

마인드 맵을 확장한 요구사항 명세 및 분석 도구 제안

Jiang Hua, Scott Uk-Jin Lee

한양대학교 컴퓨터공학과
경기도 안산시 상록구 사3동
wjhyyanan@hanyang.ac.kr, scottlee@hanyang.ac.kr

요약: 요구사항 유도, 분석 및 명세는 소프트웨어 개발 과정에서 매우 중요한 단계이며, 요구사항 명세는 현재 기본적으로 자연어를 사용하여 작성되고 있다. 하지만 정형화되지 않은 자연어를 사용하여 작성된 소프트웨어 요구사항은 애매모호하여 요구사항에 대한 개발자와 사용자의 잘못된 의도가 반영되는 경우가 빈번히 발생하며, 이는 소프트웨어 개발이 실패하는 주 원인이 된다. 본 논문에서는 기존에 개발된 마인드 맵을 확장하여 개발자와 사용자 상호간의 정확한 소프트웨어 요구사항 명세 및 분석을 통하여 상호간의 요구사항 이해도와 참여도를 높일 수 있는 요구사항 명세 및 분석도구를 제안한다.

핵심어: 소프트웨어 요구공학, 소프트웨어 요구사항 명세, 마인드 맵.

1. 서론

소프트웨어 공학의 용어 중 요구사항은 사용자의 문제 혹은 목적을 나타내고 문제를 해결하거나 목적을 달성하는 방법을 정형화된 문서로 표현한 것이다 [1]. 대규모 소프트웨어 개발에서 프로젝트가 실패하는 원인 중 하나는 요구사항의 변경으로 인한 재 개발이다. 소프트웨어 개발에서의 재 개발 비용은 전체 개발비용의 30 ~ 50%를 점유하며 [2], 요구사항 오류에 의해 생성된 재 개발은 전체 재 개발 비용의 70 ~ 85%를 차지한다 [3].

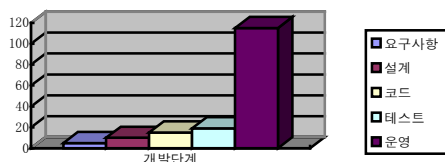


그림 1. 요구사항 오류가 발견된 시기에 따른 수정 비용

또한, 그림 1에서 볼 수 있듯이, 소프트웨어 개발 중 요구사항의 변경으로 발생하는 문제는 사전에 미리 발견될수록 수정 비용이 줄어든다 [4]. 요구사항 변경을 일으키는 위험요소로는 부족한 사용자 참여,

사용자 요구사항의 확장, 불분명한 요구사항, 지나친 외양, 최소한의 명세, 간과된 사용자 계층, 부정확한 계획 등을 들 수 있다.

표 1. 소프트웨어 프로젝트의 성공 및 실패의 이유

Project success factors	Reasons for project failure
User Involvement	Incomplete Requirements
Management Support	Lack of User Involvement
Clear Statement	Moving Targets
Proper Planning	Lack of Resources
Realistic Expectations	Insufficient Communication

표 1은 Standish CHAOS 보고서 [5]에 의한 소프트웨어 프로젝트의 실패 요인들을 정리한 것으로, 사용자 참여 부족 및 불분명한 요구사항과 같이 비효과적인 요구사항의 명세와 분석이 프로젝트가 실패하게 되는 주요 원인이라는 것을 보여준다. 요구사항의 명세 및 분석이 소프트웨어 프로젝트의 실패 원인이 되는 주 이유는 현재 소프트웨어 개발에 필요한 요구사항이 대부분 자연어로 기술되었기 때문이다. 요구사항의 명세에 자연어 사용하는 이유는 개발자 측면에서 사용자의 요구를 알기 쉽게 하고, 사용자 측면에서 시스템의 요구 사항을 이해하기 쉽게 하는 효율적인 공통언어 없기 때문이다. 하지만 자연어의 모호함(ambiguity), 부정확함(impreciseness) 그리고 불완전성(incompleteness) 때문에 완전한 요구사항을 정확히 추출하고 지정하는 것이 어렵다. 이와 더불어 요구사항 명세를 위한 효율적인 공통언어의 부재는 사용자와 개발자의 소통을 어렵게 하여 사용자의 프로젝트 참여를 방해한다. 위의 문제는 빈번한 요구사항의 변경을 초래하며 심지어는 프로젝트를 실패로 이끌어 갈 수도 있다.

