

요구사항추적테이블의 확장 및 정규화 방안

김 주 영[†] · 류 성 열^{**}

요 약

소프트웨어 개발시 요구사항의 일관성 및 완전성을 검증하기 위해 다양한 방법으로 산출물을 추적하는데, 기존 연구에서는 추적메타모델 또는 자동화된 도구를 제시하되 구체적인 추적산출물 및 추적요소 선정이 미흡하며, 추적요소를 제시하는 추적테이블 연구의 경우는 전체공정이 아닌 일부 공정만을 대상으로 추적하고 있다. 이에 본 연구는 일부 공정만 추적한 저자의 이전 추적테이블연구에 이어 프로젝트시작단계에서 아키텍처단계, 인도단계까지의 산출물을 추적한 확장된 요구사항추적테이블을 제시하였다. 또한 추적테이블이 확장됨에 따라 추적필드가 많아져서 추적이 복잡해지는 문제점을 해결하기 위해 개발프로세스단위로 통합/분리할 수 있는 추적테이블의 정규화 방안을 제시하였다. 본 연구를 D사의 H시스템개발프로젝트에 적용함으로써 연구의 적용가능성을 검증하고 요구사항의 오류를 쉽게 발견하는 추적의 효과를 확인하였다. 이에 연구를 통해 요구사항의 일관성과 완전성을 검증하는 추적의 정확성을 높여 소프트웨어 개발의 실패를 최소화하고자 하였다.

키워드 : 요구사항, 요구사항추적, 추적테이블, 추적성, 일관성, 완전성, 마르미방법론

Requirements Trace Table Expansion and How to Normalization

Kim Ju Young[†] · Rhew Sung Yul^{**}

ABSTRACT

There are various methods to trace output in software development to verify the consistency and completeness of requirements. Existing studies do present the trace meta-model or automated tools, but fail to list specifically traced output or traced items. Also, in regards to trace tables, which contain traced items, existing studies don't consider the whole software development process but merely its sub-process.

Given this, the present study suggests an extended requirements tracetable that tracks output from the inception of the project through the architectural design phase to the application delivery, following up on the researcher's previous study on a tracetable that considered only a sub-process of the whole development process. In addition, in order to address the problem of the tracking process becoming complicated with increased tracefields due to an extended trace table, the researcher suggests a method for normalizing a requirements tracetable that can integrate and separate by development process phase. Apply it to the H system development project of a D company, and this study caseverified application possibility of study, confirmed an effect of a chase to easily find out an error of requirements. Improve precision of a traceto verify consistency of requirements and completeness through this study, and will minimize failure of a software development.

Keywords : RequireMents, Requirements Traceability, Trace Table, Traceability, Consistency, Completeness, Marmi Methodology

1. 서 론

Standish 그룹에서는 소프트웨어개발프로젝트의 실패이유로 요구사항이 불완전하고 자주 변경된다는 게 전체의 36%를 차지한다고 보고하였고 A.Davis는 소프트웨어 개발 생명주기 후반에 에러가 발견되는 경우 수정에 많은 비용이 소요되는데, 이때 발생하는 에러의 대부분이 요구사항 관리의

소홀로 발생한다고 조사하였다[1]. 이에 성공적인 소프트웨어개발을 위해서는 요구사항을 잘 명세하고 변경을 잘 관리해야 하는데, 이를 위해서는 산출물간의 일관성 및 완전성을 정확하게 추적하는 것이 필요하다.

요구사항을 추적하는 일반적인 방법으로는 추적링크에 따라 요구사항을 순방향 또는 역방향으로 추적하거나 상용도구를 이용한 추적방법을 들 수 있다. 이중 추적링크로 추적하는 연구에서는 추적 메타모델을 제시하고 추적링크를 생성하거나 추적테이블을 이용한 추적 방법 등을 제안한다.

그런데 추적 메타 모델을 통한 추적링크에 관한 연구인 Ramesh의 참조모델에서는 모델을 제시하되, 그 구체적 실

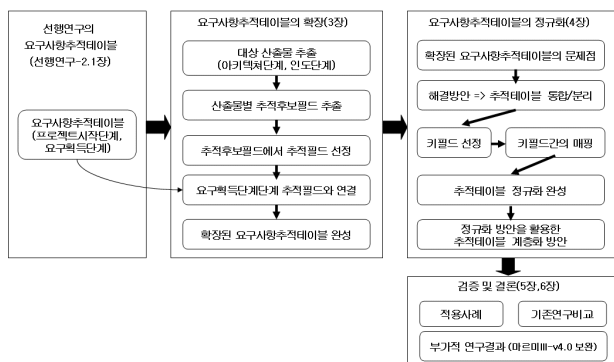
※ 본 연구는 숭실대학교 교내 연구비 지원으로 이루어졌음.
† 정 회 원 : 숭실대학교 컴퓨터학과 박사수료
** 종신회원 : 숭실대학교 컴퓨터학부 교수
논문접수 : 2008년 9월 29일
수 정 일 : 1차 2008년 12월 12일
심사완료 : 2008년 12월 12일

협사례가 부족하여 연구의 결과를 활용하고자 할 때 한계가 있으며, 안상임의 추적메타모델에서는 추적링크 생성시 요구사항의 완전성을 전제하지 않아 요구사항 추적의 모순을 내포하고 있다 [2][3]. 또한, 추적테이블을 이용한 연구로써 본 연구의 선행연구이기도 한 김주영의 추적테이블 연구는 전체 개발 프로세스를 대상으로 하고 있지 않다[4]. 한편, 상용도구에서는 추적모델에 따른 구체적 추적링크를 제시하지 않아 사용자에게 주관적 판단에 의존하여 추적링크가 생성되는 한계가 있다[5].

이에 본 연구에서는 프로젝트시작단계에서 아키텍처단계까지의 산출물을 주요 대상으로 하되, 인도단계산출물도 포함한 요구사항추적테이블을 연구하였는데, 이는 본 연구의 이전 연구에서 제안된 요구사항추적테이블을 확장한 것이다. 또한 추적테이블 확장으로 인해 추적필드가 많아져 추적이 복잡해지는 문제점이 야기되었는데, 이를 해결하기 위해 개발프로세스단위로 키가 되는 추적필드를 선정하고, 선정된 키필드간의 매핑을 통해 추적테이블을 개발프로세스단위로 통합/분리하여 사용할 수 있는 추적테이블 정규화 방안을 제시하였다.

한편, 연구를 위해 마르미-III v.4.0의 산출물을 사용하였는데, 이는 특정 방법론을 정한한계가 있으나 추적의 구체성과 추적필드의 객관적 도출을 위해 의도하였다. 연구를 진행하면서 인용한 마르미 산출물에 보완이 필요한 요소를 발견하였기에 추적테이블과 함께 부가적 연구결과로 제시한다.

연구의 구성은 (그림 1)과 같다. 2장에서 관련연구를 열거하고 3장에서 요구사항추적테이블 작성과정 및 결과를 제시하였는데, 먼저 추적테이블을 만들기 위한 추적필드를 추출하는 과정을 연구하였고, 여기서 추출된 추적필드를 토대로 확장된 추적테이블을 완성하였다. 4장에서는 추적테이블 확장시 문제점을 발견하고, 해결방안으로 추적테이블을 정규화하여 개발프로세스 단위로 통합/분리하는 방안을 제시하였다. 5장에서는 이렇게 연구된 결과를 검증하는 방법으로 D사의 H시스템개발프로젝트에 적용해 보았으며, 기존 연구와의 비교를 통해 본 연구의 한계점과 장점을 분석하였다. 마지막 6장에서는 본 연구의 부가적 연구결과 및 결론을 제시하고 향후 연구방향을 정리하였다.



(그림 1) 연구의 구성

2. 선행 및 관련 연구

2.1 선행연구

2.1.1 김주영의 요구사항추적테이블의 경험적 연구[4]

본 연구의 이전 연구인 김주영의 요구사항추적테이블의 경험적 연구에서 마르미방법론III-v4.0의 프로젝트시작단계에서 요구사항단계 산출물을 대상으로한 요구사항추적테이블의 개선연구가 (그림 2)와 (그림 3)과 같이 진행되었다.

