

## 기업 아키텍처(Enterprise Architecture)를 지원하는 요구사항 분석 프로세스에 관한 연구

최봉균\*, 임춘성\*

### A Study on Requirement Analysis Process that support Enterprise Architecture Design

Bong Kyun Choi, Choon Seong Leem

#### Abstract

소프트웨어 개발공정 상의 요구사항 문서적 분석에 치중되어 있던 요구사항 분석에 관한 논의는 기업 통합을 더욱 효과적으로 하기 위해 소프트웨어적 차원을 넘어 기업 아키텍처 기반의 정보시스템 개발 및 통합의 필요성에 대한 논의로 확대되고 있다.

본 연구에서는 궁극적으로 기업 아키텍처의 효과적인 수립을 지원하는 요구사항 분석, 관리 프레임 워크 분석 프로세스를 제시한다. 요구사항 분석 프로세스는 기업 아키텍처의 개념을 바탕으로 다양한 사용자와 관점이 구별하고, 이를 다시 요구사항 추출을 위해 업무적 관점과 정보시스템 관점으로 구분한다. 이러한 요구사항 분석을 위한 프로세스는 기업 아키텍처의 수립을 효율적으로 지원할 것이다.

*Key Word* : Enterprise Architecture 설계, 요구사항 분석 프레임 워크, 요구사항 분석 프로세스,  
요구사항 관리

---

\* 연세대학교 컴퓨터산업시스템공학과

## 1. 서론

### 1.1. 연구배경 및 목적

e-Business가 기업의 핵심적인 경쟁 요소로 대두되면서 기업 내부의 통합뿐만 아니라 기업이 속한 공급사슬망 내의 통합 등이 이슈로 떠오르고 있다. 이러한 기업통합에 대한 논의는 기업의 현재 모델(As-Is Model)을 바탕으로 다양한 정보시스템 분석, 설계 방법론 등을 거쳐 목표 모델(To-Be Model)을 정의하는 과정으로 나타났으며, 이러한 기업 변화에 유연한 기업 아키텍처의 정의가 필요한 현실이다.

이에 본 연구는 기업 아키텍처의 효과적인 수립을 지원하기 위하여 요구사항 관리가 필수적이며, 요구사항 관리를 위하여 요구사항의 계층을 기업 아키텍처의 계층과 분류체계에 따른 요구사항 관리가 이루어져야 하므로, 기업의 요구사항 추출 시 초기 요구사항을 계층에 맞는 분류 속성에 따라서 분류, 관리할 수 있는 프레임 워크를 제시하고, 이에 따른 요구사항 분석 프로세스를 정의하고, 기업 아키텍처의 수립을 지원할 수 있도록 요구사항 관리 가이드 라인을 제시하여 기업 아키텍처 수립의 상호운용성, 재사용성, 유연성을 높일 수 있는 방안을 제시한 것을 목적으로 한다.

### 1.2. 연구 범위 및 방법

연구의 범위는 기업 아키텍처와 기업 참조모형, 정보기술 아키텍처에 관한 연구, 소프트웨어 요구공학 프로세스에 관한 연구,

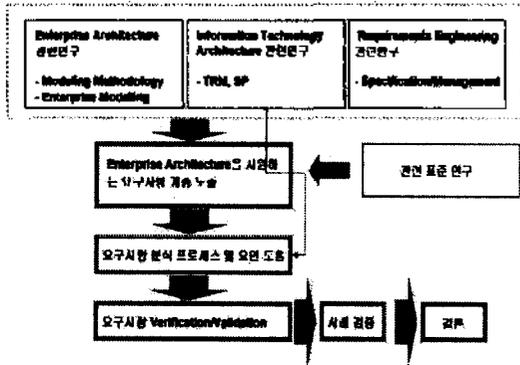
정보시스템 개발 방법론, 요구사항 관리 등을 포함한다.

본 연구는 기업 아키텍처와 기업 참조모형, 정보기술 아키텍처에 대한 개념을 정의하여, 각 모형의 관점을 분류·정의하고, 이러한 분류체계에 따른 요구사항 분석 방법과 가이드라인을 제시하고자 한다.

이를 위하여 다음과 같은 연구의 접근 방법을 따른다.

- △ 기업 아키텍처를 지원하는 요구사항 분석 프레임 워크 도출
- △ 기업 아키텍처와 기업 참조모형, 정보기술 아키텍처에 대한 개념 정립
- △ 기업 아키텍처의 관점 분류 및 연관성 도출
- △ 기업 아키텍처 관련 동향 조사
- △ 기술 참조 모델, 표준 모델 연구, 조사
- △ 요구사항 분석 프로세스 및 가이드라인 도출
- △ 소프트웨어 요구사항 분석 및 관리 프로세스 연구 조사
- △ 기업 아키텍처와의 지원성 및 요구사항 검증(평가)방안 도출
- △ 요구사항 분석 프로세스 도구 프로토타이핑
- △ 요구사항 분석 프로세스 사례 적용

이를 도식화하면 아래 그림과 같으며, 이러한 접근 방법을 통하여 기업 아키텍처를 지원하는 요구사항 분류체계 및 규칙, 관리중점(방법론적) 가이드 등을 도출할 것이며, 사례 적용을 통하여 요구사항 분석 프로세스를 적용할 것이다.



<그림 1> 연구 방법

본 연구는 위 그림과 같은 연구 방법으로 2장에서는 기업 아키텍처와 소프트웨어 요구사항 분석에 대한 관련 연구를 정리하여, 요구사항 분석 프레임워크의 관점을 도출하고, 3장에서는 2장에서 도출된 관점을 기준으로 요구사항 분석 프로세스와 가이드라인을 정의하고, 4장에서는 3장의 도출된 분석 프로세스에 따른 분석도구 프로토타이핑과 사례분석을 실시한다. 마지막 5장에서는 논문을 요약하고 향후 연구방향을 제시한다. 적인 분류기준을 도출하고자 한다.

## 2. 기존연구

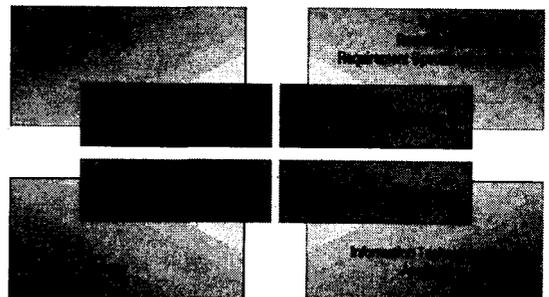
### 2.1 Enterprise Architecture

정보화 또는 정보시스템의 개발, 적용 등에서 기업의 핵심적인 경쟁우위 요소로 떠오르고 있는 기업통합은 다양한 관점의 연구가 진행되어 왔다. 기업통합은 주로 기업 내부의 시스템간의 데이터의 상호교환 또는

기업 외부 파트너와의 정보공유 측면에서 정보시스템 운용환경의 통일성과 호환성을 지칭하는 상호운용성(interoperability)과 정보시스템의 운영적 유연성과 적용성을 나타내는 기민성(agility)의 필요성에서 논의되었다.

기업통합의 관점에서 기업 아키텍처(Enterprise Architecture)는 기업 모델의 유기적인 집합으로 설명될 수 있으며, 기업 아키텍처에 대한 정의는 아키텍처의 기본 개념, 소프트웨어 아키텍처, 시스템 아키텍처, 기술 아키텍처, 기업 아키텍처 등과 혼재된 개념으로 정의되어 왔다. 이러한 혼재된 개념 속에서 기업 모델과 모델 사이의 시스템적인 접근방법을 가지고 조직의 기능 및 기술을 통합하여 전사적인 통합을 지원하는 구조와 체계 등을 통괄하는 개념의 연구가 진행되어 왔다.

이러한 연구의 관점은 크게 CIM, Software Engineering, Information Management, Technical Architecture와 같이 4가지로 분류할 수 있으며 이를 도식화하면 아래 그림과 같다.