본 논문에서는 이러한 변경사항 위험요소를 감소하고 프로젝트 성공률을 높이하고자 마인드 맵을 확장하여 요구사항 명세 시 명확하고 완전한 요구사항을 작성 가능하게 하는 도구를 개발하였다. 본 연구에서는 자연어 대신에 마인드 맵의 그래프 형태를 사용하여 요구사항 명세에 필요한 요소들을 나타내며 이로 인하여 사용자의 프로젝트 참여도를 높이고, 요구사항의 완전성과 정확성을 증대한다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 자연어로 작성한 요구사항의 완전성과 정확성을 높이기 위한 관련 연구를 논의한다. 제 3 장에서는 요구사항 명세와 마인드 맵에 대해 소개한다. 제 4 장에서는 마인드 맵을 확장하여 개발한 도구를 소개하고 제 5 장에서는 본 논문의 결론을 맺는다.

2. 관련 연구

자연어로 작성된 요구사항의 모호함과 부정확성 때문에 발생하는 여러 문제점을 해결하기 위한 여러 연구가 진행되었으며 다양한 방법들이 제안되었다[6-8].

Friedrich 외 2 명은 정확한 요구사항을 획득하기 위해 자연어 처리 도구를 사용하여 자동으로 자연어로 된 요구사항 문서를 비즈니스 프로세스 모델링(Business Process Modeling)으로 변환하는 방법을 제안하였다[6]. 비즈니스 프로세스 모델링은 요구사항 획득할 때 많이 사용되고 있으며 BPMN(Business Process Model and Notation)을 사용하여 자연어로 작성된 요구사항을 나타낸다. E. Chioaşc 는 자연어 고유의 문제인 모호성, 불완전성, 부정확성 등을 극복하기 위해 자연어 처리 도구를 사용하여 자연어를 OSMs(Object System Models)로 자동으로 변환하는 방법을 제안하였다[7]. 이 연구에서는 추상화 대상 모델을 사용하여 요구사항의 사양을 최소한으로 논의하도록 하고, 이에 대한 적절한 구조를 결정하여 요구사항을 재사용한다. 앞서 설명한 두 방법들은 요구사항 사양의 이해도를 높일 수 있으나 모델의 자동 생성으로 인하여 자연어로 작성된 요구사항들의 의미 손실이 일어날 수 있다. 또한, 사용자는 프로젝트 참여 시 모델의 대한 전문지식이 필요하기 때문에 전문지식 없이 프로젝트에 참여하기는 매우 어렵다.

F. Wanderle 외 3 명은 요구사항을 마인드 맵으로 변환한 다음 MDE(Model-Driven Engineering)을 통하여 마인드맵을 FODA(Feature-Oriented Domain Analysis) 모델로 변환하는 방법을 제안하였다[8]. 이 연구에서는 도메인 전문가(domain expert)와 소프트웨어 엔지니어 사이 의사 소통을 강화하여, 도메인 전문가의 프로젝트 참여도를 높였다. 그러나 도메인 전문가가 아니거나 전문지식이 없는 일반 사용자는 프로젝트에 참여하기 어렵다.

앞서 설명한 것과 같이 관련 연구에서 현재까지 제안된 방법들은 자동 모델 생성으로 인한 요구사항의 의미 손실 및 사용자가 프로젝트 참여 시 전문지식 필요 등의 단점을 가지고 있다. 따라서 본 논문에서는 마인드 맵을 확장하여 요구사항의 완전성 및 정확성을 높이고, 일반 사용자도 프로젝트에 참여할 수 있도록 사용자 경험 중심의 매우 직관적인 요구사항 명세 및 분석 도구를 제안한다.

3. 요구사항 명세와 마인드 맵

본 장에서는 요구사항 명세와 마인드 맵의 기본 개념과 특징에 대해 간단히 설명한다.

