

OOPT Stage 1000

<Plan and Elaboration>

Software Modeling & Analysis

소프트웨어 모델링 및 분석

보고서 Version. 1

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..... | 5 |
| 4. Activity 1004. Record Terms in Glossary..... | 8 |
| 5. Activity 1005. Implement Prototype | 8 |
| 6. Activity 1006. Define Draft System Architecture..... | 9 |
| 7. Activity 1007. Define Business Use Case..... | 9 |
| 8. Activity 1008. Define Business Concept Model | 16 |
| 9. Activity 1009. Define System Test Case | 17 |
| 10. Activity 1010. Refine Plan | 18 |

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 비교율을 알기 쉽게 보여주는 Clone Checker 프로그램을 개발하여 비교 정확도를 높이는 것을 목표로 한다.

3. Functional Requirements

- A. 초기화
- B. 폴더 지정
- C. 분석 시작
- D. 결과 출력
- E. 비교 기준 소스 코드 파일 변경

4. Non-Function Requirements

- A. 설명이 필요 없을 정도의 조작하기 쉬운 인터페이스
- B. 한 눈에 이해 가능한 결과 출력
- C. 다른 언어의 소스 파일도 비교 가능하도록 하는 확장성

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 | Init | 프로그램을 초기화한다. |
| R.2 | Display Main | 폴더를 지정할 수 있는 버튼과 '시작하기' 버튼이 제공되는 메인 화면을 출력한다. |
| R.2.1 | Select Folder | 비교할 소스 코드 파일들이 들어있는 폴더를 지정한다. |
| R.3 | Analyze Files | 비교할 소스 코드 파일들을 분석하여 일치율 계산에 필요한 정보로 가공한다. |
| R.3.1 | Analyze Num Of Line | 폴더 안에 들어있는 소스 코드 파일들의 라인 수를 분석한다. |
| R.3.2 | Analyze Num Of Function | 폴더 안에 들어있는 소스 코드 파일들의 함수 개수를 분석한다. |
| R.3.3 | Analyze Num Of Variable | 폴더 안에 들어있는 소스 코드 파일들의 변수 개수를 분석한다. |

OOPT Stage 1000 <Plan and Elaboration>

| | | |
|---------|--------------------------------|--|
| R.3.4 | Analyze Num Of Global Variable | 폴더 안에 들어있는 소스 코드 파일들의 전역 변수 개수를 분석한다. |
| R.3.5 | Analyze Num Of Local Variable | 폴더 안에 들어있는 소스 코드 파일들의 지역 변수 개수를 분석한다. |
| R.3.6 | Analyze Num Of Static Variable | 폴더 안에 들어있는 소스 코드 파일들의 Static 변수 개수를 분석한다. |
| R.3.7 | Analyze Num Of Brace | 폴더 안에 들어있는 소스 코드 파일들의 중괄호 개수를 분석한다. |
| R.3.8 | Analyze Preprocessor | 폴더 안에 들어있는 소스 코드 파일들의 전처리를 분석한다. |
| R.3.8.1 | Analyze Num Of Preprocessor | 폴더 안에 들어있는 소스 코드 파일들의 전처리의 개수를 분석한다. |
| R.3.8.2 | Analyze Name Of Preprocessor | 폴더 안에 들어있는 소스 코드 파일들의 전처리의 이름을 분석한다. |
| R.3.9 | Analyze Num Of Annotation | 폴더 안에 들어있는 소스 코드 파일들의 주석 개수를 비교한다. |
| R.4 | Display Result | 첫 번째 소스 코드 파일을 기준으로 태그 클라우드형 비교 결과 화면을 출력한다. |
| R.4.1 | Compare Info | 분석하여 가공한 정보를 활용(비교)하여 일치율을 계산한다. |
| R.4.2 | Display Sync | 일치율과 유사 항목을 출력한다. |
| R.4.3 | Change Center | 비교 기준이 되는 소스 코드 파일을 변경하여 다시 비교한다. |

