

OOPT Stage 1000

<Plan and Elaboration>

Software Modeling & Analysis

소프트웨어 모델링 및 분석

보고서 Version. 7

Team. T1

201111388 조연호

201211374 이창오

201211379 장종훈

201314196 양동혁

Stage 1000. Plan and Elaboration

1. Activity 1001. Define Draft Plan	3
2. Activity 1002. Create Preliminary Investigation Report.....	4
3. Activity 1003. Define Requirements.....	6
4. Activity 1004. Record Terms in Glossary.....	9
5. Activity 1005. Implement Prototype	9
6. Activity 1006. Define Draft System Architecture.....	10
7. Activity 1007. Define Business Use Case.....	11
8. Activity 1008. Define Business Concept Model	18
9. Activity 1009. Define System Test Case	19
10. Activity 1010. Refine Plan	21

Activity 1001. Define Draft Plan

1. Motivation

시중의 Clone Checker 프로그램들은 대부분 소스 코드 파일 두 개를 비교한다. 반대로 다수의 소스 코드 파일을 한 번에 검사하는 Clone Checker 프로그램은 그 결과를 보여주는 데 있어서 단순한 결과만을 출력할 뿐 각각 소스 코드 파일의 인과 관계를 보여주는 것이 미흡하다. 그러나 정작 중요한 것은 어떤 소스 코드 파일을 기준으로 그것과 비슷한 소스 코드 파일들을 찾아내어 복제와 연관된 인과 관계를 찾아내는 것이다. 이러한 인과 관계를 직관적이며 편하게 볼 수 있는 프로그램이 필요하다고 판단하였다.

2. Project Objectives

하나의 소스 코드 파일을 기준으로 설정하고, 다른 소스 코드 파일과의 일치율을 알기 쉽게 보여주는 Clone Checker 프로그램을 개발하여 비교 정확도를 높이는 것을 목표로 한다.

3. Functional Requirements

- A. Select Folder
- B. Start
- C. Analyze File
- D. Change Annotation
- E. Analyze Annotation
- F. Analyze Line
- G. Delete Annotation
- H. Divide Code Line
- I. Delete Printf
- J. Find Type
- K. Analyze Function
- L. Analyze Variable
- M. Analyze Preprocessor
- N. Change Center
- O. Calculate File
- P. Calculate Sync-Rate

- Q. Calculate Line Sync-Rate
- R. Calculate Function Sync-Rate
- S. Calculate Variable Sync-Rate
- T. Calculate Preprocessor Sync-Rate
- U. Calculate Annotation Sync-Rate
- V. Display Sync

4. Non-Function Requirements

- A. 조작하기 쉬운 인터페이스
- B. 한 눈에 이해 가능한 결과 출력
- C. C 소스 코드 파일만 비교
- D. 마우스 왼쪽 클릭으로만 조작

5. Resource Estimation

- A. Human Resource : 4명
- B. Project Duration : 3개월(12주)
- C. Human Efforts(Man-Month) : 12
- D. Cost : 식대 1,200,000원 (5,000원/일 × 4명 × 5일/주 × 12주)

6. Other Information

A. Future Version

다양한 시각에서 엄격한 비교 기준들을 추가하여 두 소스 코드 파일 내에서 프로그램 구현에 사용한 동일 알고리즘을 찾아낼 수 있는 Clone Checker 프로그램으로 확장한다.

Activity 1002. Create Preliminary Investigation Report

1. Alternative Solutions

- A. 프로그램 외주 제작
- B. 과제 대행 서비스
- C. 현재 존재하는 Clone Checker 프로그램

2. Project Justification(Business Demands)

- A. Cost : 식대 외의 다른 비용이 필요하지 않아 다른 대안보다 훨씬 저렴하다.
- B. Duration : 3개월(12주)
- C. Risk : UML 및 JAVA 숙련도 부족, 개인적인 사정 등
- D. Effect

전체 소스 코드 파일들의 비교가 아닌 기준 소스 코드 파일 하나에 초점을 맞추어 진행하여 원본 파일을 제공한 학생과 상습적으로 복제를 하는 학생을 쉽게 찾아낼 수 있다.

3. Risk Management

Risk	Probability	Significance	Weight
UML 숙련도 부족	4	5	20
JAVA 숙련도 부족	5	5	25
Clone Checker 원리 이해도 부족	4	4	16
아르바이트	3	4	12
동아리 및 학생회 활동	3	4	12
팀원 간 의견 충돌	5	4	20
연애 사업	1	8	8

4. Risk Reduction Plan

Risk	Way of Reduction
UML 숙련도 부족	적극적인 수업 참여와 관련 도서 및 인터넷을 통하여 학습한다
JAVA 숙련도 부족	적극적인 수업 참여와 관련 도서 및 인터넷을 통하여 학습한다
Clone Checker 원리 이해도 부족	관련 도서 및 인터넷을 통하여 학습한다.
아르바이트	각자 조정할 수 있는 스케줄은 조정한다.
동아리 및 학생회 활동	각자 조정할 수 있는 스케줄은 조정한다.
팀원 간 의견 충돌	팀장이 강력한 리더십을 발휘한다.
연애 사업	연애 문제로 인한 감정 기복이 생기지 않도록 연애 사업을 하는 팀원을 적극적으로 도와준다.

