

CSCW 환경에 기반한 요구공학 프로세스 모델 설계

황만수[†]·이원우^{††}·류성렬^{†††}

요 약

소프트웨어 개발과 운영이 분산화, 대형화됨에 따라 정확하고 완전한 요구사항의 추출과 명세는 시스템의 가장 중요한 요소가 되고 있다 또한 인터넷을 통한 공동작업 환경에서 계속적인 시스템 변경요청은 더욱 효율적인 요구사항 관리를 필요로 한다. 본 논문에서는 이러한 공동작업 환경에서 자연어 기반 요구사항 명세와 권리의 효율성을 향상시키기 위한 요구사항 명세구조와 기법 등을 정의하고, 요구공학 활동과 주기를 바탕으로 하는 요구공학 프로세스와 환경을 제안한다. 그래서 CSCW(Computer Supported Cooperative Work) 환경에서 요구사항을 정확하게 추출하고 효율적으로 관리하며 분석단계로 자연스럽게 전이가 가능하도록 한다.

A Design of Requirement Engineering Process Model Based on CSCW Environment

Man-Soo Hwang[†] · Won-Woo Lee^{††} · Sung-Yul Rhew^{†††}

ABSTRACT

According to distributed, large-scaled environment of software development and operation, the elicitation and specification of correct and complete requirement is the most important factor for the system. In addition, continuous and dramatic system changing requests in cooperative environment with internet require more efficient requirement management. In this paper we define the specification architecture and techniques for requirements, so that we improve the efficiency of natural language-based requirement specification and management in a cooperative work environment. Also, we propose a software requirement engineering process model and environment based on requirement engineering activity and phase. Then the above model allows us to elicit accurately, manage requirements in a CSCW(Computer Supported Cooperative Work) environment, therefore transfer them into analysis phase.

1. 서 론

대형화, 분산화되는 소프트웨어 개발환경에서 다양한 작업그룹으로부터 발생하는 요구사항들을 시간적, 공간적 제약없이 체계적으로 관리하기 위하여 요구공학(Requirement Engineering) 분야에서 분석, 추출, 명세, 관리, 검증 등의 활동과 원칙들에 관한 많은 연구가 진행되고 있다[1, 2]. 그러나 대부분 자연어로 기술

된 요구사항들은 의미의 모호성(ambiguity)이 발생하거나 공동작업 환경에서 다른 그룹의 참여자들에 의해 부정확하게 정의될 수 있고, 또한 변경으로 인하여 다른 요구사항과의 충돌(conflict), 일관성의 결여(inconsistency) 같은 문제가 발생할 수 있다[3].

본 연구에서는 이러한 문제점을 해결하기 위하여 자연어 기반의 요구사항 명세구조를 정의하고 요구공학적 활동과 주기를 기반으로 분산된 작업그룹으로부터 제시되는 요구사항을 추출, 처리할 수 있는 공동작업(CSCW) 기반의 요구공학 프로세스 모델을 제안하여 사용자 접근에 대해 시간적, 공간적인 투명성을 제공

† 김희원 · 신동대학 컴퓨터정보계열 교수

†† 정희원 · 대신정보통신 금융사업본부

††† 송진희원 · 숭실대학교 컴퓨터학부 교수

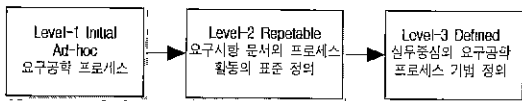
논문접수 : 2000년 4월 20일, 심사완료 : 2000년 9월 27일

하고 향후 요구사항의 변경에 대한 분석, 연관성 추적, 검증 등의 관리를 총체적으로 지원할 수 있도록 한다

2. 관련 연구

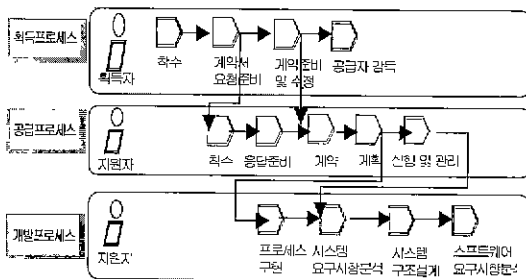
2.1 기존 요구공학 프로세스 연구

기존의 요구공학 프로세스에 관한 연구는 관리적 측면보다는 개발 초기단계에서 표준문서 형태의 요구사항 명세서를 산출하는 요구사항 분석 방법론에 의거한 분석지원 도구들로, Ian Sommerville은 요구공학 프로세스에 요구사항 추출, 분석 및 협력, 그리고 검증과 같은 활동을 포함할 것을 권고하며 요구공학 프로세스 성숙도를 판단하기 위한 모델을 (그림 1)과 같이 3단계로 제안하였다[4].



(그림 1) Sommerville의 요구공학 프로세스 성숙도 모델

또한 소프트웨어 개발 생명주기 표준인 IEEE/EIA 12207에서는 전체 프로세스를 획득, 공급, 개발 프로세스로 구성하고, 각 프로세스별 활동과 참여자들을 (그림 2)와 같이 정의하고 있다[12].