<그림 2> Enterprise Architecture 연구 관점

또한, 기업 아키텍처에 대한 연구에서는 CIMOSA, PERA, ARIS, Zachman's Framework, EAP, TOGAF(The Open Group Architecture Framework), Federal Enterprise Architecture Framework 등과

같은 아키텍처에 관하여 모델링 방법론을 포함하거나 정보시스템의 생명주기의 각 단계의 관점에 따라서 아래 그림과 같은 연구의 관점을 비교할 수 있다. 또한 각 아키텍처에서 표현되는 기업 모델의 정보를 그림

<표 1> Enterprise Architecture의 정의

Eberhardt Fochlin, 1991 [11]	System : as a set of different elements so connected or related as to perform a unique function not performable by the element alone Architecture : a top-down description of the structure of the system
F.B. Vernadat, 1996 [12]	Reference Architecture : 모든 시스템의 일반적인 기능, 행위를 나타내는 프레임워크의 하나로 특정한 기업의 특정한 행위와는 독립적인 아키텍처를 의미 Enterprise Architecture : 기업 또는 기업의 시스템의 기능과 각 구성요소의 연관관계를 보여주는 프레임워크를 제공하는 모든 방법의 총칭이고, 이는 그림, model, description 등을 모두 포함 Framework : 특정한 목적을 위해 서로 묶어진 구성 요소들의 집합체 Architecture : 서로간의 관계가 분명하고, 서로 묶어서 하나의 전체를 형성할 수 있는 구성 요소들의 집합체
Bass, Clements, and Kazman, 1997 [5]	Software Architecture of a program or computing system is the structure or structure of the system, which comprise software component, the externally visible properties of those components, and the relationships among them.
Booch, Rumbaugh, and Jacobson, 1999 [7]	An architecture is the set of significant decisions about the organization of a software system, the selection of the structural elements and their interfaces by which the system is composed, together with their behavior as specified in the collaborations among those elements, the composition of these structural and behavioral elements into progressively larger subsystems, and the architectural style that guides this organization—these elements and their interfaces, their collaborations, and their composition
Mehdi Jazayeri, Alexander Ran, Frank van der Linden, 2000 [19]	Software architecture is a set of concepts and design decisions about the structure and texture of software that must be made prior to concurrent engineering to enable effective satisfaction of architecturally significant explicit functional and quality requirements and implicit requirements of the product family, the problem, and the solution domains.
Bernard H. Boar, 1999 [6]	IT Architecture : a set of principles, guidelines and rules that guides an organization through acquiring, building, modifying and interfacing IT resources throughout the enterprise
Open Group [26]	IT Architecture : a formal description of an information technology system, organized in a way that supports reasoning about the structural properties of the system
Glinger-Cohen Act, 1996 [8]	IT Architecture는 조직의 전략적 목표와 정보 자원관리 목표를 달성하기 위해 정보기술을 획득, 유지, 진화 시키기 위한 통합된 프레임 워크
OMB, 1996 [20]	IT Architecture는 조직의 목적과 목표를 지원하는 정보시스템의 요구사항을 확보하고, 정보시스템의 보안, 중복, 상호운용의 직렬성을 보증하며, 새로운 시스템을 확보, 평가하기 위한 표준들의 유지,용용을 지원하기 위하여 정보기술, 관리 프로세스, 비즈니스 프로세스들 간의 관계를 체계화한 것
DOD, 1996 [10]	IT Architecture는 작업(world)과 위치(location), 정보 집합(information set), 응용(application), 기술 하부구조(technology infrastructure) 등의 구성요소 또는 관점을 특성화한 것
IEEE, 1996 [15]	System Architecture : 시스템 구성요소간 관계와 구조를 말하며, 운영환경과 시스템 인터페이스를 포함 System : 사람(people), 기계(machine), 구체적 기능집합을 실행하는 조직화된 방법(method)
INCOSE, 1998 [17]	IT Architecture는 시스템 요소(elements), 인터페이스, 프로세스, 제약조건(constraints), 행동(behavior)으로 정의된 근본적, 통일적 시스템 구조
Federal CIO Council, 1998 [13]	IT Architecture는 새로운 정보기술을 획득, 유지, 진화시키기 위한 통합된 프레임 워크으로, 정보흐름과 작업 프로세스를 통합하여 조직 전략과 목표를 달성하는 수단 Architecture는 정보교환과 자원공유를 가능케 하는 표준(standards)을 구체화 한 것

의 오른쪽과 같이 정보, 기능, 조직, 조정, 자원의 관점으로 분류할 수 있으며 이러한 분류는 다양한 모델링 표현 방법으로 나타낼 수 있다.

기업 아키텍처의 수립 프로세스와 모델링 표현에 대하여 이러한 관점으로 비교를 하기전에 먼저 기업 아키텍처, 기업 모델, 기업 참조 모형 등의 다양한 용어상의 혼재를 없애기 위하여, 용어상의 개념 정립과 다양한 학문 관점에서의 개념을 정리하였다.

여러 아키텍처의 개념은 정보시스템 생명

주기에 따른 모델링 방법론, 표준화 적용성, 참조 모형 활용, 정보 기술 구조 등의 개념이 혼재되어 사용되고 있으며, 아래와 같은 연구에 의해서 다양한 접근 방식으로 연구되어 왔으며, 이는 system, architecture, reference architecture, IT architecture 등의 다양한 용어로 사용된 것을 알 수 있다.

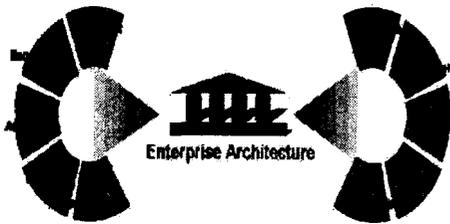
본 연구에서의 기업 아키텍처는 구조 개념을 바탕으로 조직의 기능 및 기술을 통합하여 전사적인 통합을 지원하는 구조와 체계 등을 통괄하는 개념을 지칭하며, 이러한 개

<표 2> Enterprise Architecture의 관점 비교

Generic, Partial, Particular Model (each model has 3 levels)	Function, Data, Orga., Control view (each view has 3 levels)	not defined	Information, Manufacturing Archi. (Concept, Definition, Design layer)	Data, Function, Network, People, Time, Motivation	Function, Implementation, Physical (more detailed views)	Data, Systems, Infrastructure (each view has five user perspectives)
Enterprise Activity	Function	Activity, Function, (Action)	Task Module	Business Process, App. Archi., System Design, Program	Business process, Service layer & application	Adopted Zachman Framework
Bus. Process, Event, (Sequential Relationships)	Process Chain, Event, (Connectors), Cluster	Funct. Chain, Funct. Auton. Unit, (Connect Constructs)	not defined	Work Flow Model, Control Archi., Schedule, Processing Structure	Business Process model, building block selection method	-
Organizational Unit	Organ. Level, Organ. Unit, Attribute, Location, Network, NetworkNode, NetworkUnit, TechResource	Object Class: Special Resource	not defined	Organization Unit, Work Product, Human Interface, Security Archi.	Location, Organ. Unit (in Architecture Development Method)	-
Enterprise Object, Product, Order, Object View, Relation	Entity, Attribute, Relation, Terminology, Table, (Cardinality, Operators)	Object Class: Product, Order, Relation (Operators)	Enterprise Business Entity	Semantic model, Logical/Physical Data model	information base, building block	-
Capability Set, Resource	part of Organisation View	Object Class: Resource		Bus. Location, Distributed System Archi., Technology Archi.	Building block information base	-
not defined	not defined	not defined	not defined	Bus. goal, Bus. Plan, Bus. rule model	Business Senario & Process	-
not defined	not defined	not defined	not defined	not defined	Service layer & application	Technology, application archi.
not defined	not defined	not defined	not defined	not defined	Standards information base	Some guidelines

념을 바탕으로 아래 표와 같은 주요 기업 아키텍처에 포함되는 개념을 비교 분석하였다.