5. Market Analysis

창의적인 Clone Checker 프로그램으로써 아직까지 비슷한 대체제가 없어 시장성이 높다.

6. Other Managerial Issues

A. 2016년 6월 안에 프로젝트가 종료되어야 한다.

B. 다음 학기부터 곧바로 이용할 수 있도록 완벽성을 추구해야 한다.

Activity 1003. Define Requirements

1. Functional Requirements

Ref	Function	Description
R.1	Select Folder	비교할 소스 코드 파일들이 들어있는 폴더를 선택한다.
R.2	Start	지정된 경로의 파일들에 대한 객체들을 생성한다.
R.3	Analyze File	소스 코드 파일을 분석하여 일치율 계산에 필요한 정보로 가공한다.
R.3.1	Change Annotation	소스 코드에서 '/**/'으로 주석 처리된 줄을 '//'으로 변경한다.
R.3.2	Analyze Annotation	소스 코드 파일의 주석의 개수를 분석하여 결과값을 저장한다.
R.3.3	Analyze Line	소스 코드 파일의 라인 수를 분석하여 결과값을 저장한다.
R.3.4	Delete Annotation	소스 코드 파일 분석에 불필요한 주석 처리된 부분을 삭제한다.
R.3.5	Divide Code Line	소스 코드 파일에서 한 줄에 여러 개의 코드가 있는 경우 ';'을 기준으로 나눈다.
R.3.6	Delete Printf	소스 코드 파일 분석에 불필요한 printf문의 괄호 안의 부분을 삭제한다.
R.3.7	Find Type	구조체 또는 typedef로 새로 선언된 타입을 찾는다.
R.3.8	Analyze Function	소스 코드 파일의 함수의 개수와 이름을 분석하여 결과값을 저장한다.
R.3.9	Analyze Variable	소스 코드 파일의 변수의 개수와 이름을 분석하여 결과값을 저장한다.
R.3.10	Analyze Preprocessor	소스 코드 파일의 전처리기의 개수와 이름을 분석하여 결과값을 저장한다.
R.4	Change Center	비교 기준이 되는 소스 코드 파일을 변경한다.
R.5	Calculate File	분석하여 가공한 정보를 비교하여 일치율을 계산한다.

OOPT Stage 1000 <Plan and Elaboration>

R.5.1	Calculate Line Sync-Rate	분석한 라인 수를 비교하여 일치율을 계산한다.
R.5.2	Calculate Function Sync-Rate	분석한 함수의 개수와 이름을 비교하여 일치율을 계산한다.
R.5.3	Calculate Variable Sync-Rate	분석한 변수의 개수와 이름을 비교하여 일치율을 계산한다.
R.5.4	Calculate Preprocessor Sync-Rate	분석한 전처리기의 개수와 이름을 비교하여 일치율을 계산한다.
R.5.5	Calculate Annotation Sync-Rate	분석한 주석의 개수를 비교하여 일치율을 계산한다.
R.6	Display Sync	기준 소스 코드 파일과 비교할 소스 코드 파일의 일치율과 유사 항목을 출력한다.

Ref	Function	Category
R.1	Select Folder	Evident
R.2	Start	Evident
R.3	Analyze File	Hidden
R.3.1	Change Annotation	Hidden
R.3.2	Analyze Annotation	Hidden
R.3.3	Analyze Line	Hidden
R.3.4	Delete Annotation	Hidden
R.3.5	Divide Code Line	Hidden
R.3.6	Delete Printf	Hidden
R.3.7	Find Type	Hidden
R.3.8	Analyze Function	Hidden
R.3.9	Analyze Variable	Hidden
R.3.10	Analyze Preprocessor	Hidden
R.4	Change Center	Evident
R.5	Calculate File	Hidden
R.5.1	Calculate Line Sync-Rate	Hidden
R.5.2	Calculate Function Sync-Rate	Hidden
R.5.3	Calculate Variable Sync-Rate	Hidden
R.5.4	Calculate Preprocessor Sync-Rate	Hidden
R.5.5	Calculate Annotation Sync-Rate	Hidden
R.6	Display Sync	Evident

2. Performance Requirements

- A. 소스 코드 파일 분석 작업은 5초 이내로 수행되어야 한다.
- B. 기준 소스 코드 파일 변경 시 2초 이내로 비교 결과가 변경되어야 한다.