3.1 요구사항 명세

표 2. 요구사항 명세 예

Function Name	Order Ticket
Function Description	Providing ticketing services according to the required information inputted by a user.
Pre-Condition	User is already in the 'Order Ticket' page
Post-Condition	N/A
Basic Flow	1. Click 'Order Ticket' button 2. Go to 'Order Ticket' page 3. Input required information 4. Select a flight 5. Click 'Order' button 6. Make a payment 7. Receive the flight information
Constraints	1. Real Name: String 2. ID Number: String Verification: - string length must be 13 - ID format must be correct 3. Gender 4. Work Unit 5. Travel Time 6. Destination

요구사항 명세는 개발될 시스템의 전체 동작을 설명한다. 시스템과 사용자의 상호 작용을 구체적으로 설명하기 위해 사용 사례(Use Case)도 요구사항 명세에 포함할 수 있다. 또한, 요구사항 명세에 제약 및 검증의 형태로 다양한 비기능적 요구사항들도 포함할 수 있다. 표 2 는 관광 웹 사이트의 티켓 주문 기능을 나타내는 요구사항 명세의 예이다.

3.2 마인드 맵

마인드 맵은 단어나 문장의 제한 없이 연결되어 다양한 아이디어와 개념을 시각화하여 나타내는 도면이다. 즉, 이는 개념 또는 특정 도메인을 특정된 순서로 된 하나의 단어 또는 문자를 통해 모델링하고 표현하는 방사형 도면이다[8]. 마인드 맵을 사용하는 주요 장점으로는 키워드나 단어를 사용한 아이디어 수집 능력 증진, 아이디어 시각화 제공, 풍부한 창의력과 혁신 등이 있다[9]. 이뿐만 아니라, 마인드 맵은 직관적이고 간단하며 누구나 쉽게 이해할 수 있다.

4. 마인드 맵을 확장한 도구

요구사항 명세의 목적은 프로젝트 개발에 필요한 모든 요구사항을 나타내고 프로젝트의 개발자와 시스템의 사용자들이 요구사항을 명확하고 철저히 이해하는 것이다. 이를 위하여 본 연구에서는 개념이나 아이디어의 직관적 명세 및 이해를 가능하게 하는 마인드 맵을 확장하여 요구사항 명세중 기능에 대한 요구사항들을 하나의 마인드 맵에 나타나게 한다.

본 장에서는 요구사항 명세를 마인드 맵으로 쉽게 나타낼 수 있도록 확장한 도구를 설명한다. 이는 기능, 모듈, 제약의 속성 표현 형식, 표기법, 텍스트 상자, 플로우 차트 등을 확장하여 요구사항 명세에 포함한 요소들을 마인드 맵에 나타나게 한다. 다음절들은 표 2의 요구사항 명세 예와 비교하여 제안하는 도구를 설명한다.

4.1 기능, 모듈, 제약의 속성 표현 형식

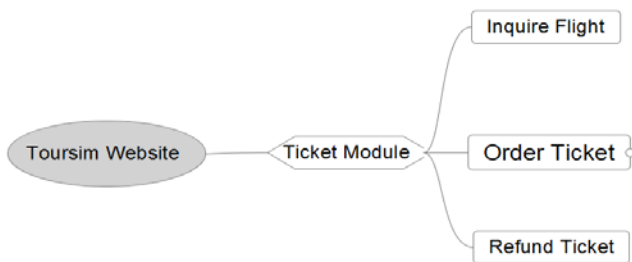


그림 2. 표현 형식

기능 모듈(Function Module)은 복잡한 문제를 여러 가지 쉬운 문제로 분해하여 해결하기 위해 도입한 개념이다. 위의 요구사항 명세 예에서 Order Ticket은 하나의 기능이고 Ticket Module 모듈에 포함된다. 이러한 요구사항의 개념과 아이디어는 마인드 맵에서 제공하는 다양한 노드 포맷을 사용하여 나타내며, 그림 2와 같이 는 프로젝트를 표시하고 는 기능 모듈을 표시하며 는 기능을 표시한다.

4.2 기능 설명

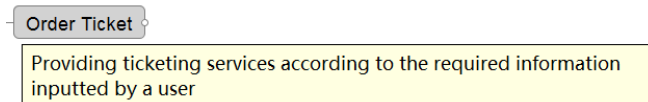


그림 3. 기능 설명

마인드 맵에는 개념과 아이디어에 대한 상세한 설명을 나타낼 수 있는 텍스트 상자가 있다. 본 연구에서는 이 텍스트 상자를 사용하여 명세된 요구사항 중 기능에 대한 설명을 표현하며, 해당 기능 위에 마우스 포인트가 멈추었을 때 이를 나타내도록 하였다. 예를 들어 그림 3은 위의 요구사항 명세 예 Order

Ticket의 기능설명을 텍스트 상자로 표시한 것이다.