위 <표 2>의 비교 분석은 CIMOSA, PERA, ARIS, Zachman's Framework, EAP, TOGAF(The Open Group Architecture Framework), Federal Enterprise Architecture Framework 등이 비교 대상으로 <그림 3>의 비교 관점에 따라서 분석하였으며, 본 연구에서는 주로 모델의 관점에서 요소들의 정의 여부에 중점을 두어 분석하였음을 밝힌다.



<그림 3> Enterprise Architecture 연구 비교의 관점

## 2.2 요구사항 분석

일반적인 요구사항에 대한 정의를 살펴보면, Robertson과 Robertson은 요구사항을 '제품을 만들기 전에 발견해야 할 모든 것들', '제품이 해야 할 모든 것들과 가져야 할 품질'이라고 정의하였다.[22] 소프트웨어 중심의 연구관점인 요구공학에서는 요구사항의 추출, 분석, 서술, 검증 및 이들의 관리 등을 포함한 요구사항에 관한 모든 활동과 원칙에 대한 체계적이고 총괄적인 접근에 대한 연구를 하고 있는데, 이러한 요구공학은 사용자 요구사항의 획득, 추가 요구사항을 유도하기 위한 요구사항의 분석,

문서화를 위한 요구사항에 대한 명세, 그리고 변경 요구에 대한 관리 등에 포함되는 모든 생명주기 프로세스와 이를 지원하는 행위를 포함한다.[3][18][14]

Ian Sommerville은 요구공학 프로세스에 요구사항 추출, 분석 및 정의, 그리고 문서화와 같은 활동을 포함할 것을 권고하며, 요구공학 프로세스 성숙도를 판단하기 위한 모델을 아래 그림과 같이 3 단계로 제안하였다.[14]

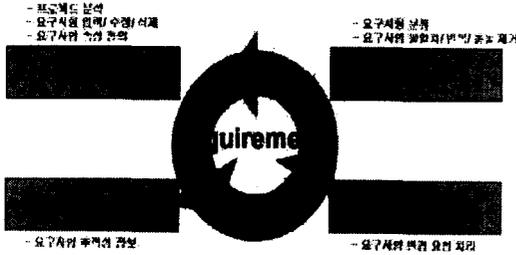


<그림 4> 요구공학 프로세스 성숙모델

Rzepka는 요구사항 명세를 위하여 여러 개별 소스로부터 발생하는 요구사항을 추출하고 모든 사용자의 요구가 일관성과 타당성을 가지는지에 대한 확인 및 사용자 요구를 정확하게 반영하는지에 대한 검증의 세 단계로 요구공학 프로세스를 제시하였다.[23]

요구사항은 시스템에 대한 고객의 요구를 정확하게 반영하지 못하거나, 고객과 시스템 요구사항을 개발하고 유지하는 개발자 사이에 잘못된 이해가 발생하여 요구사항 사이에 불일치와 불완전성을 가질 수 있다. 이러한 요구사항의 관리에 대한 연구에 대하여 A. Davis는 요구사항의 식별, 재구성, 추적성 관리, 변경관리 등의 연구로 구분하여 연구의 관점을 분류하였으며, A. Davis의 연구는 주로 요구사항의 재구성과 추적성 정보 관리에 초점을 맞추어 진행되었으

며, 이는 요구사항이 시스템화 되었을 때의 지속적인 관리에서는 한계를 나타내었다.[3]



<그림 5> 요구사항 관리의 연구관점 [3]

또한, A. Davis는 소프트웨어 개발 생명주기 후반에 에러가 발견되는 경우 수정에 많은 비용이 소요되는데, 이때 발생하는 에러의 대부분이 요구사항 관리의 소홀로 발생한다고 조사하였다.[4] 일반적으로 이와 같이 발생하는 에러에 대한 요구사항 관리 문제를 정리하면 다음과 같다.

- 문제영역에 대한 명확한 이해의 부족
- 그룹내 또는 그룹간에서 참여자 사이에 이해의 문제
- 사용자간의 의사소통에 대한 문제
- 요구사항의 휘발성 문제

즉, 이러한 요구사항 관리는 고객과 고객의 요구사항을 담당할 소프트웨어 개발 팀 사이의 공통 이해를 확립하기 위한 목적으로 사용된다. 요구사항을 관리하는 정형 기법은 다른 관련자들 사이에 발생할 수 있는 견해 차이 또는 명세 에러를 개발 프로세스의 초기 단계에서 나타낼 수 있고, 정확한 요구사항 문서에 의해 소프트웨어 개발 프로세스의 이후 단계에 대해 충실한 기초를

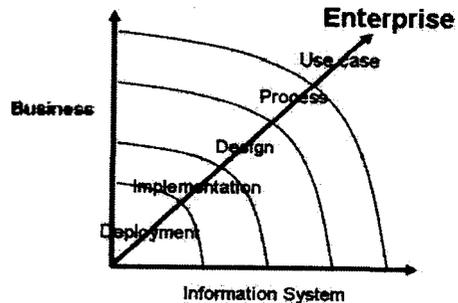
제공할 수 있으며, 다음과 같은 요소로 필요요성을 정리할 수 있다.

- 관련자들 사이에 다른 관점 식별
- 초기 에러 발견
- 개발 이후 단계를 위한 정보 제공

### 3. 기업 아키텍처를 지원하는 요구사항 분석 프로세스

#### 3.1 요구사항 분석 프로세스

위와 같은 기존 연구를 바탕으로 본 연구에서는 요구사항 분석 프레임 워크 분석 프로세스를 제안 하고자 한다. 아래 그림은 기업의 업무 아키텍처의 개념과 정보시스템 아키텍처의 개념의 매핑을 보여주는 그림으로서 기업의 업무와 정보시스템 사이의 차이를 줄임으로써 다양한 관점과 참여자 사이의 의사소통을 원활하게 하도록 도울 수 있다.

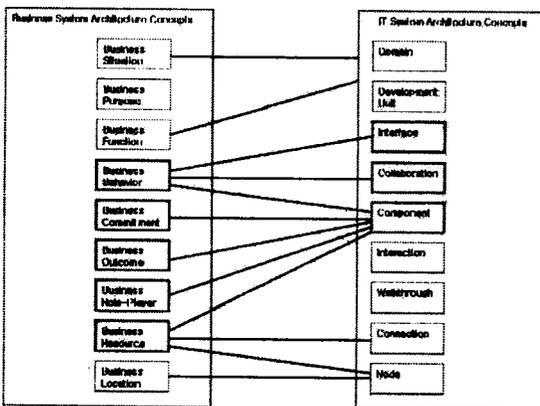


<그림 6> Business Concept과 Information System Concept의 매핑

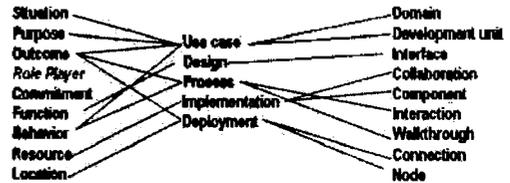
위와 같은 개념의 정립을 위하여 본 연구는 UML에서 시스템을 설명하는 5가지 뷰

를 도입하여 업무와 정보시스템의 차이를 줄이도록 한다. UML의 다섯 가지 뷰는 Use case 뷰는 시스템행동을, Design 뷰는 시스템의 기능적 요구사항을 주로 다루며, Process 뷰는 동적 측면, 스레드와 프로세스 중심의 정보와 관점을, Implementation 뷰는 하드웨어 상태, 그리고 마지막으로 Deployment 뷰는 하드웨어의 물리적 형태를 나타낸다.[7][21]

이는 UML에서의 각각의 뷰를 업무 관점과 시스템 관점에서 공통으로 적용하도록 하며, 이를 도출하는 과정은 아래의 그림과 같이 D. McDavid의 연구에서의 Business Architecture와 IT system의 매핑과 함께 UML에서의 5가지 시스템 뷰를 통하여 다양한 필요 정보가 전체 업무 아키텍처와 시스템 아키텍처 사이에서 표현 가능한 요구사항이 도출 되도록 하기 위하여 모든 정보를 UML의 5가지 뷰에 매핑시키는 단계를 거쳤다.