연구에서는 추적필드 추출과정과 추출된 추적필드로 구성된 추적테이블을 기능 및 비기능요구사항으로 나눠서 제시하였는데, 개발 프로세스 중에서 요구사항단계까지의 산출물만으로 정한 한계가 있다. 이에 본 연구는 선행연구의 확대연구로써 아키텍처단계 산출물을 포함한 전체 프로세스의 주요 산출물로 그 추적대상을 확장하였는데, 이로 인해 추적테이블의 추적필드가 많아져 추적이 복잡해지는 문제점이 발생하여 추적테이블을 정규화하는 방안도 연구하였다.

2.2 관련 연구

2.2.1 안상임의 가치분석을 통한 휘저기반의 요구사항 추적 및 변경영향 분석 [3]

안상임의 가치분석을 통한 휘저기반의 요구사항 추적 및 변경 영향 분석에서는 요구사항을 휘저기반으로 분류하고, 분류된 휘저기반 요구사항에 대해 가치분석을 한다. 이 가치분석으로 우선순위를 매겨서 추적링크의 상세화 정도를 결정한다. 다음 요구사항들의 추적링크 생성하는 절차를 제시하고 있는데, 추적링크 생성 위한 방법을 비효율적 측면에서의 메타모델로 제공했다는 점에서 연구가치가 있다. 한편 안상임의 연구는 요구사항의 가치정도에 따라 추적링크 생성여부 및 상세화 정도를 구분하기 때문에 이는 모든 요구사항이 지켜지는지를 확인하는 요구사항 완전성 추적에는 한계가 있다.

요구사항 ID	프로젝트준비			프로젝트시작			요구사항					
	프로젝트 기안서 프로젝트 정의서	프로젝트 제안서	프로젝트 수행계획서	요구사항	사용자/관련 부서	구현방안/해결방안	우선 순위	범위 여부	유스 케이스	클래스	컴포넌트	인수 카운 ID
R					A							

(그림 2) 기능요구사항추적테이블

요구사항 ID	프로젝트준비			프로젝트시작			요구사항				
	프로젝트 기안서 프로젝트 정의서	프로젝트 제안서	프로젝트 수행계획서	품질관리 계획서	요구사항	사용자/관련 부서	구현방안/해결방안	우선 순위	범위 여부	인수 카운 ID	
R											

(그림 3) 비기능요구사항추적테이블

2.2.2 Balasubramaniam Ramesh의 Toward Reference Models for Requirements Traceability [2]

IEEE에 발표된 Balasubramaniam Ramesh 의 Toward Reference Models for Requirements Traceability 에서는 요구사항 추적을 위한 참조모델을 이해당사자(Stakeholder), 객체(Object), 코드(Source) 등 3가지 관점의 추적 메타 모델로 제시한다. 각 관점들과 그 세부요소들간의 관계(Relationship)을 통해 추적링크를 생성하는 모델로 제약사항, 정책, 표준 등도 요구사항의 객체로 간주하여 추적링크를 생성하고 있는데, 이해당사자를 비롯하여 요구사항 추출시 관련된 많은 객체들을 추적하는 한편, 코드까지 산출물이 개발단계를 진화해가면서 요구사항이 틀리지 않게 지속적으로 지켜지는지에 관한 일관성을 추적하고 있어 추적의 시작과 링크 선정에 있어 좋은 참고가 되고 있다. 그러나 메타모델만 제시하고 있어 구체적인 추적 산출물을 가능하는데 한계가 있다.

2.2.3 Lee M. Williams의 요구정보관리시스템의 개념 설계 [6]

INCOCE Atlanta Chapter에서 발표된 Lee M. Williams 의 The Elements of Technical Requirements Management 에서는 요구사항을 관리하기 위한 정보시스템의 개념을 제안하고 있다. 요구사항을 데이터베이스로 저장하면서 다른 산출물간의 일관성과 완전성을 검증하기 위해 각 산출물 마다 항목단위별로 식별자를 두고, 그 식별자를 각 산출물간에 참조하도록 설계하고 있다. 즉 산출물마다에 포함된 요구사항항목의 식별자를 통해 요구사항을 추적하는 것이다. 요구사항 각 항목마다 식별자를 두는 것은 통상적인 추적테이블이나 상용 도구 등에서 많이 쓰이는 방법인데, 이 연구에서는 특히 설계 및 개발명세뿐만 아니라 위험명세와 기술성능측정명세 산출물과 요구사항간의 추적도 보여주고 있어서 개발산출물과 관리산출물간의 연결을 시도한 좋은 참조가 되고 있다.

2.2.4 상용도구를 이용한 요구사항관리 [5]

일반적으로 요구관리도구로 사용되는 상용도구에는 Requisite Pro, Doors(Dynamic Object Oriented Requirements System), Caliber RM (Caliber Requirements Management), RTM workshop 등이 있는데, 자동화된 추적이 주요 기능으로써 변경 등에 따른 수작업을 줄여주며, 기록과 저장이 탁월하여 프로젝트 팀원이 공유하여 사용할 수 있다는 장점이 있다. 그러나 산출물에서 추적필드를 추출하는 것은 자동화되지 않기 때문에 사용자의 판단에 따라 추적필드를 정의해야 하며, 경제적 요소로 구입비용 및 사용법 숙지에 따른 간접비용(overhead)이 발생한다는 단점이 있다.

5장에서 상용도구를 포함한 기존 연구들과 본 연구의 추적테이블을 비교할 것이다.

3. 요구사항추적테이블의 확장

3.1. 추적대상 산출물 및 추적 필드 추출

3.1.1 아키텍처단계 추적 대상 산출물 및 추적후보 필드 추출

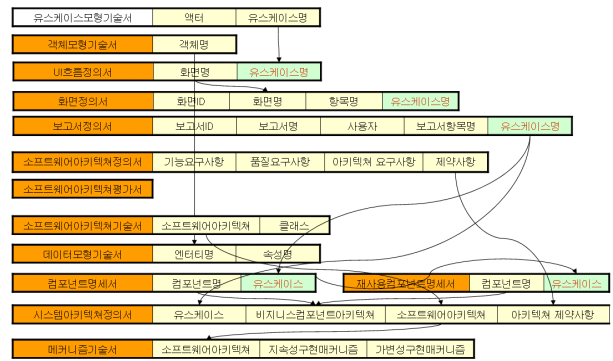
요구사항추적테이블의 확장은 요구획득단계까지의 산출물로 추적테이블이 작성된 선행연구에 이어서 아키텍처단계를 거쳐 인도단계까지의 산출물까지 추적테이블에 포함하겠다는 의미이다. 이에 확장방법은 선행연구의 추적테이블은 그대로 수용하고, 아키텍처단계에서 인도단계까지의 산출물에서 추적대상 산출물을 추출하고 산출물별 추적후보필드를 선정, 그 중에서 추적필드를 식별하여 선행연구의 추적테이블에 이어 붙임으로써 완성된다.

따라서 본 3장은 아키텍처단계에서 인도단계까지의 추적필드를 추출하는 것부터 시작한다. 마르미-III v4.0에서 정의하고 있는 아키텍처는 크게 컴포넌트아키텍처, 소프트웨어 아키텍처, 기술아키텍처로 나누고 있다[7]. 이 세가지 산출물을 포함한 아키텍처단계의 전체 산출물들간의 연관성을 토대로 산출물간의 추적대상 산출물과 추적필드를 추출하기 위해 산출물의 입력속성을 <표 1>과 같이 분석한다.

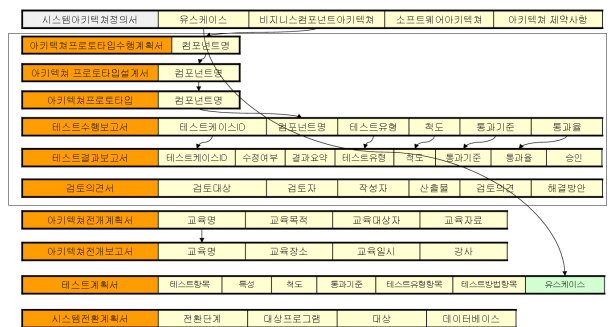
입력속성들 중 필수입력속성으로 정해진 속성은 산출물을 추적할 때 해당 산출물의 주요 추적대상 속성이 될 추적필드 후보이다.

