

# 도메인 기반 모델링과 구조 유니트를 이용한 기업 구조 프레임워크의 설계방법

## Design of Enterprise Architectures Framework using Architecture Unit and Domain Specific Method

채희권(Heekwon Chae)\*, 김광수(Kwangsoo Kim)\*, 김철한(Cheolhan Kim)\*\* †,  
최영환(Youngwan Choi)\*\*\*

### 초 록

엔터프라이스 아키텍처 (EA) 프레임워크는 기업의 정보기술 요소들 사이의 상호 운영성을 증가시키기 위해 활용되는 EA를 구축하기 위한 도구이다. 본 논문은 엔터프라이스 아키텍처 유닛(AU), 참조모델, 그리고 도메인 모델 사이의 연관관계를 조합하여 정의한 ENAF (ENterprise Architecture Framework)이라는 엔터프라이스 아키텍처 프레임워크의 설계 기법을 정의한 것이다. 이 기법에서 정의하는 AU는 하나의 비즈니스 프로세스와 이를 지원하는 시스템 및 기술요소들을 연관관계를 포함하여 조합한 최소단위의 EA로서, 서로 연관이 있는 AU들의 조합을 통하여 EA의 구축이 가능하도록 하였다. UML과 같이 범용성있는 모델링 언어의 사용은 비즈니스 도메인을 표현하는데 한계가 있으므로, 비즈니스 도메인의 기술에 적합한 DSM의 개념을 도입하였으며, DSM으로 정의된 AU 들은 토픽 맵을 이용하여 EA를 생성할 수 있다. 논문의 2장은 EA 관련 연구 및 DSM, 토픽 맵에 대하여 기술하였으며, 3장은 본 연구에서 제안하는 EA 프레임워크의 설계 방법에 대하여 설명하였으며, 4장에서는 본 설계 기법을 적용한 사례에 대하여 설명하였다.

### ABSTRACT

An Enterprise Architecture (EA) Framework is a tool which supports implementation of the Enterprise architecture that is used to enhance the interoperability of the IT components. In this paper, we propose a framework named as ENAF (ENterprise Architecture Framework) which combines enterprise architecture unit (AU), reference model, and association relationship between domain model. Architecture Unit is defined as a minimum set of a business process and its associated components such as application system and technical components. An EA can be designed and implemented by the aggregating the related AUs including association relationship between Architecture Units. Because UML model has limitations to describe business domain semantics because it is designed for general purpose, we adapt the DSM (Domain Specific Modeling) concept. We describe association relationship between Architecture Units designed by Domain Specific Modeling through Topic Map.

Session 2 describes related works about Enterprise Architecture frameworks, Domain Specific Modeling, and Topic Map, while Session 3 explains components of the ENAF. Finally Session 4 shows the case study for implementation of the new Framework called ENAF.

키워드 : 기업구조, 기업 구조프레임워크, 도메인 기반 모델링, 토픽 맵  
Enterprise Architecture, Enterprise Architecture Framework,  
Domain Specific Modeling, and Topic Map

\* 포항공과대학교 산업공학과

\*\* 대전대학교 인터넷정보공학과, 교신저자

\*\*\* 서울대학교 컴퓨터연구소 e-비즈니스 기술연구센터

## 1. 서론

EA(Enterprise Architecture)는 기업 내의 업무 및 관리 프로세스와 정보기술 간의 현재 및 목표관계에 대한 명확한 설명과 문서[1]로서 정보 전략을 위한 기본적인 기술 및 프로세스 구조를 제공한다[2]. EA 프레임워크는 EA 구축을 지원하는 도구로서 EA가 다루어야 할 항목, 즉 비즈니스 애플리케이션, 정보 기술, 그리고 정보 등을 대상으로 기업의 비즈니스에 대한 IT활용측면에서의 기업 구조를 정의하기 위하여 사용된다. EA를 구축하는 방법에는 참조모델을 이용하는 방법 [2]과 Template을 이용하는 방법[1]으로 나뉜다. 전자의 경우에는 참조모델을 구성하고 있는 요소들 간의 연관관계를 파악하는데 어려움이 있으며, 후자의 경우에는 특정 도메인에 표현하는 템플릿보다는 일반화된 템플릿을 적용하기 때문에, 특정 기업에 해당하는 EA를 구축하는데 어려움이 있다. 따라서 본 연구에서는 이러한 두 가지 문제를 해결하는 방안으로 두 가지 방안을 혼합하는 EA 구축 방안 즉, 참조모델과 도메인 모델들 간의 연관관계를 포함하고 이들을 이용하여 EA를 개발하는 ENAF를 제시하고자 한다. 연관관계 모델은 EA에서 사용되는 이들 간의 연관 관계를 정의하고 모아놓은 것으로서, 본 논문에서는 사용한 연관관계는 Henk[3]등이 정의한 연관관계의 개념을 이용하였다. AU(Architecture Unit)는 하나의 비즈니스 프로세스와 이를 지원하는 시스템, 기술 요소들 간의 연관관계를 포함하여 조합한 최소단위의 EA로 AU를 조합함으로써 EA의 생성이

가능하다. AU 구성요소들 간의 연관관계를 포함함으로써, 연관관계의 고려 없이 EA의 생성이 가능하게 하며 구조의 일관성을 유지해준다.

EA 모델에 UML과 같이 도메인을 고려하지 않는 모델링 언어의 사용은 특정 도메인에 한정된 개념을 빠르고 정확하게 표현하는데 한계가 있다. 이를 해결하기 위하여 본 연구에서는 DSM(Domain Specific Modeling)[4]을 이용하여 EA로 구현하였다. Domain Specific Model로 정의된 AU들은 토픽맵(Topic Map)을 이용하여 EA내의 각 관점들의 연관관계를 표현에 이용하고, "merging" 과정을 이용하여 EA의 빠른 생성과 관리를 가능하게 한다. 개발된 방법론을 검증하기 위하여 공급망(supply chain)을 위한 EA 모델의 Domain Specific Meta Model을 정의하였고, 이를 바탕으로 비즈니스 프로세스 참조 모델인 SCOR[5]모델을 참조하여 DSM인 EA 참조 모델, 연관모델 그리고 AU를 생성 하였다. 그리고 생성된 EA모델들을 기반으로 국제 표준인 [ISO13250] 토픽 맵[6]형태로 EA들을 구현하였다.

본 논문의 2장에서는 널리 사용되고 있는 EA 프레임워크들의 장단점에 대한 분석과 이를 보완하기 위하여 제시한 DSM과 토픽맵에 관련된 연구를 소개하였고, 3장에서는 ENAF의 구성요소에 대하여 언급하였으며, 4장에서는 ENAF 내의 AU와 관계모델을 공급망의 EA를 위한 DSM로 표현하였으며, 5장에서는 AU를 토픽맵으로 구현한 예를 설명하였다.

## 2. 관련연구

EA를 구축하는 방법에는 FEAF와 같이 참조모델을 이용하는 방법과 Zachman 프레임워크와 같이 Template을 이용하는 방법으로 나뉜다. EA 프레임워크들 중 FEAF는 보다 체계적이고 정확한 EA의 생성을 위하여 참조모델을 EA생성의 도구로서 제공한다. 참조모델 방법은 EA 프레임워크를 기초로 하여 관점 별로 참조 모델을 만들고 이를 이용하여 EA를 생성하는 방법이다. 참조모델 방법은 참조모델내의 요소들이 컴포넌트로 구성되어있기 때문에, 컴포넌트들 간의 연관관계를 고려하여 조합하면 EA가 생성된다. 참조모델 방법은 참조 모델을 이용하는 참가자들 간의 상호운용성이 확보되고, 컴포넌트의 특징인 모델의 재사용성(reusability)이 확보됨으로써 EA의 초기 모델 생성이 용이한 장점을 갖는다. 이러한 참조모델 방법은 EA의 생성에 많은 장점을 가지고 있으나, 참조모델에 EA의 생성 시에 필요한 도메인 모델들 간의 연관관계가 포함되지 않기 때문에, EA생성에 필요한 연관관계를 사용자가 직접 찾고 분석해야 하는 어려움이 있다.