| Ref | Function | Category |
|---------|--------------------------------|----------|
| R.1 | Init | Hidden |
| R.2 | Display Main | Evident |
| R.2.1 | Select Folder | Evident |
| R.3 | Analyze Files | Hidden |
| R.3.1 | Analyze Num Of Line | Hidden |
| R.3.2 | Analyze Num Of Function | Hidden |
| R.3.3 | Analyze Num Of Variable | Hidden |
| R.3.4 | Analyze Num Of Global Variable | Hidden |
| R.3.5 | Analyze Num Of Local Variable | Hidden |
| R.3.6 | Analyze Num Of Static Variable | Hidden |
| R.3.7 | Analyze Num Of Brace | Hidden |
| R.3.8 | Analyze Preprocessor | Hidden |
| R.3.8.1 | Analyze Num Of Preprocessor | Hidden |

OOPT Stage 1000 <Plan and Elaboration>

| | | |
|---------|------------------------------|---------|
| R.3.8.2 | Analyze Name Of Preprocessor | Hidden |
| R.3.9 | Analyze Num Of Annotation | Hidden |
| R.4 | Display Result | Evident |
| R.4.1 | Compare Info | Hidden |
| R.4.2 | Display Sync | Evident |
| R.4.3 | Change Center | 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 지정 버튼

화면 중앙에 위치하며 클릭 시 폴더 선택 화면을 출력한다.

ii. 비교 시작 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

기준 소스 코드 파일 이름을 중심으로 일치율에 따라 크기가 달리 출력된다.

iii. 일치율과 유사 항목

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

iv. 이전 화면 버튼

화면 하단에 위치하며 클릭 시 메인 화면을 출력한다..

C. 그 외 메시지 출력은 대화상자(Dialog) 활용

6. Other Requirements

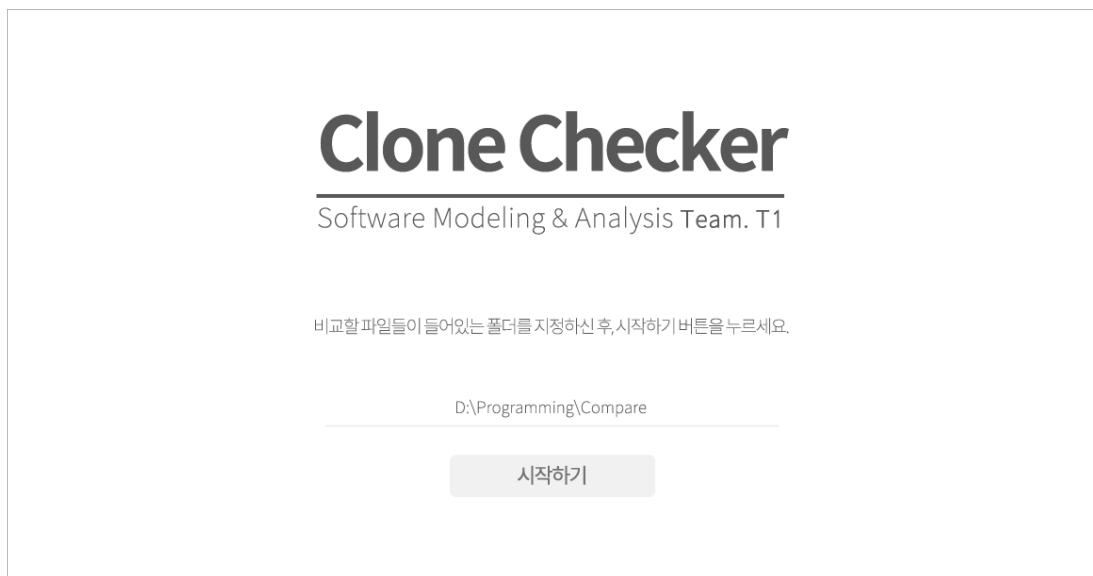
A. 비교 결과(일치율, 유사 항목)를 한 눈에 알아볼 수 있도록 비교 결과 출력 화면이 직관적이어야 한다.