3. Operating Environments

Microsoft Windows 7 이상

4. Develop Environments

- A. 운영체제 : Windows 7, 10
- B. CPU : Intel
- C. IDE : Eclipse
- D. 개발 언어 : JAVA
- E. UML 툴 : StarUML

5. Interface Requirements

A. 메인 화면

i. 폴더 선택 버튼

화면 중앙에 위치하며 클릭 시 폴더 선택 화면을 출력한다.

ii. '시작하기' 버튼

화면 하단에 위치하며 클릭 시 소스 코드 파일 분석 후 비교 결과 화면을 출력한다.

B. 비교 결과 화면

i. 기준 소스 코드 파일 이름

화면 중앙에 위치한다.

ii. 비교 소스 코드 파일 이름

일치율이 52% 이상이면 일치율에 따라 크기가 달리 출력된다.

iii. 일치율과 유사 항목

비교 소스 코드 파일 이름에 마우스 커서를 위치할 경우 일치율과 유사 항목을 출력한다.

C. 그 외 메시지 출력은 대화상자(Dialog) 활용

6. Other Requirements

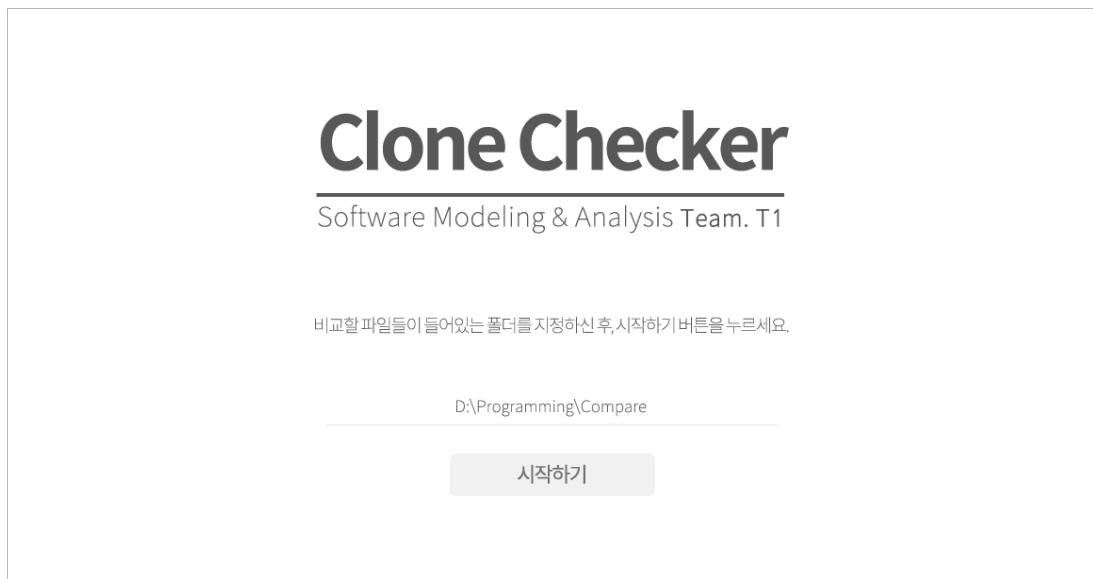
- A. 새로운 폴더를 지정하기 위해서는 프로그램을 종료하고 다시 실행해야 한다.

Activity 1004. Record Terms in Glossary

Term	Description	Remarks
소스 코드	컴파일 가능 여부와 상관 없이 프로그래밍 언어로 작성된 코드	
소스 코드 파일	파일 크기가 0바이트 초과인 소스 코드가 담긴 .c 확장자 파일 파일명 양식은 '학번_이름.c' 형태여야 한다.	

Activity 1005. Implement Prototype

1. 메인 화면



2. 비교 결과 화면

Clone Checker

Software Modeling & Analysis Team. T1

오른쪽 List에서 **기준 소스 코드 파일**을 선택하세요.

List

- 광희
- 김속
- 김영철
- 김태호
- 민아
- 박명수
- 서지영
- 송은이
- 안영미
- 예린
- 유재석
- 이경규
- 정준하
- 정형돈
- 태연
- 하하
- 현아
- 혜리

Clone Checker

Software Modeling & Analysis Team. T1

종료하기

List

- 광희
- 김속
- 김영철
- 김태호
- 민아
- 박명수
- 서지영
- 송은이
- 안영미
- 예린
- 유재석
- 이경규
- 정준하
- 정형돈
- 태연
- 하하
- 현아
- 혜리