(그림 2) IEEE/EIA 12207의 요구공학 프로세스 모델

2.2 CSCW(Computer Supported Cooperative Work)

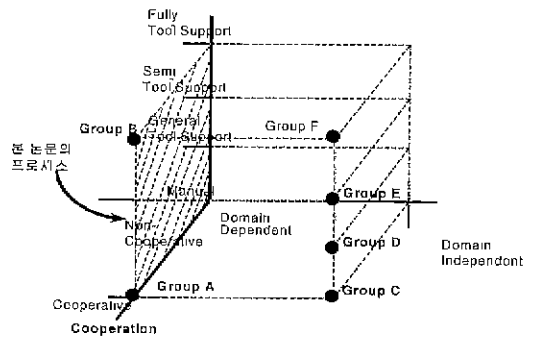
2.2.1 CSCW 개발 환경

CSCW 환경에서의 소프트웨어 개발은 시간적 측면에서는 동기적인 즉시처리, 실시간 처리가 필요하고, 공간적 측면으로는 지역적 분산에 대한 협력작업이 요구되며, 또한 특정 플랫폼에 대해 독립적이어야 한다

[5]. 본 논문에서는 관련자들이 지역적으로 분산되고, 시간적으로 비동기적 또는 동기적으로 프로세스에 접근할 수 있고, 또한 다양한 이종그룹 내에서 작업단위를 중심으로 하는 동종그룹을 지원하며 요구사항 문제의 해결을 위하여 그룹 내의 참여자들이 서로 협력하거나 경쟁하게 되는 4가지 조건을 가정한다.

2.2.2 CSCW 기반 요구공학 연구에 대한 분류

기존 CSCW 기반 요구공학에 관한 연구는 추출, 관리, 협력 알고리즘에 관한 것으로 이것을 요구사항의 분산적 처리 가능성인 공동작업(Cooperation) 정도와 요구사항이 발생하는 도메인의 종속성에 대한 도메인 의존(Domain Dependence) 정도, 그리고 요구사항의 생성과 관리를 자동으로 수행하는 도구사용의 의존에 대한 자동화(Automation) 정도에 따라 구분하면 (그림 3)과 같이 분류할 수 있다



(그림 3) CSCW기반 요구공학 프로세스 분류

<표 1>은 (그림 3)의 분류에 따른 관련 연구로, RTM (Requirements Trace Management)은 데이터베이스를 사용하여 요구 사항의 관리와 연결(Traceability) 설정을 하여 대형 시스템 개발에 효과적이지만 사용자 접근이 어렵고 분산적 처리가 효과적이지 못하다. DOORS (Distributed Object Oriented Requirements System)는 PC 환경에서 객체 지향적 데이터베이스를 기본으로 운용되어 사용이 쉽지만 공동작업에 대하여 효과적이지 못하고 대량의 요구사항을 관리하는 것이 쉽지 않다. 또한 RequisitePro도 PC에서 MS Word 파일을 기본으로 하여 요구 사항들을 관리하므로 작은 프로젝트에 적합하나 대형 프로젝트에는 효용성이 떨어진다 [6, 15].

〈표 1〉 CSCW 기반 요구공학 프로세스 관련 연구

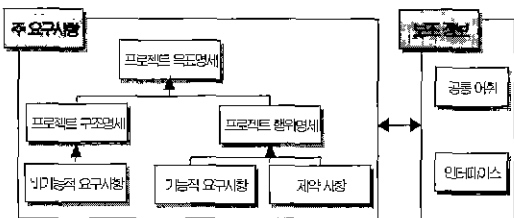
그룹	공동 작업	도메인 의존	도구지위	관련 연구
A	No	의존적	수동	Rational Unified Process
B	Yes	의존적	자동	
C	Yes	독립적	민자용	W 이론
D	Yes	독립적	지원도구 제공	GIBIS, WIN-WIN
E	Yes	독립적	민자용	DOORS, RequisitePro, RTM
F	Yes	독립적	자동	REMAP

3. 요구공학 프로세스 설계 환경

3.1 요구사항 명세구조

3.1.1 요구사항 명세구조

일반적으로 자연어 기반의 요구사항 명세는 공동작업 수행 시 발생하는 요구사항에 대해 표현의 불명확성이나 이해의 모호성 그리고 의미의 불일치를 발생시킬 수 있다. 따라서 이러한 문제를 해결하기 위하여 요구사항 명세구조를 정의하는데, (그림 4)와 같이 요구사항의 추상적인 부분으로부터 구조적인 측면과 행위적인 부분으로 분리하고 이들간의 연관관계를 표시하여, 요구사항 구성에 직접적으로 관련된 주 요구사항과 요구사항 구성을 보조적으로 지원하는 보조 정보로 구성하여 정의한다.



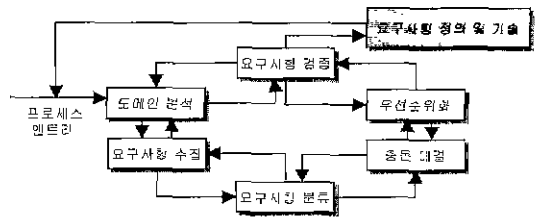
(그림 4) 요구사항 명세 구조

주 요구사항에서는 명세자의 관점에서 추상적인 부분으로부터 비 기능적 요구사항의 구조명세와 기능적 요구사항과 제약사항에 관한 행위명세로 분리하여 이들 간의 연관관계를 표시한다. 요구공학 프로세스 관계자 측면에서 가장 추상적인 프로젝트 명세는 프로젝트의 주요 기능군의 명세인 프로젝트 행위명세와 주요 비 기능적 요구사항의 집합인 프로젝트 구조명세를 생성하고, 프로젝트 행위명세는 세부 기능적 요구사항과 특정 기능적 요구사항 및 요구사항 군에 미치는 제약사항을 파생한다. 또한 보조정보인 공동어휘와 인터페

이스 정의는 주 요구사항의 각 요소와 연관관계를 가지게 된다.

3.1.2 요구사항 명세 작업흐름

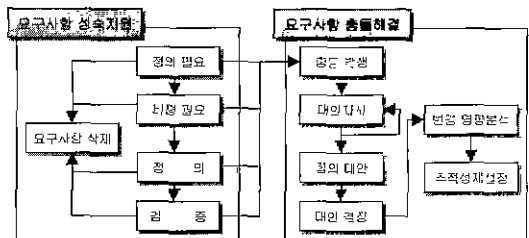
요구사항 명세를 위한 작업흐름은 요구사항이 발생하는 도메인을 분석하여 요구사항을 수집한 후 역할과 기능에 따라 분류한다. 그리고 이 요구사항이 기존의 요구사항과 충돌이 발생하는지에 대해 연계설정에 따라 추적을 하고 만일 충돌이 발생하면 충돌해결을 수행하고 검증을 한다. 검증이 완료되면 요구사항에 대한 정의 및 기술을 수행한다. (그림 5)에서는 요구사항 명세를 위한 전체적인 작업 흐름을 보여주고 있다[7,8].