Activity 1004. Record Terms in Glossary

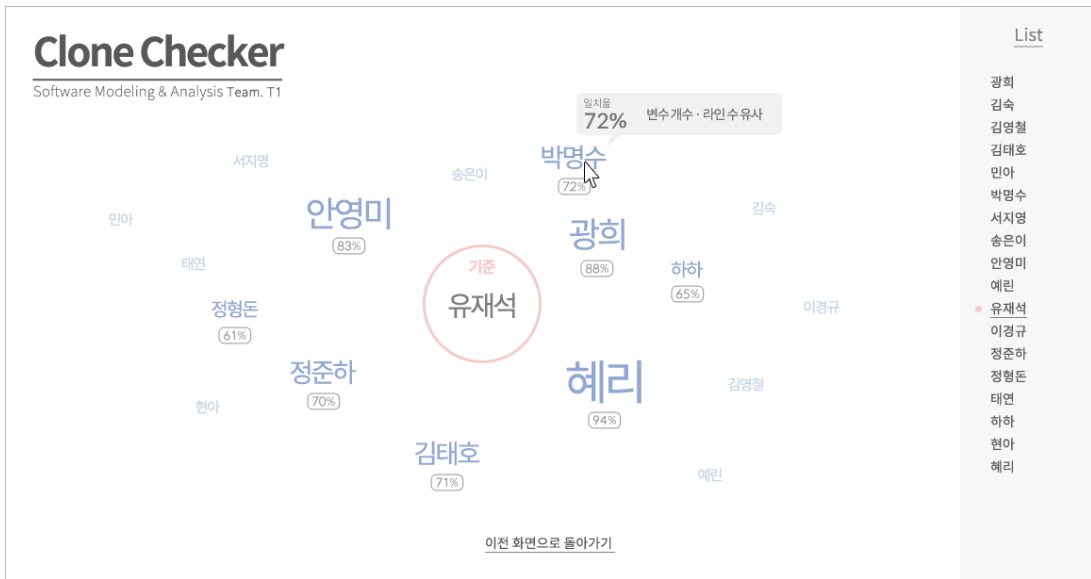
| Term | Description | Remarks |
|-------|----------------------|---------|
| Clone | 비교 결과 일치율 85% 이상인 파일 | |

Activity 1005. Implement Prototype

1. 메인 화면



2. 비교 결과 화면



Activity 1006. Define Draft System Architecture



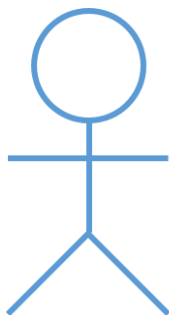
User



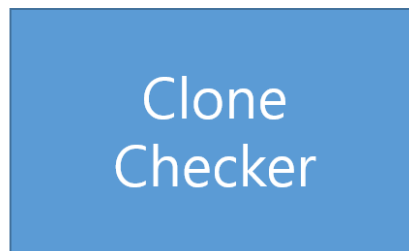
Clone Checker
(Program)

Activity 1007. Define Business Use Case

1. Define System Boundary



User



Clone Checker
(Program)

2. Identify and Describe Actors

A. User

User는 다음과 같은 행동을 할 수 있다.

i. 메인 화면

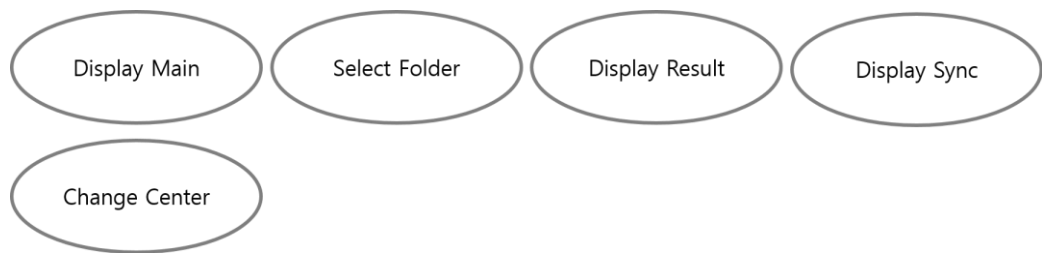
비교하고자 하는 소스 코드 파일들이 들어있는 폴더를 지정하고, '시작하기' 버튼을 클릭하여 유사한 소스 코드 파일이 있는지 비교할 수 있다.

