

# <Software Modeling & Analysis>

시스템 테스트 대응 보고서

- 그놈! Clone Checker -

Project Team

T4

Date

2016-06-09

---

Team Information

201411258 강태준

201411265 김서우

201411321 홍유리

## <Contents>

1. 문서 분석 대응
2. Category-partition testing 대응
3. Pairwise Testing 대응
4. Brute Force Testing 대응

## 1. 문서 분석 대응

### - Stage 1000\_T4\_ver6

#	1
문서 분석 결과	3페이지, (전략) 가중치를 두어 프로그램을 실행하면 유사도를 비율(%)로 사용자에게 알려주는 것을 목표로 한다 : STR에서는 백분율을 점수로 수정하였다고 하였으나, 실제로는 이전 버전(Stage1000_T4_ver5)에서 변하지 않았음
대응	최종 유사도는 백분율(%)로 계산하기로 이미 그 전에 구현을 해 놓았었고 실제로 이전의 코드를 실행시켰을 때, Show Detail 버튼을 열어보면 최종 유사도가 백분율(%)로 나옴을 알 수 있다. 실제로 T4_5_System_Test.pdf의 아래쪽 Pairwise 부분에 첨부되어 있는 캡처를 보면 백분율이 쓰여 있음이 보인다. 따라서 수정하지 않음

#	2
문서 분석 결과	4페이지, 이전 버전과 비교하였을 때 내용이 변경된 점이 보임, 수정 내역이 없어 쉽게 알기 어려움
대응	수정 내역을 Stage 1000이 아닌 2030에서 적는 것이라 생각하여 수정하지 않음

#	3
문서 분석 결과	4페이지, 실제 프로그램 사용 시 7개 파일 분석에 4분 소요
대응	50개 파일 분석에 5분 소요로 수정

#	4
문서 분석 결과	8페이지, 실행 환경에 IDE를 넣는 것은 적절하지 않아 보임
대응	OS : Windows 7, 개발 언어 : JAVA 1.8.0_77 로 수정

#	5
문서 분석 결과	10페이지, 이전에 언급된 적 없는 Use-Case 인 Draw Result가 등장함
대응	Draw Result( UseCase )를 삭제

#	6
문서 분석 결과	11페이지, 다이어그램에 Draw Result가 존재 하지 않음
대응	다이어그램 수정

### - Stage 2030\_T4\_ver4

#	1
문서 분석 결과	5페이지, Ref. # 표가 이전 버전 문서를 반영하고 있지 않음
대응	반영하고 있으므로 수정하지 않음

#	2
문서 분석 결과	7페이지, (A)에서 정의한 Use-Case를 완전하게 표현하고 있지 않음
대응	수정된 Use-Case를 바탕으로 수정함

#	3
문서 분석 결과	21,22페이지, Show Detail 항목의 Typical Courses of Events의 설명 중, 3번과 4번의 순서가 잘못 되어 있음. 실제로는 txt 파일을 닫는 것과 버튼이 비활성화 되는 것에 상관 관계가 없음.
대응	<ol style="list-style-type: none"> <li>1. (A) Actor가 Show Detail 버튼을 누른다.</li> <li>2. (S) Show Detail 버튼을 비활성화 시킨다.</li> <li>3. (S) 유사도 검사의 세부 내용이 담긴 .txt 파일의 내용을 보여준다.</li> </ol> 로 수정함

#	4
문서 분석 결과	<p>23페이지.</p> <p>A. MainController에 실제 코드 상에 존재하는 start_analyze():int가 존재하지 않음</p> <p>B. AnalysisSystem의 make_Detail()의 반환형은 boolean이나, 이것이 적절히 반영되어 있지 않음.</p> <p>C. GUI 클래스가 반영되어 있지 않음</p>
대응	<p>클래스 다이어그램 수정</p>

#	5
문서 분석 결과	<p>25페이지, Variable 항목이 UML과 일치하지 않음</p> <p>A. count는 UML 상에서 attribute이나, 표에서는 operation으로 표현됨</p> <p>B. Variable() 메서드가 표현되어 있지 않음</p>
대응	수정함

#	6
문서 분석 결과	26~28페이지, 내부 시스템 동작에 대해서는 작성하지 않았음.
대응	System Sequence Diagram은 System 자체를 블랙박스로 보고 시퀀스를 쓰는 것이라 생각하여 내부 시스템의 동작에 대해서는 쓰지 않기로 하였음. 즉 수정하지 않음