(그림 5) 요구사항 명세를 위한 작업 흐름도

3.2 요구사항 협력 기술지원 체계

공동작업 환경에서 여러 그룹으로부터 상이하게 제시되는 연관된 요구사항에 대해 일치성과 완전성 유지를 위하여 요구사항 상태의 성숙을 지원하기 위한 명세요소의 기술지원 체계와 충돌해결 체계를 (그림 6)과 같이 정의한다.



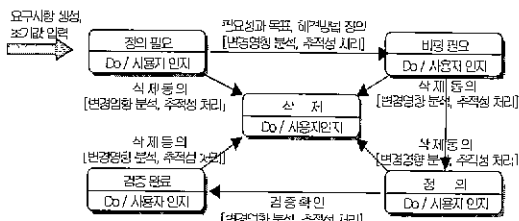
(그림 6) 요구사항 협력 기술 지원 체계

3.2.1 요구사항 성숙 지원체계

- (1) 정의필요 : 제시된 요구사항의 필요성과 해결가능 방법을 제시하여 정의를 유도한다.
- (2) 비평필요 : 제시된 요구사항에 대해 초기 값이 입력된 상태로, 반복적으로 비평과 수정을 한다.

- (3) 정의 : 요구사항 설계단계를 거쳐 요구사항의 명세를 정의하고 확정한다
- (4) 검증 : 요구사항 검증 활동에 의해 검증을 수행한다
- (5) 삭제 : 관련자들의 동의에 의하여 불필요한 요구사항을 삭제상태로 변경하여 관리한다

이와 같은 상태들 간의 전이체계는 (그림 7)과 같이 정의된다.



(그림 7) 요구사항 신속지원 상태전이 체계

3.2.2 요구사항 충돌 해결체계

요구사항의 상태전이가 발생되면 변경으로 인한 영향분석과 추적성 실행을 위한 기능설정을 하여 요구사항을 일관성 있게 유지한다. 이때 발생하는 불일치에 대한 해결체계는 다음 요소를 포함한다.

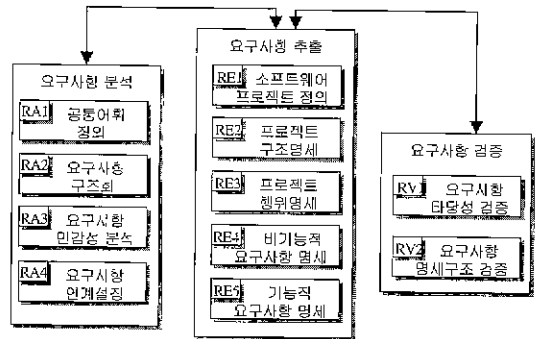
- (1) 충돌발생 : 새로운 요구사항 또는 변경 요구에 따른 정의가 기존의 요구사항과 상충될 경우 충돌을 인지한다.
- (2) 대안 제시 : 프로세스 개발자가 관련된 개발 그룹들간에 발생된 충돌에 대한 대안을 제시하고 합의된 대안을 재 정의한다.
- (3) 합의대안도출 : 제안된 대안들 중에서 합의대안을 도출한다.
- (4) 대안결정 : 관련자들간의 협의에 의해 합의대안 중에서 대안을 결정한다.
- (5) 변경영향 분석 : 변경된 요구사항이 기존의 요구사항들에 대한 영향을 검토하여 연관된 요구사항들을 식별한다.
- (6) 추적성 재 설정 : 관련된 요구사항들 간의 연계성을 재 설정한다

4. CSCW 기반 요구공학 프로세스 설계

4.1 CSCW 기반 요구공학 프로세스 활동과 주기

본 논문에서 제안하는 프로세스는 IEEE/EIA 12207

과 Sommerville의 요구공학 활동을 근간으로 (그림 8)과 같이 요구사항의 추출, 분석, 검증의 세 단계로 구성하여 추출 프로세스를 중심으로 분석과 검증 프로세스를 병행 수행한다. 요구사항 추출활동은 프로세스 개발과 관계되는 모든 그룹으로부터 발생하는 요구사항에 대해 각 요소와 시스템 제약사항 등을 추출 및 기술하고, 분석활동에서는 추출된 요구사항의 분석을 통하여 제사용 여부와 요구사항간의 추적 가능성과 일관성을 설정하고 유지한다. 또한 검증활동에서는 추출과 분석 활동을 통한 요구사항의 시험 가능성, 소프트웨어 설계의 타당성 및 운영과 유지보수의 타당성 등을 확인하고 검증한다



(그림 8) 제안된 CSCW 기반 요구공학 프로세스 모델

4.2 요구사항 분석 프로세스

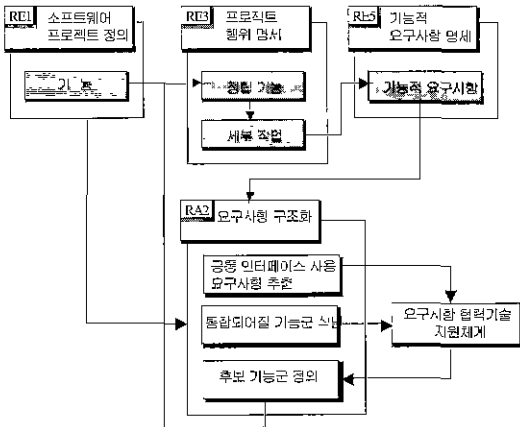
4.2.1 공통어휘 정의

공통어휘 정의는 시스템의 일관성 유지를 위하여 요구사항에서 사용되는 주요 이휘와 약자를 체계로 선택하여 정의와 명세를 한다. 이러한 정의는 요구사항 추출의 모든 단계에서 지속적으로 진행하며, 추출된 공통어휘는 구분 가능한 태그를 포함하여 테이블(The-saurus Table)에 저장한다.

