

디자인 씽킹 기반 스켈레톤 코드 생성

서채연^{1*} 김장환² 장우성³ 박보경⁴ 김영철⁵
 홍익대학교 소프트웨어 공학연구소, 유원대학교 스마트 IT 학과
 e-mail :{chaeyun^{1*}, bob⁵}@hongik.ac.kr
 {janghwan², jang³}@selab.hongik.ac.kr, ⁴parkse@u1.ac.kr

Skeleton Code Generation based on Design Thinking

ChaeYun Seo^{1*}, J. Kim², W. Jang³, B. Park⁴, R. Young Chul Kim⁵
 SE Lab, Dept. of Software and Comm. Engineering Hongik University
 Dept. of Smart IT, U1 University

요약

현재 대학에서 비전공자들에게 창의적 사고 기반의 코딩 능력이 부재하다. 비전공자들에게 창의적 문제해결을 위한 디자인 씽킹을 활용하여 소프트웨어 교육을 하지만, 디자인 씽킹에서 직접적 코딩을 이 불가능하다. 이를 위해, 소프트웨어 공학 개발 공정 단계를 디자인 씽킹 프로세스에 적용하고자 한다. 이를 통해, 소프트웨어 개발주기인 요구사항부터 설계(클래스, 순차와 상태 다이어그램)를 통한 코딩을 디자인 씽킹의 각 단계의 요소에 적용하고자 한다. 결국, 디자인 씽킹 프로세스 공정에 따르면 스켈레톤 코드 생성이 가능하도록 한다.

1. 서론

최근, 기술의 발전이 가속화되면서 컴퓨터 프로그래밍에 관한 관심이 높아지고 있다. 이는 사회의 모든 분야에서 창의적인 사고와 코딩을 통해 컴퓨팅을 이용한 서비스들이 지속해서 등장하고 이에 따라 세계가 기술 중심으로 성장하고 있기 때문이다[1]. 하지만 일반인들은 전공자나 현업에서 일하는 개발자처럼 기술의 발전을 따라가는 것은 거의 불가능하고 이와 함께 기업도 지능정보기술을 기반으로 한 응용 분야에서 기업 간의 기술격차가 확대되고 있다[2]. 따라서 기업 간의 기술격차 및 사람들의 기술격차를 줄이고 비전공자들에게도 기회를 줄 수 있는 연구들이 지속하고 있다[3]. 이러한 문제들을 해결하기 위해 본 논문에서는 디자인 씽킹 기반 스켈레톤 코드 생성을 제안한다. 이 방법은 디자인씽킹을 통해 소프트웨어 개발 프로세스, 각 공정에 따른 문서기법, 설계기술 등을 완벽히 알지 못해도 스켈레톤 코드를 생성해 코드를 생성할 수 있다. 창의적 사고 기반의 디자인씽킹 방법은 문제해결을 위한 솔루션 기반 접근방식을 제공함으로써 제품이나 서비스를 디자인하는데 사람들의 이해를 발전시키는데 크게 이바지하고 있다[4].

본 논문은 2장 관련 연구인 디자인씽킹을 언급하고, 3장 디자인 씽킹 기반의 스켈레톤 코드 생성 방법을 설명한다. 4장은 결론 및 향후 연구에 관해 서술한다.

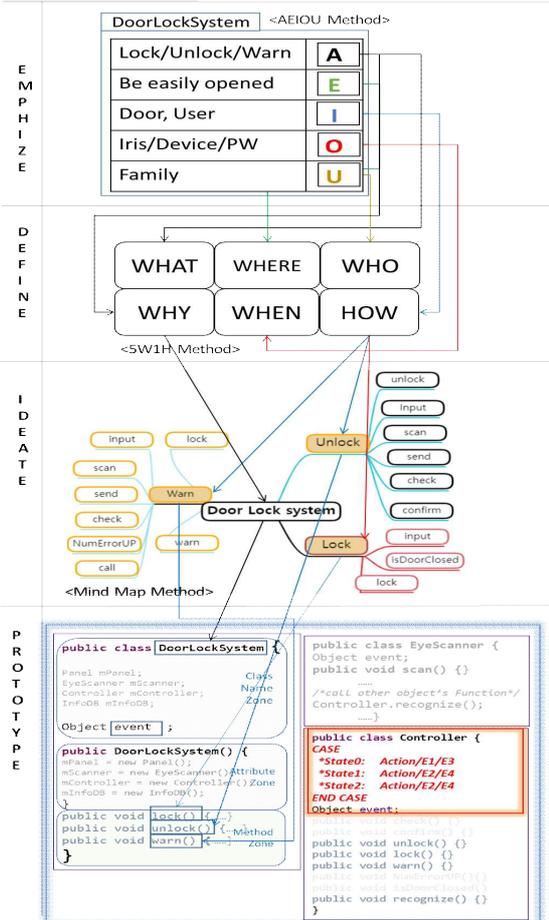
2. 관련 연구

2.1 Design Thinking(DT)

디자인 씽킹은 개발자가 사용자를 이해하고, 문제를 재정의하고, 프로토타입 및 테스트를 위한 혁신적인 솔루션을

생성하는 데 사용하는 비선형 반복 프로세스이다.