3.1.2 아키텍처단계 산출물의 추적 후보 필드에서 추적 필드 추출

<표 1>의 필수입력속성 중에서 추적테이블에서 추적용으로 사용될 필드를 정하기 위해 (그림 4), (그림 5)에서 보는 바와 같이 산출물간에 추적성을 도식화하였다.



(그림 4) 아키텍처단계 산출물간의 요구사항 추적



(그림 5) 아키텍처단계 산출물간의 요구사항 추적II

요구획득단계에서 최초로 나온 유스케이스모형기술서의 액처정의서, 테스트계획서로 추적되고 있다. 유스케이스가 UI설계서를 거쳐 컴포넌트명세서, 시스템아키텍처 정의서, 테스... 한편, 마르미-III v4.0에서 UI설계서, 컴포넌트명세서, 테...

〈표 1〉 아키텍처단계 산출물 간의 관계

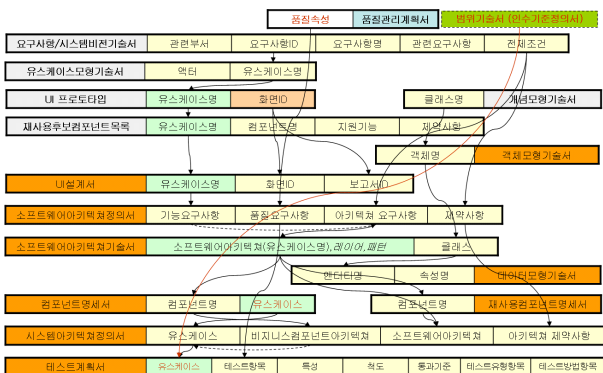
산출물명	입력물	입력 속성	필수입력 속성	선택입력 속성
유스케이스 모형기술서	업무유스케이스모형기술서 프로젝트계획서 아키텍처단계작업계획서 아키텍처단계평가서 유스케이스모형기술서(기준)	시스템개요 액터,액터구분,액터설명 유스케이스도 유스케이스(이름,개요,사전호름,사전/후조건,확장점,가변성,시나리오,비기능적요구사항,관련화면,관련 다이어그램)	시스템개요 액터 유스케이스(이름,사전호름)	시스템개요 액터구분,액터설명 유스케이스도 유스케이스(개요,사전/후조건,확장점,가변성,시나리오,비기능적요구사항,관련화면,관련 다이어그램)
객체모형기술서	유스케이스모형기술서 개념모형기술서	클래스도 클래스(이름,개요,속성,속성설명)	클래스(이름)	클래스도 클래스(개요,속성,속성설명)
UI흐름정의서		UI흐름도,화면명	UI흐름도,화면명	UI흐름도,화면명
화면정의서	유스케이스모형기술서 UI 프로토타입	화면ID,화면명,설명,양식 화면항목설명(항목명,속성,I/O,길이,관련데이터명)	화면ID,화면명, 화면항목설명(항목명)	화면설명,양식 화면항목설명(속성,I/O,길이,관련데이터명)
보고서정의서		보고서ID,보고서명,형식,사용자,설명,양식,보고서항목설명(항목명,속성,길이,관련데이터명)	보고서ID,보고서명,사용자,보고서항목설명(항목명)	보고서형식,설명,양식,보고서항목설명(속성,길이,관련데이터명)
소프트웨어 아키텍처정의서	비전기술서 요구사항기술서 유스케이스 모형기술서 객체모형기술서 시스템아키텍처정의서(초기)	기능요구사항,품질요구사항,아키텍처요구사항,제약사항,아키텍처원형,연관관계,시스템 인스턴스 모형	기능요구사항,품질요구사항,아키텍처요구사항,제약사항,	아키텍처 원형,연관관계,시스템 인스턴스 모형
소프트웨어 아키텍처평가서	요구사항기술서 유스케이스모형기술서 소프트웨어아키텍처정의서	평가품질,요구수준정의,품질프로파일목록,대안아키텍처정의,아키텍처평가결과,최종 아키텍처 정의		
메커니즘 기술서	요구사항기술서 유스케이스모형기술서 소프트웨어아키텍처정의서	소프트웨어아키텍처,지속성구현메커니즘,가변성구현메커니즘,기타메커니즘	소프트웨어아키텍처,지속성구현메커니즘,가변성구현메커니즘,	기타메커니즘
소프트웨어 아키텍처기술서	시스템환경정의서 소프트웨어아키텍처정의서 소프트웨어아키텍처평가서	소프트웨어아키텍처,클래스	소프트웨어아키텍처, 클래스	
데이터모형설계서	객체모형기술서 메커니즘기술서	ER(엔터티,속성)	ER(엔터티,속성)	
컴포넌트명세서	유스케이스모형기술서 객체모형기술서 소프트웨어아키텍처정의서	컴포넌트	컴포넌트	
시스템아키텍처 정의서	요구사항기술서 시스템아키텍처정의서(초기) 소프트웨어아키텍처정의서 메커니즘기술서	유스케이스,비즈니스컴포넌트아키텍처,소프트웨어 아키텍처, 병행구조, 분산구조,배치구조,아키텍처 제약사항	유스케이스,비즈니스컴포넌트아키텍처,소프트웨어 아키텍처,아키텍처 제약사항	병행구조, 분산구조,배치구조
아키텍처 프로토타입핑 수행계획서	유스케이스모형기술서 시스템아키텍처정의서	컴포넌트	컴포넌트	
아키텍처 프로토타입설계서	유스케이스모형기술서 시스템아키텍처정의서	컴포넌트	컴포넌트	
아키텍처 프로토타입	시스템아키텍처정의서 아키텍처프로토타입핑 수행 계획서	컴포넌트	컴포넌트	
테스트 수행보고서	UI설계서 - UI흐름정의서 - 화면정의서	테스트수행과정 테스트수행결과(테스트케이스ID,컴포넌트명,타입,값),평가(테스트유형,특성,척도,통과기준,통과율)	테스트수행결과(테스트케이스ID,컴포넌트명),평가(테스트유형,척도,통과기준,통과율)	테스트수행과정 테스트수행결과(타입,값),평가(특성)
검토의견서	테스트수행보고서 시스템아키텍처정의서 아키텍처 프로토타입설계서 아키텍처프로토타입	검토대상,검토자,작성자,산출물,작성일,검토의견,근거/관련내용,해결방안	검토대상,검토자,작성 자,산출물,검토의견,해 결방안	작성일,근거/관련
아키텍처 전개계획서	프로젝트수행계획서 소프트웨어아키텍처기술서 시스템아키텍처정의서	교육명,교육목적,교육대상자,교육자료,교육기자재,교육장소,교육일정	교육명,교육목적,교육 대상자,교육자료	교육기자재,교육장소,교육일정
아키텍처 전개보고서	아키텍처전개계획서 소프트웨어아키텍처기술서 시스템아키텍처정의서	교육명,교육장소,교육일시,강사,교육대상자(순번,수강자,부서명,직책,서명,평가),문제점,해결방안	교육명,교육장소,교육 일시,강사	교육명,교육장소,교육일시,강사
테스트계획서	프로젝트수행계획서 유스케이스모형기술서 시스템아키텍처정의서	테스트배경,테스트범위,테스트항목,특성,척도,통과기준,테스트케이스유형(테스트레벨,항목,특성,입력테스터유형,출력테스터유형),테스트방법(항목,특성,테스트케이스선정기준),테스트예외사항,테스트환경	테스트항목,특성,척도, 통과기준,테스트케이스 유형(항목),테스트방법 (항목)	테스트배경,테스트범위,테스트케이스유형(테스트레벨,특성,입력테스터유형,출력테스터유형),테스트방법(특성,테스트케이스선정기준),테스트예외사항,테스트환경
시스템전환 계획서	프로젝트수행계획서 유스케이스모형기술서	전환단계,전환기간,대상프로그램,대상데이터베이스,전환일시,설치장소,시설/장비,접속시스템명,전환절차,비상대책방안	전환단계, 대상프로그램, 대상 데이터베이스	전환기간,전환일시,설치장소,시설/장비, 접속시스템명,전환절차,비상대책방안

스트계획서에 유스케이스입력항목이 없는데, 추적을 위해 추가할 것을 제안하고, 부가적 연구결과로 6장에 설명되어 있다.

