디자인씽킹 메커니즘을 사용한 불분명 요구사항명세 개선

김장환[1*], 장우성[2], 박보경[3], 손현승[4], 서채연[5], 김영철[6]
홍익대학교 소프트웨어공학 연구실[1,2,5,6]
U1대학교[3], 목포대학교[4]
{janghwan[1], jang[2]}@selab.hongik.ac.kr, parkse@u1.ac.kr, hson@mokpo.ac.kr, {chaeyun[5], bob[6]}@hongik.ac.kr

Enhancing ill-defined/unknown Requirement Specification with Design Thinking Mechanism

Janghwan Kim[1*], Woo Sung Jang[2], Bokyung Park[3], Hyun Seung Son[4], ChaeYun Seo[5], R. Young Chul Kim[6]
SE Lab, Dept. of Software and Comm. Engineering Hongik Universit[1,2,5,6]
U1 University[3], Mokpo National University[4]

요 약

현재 소프트웨어 프로젝트에서 명료한 요구사항 추출 및 정의는 매우 힘든 영역이다. 특히, 자연어 기반의 요구사항문서는 요구사항 분석가의 이해도, 성숙도에 따라 다른 의미로 해석할 수 있다. 또한 빈번한 요구사항 변경 및 비 정의된, 비 명료한 요구사항들에 대해서, 명료한 요구사항을 만드는 것이 현재의 이슈이다. 이를 해결위해, Design Thinking 메커니즘을 요구공학에 접목하여 명료한 요구사항을 추출 및 재정의를 제안한다. 이는 창의적 사고 기반으로 ill-defined/unknown 문제에 대해서 요구사항에 대한 재정의 및 개선이 가능하다.

1. 서 론

4차 산업 혁명 시대가 오면서 소프트웨어 개발 프로세스에서 요구사항 분석과 설계는 매우 소프트웨어 개발에 매우 중요한 이슈이다. 소프트웨어 개발 프로세스는 요구사항을 분석하고 분석된 정보를 바탕으로 소프트웨어를 설계하며 설계를 바탕으로 구현되기 때문에 프로세스의 각 단계는 서로 유기적인 연관관계를 갖는다. 하지만 발주자는 개발자가 아니기 때문에 많은 경우에 개발자의 언어로 말하기보다는 일반인 관점에서 소프트웨어를 설명(자연어 요구사항)한다. 따라서 발주자의 요구사항은 많은 경우에 그 의미가 명료하지 못하고 다른 의미에 대한 해석을 가능하게 한다[1]. 이러한 문제는 요구사항 분석이 대부분 개발자의 성숙도레벨에 의해 분석되어 왔기 때문에 발생한다. 이 문제를 해결하기 위해서, 본 논문에서는 소프트웨어공학에 기반하고 디자인씽킹의 창의적사고 기법을 이용해 명료하지 않은 요구사항들을 보다 명료하게 분석하는 방법을 제안한다. 또한 Gary E. Mogyorodi는 요구사항에서 모호성을 제거하는 것, 즉 요구사항을 명료하게 정의하면 요구사항의 품질이 향상된다고 언급한다[2].

본 논문에서 제안하는 방법은 창의적 사고기반의 디자인씽킹 메커니즘을 이용해 요구사항 모호성을 간결하게 하여 요구사항의 명료성을 높이고 이에 따라 요구사항의 품질향상을 위한 시도 연구이다. 이 방법을 통해 개발자에게 전달되는 요구사항들이 보다 명료해지고 개발자로 하여금 명료해진 방법을 이용해 설계의 정확도 향상을 기대하여 소프트웨어 품질의 향상을 기대한다. 본 논문은 다음과 같은 순서로, 2장에서는 관련 연구로서 소프트웨어 개발 생명주기와 디자인씽킹의 창의적 사고기법을 소개한다. 3장에서는 제안하는 요구사항명세 향상 방법에 대해 제안한다. 4장에서는 사례연구를 통해 적용하고, 5장에서는 결론과 함께 향후 연구 방향에 관해 서술한다.

2. 관련연구

디자인씽킹이란 '디자이너처럼 생각하는 방법'으로써, 디자이너들이 최종 결과물을 만들어 내기 위한 '사고 과정'(Thinking Process)이다[3]. 디자인씽킹은 다음과 같은 프로세스를 통해 창의적인 사고를 돕는다. 공감 (Empathize) - 정의(Define) - 아이디어 만들기(Ideate) - 모형제작(Prototype) - 검증(Test)등의 단계들을 포함한다. 또한, 디자인씽킹은 불분명하고 알려지지 않은 문제들을 재정의함으로써 문제를 해결하는 데 매우 유용한 기법이다[4]. 각 단계는 항상 순차적일 필요가 없으며, 꼭 병렬로 발생하고 각 단계를 반복적으로 수행할 수 있다[3]. 따라서 각 단계를 계층적 또는 단계별 프로세스로 이해해서는 안 된다. 소프트웨어 개발 생명주기는 소프트웨어공학의 소프트웨어 개발과정을 뜻하는 용어로써 소프트웨어 개발 전반에 걸친 공정을 나타낸다. 소프트웨어 개발 생명주기는 요구사항 분석-설계-구현-테스팅-운영 단계로 구성되어 있다[5].