일치율 72% 변수 개수 · 라인 수 유사

Activity 1006. Define Draft System Architecture



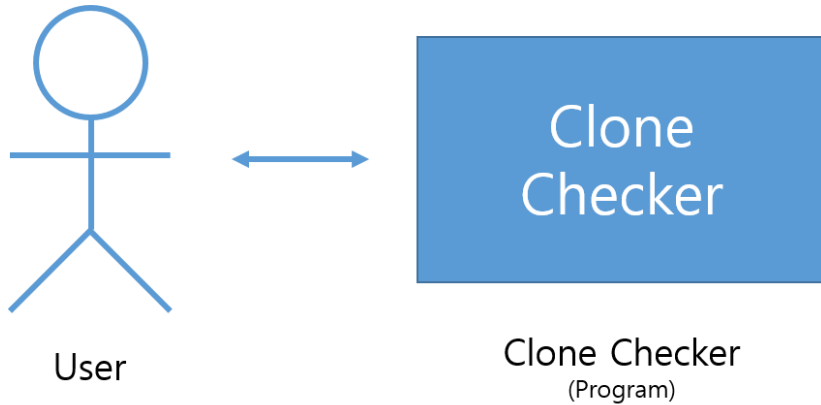
User



Clone Checker
(Program)

Activity 1007. Define Business Use Case

1. Define System Boundary



2. Identify and Describe Actors

A. User

User는 다음과 같은 행동을 할 수 있다.

i. 메인 화면

비교하고자 하는 100개 이하인 소스 코드 파일들이 들어있는 폴더를 선택하고, '시작하기' 버튼을 클릭하여 유사한 소스 코드 파일이 있는지 비교할 수 있다.

ii. 비교 결과 화면

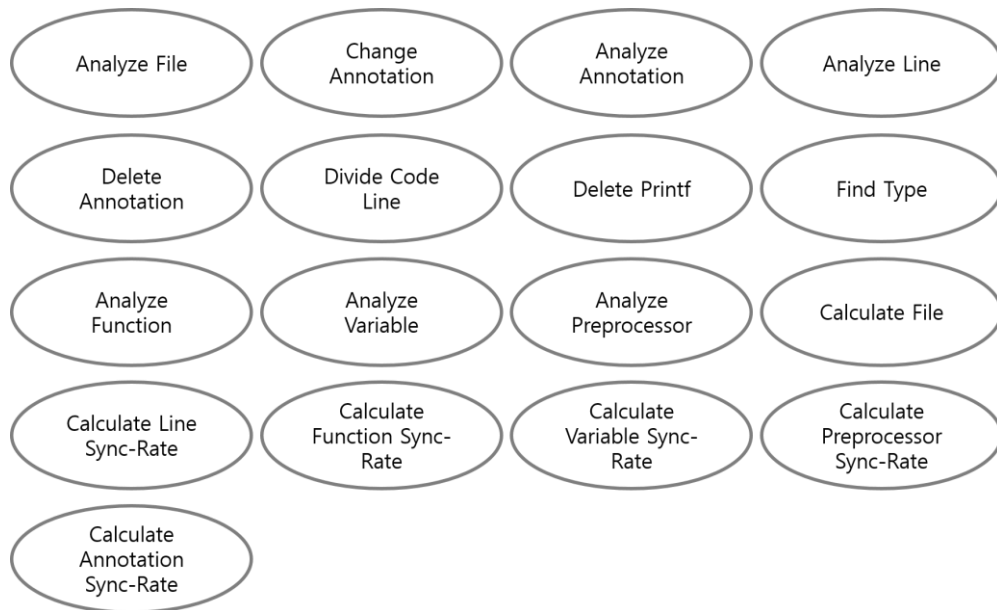
1. 소스 코드 파일 이름에 마우스를 올려 일치율과 유사 항목을 확인한다.
2. 태그 클라우드 또는 오른쪽 리스트에서 소스 코드 파일 이름을 클릭하여 기준을 변경한다.

