

유즈케이스 시나리오에서 유즈케이스 패턴을 이용한 요구사항 대체 흐름 추천 기법

김유림*, 고덕윤*, 박수용*, 김진태**

*서강대학교 컴퓨터공학과

**소프트웨어공학엑스퍼트그룹(주)

{godrrna, maniar.k}@gmail.com, sypark@sogang.ac.kr, jtkim@swexpertgroup.com

요약: 본 논문은 유즈케이스 시나리오의 패턴을 이용하여 요구사항 대체 흐름을 추천하는 방안을 제안한다. 대체 흐름은 요구사항의 완성을 만족시키는 조건으로서 소프트웨어 시스템에도 영향을 미친다. 대체 흐름을 요구사항 명세서에 작성하기 위해서는 대체 흐름을 도출할 수 있는 명확한 기준이 선행되어야 하나 기존의 연구들은 이를 다루고 있지 않다. 유즈케이스 시나리오는 소프트웨어 시스템 프로젝트에서 일반적으로 사용되는 요구사항 명세서의 형식이다. 또한, 유즈케이스 시나리오는 사용자와 시스템 등의 주체가 행하는 행위를 위주로 작성된다. 그러므로 유즈케이스 시나리오의 주체와 행위를 식별할 수 있다면, 대체 흐름을 추천하는 기준점을 마련할 수 있다. 이에 본 논문에서는 유즈케이스 시나리오의 패턴을 기반으로 요구사항 대체 흐름을 도출하는 방안을 제안한다. 제안된 방안은 도메인이 상이한 세 가지 시스템 소프트웨어 요구사항 예제를 통해 적용 방안을 설명한다.

핵심어: 요구사항, 대체 흐름, 유즈케이스 시나리오, 유즈케이스 시나리오 패턴

1. 서론

Standish 보고서에 의하면 프로젝트가 실패하는 대부분의 이유는 불명확한 요구사항과 불완전한 요구사항 때문으로 밝혀졌다[1]. 이와 같이 소프트웨어의 요구사항은 프로젝트의 성패를 결정하는 중요한 요인 중 하나이다[2].

요구사항의 완성을 높이기 위해서는 요구사항의 기본흐름(Basic flow)과 대체흐름(Alternative flow)이 식별되어 기술되어야 한다. 기본흐름은 사용자가 원하는 목적을 성공적으로 수행하는 시나리오이다. 대체흐름은 기본흐름과 같은 목적을 성공적으로 수행하는 또 다른 시나리오 혹은 실패할 경우의 시나리오이다[3].

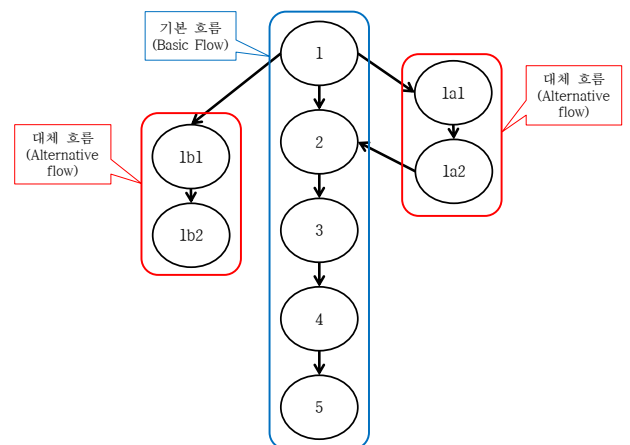
대체 흐름이 요구사항의 완성도에 있어 중요함에도 불구하고 간과되는 이유들은 다음과 같다.

- 1) 대체 흐름에 대한 이해부족
- 2) 중요성에 대한 인식 미흡
- 3) 대체 흐름을 도출할 수 있는 불명확한 기준
- 4) 작성자의 인식 한계 또는 경험으로 인한 불완전한 요구사항

2. 접근 방안

2.1 용어 정의

본 논문에서는 [그림 1]과 같이 기본 흐름은 하나의 유즈케이스 흐름에서 다른 선택의 여지가 없는 하나의 경로로 정의한다. 즉, “확장”이 없는 유일한 하나의 경로이다. 그리고 대체 흐름이란 기본 흐름에 해당하지 않는 그 밖의 모든 경우를 일컫는 것으로 일반적으로 기본 흐름에서 확장해 나가 다른 선택의 여지가 있는 경우 혹은 예외 상황이 발생하여 기본 흐름으로 끝나지 않는 경우에 해당한다.