ii. 비교 결과 화면

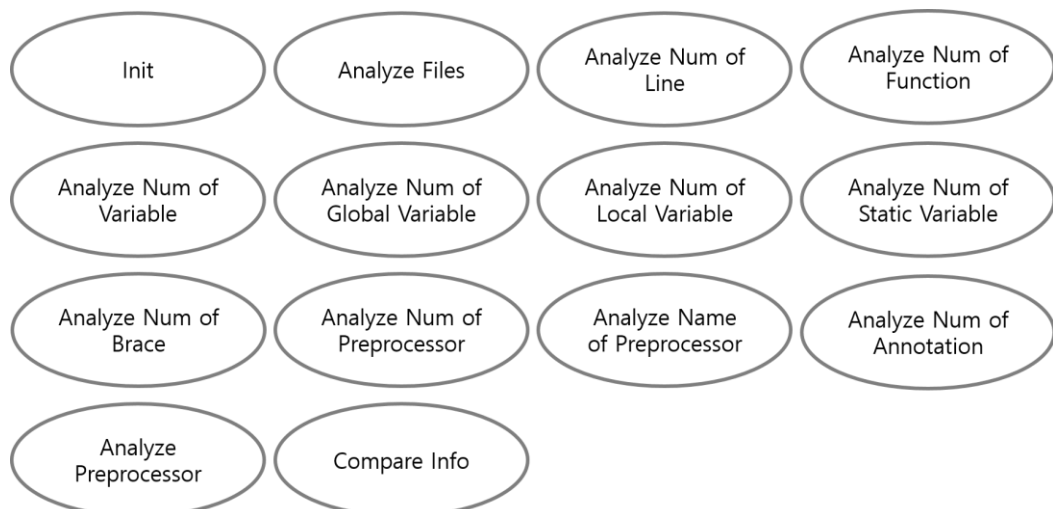
1. 소스 코드 파일 이름에 마우스를 올려 일치율과 유사 항목을 확인한다.
2. 태그 클라우드 또는 오른쪽 리스트에서 소스 코드 파일 이름을 클릭하여 기준을 변경한다.

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 | Init | 1. Init |
| R.2 | Display Main | 2. Display Main |
| R.2.1 | Select Folder | 3. Select Folder |
| R.3 | Analyze Files | 4. Analyze Files |
| R.3.1 | Analyze Num Of Line | 5. Analyze Num Of Line |
| R.3.2 | Analyze Num Of Function | 6. Analyze Num Of Function |
| R.3.3 | Analyze Num Of Variable | 7. Analyze Num Of Variable |
| R.3.4 | Analyze Num Of Global Variable | 8. Analyze Num Of Global Variable |
| R.3.5 | Analyze Num Of Local Variable | 9. Analyze Num Of Local Variable |
| R.3.6 | Analyze Num Of Static Variable | 10. Analyze Num Of Static Variable |
| R.3.7 | Analyze Num Of Brace | 11. Analyze Num Of Brace |
| R.3.8 | Analyze Preprocessor | 12. Analyze Preprocessor |
| R.3.8.1 | Analyze Num Of Preprocessor | 13. Analyze Num Of Preprocessor |
| R.3.8.2 | Analyze Name Of Preprocessor | 14. Analyze Name Of Preprocessor |
| R.3.9 | Analyze Num Of Annotation | 15. Analyze Num Of Annotation |
| R.4 | Display Result | 16. Display Result |
| R.4.1 | Compare Info | 17. Compare Info |
| R.4.2 | Display Sync | 18. Display Sync |
| R.4.3 | Change Center | 19. Change Center |