3.1.3 요구획득단계 산출물과 아키텍처단계 산출물간의 추적

선행연구의 결과와 3.1.2 의 (그림 4), (그림 5) 아키텍처 단계 산출물간의 추적성을 결합하여 두 단계 산출물간의 요구사항을 추적해 보면 (그림 6)과 같다.

요구획득단계의 요구사항정의서에서 최초 공식적으로 명세된 요구사항은 유스케이스로 추적되고, 다시 화면ID로 추적적이 이어진다. 화면이 정의된 다음 아키텍처단계의 소프트웨어아키텍처는 요구사항, 유스케이스를 토대로 아키텍처요구사항을 식별하고 레이어, 패턴, 클래스를 정의한다. 정의된 레이어와 패턴에 따라서 식별된 비즈니스컴포넌트는 다시 유스케이스와 추적되며, 테스트계획서까지 추적이 이어진다. 테스트계획서는 요구사항의 공식적 승인문서인 범위기술서(인수기준정의서)와 최종 추적이 되면서 요구사항의 모든 범위에 대해 테스트계획을 수립하였는지에 대해 일관성과



(그림 6) 요구획득단계 산출물과 아키텍처단계 산출물간의 추적

완전성을 추적하게 된다.

하드웨어, 네트워크와 같은 기술아키텍처는 비기능요구사항으로서 소프트웨어아키텍처정의서의 품질요구사항, 제약사항과 시스템아키텍처정의서의 아키텍처제약사항과 함께 품질관리계획서의 품질속성에 따랐는지 추적이 되어야 하며, 테스트계획서에서도 품질속성에 따른 테스트항목이 있는지 추적되어야 한다.

3.2 요구사항추적테이블의 확장

3.2.1 기능요구사항추적테이블

요구사항은 기능요구사항과 비기능요구사항으로 구분하여 관리되는데, 이에 요구사항추적테이블은 기능요구사항인지 비기능요구사항인지에 따라 추적필드 구성에 차이가 있다.

우선 (그림 7)은 기능요구사항추적테이블이다.

요구획득단계 산출물에 이어서 아키텍처단계 산출물까지 추적된 (그림 7)의 기능요구사항추적테이블은 이미 추적된 요구사항과 유스케이스에 이어서 UI설계서의 화면, 소프트웨어아키텍처정의서의 아키텍처요구사항, 소프트웨어아키텍처기술서의 레이어, 패턴, 컴포넌트명세서의 컴포넌트, 데이터모형기술서에서 엔터티, 테스트계획서에서 테스트ID 순으로 추적필드를 정하고 순차적으로 추적한다. 아울러, 재사용여부 추적필드는 재사용컴포넌트명세서에서 추출된 필드로, 컴포넌트 중에 재사용이 가능한 것으로 명세된 컴포넌트를 “Y” 또는 “N”으로 나타낸다. 추적필드 선정은 3.1.3 의 (그림 6)에서 순방향으로 추적된 필드에 근거한다.

변경요구사항필드는 기준선이 정해진 다음에 변경된 요구사항에 대해 변경코드, 변경내용, 변경날짜를 기재하는 필드로, 변경관리프로세스에 따라 처리된 변경코드를 링크하여 추적할 수 있게 하는 것이 목적이다. (그림 7)에서 ‘기존 연구’로 표시된 추적필드는 선행연구인 김주영[2]의 연구 결과이다.

프로젝트준비 프로젝트시작				요구획득단계			아키텍처설계단계														
요구 사항 ID	프로젝트 기안서 프로젝트 정의서	프로젝트 제안서	프로젝트 수행계획서	요구 사항	관련부서	구현방안	변경요구 사항	유스 케이스	클래스	화면 / 보고서	아키텍처 요구 사항	레이어	패턴	컴포넌트	재사용 여부	엔터티	테스트 ID	우선 순위	범위 여부	인수 기준 ID	
																					R

(그림 7) 기능요구사항추적테이블

3.2.2 비기능요구사항추적테이블

(그림 8)의 비기능요구사항추적테이블은 기능요구사항추적테이블에 없었던 품질관리계획서의 사전 요구사항을 시작하여 요구획득단계의 요구사항필드에 이어서 아키텍처단계에서 나온 품질요구사항과 아키텍처제약사항, 테스트ID로 추적한다.

3.2.3 확장된 요구사항추적테이블 작성규칙

본 연구의 요구사항추적테이블은 다음과 같은 규칙을 전제하여 작성하여야 한다.

(규칙1) 범위기술서의 인수기준ID는 요구사항ID와 매핑되어야 한다. 이때

- 인수기준ID는 기능요구사항ID와 비기능요구사항ID를 포함한다
- 모든 요구사항ID는 최소한 1개 이상의 인수기준ID에 포함된다.
- 인수기준ID에 포함되지 않은 기능요구사항ID는 범위 여부에 OUT으로 기재하되, 삭제하지 않는다.

(규칙2) 유스케이스ID는 컴포넌트ID와 매핑되어야 한다. 이때

- 모든 유스케이스ID는 최소한 1개 이상의 컴포넌트에 포함된다.

(규칙3) 모든 컴포넌트ID는 하나의 레이어에 나타나야 한다

(규칙4) 테스트ID는 테스트계획ID와 결과ID를 포함하며,

이 두 ID는 매핑되어야 한다. 이때

- 모든 테스트계획ID는 동일한 이름의 테스트결과ID를 가진다.
- 테스트ID는 1개 이상의 컴포넌트ID를 포함한다.

(규칙5) 테스트ID는 인수기준ID와 매핑되어야 한다. 이때

- 모든 테스트ID는 1개 이상의 인수기준ID에 포함된다.

(규칙6) 요구사항 변경이 발생했을 경우 기존에 기재된 내용을 지우지 않은 상태에서 변경요구사항필드에 새로 변경된 내용을 추가하여 작성하고 변경날짜를 함께 기재한다. 만약 변경관리프로세스에 의해 변경코드(변경ID)가 있는 경우는 변경코드를 같이 기재하여 변경관리프로세스와 링크가 되게 한다.

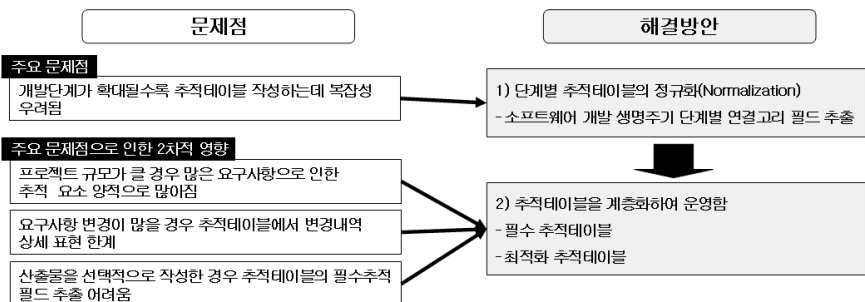
4. 요구사항추적테이블의 정규화

4.1 추적테이블 확장시 문제점

한편, 확장된 요구사항추적테이블은 개발프로세스가 확대될수록 추적필드가 많아져서 추적테이블을 복잡하게 할 우려가 있다. 이렇게 복잡해졌을 때 요구사항이 너무 많으면 추적작업에 노력이 많이 들며, 요구사항 변경 내역도 많아져 표현에 어려움이 발생한다. 이를 해결하는 방안으로 (그림 9)와 같이 추적테이블을 개발프로세스단위로 나누거나 합칠 수 있는 정규화 방안을 제시한다.



(그림 8) 비기능요구사항추적테이블



(그림 9) 추적테이블 작성시 문제점 및 해결방안

4.2 추적테이블을 개발프로세스단위로 정규화

정규화(Normalization)는 관계형데이터베이스에서 테이블 하나하나를 주 키에서 직접 연상되는 데이터만으로 구성되도록 설계하는 것이다[8]. 이에 추적테이블 정규화는 추적테이블을 개발프로세스단위로 키를설정하여 나누는 것이다. 정규화하는 방법은 먼저 (그림 10)에서처럼 개발프로세스단위로 키가 되는 추적필드를 선정하고, 선정된 키필드간의 매핑을 정의한다.