4.2.2 요구사항 구조화

요구사항 구조화는 획득된 기능적 요구사항의 체계를 설정하는 활동으로, 비 기능적 요구사항은 정의된 체계를 기반으로 추출 및 관리가 가능하지만, 기능적 요구사항은 요구사항 추출이 진행되는 동안 기능 군은 세부기능으로, 각 기능은 세부작업으로 구성되는 체계를 지속적으로 수행한다 즉 프로젝트 정의단계에서 정의한 핵심기능을 중심으로 기능 군을 정의하고, 공

통적인 외부 인터페이스를 사용하는 요구사항을 수집하여 후보기능 군으로 정의한 후 정제작업을 거쳐 기능 군으로 설정한다. (그림 9)에서는 요구사항 구조화 세부작업과 관련 형상을 보여준다



(그림 9) 요구사항 구조화 세부 작업과 관계된 형상

4.2.3 요구사항 민감성 분석

민감성 분석은 요구사항간의 불일치로 인한 충돌을 최소화하기 위하여 내용이 변경되었을 때 다른 형상에 영향을 미치는 요구사항을 식별하고, 이를 예상 충돌 요구사항으로 정의한다[9, 10]. 그리고 다른 요구사항들이 받는 영향을 개별 요구사항들과 내역을 바탕으로 분석하고 기술하는 작업으로, 연계 설정된 요구사항을 기준으로 그 요구사항이 변경되었을 때에 연계 설정에 대한 타당성을 분석하고 내용을 기술하게된다. 이후에 충돌이 감지되면 관련자들에게 보고하여 요구사항 변경으로 인한 영향을 최소화 줄일 수 있도록 한다

4.2.4 요구사항 연계 설정

연계설정은 연관된 요구사항의 내용에 의해 색인추출(index retrieval), 슬라이딩 윈도우(sliding window)에 의한 단어 쌍 추출, 또는 의존구조에 의한 단어 쌍 추출과 같은 유사도 검색 엔진을 통하여 요구사항 간의 유사도 계산을 한 후 연관관계를 설정한다[16]. 그리고 설정된 연관성에 따라 주어진 요구사항의 내용과, 작성자, 작업그룹 등의 정보를 이용하여 추적을 수행하는데, 이러한 연계설정은 기능적 요구사항의 경우 기능군 내의 요구사항 집합을, 비 기능적 요구사항의 경우 분류 내 요구사항 집합을 대상으로 하여 이를 프

로세스 관계자에게 보고, 인지하도록 한다.

4.3 요구사항 추출 프로세스

추출 프로세스는 요구사항 및 요구사항 명세를 이루는 각 요소를 명세, 기술을 위하여 추출하는 작업으로 다음과 같은 활동을 수행한다.

- 시스템 요구사항 분석 : 시스템의 기능과 능력, 안정성, 보안성, 인터페이스, 운영 및 유지보수 요구사항, 설계 제약사항, 자격 요구사항 등을 기술한다.
- 소프트웨어 요구사항 분석 : 소프트웨어 수행 환경 조건, 성능, 물리적 특성을 포함한 기능명세, 소프트웨어 항목에 대한 외부 인터페이스, 안전명세, 보안명세, 데이터베이스 요구사항, 설치 및 인수 요구사항, 운영과 실행 요구사항, 유지보수 요구사항 등을 기술한다.

4.3.1 소프트웨어 프로젝트 정의

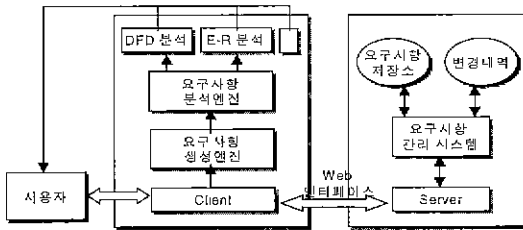
소프트웨어 프로젝트 정의는 요구사항의 행위적 측면을 추출하여 세부 요구사항 명세단위를 추출하기 위한 것으로, 개발할 시스템의 기능을 중심으로 명세한다. 이 단계에서는 요구사항의 주체어, 식별자, 생성일, 자연어 형태의 요구사항 내용, 프로젝트 기능, 기능별 입력물 및 산출물 그리고 관련자 등을 추출하며, 다음 단계에서 세부 요구사항을 산출하는 주요 형상으로 사용된다[11]. 이 프로세스에서 수행되는 세부 기능은 다음과 같다.

- 요구사항의 목적 기술 : 사용자 문제 또는 프로젝트 효과에 대한 배경을 기술하고, 요구사항이 산출될 목적과 이유를 기술한다.
- 요구사항 번호설정 : 각 요구사항이 개발과정 동안 추적이 가능하도록 식별번호(Requirement Identifier Number : RID) 부여에 대한 기준을 설정하고 번호를 부여한다.
- 요구사항의 제약사항 설정 : 요구사항 명세를 위하여 문맥 다이어그램(context diagram)을 작성하고, 시스템을 세분화하여 사건 리스트(event list), use case 리스트 및 시스템이 지원해야 하는 모든 사건을 식별한다.
- 기능과 데이터 요구 표현 : 각각의 기능적 요구사항과 데이터 요구사항 즉 비 기능적 요구사항에 대해 기술한다.