3. 불분명 요구사항명세 향상 방법 with example

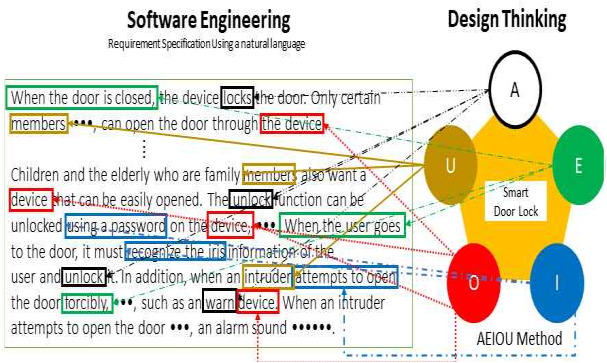

그림 1. Mapping NLR with AEIOU method

소프트웨어 개발은 발주자의 요구사항으로부터 시작된다. 요구사항명세서가 아닌 계획단계의 요구사항은 개발자 혹은 기

획자의 소프트웨어에 대한 성숙도레벨에 따라 요구사항이 분석된다. 이렇게 분석된 정보는 개발자의 소프트웨어 개발 성숙도에 따라서, 요구사항들은 개발자 개인의 업무경험과 능력에 매우 의존적이며 명료하지 않은 단점을 가진다[6].

Step 1. Requirement Analysis with Empathize

먼저, 자연어로 이루어진 요구사항(Natural Language Requirement: NLR)들을 문장단위로 나눈다. 문장 단위로 구분된 요구사항들을 디자인씽킹의 공감하기 기법을 통해 요구사항 명세를 한다. 디자인씽킹의 공감단계 방법을 'AEIOU 기법' 을 사용한다. **'A'** 는 Activity의 요구사항의 주제로 맵핑(Mapping)된다. **'E'** 는 Environment의 약자로 요구사항이 수행되는 '환경' 을 설정한다. **'I'** 는 Interactions의 약자로 User와 상호작용하는 부분과 맵핑된다. **'O'** 는 Objects의 약자로 요구사항에서 User와 Interaction하는 사물들과 맵핑된다. **'U'** 는 Users의 약자로 요구사항에서 소프트웨어의 사용자 또는 Object와 상호작용의 주체와 맵핑된다.

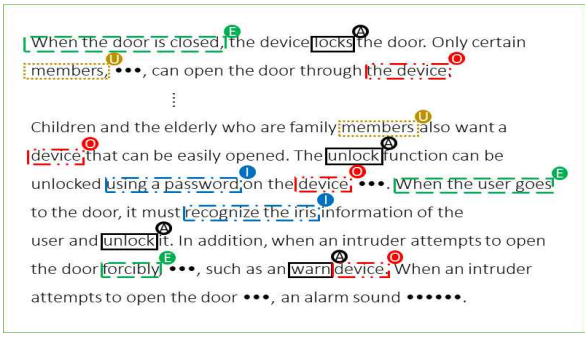

그림2. Mapping Result of NLR

위의 그림2와 같이 'AEIOU기법' 을 사용해 자연어로 이루어진 요구사항에서 각 범주에 따라 나눈 문장들을 분류하고 맵핑한다.

Step 2. Redefine Requirements with 5W1H

자연어로 된 요구사항들을 Step 1단계에서 AEIOU기법을 ㅌ 통해 추출하면, 추출된 요구사항 정보들을 그림 4처럼 다시 하나로 묶어 문장을 재 정의한다. 이때, 디자인씽킹 기법 중 하나인 5W1H기법을 통해 문장을 재 정의한다. 5W1H 방법은 흔히 '육하원칙' 이라고 불리는 방법으로서 이 방법을 사용하면 문장이 명료해진다[7]. 문장이 '명료하다' 라는 말은 "*한 문장이 여러 가지로 해석될 수 있는 표현은 삼가고 한 가지 의미만을 전달해야 함*" 을 의미한다[8]. 5W1H기법을 통해 재 정의된 요구사항들은 자연어로 나열되어 있을 때 보다 의미적으로 명료하게 된다.