Design Thinking



(그림 1) 스켈레톤 코드 생성을 위한 디자인 씽킹 프로세스

디자인 씽킹의 프로세스는 잘못 정의되거나 알려지지 않은 문제를 해결하는 데 가장 유용하다[4].

그림 1은 스키텔론 코드 생성을 위한 디자인 씽킹 프로세스이다. 스키텔론 코드 생성을 위한 단계별 요소는 다음과 같다[5].

- 공감-AEIOU방법론으로 각 요소 식별
- 문제 정의-공감에서 식별한 요소를 기반으로 5W1H 방법론의 요소 식별
- 아이디어-마인드맵 방법론 기반 시스템 기능 추출
- 프로토타입-마인드맵으로 추출한 결과를 기반으로 스키텔론 코드 매핑

그림 1은 스키텔론 코드 생성을 위한 디자인 씽킹 프로세스의 수직적 매핑관계를 보여준다.

3. 디자인 씽킹 기반 스키텔론 코드 생성

본 장에서는 소프트웨어공학 개발 프로세스를 모르더라도 코딩할 수 있는 디자인 씽킹 기반 스키텔론 코드 생성에 대해 설명한다. 디자인씽킹의 단계별 요소와 소프트웨어공학 개발 프로세스의 단계별 요소를 수평적으로 매핑하여 스키텔론 코드 생성이 가능하다. 그림 2는 유스케이스 다이어그램이다. 그림 3은 디자인 씽킹 요소와 소프트웨어개발 프로세스 단계 매핑을 위한 절차이다.

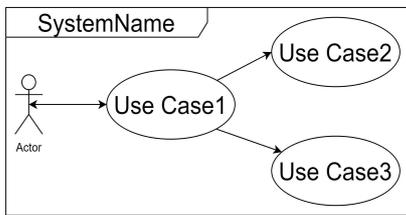
- Use-case Diagram
- Class Diagram
- Sequence Diagram
- State Diagram
- State Table
- Skeleton Code generation

3.1 소프트웨어공학 “요구사항”

소프트웨어공학 개발 프로세스의 요구사항에서 시스템의 기능과 사용자를 찾는다.

3.2 소프트웨어공학 “분석”

a. Use-case Diagram: 요구사항에서 추출한 시스템 기능을 이용해서 유스케이스 다이어그램을 그린다.



