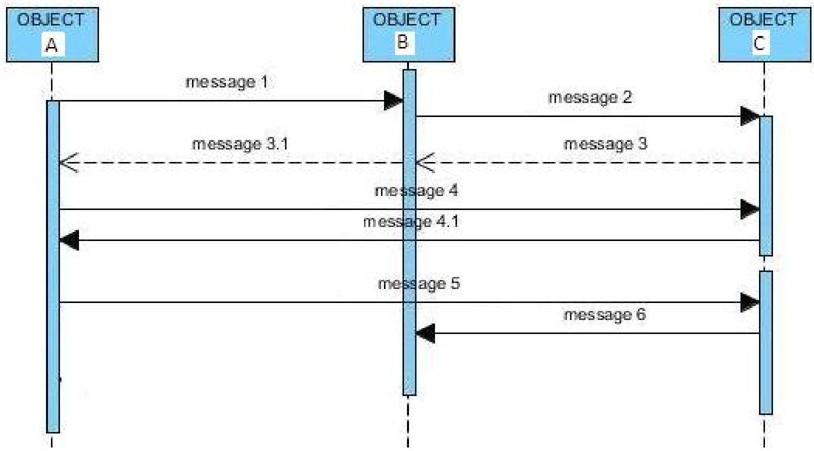

Fig. 1. Sample Sequence Diagram

Table 1. Scenarios with start and end state

Scenario 1	Scenario 2	Scenario 3
Start state x	Start state x	Start state x
S1 : (m1,a,b)	S5 :(m4,a,c)	S7 : (m5,a,c)
S2 : (m2 ,b ,c)	S6 : (m4.1,c,a)	S8 : (m6,c,b)
S3 : (m3 ,c ,b)		
S4 : (m3.1 ,b, a)		
End state x	End state x	End state z

The message requests the service from one UML actor to another. In table 1 different scenario were taken according to the sequence diagram in fig 1. If the starting and ending object of the scenario is A then the Start State and End State is X. If the starting and ending object of the scenario is B then the Start State and End State is Z and if the starting and ending object of the scenario is C then the Start State and End State is Y. The SD is then transformed into SDG by using the scenarios and states.This approach refers the one proposed in [5]. States S1, S2, S3, S4, S5, S6, S7, S8 represents the transition from one object to another as shown in figure 2a.

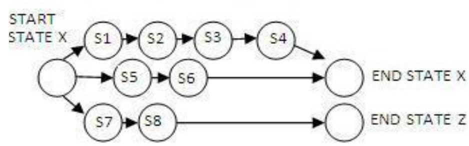

Fig. 2a. Sequence Dependency Graph

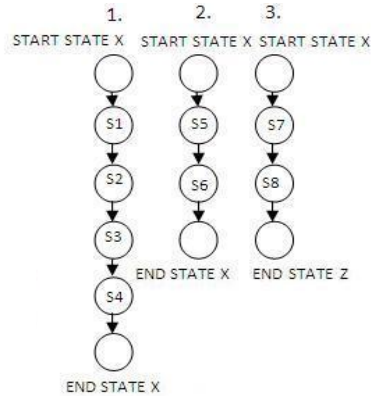

Fig. 2b. The Dynamic slices of SDG

4 Dynamic Slicing of SDG

Static Slicing is applied on the source code with no other information than the source code itself. This makes use of information about a particular execution of a program. A dynamic slice contains all statements that actually affect the value of a variable at a program point, for a particular execution of the program, rather than all statements that may have affected the value of a variable, at a program point, for any arbitrary execution of the program. Sequence diagrams shows the dynamic function and if we make the slices of the sequence dependencies by taking the criteria as the start and end state of the each scenario then in figure2(b) slice 1 , slice 2 , slice 3 is formulated from the fig 2(a) as the dynamic program slices of SDG. These slices can then be used to calculate cohesion and coupling in the coming sections.

5 Calculating Cohesion Using Slicing

This section details the calculation of Cohesion. There are three slices in figure 3 with their corresponding number of nodes as 4, 2, and 2 respectively. The total number of nodes by combining all the slices is called Module Size which is 8 as shown in fig 2b. The various terms used in the formulas are Size of slice (Common node in every slice), Number of slices (Total number of slices made), Min slice size (Number of nodes in the smallest slice), Max slice size (Number of nodes in the largest slice).