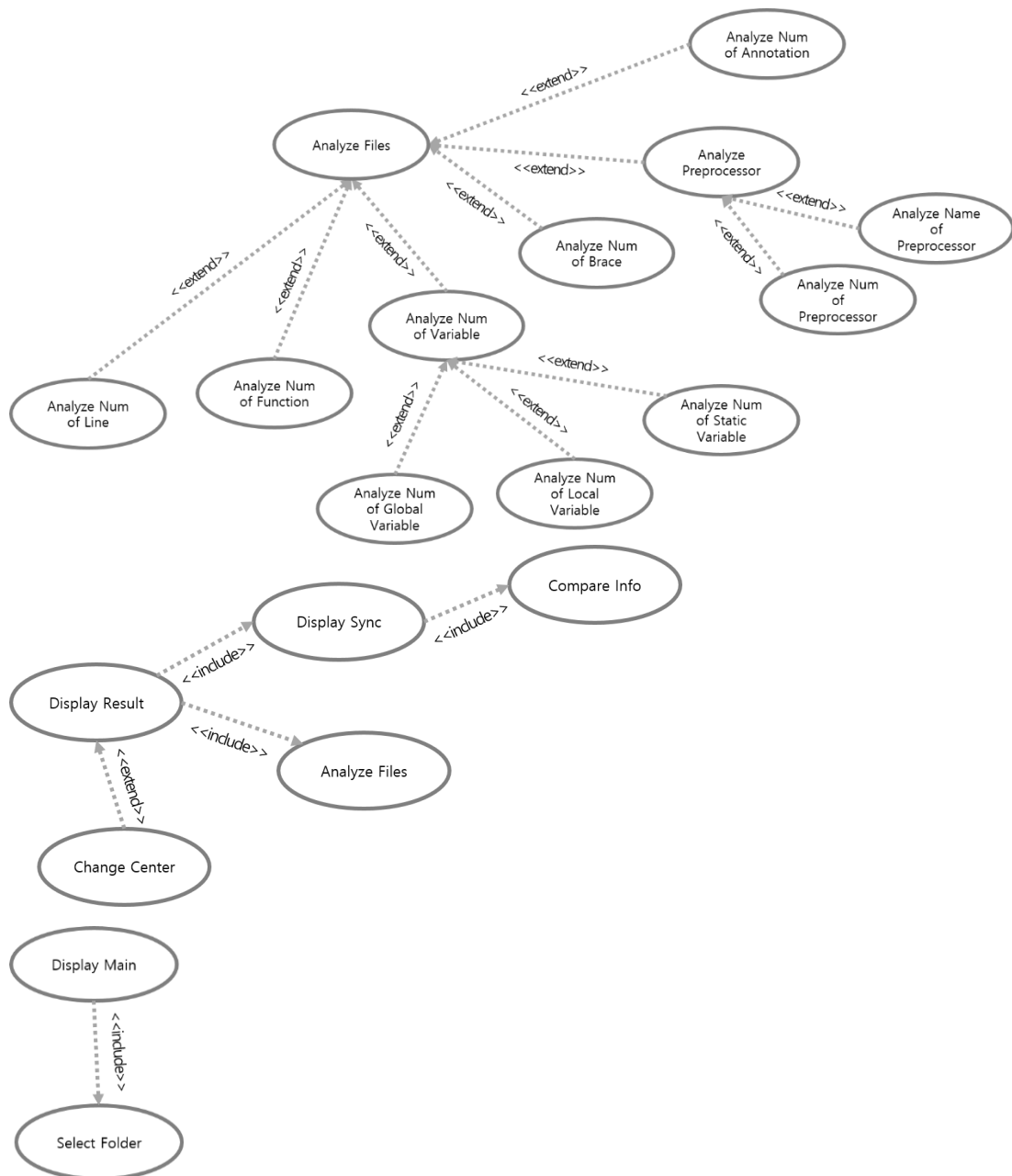
5. Categorize Use-Cases

| Use-Case Number & Name | Category |
|------------------------------------|-----------------|
| 1. Init | Primary |
| 2. Display Main | Primary |
| 3. Select Folder | Primary |
| 4. Analyze Files | Primary |
| 5. Analyze Num Of Line | Primary |
| 6. Analyze Num Of Function | Primary |
| 7. Analyze Num Of Variable | Primary |
| 8. Analyze Num Of Global Variable | Primary |
| 9. Analyze Num Of Local Variable | Primary |
| 10. Analyze Num Of Static Variable | Primary |
| 11. Analyze Num Of Brace | Primary |
| 12. Analyze Preprocessor | Primary |