#	7
문서 분석 결과	26페이지, Input Path 이후 Setting Files이 실행되어야 하나, 그것에 관한 언급이 없음
대응	6번과 같은 이유

#	8
문서 분석 결과	CK.STC.321 Test Case와는 달리, 실제 프로그램은 창을 닫기 전에 버튼이 비활성화 됨.
대응	경로 입력을 성공하고 Start 버튼을 누르고, 분석 완료 알림창이 뜬 뒤에 Show Detail 버튼을 누르면 <b>Show Detail 버튼이 비활성화</b> 되는지 확인한다. 로 수정함

#	9
문서 분석 결과	Actor가 user인 것에 대해서만 use-case를 operation에 대응시켰으나, 이 과정에서 system이 actor인 use-case에 대해서는 operation의 이름을 어떻게 바꾸는지 명세하지 않았음.
대응	operation의 이름을 바꾼다는 것이 이해가 안되어서 수정 못함

#	10
문서 분석 결과	System이 actor인 use-case에 대해, operation name을 표기하지 아니하고 서술함
대응	이해가 안되어서 수정 못함

#	11
문서 분석 결과	클래스 다이어그램보다 도메인 모델이 거대하고 상세하게 표현되어 있음
대응	수정함

## - Stage 2040\_T4\_ver3

#	1
문서 분석 결과	3~11 페이지, A. Input Path는 실질적으로 경로 유효성을 판단하고, Setting Files을 호출하는 역할을 하나, 표에는 그러한 서술이 존재하지 않음 B. Setting Files : 함수의 지역변수 저장은 SourceCode 내부의 변수가 아닌, Function 내부의 변수에 저장하는 것으로 보이나, 표의 서술은 SourceCode 내부의 변수에 저장한다고 되어 있음 C. Start Analyze : OSP Stage 1000 ver6에서 Start Analyze Code로 변경된 Function임.
대응	A, B : 수정함 C : 2041은 Define Real Use Cases로 UseCase에 대한 내용을 적는 것이라 판단. 본 프로그램상 UseCase에는 Start Analyze, Functional Requirement에는 Start Analyze Code라 되어 있음 따라서 이름이 Start Analyze라 하는 것이 맞다고 판단하여 수정 안함

#	2
문서 분석 결과	16 ~ 19페이지, 전체적으로 실제 작성된 코드를 적절히 반영하고 있지 않음
대응	반영하고 있다고 생각하여 수정하지 않았음

#	3
문서 분석 결과	18페이지, Start Analyze 그림에서 코드 상에는 조건문이 존재하나, 그림에선 표현되지 않음.
대응	이해가 안되어서 수정 못함

#	4
문서 분석 결과	19페이지, Show Detail 그림에서, 메모장을 호출하여 txt 파일의 내용을 보여주는 부분이 명확하게 나와있지 않음
대응	System함수를 이용하는 것이어서 따로 표현하지는 않았음

## - Stage 2050\_T4\_ver2

#	1
문서 분석 결과	3~5페이지, Cross Reference의 Start Analyze를 Start Analyze Code로 변경이 필요함
대응	Start Analyze Code는 Functional Requirements의 이름이고, Start Analyze는 Usecase이름이 맞으므로 수정 안함

#	2
문서 분석 결과	6~19페이지, A. Input, Output의 형태가 메서드로 나와 있지만, 실제 표의 내용은 변수 형임 B. analyzeCode : 배열의 사이즈를 선정한 이유가 명시되지 않음. C. SourceCode, Variable, Loop, Conditional, Function 항목의 메서드는 모두 동일한 기능(클래스 변수 초기화)을 수행하나, 실제 코드 상에서는 클래스 내부의 변수들의 접근 제어자가 모두 public, 또는 default이고, Getter, Setter 또한 존재하지 않음
대응	A, B : 수정함 C : 변수 초기화를 하는 점은 맞지만, 코드 상에서 public으로 접근하기에 편하다 생각하여 제어자를 private으로 지정하지 않았음.