4.3.2 프로젝트 구조명세

소프트웨어 프로젝트에 대한 구조적 정보 및 요구사항을 기술하는 것으로, 요구사항 변경시 발생할 수 있는 불일치 및 연관된 요구사항에 대한 영향을 최소화하기 위하여 다른 형상에 영향을 미칠 수 있는 요소를 구조적 요소로 분리하여 미리 추출한다. 이와 관련된 구조적 요소로는 프로젝트 비용, 프로젝트 수행기간, 개발 방법론, 외부 인터페이스(하드웨어, 소프트웨어, 통신, 사용자 인터페이스 등) 및 소프트웨어 시스템 사용자 등으로 분류된다

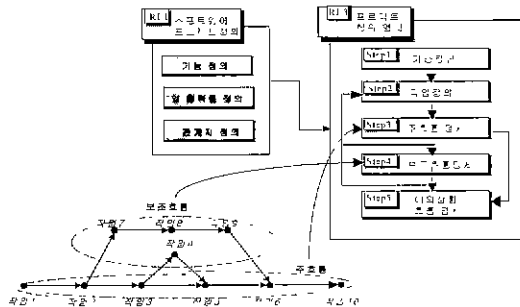
본 논문에서는 다중 사용자 환경에서 발생하는 요구사항 정보의 원활한 수집과 기술을 위하여 (그림 10)와 같은 다수의 대등한(peer) 클라이언트와 하나의 관리 시스템으로 구성되는 Client/Server 시스템 환경 구조를 제시한다.



(그림 10) 다중 사용자 요구사항 관리 시스템 구조

4.3.3 프로젝트 행위명세

프로젝트에 포함되는 주요 기능의 흐름을 이해하고 표현하는 활동으로 프로젝트 정의단계에서 획득된 주 기능을 중심으로 기능을 정립하고, 정립된 기능에 관한 세부작업을 정의하며 관련된 작업들 간의 흐름을 주 흐름, 보조 흐름, 예외 흐름으로 구분하여 기술한다 (그림 11) 이때 정립된 기능내의 세부작업에 대한 정



(그림 11) 프로젝트 행위 명세 단계 작업 흐름

의는 주 흐름명세와 보조 흐름명세의 진행과 더불어 반복적이고 점진적으로 획득한다

4.3.4 비 기능적 요구사항 명세

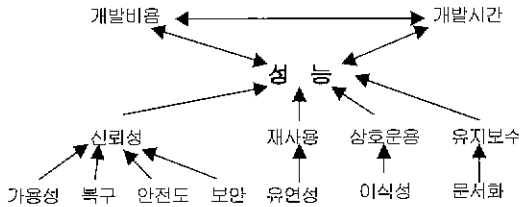
비 기능적 요구사항은 기능적 요구사항의 구현 시 고려할 히는 제약사항으로, 표준 명세요건을 만족하면서 정확하게 분류하는 것이 필요하다. 추출되는 명세요건과 내용은 각 표준에 따라 다르게 정의되는데 IEEE 830에서는 성능, 신뢰성, 가용성 유지보수성, 이식성과 보안 및 안전에 대한 것을 명세요건으로, BS 6719에서는 성능, 시간, 비용, 유연성, 이식성 및 보안 등에 관한 사항을 주요 명세요건으로 정의하고 있다. 또한 DoD 7935A에서는 성능, 비용, 유연성 및 보안과 안전도에 대한 내용을 명세요건으로 정의한다[12-14]. 이와 같이 각 표준에서 제시하는 명세요건이 다르기 때문에, 본 논문에서는 기존의 각 표준 명세요건들을 상호 비교, 분석하여 제시되지 않거나 일부만 정의하는 항목을 추가하여 요구사항 명세요건을 제안한다 <표 2>에서는 IEEE 830, BS 6719, DoD 7935A에서 제시하는 명세 요건과 본 논문에서 제시하는 요건을 나타내고 있다

<표 2> 비 기능적 요구사항 비교

표준/명세	IEEE 830-1993	BS 6719-1986	DoD-7935A	제안 분류
성능	기술	기술	기술	○
시간	×	기술	×	○
비용	×	기술	기술	○
상호운용	×	×	×	○
제사용	×	×	×	○
표시화	×	×	×	○
신뢰성	기술	×	×	○
가용성	기술	×	×	○
복구	기술	기술	기술	○
유지보수	기술	×	×	○
유연성	×	기술	기술	○
이식성	기술	기술	×	○
보안	기술	기술	기술	○
안전도	기술	기술	기술	○

기 비 기능적 요구사항은 다른 요구사항과 상호 연관관계를 가지고 있기 때문에 연관관계가 많은 요구사항 요건부터 추출하여 변경요구 때 발생할 수 있는 변경작업과 요구사항간의 불일치 등을 줄일 수 있게 하고 이후에 신뢰성, 제사용, 상호운용, 유지보수 등과 관련된 요구사항을 추출한다. (그림 12)에서는 비 기능적

요구사항간의 관계를 나타낸다.



(그림 12) 비 기능적 요구사항 간의 관계

4.3.5 기능적 요구사항 명세

기능적 요구사항 명세는 프로젝트 행위 명세에서 정의된 기능과 세부작업을 중심으로 세부 요구사항을 명세화한다. 추출되는 명세요건과 내용은 각 표준에 따라 다르게 정의되는데, 요구사항 내용의 기술, 우선 순위, 입력 및 출력물은 모든 표준에서 제시하고 있다. 그러나 IEEE 830에서는 식별자와 제목을 기술하는데 비해 다른 표준은 제시하지 않고 있고, 타 형상과의 관계는 DoD에서만 제시하고 있다. 따라서 본 논문에서는 기존의 표준으로부터 제시되지 않거나 일부에 정의된 항목을 추가한 요구 명세요소를 제시한다 <표 3>에서는 IEEE 830, BS 6719, DoD 7935A에서 제시하는 기능적 요구사항의 명세 요건을 분석하고, 이 요건들을 기준으로 본 논문에서 제안하는 요건을 제시하고 있다[12-14].