- 키 필드는 3.2.3 에서 정의한 추적테이블의 작성규칙 중에서 (규칙1) 범위기술서의 인수기준ID는 요구사항ID와 매핑되어야 한다.
- (규칙2) 유스케이스ID는 컴포넌트ID와 매핑되어야 한다.
- (규칙4) 테스트ID는 테스트계획ID와 결과ID를 포함하며, 두 ID는 매핑되어야 한다.
- (규칙5) 테스트ID는 인수기준ID와 매핑되어야 한다.

등의 4가지 규칙에 근거하여, (그림 10)에서 보는 바와 같이 프로젝트시작단계에서는 인수기준ID, 요구획득단계에서는 요구사항ID와 유스케이스ID, 아키텍처단계에서는 컴포넌트ID와 테스트ID, 인도단계에서는 프로젝트단계에서와 동일한 인수기준ID를 선정하였다.

그런 다음 (그림 11)과 같이 개발프로세스별로 분리된 추적테이블마다 선행단계의 키필드를추적테이블 맨 왼쪽에 추가하면 개발프로세스단위로 정규화된 추적테이블이 생성된다.

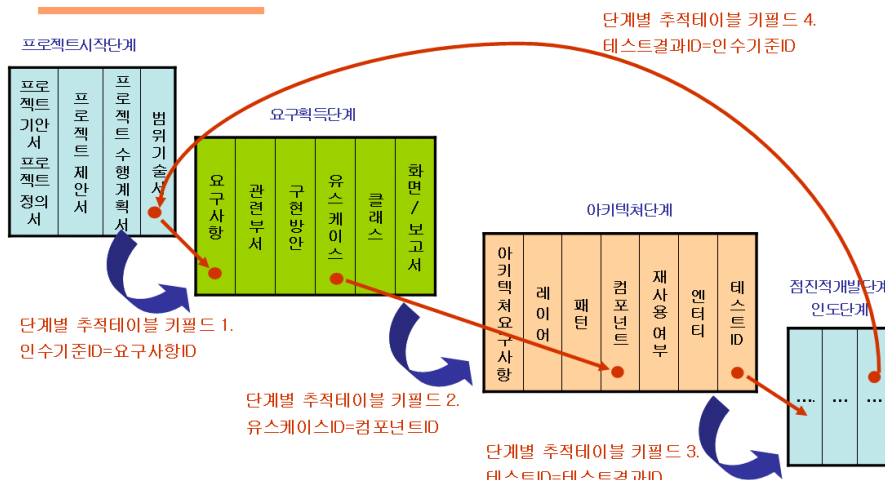
이 추적테이블 정규화를 통해 추적테이블이 확대되고 복잡해졌을 경우 추적테이블을 개발프로세스단위로 나누어서 작성하였다가, 역으로 필요시점에서 개발프로세스단위로 나누어진 추적테이블을 키필드간의 매핑으로 통합할 수 있다.

4.3 추적테이블의 계층화

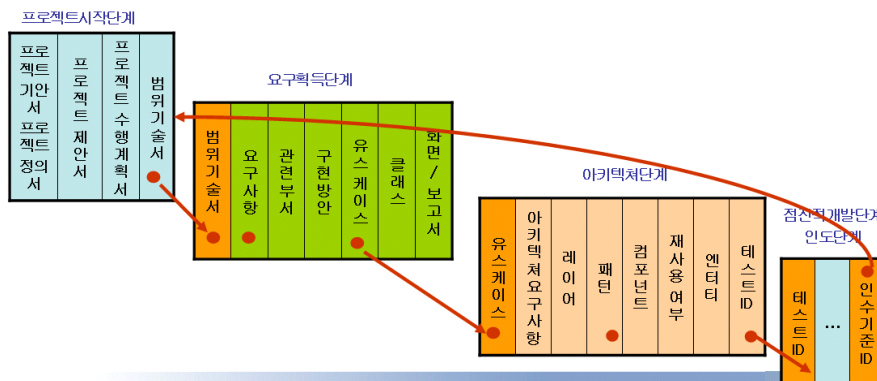
추적테이블 확장시 문제점을 해결하는 두번째 방안은 추적테이블의 정규화를 이용하여 추적테이블을 계층화하는 방법이다. 계층화 방법에는 <표 2>와 같이 추적테이블 키필드 중심의 필수추적필드로만 구성된 필수추적테이블을 작성하는 방법과 문제점에 따라 추적필드를 묶거나 추가하여 문제점 해결에 최적화된 추적테이블을 작성하는 방법이 있다.

먼저, 개발방법론에서 정의한 산출물을 모두 작성하지 않고 선택적으로 산출물을 작성한 경우 추적테이블을 작성시 추적필드 선정이 본 연구와 같이 되지 않을 것이다. 이럴 경우에는 (그림 12)에서처럼 키 필드만 구성된 필수추적테이블을 활용하면 추적테이블 작성이 가능하다. 단, 선택한 산출물에는 키필드가 포함된 산출물이 포함되었음을 전제한다.

한편 요구사항이 많은 경우 추적량이 많아 추적테이블 작



(그림 10) 개발프로세스별 추적테이블 키필드 선정



(그림 11) 개발프로세스별 추적테이블에 키필드 추가

성에만 과도한 노력이 들 수 있으므로 (그림 13)과 같이 요구사항을 묶어서 그룹화한 최적화테이블의 예이며, (그림 14)는 요구사항에 변경이 많은 경우 추적테이블내에서 프로젝트 전체의 변경관리현황문서와 매핑하여 요구사항변경내역을 변경관리자가 즉시적으로 볼 수 있게 하는 최적화테이블의 예이다.

5. 연구에 대한 검증

5.1 D사의 H시스템개발프로젝트 적용 사례

5.1.1 요구사항추적테이블의 적용 사례

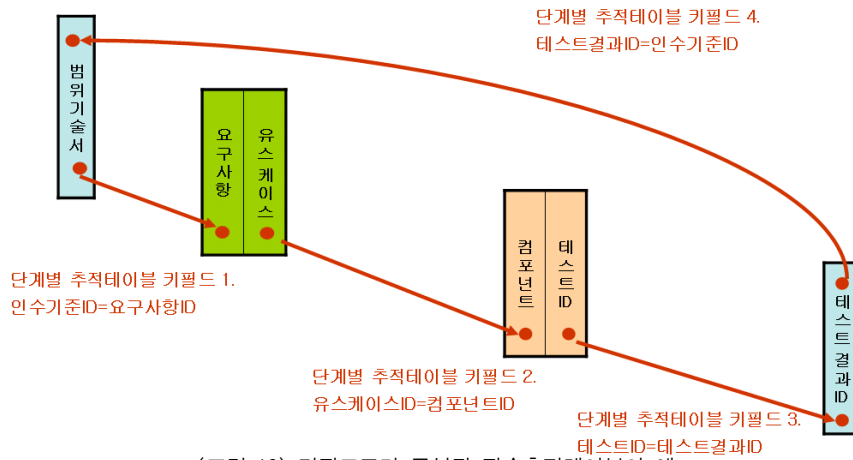
(그림 15)는 기능요구사항추적테이블을 실제 프로젝트에

적용한 사례이다. 사례 프로젝트는 D사의 H시스템(홈네트워크관리시스템)을 구축한 사례로, 대상시스템은 아파트 입주민 및 관리자를 위한 각종 주민편의서비스를 제공하고 아파트단지내 홈네트워크 관련 시스템을 감시하는 기능을 제공하는 웹기반의 시스템이다.