3. Identify Use-Case

A. Use-Cases by Actor-Based



B. Use-Cases by Event-Based



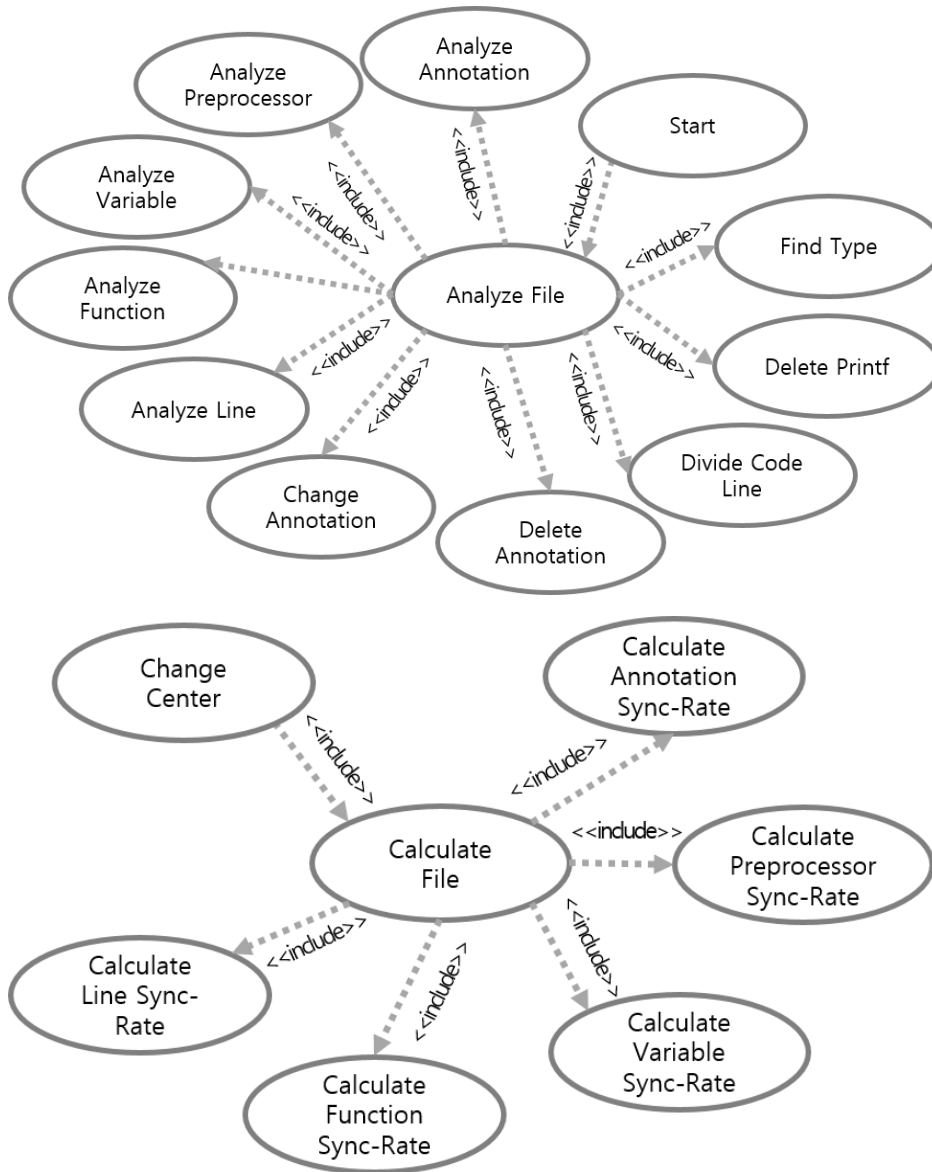
4. Allocate System Functions into Related Use-Cases

Ref	Function	Use-Case Number & Name
R.1	Select Folder	1. Select Folder
R.2	Start	2. Start
R.3	Analyze File	3. Analyze File
R.3.1	Change Annotation	4. Change Annotation
R.3.2	Analyze Annotation	5. Analyze Annotation
R.3.3	Analyze Line	6. Analyze Line
R.3.4	Delete Annotation	7. Delete Annotation
R.3.5	Divide Code Line	8. Divide Code Line
R.3.6	Delete Printf	9. Delete Printf
R.3.7	Find Type	10. Find Type
R.3.8	Analyze Function	11. Analyze Function
R.3.9	Analyze Variable	12. Analyze Variable
R.3.10	Analyze Preprocessor	13. Analyze Preprocessor
R.4	Change Center	14. Change Center
R.5	Calculate File	15. Calculate File
R.5.1	Calculate Line Sync-Rate	16. Calculate Line Sync-Rate
R.5.2	Calculate Function Sync-Rate	17. Calculate Function Sync-Rate
R.5.3	Calculate Variable Sync-Rate	18. Calculate Variable Sync-Rate
R.5.4	Calculate Preprocessor Sync-Rate	19. Calculate Preprocessor Sync-Rate
R.5.5	Calculate Annotation Sync-Rate	20. Calculate Annotation Sync-Rate
R.6	Display Sync	21. Display Sync