<표 3> 기능적 요구사항 획득 요건

명세요건 \ 표준	IEEE 830-1993	BS 6719-1996	DoD-7935 A	제안 분류
식별자	상세기술	×	×	○
요구사항 제목	상세기술	×	×	○
요구사항 기술	상세기술	상세기술	상세기술	○
생성자	×	×	×	○
생성일자	×	×	×	○
타형상과의관계	×	×	기술	○
우선순위	상세기술	상세기술	기술	○
안정성	상세기술	×	×	○
출물여부	×	×	×	○
주제어	×	×	×	○
외부인터페이스	×	×	×	○
입력물	상세기술	상세기술	상세기술	○
출력물	상세기술	상세기술	상세기술	○

4.4 요구사항 검증

요구사항의 검증 단계는 추출과 분석 활동의 정당성

과 타당성을 확인 및 검증하며 생성된 요구사항들 간의 일치성을 확립하는 것으로, 만일 요구사항 간에 충돌이 발생하면 불일치로 보고되어야 한다.

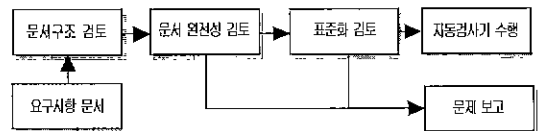
4.4.1 요구사항 타당성 검증

요구사항 타당성 검증단계에서는 요구사항 협력기술 지원체계를 통하여 명세된 요구사항의 구현 가능성, 명세표현의 정확성 및 완전성, 표준과의 일치성, 요구사항간의 충돌, 기술적 결합 등에 대한 검증을 수행한다. 이 단계에서 수행되는 행위는 다음과 같다.

- 시스템 요구사항 검증 : 시스템 설계의 타당성, 운영 및 유지보수의 타당성 등을 검증한다.
- 시스템 구조 설계 검증 : 사용된 설계 표준과 방법의 적절성, 할당된 요구사항을 충족시킬 수 있는 소프트웨어 항목의 타당성, 운영 및 유지보수의 타당성 등을 검증한다.
- 소프트웨어 요구사항 검증 : 소프트웨어 설계의 타당성, 운영과 유지보수의 타당성 등을 검증한다.

4.4.2 요구사항 명세 구조 검증

각 단계별 명세 요건들이 완전하고 정확하게 명세되었는가에 대한 요구사항 명세구조 검증을 수행한다. 이와 같은 검증에 대한 처리 순서는 (그림 13)과 같다



(그림 13) 요구사항 명세 검증 흐름도

4.5 요구공학 프로세스와 산출물 연계

제안된 요구사항 프로세스 모델을 통하여 생성되는 산출물은 요구공학 프로세스 이후 단계에서의 사용에 대한 정의를 통하여 분석 및 설계 단계로의 빠른 전이와 일관성을 가지고 실행될 수 있도록 한다.

4.5.1 객체지향 방법론 연계를 이용한 개발

대표적인 객체지향 개발 방법론인 Use Case 모델을 이용하여 소프트웨어 프로젝트의 기능적인 측면을 표현하고 산출물을 통하여 다음과 같은 요소를 식별할 수 있다[17].

- Actor : 외부 인터페이스를 중심으로 식별한다.
- Use Case . 기능군을 중심으로 식별한다.
- Use Case 간의 관계 : 기능적 요구사항의 체계를 중심으로 식별한다. 체계 내 공통기능은 Uses, 예외사항은 Extends로 할당한다.
- 작업흐름(주, 보조, 예외사항) 정의한 작업흐름으로 식별한다.
- 시나리오 . 작업흐름을 실제화한다.

4.5.2 구조적 방법론 연계를 이용한 개발

대표적인 구조적 방법론인 DFD 다이어그램을 이용하여 기능을 중심으로 자료 흐름을 표현하고 산출물을 통하여 다음과 같은 요소를 식별할 수 있다

- 단말 : 외부 인터페이스를 중심으로 식별한다
- 프로세스 . 기능적 요구사항의 체계를 중심으로 각 단계를 식별한다.
- 데이터 흐름 . 기능적 요구사항의 산출물을 중심으로 식별한다

4.6 프로세스의 평가

4.6.1 IEEE 12207 프로세스

IEEE 12207 프로세스는 요구공학 절차와 지침을 정의하고 표준문서 형태의 명세를 하여 요구사항이 체계적으로 분류되고 적용이 쉽지만, 요구사항간의 연관관계에 대한 파악이 어렵고 문서 양이 많아지기 때문에 요구사항의 변경에 대해 효율적으로 대처하지 못한다. 그리고 분석 단계로의 전이에 대한 정의가 없어 요구사항 문서를 이해하기 위하여 많은 비용과 시간이 소모되며 공동작업에 대한 정의가 없어 참여자간의 일관성 결여와 불완전성의 결과를 초래할 수 있다.

4.6.2 Sommerville의 프로세스

Sommerville의 프로세스는 요구사항 절차와 프로세스 활동에 대한 일반적인 정의는 하고 있으나 세부 지침에 대한 기술이 없기 때문에 명세를 이해하기 어렵고 분석단계로의 연계성도 낮아 실제 적용에 있어 효율성이 적다.

4.6.3 Rational Unified Process

RUP는 업무 모델링(Business Modeling)과 요구사항 작업흐름(Requirements Work-flow) 단계에서 공통어,

요구사항 특성, 소프트웨어 구조 문서를 추출하여 초기에 정확한 명세로 요구사항의 변경과 유지보수에 따른 비용과 시간을 절약할 수 있다 그러나 명세를 위한 작업이 많고 각 산출물간의 연계가 명확하지 못하며, 분석 단계에서의 구체적인 활용에 대한 제시없이 입력물, 출력물로만 명기하고 있어 실제적인 적용은 용이하지 못하다. 또한 Use Case 모델을 업무 모델 작업흐름에 적용하여 업무 Use Case 모델을 정의하지만, 이후 작업흐름에 실제적으로 적용할 수 있도록 하는 장치가 없다.