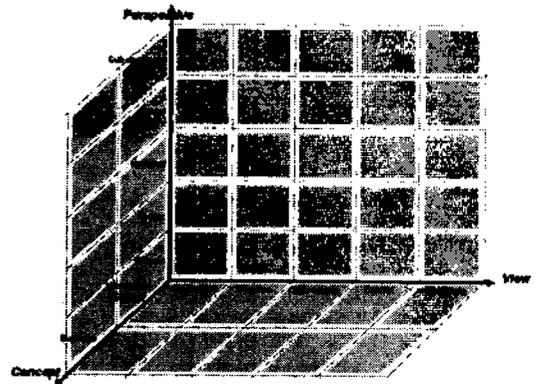


<그림 7> Architecture Concept과 IT System Concept 매핑[9]



<그림 8> UML view와 Business/Information system Concept 매핑

위 그림과 같은 매핑을 통하여 다음 절에서 제안하고자 하는 요구사항 분석 프레임워크의 완결성을 뒷받침하고, 전체 시스템 생명주기 상에서 발생하는 모든 요구사항을 위와 같은 개념으로 표현한다. 지금까지 요구사항의 관리와 기업 아키텍처의 관리는 시스템적인 측면 혹은 업무에 관한 비즈니스 아키텍처에 한정되어 있었으며, 이러한 한계점을 극복하고자 본 연구에서는 아래 그림과 같은 요구사항 분석 프레임워크를 제안하고자 한다.



<그림 9> 기업 아키텍처를 지원하는 요구사항 분석 프레임워크

그림의 분석 프레임 워은 크게 분석하고자 하는 영역, 모델 등을 구분하는 View, 분석하고자 하는 참여자의 관점을 구분하기 위한 Perspective, 그리고 정보시스템 아키텍처 또는 업무 아키텍처의 개념적인 구조를 정의할 수 있도록 하는 Concept의 세가지 차원을 정의 하였다.

프레임 워은 실제로 요구사항을 추출, 관리하기 위한 하나의 틀로써 기능을 수행하게 되며, 실제로는 평면의 구조로 요구사항을 분류하게 되고, 구분된 요구사항은 요구사항 세부항목을 정보시스템 항목에 따라 정의하고, 이를 각 세부 요구사항에 맞추어 기술참조모과 표준화 정도를 기입함에 따라 기업 아키텍처를 지원하게 된다.

Perspective는 기업 아키텍처 수립에 참여할 수 있는 다양한 참여자의 이해관계에 따른 관점의 차이를 극복할 수 있도록 Planner, Owner, Designer, Builder, Sub-contractor로 구분하였으며, 각 관점에서 시스템 또는 아키텍처를 개발하는데 있어서의 역할, 관점 등을 정의하여 아래 표에 정리하였다. 이는 Zachman의 연구에서 정의한 것을 수정, 재인용하여 정의하였으며, Zachman의 연구에서 정의한 관점을 기업 아키텍처에 맞게 수정한 것이다.[24][25] 이는 본 연구에서 제안하는 프레임 워의 목적상 Zachman의 연구에서 다루고 있는 정보시스템 개발의 참여자의 성격과 많은 유사함을 가지고 있기 때문이다.

또한, View는 다양한 요구 분석을 목적으로 하는 본 연구의 프레임 워의 특성을 위하여 다양한 모델이 모두 포함되도록 정보, 기능, 조직, 제어, 자원 모델로 나누어 모델

<표 3> Perspective 상세분류

Information strategist, information strategy planner, system architect	계획자는 정보화 전략 수립자, 또는 project manager 등의 관점을 가지고 기업 아키텍처의 전체 계획과의 조율이 가능하며, 아키텍처의 기본 원칙과 아키텍처를 정의한다. 또한 예산, 일정 계획, 투자타당성, 위험성 등을 평가할 수 있는 관점을 가진다.
CIO, CTO, end-user	수립되는 시스템, 아키텍처를 사용하며 어떻게 그들의 필요에 맞는지를 평가한다. 또한 시스템, 아키텍처의 성능, 신뢰성, 상호운용성 등을 검증한다.
business analyst, system architect, database administrator, network administrator	각 시스템, 아키텍처의 요구사항에 따라 알맞은 기술을 선택하고, 만들어질 시스템 아키텍처의 실재를 담당한다. 시스템 아키텍처의 정확성, 완전성, 일관성, 결합력 등을 평가한다.
developer, programmer	설계된 시스템, 아키텍처의 설계 정보에 따라 시스템 또는 아키텍처를 구축한다. 기존의 시스템과의 상호운용성, 유지보수, 기술적 세부 가이드라인을 구축한다.
system maintainer, other subcontractor	다양한 목적으로 참여하는 아웃소싱 업체 등이 있을 수 있다. 시스템, 아키텍처의 유지보수, 하드웨어적 지원, 소프트웨어의 수정 등으로 한정된다.

을 분류하고 이 모델에 대한 요구사항을 나누어 구분한다. 각 view는 요구사항 분석 이후의 시스템 또는 아키텍처의 분석, 설계시에 이용되는 모델의 요구사항을 추출한다. 기능 모델은 아키텍처의 기능을 나타내는 것으로 각 관점에 따른 시스템의 수행 기능에 초점을 맞춘 모델들의 요구사항이, 정보 모델은 다양한 관점에서 어떠한 데이터가 아키텍처를 구성하는지를 나타낸다. 조직 모델은 조직 구성원에 대한 정보와 위치, 역할 등의 모델을 나타내며, 제어 모델은 프로세스, 워크 플로우 등 프로세스에 해당하는 정보를 포함하며, 자원 모델은 기

업내 정보시스템과 관련된 모든 자원에 대한 정보에 대한 모델을 포함한다. 아래 표는 표현 가능한 모델에 대한 분류이며, 모델에 언급되지 않은 모델이라 해도 정의에 부합하는 모델은 포함될 수 있다.

<표 4> View 상세분류

data flow, entity-relationship model, class diagram, database schema, data cluster, etc.
function tree, IDEF0, use case model, etc.
organization chart, decision chart, location diagram, organization def., etc.
process diagram, IDEF3, petri-net, sequence diagram, state transition, collaboration diagram, event process control, process chain, work flow, etc.
Hardware spec., software spec., network spec., network topology, deployment diagram, etc.

이렇게 정의한 요구사항 분석 프레임 워크에 대한 정의에 맞추어 평면 구조의 요구사항

분석 프레임 워크의 사분면내의 각각의 셀 안에 아래 표와 같은 항목 세부 구조에 따른 세부 요구사항 항목을 정의한다.

사용자는 요구사항의 세부 항목은 앞서 정의한 Perspective와 View에 따라 구분하고, Business Concept을 요구사항을 식별한 후 이에 맞는 Semi-spec을 정하게 된다. 이는 필수적인 선택이 아닌 선택적 사양으로 일치하는 요구사항이 없는 경우에는 생략할 수도 있다. 또한 요구사항에 대한 TRM/SP에 대한 Adaptability를 기입하고, 우선순위와 사분면 셀 안의 요구사항 또는 다른 셀에 해당하는 요구사항과 연결하여 서로에게 참조가 될 수 있도록 Traceability를 설정하게 된다.

예를 들어, 사용자가 계획자이며 정보모델에 대한 경우에, Use case Business Concept을 선택하고 요구사항을 입력한다고 가정하면, 지식관리 시스템 프로젝트 진행 중에 이에 대한 요구사항 중 지식관리 시스템의 목적이 사내 인트라넷과 상호 연

<표 5> 요구사항 세부 항목 구조

Planner Perspective : Information View					ver 1.1			
Business Concept	Semi-spec	Specification	TRM	SP	Priority	Update	Traceability	
Use Case (Situation, Purpose, Outcome, Behavior, Role player)	Use case	Requirements...				Data...		
	Process							
	Design							
	Implementation							
	Deployment							
Design (Commitment, Function)	Use case							
	Process							
	Implementation							
	Deployment							
Process (Outcome, Behavior)	Use case							
	Process							
	...							