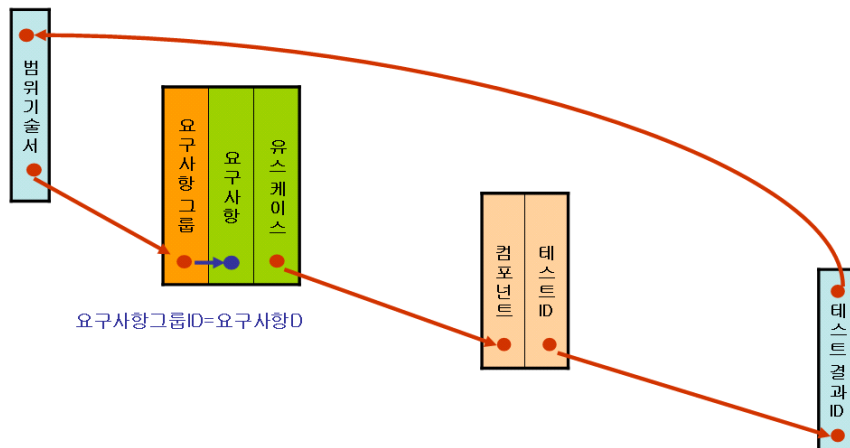
사례에서는 프로젝트 시작단계 산출물인 제안서에서부터 인도단계 산출물인 인수기준정의서(인수기준ID)까지를 추적하였는데 프로젝트기안서 대신 건설산업에 맞게 시방서 산출물을 사용하였다. 또한 사례 프로젝트에서는 요구사항변경이 잦아서 추적테이블에 변경요구사항 필드를 추가하였고, 변경관리코드를 기재한 후, 변경관리현황 문서와 매핑되도록 하였는데, 연구에서의 최적화추적테이블을 인용하였다. 한편, 3.2.3의 추적테이블 작성규칙에 따라 인수기준에 포함

<표 2> 계층화된 추적테이블의 종류 및 용도

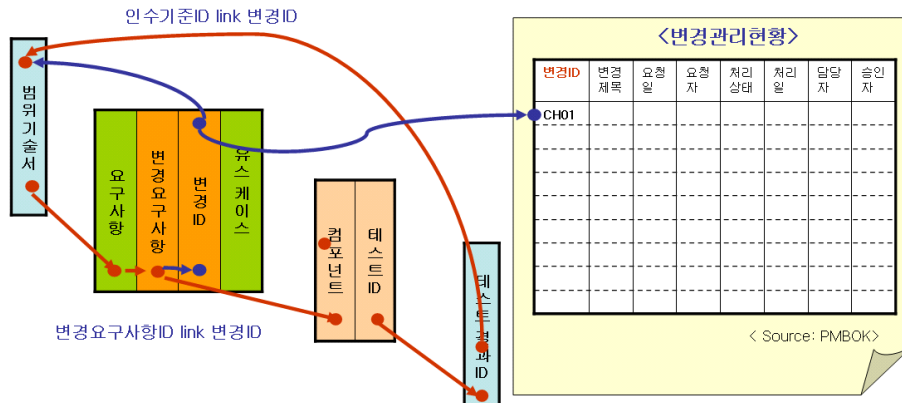
계층화 종류	정의	용도	해결요소
필수추적테이블	추적필드 중 키필드에 근거한 필수 추적필드로 구성된 추적테이블	방법론에서 정한 산출물을 선택적으로 작성할 경우, 요구사항 추적테이블에 필수적으로 필요한 추적필드만 구성하여 최소한의 추적이 이루어지게 함	선택적 산출물에도 요구사항 추적방법 제시 선택적 산출물 정의할시, 필수 산출물 가이드 제시
최적화추적테이블	추적필드에서 분류필드를 추가하거나 키필드에 근거한 추적필드간, 다른 산출간의 링크필드를 구성한 추적테이블	예1)요구사항이 많을 경우 요구사항을 그룹을 지어 추적할 대상 요구사항의 양을 줄여줌 예2)변경이 많을 경우 변경관리현황문서와 링크 변경내역을 알기 쉽게 함	추적대상의 간소화 추적의 가독성



(그림 12) 키필드로만 구성된 필수추적테이블의 예



(그림 13) 요구사항을 그룹화한 최적화추적테이블의 예

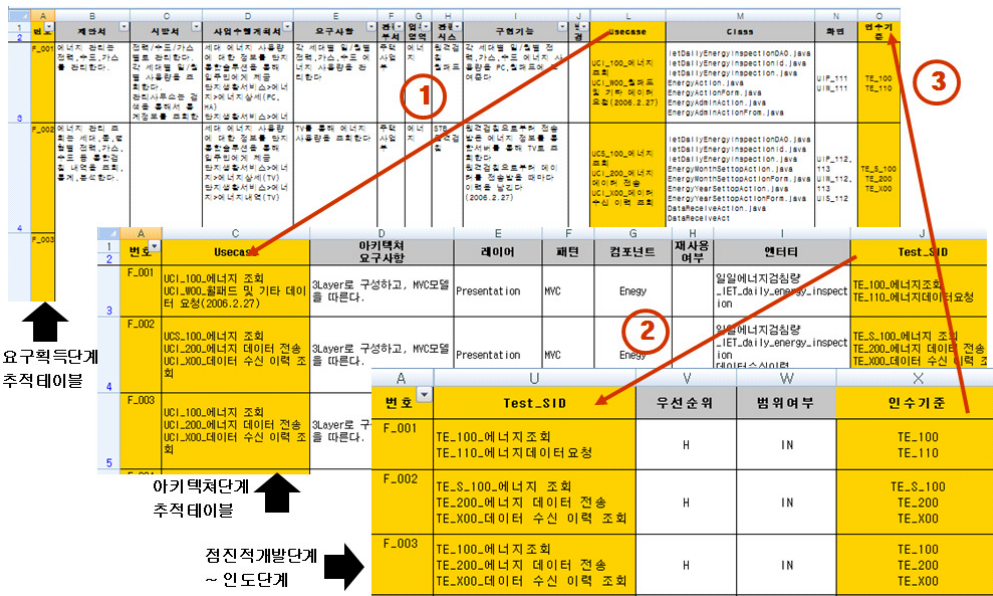


(그림 14) 변경관리현황과 연계한 최적화추적테이블의 예

번호	제안서	시방서	사업수행계획서	요구사항	관리부서	업무영역	관련시스템	구현기능	변경요구사항	Usecase
F-017		검색을 통해서 통계정보를 조회할 수 있어야 한다.	단지내 생활서비스>관리비관리>월별관리비조회	검색을 통해서 통계정보를 조회할 수 있어야 한다.	주력사업부	관리비	관리비 SW	관리사무소에서 PC로 각 세대별 관리비를 동/동형, 월/월연간 추이, 적남대비, 전월대비 등의 통계정보를 조회한다	동별 검색은 의미가 없으므로 제외한다 (2006.2.20) =>CH009	UC1_300.관리비 조회
F-018	CCTV 시스템은 실시간 동영상 데이터(주요 장소 원격감시, 카메라 상면현황, 보관 데이터 조회), DVR 하드 공간 관리를 연동한다.	영상 서비스 접속정보 연동	단지내 주요시설(블루터, 단지앞 등)에 설치된 CCTV 영상 서비스를 TV와 PC로 입주인에게 온라인(실시간)으로 제공한다.	단지내 주요 장소에 대한 실시간 동영상 데이터를 영상서버로 제공한다.	통신사업부	단지영상	CCTV	영상의 접속정보를 통합서버를 통해 받아서 PC에서 조회한다		UC1_500.CCTV 영상 조회 UC1_600.CCTV 영상 관리
F-019				DVR 서버의 보관 영상 데이터를 연동할 수 있어야 한다.	영상정보		CCTV	기술검증후 재검토할 관리자는 DVR시스템에서 직접 작업하고, 통합은 구현하지 않는다	DVR 자체기능으로 사용한다 =>CH014	
F-020				DVR 하드 공간에 대한 정보를 연동할 수 있어야 한다.	영상정보		CCTV	기술검증후 재검토할 관리자는 DVR시스템에서 직접 작업하고, 통합은 구현하지 않는다		
F-021		CCTV 시스템 상태/알람정보 등		CCTV 카메라 상면 현황 정보를 연동할 수 있어야 한다.	공공시스템상	상태 감시	CCTV	기술검증후 재검토할 관리자는 DVR시스템에서 직접 작업하고, 통합은 구현하지 않는다		
F-022			시스템관리>공공시스템관리>공공상태감시	CCTV 시스템 서버상태/알람정보를 연동할 수 있어야 한다.	통신사업부	공공시스템관리	CCTV	공공시스템 서버상태관리 기능에 통합되어 조회한다		UC1_000.공공시스템 감시 UC1_300.공공시스템 알람 조회
F-023	출입통제/DVR 조하는 입주인 카드 관리, 카드변조/비밀번호 관리, DVR 영상 조회를 한다.		입주인에게 CCTV 영상을 TV로 통해 볼 수 있도록 서비스	TV를 통하여 단지내 주요 장소를 실시간으로 조회할 수 있어야 한다.	통신사업부	단지영상	CCTV STB	TV에서 단지 내/외부의 정해진 CCTV의 영상을 받아 보여준다 (2006.9.30)		UCS_800.CCTV 영상 조회
F-024			단지내 주요시설(블루터, 단지앞 등)에	인터넷을 통하여 단지내 주요 시설을 실시간으로 조회할 수 있어야 한다.	통신사업부	단지영상	CCTV	영상의 접속정보를 통합서버를 통해 받아서 PC에서		UC1_500.CCTV