[그림 1] 기본 흐름 및 대체 흐름

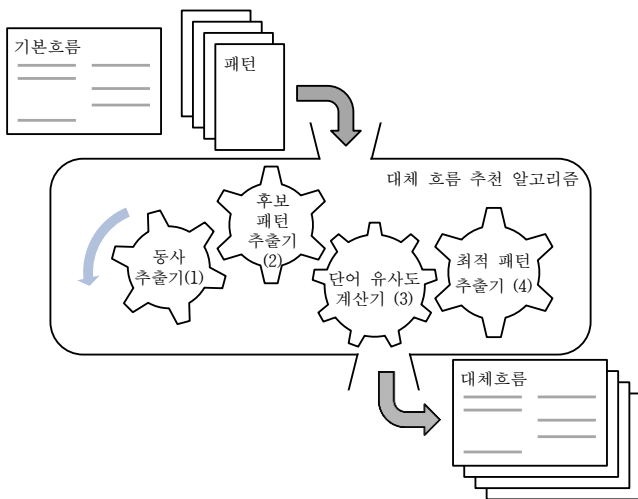
2.2 필수 유즈케이스 및 필수 유즈케이스 패턴

필수 유즈케이스를 통해 주체를 사용자와 시스템으로 명확하게 나누고, 이를 기반으로 요구사항을 주체와 관련된 행위를 위주로 작성한다.

사용자와 시스템간의 대화 진행순서 및 대화가 내포하고 있는 의미를 기준으로 필수 유즈케이스 패턴이 개발되었다[4]. 필수 유즈케이스 패턴은 요구사항 명세서 작성 시 누락될 수 있는 부분을 최소한으로 줄일 수 있다[5].

3. 제안 방안

대체 흐름을 도출할 수 있는 불명확한 기준과 작성자의 인식 한계로 인하여 대체 흐름이 누락되어 불완전한 요구사항이 발생한다. 따라서 이러한 문제점들을 해결하기 위해서 본 논문에서는 요구사항 명세서 작성자가 입력하는 요구사항을 식별하여 대체 흐름을 추천하여 줄 수 있는 필수 유즈케이스 패턴의 확장 및 대체 흐름 추천 알고리즘을 제안한다. [그림 2]는 본 논문에서 제안하는 방안을 개괄적으로 보여준다.



[그림 2] 제안 방안 개요

- 1) 입력된 기본 흐름의 각 문장을 동사 추출기(1)를 통해 문장을 분석하여 문장 내 본동사를 추출한다.
- 2) 입력된 기본 흐름의 문장들 간 사용자 의도와 시스템 반응의 상호작용(사용자와 시스템 간의 대화순서)을 패턴 추출기(2)를 통해 분석하여 제안된 패턴을 기반으로 적합한 후보 패턴을 도출한다.
- 3) (1)에서 도출된 동사와 본 논문에서 제안한 패턴 내에 사용된 동사들 간의 유사도를 유사도 계산기(3)를 통해 계산한다.
- 4) (3)에서 계산된 동사 유사도를 기반으로 패턴 추출기(4)를 이용하여 최적의 패턴들을 도출한다.

5. 결론

본 논문은 요구사항 대체 흐름을 도출하는 불명확한 기준점으로 인해 발생하는 실수 및 작성자의 인식 한계로 인한 대체 흐름의 누락이나 생략 때문에 발생하는 불완전한 요구사항을 줄이는 방안으로 대체 흐름 도출 기법을 제안 하였다.

본 논문에서 제안하는 기법을 소프트웨어 시스템의 요구사항에 적용해 본 결과 대체 흐름 추천 알고리즘을 사용하여 도출된 대체 흐름을 기반으로 도메인 전문가가 대체 흐름을 도출하는 경우, 도출된 대체 흐름의 개수 및 중요도 모두에서 각기 제안된 방안 중 가장 좋은 결과를 얻음을 확인할 수 있었다. 이 결과로 본 논문에서 제안한 알고리즘을 사용하여 도출된 대체 흐름이 작성자에게 대체 흐름을 도출하는 기준점을 마련하여 대체 흐름 도출에 기여한다는 것을 확인하였다. 또한, 작성자의 인식 한계로 놓칠 수 있는 대체 흐름에도 영향을 끼쳐 보다 나은 결과의 대체 흐름을 도출 해낼 수 있었다.

사사

이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술 개발사업의 지원을 받아 수행된 연구임 (No. 2012M3C4A7033348).

참고문헌

- [1] Robert L. Glass, The Standish Report: Does It really describe a software crisis?, COMMUNICATIONS OF THE ACM August 2006/Vol. 49, No. 8
- [2] The Standish Group, "Software Chaos", <http://www.standishgroup.com/chaos.html>
- [3] Karl E. Wiegers, Software Requirements 2, 2003.
- [4] Robert Biddle, James Noble, Ewan Tempero, Pattern for Essential Use Case, CRPIT '02 Proceedings of the 2002 conference on Pattern languages of programs - Volume 13, Pages 85-98
- [5] Biddle, R., Noble, J. and Tempero, E. 2002. "Essential use cases and responsibility in object-oriented development." In *Proceeding of the twenty-fifth Australasian conference on Computer science*, Melbourne, Victoria, Australia, 2002, ACM, pp. 7-16.