<표 6> 요구사항 분석 프로세스 세부 정의

<ul style="list-style-type: none"> <li>- 프로젝트 영역에 대한 식별</li> <li>- 어떤 시스템 또는 문제영역에 대한 분석 단계</li> </ul>	<ul style="list-style-type: none"> <li>- 기존 시스템 문서</li> <li>- 업무 영역 문서</li> </ul>	<ul style="list-style-type: none"> <li>- 프로젝트 정의</li> </ul>
<ul style="list-style-type: none"> <li>- 참여 프로젝트 인원에 대한 정의와 분석 모델 선정</li> <li>- 모델의 표현 방법 정의</li> </ul>	<ul style="list-style-type: none"> <li>- 프로젝트 정의</li> <li>- 인원구성</li> </ul>	<ul style="list-style-type: none"> <li>- 사용자 정의, 권한설정</li> <li>- 모델 표현 정의</li> </ul>
<ul style="list-style-type: none"> <li>- 사용자마다의 업무, 비즈니스 요구사항에 대한 식별</li> <li>- 요구사항 간의 상충 고려</li> </ul>	<ul style="list-style-type: none"> <li>- 기존 시스템 업무 문서</li> <li>- 모델표현 정의</li> </ul>	<ul style="list-style-type: none"> <li>- Semi-spec 정의</li> </ul>
<ul style="list-style-type: none"> <li>- Semi-spec 요구사항 간의 상충 분석</li> <li>- 사용자 Perspective &amp; View 에 대한 일관성 유지</li> </ul>	<ul style="list-style-type: none"> <li>- 기존 시스템 업무 문서</li> <li>- 프로젝트 정의</li> <li>- 사용자정의, 권한</li> </ul>	<ul style="list-style-type: none"> <li>- Re-vision 프로젝트 정의</li> <li>- Re-vision 사용자 정의</li> </ul>
<ul style="list-style-type: none"> <li>- 다양한 정보를 이용한 요구사항 추출</li> <li>- 요구사항 우선순위 도출</li> <li>- 요구사항간 Traceability 설정</li> </ul>	<ul style="list-style-type: none"> <li>- 기존 시스템, 업무 문서</li> <li>- 프로젝트 정의</li> <li>- 요구사항 세부항목</li> </ul>	<ul style="list-style-type: none"> <li>- 요구사항 세부항목</li> <li>- 요구사항 traceability</li> </ul>
<ul style="list-style-type: none"> <li>- 도출된 요구사항의 TRM/SP adaptability 설정</li> <li>- 요구사항마다의 Capa. 설정</li> </ul>	<ul style="list-style-type: none"> <li>- 요구사항 세부항목</li> <li>- 요구사항 traceability</li> </ul>	<ul style="list-style-type: none"> <li>- TRM/SP adaptability</li> </ul>
<ul style="list-style-type: none"> <li>- 추출된 요구사항의 검증</li> </ul>	<ul style="list-style-type: none"> <li>- 요구사항 세부항목</li> <li>- 요구사항 traceability</li> <li>- TRM/SP adaptability</li> </ul>	<ul style="list-style-type: none"> <li>- 요구사항 verification report</li> </ul>

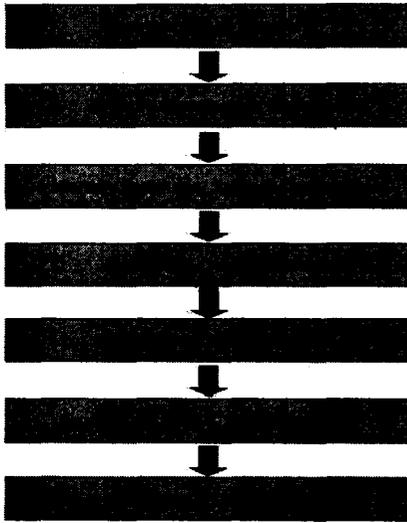
동성이 보장이 되어야 한다는 요구사항이 있다면, 이에 대한 semi-spec이 use case인 경우는 상호 연동성이 보장되어야 한다는 목적과 개발되어야 하는 영역을, design인 경우에는 인트라넷의 현재 개발된 부분이 어떻게 상호연동을 지원하는지에 대한 요구사항이 deployment에는 상호연동성의 보안상의 요구사항이 식별되게 지원하는 것이다. 이렇게 정의한 후에는 TRM/SP Adaptability와 우선순위를 선정하고, 요구사항이 re-vision될 때마다 update 시간, 작성자를 기록하며, 마지막으로 요구사항 사이의 참조된 요구사항을 정의하는 traceability를 작성한다

### 3.2 요구사항 분석 프로세스

본 연구에서는 앞서 정의한 요구사항 세부구조와 함께 아래 그림과 같은 요구사항 분석 프로세스를 제안하는데, Ian Sommerville의 연구와 앞서 살펴본 요구사항 관리 프로세스의 문제점을 보완하기 위하여 다양한 관점의 요구사항을 모두 수용하며, 이에 대한 기업 아키텍처의 지원성을 식별하기 위하여 TRM/SP Adaptability와 요구사항의 검증 과정을 거치게 된다.

요구사항 분석 프로세스는 각 task마다 입력물, 산출물을 정의하였으며, 이는 기존 연구에서 부족하였던 요구사항 분석 과정에

참여하는 사용자의 관점을 구조적으로 구분하는데 도움을 줄 것이다. 아래 표는 요구사항 분석 프로세스의 task의 정의와 입력물, 산출물을 정리한 것이다.



<그림 10> 요구사항 분석 프로세스

요구사항 분석 프로세스에서 TRM/SP Adaptability를 정의하기 위해서 한국전산원에서 정의한 TRM 서비스 분류를 이용한다. 이 분류는 미국방성의 ITSG (Information Technology Standards Guidance), 미에너지성의 채택된 표준 프로파일, IEEE OSE(Open System Environment) 참조모델을 정리한 것으로, 정보서비스를 응용 서비스, 데이터 교환 서비스, 데이터 관리 서비스, 플랫폼 서비스, 통신 서비스, 보안 서비스, 관리 서비스 등으로 분류하였다.[2] 또한, 표준프로파일은 기술참조모델에 명시된 서비스를 지원하는 정보기술 표준들의 집합으로 정

의한다. 이는 7개 분야에서 국내의 10개 표준화 기구의 8000여개 표준을 정리한 것으로 이를 정리하면 아래와 같다.

<표 7> 표준 프로파일 참여기관 분류 [2]

기관	IEEE	IECF	ISO	JTCS	KTTC	ONIS	Open G	NSC	KS	TTA	기타	계
횟수	76	2320	431	889	2063	16	64	64	473	479	336	7714

\* 기타 : ANSI, ARM Forum, ECMA, France Relay, ITU-R, ITX Consortium, NATO, NIST, NSC, ITSC 등

기관	음성	데이터교환	데이터관리	플랫폼	통신	관리	보안	기
횟수	140	304	125	382	940	204	363	7787

이렇게 정리된 표준 프로파일을 요구사항 분석 프로세스 중 기업 아키텍처의 지원성을 평가하기 위하여 TRM/SP Adaptability를 아래 표와 같이 정의하여 평가한다. 아래 표는 기술 참조 모델과 표준 프로파일의 적용성, 유연성의 정도를 나타내는 것으로, 향후 요구사항의 추적성과 관리 용이성에 도움을 준다. TRM/SP Adaptability를 평가하기 위해서는 하나의 요구사항에 대하여, 정보서비스에 따라 시간, 규모에 따른 상증하로 표기하고, 각 요구사항에 표준명(표준번호, 표준명, 제정연도 등), 설명(표준에 대한 간단한 설명 및 적용 범위), 선정이유(표준을 선택하게 된 이유), 효과(표준 적용시 장점 및 효과), 기타(제한사항, 적용시 고려사항 등) 등의 표준 관련 정보를 표시한다 또한 요구사항의 Capa.를 표시함으로써 요구사항 사이의 Traceability를 설정하는데 도움을 준다.