Usecase	Class	화면	아키텍처 요구사항	레이어	패턴	컴포넌트	재사용 여부	엔티티	Test_SID	우선순위	범위 여부	인수기준
UC1_300.관리비 조회	letRateItemDAO.java letRateDAO.java letRateItemId.java letRateId.java letRateItem.java letRate.java RateAdminAction.java RateAdminActionForm.java	UIP_129	3layer로 구성하고, MVC모듈을 따른다.	Presentation	MVC	Rate		관리금항목 _let_rate_item 관리금_let_rate	TE_300.관리비 조회	H	IN	TE_300
UC1_500.CCTV 영상 조회 UC1_600.CCTV 영상 관리	IFTCctvCameraDAO.java IFTCctvCamera.java CctvCameraAction.java CctvCameraAdminActionForm.java	UIP_131	3layer로 구성하고, MVC모듈을 따른다.	Presentation	MVC	CCTV		영상카메라 _IFT_cctv_camera	TE_500.CCTV 영상 조회 TE_600.CCTV 영상 관리	H	IN	TE_500 TE_600
											OUT	
											OUT	
											OUT	
UC1_000.공공시스템 감시 UC1_300.공공시스템 알람 조회	listDisplayToIDDAO.java listDisplayToID.java listCounterData.java CounterDataAction.java listCounterData.java CounterDataActionForm.java PublicSystemAlarmAction.java PublicSystemAlarmActionForm.java	UIP_132	3layer로 구성하고, MVC모듈을 따른다.	Presentation	MVC	Memnt		공공시스템감시_항목 1_list_display_to_ID 공공시스템감시_항목 2_list_counter_detail 공공시스템감시_로그 _list_counter_data	TE_000.공공시 스템 감시 TE_300.공공시 스템 알람	M	IN	TE_000 TE_300
UCS_800.CCTV 영상 조회	CctvCameraSettopAction.java CctvCameraSettopActionForm.java	UIP_133 UIW_133	3layer로 구성하고, MVC모듈을 따른다.	Presentation	MVC	CCTV		영상카메라 _IFT_cctv_camera	TE_S_800.CCTV	H	IN	TE_S_600
UC1_500.CCTV	IFTCctvCameraDAO.java IFTCctvCamera.java								TE_500.CCTV			

(그림 15) 기능요구사항추적테이블 사례



(그림 16) 추적테이블의 정규화 사례

되지 않게 된 요구사항은 범위여부에서 ‘OUT’ 으로 표시하고 삭제하지 않음으로써 이력이 남도록 하였다.

사례를 통해 본 연구의 적용가능성과 요구사항 추적의 효과를 보고자 하였는데, 우선 전체 개발프로세스 산출물인 연구의 추적테이블에서 추적이 가능했으며, 작성된 추적테이블을 통해 단계를 지날때마다 산출물이 진화하면서 모든 요구사항이 산출물마다 존재하는지의 완전성과 요구사항의 의미하는 바를 제대로 산출물에 나타내고 있는지의 일관성을 검증할 수 있었다. 또한 요구사항이 추적되는 과정을 한눈에 알 수 있었고, 이를 통해 오류를 보다 쉽게 발견함으로써 요구사항의 일관성을 완전성을 보다 정확하게 검증할 수 있었다.

5.1.2 추적테이블 정규화 사례

(그림 16)은 (그림 15)의 기능요구사항추적테이블을 프로젝트시작단계-요구획득단계, 아키텍처단계, 점진적개발단계-인도단계로 나눠 단계별로 추적테이블의 키워드를 중복하여 작성한 정규화된 추적테이블의 사례이다.

(그림 16)에서 보면 ①요구획득단계 추적테이블의 추적필드 중 Usecase필드가 아키텍처단계 추적테이블의 Usecase 필드와 매핑되며, ②아키텍처단계 추적테이블의 Test_SID 필드가 점진적개발단계-인도단계 추적테이블의 Test_SID 필드와 매핑된다. 마지막으로, ③점진적개발단계-인도단계 추적테이블의 인수기준필드는 요구획득단계 추적테이블의 추적필드의 인수기준필드와 매핑된다. 사례 프로젝트에서는 이렇게 추적테이블을 정규화하여 사용함으로써 단계별로 추적필드가 나뉘졌기 때문에 복잡하지 않게 추적활동이 진행되어졌으며, 최종단계에서는 (그림 15)와 같이 통합하여 추적결과를 한눈에 볼 수 있었다.

아울러 본 사례에서는 최적화된 추적테이블을 활용함으로써 요구사항변경내역을 쉽게 파악할 수 있었고 변경관리현

황 문서 작성시 변경영향분석을 용이하게 하였다.

이에 사례 프로젝트는 추적테이블을 통해 요구사항관리 미흡에서 오는 소프트웨어 개발 실패 원인을 최소화할 수 있었다.

5.2 기존 추적연구와의 비교

확장된 요구사항추적테이블 및 정규화 방안을 제시한 본 연구와 앞서 2장에서 열거한 기존의 추적연구들을 추적사상, 추적기능구성, 정성적 측면 등을 중심으로 <표 3>에서 비교하여 본 연구의 장점과 한계점을 보고자 한다. 우선 비교항목에서 추적기능구성, 정성적 측면의 세부항목은 INCOSE와 Ralph H. Acaba 각각에서 제시한 요구관리도구 선정기준 중에서 추적기능에 관한 기준을 추출하여 선정하였고, 추적사상의 세부항목은 본 연구의 배경으로써 연구결과를 검증하는 항목이다[9][10].

비교항목 중 정성적 측면은 비기능적 요소로 경험적 판단이 포함되기 때문에 비교의 객관성을 위해 정보시스템 분석 및 설계 경력 7년이상(중급기술자)의 전문가 5인을 대상으로 설문을 진행한 결과이다.

<표 3>에서 보면 상용도구와 Lee M. Williams 의 요구정보관리시스템이 추적기능구성면에서 많은 잇점이 있으나 추적필드나 추적링크를 필요에 따라 조절하는 유연성이 미흡하며, 추적모델을 제공하는 안상임의 추적메타모델 및 Ramesh의 참조모델은 추적링크를 생성하는 방법에 대한 훈련이 필요하기 때문에 사용편의성이 부족하다. 한편 본 연구의 추적테이블은 추적필드 유연성 및 사용편의성을 제공하는 점에서 잇점이나 추적모델을 제시하지 않기에 연구의 한계가 있다.

비교결과를 볼 때 특정 방법론을 인용하여 추적필드를 구체적으로 제시한 본 연구와 추적기능구성이 유리한 상용도구를 함께 사용하는 것도 가능할 것이다.