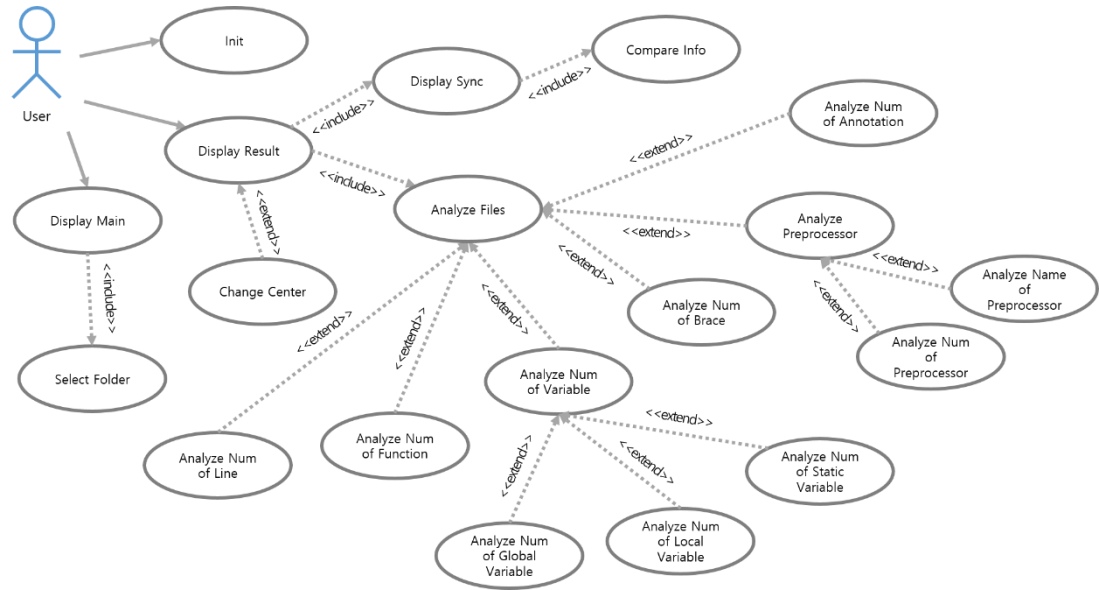
OOPT Stage 1000 <Plan and Elaboration>

| | |
|----------------------------------|---------|
| 13. Analyze Num Of Preprocessor | Primary |
| 14. Analyze Name Of Preprocessor | Primary |
| 15. Analyze Num Of Annotation | Primary |
| 16. Display Result | Primary |
| 17. Compare Info | Primary |
| 18. Display Sync | Primary |
| 19. Change Center | Primary |

6. Identify Relationships between Use-Cases



7. Draw a Use-Case Diagram



8. Describe Use-Cases

| Use-Case Name | Description |
|---------------|--|
| 1. Init | 프로그램 실행 시 또는 재실행 시 프로그램의 모든 내용을 초기화한다. |
| Actor | |
| None | |

| Use-Case Name | Description |
|-----------------|-----------------------------------|
| 2. Display Main | 폴더를 지정할 수 있는 버튼과 '시작하기' 버튼이 제공된다. |
| Actor | |
| User | |

| Use-Case Name | Description |
|------------------|-------------------------------|
| 3. Select Folder | 비교할 소스 코드 파일들이 들어있는 폴더를 지정한다. |
| Actor | |
| None | |

| Use-Case Name | Description |
|------------------|---|
| 4. Analyze Files | 비교할 소스 코드 파일들을 분석하여 일치율 계산에 필요한 정보로 가공한다. |
| Actor | |
| None | |