본 연구에서는 앞서 정의한 요구사항 분석 프로세스 중 Verification & Validation 단계의 가이드 라인을 제공하기 위하여 IEEE에서 제안하고 있는 소프

트웨어 요구사항의 구조를 받아들여 가이드라인을 작성한다.[16] 이는 요구사항의 가이드라인이라는 것의 특성이 비기능적 성질이 많은 부분을 차지하고 있기 때문에 실제 소프트웨어로 만들어지는 특성을 반영하고자 함이다.

<표 8> TRM & SP Adaptability 항목 구조

Service	Tating	Scap	Standard Profile				Quality		
			Standard	Description	Release	Bandwidth	Type	Frequency	Use
Application							20%	20%	CRUD
Dataexchange	● 상						20%	20%	CRUD
Datamanagement	⊗ 중						20%	20%	CRUD
Platform	○ 하								
Network									
Security									
Management									

먼저 IEEE에서는 소프트웨어 요구사항에 대한 평가 영역은 기존의 소프트웨어 요구사항의 기능적 요소와 비기능적 요소로 평가하던 것에서, 더욱 범용적인 요구사항의 평가를 위하여 좋은 요구사항이 가져야 하는 성질로 분류하였다. 이러한 요소들에 대하여 Verification & Validation 단계의 가이드 라인을 아래 표와 같이 도출하였다. 이는 비기능적 요구사항에 대한 검증에 포함하고 있으므로 실제 세부 요소 중에 정량적 요소로 포함되지 못하는 요소는 사용자가 평가할 수 있도록 가이드 라인을 제공하며, 이렇게 제공되는 요구사항의 Verification & Validation 단계의 가이드 라인은 요구사항 분석 프로세스가 적용되면서 프로젝트가 진행 중이거나 향후 프로젝트의 종료 이후 요구사항의 관리에 활용되면 효율이

더 높아질 수 있다.

<표 9> Requirement Verification & Validation 가이드 라인

- 사용자의 실제 필요성을 요구사항이 정확하게 반영하고 있는지에 대한 정보
- 기존 요구사항과의 비교, 시스템 요구사항과의 비교, 또는 기존 프로젝트와의 비교로 작성
- 해당 TRM/SP의 적용사항, 제한사항과의 gap 분석으로 만족도 도출
- < Description, Satisfaction % >
- 요구사항의 모호성에 대한 확인 정보
- 요구사항에 사용된 용어가 용어사전에서 등록되어 있는지, 같은 용어인데 다른 사용자가 다른 의미로 쓰인 것은 없는지에 대해 평가
- 용어에 대한 정정 횟수를 척도로 이용
- < # of adjust requirement >
- 요구사항이 요구사항으로서 완전한지에 대한 정보
- 요구사항에 대하여 모든 참여자의 요구사항이 포함되었는지, TRM/SP가 작성되었는지, priority/update/traceability 완전하게 작성되었는지 확인한다.
- < # of missing information >
- 요구사항의 내부적인 일관성에 대한 정보
- 같은 프로젝트 내부의 연결된 요구사항들의 우선순위에 대한 중복, 실수 횟수
- < illegal 'priority', illegal 'traceability' >
- 요구사항의 안정성을 나타내는 것으로, 프로젝트 진행 중에 요구사항의 변경횟수를 확인한다.
- < # of changes to any requirement >
- 요구사항의 필요성에 대한 정보로, 우선순위에 대한 검증절차로 요구사항을 세가지 필요성으로 분류
- < Essential /Conditional / Optional >
- 요구사항의 검증이 가능한 지에 대한 정보
- 요구사항의 비용-효과 분석에 대한 정보로 실제로 측정가능한 정량적 정보와 측정 source를 언급
- < measurable information, information sources >
- 요구사항이 수정되었을 경우, 구조의 종류에 대한 정보
- 연결된 요구사항의 구조가 계층적인 경우 하위 요구사항과 상위 구조에 대한 정보
- < Description >

## 4. 사례 적용 및 활용방안

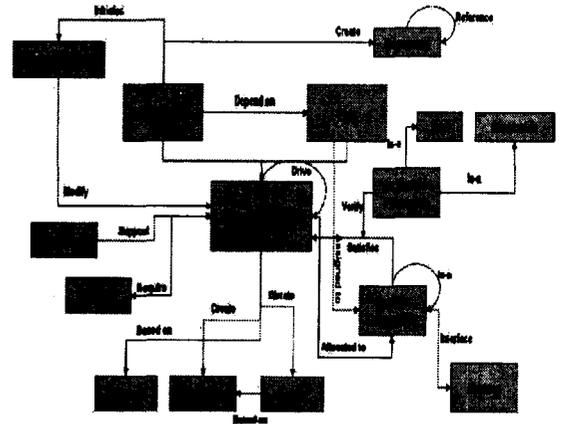
### 4.1 요구사항 관리도구

본 논문에서는 개발에 협력하고 참여할 수 있도록 관련된 참여자들로부터 발생하는 요구사항의 추출 및 관리를 지원하는 요구사항 관리도구를 제시한다. 기존의 대부분의 요구사항 관리도구들이 요구사항 분석 방법론에 의거한 분석지원도구들로 다양한 요구사항들을 효과적으로 지원하지 못하는 단점을 극복하기 위하여, 본 논문에서 제시된 요구공학 프로세스의 단계를 이용하여 시스템 개발에 참여하는 여러 그룹들의 의견이나 요구사항들이 효과적으로 반영되어 관리되도록 구현함으로써 생산성을 향상시키고 신속하고 적은 비용으로 소프트웨어를 개발할 수 있도록 한다.

요구사항 관리 도구는 사용의 용이성을 갖추고, 분산된 개발집단의 다양한 요구를 충족하기 위하여 요구사항들을 분산 처리할 수 있어야 하며, 또한 변경요청에 대한 동기화 기능도 제공하여야 한다. 이러한 도구가 기본적으로 갖추어야 할 기능은 요구사항 추출, 구성 등의 획득기능과 변경수행, 추적관리, 변경 영향분석, 연계성 관리와 같은 변경관리 기능, 형상관리 및 사용자 관리 기능 등이다.

요구사항 관리도구는 요구사항 사이의 연계성을 위해서 각 요구사항의 내용을 기반으로 사용자가 연계성을 설정하는데, 이는 기존 시스템 또는 컴포넌트가 있는 경우에 기존 시스템의 요구사항을 현재

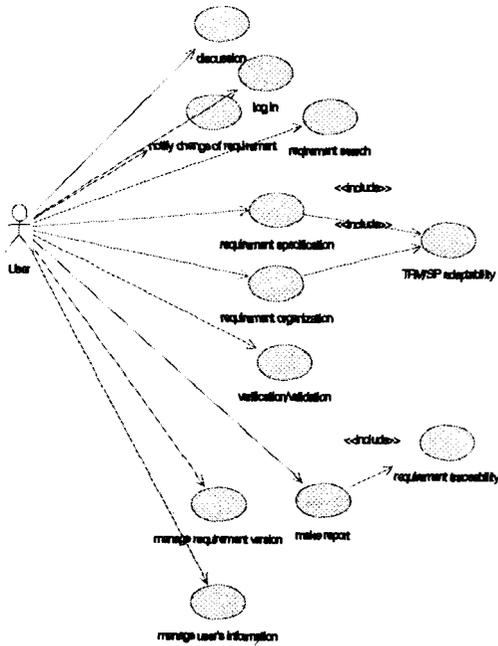
진행중인 프로젝트에 반영하기 위해서도 필요한 기능이다. 이를 위하여 본 연구에서는 아래 그림과 같은 개념적 수준의 연계성 모델을 제시한다.



<그림 11> 개념적 수준의 Traceability 모델

이는 요구사항의 변경 또는 삭제 등으로 인하여 요구사항의 내용이 변경되었을 때 그로 인하여 다른 요구사항에 미치는 영향을 분석하고 기술할 때 활용되며, 요구사항 관리도구의 프로토타이핑의 자료 구조에 기본적인 구조를 이루게 된다.