4.6.4 제안된 프로세스

본 논문에서는 IEEE/EIA 12207과 Sommerville의 연구를 기초로 공동작업 환경에서 기존 프로세스가 정의하지 않거나 일부만 정의하는 요구공학 구조를 세부적으로 정의하고, CSCW 환경과 기반을 명시하여 공동작업을 통한 요구사항의 정확하고 체계적인 요구사항 요소 및 명세를 추출, 관리하는 요구공학 프로세스 모델을 제안한다. 그리고 각 요구사항 요소와 분석단계의 산출물간의 연관관계를 명시하여 분석단계로의 전이가 가능하도록 한다. 또한 기존의 프로세스들은 특정 도메인에 대한 도구 및 환경지원 없이 요구공학을 진행하고 있지만 본 논문에서는 특정 도메인의 설정 후 프로세스 적용이 진행되도록 하며, 공동작업을 기반으로 자동화 강도를 선택적으로 구현하도록 한다 <표 4>에서는 IEEE/EIA 12207, Sommerville, Ration-

<표 4> 타 프로세스 모델과의 비교

평가기준\프로세스	IEEE/EIA 12207	Sommerville	RUP	제안 프로세스
요구공학 작업정의	정의	정의	정의	요구공학 프로세스 정의
요구공학 시정정의	일부 정의	미 정의	일부 정의	요구공학 프로세스 활동과 주기 정의
요구사항 이해도	낮음	낮음	높음	공동작업을 통해 정제된 요구사항 도출
요구사항 적용성	높음	낮음	높음	높음
분석단계로 연계성	낮음	비지원	낮음	요구공학 프로세스와 산출물 연계
공동작업 지원도	비지원	비지원	비지원	요구사항 협력 지원 체계 지원
요구사항 명세효율성	중간	낮음	중간	명세구조 상의에 의해 효율성 증대
기반 언어	자연어	자연어	자연어, UML	자연어, UML
요구사항 명세간성	높음	낮음	높음	도구에 의한 자동이 아닌 수동으로 명세

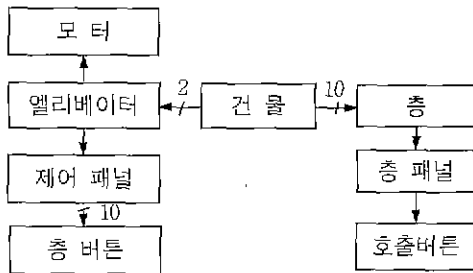
al Unified Process와 제안된 프로세스의 비교 결과를 보여준다

4.7 요구사항 명세 예

본 논문에서 제시하는 요구공학 프로세스를 이용하여 다음과 같이 수집된 내용을 분석하여 요구사항을 명세 요건에 따라 생성한다.

10층으로 구성된 건물에 각 층에는 2개의 엘리베이터가 있고, 각 층에는 층 패널과 엘리베이터 호출 버튼이 있다. 또한 엘리베이터에는 제어 패널 층 버튼이 있다

생성된 요구사항을 분석을 위하여 E-R 다이어그램으로 표현하면 (그림 14)와 같이 나타난다.



(그림 14) E-R 다이어그램을 이용한 요구사항 분석

위에서 제시된 사용자 요구를 행위사항과 제약사항으로 구분하여 기능적 요구사항과 비기능적 요구사항으로 분류하고, 보조정보인 공동어휘를 식별하며 제시된 명세요건에 따라 (그림 15)와 같이 요구사항을 생성한다.

변경요구에 의해 수정된 요구사항이 다른 요구사항과 충돌하는 것을 최소화하기 위하여 다른 형상에 영

향을 미칠 수 있는 요구사항을 식별하여 추적할 수 있도록 유사도 검색 엔진을 통하여 유사도 계산을 하여 두 요구사항간의 연계관계를 설정한다. 생성된 요구사항을 비교하여 요구사항 간의 연관성을 찾아내어 요구사항을 연계 설정하면 (그림 16)과 같다.

연계할 요구사항 번호 #1

RN21

인계할 요구사항 번호 #2

RF23

연계설정

RID RN21
Sentence 각각의 엘리베이터는 제어패널을 가진다.
Trace Sentence
RID RN22
Sentence 제어패널에는 층 당 하나씩의 10개의 층 버튼이 있다
RID RF23
Sentence '사용자가 층버튼을 누르면 제어패널은 요구된 층 번호를 전송한다'
RID RF41
Sentence '제어기는 운행 명령을 엘리베이터 모터로 전송한다'
Trace Sentence
There is no candidate sentence

(그림 16) 요구사항 간의 연계 리스트

5. 결 론

본 논문에서는 공동작업 환경에서 요구사항을 분석, 추출, 검증함으로써 소프트웨어 생명주기를 통하여 일관성 있게 관리할 수 있는 요구공학 프로세스를 제안하였다 이러한 프로세스를 통하여 요구사항에 대한 잘못된 분석 및 불일치를 해소하고 개발단계로의 자연스러운 전이를 함으로써 시스템 완성도를 높이고 개발 비용 및 시간을 절감할 수 있는 기초를 이룰 수 있다. 제시된 요구공학 프로세스 모델을 통하여 얻어지는 효과는 다음과 같다.