4.3 전제 조건과 사후 조건



그림 4. 전제 조건과 사후 조건

요구사항 명세에서 기능을 설명할 때 기능에 전제 조건과 사후 조건이 있을 수 있다. 이를 그림 4에서 보이는 것과 같이, 마인드 맵에서 개념 혹은 아이디어를 연결할 수 있는 아크 라인을 사용하여 나타낸다. 조건표시는 사용자에게 간결하고 직관적인 요구사항 전체를 보여주기 위하여 숨겨져 있다가 기능이 선택되면 나타난다. 또한, 조건표시와 함께 조건에 대한 상세 설명을 추가할 수 있으며 아래 그림 5에서 보여지는 오른쪽 창은 전제 조건과 사후 조건에 대한 상세한 설명을 나타낸다. 이는 관광 웹 사이트의 결제기능을 시행하기 위해서는 반드시 위의 예 중 Order Ticket 기능을 통하여 티켓을 주문해야 한다는 전제 조건을 나타낸다.

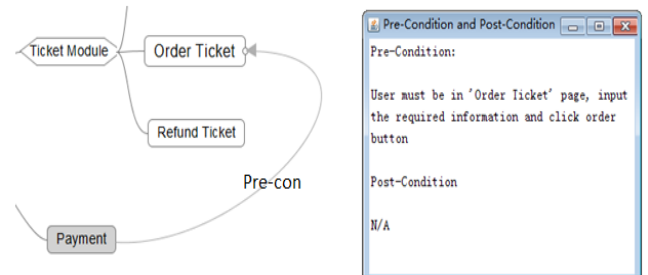


그림 5. 전제 조건과 사후 조건

4.4 기본 흐름

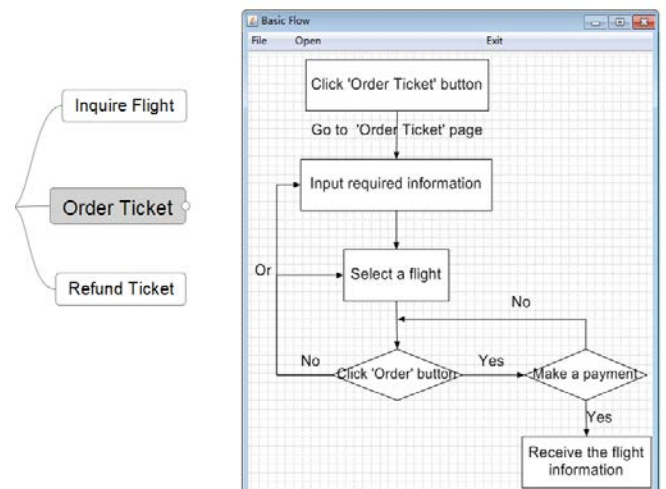


그림 6. 기능의 기본 과정

요구사항에 명세 된 기능의 기본 흐름(Basic Flow)을 표현하기 위하여 본 연구에서는 마인드 맵에 플로우 차트를 추가하였다. 요구사항 명세 시 각 기능의 기본 흐름은 플로우 차트로 표현할 수 있으며 해당 기능이 선택되면 새로운 창에 기술된 플로우 차트가 나타난다. 그림 6 은 Order Ticket 기능의 기본 흐름을 표현한 플로우 차트이다.

4.5 제약

요구사항 명세에 제약 사항은 마인드 맵에 배경 색상 있는 상자를 추가하여 나타낼 수 있다. 제약 조건 중 필수적인 속성은 *표시를 앞에 붙이며 볼드체로 표현하고 선택적인 속성은 특별한 표시 없이 그대로 표현한다. 속성의 개수에 대한 제약이 필요할 경우 표 3 과 같이 정의한다. ✓아이콘은 제약의 속성에 검증이 필요 한다는 것을 나타내고, 뒤의 검증할 내용은 키워드 또는 간결한 문장으로 설명한다.