이러한 도메인 모델들 간의 연관관계에 대한 연구는 Henk[3]등에 의하여 이루어 졌다. Henk[3]등은 EA의 영역을 비즈니스, 애플리케이션, 기술 레이어 (layer)로 표현하고, 레이어 내부와 외부의 연관관계를 나누어서 표현함으로써 EA내의 모든 연관관계를 표현하였다. 그러나 Henk[3]등의 연구는 EA내의 연관관계의 표현은 적절하나 참조모델 방법과 같이 컴포넌트에 의한 체계적이고 쉬운 EA

의 개발을 지원하지는 않는다.

### 2.1 Zachman 프레임워크

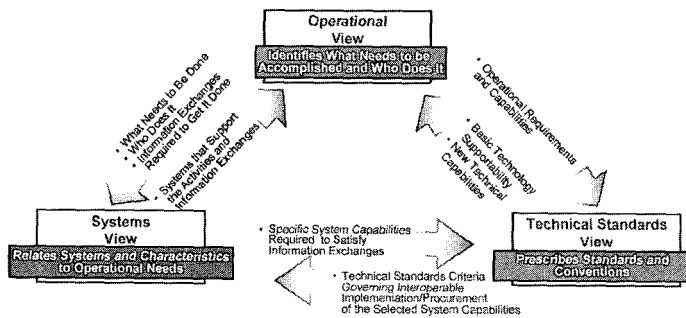
Zachman 프레임워크는 EA의 대표적인 프레임워크의 하나이며, 하나의 기업의 아키텍처를 이차원적으로 분류[7,8]하는 프레임워크다. <그림 1>과 같이 Zachman 프레임워크의 세로축은 관점(perspective)에 따라 scope (planner), business model (owner), system model (designer), technology model (builder), detailed representation (sub-contractor)로 분류하여 상위로 갈수록 조직의 일반적인 관점을 표현하며 하위로 갈수록 현실적인 관점을 표현한다. 가로축은 육하원칙에 따라 data (what), function (how), network (where), people (who), time (when), motivation (why)로 분류하였다. 이러한 방법으로 생성된 셀에 이 셀을 지원하는 방법들을 매핑하여 프레임워크를 정의하였다. 따라서 비즈니스 도메인에 따라서, 그리고 생성되는 셀의 특징에 따라서 다른 템플릿이 생성되므로, 이를 관리하는 문제가 발생하게 된다.

### 2.2 DoDAF(Department of Defense Architecture Framework)

DoDAF[9]는 미 국방성의 표준 EA 프레임워크며 <그림 2>와 같이 크게 3가지의 관점(view)로 구성되어 있다. 3가지 관점은 OV(Operational View), SV(System View), TV(Technical standards view)이며 이들 3가지 관점을 총괄하는 AV(All view)가 합쳐져

	DATA What?	FUNCTION How?	NETWORK Where?	PEOPLE Who?	TIME When?	VOLITION Why?
SCOPE (Planner)	List of things	List of process	List of locations	List of stakeholders and agents	List of business events	List of business goals and metrics
ENTERPRISE MODEL (Owner)	Conceptual model	Business process models	Business process system	Workflow model	Master schedule	Business plan
SYSTEM MODEL (Designer)	Concept Data model	Application architecture	Distributed system architecture	Human interfaces architecture	Producting schedule	Business case studies
TECHNOLOGY MODEL (Builder)	Concept Data model	System design	Technology architecture	Presentation architecture	Cost of structure	Real design
COMPONENTS (Vendor)	Data distribution	Product	Network architecture	Service architecture	Plans, schedules	Rule specifications
Functional System (Product)	Data	Function	Network	Organization	Schedule	Strategy

〈그림 1〉 Zachman Framework



〈그림 2〉 DoDAF 뷰들 간의 연관관계

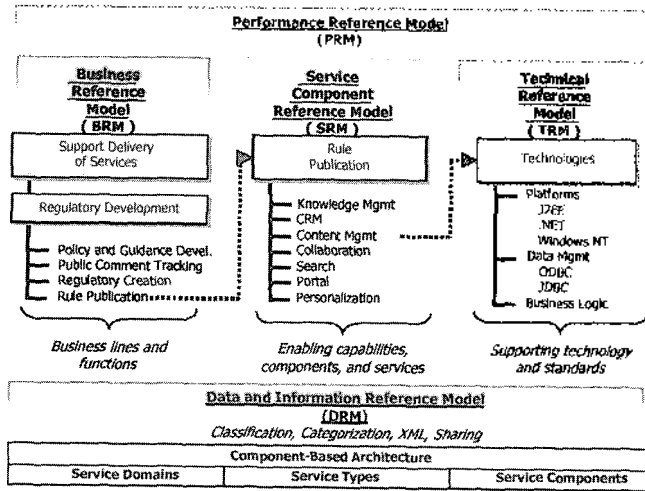
서 DoDAF를 구성한다. 각각의 뷰는 뷰를 상세하게 기술하는 여러 개의 product들로 구성된다. OV, SV, TV는 서로 관련을 가지고 있으며 EA를 구성할 경우 관점에 따라 각 관점에 속한 필요한 product를 조합하여 EA를 구성한다.

OV는 미 국방부의 목표를 달성하기 위한 업무와 활동, 조직요소, 그리고 정보교환을 기술하며, SV는 미 국방부의 업무를 위한 상호 관련 및 시스템을 기술하고 시스템 리소스를 OV에 연결시킨다. TV는 시스템을 실행하는

과정 있어서 기술적인 근거를 제공한다. AV는 OV, SV, TV를 지배하는 형태를 지니며, 전체적인 아키텍처에 필요한 정보를 제공한다. DoDAF는 각 관점들 간의 연관관계가 정의되어 있어서 EA생성을 보다 체계적이고 쉽게 한다.

### 2.3 FEF(Federal Enterprise Architecture Framework)

FEF[10]는 미 연방정부의 표준 EA 프레임



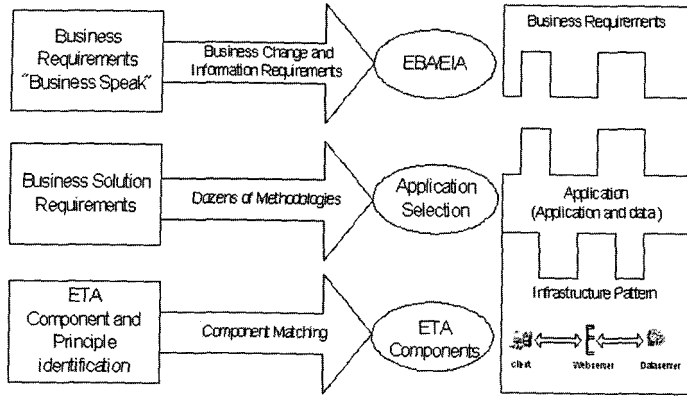
〈그림 3〉 FEA의 참조모델

임위크며 정부 내의 IT에 대한 중복투자를 방지하여 효율적인 투자를 이끌어 내며 IT의 재사용성을 높이기 위하여 미 연방정부가 재정한 EA 프레임워크다. FEA는 크게 BRM(Business Reference Model), SRM(Service Reference Model), TRM(Technical Reference Model), PRM(Performance Reference Model), DRM(Data Reference Model) 등 5가지의 참조모델로 구성되어 있다(그림3). BRM은 비즈니스 참조모델로서 FEA의 기본이 되는 모델이며 미 정부 내의 각 부서들의 공통적인 비즈니스들로 구성되어 있다. SRM은 컴포넌트를 기반의 모델로서 애플리케이션, 애플리케이션 성능, 컴포넌트, 그리고 비즈니스 서비스의 재사용을 가능하게 한다. TRM은 기술적인 요소들로 구성되어 있으며 CBA(Component Based Approach)를 지원한다. PRM은 IT의 성능 측정지표로서 활용되며, DRM은 FEA에서 사용되는 데이터와 정보

의 분류하기 위한 모델이다. 이러한 참조모델 방법을 이용하면 쉽게 EA를 생성할 수 있으나 각 참조모델 간의 연관관계가 표현되어 있지 않기 때문에 EA생성 시 연관관계를 찾아야 하는 불편함이 있다.