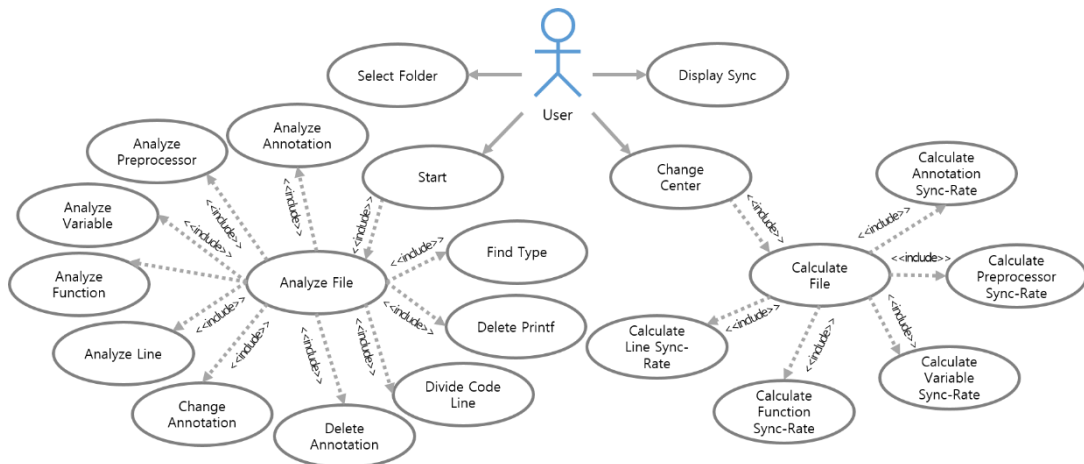
5. Categorize Use-Cases

Use-Case Number & Name	Category
1. Select Folder	Primary
2. Start	Primary
3. Analyze File	Primary
4. Change Annotation	Primary
5. Analyze Annotation	Primary
6. Analyze Line	Primary
7. Delete Annotation	Primary
8. Divide Code Line	Primary
9. Delete Printf	Primary
10. Find Type	Primary
11. Analyze Function	Primary
12. Analyze Variable	Primary
13. Analyze Preprocessor	Primary
14. Change Center	Primary
15. Calculate File	Primary
16. Calculate Line Sync-Rate	Primary
17. Calculate Function Sync-Rate	Primary
18. Calculate Variable Sync-Rate	Primary
19. Calculate Preprocessor Sync-Rate	Primary
20. Calculate Annotation Sync-Rate	Primary
21. Display Sync	Primary

6. Identify Relationships between Use-Cases



7. Draw a Use-Case Diagram



8. Describe Use-Cases

Use-Case Name	Description
1. Select Folder	비교할 소스 코드 파일들이 들어있는 폴더를 선택한다.
Actor	
User	

Use-Case Name	Description
2. Start	지정된 경로의 파일들에 대한 객체들을 생성하고, 비교 결과 화면을 출력한다.
Actor	
User	

Use-Case Name	Description
3. Analyze File	소스 코드 파일을 분석하여 일치율 계산에 필요한 정보로 가공한다.
Actor	
None	

Use-Case Name	Description
4. Change Annotation	소스 코드 내 /* */ 주석 안의 문장을 // 문장으로 바꾼다.
Actor	
None	

Use-Case Name	Description
5. Analyze Annotation	소스 코드 파일의 주석의 개수를 분석하여 결과값을 저장한다.
Actor	
None	

Use-Case Name	Description
6. Analyze Line	소스 코드 파일의 라인 수를 분석하여 결과값을 저장한다.
Actor	
None	

Use-Case Name	Description
7. Delete Annotation	소스 코드 파일의 주석을 삭제한다.
Actor	
None	

OOPT Stage 1000 <Plan and Elaboration>

Use-Case Name	Description
8. Divide Code Line	소스 코드 파일에서 한 줄에 여러 개의 코드가 있는 경우 ';'을 기준으로 행을 나눈다.
Actor	
None	

Use-Case Name	Description
9. Delete Printf	Printf문의 괄호 안의 내용을 삭제한다.
Actor	
None	

Use-Case Name	Description
10. Find Type	구조체 또는 typedef로 새로 선언된 타입을 찾는다.
Actor	
None	

Use-Case Name	Description
11. Analyze Function	소스 코드 파일의 함수의 개수와 이름을 분석하여 결과값을 저장한다.
Actor	
None	

Use-Case Name	Description
12. Analyze Variable	소스 코드 파일의 변수의 개수와 이름을 분석하여 결과값을 저장한다.
Actor	
None	

Use-Case Name	Description
13. Analyze Preprocessor	소스 코드 파일의 전처리의 개수와 이름을 분석하여 결과값을 저장한다.
Actor	
None	

Use-Case Name	Description
14. Change Center	비교 기준이 되는 소스 코드 파일을 변경한다.
Actor	
User	

OOPT Stage 1000 <Plan and Elaboration>

Use-Case Name	Description
15. Calculate File	분석하여 가공한 정보를 비교하여 일치율을 계산한다.
Actor	
None	

Use-Case Name	Description
16. Calculate Line Sync-Rate	분석한 라인 수를 비교하여 일치율을 계산한다.
Actor	
None	

Use-Case Name	Description
17. Calculate Function Sync-Rate	분석한 함수의 개수와 이름을 비교하여 일치율을 계산한다.
Actor	
None	

Use-Case Name	Description
18. Calculate Variable Sync-Rate	분석한 변수의 개수와 이름을 비교하여 일치율을 계산한다.
Actor	
None	