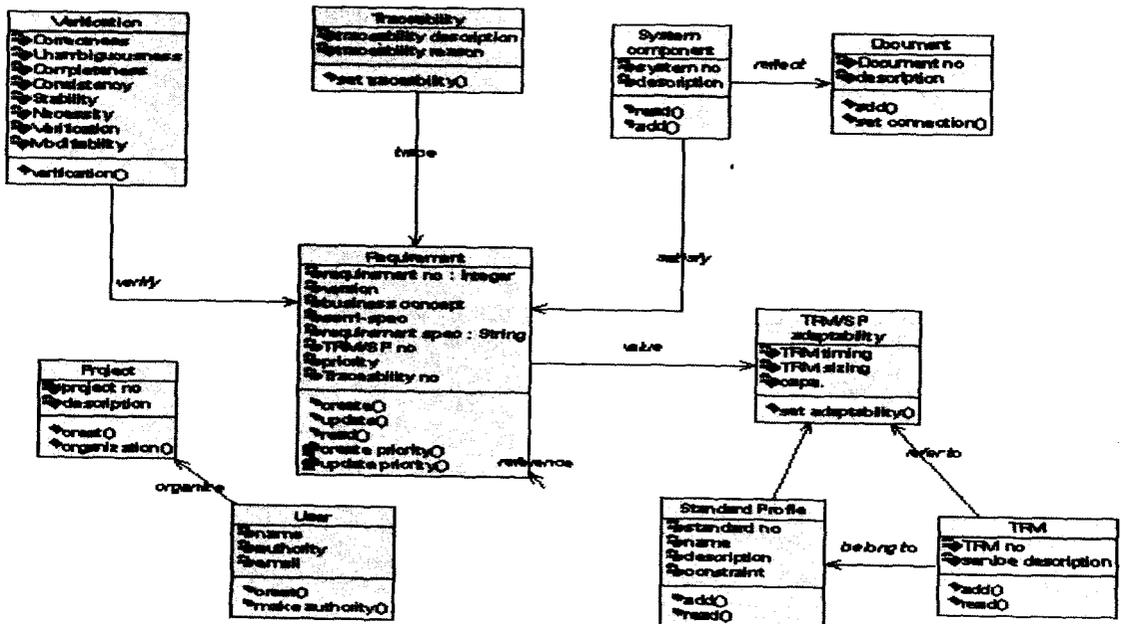
본 논문에서 제시하고 있는 요구사항 분석 프레임 워크 및 요구사항 분석 프로세스에 대하여 요구사항 관리도구는 앞서 정의한 기능을 기초로 하여, 먼저 use case 모델을 작성하였고, 이들 중 우선순위가 높은 use case에 대하여 sequence diagram을 거쳐 class 모델을 도출하였다. 아래 그림은 요구사항 관리도구의 use case 모델이다.



<그림 12> Use case 모델

이렇게 정의한 use case 중에서 기업 아키텍처를 지원하는 요구사항 관리도구의 핵심적인 요구사항 추출, 요구사항 재구성, TRM/SP설정, traceability 설정, 로그인 등의 use case에 대하여 프로토타이핑을 수행하였다. 이들 use case는 모두 RUP(Rational Unified Process, Version 2002. 5.)에 따른 산출물을 참조하여 작성하였으며, 이 중 요구사항 추출에 대한 sequence diagram은 아래와 같다.

이와 같은 단계를 거쳐 아래 그림과 같은 Class diagram을 도출하였으며, 이는 관리도구 프로토타이핑 전체 클래스를 표현한 것으로, 요구사항 관리도구 use case 또는 연계 모델에서 등장한 클래스가 모두 표현되지 않았음을 밝힌다.



<그림 13> Class diagram

4.2 사례적용 및 활용방안

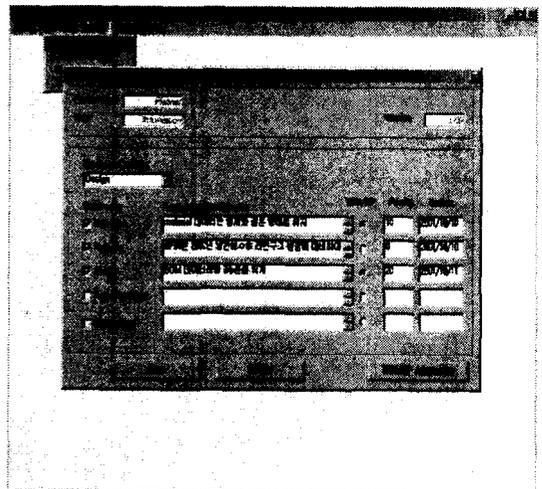
본 논문에서 제시한 관리도구를 이용한 UniERP의 생산 관리 모듈의 적용 사례를 통하여 요구사항 분석 프레임 워크 및 프로세스의 활용 방안을 모색하고자 한다.[1] 그러나 사례의 적용은 개발 프로세스 전체에서 활용된 결과를 바탕으로 결과가 도출되어야 하나, 투입된 모든 요구사항이 이미 개발이 끝난 시점에서의 요구사항을 본 연구에서 제시하는 요구사항 관리도구를 이용하여 재구성 하는 것으로 대신했다.

구성되는 요구사항은 한 컨설팅 업체가 A사에 UniERP 생산모듈을 도입하는 과정에서 기존에 작성된 요구사항으로 아래 표와 같은 세부 항목을 가지고 있으며, 요구사항의 추출은 요구사항이 정리되는 회의자료와 세 가지 단계에 따르는 문서를 기준으로 분류하였다. Perspective와 View 역시 작성자와 토의된 내용 중 특정 관점의 요구사항이 부각된 경우를 기준으로 분류하여 요구사항 관리도구에 적용하였다.

<표 10> 사례 적용 세부 항목

- 2개월	
- 컨설팅 사 (Planner 1명, Designer 4명, Builder 3명)	sub-contractor 역할은 있으나 요구사항 분석단계에서 참여하지 않음
- H사 (Planner 1명, Owner 5명)	
- General design - 7단계에 따른 주간 회의록	
- Customizing - Business rule 정의서, Configuration List	
- Training - Manual 작성지침	
- UniERP reference model	
- 데이터 입력 지침	

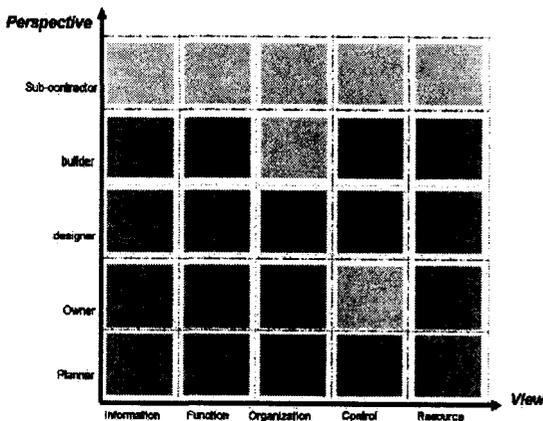
제시한 도구의 프로토타입을 적용하는 과정은 프로젝트 관리자가 프로젝트를 생성하고 다양한 Perspective를 가진 참여자를 사용자로 등록하여 권한을 설정하면, 사용자들은 해당 프로젝트를 선택하여 요구사항의 추출(생성), 재구성(수정), TRM/SP adaptability 설정, traceability 설정 등을 수행하고 최종적으로 수집된 요구사항에 대하여 verification을 수행한다. 이러한 사례의 적용은 실제 요구사항 분석 단계, 즉 프로젝트의 시작과 함께 관리되어야 하는 특성이 반영되지 못하였으나, 요구사항의 추출되는 과정과 요구사항 분석 프레임 워크, TRM/SP adaptability 설정, traceability 설정 등의 주요 과정을 거치면서 요구사항 분석 프로세스 자체에 대한 적용이 이루어졌다고 할 수 있다.