#### 2.4 ESA(Enterprise Solution Architecture)

ESA[11]는 IT투자에 대한 결정과 IT 솔루션(solution)의 지침을 공급하는 것을 목적으로 한다. ESA는 비즈니스 솔루션에 대한 요구에 반응하여 개발되며, EBA(enterprise business architecture), EIA(enterprise information architecture), 미래 지향적인 컴포넌트 모델인 ETA(enterprise technical architecture)과 연결되어 있다. ESA에서 솔루션은 그림4와 같이 비즈니스 요구사항이 발생하면 EBA와 EIA가 만들어지며, EBA와 EIA의 비즈니스 요구사항에 따라 비즈니스 솔루션에 대한 요



〈그림 4〉 ESA(enterprise solution architecture) 모델

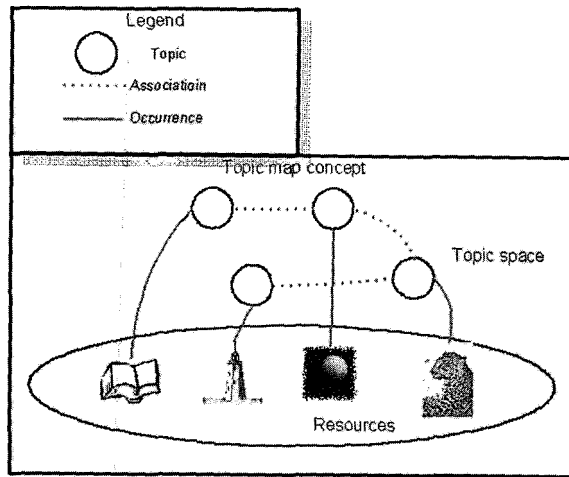
구가 발생하면 적절한 애플리케이션과 데이터, 그리고 ETA 컴포넌트가 선택된다. 완성된 솔루션 모델들은 기업 내의 각 조직에 따라 다를 수 있으며 EA 내에서 서로 연관관계가 표현된다.

ESA는 솔루션 안에 여러 가지 관점에 따른 모델이 합쳐져 있고 서로 연관관계를 가지고 있는 점에서 AU와 같다. 그러나 ESA는 애플

리케이션 중심의 방법론이며 EA 전체가 컴포넌트로 모델링 되지는 않는다.

### 2.5 Topic Maps

토픽맵[6, 12]은 주체(Subject)중심의 언어로서 Topic, Association, Occurrence를 기본으로 연관관계의 표현이 가능하며, 정보가 여러



〈그림 5〉 토픽 맵의 구조

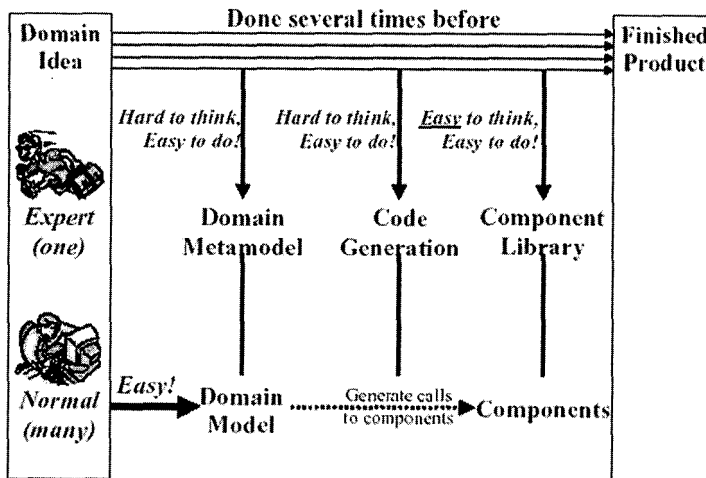
가지 관점에서 표현되어 정보 검색에 유용하다. 토픽맵은 ISO 13250에 정의 되어 있고, 오늘날 대부분은 XTM 1.0 [6]에서 정의된 문법을 사용한다. 토픽맵의 기본 구조는 그림5와 같으며, 적용분야는 (1)정보의 접근성 향상 (2)유연하고 확장 가능한 구조 (3)기존의 애플리케이션 간의 통합이다 [12].

본 논문에서는 토픽맵을 EA내의 각 관점들의 연관관계를 표현에 이용하고, "merging" 과정을 이용하여 EA의 생성에 이용한다. 토픽맵은 EA내에서 비즈니스와 시스템 그리고 기술 등 각 모델들 간의 연관관계를 표현함으로써, EA내의 각 모델들이 서로 연결이 되어 통합될 수 있도록 한다. 그리고 각각의 AU들이 토픽맵의 "merging" 과정을 이용하여 특정한 EA를 생성하도록 하였다.

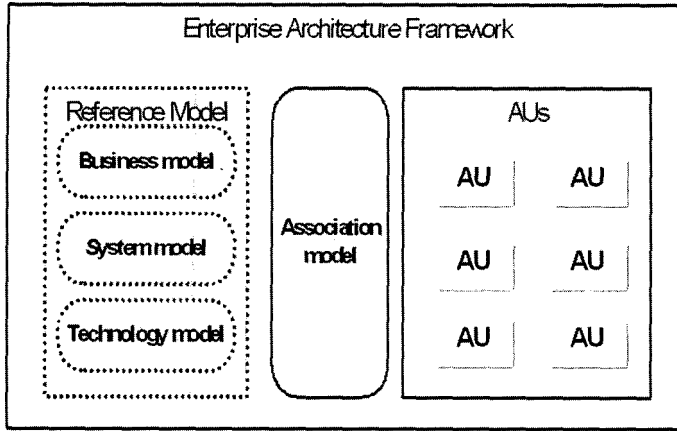
## 2.6 DSM (Domain Specific Modeling)

하나의 도메인 내의 개념과 이러한 개념을

실행하는 최종 제품사이의 연결은 매우 어려운 일이다.[4] UML과 같은 대중적인 언어들을 개념 모델에 사용하면, 언어가 도메인과 직접적으로 연관이 되어 있지 않기 때문에, 부가적으로 해당 도메인과의 매핑이 요구된다. 이에 따라 최종 제품 즉 코드 생성을 하는데 <그림 6>와 같이 여러 단계를 거쳐야 한다. 그러나 DSM은 해당 도메인의 개념을 그대로 반영하여 모델을 만들고, 이러한 모델들이 해당 도메인의 최종 제품과 직접적으로 연결되어 있기 때문에 빠른 최종제품의 생성과 반응성 그리고 지식습득시간을 감소시킬 수 있다. DSM은 Domain Specific Model의 생성을 위한 메타모델인 Domain Specific Meta Model, 이를 이용하여 생성된 모델인 Domain Specific Model, 그리고 라이브러리 형태로 Domain Specific Model과 최종 제품의 연결을 도와주는 컴포넌트들로 구성된다.