#	3
문서 분석 결과	19~20페이지, Pre-Conditions와 Post-Conditions가 반대로 적혀 있음
대응	수정함

#	4
문서 분석 결과	20~24페이지, 절대 경로를 사용하여, 특정 컴퓨터에서만 테스트 할 수 있도록 해두었고, 따라서 리눅스 기반의 CTIP 환경에서는 에러를 발생시킨다. 상대 경로를 이용하는 코드로 변경하여, CTIP 환경에서 테스트 가능하도록 하는 것이 적절하다 판단함.
대응	수정하지 않음



#	5
문서 분석 결과	29페이지, 바디 모음을 비교한다고 서술하였으나, 구체적인 비교 방법이 서술되어 있지 않음
대응	소스 코드의 전체 함수 body를 하나로 모으고 모든 변수의 이름을 "attr"로, 모든 함수의 이름을 "oper"로 바꾼 상태에서 두 함수 body 모음을 Levenshtein distance 알고리즘을 통하여 비교한다. 유사도에 따라서 점수를 부여한다. 라고 수정함

## - Stage 2060\_T4\_ver2

문서 분석 결과 문제점은 없었으나, 2050을 수정하면서 내용을 수정함

## 2. Category-partition testing 대응

#	1
Testing 결과	<p>Single 범칙에 의해 생성된 테스트 케이스는 모두 통과 하였으나, Struct 또는 typedef에 의해서 생성된 변수 테스트에서는 정확하지 않은 결과가 발생하였다.</p> <ul style="list-style-type: none"> <li>■ 이는, 변수에 대한 명세만 있고, 구조체에 대해서는 어떻게 검사한다는 specification이 존재하지 않았기 때문.</li> </ul>
대응	<p>구조체 또한 하나의 자료형으로 Parsing을 진행하는데, 변수 검사와 Specification이 동일하므로 따로 적을 내용이 없다고 판단하여 따로 적지 않음.</p> <p>변수 테스트에서 정확하지 않은 결과가 발생한 부분에 대해서는 코드를 수정함으로써 옳게 고침</p> <p>[기존 코드]</p> <pre> if(exist_flag==0) {     sc[index].type_list.add(temp_str);     sc[index].type_list.add(temp_str+"*");     sc[index].type_list.add(temp_str+"**");     sc[index].type_list.add(temp_str+"[]");     sc[index].type_list.add(temp_str+"[][]"); } </pre> <p>[수정 코드]</p> <pre> if(exist_flag==0) {     temp_str = temp_str.trim();     sc[index].type_list.add(temp_str);     sc[index].type_list.add(temp_str+"*");     sc[index].type_list.add(temp_str+"**");     sc[index].type_list.add(temp_str+"[]");     sc[index].type_list.add(temp_str+"[][]"); } </pre>

### 3. Pairwise Testing 대응

#	1
Testing 결과	<p>테스트에 앞서</p> <p>본 프로그램은 변수, 함수, 조건문, 반복문에 대한 개수 확인을 제대로 진행하지 못하므로, 이를 고려하여 이번 Testing에서는 유사도 검사에서 정확한 동작을 수행하는지에 대한 확인을 목적으로 하였다.</p> <p>Pairwise testing 테스트 중 변수나 함수, 조건문 등의 컴포넌트를 정확히 카운트 하지 못하는 현상을 발견하여, 대안으로 유사도 검사를 정확히 수행하는지에 대한 case를 만들었다. 그러나 모든 case에 대해 유사도 검사를 정확하게 실행하는 case가 존재하지 않았다.</p> <p>위 결과 보고서에서, 함수 내용 유사도 항목이 존재하지 않는 것을 확인할 수 있다.</p>
대응	<p>1. 변수 개수 확인 : Category-partitioning Testing 에서 수정한 코드를 바탕으로 총 개수를 정확히 셀 수 있게 수정</p> <p>2. 동일한 변수의 개수 계산</p> <p>[기존 코드]</p> <pre>//지역변수에서 동일한 변수의 개수 모음 for(Function f1: s1.func){     for(Function f2: s1.func){         for(Variable v1 : f1.var_list){             for(Variable v2 : f2.var_list){                 if(v1.name.equals(v2.name) &amp;&amp; v1.data_type.equals(v2.data_type) &amp;&amp; f1.name.equals(f2.name)){                     //System.out.println("지역변수"+f1.name+":"+v1.data_type+" "+v1.name+"===="+f2.name+":"+v2.data_type+" "+v2.name);                     sum++;                 }             }         }     } } </pre> <p>[수정 코드]</p> <pre>//지역변수에서 동일한 변수의 개수 모음 for(Function f1: s1.func){     for(Function f2: s2.func){         for(Variable v1 : f1.var_list){             for(Variable v2 : f2.var_list){                 if(v1.name.equals(v2.name) &amp;&amp; v1.data_type.equals(v2.data_type) &amp;&amp; f1.name.equals(f2.name)){                     //System.out.println("지역변수"+f1.name+":"+v1.data_type+" "+v1.name+"===="+f2.name+":"+v2.data_type+" "+v2.name);                     sum++;                 }             }         }     } } </pre> <p>동일한 이름을 갖는 지역 변수의 개수를 세는 부분에서 같은 소스코드의 지역변수를 비교하여 개수가 정확하지 않아서 수정함.</p> <p>3. 함수 내용 유사도 항목</p> <p>[기존 코드]</p> <pre>result = "2) 함수 이름 유사 검사 =&gt; 동일한 이름의 함수 개수 : " + Double.toString(score[1][3]) + "개"; temp = " ∴ 검사 결과 =&gt; " + Double.toString(score[1][4]) + "점"; out.write(result); out.newLine(); out.write(temp); out.newLine(); result = "3) 함수 Body 유사 검사 =&gt; " + Double.toString(score[1][5]) + "%"; temp = " ∴ 검사 결과 =&gt; " + Double.toString(score[1][6]) + "점"; result = " ∴ 함수 유사도 검사 최종 점수 =&gt; " + Double.toString(score[1][7]) + "점"; out.write(result); out.newLine(); </pre>