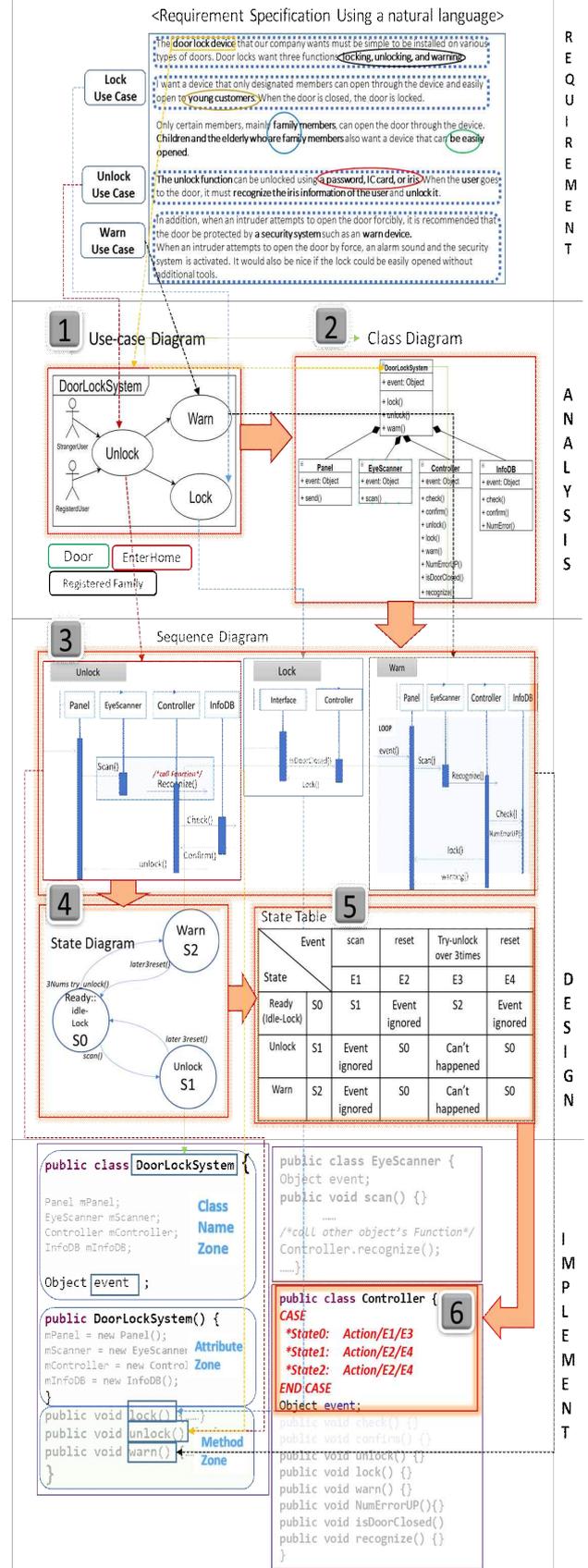
(그림 2) 유스케이스 다이어그램

요구사항에서 추출한 시스템 기능의 수 만큼 유스케이스를 n개까지 만들 수 있다. 유스케이스를 만든 후, 프레임워크를 띄워 시스템 이름을 정의한다.

b. Class Diagram:

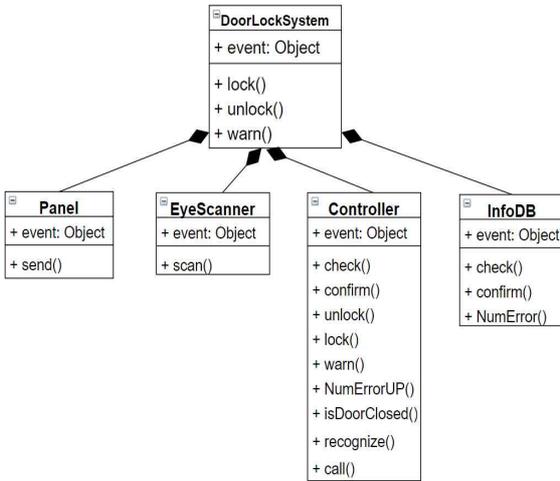
유스케이스 다이어그램을 기반으로 클래스 다이어그램을 만든다. 시스템 이름을 기준으로 유스케이스에서 추출한 기능과 그 기능 안에 들어가는 메소드를 정의해서 클래스 다이어그램을 그린다.

Software Engineering



(그림 3) 스키텔론 코드 생성을 위한 소프트웨어개발 프로세스
그림 4는 클래스다이어그램으로 시스템 전체 구조를 표현한다[6]. 클래스 다이어그램의 컴포지션 관계는 생명주

기가 같아지는 클래스 사이의 “전체-부분” 관계를 표현한다[7].



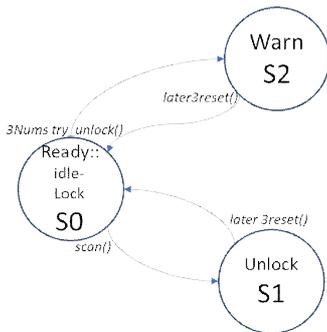
(그림 4) 클래스 다이어그램

클래스 다이어그램에서 기본 코드 템플화가 가능하다.

3.3 소프트웨어공학 “디자인”

c. **Sequence Diagram:** 클래스 다이어그램에서 표현된 기능들을 순차 다이어그램으로 나타낸다. 클래스 다이어그램으로 표현된 기능들과 그 안에 사용되는 메소드들로 순차 다이어그램을 그린다.

d. **State Diagram:** 순차 다이어그램에서 컨트롤 오브젝트를 찾아 컨트롤 오브젝트의 상태 다이어그램이다[8]. 그림 5는 상태 다이어그램을 보여준다.



(그림 5) 상태 다이어그램

e. **State Table:** 상태 다이어그램을 이용해서 상태 테이블을 만든다. 상태 테이블은 CASE 구문으로 바꾸어 if then else로 스킴레톤 코드를 생성할 수 있다[8]. 그림 6은 상태 테이블이다.

Event \ State		scan	reset	Try-unlock over 3times	reset
		E1	E2	E3	E4
Ready (Idle-Lock)	S0	S1	Event ignored	S2	Event ignored
Unlock	S1	Event ignored	S0	Can't happened	S0
Warn	S2	Event ignored	S0	Can't happened	S0

(그림 6) 상태 테이블

S0 상태에서 E가 발생하면 S1 상태가 된다. S0 상태에서 E3가 발생하면 S2 상태가 된다. S1 상태에서 E2가 발생하면, S0 상태가 된다. S1 상태에서 E4가 발생하면, S0 상태가 된다. S2 상태에서 E2가 발생하면, S0 상태가 된다. S2 상태에서 E4가 발생하면, S0 상태가 된다. 상태 테이블에서 “Can't be happened”는 Error code를 뜻하고, “Event Ignored”는 발생하지 않는 동작을 의미한다[8].

