

추적성 매트릭스 메카니즘 기반의 요구사항 추적 구현 사례

이진협[○] 김기두¹ 박보경[○] 김영철[○]

[○]홍익대학교 소프트웨어공학 연구실, ¹한국정보통신진흥협회(TTA)

[○]{jh, park, bob}@selab.hongik.ac.kr, ¹kdkim@tta.or.kr

An Implementing Practice of Requirement Traces based on Traceability Matrices Mechanism

[○]Jin Hyub Lee, ¹Kidu Kim, [○]Bo Kyung Park, [○]R. Young Chul Kim

[○]Software Engineering Lab. Hongik University, ¹Telecommunications Technology Association

요 약

소프트웨어 개발 분야에서 개발 프로세스는 복잡성과 비가시적 특성으로 인해 관리가 어렵다. 전 단계에서의 추적성이 확보되면, 가시화하고 전체 프로세스를 더 쉽게 이해할 수 있다. 추적성 부여는 가장 보편적으로 '추적 링크'가 사용된다[1]. 추적 링크를 부여하기 위해 Redmine에서는 '연관 일감'을 사용하지만, 추적 링크의 수가 많아질 수록 복잡해지고 알아보기 어렵다. 본 논문에서는 필요한 데이터만 추출하여 체크만으로 추적성을 부여할 수 있는 추적성 매트릭스 메카니즘 기반의 요구사항 추적 시스템을 제안한다. 이를 통해 사용자가 쉽고 빠르게 추적성을 부여 및 관리할 수 있어, 소프트웨어 프로세스와 제품의 품질을 높일 수 있을 것으로 기대한다.

1. 서 론

최근 국내에서 추적성에 대한 관심이 더욱 높아지고 있는 추세이다. 추적성을 통해 소프트웨어 개발 프로세스 내에서 요구사항 추가/변경이나 이슈 발생 등 변경이 발생하는 단계가 영향을 미치는 다른 단계의 해당 요소를 쉽게 찾을 수 있다. 이는 설계 단계 이후에는 프로젝트의 시간적/인적 비용을 상당히 줄여준다. 소프트웨어 개발 프로세스에서 추적성은 각 단계에서의 산출물을 기반으로 이루어진다. 산출물은 단계에 따라 소스 산출물, 타겟 산출물로 나누어진다. 소스 산출물은 추적을 위해 시작이 되는 데이터이다. 타겟 산출물은 추적의 대상이 되는 데이터이다. 이 소스 산출물과 타겟 산출물을 이어 주는 것이 '추적 링크'이다[1].

현재 많은 상용 소프트웨어와 오픈 소스 소프트웨어에서 이미 추적성을 지원하고 있다. 이들 도구들의 대부분은 링크를 이용해 추적성을 부여한다. 이전 연구에서 소프트웨어 프로세스 자동 구축에 프로젝트 관리 도구로 Redmine을 이용하였으므로, 이 논문에서는 Redmine을 기반으로 추적성을 관리한다[2]. Redmine에서는 '연관 일감'을 이용해 링크를 생성하고 추적성을 부여한다. 이를 사용해본 결과 모든 일감에 일일이 접근하여 일감을 서로 구분해주는 기준인 '일감 ID'를 연관 일감에 등록해 주는 방법은 간단하고 쉽지만, 추적 링크가 많아질 수록 상당히 불편하고 시간이 오래 걸렸다.

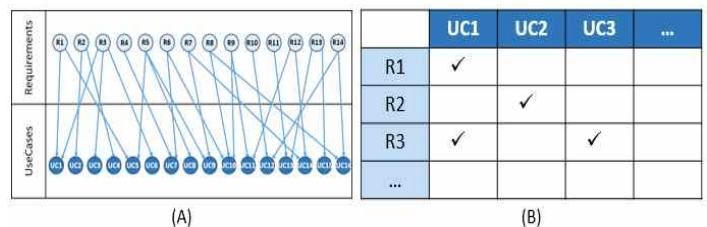
이를 개선하기 위해 이 논문에서는 이전 연구에서 소프트웨어 프로세스 자동화를 통해 구축한 시스템에 추적성을 좀 더 쉽고 편리하게 부여할 수 있고, 수정, 삭제할

수 있는 방안 제안한다[2].