**[수정 코드]**

```
result = "2) 함수 이름 유사 검사 => 동일한 이름의 함수 개수 : " + Double.toString(score[1][3]) + "개";
temp = " ∴ 검사 결과 => " + Double.toString(score[1][4]) + "점";
out.write(result);
out.newLine();
out.write(temp);
out.newLine();
result = "3) 함수 Body 유사 검사 => " + Double.toString(score[1][5]) + "%";
temp = " ∴ 검사 결과 => " + Double.toString(score[1][6]) + "점";
out.write(result);
out.newLine();
out.write(temp);
out.newLine();
result = " ∴ 함수 유사도 검사 최종 점수 => " + Double.toString(score[1][7]) + "점";
out.write(result);
out.newLine();
```

세부 내용 파일에 함수 Body 유사 검사 내용을 입력하도록 수정

## 4. Brute Force Testing 대응

Test Case No#	1
문제	경로 입력을 하지 아니하고 Input 버튼 클릭, '경로 입력 실패' 팝업 표시를 해야하는데, '.C 파일 말고 다른 파일이 존재' 팝업 표시
원인	GUI 부분과 Setup System 부분을 수정하는 과정에서 문제 발생
대응	수정 완료

Test Case No#	2
문제	존재하지 않는 경로를 입력하고 Input 버튼 클릭, '경로 입력 실패' 팝업 표시를 해야하는데, '.C 파일 말고 다른 파일이 존재' 팝업 표시
원인	GUI 부분과 Setup System 부분을 수정하는 과정에서 문제 발생
대응	수정 완료

Test Case No#	3
문제	C 파일이 하나만 존재하는 경로를 입력하고 Input 버튼 클릭, '경로 입력 실패' 팝업 표시를 해야하는데 '유효하지 않은 경로' 팝업 표시
원인	GUI 부분과 Setup System 부분을 수정하는 과정에서 문제 발생
대응	수정 완료

Test Case No#	4
문제	프로그램을 1회 정상적으로 동작 후 존재하지 않는 경로를 입력하고 Input 버튼 클릭, '경로 입력 실패' 팝업 표시 후 Start, Show X_File, Show Detail 버튼 비활성화 되어야 하는데. 모든 버튼 활성화 상태, 이전 분석 결과 출력
원인	경로 입력 실패 시 모든 버튼의 비활성화를 해두지 않음
대응	수정 완료

Test Case No#	5
문제	Show X_File 버튼을 연속 클릭, Show X_File 버튼 1회 클릭 후 비 활성화, 팝업 화면의 확인 버튼 클릭 전까지 활성화 상태 유지되며 클릭 시, 팝업 화면이 무제한으로 생성됨.
원인	버튼이 누름과 동시에 비활성화 해주지 않아서 발생
대응	수정 완료

Test Case No#	6
문제	C 파일만 존재하는 올바른 경로를 입력, Input 버튼 클릭 후 Start 버튼 클릭, 소스 코드 분석 후 Show X_File, Show Detail 버튼 활성화 되어야 하는데, 에러 발생
원인	분석하는 과정에서의 문제
대응	수정 완료