3.4 소프트웨어공학 “구현”

f. **Skeleton Code Generation:**

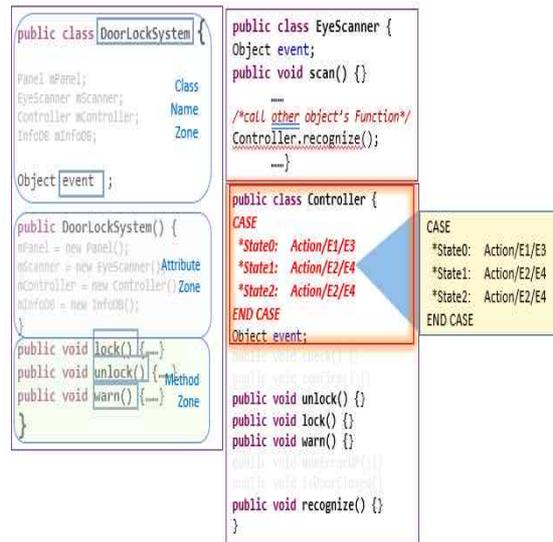
상태 테이블의 상태도를 기반으로 상태 테이블의 CASE 스킴레톤 코드를 생성할 수 있다[8]. 표 1은 상태 테이블의 스킴레톤 코드이다.

(표 1) 상태 테이블의 Skeleton Code

```

CASE
    *S0: Action/E2/E4
    *S1: Action/E1
    *S2: Action/E2/E4
END CASE
    
```

E2 또는 E4가 발생하면 S0 상태로 간다. 이를 조건문으로 변경하면, “E2 또는 E4 조건 만족하면, S0를 실행“이 된다. 이를 통해, CASE 구문의 스킴레톤 코드가 생성할 수 있다.



(그림 7) 스킴레톤 코드 생성

즉, 코드 생성은 클래스 다이어그램으로부터는 객체의 구조, 순차 다이어그램으로부터 객체 간의 호출 관계, 상태 다이어그램으로부터 컨트롤 객체의 상태 변화도를 추출된 스킴레톤 코드이다. 그림 7은 전체 코드를 나타낸다.

4. 결 론

기술의 발전이 가속화되면서 컴퓨터 프로그래밍에 관한 관심이 높아지고 있다. 현재 대학에서는 비전공자들을 대상으로 디자인 씽킹을 활용한 소프트웨어 교육하지만, 디자인 씽킹 프로세스를 활용해 직접 코딩을 하지 못한다는 문제점이 있다. 이러한 문제들을 해결하기 위해 본 논문에서는 디자인 씽킹 기반 스킴레톤 코드 생성을 제안한다.

이 방법은 디자인씽킹을 통해 소프트웨어 개발 프로세스, 각 공정에 따른 문서기법, 설계기술 등을 완벽히 알지 못해도 스킴레톤 코드를 생성해 코드를 작성할 수 있다. 이를 통해 비전문가들은 소프트웨어공학 개념 없이도, 디자인씽킹 메커니즘의 코딩화를 제시한다. 현재 디자인 씽킹의 공감 요소를 입력하면 코드 템플릿을 추출할 수 있는 연구가 진행중이다.

Acknowledgement

이 논문은 2020년도 정부(교육부)의 재원으로 한국 연구재단의 지원을 받아 수행된 기초 연구 사업(NRF - 2020R111A1A01072928)임.

참고문헌

[1]"Forbes", "Should Everyone Learn To Code? 15 Tech Pros Weigh In On Why Or Why Not", Mar 20, 2020, <https://www.forbes.com/sites/forbestechcouncil/2020/03/20/should-everyone-learn-to-code-15-tech-pros-weigh-in-on-why-or-why-not/?sh=409afd7a693e>

[2] 최창열, "제4차 산업혁명과 e-비즈니스 기업의 발전 전략과 시사점", e-비즈니스연구, 18(3), 39-54, 2017

[3] 서채연, 김영철, "디자인 씽킹 메커니즘과 소프트웨어공학 접목에 관한 연구", 2020온라인춘계학술발표대회, 한국정보처리학회, 2020

[4]<https://www.interaction-design.org/literature/article/what-is-design-thinking-and-why-is-it-so-popular>

[5] 서채연, 김영철, "Adapting Design Thinking to Software Design for solving ill-defined requirement problem on Creative Thinking, 2021, IJCC2021 국제학술대회"

[6] Y. J. Kwak and Jongbeom Oh, "UML object-oriented design for beginners", Seoul: Information Culture History, 2006

[7] Hye-young Yoo, "SoftwareEngineering", 4rd ed., Seoul: SciTech, 1998

[8] Choi Eun Man, "Object Oriented Software Engineering", Hanbit Academy, 2017