이 논문의 구성은 다음과 같다. 2장에서 Redmine에서 기본적으로 사용되는 추적 생성 방법과 추적성 매트릭스에 대해 언급하고, 3장에서 추적성 매트릭스 기반의 추적성 부여 방안과 구현된 시스템에 대해 설명한다. 4장에서 결론 및 향후 연구에 대해 언급한다.

2. 기존 요구사항 추적 표현 방법

Redmine에는 일감 기능이 있다. 일감 내에 속성인 일감 유형과 일감 상태 등을 이용해 이슈 관리, 프로젝트 관리, 일정 관리 등 프로젝트 상에서 다양한 부분에 활용되고 있다. 이 중 '연관 일감'은 일감 간에 상관 관계를 정의할 때 사용된다. Redmine에서 연관 일감 생성 시에 이미 등록되어있는 일감의 ID인 번호를 입력해 주어야 한다. 때문에 ID번호를 일일이 확인 후 해주어야 하는 불편함이 있다.



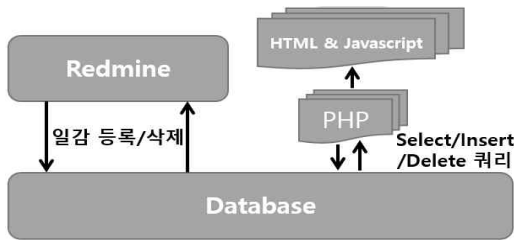
(그림 1) 요구사항-유스케이스 추적성 및 요구사항 추적 매트릭스 예

추적성 매트릭스는 추적 관계를 구성하는 추적을 기록하는 매트릭스로, 추적 링크를 통해 연결된 추적 산출물 쌍을 표시한다[1].

그림 1은 요구사항과 유스케이스 간의 추적성과 요구사항 추적성 매트릭스를 간단히 보여주는 예이다[3]. (A)에서 동그라미를 산출물이라 정의하였을 때, Requirements 내 동그라미들은 소스 산출물이고, Use Cases 내의 동그라미들은 타겟 산출물이다. 그 사이를 연결하는 선들은 추적 링크이다. 이로써 요구사항 추적성을 의미한다. 반면, 그림 2의 (B)에서는 표의 column은 소스 산출물이고, row는 타겟 산출물이다. 체크 표시는 소스 산출물과 타겟 산출물 간의 링크 존재 여부를 알려준다. 이는 연관관계가 많아져도 R1에 영향을 받는 Use Case들이 무엇인지 (A)에 비해 쉽게 파악할 수 있다.

3. 추적성 매트릭스 기반 요구사항 추적 시스템 구현

요구사항 추적 시스템은 이전 연구인 소프트웨어 프로세스 자동화에 포함된 오픈소스 소프트웨어인 Redmine을 기반으로 하고 있으며, DB에 저장된 일감 데이터를 추출하여 추적성 매트릭스를 표현한다. Redmine이 웹 기반이므로, 이 시스템 또한 웹 기반으로 구현하였다. 이 시스템의 구성도는 그림 2와 같다. Redmine에서는 일감 등록/삭제를 통해 데이터베이스에 저장하고, Select/Insert/Delete 쿼리로 데이터를 추출하여 표현한다.



(그림 2) 요구사항 추적 시스템 구성도

일감과 custom_values 테이블을 조인한 결과를 매트릭스의 column과 row에 추가한다. 표 내에 상관 관계를 나타낼 모든 cell에 checkbox를 추가하고, 체크 여부에 따라 column 값과 row 값을 각각 추출하여 column에 해당하는 일감의 id를 issue_relation 테이블의 issue_from_id 필드에, row에 해당하는 일감의 id를 issue_to_id 필드에 insert한다.

추적 링크를 수정할 때는 수정 버튼을 클릭하면 custom_value에서 customized_id를 추출하여 issue_relations 테이블에 해당하는 issue_from_id, issue_to_id로 구성된 row가 있는지 여부를 exists 쿼리로 탐색하여 있으면 1을 반환받아 체크박스에 checked 상태로 출력한다. 수정 후 적용 버튼 클릭 시, 해당 추적성 매트릭스의 column, row들에 해당되는 모든 추적 링크를 삭제한다. 그 후에 해당 관계들을 다시 insert하는 방법으로 수정된다.