<그림 14> 요구사항 입력화면

본 논문에서 적용한 도구의 프로토타입에 대한 사례 적용에 대한 결과를 살펴보면 아래 그림과 같은 요구사항의 분류가 가능했

다. 이는 전체 요구사항 분석 프로세스를 거치지 못하여 verification 단계가 실행되지 않았지만 요구사항 분석 프레임 워크 제시하는 틀이 다양한 요구사항을 효과적으로 분류함을 보여주고 있으며, Perspective를 중심으로 볼 때 builder와 designer의 요구사항이 두드러지게 많았으며, View에서는 정보모델과 기능모델에 대한 요구사항이 높았다. 이러한 결과는 실제 사용자에게 해당하는 owner가 추출한 요구사항에 대하여 designer와 builder가 이를 세분화하거나, 실제 정보시스템으로 분석, 설계하는 과정에서 일어날 수 있는 시스템과 관련하여 파생되는 요구사항이 많음을 보여주고 있다.



<그림 15> 사례적용 결과  
- 추출된 요구사항 개수와 영역

또한, 정보 모델, 기능모델에 대한 요구사항이 높은 것은 적용된 사례의 특성상 기존에 만들어진 시스템의 구현을 다루고 있기 때문에 자원이라든가 조직적 특성이 이미 상당 부분 반영되어 있기 때문으로 볼 수 있다.

이에 본 연구에서 제안하는 요구사항 분석 프레임 워크는 요구사항 관리의 틀을 제공하는 것으로 활용 가능하며, 비즈니스 개념과 정보시스템 개념을 식별하는 과정은 요구사항의 추출에 대한 분석 방법을 제공한다. 또한 요구사항 분석 프로세스는 기업 아키텍처의 TRM/SP의 특성과 verification에 따른 가이드 라인을 제공하므로 요구사항 변경 관리 및 향후 설계되는 기업 아키텍처의 강건성을 확립하는 것으로 활용 가능하다.

### 5. 결론 및 활용방안

본 연구에서는 기존의 요구사항 관리가 문서의 생성에 관련되어 전체 개발 공정에 대한 해결책이 미흡하였던 문제를 해결하기 위하여 Enterprise Architecture의 개념을 도입하여 요구사항을 완전하게 관리할 수 있도록 함으로서, 기업 아키텍처의 설계 위험부담을 경감시키고, 정보시스템의 품질 향상을 이룰 수 있도록 요구공학 기반의 관리 프로세스와 안정적인 요구사항의 생성 및 변경에 대한 관리 기법 및 요구사항을 처리할 수 있는 관리 도구 등이 포함된 프레임 워크의 설계를 제시하였다. 이를 통하여 연구에 포함된 내용은 다음과 같다.

- △ 도입 단계에 있는 Enterprise Architecture의 이해와 정보시스템 생명주기 상의 다양한 참여자와 관점의 이해
- △ Enterprise Architecture 설계를 지원하는 요구사항 생성 및 변경에 대한 관리 기법 및 프레임 워크 제시

△ 요구사항 분석 프로세스의 정의로 체계적인 요구사항 관리에 의한 소프트웨어 개발 생산성의 향상

△ 요구사항 관리도구의 프로토타이핑과 사례 적용으로 활용 가능성 평가

본 연구에서는 기업 아키텍처의 개념을 도입하여 요구사항 분석 프레임 워크 프로세스를 정의하고 사례 적용을 하였으나, 요구사항 분석 단계의 특성상 프로젝트의 시작과 함께 지속적인 사례에 대한 관찰이 되고 있지 못한 것과 요구사항의 verification 과정이 적용되기 위해서도 필요한 과정을 완전히 적용하고 있지 못하다

그러므로, 향후 연구에서는 제시된 요구사항 분석 프레임 워크에 대해 실제 업무에 적용 및 사례분석을 통한 정량적 평가와 정형적 기법의 개발에 대한 연구가 수행되어야 할 것이다. 또한 제시된 관리 도구의 프로토타입을 근간으로 기존의 관리 도구와의 비교 분석 및 통합을 위한 연구가 필요하다.

## 참 고 문 헌

- [1] 삼성SDS, Uni-ERP Reference Model, 2001.
- [2] 한국전산원, 정보화 표준 프로파일, NCA IV-RER-99039, 1999. 12.
- [3] A. M. Davis, Software Requirements - Objects, Functions, & States, Englewood Cliffs, NJ Prentice Hall, 1993.
- [4] A. M. Davis, Specifying Behavioral Requirements - Software Requirements Objects, Functions, and States, University of Colorado at Colorado Springs, Prentice hall, 1993.
- [5] Bass, Clements, and Kazman, Software Architecture in Practice, Addison-Wesley, 1997.
- [6] Bernard H. Boar., Constructing Blueprints for Enterprise IT Architectures, John Wiley & Sons, 1999.
- [7] Booch, Rumbaugh, and Jacobson, The UML Modeling Language User Guide, Addison-Wesley, 1999.
- [8] Clinger-Cohen Act., National Defense Authorization Act for Fiscal Year 1996, Division E-information Technology Management Reform Act of 1996., Public Law 104-106(S. 1124)
- [9] D. McDavid, A Standard for business architecture description, IBM System Journal, Vol 38 No 1, 1999.
- [10] DoD, Technical Architecture Framework for information Management(TAFIM), Vol. 4 : DoD Standards-Based Architecture Planning Guide Ver. 3.0, April,1996.
- [11] Eberhardt Rechtin, System Architecting: Creating and building complex systems, Prentice-Hall, 1991.
- [12] F.B. Vernadat, Enterprise Modeling and Integration: principle and applications, 1996.
- [13] Federal CIO Council, Federal Information Technology Architecture

- Conceptual Model, Draft, June 1998.
- [14] G. Kotonya, Ian Sommerville, Requirements Engineering, John Wiley & Sons Inc, 1998.
- [15] IEEE Std 100-1996, IEEE Standard Dictionary of Electrical and Electronics Terms, Sixth Edition, The Institute of Electrical and Electronics Engineers, 1996.
- [16] IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications, 1998
- [17] International Council of Systems Engineers(INCOSE), Systems Architecture Working Group, Systems Architecture Working Group Definitions, August 1998.
- [18] K. Pohl, Process-Centered Requirements Engineering, John Wiley & Sons Inc, 1996.
- [19] Mehdi Jazayeri, Alexander Ran, Frank van der Linden., Software Architecture for Product Families: Principles and Practice, Addison Wesley Longman, 2000.
- [20] OMB Memorandum M97-16, Information Technology Architecture, 1997.
- [21] Philippe B. Kruchten, The 4+1 View Model of Architecture, IEEE Software, Vol 12, No. 6, 1995.
- [22] Robertson, S., Robertson, J., Mastering the requirements process, ADDISON-WESLEY, 1999.
- [23] Rzepka, William. E, A Requirement Engineering Testbed: Concepts, Status, and First Results, In Proceeding of the Twenty-Second Annual Hawaii International Conference on System Sciences, IEEE Computer Society, 1989.
- [24] Zachman, J.A., A Framework for Information Systems Architecture. IBM Systems Journal, Vol. 26, No. 3, 1987, 276-292
- [25] <http://www.zifa.com/>
- [26] <http://www.opengroup.org/>

## 저자 소개

### 최봉균

연세대학교 컴퓨터·산업시스템공학과 석사과정

연세대학교 지식정보화센터 보조연구원

한성대학교 산업시스템공학과

관심분야: 비즈니스 프로세스 표준화, 비즈니스 컴포넌트 아키텍처

### 임춘성

연세대학교 컴퓨터·산업시스템공학과 부교수

(사)기업정보화지원센터장

미국 Rutgers University 산업공학 조교수

미국 University of California at Berkeley, ESRC 방문교수

미국 University of California at Berkeley 산업공학 박사

서울대학교 산업공학과 석사

서울대학교 산업공학과 학사

관심분야: 기업정보화컨설팅 방법론, 기업가치평가, 전자상거래, 지식경영시스템,  
인터넷 비즈니스