표 3. 개수에 대한 정의

[n]	속성을 n 개로 제약할 경우
[m..n]	속성을 m-n 개로 제약할 경우
[1+]	속성을 1 개이상으로 제약할 경우

그림 7 은 Order Ticket 기능의 제약 조건을 나타낸다.

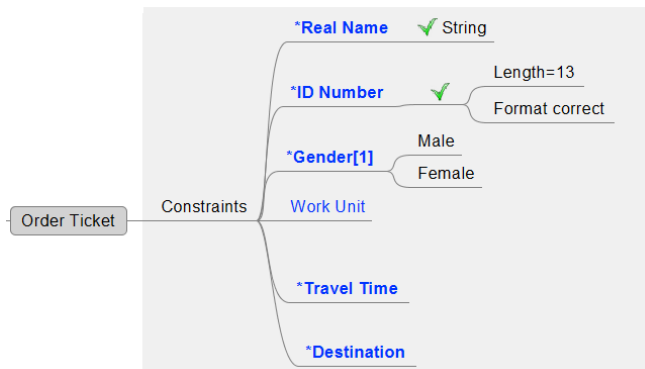


그림 7. Order Ticket 의 제약 조건

5. 결론 및 향후 계획

현재 소프트웨어 개발에서는 사용자의 활발한 프로젝트 참여를 위하여 요구사항을 자연어로 명세한다. 하지만 자연어의 모호함과 부정확함은 요구사항의 불완전성, 부정확성과 사용자의 참여 부족 등 소프트웨어 프로젝트가 실패하게 하는 문제들을 일으킬 수 있다. 이 문제들이 발생하게 되는 근본적인 원인은 요구사항 명세 시 사용자와 개발자 사이에 서로의 정보를 정확하고 완전하게 교환할 수 있는 효율적인 공통 언어가 없기 때문이다. 본 논문에서는

마인드 맵을 확장하여 요구사항 명세에 필요한 공통 언어를 제안함으로써 모호함을 감소시키며 정확성, 완전성, 그리고 사용자의 프로젝트 참여도를 높인다. 이는 효과적인 요구사항 명세를 가능하게 하여 소프트웨어 프로젝트의 성공률을 높인다.

향후 연구로는 우선 개발한 도구를 여러 실제 프로젝트에 적용하여 일반 자연언어로 된 요구사항 명세에 비해 사용자의 프로젝트 참여도와 요구사항의 정확도 향상을 분석하고 평가한다. 또한 추가적으로 개발한 도구의 가독성 및 이해성의 향상, 전체 시스템의 처리 흐름을 자동으로 생성하는 알고리즘 개발, 사용자 역할에 따른 기능 및 해당 모듈 기입, 기능의 우선순위 및 위험성 탐지 등을 향후 연구로 계획하고 있다.

참고문헌

- [1] IEEE Standard Glossary of Software Engineering Terminology, 610.12, 1990.
- [2] Boehm, Barry W., Philip N. Papaccio., "Understanding and Controlling Software Costs", IEEE Transactions on Software Engineering, vol.14, 1998.
- [3] Dean Leffingwell, "Calculating the Return on Investment from More Effective Requirements Management", American Programmer, no. 4, pp.13-16, 1997.
- [4] Robert B. Grady, "An Economic Release Decision Model: Insights into Software Project Management", Applications of Software Measurement, 1999.
- [5] I. Attarzadeh, "Project Management Practices: The Criteria for Success or Failure", Communications of the International Business Information Management Association(BIMA), vol.234-241, 2008.
- [6] F. Friedrich, J. Mendling, and F. Puhlmann, "Process Model Generation from Natural Language Text", Lecture Notes in Computer Science, vol. 6741, pp.482-496, 2011.
- [7] E. Chioaşcă, "Using Machine Learning to Enhance Automated Requirements Model Transformation", In the proceedings of the 34th International Conference on Software Engineering(ICSE), pp.1487-1490, 2012
- [8] F. Wanderley, D. S. Silveira, J. Araujo, and M. Lencastre, "Generating Feature Model from Creative Requirements using Model Driven Design", In the proceedings of the 16th International Software Product Line Conference(SPLC), pp18-25, 2012.
- [9] T. Buzan, "The Mind map Book", BBC Active, 2006.