4. 결론 및 향후 연구

추적성을 부여하는 방법으로 ‘추적 링크’가 가장 많이 사용되고 있다. Redmine에서는 이 추적 링크를 issue_relations 테이블을 이용해 소스 산출물은 issue_from_id 필드에, 타겟 산출물은 issue_to_id 필드에 저장한다. 하지만, 기존에 각 일감에 접근하여 연관 일감을 직접 추가해주는 추적 링크를 생성하는 방법은 링크 수가 많을 수록 복잡하고, 가시적으로 알아보기도 어렵다. 본 논문에서 구현된 시스템을 이용하면 추적성 매트릭스 형태로, 체크만으로 소스 산출물과 타겟 산출물 간의 링크를 관리할 수 있다. 또한 매트릭스이므로 하나의 요구사항이 어떤 유스케이스들과 연관있는지 쉽게 파악할 수 있다. 이는 소프트웨어 개발 프로세스에서 변경 발생 시, 추적성을 통해 변경 영향을 쉽게 파악하여 시간적/인적 비용을 줄일 수 있을 것으로 기대한다.

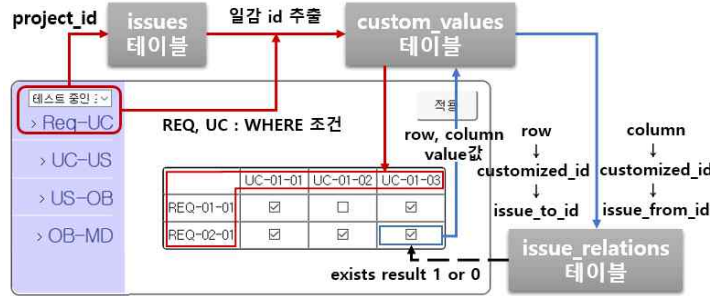
향후 연구에서는 현재 customized_value 테이블 내에 저장된 식별자들만으로 산출물들을 구분해야하는데, 그 수가 많아 일일이 확인해야하는 문제점을 해결하는 방안을 연구할 예정이다.

ACKNOWLEDGEMENT

이 논문은 2017년도 정부(교육부) 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NRF-2017R1D1A3B03035421)과 2018년도 정보통신산업진흥원의 정보통신, 방송 연구개발 사업(개방형OS 환경개발 및 보급, 확산 사업)의 지원을 받아 수행된 연구임(S1113-18-1001).

참고 문헌

[1] Orlena Gotel, Jane Cleland-Huang, Jane Huffman Hayes, et al., “Traceability Fundamentals”, Software and Systems Traceability, pp3-22
 [2] 이진협, 장우성, 박지훈, 박용범, 김영철, “자동 소프트웨어 프로세스 구축을 위한 공개 소프트웨어 툴 체인 연구”, 제 20회 한국 소프트웨어공학 학술대회 논문집, Vol.20 No.1, pp148-149, 2018
 [3] 변은영, 손현승, 문소영, 박지훈, 김영철, “클로즈 아키텍처 메커니즘 기반의 요구사항 추적성 매트릭스”, 한국정보처리학회 2017년 추계학술발표대회 논문집 제24권 제2호, pp.631-634, 2017



(그림 3) 요구사항 추적 시스템 흐름도

그림 3은 요구사항 추적 시스템의 흐름도이다. Redmine에 산출물에 해당하는 요구사항, 유스케이스, 유스케이스 시나리오 등을 일감으로 등록한다. 그 후 일감 데이터가 DB 내 issues 테이블에 저장된다. 그 후 추적 링크 관리 시스템에 접속하면 Redmine에 저장된 프로젝트 목록이 나타난다. 프로젝트를 선택하고 Req-UC, UC-US, US-OB, OB-MD 중 하나를 클릭한다. php에서 쿼리를 통해 issues 테이블에서 선택된 프로젝트 id에 포함된 select된