<그림 6> 아이디어 구현을 위한 DSM 방법



〈그림 7〉 ENAF 구성요소

### 3. ENAF (ENterprise Architecture Framework)

본 논문에서 제시하는 프레임워크인 ENAF는 〈그림 7〉과 같이 참조모델과 관계 모델 그리고 AU들이 결합되어 있는 구조를 가지며, EA개발 방법으로는 DSM방법을 이용한다. ENAF내의 참조 모델들과 AU들은 관계모델을 통하여 서로 호환이 가능한 구조를 가진다. 즉 참조모델과 관계모델을 결합하면 AU들이 생성되며, AU들은 관계모델을 고려하여 분해하면 참조모델로 변환이 가능하다. 따라서 ENAF는 관계모델과 AU들만으로 구성이 가능하며, 참조모델의 생략이 가능하다.

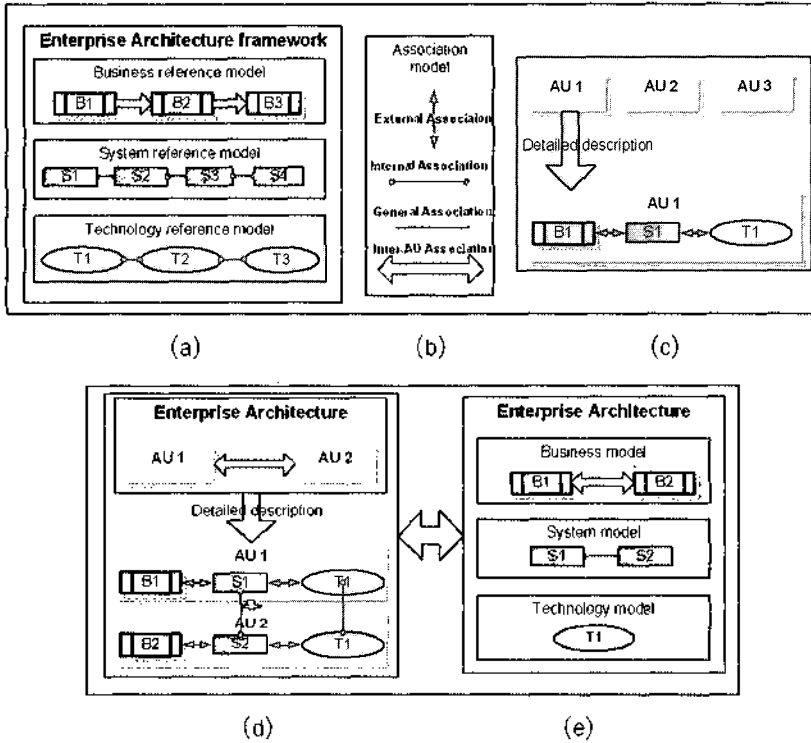
ENAF내의 관계모델과 AU를 사용하여 EA를 생성하는 과정을 설명하면 〈그림 8〉과 같다. 기업 내에 존재하는 모든 구성요소들은 서로 복잡하게 얽혀있는 구조를 가지고 있다. 이러한 구성요소들에 EA 프레임워크를 적용하여 EA를 생성하면, EA가 〈그림 8〉의 (a)

와 같이 참조모델들로 나누어져서 모델링 된다. 〈그림 8〉의 (b)는 AU를 생성하는데 기본이 되는 연관관계를 모아놓은 연관 관계 모델을 표현한다. 〈그림 8〉의 (c)는 AU를 설명한 것으로, AU는 (a)의 참조모델들과 (b)의 연관 관계모델을 이용하여 생성된다. AU를 비즈니스 프로세스 중심으로 조합하면 (d)과 같은 하나의 특정 EA를 생성할 수 있다. 이와 같이 생성된 EA는 〈그림 8〉의 (e)와 같이 참조모델에서 생성된 EA 모델로 변환이 가능하다.

#### 3.1 AU

AU는 EA의 체계적이고 쉬운 생성을 목적으로 하는 비즈니스 프로세스 중심의 EA Unit으로서 Tony Clark등의[13] language unit과 유사한 개념이다. AU는 〈그림 8〉의 (d)와 같이 비즈니스 중심으로 관련된 시스템 요소 기술 요소 그리고 이러한 요소들 간의 연관관계가 표현된 가장 작은 단위의 EA로서 다음





〈그림. 8〉 ENAF를 이용한 EA 구축

과 같은 특징을 갖는다.

### AU의 특징

(1) 기업 내의 모든 AU들의 합집합은 기업의 EA가 된다.

기업의 EA가 비즈니스 시스템 그리고 기술요소들로 구성이 되어 있으므로, AU는 비즈니스 프로세스 단위를 중심으로 관련된 시스템과 기술 요소들을 외부연관관계를 참조하여 생성된다. 즉 AU는 기업의 EA에서 비즈니스를 분리하여 먼저 모델링하고 시스템과 기술이 연관된 외부연관관계를 찾고, 외부연관관계에 따라 시스템 그리고 기술모델들을 찾아서 연결하면 완성된다. 이렇게 완성된

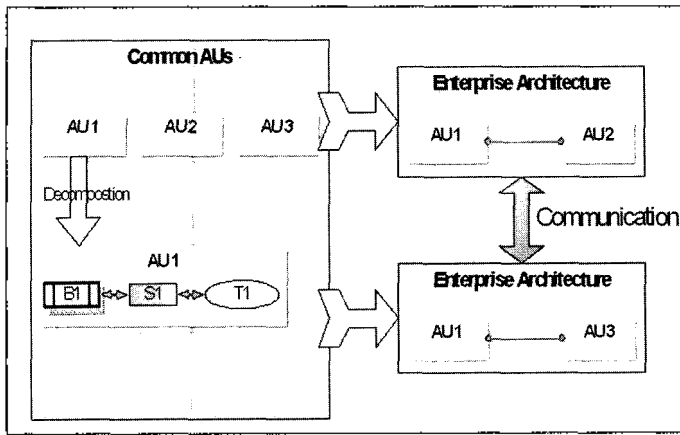
모든 기업 내의 AU들의 합집합은 기업의 전체적인 EA가 된다.

(2) AU는 그 자체로도 EA의 특징을 갖는다.

AU는 EA의 구조를 기반으로 하여 비즈니스 프로세스 중심으로 관련 요소들을 외부연관 관계를 통하여 재구성하였기 때문에, 비즈니스 프로세스 중심의 소규모 EA로서 전체적인 EA의 특징을 그대로 갖는다.

(3) AU는 EA의 컴포넌트와 같은 특징을 갖는다.

AU는 컴포넌트와 같이 필요한 AU를 조합하면 원하는 EA를 구성할 수 있다. 두 기업 또는 한 기업 내의 두 부서가 공통의 AU를 사용할 경우, 각 회사 또는 각 부서 마다 필요



〈그림 9〉 AU들간의 상호운용성

한 비즈니스 프로세스를 중심으로 AU를 조합하면 각 조직의 운영을 정보기술의 관점에서 기술한 EA가 생성된다.

(4) AU을 사용하는 기업들 또는 기업 내의 부서들은 서로 상호운용성을 갖는다.

기업에서 EA가 구축되고 이를 바탕으로 AU들이 생성 되면, AU는 컴포넌트와 같이 필요에 따라 여러 가지 조합으로 기업의 EA와 상호운용성을 보장하는 형태로 EA를 형성한다. 비즈니스 조직의 AU를 중심으로 EA가 형성이 되면, 기업의 각 부서들은 관련된 비즈니스를 중심으로 AU들을 조합하여 각 부서의 기능을 반영하는 EA를 생성한다. 이와 같이 생성된 부서별 EA들은 기업 전체적인 EA를 기준으로 만들어 졌기 때문에 각 부서별 EA들 간의 상호운용성이 보장된다.