〈표 3〉 본 연구와 선행 및 관련연구들과의 비교

비교항목	본 연구의 추적 테이블	추적기법/추적테이블				사용 도구
		김주영의 요구사항추적테이블	안상인의 요구사항 추적 및 변경 영향 분석	Ramesh의 Reference Models	Lee M. Williams의 요구정보관리시스템	
추적 사상						
전체 공정 추적	O	X	O	O	O	O
추적모델 제시	X	X	O	O	△	△
추적 기능 구성						
추적링크 or 필드유연성	O	X	O	X	△	X
변경 추적	O	△	O	△	O	O
요구사항 그룹핑	O	X	△	X	X	O
위험 영향 식별	X	X	X	O	O	△
시험가능성 표시	O	O	O	O	O	O
정성적 측면						
사용 편의성	O	O	X	X	X	X
추적링크 가독성	O	△	O	O	△	△

〈표 4〉 마르미-III v.4.0 산출물의 보완점

구분	산출물명	연구 제안	목적
산출물 보완	UI설계서 (UI흐름서 화면정의서 보고서정의서)	상위산출물과 추적되는 항목 없음. 요구사항 추적을 위해 유스케이스ID 항목을 추가함	선행 산출물과의 일관성 검증
	컴포넌트명세서	상위산출물과 추적되는 항목 없음. 요구사항 추적을 위해 유스케이스ID 항목을 추가함	선행 산출물과의 일관성 검증
	제사용컴포넌트 명세서	상위산출물과 추적되는 항목 없음. 요구사항 추적을 위해 유스케이스ID 항목을 추가함	선행 산출물과의 일관성 검증
	테스트계획서	상위산출물과 추적되는 항목 없음. 요구사항 추적을 위해 유스케이스ID 항목을 추가함	선행 산출물과의 일관성 검증

출물을 대상으로 하였다. 연구의 제안대로 요구사항추적테이블을 작성하기 위해서는 산출물에 일부 보완이 필요함을 본문의 연구진행과정에서 부분부분 기술하였는데, <표 4>에서 이를 취합하여 정리하였다.

6.2 결론 및 향후 연구 계획

본 연구는 일부 개발프로세스가 아닌 전체 개발프로세스의 추적대상이 되는 산출물을 추적테이블에 작성하여 요구사항의 일관성과 완전성을 검증하게 하였고, 아울러 사용상 편의와 추적의 가독성을 제공하여 추적의 정확성을 높였다. 또한 추적테이블의 정규화 방안을 통해 추적테이블 작성시 초래되는 추적테이블의 복잡성을 해결하여 요구사항추적테이블의 활용성을 제고하고자 하였다. 이런 요구사항 추적활동을 통해 소프트웨어 개발의 실패를 최소화할 것이다. 한편, 비교 결과에서도 밝혔듯이 본 연구는 마르미 방법론을 인용하여 구체적인 추적필드를 제시한 바, 추적기능구성면에서 우수한 상용도구 등과 결합하여 사용하는 것도 가능할 것이다.

한편, 아키텍처단계 산출물을 대상으로 추적필드를 추출하는 과정에서 패턴, 레이어 등은 기능필드인지 비기능필드인지를 정할 때 애매한 요소가 많았으며, 기능/비기능 모두에 속하기도 하여 정확한 구분이 어려웠다. 이에 본 연구에서는 대체로 기능요소가 많은 것으로 판단하고 기능요구사항추적테이블의 추적필드로 정의하였는데 향후 연구에서 이를 좀더 면밀히 연구하여 그 기준을 정하고, 정확하게 분류하고자 한다.

본 연구는 마르미 방법론을 인용하여 추적테이블 작성규칙을 정하고, 정규화 방안 등을 제시하였는데 이를 일반화시킬 수 있는 연구결과를 위해 이론 및 연구 검증기법을 도입, 이론화하는 연구를 진행해 보고자 한다.

또한, 내용적인 면에서 변경관리프로세스와 일부 연결하려는 일부 노력을 시도해 보았는데, 이를 좀더 확대하여 추적테이블을 이용하여 소프트웨어 개발산출물과 관리산출물

범례.

- 1) 추적사상/추적기능구성 항목 : O (전적으로 지원함), △ (일부 지원함, 인정의 여지가 있음), X (거의 지원하지 않음)
- 2) 정성적측면 항목 : 전문가5인의 3점척도 설문조사 결과, 그 합에 따라 O (15-14점), △(13점-7점), X (5점-6점), 설문조사 상세결과는 부록 참조

6. 결 론

6.1 부가적 연구결과로 마르미-III v.4.0 의 산출물의 보완점 제시

본 연구는 서론에서 밝힌 바와 같이 마르미-III v.4.0 산

을 통합관리하는 프레임워크를 연구하고자 한다.

참 고 문 헌

[1] P. Hsia, A. Davis, D. Kung, "Status report : Requirement Engineering," IEEE Software, 1993.
 [2] Balasubramaniam Ramesh, "Toward Reference Models for Requirements Traceability," IEEE Transactions on software Engineering, Vol.27, No.1, 2001.1.
 [3] 안상임, 가치분석을 통한 휘저기반의 요구사항 추적 및 변경 영향 분석, 숭실대학교 일반대학원 박사학위논문, 2007.12.
 [4] 김주영, 류성열, "산출물간의 일관성과 완전성 검증을 위한 추적데이터의 경험적 연구", 한국정보과학회 논문지, 제34권 제5호, pp.419-430, 2007.05.
 [5] Karl E.Wiegers, Automating Requirements Management, Process Impact, 1999.
 [6] Lee M. Williams, The Elements of Technical Requirements Management, INCOSE Atlanta Chapter, 2006.02.
 [7] 한국전자통신연구원, 마르미-III 버전 4.0, 2002.
 [8] 국가지식정보통합검색시스템, <http://www.knowledge.go.kr>
 [9] INCOSE, INCOSE Requirements Tool Evaluation Criteria, Integrated Chipware, 1997.
 [10] Ralph H. Acaba, "Lessons Learned in the Selection of a Company Standard RM Tools," Quality System&Software, 2000.



김 주 영

e-mail : gogumacake@ssu.ac.kr
 1995년 동아대학교 컴퓨터공학과(학사)
 2004년 숭실대학교 소프트웨어공학과(공학 석사)
 2007년 숭실대학교 컴퓨터학과 박사수료
 관심분야: 요구공학, 소프트웨어 개발방법론



류 성 열

e-mail : syrheew@ssu.ac.kr
 1976년 숭실대학교 전자계산학과(학사)
 1980년 연세대학교 전자계산학과(공학석사)
 1997년 아주대학교 컴퓨터공학과(공학박사)
 1982년~1995년 숭실대학교 전자계산연구소 및 중앙전자계산소 소장
 1992년~1993년 한국정보과학회 총무 이사
 1997년~1998년 George Mason University 객원 교수
 1998년~2001년 숭실대학교 정보과학대학원 원장
 1998년~현 재 숭실대학교 소프트웨어 공학센터 소장
 2006년~현 재 공정거래위원회 성과관리위원회 위원
 2006년~현 재 기획예산처 정보화예산협의회 위원
 2007년~현 재 사회복지정보화추진 위원회 위원
 1981년~현 재 숭실대학교 컴퓨터학부 교수
 관심분야: 소프트웨어 유지보수, 소프트웨어 재사용, 소프트웨어 재공학/역공학, 오픈소스 소프트웨어

부 록

- 설문자 구성
 - A: 고급기술자, 소프트웨어개발프로젝트 10년, 현 한국지역정보개발원 근무
 - B: 중급기술자, 소프트웨어개발프로젝트 7년, 현 대림대학 전산실 근무
 - C: 고급기술자, 소프트웨어개발프로젝트 8년, 현 T-Free 근무
 - D: 중급기술자, 소프트웨어개발프로젝트 7년, 현 대학교 강사 근무
 - E: 고급기술자, 소프트웨어개발프로젝트 11년, 현 우리은행 전산실 근무
- 비교항목 중 정성적 측면에 대한 전문가 5인의 설문조사 3점척도 결과
 - 3점: 전적으로 지원함
 - 2점: 일부 지원함
 - 1점: 거의 지원하지 않음

연구	본 연구의 추적 테이블		김주영		안상임		Ramesh		Lee M. Williams		상용 도구	
	사용 편의성	추적링크 가독성	사용 편의성	추적링크 가독성	사용 편의성	추적링크 가독성	사용 편의성	추적링크 가독성	사용 편의성	추적링크 가독성	사용 편의성	추적링크 가독성
A	3	3	3	3	1	3	1	3	1	1	1	2
B	3	2	3	2	1	2	1	3	1	2	1	1
C	3	3	3	3	1	3	1	3	1	2	1	1
D	3	3	3	3	1	3	1	2	1	1	1	2
E	2	3	2	2	1	3	1	3	1	2	1	1
계	14	14	14	13	5	14	5	14	5	8	5	7
비교 결과	O	O	O	△	X	O	X	O	X	△	X	△