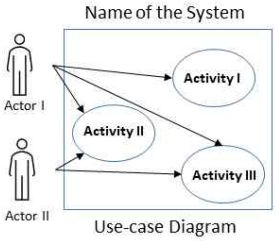

그림 3. Use-case Diagram

Step 3. Use-case Diagram with Redefined Requirement

새롭게 다시 정의된(Redefined) 요구사항들은 Step 3에서 Use-case Diagram으로 나타낸다. 그림 3과 같이 Activity + Object의 결합으로 Use-case scenario를 만들고 User를 통해 Actor를 나타낸다. 이렇게 만들어진 Actor와 Scenario들의 관계를 맵핑해 Use-case Diagram을 나타낸다. Use-case Diagram은 소프트웨어의 기능들을 직관적으로 보여주고 사용자와 소프트웨어의 상호작용을 나타낸다[9]. 이 단계에서는 Step 1의 공감단계에서 추출한 Actor와 Object, Interaction 등의 관계를 Activity와 Object를 묶어 Use-case scenario를 만들고 Scenario와 상호작용하는 User는 Actor로, Interaction은 화살표로 맵핑하여 나타낸다.

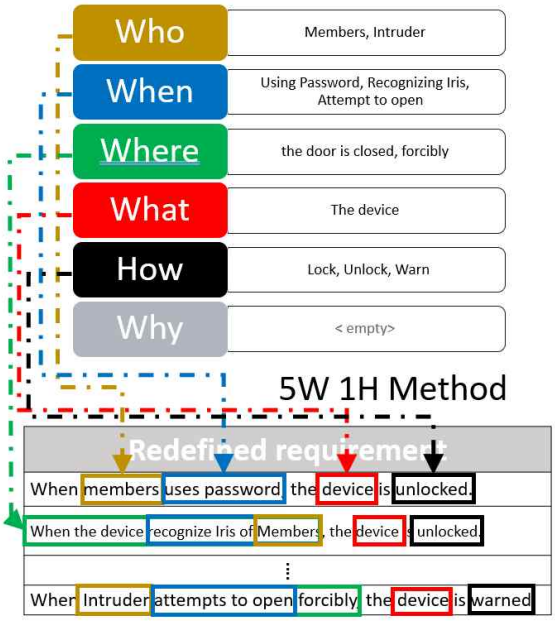

그림 4. Redefine requirements by 5W1H method

이와 같은 방법은 요구사항을 요소별로 나눈 정보를 바탕으로 Use-case Diagram으로 나타내고 5W1H로 재정의된 요구사항으로부터 설계하기 때문에 본 논문에서 제안한 분석방법을 통해 맵핑하지 않고 바로 요구사항으로부터 설계한 것보다 설계의 명료성을 향상시킨다.

Step 4. Abstract Design by Mind Mapping

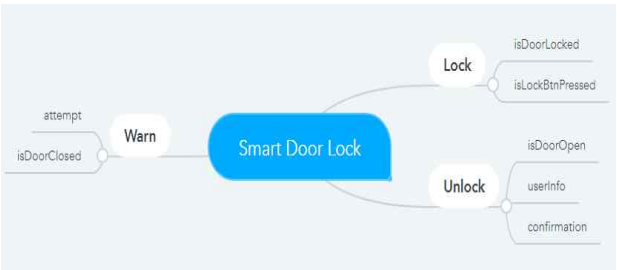

그림 5. Requirement Analysis by Mindmap

Step 4는 SW 설계단계를 추상설계(Use-case diagram)을 바탕으로 마인드맵을 작성한다. 그림 5처럼 마인드맵은 Step 2에서 재 정의한 요구사항과 Step 1에서 추출한 요구사항요소

정보들을 통해 클래스 다이어그램을 생성한다. 먼저 Step 2에서 재 정의된 요구사항의 기능을 단어 단위로 추출한다. 추출된 기능단어를 마인드맵의 중앙에 위치하고 기능수행에 필요한 요소들을 마인드맵으로 작성한다. 이렇게 표현된 기능을 기반으로 도어락 시스템의 전체 구조를 그림 6과 같이 소프트웨어공학의 클래스다이어그램으로 작성한다.

Step 5. Design method flow by Sequence Diagram

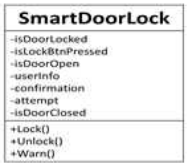

그림 6. Class Diagram for Smart Door Lock

Step 5. 마인드맵을 통해 클래스다이어그램을 그렸으면 재 정의된 요구사항과 클래스다이어그램을 바탕으로 시퀀스 다이어그램을 생성한다. 그림 7처럼 시퀀스다이어그램은 기능이 '액터' 로부터 수행되는 과정을 순차적으로 보여준다. 시퀀스 다이어그램을 통해 해당 기능을 가진 Function이 어떤 순서로 기능을 수행하는지 보여줄 수 있으므로 코드의 순차적 수행순서를 이해도를 향상시킨다.