일반적으로, 공통의 EA에서 개별적인 EA의 생성 시, 공통의 EA내의 구성 요소들의 관점은 개별적인 EA의 특성에 따라 각 구성요소들의 관점이 바뀌기 때문에 구조의 일관성이 유지가 어렵다. 예를 들어 공통의 EA (기

업의 공통적인 EA)에서 기술모델에 "barcode"이 정해져 있다고 가정하면, 개별적인 EA(한 기업 내의 부서 또는 기업과 관련 있는 기업으로, 예를 들면 "barcode"를 개발하는 부서 및 하청기업) 입장에서는 "barcode" 자체가 비즈니스가 되는 경우가 있다. 이럴 경우 개별적인 EA들 간의 상호운용성이 보장되지 않는다. 그러나 AU를 이용하여 EA를 생성하면 개별적인 EA들은 공통의 EA의 구조를 그대로 가져오기 때문에 모델의 일관성이 확보되고 상호운용성도 보장된다. 그림 9에 나타난 바와 같이, 기업과 기업 간의 연관 관계에 있어서 기업의 참조모델 또는 관련 기업들 간의 참조모델형태로 AU를 형성하고 각 기업들이 이를 활용하면, 기업들 간의 상호운용성이 보장되어서 협업이 용이해진다.

(5) AU은 분해(decomposition)가 가능하며 분해 된 AU들은 AU의 특징을 그대로 갖는다.

비즈니스 프로세스가 필요에 따라 하위의 프로세스로 분해되면, AU들도 보다 상세한 하위 AU들로 분해된다. 분해된 하위 AU들도

상위의 AU과 같은 특징을 지니며, 하위의 AU들을 조합하면 하위의 EA가 생성된다.

(6) 비즈니스 요구사항에 따른 빠른 응답성

AU는 비즈니스 프로세스 중심으로 모델링 되어있고, 하나의 비즈니스 프로세스만을 기준으로 모델링 되어 있기 때문에, 비즈니스 요구가 발생하면 해당하는 비즈니스 프로세스를 수정하여 비즈니스의 요구사항에 빠르게 반응할 수 있다.

(7) AU들은 연관관계가 포함되어 있다.

AU내에는 AU내외의 모든 요소들과 관련된 연관관계들이 포함되어 있다. 이러한 연관관계는 AU의 이해에 도움을 주며, AU들이 결합하여 EA를 생성할 경우 판단의 근거를 제공한다.

### 3.2 연관 관계모델

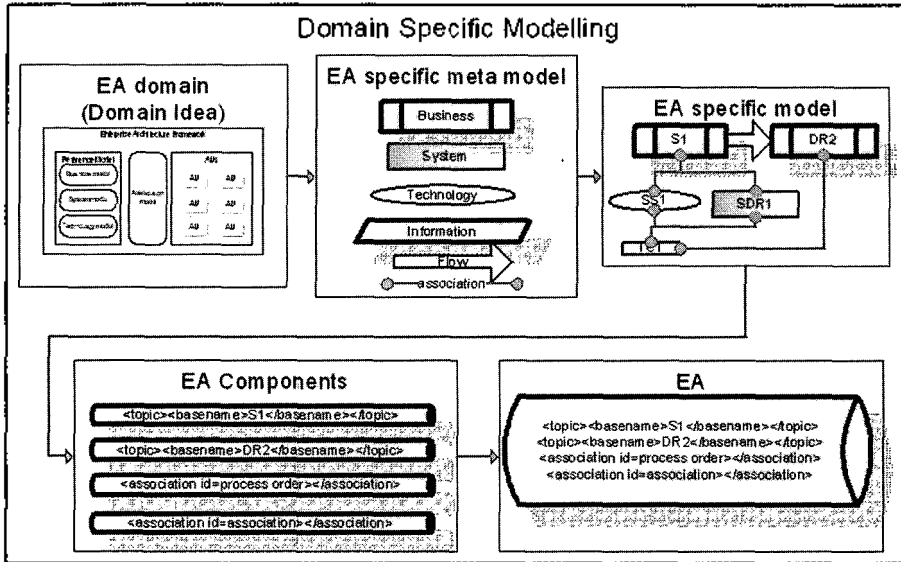
ENAF에서 관계모델은 비즈니스, 시스템, 데이터, 기술 참조모델들 간의 연관관계들을 표현한 것으로서, AU의 기본이 되는 연관관계를 표현한다. 관계모델은 EA에 필요한 연관관계를 정의한 일종의 템플릿으로서 EA를 생성할 때 명확하고 체계적인 연관관계의 설정을 돕는다. <그림 8> (b)에 표현한 바와 같이, 관계모델은 크게 AU안에서 각 관점별 요소들 간의 연관관계를 표현한 외부연관관계(external association), 기존의 EA 프레임워크내의 모델들 내의 요소들 사이의 연관관계를 정리한 내부연관관계(internal association), 일반적인 연관관계를 표현하는 일반연관관계 그리고 AU들 간의 연관관계를 정의한 AU연관관계 4가지의 종류를 갖는다.

외부 연관관계는 EA 프레임워크를 나타내는 필수적인 기본 구조를 표현하는 연관관계로서 AU내에서 EA 프레임워크에 나타나 있는 관점들을 표현한다. 예를 들어 “system-technology” 연관관계는 “system”과 “technology”의 역할을 갖는 외부연관관계로 정의되며, “business-system” 연관관계는 “business”와 “system”의 역할을 갖는 외부연관관계로 정의된다. AU내에서의 외부연관관계는 EA 프레임워크의 구조를 표현한다.

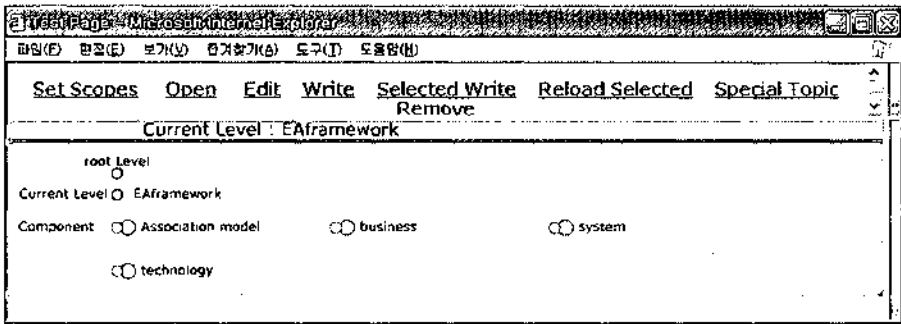
내부연관관계는 기존의 모델들 내에서 정의되어 있는 연관관계를 AU들 간의 연관관계로 나타낸 것이다. AU연관관계는 AU들 간의 연관관계를 나타내는 것이다. 이러한 AU들 간의 연관관계는 AU가 비즈니스 중심으로 형성되어 있기 때문에 주로 비즈니스 간의 연관관계를 표현한 것이 된다. 예를 들면 비즈니스모델에서 “buy”, “sell”은 “process flow”란 연관관계에 “previous process”, “next process”란 역할을 갖는 연관관계를 정의함으로써 비즈니스의 순서를 정의할 수 있다.