Use-Case Name	Description
19. Calculate Preprocessor Sync-Rate	분석한 전처리기의 개수와 이름을 비교하여 일치율을 계산한다.
Actor	
None	

Use-Case Name	Description
20. Calculate Annotation Sync-Rate	분석한 주석의 개수를 비교하여 일치율을 계산한다.
Actor	
None	

Use-Case Name	Description
21. Display Sync	기준 소스 코드 파일과의 일치율 및 유사 항목을 출력한다.
Actor	
User	

9. Rank Use-Cases

Use-Case Number & Name	Rank
1. Select Folder	High
2. Start	High
3. Analyze File	High
4. Change Annotation	High
5. Analyze Annotation	High
6. Analyze Line	High
7. Delete Annotation	High
8. Divide Code Line	High
9. Delete Printf	High
10. Find Type	High
11. Analyze Function	High
12. Analyze Variable	High
13. Analyze Preprocessor	High
14. Change Center	High
15. Calculate File	High
16. Calculate Line Sync-Rate	High
17. Calculate Function Sync-Rate	High
18. Calculate Variable Sync-Rate	High
19. Calculate Preprocessor Sync-Rate	High
20. Calculate Annotation Sync-Rate	High
21. Display Sync	High

Activity 1008. Define Business Concept Model

Display	Files	Analyze	Calculate

Activity 1009. Define System Test Case

1. Functional Requirements Test Case

Ref	Function	Test Case
R.1	Select Folder	폴더 선택 화면이 나타나고, 정상적으로 선택되었는지 확인한다.
R.2	Start	'시작하기' 버튼을 눌렀을 때 Files 인스턴스가 실제 소스 코드 파일 개수만큼(최대 100개) 생성되는지 확인한다. 파일 분석 및 계산, 비교결과 화면 출력이 기능 실행되는지 확인한다.
R.3	Analyze File	비교 결과 출력 화면이 나오는지 확인한다.
R.3.1	Change Annotation	소스 코드 파일의 주석 형태만 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
R.3.2	Analyze Annotation	소스 코드 파일의 주석을 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
R.3.3	Analyze Line	소스 코드 파일의 라인 개수만 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
R.3.4	Delete Annoation	소스 코드 파일의 주석 개수만 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
R.3.5	Divdie Code Line	소스 코드 파일의 한 줄로 같이 작성된 코드의 개수를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
R.3.6	Delete Printf	소스 코드 파일의 'printf()' 메소드로 출력된 코드의 개수를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
R.3.7	Find Type	소스 코드 파일의 구조체 또는 typedef로 새로 선언된 타입을 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
R.3.8	Analyze Function	소스 코드 파일의 함수를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
R.3.9	Analyze Variable	소스 코드 파일의 변수를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
R.3.10	Analyze Preprocessor	소스 코드 파일의 전처리기를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.

OOPT Stage 1000 <Plan and Elaboration>

		다.
R.4	Change Center	기준 소스 코드 파일 이름이 변경되는지 확인한다.
R.5	Calculate File	비교 소스 코드 파일 이름에 마우스 커서를 올리면 일치율과 유사 항목이 출력되는지 확인한다.
R.5.1	Calculate Line Sync-Rate	소스 코드 파일의 라인 개수만 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
R.5.2	Calculate Function Sync-Rate	소스 코드 파일의 함수를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
R.5.3	Calculate Variable Sync-Rate	소스 코드 파일의 변수를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
R.5.4	Calculate Preprocessor Sync-Rate	소스 코드 파일의 전처리기를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
R.5.5	Calculate Annotation Sync-Rate	소스 코드 파일의 주석을 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다.
R.6	Display Sync	마우스 커서가 올라간 소스 코드 파일의 비교율이 정상적으로 출력되는지 확인한다.

2. Non-Functional Requirements Test Case

Category	Test Case
Interface	<ul style="list-style-type: none"> ● 태그 클라우드 출력 화면에서 출력되는 글씨가 겹치지 않는지 확인한다. ● 태그 클라우드 출력 화면에서 일치율과 소스 코드 파일 이름의 크기가 비례하는지 확인한다. ● 태그 클라우드 출력 화면에서 소스 코드 파일 이름에 마우스 커서를 올릴 경우 일치율과 유사 항목이 정상적으로 출력되는지 확인한다. ● 태그 클라우드 출력 화면에서 소스 코드 파일 이름 클릭 시 기준이 변경되는지 확인한다. ● 리스트 출력 화면에서 소스 코드 파일 이름 클릭 시 기준이 변경되는지 확인한다. ● 리스트 출력 화면에서 소스 코드 파일 이름이 리스트 가로 길이를 넘지 않는지 확인한다. ● 키보드 입력으로 작동되는지 확인한다. ● 마우스 오른쪽 클릭으로 작동되는지 확인한다.
Speed	<ul style="list-style-type: none"> ● 소스 코드 분석 시간이 5초 이내인지 확인한다. ● 일치율과 유사 항목 계산 시간이 2초 이내인지 확인한다. ● 비교 결과 화면이 8초 이내인지 확인한다. ● 기준 소스 코드 파일 변경 후 2초 이내에 결과가 바뀌는지 확인한다.