- Tightness is the number of nodes included in every slice compared to the number

  Tightness (M) = $\frac{|SL\text{int}|}{length(M)} = \frac{sizeofslice}{\text{mod}\,ulesize} = \frac{0}{8} = 0$

- Coverage is a comparison of the length of the slices to the length of the module.

$$\text{Coverage (M)} = \frac{1}{V\circ}\sum_{i=1}^{v\circ}\frac{SL}{length(M)} = \frac{sumofslices}{(no.ofslices)\times(modulesize)} = \frac{8}{3\times 8} = 0.33$$

- Min coverage is the ratio of the size of the smallest slice to the module size.

$$\text{Min coverage (M)} = \frac{1}{length(M)}\min i|SLi| = \frac{\min slicesize}{modulesize} = \frac{2}{8} = 0.25$$

- Max coverage is the ratio of the size of the largest slice to the module size.

$$\text{Max coverage (M)} = \frac{1}{length(M)}\max i|SLi| = \frac{\max slicesize}{modulesize} = \frac{4}{8} = 0.5$$

- Overlap is the average ratio of the number of the nodes in the intersection of all slices to the size of each slice.

$$\text{Overlap (M)}= \frac{1}{|V\circ|}\sum_{i=1}^{|v\circ|}\frac{|SL\text{int}|}{|SLi|} = \frac{\sum intersectionsize}{slicesize} \Big/ no.ofslices = \frac{0}{4}+\frac{0}{2}+\frac{0}{2} \Big/ 3 = \frac{0}{3} = 0$$

6 Calculation of Slice Based Coupling of SDG

This section explains slice-based coupling metrics of the sequence dependency graph as in [6]. Worked examples are shown as under with calculations. The coupling between two functions f and g can be calculated as:

$$\text{Coupling (f, g)} = \frac{FF(f,g)\times length(f) + FF(g,f)\times length(g)}{length(f)\times length(g)}$$

Table 2. Intra Slice Coupling between slice 1 and slice 2

Module (G)	Module Size	Module (F)	Common Nodes	Total
1	4	2	0	0
2	2	1	0	0

$$\text{Coupling}= \frac{0+0}{6} = 0$$

The slice based coupling is of two types: Inter slice Coupling and Intra Slice Coupling. Inter Slice Coupling is the coupling between the slices having different end states e.g. Slice 1 and Slice 3, Slice 2 and Slice 3. Intra Slice Coupling is the coupling between the slices having the same end state e.g. Slice 1 and Slice 2. The Final coupling is shown in table 2, 3 and 4.

Table 3. Inter Slice Coupling between slice 2 and slice 3

Module (G)	Module Size	Module (F)	Common Nodes	Total
2	2	3	0	0
3	2	2	0	0

$$\text{Coupling}= \frac{0+0}{4} = 0$$

Table 4. Inter Slice Coupling between slice 1 and slice 3

Module (G)	Module Size	Module (F)	Common Nodes	Total
1	4	3	0	0
3	2	1	0	0

$$\text{Coupling}=\frac{0+0}{6}=0$$

7 Conclusion

The method to calculate cohesion and coupling directly from the Sequence Diagram has been proposed in this paper. We are planning to extend this approach to other diagrams of UML to capture static and dynamic aspects of the system.

References

1. Noda, K., Kobayashi, T., Agusa, K.: Sequence Diagram Slicing, Grant-in-Aid or Scientific Research of MEXT Japan (2009)
2. Simon Allier, S., Vaucher, S., Dufour, B., Sahraoui, H.: Deriving Coupling Metrics from Call Graphs. Natural Sciences and Engineering Research Council of Canada (2010)
3. Green, P., Lane, P.C.R., Rainer, A., Scholz, S.: An Introduction to Slice-Based Cohesion and Coupling Metrics, Technical report no. 488 School of Computer Science, University of Hertfordshire (2009)
4. Samuel, P., Mall, R.: A Novel Test Case Design Technique Using Dynamic Slicing of UML Sequence Diagrams. e-Informatics Software Engineering Journal 2(1) (2008)
5. Sharma, M., Mall, R.: Automatic Test Case Generation from UML Models. In: 10th International Conference on Information Technology, pp. 196–201. IEEE Computer Society, Washington (2007)
6. Harman, M., Okuiawon, M., Sivagurunathan, B., Danicic, S.: Slice Based Measurement of Function Coupling. In: 18th International Software Quality Week (QW 1995), San Francisco, CA (1995)
7. Lallchandani, J.T., Mall, R.: Computation of dynamic slices for Object oriented concurrent programs. In: 12th Asia Pacific Software Engineering Conference, pp. 341–350. IEEE Computer Society, Taipei (2005)