OOPT Stage 1000 <Plan and Elaboration>

| Use-Case Name | Description |
|------------------------|-----------------------------------|
| 5. Analyze Num Of Line | 폴더 안에 들어있는 소스 코드 파일의 라인 개수를 분석한다. |
| Actor | |
| None | |

| Use-Case Name | Description |
|----------------------------|-----------------------------------|
| 6. Analyze Num Of Function | 폴더 안에 들어있는 소스 코드 파일의 함수 개수를 분석한다. |
| Actor | |
| None | |

| Use-Case Name | Description |
|----------------------------|-----------------------------------|
| 7. Analyze Num Of Variable | 폴더 안에 들어있는 소스 코드 파일의 변수 개수를 분석한다. |
| Actor | |
| None | |

| Use-Case Name | Description |
|-----------------------------------|--------------------------------------|
| 8. Analyze Num Of Global Variable | 폴더 안에 들어있는 소스 코드 파일의 전역 변수 개수를 분석한다. |
| Actor | |
| None | |

| Use-Case Name | Description |
|----------------------------------|--------------------------------------|
| 9. Analyze Num Of Local Variable | 폴더 안에 들어있는 소스 코드 파일의 지역 변수 개수를 분석한다. |
| Actor | |
| None | |

| Use-Case Name | Description |
|------------------------------------|--|
| 10. Analyze Num Of Static Variable | 폴더 안에 들어있는 소스 코드 파일의 Static 변수 개수를 분석한다. |
| Actor | |
| None | |

| Use-Case Name | Description |
|--------------------------|-------------------------------------|
| 11. Analyze Num Of Brace | 폴더 안에 들어있는 소스 코드 파일의 중괄 호 개수를 분석한다. |
| Actor | |
| None | |

OOPT Stage 1000 <Plan and Elaboration>

| Use-Case Name | Description |
|--------------------------|----------------------------------|
| 12. Analyze Preprocessor | 폴더 안에 들어있는 소스 코드 파일의 전처리기를 분석한다. |
| Actor | |
| None | |

| Use-Case Name | Description |
|---------------------------------|-------------------------------------|
| 13. Analyze Num Of Preprocessor | 폴더 안에 들어있는 소스 코드 파일의 전처리의 개수를 분석한다. |
| Actor | |
| None | |

| Use-Case Name | Description |
|----------------------------------|-------------------------------------|
| 14. Analyze Name Of Preprocessor | 폴더 안에 들어있는 소스 코드 파일의 전처리의 이름을 분석한다. |
| Actor | |
| None | |

| Use-Case Name | Description |
|-------------------------------|-----------------------------------|
| 15. Analyze Num Of Annotation | 폴더 안에 들어있는 소스 코드 파일의 주석 개수를 분석한다. |
| Actor | |
| None | |

| Use-Case Name | Description |
|--------------------|---|
| 16. Display Result | 첫 번째 소스 코드 파일을 기준으로 태그 클라우드형 비교 결과 화면을 출력한다 출력한다. |
| Actor | |
| None | |

| Use-Case Name | Description |
|------------------|----------------------------------|
| 17. Compare Info | 분석하여 가공한 정보를 활용(비교)하여 일치율을 계산한다. |
| Actor | |
| None | |

| Use-Case Name | Description |
|------------------|-------------------|
| 18. Display Sync | 일치율과 유사 항목을 제공한다. |
| Actor | |
| None | |

| Use-Case Name | Description |
|-------------------|-----------------------------------|
| 19. Change Center | 비교 기준이 되는 소스 코드 파일을 변경하여 다시 비교한다. |
| Actor | |
| None | |