Activity 1010. Refine Plan

1. Project Scope

비교 작업과 결과 출력이 쾌적한 Clone Checker

다수의 소스 코드 파일을 비교하면 시간은 오래 걸리고, 결과 자료의 양도 방대하다. 많은 양의 결과를 확인하는 것은 사람으로 하여금 거부감이 들며 이해하는 데 적지 않은 시간이 걸린다. 따라서 비교 결과가 거부감 없이 명확하고 한 눈에 알아 볼 수 있는 쾌적한 Clone Checker 프로그램을 개발한다.

2. Project Objectives

하나의 소스 코드 파일을 기준으로 설정하고, 다른 소스 코드 파일과의 일치율을 알기 쉽게 보여주는 Clone Checker 프로그램을 개발하여 비교 정확도를 높이는 것을 목표로 한다.

3. Functional Requirements

- A. Select Folder
- B. Start
- C. Analyze File
- D. Change Annotation
- E. Analyze Annotation
- F. Analyze Line
- G. Delete Annotation
- H. Divide Code Line
- I. Delete Printf
- J. Find Type
- K. Analyze Function
- L. Analyze Variable
- M. Analyze Preprocessor
- N. Change Center
- O. Calculate File
- P. Calculate Sync-Rate
- Q. Calculate Line Sync-Rate
- R. Calculate Function Sync-Rate
- S. Calculate Variable Sync-Rate
- T. Calculate Preprocessor Sync-Rate
- U. Calculate Annotation Sync-Rate
- V. Display Sync

4. Performance Requirements

- A. 소스 코드 파일이 100개 이하로 들어있는 폴더를 선택해야 한다.
- B. 소스 코드 파일 분석 작업은 5초 이내로 수행되어야 한다.
- C. 기준 소스 코드 파일 변경 시 2초 이내로 비교 결과가 변경되어야 한다.

5. Operating Environment

Microsoft Windows 7 이상

6. User Interface Requirements

알아보기 쉬운 Interface를 통해 특별한 설명 없이 프로그램을 수행하고, 간결한 결과 화면을 통해 사용자에게 결과를 정확하게 전달한다.

7. Other Requirements

C 프로그래밍 언어에서만 사용 가능하다.

8. Resources

A. Human Resource : 4명

B. Project Duration : 3개월(12주)

C. Human Efforts(Man-Month) : 12

D. Cost : 식대 1,200,000 원 (5,000 원/일 × 4명 × 5일/주 × 12주)

OOPT Stage 1000 <Plan and Elaboration>

9. Scheduling

Stage	Phase/Activity	Schedule (Week)																
		1	2	3	4	5	6	7	8	9	10	11	12					
1000 Plan & Elaborate	1001. Define Draft Plan	█																
	1002. Create Preliminary Investigation Report	█																
	1003. Define Requirements	█																
	1004. Record Terms in Glossary		█															
	1005. Implement Prototype		█															
	1006. Define Draft System Architecture		█															
	1007. Define Business Use Case		█															
	1008. Define Business Concept Model		█	█														
	1009. Define System Test Case		█	█														
	1010. Refine Plan		█	█														
2000 Build	2010. Revise Plan			█														
	2020. Synchronize Artifacts			█														
	2030 Analyze	2031. Define Essential Use Case			█													
		2032. Refine Use Case Diagrams			█													
		2033. Refine Conceptual Model			█													
		2034. Refine Glossary			█													
		2035. Define System Sequence Diagrams			█													
		2036. Define Operation Contracts			█													
		2037. Define State Diagrams			█													
	2040 Design	2041. Define Real Use Cases					█	█	█									
		2042. Define Reports, UI and Storyboards					█	█	█									
		2043. Refine System Architecture					█	█	█									
		2044. Define interaction Class Diagrams					█	█	█									
		2045. Define Design Class Diagrams					█	█	█									
		2046. Define Database Schema					█	█	█									
	2050 Construct	2051. Implement Class & interface Definition							█	█								
		2052. Implement Methods							█	█								
		2053. Implement Windows									█	█						
		2054. Implement Reports									█	█						
		2055. Implement DB schema										█	█					
2056. Write Test Code												█	█					
2060 Test	2061. Unit Testing												█	█	█			
	2062. Integration Testing												█	█	█			
	2063. System Testing												█	█	█			
	2064. Performance Testing												█	█	█			
	2065. Acceptance Testing												█	█	█			
	2066. Documentation Testing												█	█	█			