- (1) 공동작업을 통한 신속하고 명확한 프로세스의 신

RID	Title	Requirement Specification	Name	Date	Status	Keyword
RN11	건물1	건물은 10층으로 구성된다	WG1	07-20	승인	층
RN12	건물2	건물의 각 층에는 2개의 엘리베이터가 있다.	WG1	07-20	승인	엘리베이터
RN21	제어패널	엘리베이터는 제어패널을 가진다.	WG2	07-22	승인	제어패널
RN22	층버튼	제어패널에는 층 당 하나씩의 10개의 층 버튼	WG2	07-22	승인	제어패널 층버튼
RF23	제어패널	사용자가 층버튼을 누르면 제어패널은 요구.	WG2	07-22	대기	제어패널 층버튼
RN31	층패널	건물의 각 층에는 층 패널이 있다	WG3	08-01	승인	층패널
RN32	층2	각 층 패널은 하나의 엘리베이터 호출버튼이.	WG3	08-01	제안	층패널 제어패널
RF33	호출버튼	사용자가 호출버튼을 누르면, 각 층의 패널..	WG3	08-01	대기	호출버튼 제어기
RF34	운행모터	각 엘리베이터는 운행을 제어하는 모터와 인..	WG3	08-01	제안	모터
RF41	운행명령	제어기는 운행 명령을 엘리베이터 모터로 전.	WG3	08-01	제안	모터

(그림 15) 생성된 요구사항 명세

행이 가능하다.

- (2) 소프트웨어 개발 주기의 다음 단계로의 신속하고 명확한 명세를 전이할 수 있다.
- (3) 시스템 개발의 일관성을 유지할 수 있다.

현재 거의 모든 시스템의 개발 환경이 웹을 기반으로 수행되는 추세에 있으므로 이러한 프로세스를 근간으로 하여 웹에서 공동작업을 효율적으로 수행할 수 있는 요구사항 관리 도구에 대한 개발과 요구사항 간의 충돌식별 및 이의 해결을 위한 기법의 대한 연구가 계속될 것이다.

참 고 문 헌

[1] K. Pohl, Process-Centered Requirements Engineering, John Wiley & Sons Inc, 1996.

[2] R. H. Richard & M. Dorfman, Software Requirements Engineering, 2nd Ed., IEEE Computer Society Press, 1997.

[3] A. M. Davis, D. A. Leffingwell, Using Requirements Management to Speed Delivery of Higher Quality Applications, Rational, 1996.

[4] Ian Sommerville and Pete Sawyer, Requirement Engineering ; A Good Practice Guide, Wiley, 1997.

[5] J. D. Palmer and N. A Fields, "Computer-Supported Cooperative Work," IEEE Computer, Vol 27, No 5, May, 1994.

[6] INCOSE, "Requirement Management," <http://incose.org/workgrps/rwg>, 1996.

[7] Nuseibeh, B. Kramer, "A Framework for Expressing the Relationships between Multiple Views in Requirements Specification," IEEE Trans. on Software Engineering, Vol.20, No 10, 1994

[8] C. L. Heitmeyer, R. D. Jeffords. B. G. Labaw, "Automated Consistency Checking of Requirements Specifications." ACM Trans. on Software and Methodology, Vol.5. 1996.

[9] M. Heimdahl, and N. Leveson, "Completeness and Consistency Analysis of State-based Require-

ments," IEEE Trans. on Software Engineering, Vol.22, 1996

[10] Hoh In, "Conflict Identification and Resolution for Software Attribute Requirements," Univ of Southern California, Doctoral Dissertation, 1998.

[11] I. Sommerville, P. Sawyer, and S. Viller, "Viewpoints for Requirements Elicitation : A Practical Approach," Proc. of the 3rd IEEE Int'l Conference on Requirements Engineering, April, 1998

[12] IEEE, "IEEE Standard Glossary of Software Engineering Terminology," IEEE Std 610.12-1990. 1991.

[13] IEEE, "IEEE Recommended Practice for Software Requirements Specifications." IEEE Std 830-1993, 1993

[14] DoD, Functional Description, U.S. Department of Defence (Extracted from DoD Automated Information Systems Documentation Standards). 1998.

[15] 이원우, 황만수, 박수용, "웹을 이용한 요구사항 관리 모델의 구축", 한국정보과학회 '98 가을 학술발표논문집(I), 제25권 제2호, 1998.

[16] 박수용, "문서간 유사도 측정을 통한 효율적인 사용자 요구분석", HCI '99 학술대회 발표 논문집, 1999.

[17] 박수용, "객체지향 어플리케이션 개발을 위한 UML 기반의 요구공학 프로세스", 한국 소프트웨어 공학 학술대회, 1999.



황 만 수

e-mail : mshwang@shc.ac.kr

1984년 중앙대학교 전자계산학과 (이학사)

1986년 중앙대학교 대학원 전자계산학과 (이학석사)

1995년~현재 숭실대학교 대학원 컴퓨터공학과 박사과정

1987년~1993년 LG 소프트웨어 연구원

1993년~현재 신홍대학 컴퓨터정보계열 조교수

관심분야 : 소프트웨어공학, 요구공학, 교환기통신



이 원 우

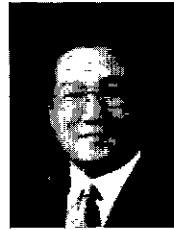
e-mail : wwlee@dsic.co.kr

1998년 송실대학교 컴퓨터학부
(공학사)

2000년 송실대학교 대학원
컴퓨터학과(공학석사)

현재 대신정보통신 금융사업본부

관심분야 : 요구공학, 분산 객체 컴퓨팅



류 성 열

e-mail : syrhew@comist.songsil.or.kr

1997년 아주대학교 컴퓨터학부
(공학박사)

1997년~1998년 George Mason
University 교환 교수

1981년~현재 송실대학교 컴퓨터
학부 교수

1998년~현재 송실대학교 정보과학대학원 원장

1998년~현재 송실대학교 전자계산원 원장

관심분야 : 리엔지니어링, 분산 객체 컴퓨팅, 소프트웨어
재사용