

UML Based Visualization based on Test Coverage Metrics

Dongho Kim
*SE Lab, Dept. of CIC(Computer Information
Communication), Hongik University*
ray@selab.hongik.ac.kr

Index

1. Motivation

2. Related Works

1. Software Visualization
2. Software Test Coverage

3. Visualizing with Test Coverage Metrics

4. Conclusion

1. Motivation

- **In recent software, the size and complexity of software rapidly increase.**
- **Clients requires many requirement spec, rapid development, and high quality[1].**
- **As the software is complicated, software testing cost is limited although requiring high quality**
- **To solve this problem, it is required to do the software development and testing simultaneously.**

2. Related Works

- **Software Visualization Techniques**
 - automatically show software architecture and quality indicator through a tool-chain.
- **Test coverage**
 - Previous Test Coverage based mostly white-box techniques (White Box) based on procedural language.
 - It is not suitable to use case-based, object-oriented test case coverage measurement, and test case generation techniques

3. Visualizing with Test Coverage Metrics

Use case test cases coverage	Message sequence			State test case coverage	Object test case coverage	Method test case coverage
	Dialogue Test case coverage	MLU test case coverage	Reusable pattern test case coverage			
use case 1 (Product)	D1	MLU 1	RP1	s0	Object 1	m1
				Object 2	m2	
		MLU 2	RP2	s1	Object 3	m4
				m5		
		MLU 3	RP3	s1	Object 3	m3
				s2	Object 5	m4
s5	Object 1	m5				
s0	Object 3	m10				
use case2 (Coins)	D2	MLU 4	RP4	s2,	Object 3	m6
			RP5	s3		m8
			RP6	s4		m9
		MLU 5	RP7	s1	Object 3	m7
				s2	Object 5	m8
				s3	Object 3	m3
MLU 6	RP10	s2	Object 5	m4		
		s3	Object 3	m3		
		s4	Object 5	m5		
use case3 (Verify Deposit)	D3	MLU 5	RP7	s2	Object 5	m8
				s3	Object 3	m9
	D4	MLU 6	RP9	s2	Object 3	m8
				s4	Object 5	m7
						m9
						m8



simple example : vending machine

3. Visualizing with Test Coverage Metrics (cont.)

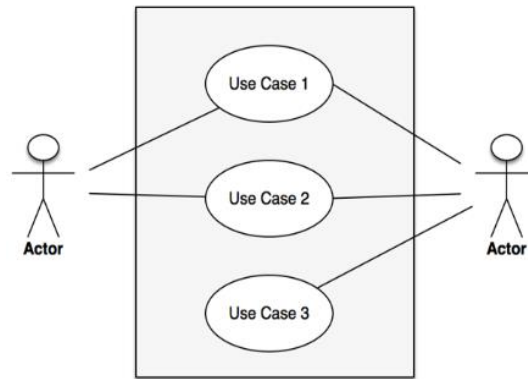


Figure 1. Use Case Test Coverage

Table 1. Use Case' Test Case Coverage

Test Case	Use Case Name	Testing Type	Pre-Condition	Input Value	Post-Condition	Expected Value
tc1	UC1	UC	Pre-C1	In1	Post-C1	Out1
tc2	UC2	UC	Pre-C2	In2	Post-C2	Out2
tc3	UC3	UC	Pre-C3	In3	Post-C3	Out3

3. Visualizing with Test Coverage Metrics (cont.)

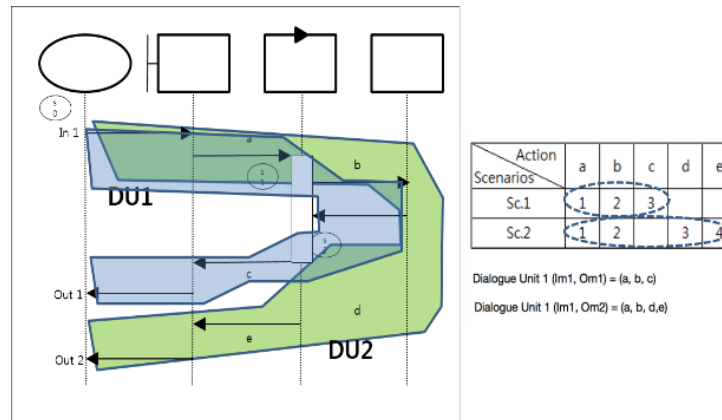


Figure 2. Dialogue Test Coverage

Table 2. 'Dialogue' Test Case Metrics

Test Case	Dialogue Name	Testing Type	Pre-Condition	Input Value	Post-Condition	Expected Value
tc1	DU1	Dialogue	S0	In1	S0	Om1
tc3	DU2	Dialogue	S0	In3	S0	Om2

3. Visualizing with Test Coverage Metrics (cont.)

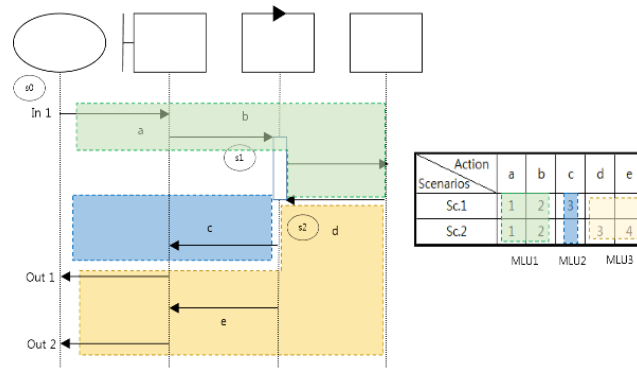


Figure 3. MLU (Maximum Linear Unit) Test Coverage

Table 3. MLU' Test Case Metrics

Test Case	MLU Name	Testing Type	Pre-Condition	Input Value	Post-Condition	Expected Value
tc1	M1	MLU	S0	In1	S1	M2 M3
tc2	M2	MLU	S1	None	S0	Om1
tc3	M3	MLU	S2	None	S0	Om2

4. Conclusion

- **The size and complexity of software are rapidly increased, but budget is limited.**
- **Moreover, the requirements for the quality is increasing.**
- **As a result, a lot of research have been in progress for the test method of high efficiency**

4. Conclusion (cont.)

- **In this paper, we suggest this visualization way through the test coverage and test case metrics.**
- **We define each step of test case metrics and level of abstraction test case.**
- **Although this is accomplished by first critical test cases of higher level, it is guarantee the Maximal Test Coverage with the Minimal Test Case.**
- **Future study will automatically extract level of test case, and apply to various cases.**

Thank You!!!