### 3.3 Domain Specific Model

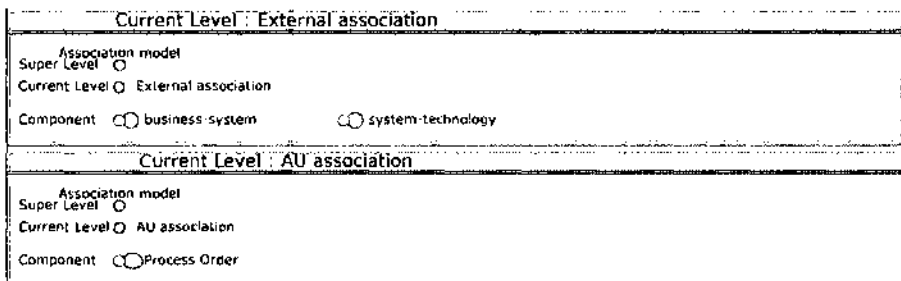
DSM 방법을 이용한 ENAF의 구현방법은 <그림 10>에서와 같이 우선 ENAF를 따르는 EA 모델을 모델링 하기위하여 domain specific meta model을 정의하고, 이 meta model을 이용하여 domain specific model을 생성한다. 그리고 domain specific model을 최종 형태인 토폭맵 파일에 직접 연결하거나 컴포넌트 라이브러리 형태로 토폭맵과 연결하게 된다. 즉, ENAF내의 EA는 연관관계 모델과 AU들이



〈그림 10〉 EA 구축을 위한 DSM 방법



〈그림 11〉 제안한 Enterprise Architecture Framework



〈그림 12〉 제안된 Association model

결합되어진 형태로 토픽 맵을 이용하여 표현하게 된다. EA에서 결합된 AU들은 분리되어 각각의 토픽 맵으로 표현될 수 있으며, 분리되어진 각각의 AU들이 결합하여 생성한 EA를 토픽 맵의 "merging" 과정을 이용하여 <그림 11>에서 나타낸 바와 같이 구현할 수 있다. <그림 11>은 SCOR를 기본으로 하는 공급망의 EA를 관계모델과 AU들이 결합되어진 형태로의 토픽맵을 보여준다. 여기서 "business", "system", "technology" 참조 모델을 가지며, 부가적으로 본 논문에서 제안된 관계모델이 추가되어 있다. <그림 12>의 관계모델은 외부연관관계인 "system-technology"와 "business-system", AU연관관계인 "process Order"가 토픽맵으로 나타낸 것이다.

#### 4. 적용 예

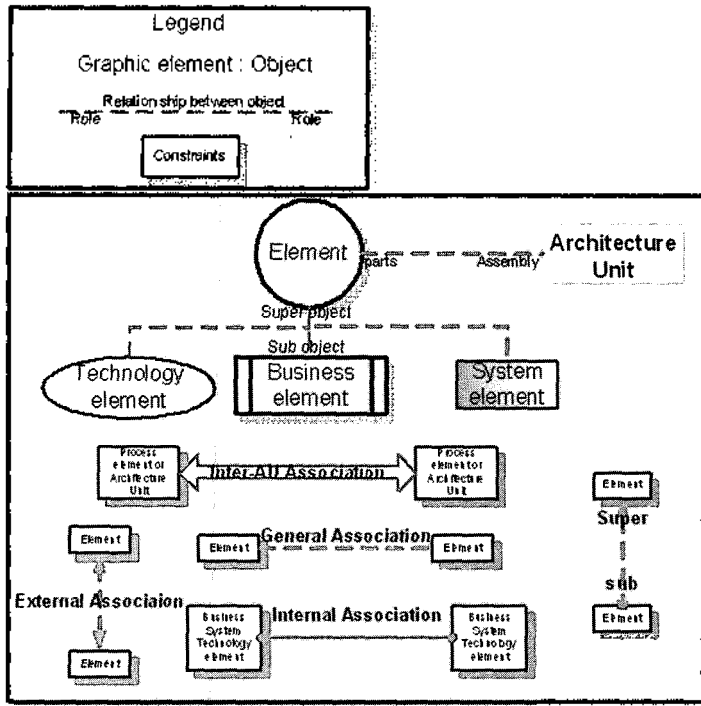
ENAF를 공급망에 적용하여 EA모델을 생성하기 위하여 SCOR모델을 이용하였다. SCOR모델은 프로세스 참조 모델로서 모든 관점이 프로세스 중심으로 결합되어 표현되어 있다. SCOR모델을 EA내로 포함시키면서 동시에 EA의 상호운용성을 확보하기 위하여 SCOR모델은 EA가 가진 관점에 따라 분해가 되어 AU형태로 표현 되어야 한다. 따라서 SCOR 모델은 비즈니스 시스템, 그리고 기술 모델로 분리 되어야 하고 그 연관관계를 나타내는 관계모델이 나타나야 EA에 적용 가능한 모델이 된다.

#### 4.1 Domain specific meta model

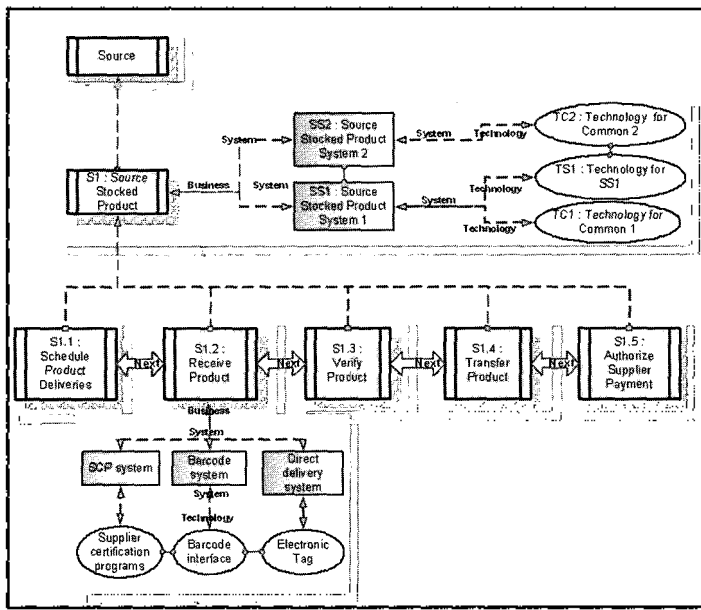
본 논문에서 사용된 SCOR모델에 대한 관계 모델 및 AU의 메타모델은 비즈니스 시스템 그리고 기술개념과 이들 사이의 관계를 정의하는 연관관계들을 정의하는 것으로서 <그림 13>과 같이 그래픽 요소들로 정의할 수 있다. ENAF기반의 EA모델에서 사용되는 요소들은 비즈니스 시스템 그리고 기술요소 등이 있으며, 관계모델에서 사용되어지는 연관관계들은 내부연관관계와 외부연관관계, 일반연관관계 그리고 AU연관관계이다. EA 모델의 관점을 표현하는 외부 연관관계는 비즈니스 모델, 시스템 모델 그리고 기술모델의 요소들을 연결시켜 주며, 내부연관관계는 참조 모델의 내부 요소들 간의 연관관계를 표현한다. 외부 연관관계는 비즈니스와 시스템의 연관관계를 표현 하는 "business-system" 연관관계와 시스템과 기술과의 연관관계를 표현 하는 "system-technology" 연관관계가, 내부연관관계로는 시스템과 시스템 그리고 기술과 기술을 연결하는 "connected to" 연관관계가, AU연관관계로는 AU의 순서를 나타내는 "Process Order" 연관관계가, 그리고 일반연관관계의 하나로서 모델 요소들의 계층을 구분하기 위하여 "super-sub"관계가 사용되었다.