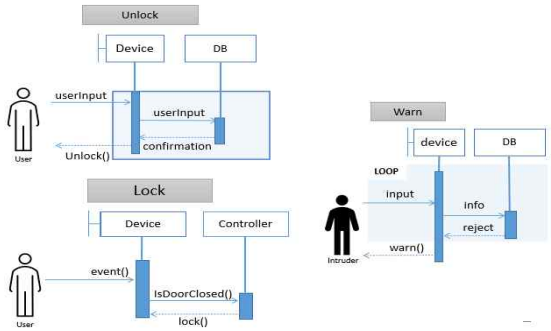

그림 7. Sequence diagram for Smart Door Lock System.

Step 6. 스켈레톤(Skeleton) 코드 추출

Step 6에서는 클래스다이어그램, 시퀀스 다이어그램을 통해 스켈레톤 코드를 추출한다[10]. 클래스 다이어그램에서 정의한 클래스이름, Field Attributes, Methods 등을 활용해 그림 8과같이 스켈레톤 코드(뼈대코드)를 작성한다. 이때 Method 설계에 필요한 설계정보들은 시퀀스다이어그램의 메시지의 흐름과 정보들을 바탕으로 작성한다.

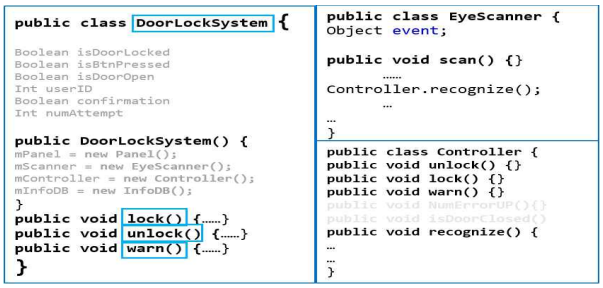

그림 8. Skeleton code extraction result

4. 결론 및 향후 연구

본 논문은 디자인씽킹 메커니즘을 사용한 불분명 요구사항명세 향상 방법에 관한 연구이다. 이 방법은 전통적인 소프트웨어개발 생명주기의 요구사항 분석 단계에서 불분명한 요구사항을 디자인씽킹 메커니즘을 사용해 의미적으로 명료하게 향상 시킨다. 향후, 창의적 사고기법의 디자인씽킹을 이용한 방법들을 구체화하고 좀 더 명료한 분석방법의 자동화에 대한 연구를 할 예정이다.

6. Acknowledgement

본 논문(저서)은 교육부 및 한국연구재단의 4단계 두뇌한국21 사업(4단계 BK21 사업)으로 지원된 연구임.

참고문헌

[1] 김홍진, 조동혁, 안태호, AHP 기법을 활용한 소프트웨어 제안평가요인의 상대적 중요도에 관한 연구 : 발주자와 수주자 비교를 중심으로, Journal of Information Technology Services, 16, 1, 48-49, 2017

[2] Gary E. Mogyorodi, B.Math., "Requirements-Based Testing - Cause-Effect Graphing", Software Testing Services, 2005

[3] INTERACTION DESIGN FOUNDATION,

https://www.interaction-design.org/literature/topics/design-thinking

[4] 창의적사고와코딩편찬위원회, 창의적 사고와 코딩 공학, Nosvos, 2018

[5] 서채연, 디자인 씽킹 메카니즘과 소프트웨어공학 접목에 관한 연구, 2020온라인춘계학술발표대회, 한국정보처리학회, 2020, p349-351

[6] 조병선, 이석원. (2020). 자연어처리와 기계학습을 이용한 요구사항 분석기술 비교 연구. 한국컴퓨터정보학회논문지 , 25(7), 27-37.

[7] TSUJI, Kayo. Implementation of the Writing Activity Focusing on 5W1H Questions: An Approach to Improving Student Writing Performance. LET Journal of Central Japan, 28: 1-12, 2017.

[8] 오병국, 미디어 글쓰기, 아시아 출판사, 2013

[9] Object Managment Group, OMG Unified Modeling Language, https://www.omg.org/spec/UML/2.5/PDF, March 2015.

[10] Abilio G. Parada, Eliane Siegert, Lisane B. de Brisolara, Generating Java code from UML Class and Sequence Diagrams, 2011 Brazilian Symposium on Computing System Engineering, 101, 2011

[11] 서채연, 김영철, Adapting Design Thinking to Software Design for solving ill-defined requirement problem on Creative Thinking, 2021, IJCC2021국제학술대회