Test Case No#	7
문제	빈 경로 입력 후 Input 버튼 클릭, '경로 입력 실패' 팝업 표시되어야 하는데, '경로 탐색 성공' 팝업 표시
원인	GUI 부분과 Setup System 부분을 수정하는 과정에서 문제 발생
대응	수정 완료

Test Case No#	8
문제	7개의 C 파일이 있는 경로 입력 후 Input, Start 버튼 클릭, 분석에 약 7.7초 소요(Activity 2064)되어야 하는데, Input에 5.18초, Start에 약 4분30초 소요
원인	분석 과정의 내용들이 늘어나서 발생
대응	수정 완료

Test Case No#	9
문제	입력된 C 파일에 변수의 개수 94개, for문의 개수 9개, do - while 문 존재, 변수의 개수 94개, for문의 개수 9개 검출, do-while문 검출 되어야 하는데, 변수의 개수 93개, for문의 개수 8개로 검출, do-while문 검출하지 못함.
원인	do-while문은 애초에 분석을 하지 않음
대응	do-while문 분석은 분석한다는 언급이 애초에 없었으므로 수정 안함. 나머지 부분은 코드 수정

Test Case No#	10
문제	동일한 변수의 개수를 파악할 때, (전체 변수 개수 > 동일한 변수 개수) 이어야 하는데, (전체 변수 개수 < 동일한 변수 개수) 로 분석
원인	동일한 소스코드의 변수끼리 비교하여 항상 큰 개수로 나눔 <pre>//지역변수에서 동일한 변수의 개수 찾기 for(Function f1: s1.func){     for(Function f2: s1.func){         for(Variable v1 : f1.var_list){             for(Variable v2 : f2.var_list){                 if(v1.name.equals(v2.name) &amp;&amp; v1.data_type.equals(v2.data_type) &amp;&amp; f1.name.equals(f2.name)){                     //System.out.println("지역변수"+f1.name+" "+v1.data_type+" "+v1.name+"=="+f2.name+" "+v2.data_type+" "+v2.name);                     sum++;                 }             }         }     } } //지역변수에서 동일한 변수의 개수 찾기 for(Function f1: s1.func){     for(Function f2: s2.func){         for(Variable v1 : f1.var_list){             for(Variable v2 : f2.var_list){                 if(v1.name.equals(v2.name) &amp;&amp; v1.data_type.equals(v2.data_type) &amp;&amp; f1.name.equals(f2.name)){                     //System.out.println("지역변수"+f1.name+" "+v1.data_type+" "+v1.name+"=="+f2.name+" "+v2.data_type+" "+v2.name);                     sum++;                 }             }         }     } } }</pre>
대응	<pre>//지역변수에서 동일한 변수의 개수 찾기 for(Function f1: s1.func){     for(Function f2: s2.func){         for(Variable v1 : f1.var_list){             for(Variable v2 : f2.var_list){                 if(v1.name.equals(v2.name) &amp;&amp; v1.data_type.equals(v2.data_type) &amp;&amp; f1.name.equals(f2.name)){                     //System.out.println("지역변수"+f1.name+" "+v1.data_type+" "+v1.name+"=="+f2.name+" "+v2.data_type+" "+v2.name);                     sum++;                 }             }         }     } } }</pre>

Test Case No#	11
문제	원본 파일 A와 원본과 유사한 파일 B를 입력함, X_File로 A가 검출 되어야 하는데, X_File로 B가 검출
원인	X_File을 찾는 데에 평균치로 찾기 때문에 1:1 로는 한계치가 있음
대응	수정 불가

Test Case No#	12
문제	main() 함수만 존재하는 C 파일 입력, 정상적으로 분석해야 하는데, 에러 발생
원인	수정 과정에서 에러가 발생하지 않아서 잘 모르겠음
대응	수정 불가

Test Case No#	13
문제	내용이 존재하는 C 파일 2개와 내용이 없는 C 파일 2 개를 입력, 정상적으로 분석해야 하는데, 에러 발생
원인	수정 과정에서 에러가 발생하지 않아서 잘 모르겠음
대응	수정 불가

Test Case No#	14
문제	분석에 시간이 많이 소요, 분석 진행중을 안내하는 팝업 표시가 되고, 프로그램의 정상적인 동작이 되어야 하는데, 분석 완료 팝업이 뜨기 이전까지 프로그램이 정지, 모든 버튼 입력 불가능
원인	GUI Frame을 띄우는 과정에서 발생하는 한계
대응	수정 불가