#### 4.2 AU의 Domain Specific Model

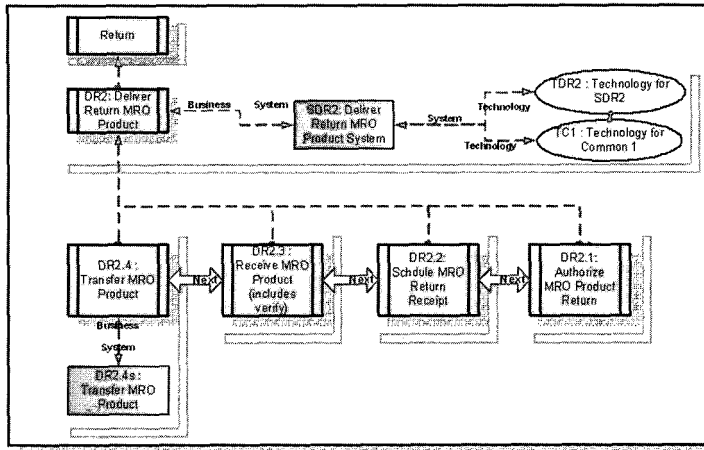
<그림 14>는 SCOR모델을 기반으로 그림 13에 정의된 메타모델에 따라 Domain specific model로 AU들을 생성한 예이다. 하나의 비즈니스 프로세스 요소와 이와 관련된 시스템 요



〈그림 13〉 ENAF의 Domain Specific meta model



(a)



(b)

〈그림 14〉 Domain specific model of AU

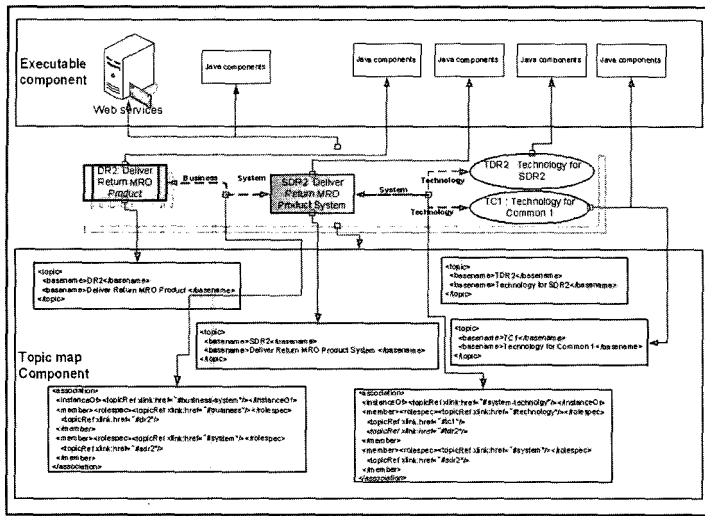
소 그리고 기술요소를 관계모델에서 정의된 형태로 표현하면 AU들이 생성된다. 〈그림 14〉 (a)는 메타모델에 기초하여 SCOR모델의 "S1: Source Stocked Product" 프로세스를 AU로 표현하고, "S1"과 "S1.2"에 대하여 자세하게 표현한 것이다. "S1"은 "S1.1", "S1.2", "S1.3", "S1.4" 그리고 "S1.5"의 하부 비즈니스 프로세스로 분해가 되며 하부 비즈니스 프로세스는 내부연관관계로 비즈니스 프로세스의 순서를 정의하는 연관관계를 가진다. 〈그림 14〉 (b)는 "DR2: Deliver Return MRO Product" 프로세스를 AU로 표현하고, "DR2"와 "DR2.4"에 대하여 자세하게 표현한 것이다. 이와 같이 SCOR모델 전체를 AU로 구성하고, 협업적 공급망에 참여하는 기업들이 AU를 활용하면 각 기업들의 EA가 협업적 공급망을 지원하게 된다.

### 4.3 토픽맵 생성

Domain specific model은 〈그림 15〉와 같이 각각의 요소들은 토픽맵의 "topic"요소로 변환되어 지고, 연관관계는 "association"요소로 변환되어 토픽맵 코드로 작성되어 최종적으로 토픽맵 형태의 EA가 완성된다. domain specific model은 topic map이외에도 웹 서비스나 자바로의 맵핑을 통하여 실행 가능한 코드로 생성이 가능하다.

〈그림 16〉은 〈그림 14〉 (a)의 "S1: Source Stocked Product" 프로세스와 〈그림 14〉 (b)의 "DR2: Deliver Return MRO Product"를 토픽맵으로 구현한 것이며, SCOR의 기본 프로세스인 "plan", "source", "make", "deliver", "return"가 나타나 있다.

〈그림 17〉 (a)는 공급망상에서 필요한 비즈니스 프로세스들을 개념적으로 나타낸 그림이며, "Consumer"가 "S1"과 "DR2" 두 가지



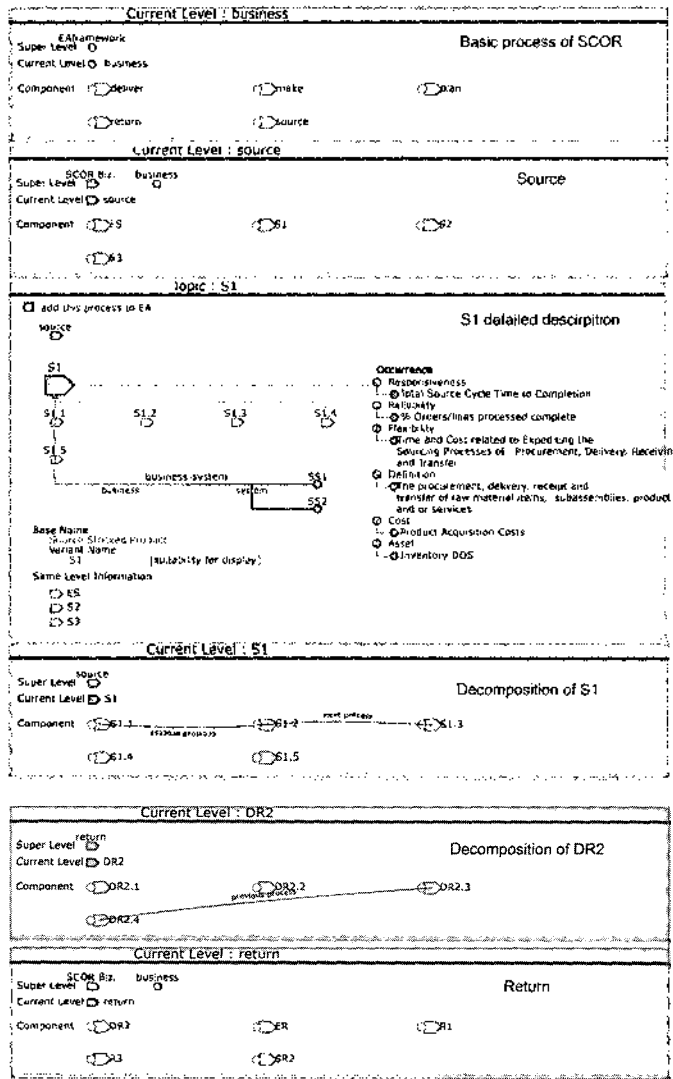
〈그림 15〉 DSM에서 topic map code로의 전환

비즈니스 프로세스를 필요로 함을 알 수 있다. 통합된 EA는 AU들이 결합되어져 있는 상태이기 때문에, 비즈니스를 선택하고 AU를 추출하면 〈그림 17〉의 (b)와 (c)같이 각각의 AU가 하나의 컴포넌트 형태인 토픽맵으로 만들어진다. 이때 하나의 비즈니스 프로세스를 선택하면 독립되어진 하나의 AU가 토픽맵으로 생성되며, 복수를 선택하면 AU들이 결합되어진 형태로 토픽맵이 생성되어진다. 토픽맵으로 생성된 독립된 AU들을 토픽맵의 "merging"을 이용하여 통합하면 〈그림 17〉의 (d)와 같이 선택된 AU들로 EA를 구축할 수 있다. 즉 원하는 비즈니스가 "SI"과 "DR2"일 경우 두 가지 비즈니스 중심의 AU 컴포넌트를 토픽맵의 "merging"을 이용하여 결합하면 두 가지 비즈니스를 중심으로 EA가 생성된다. 이와 같이 AU를 활용하여 EA를 생성하면 쉽게 특정의 EA를 생성할 수 있다.