9. Rank Use-Cases

| Use-Case Number & Name | Rank |
|------------------------------------|------|
| 1. Init | High |
| 2. Display Main | High |
| 3. Select Folder | High |
| 4. Analyze Files | High |
| 5. Analyze Num Of Line | High |
| 6. Analyze Num Of Function | High |
| 7. Analyze Num Of Variable | High |
| 8. Analyze Num Of Global Variable | High |
| 9. Analyze Num Of Local Variable | High |
| 10. Analyze Num Of Static Variable | High |
| 11. Analyze Num Of Brace | High |
| 12. Analyze Preprocessor | High |
| 13. Analyze Num Of Preprocessor | High |
| 14. Analyze Name Of Preprocessor | High |
| 15. Analyze Num Of Annotation | High |
| 16. Display Result | High |
| 17. Compare Info | High |
| 18. Display Sync | High |
| 19. Change Center | High |

Activity 1008. Define Business Concept Model

| Upload | Analyze | Compare | Display |
|--------|---------|---------|---------|
| | | | |
| | | | |

Activity 1009. Define System Test Case

| Ref | Function | Test Case |
|---------|--------------------------------|--|
| R.1 | Init | 메인 화면이 출력되는지 확인한다. |
| R.2 | Display Main | 폴더 지정 버튼과 '시작하기' 버튼이 출력되는지 확인한다. |
| R.2.1 | Select Folder | 폴더 지정 화면이 나타나고, 정상적으로 지정되었는지 확인한다. |
| R.3 | Analyze Files | 비교 결과 출력 화면이 나오는지 확인한다. |
| R.3.1 | Analyze Num Of Line | 소스 코드 파일의 라인 개수만 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다. |
| R.3.2 | Analyze Num Of Function | 소스 코드 파일의 함수 개수만 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다. |
| R.3.3 | Analyze Num Of Variable | 소스 코드 파일의 변수 개수만 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다. |
| R.3.4 | Analyze Num Of Global Variable | 소스 코드 파일의 전역 변수 개수만 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다. |
| R.3.5 | Analyze Num Of Local Variable | 소스 코드 파일의 지역 변수 개수만 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다. |
| R.3.6 | Analyze Num Of Static Variable | 소스 코드 파일의 Static 변수 개수만 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다. |
| R.3.7 | Analyze Num Of Brace | 소스 코드 파일의 중괄호 개수만 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다. |
| R.3.8 | Analyze Preprocessor | 소스 코드 파일의 전처리기에서 행해지는 부분들의 코드를 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다. |
| R.3.8.1 | Analyze Num Of Preprocessor | 소스 코드 파일의 전처리기에서 행해지는 부분 중 상수 개수만 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다. |
| R.3.8.2 | Analyze Name Of Preprocessor | 소스 코드 파일의 전처리기에서 행해지는 부분 중 상수 이름만 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다. |
| R.3.9 | Analyze Num Of Annotation | 소스 코드 파일의 주석 개수만 변경하면서 예상한 결과값과 프로그램 실행 후 얻어지는 결과값을 비교해 본다. |
| R.4 | Display Result | 태그 클라우드 형으로 비교 결과 출력 화면이 나오는지 확인한다. |
| R.4.1 | Compare Info | 비교 소스 코드 파일 이름에 마우스 커서를 올리면 일치율과 유사 항목이 출력되는지 확인한다. |
| R.4.2 | Display Sync | 비교 소스 코드 파일 이름에 마우스 커서를 올리면 일치율 |

| | | |
|-------|---------------|-----------------------------|
| | | 과 유사 항목이 출력되는지 확인한다. |
| R.4.3 | Change Center | 기준 소스 코드 파일 이름이 변경되는지 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 비교율을 알기 쉽게 보여주는 Clone Checker 프로그램을 개발하여 비교 정확도를 높이는 것을 목표로 한다.

3. Functional Requirements

- A. 초기화
- B. 폴더 지정
- C. 분석 시작
- D. 결과 출력
- E. 비교 기준 소스 코드 파일 변경

4. 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.

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.

OOPT Stage 1000 <Plan and Elaboration>

7. Other Requirements

다른 프로그래밍 언어의 소스 파일 또는 다른 분야에서도 사용이 가능할 수 있게끔 확장성을 지닐 수 있도록 한다.

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 × 5 일/주 × 12 주)

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 | | | | █ | █ | | | | | | | | | | | | | | |