## 5. 결 론

본 논문에서 ENAF는 기존 참조모델 외에 연관 관계모델을 추가하고 이를 이용하여 EA의 단위 컴포넌트인 AU의 개념을 도입한 것이다. ENAF의 관계모델과 AU를 활용한 EA생성 방법은 구조적으로 일관성이 있는 EA의 생성이 가능하며, 기존의 참조모델 방법에 비하여 모델들 간의 연관관계가 표현되어 이해하기 쉽다. 특히 EA의 생성 시에 AU들을 비즈니스 중심으로 결합하여 EA를 생성하는 구조를 갖게 함으로서 보다 쉬운 EA의 생성이 가능하게 하였다 AU는 비즈니스 중심으로 각 관점에 따른 모델과 관계모델을 조합한 컴포넌트이며 AU를 조합하면 쉽게 EA의 생성이 가능하다. 그리고 ENAF의 개념을 이용하여 EA를 구현하기 위하여 DSM 방법을 이용하였다. DSM방법은 일반적인 모델링 방법에 비하여 EA의 빠른 개발과 실행





〈그림 16〉 Topic maps for Enterprise Architecture

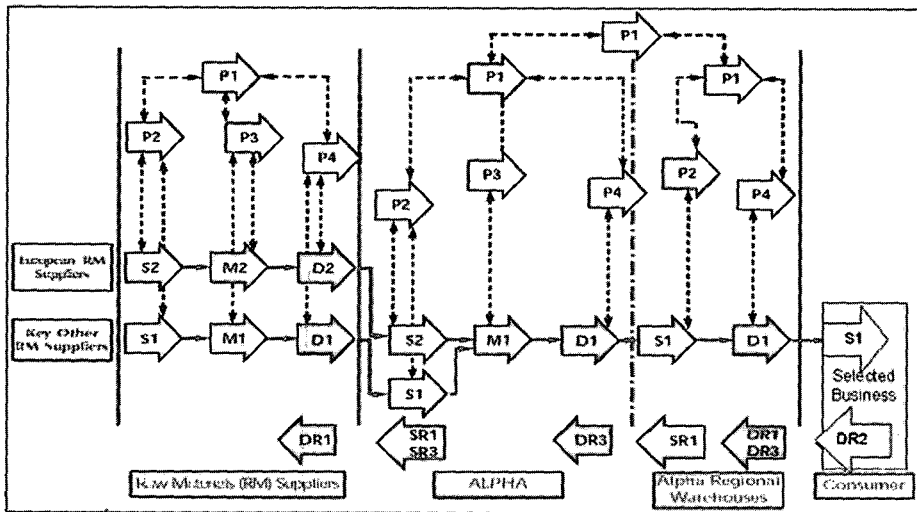
을 가능하게 한다. ENAF와 EA의 적용 예로서는 공급망을 위한 EA 모델의 Domain Specific Meta Model을 정의하였고, 이를 바탕으로 비즈니스 프로세스 참조 모델인 SCOR모델을 참조하여 Domain Specific Model인 EA 참조모델, 관계모델 그리고 AU를 생

성하였다. 그리고 생성된 EA모델들을 기반으로 Topic Maps형태로 EA들을 구현하였다. 구현된 EA에서 AU들은 컴포넌트 형태로 분리 및 결합이 이루어지며, 이를 토픽맵의 "merging" 과정을 이용하여 구현하였다.

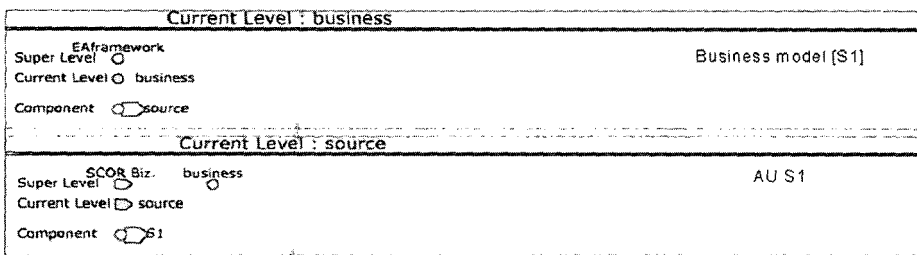
4장의 도입 예에서 알 수 있듯이, 참조 모델

정의를 통하여 도메인의 특성을 반영할 수 있는 EA를 구현할 수 있으므로, 보다 정확한 Enterprise Architecture를 구현할 수 있으며, 이렇게 구현된 모델은 토포 맵을 이용하여 검

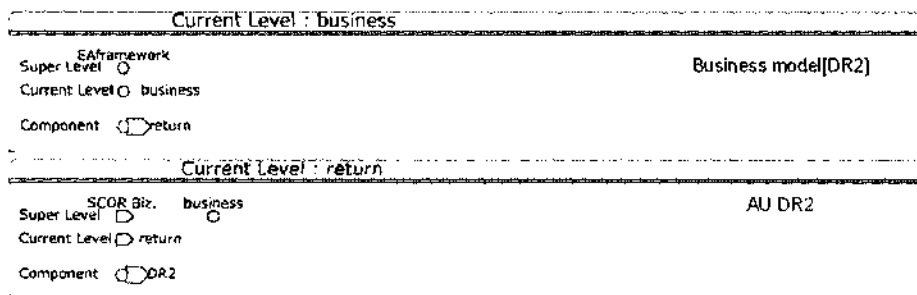
색이 가능하므로, 기업 간 거래에서 요구되는 문제 중의 하나인 기업 간에 존재하는 비즈니스 프로세스의 서로 다른 시멘틱에 대한 문제를 해결할 수 있다.



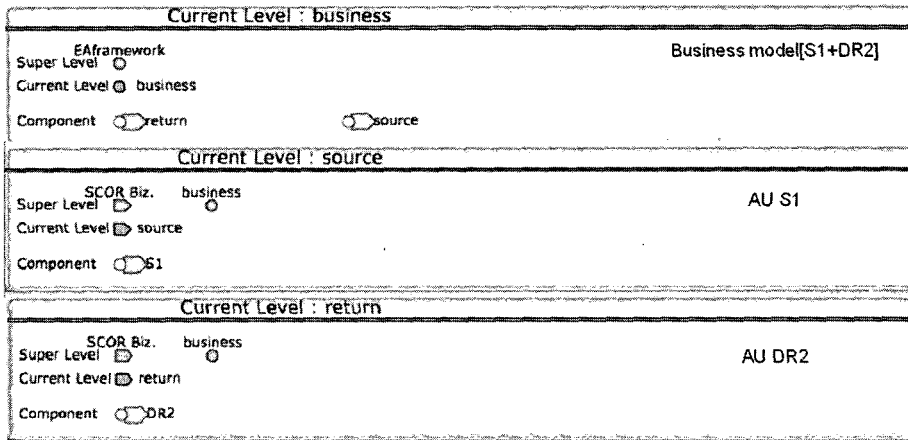
(a)



(b)



(c)



(d)

〈그림 17〉 2개의 AU로부터 EA 생성방법

## 참 고 문 헌

1. OMB Circular A-130, Transmittal Memorandum #4, Management of Federal Information Resources, 2000
2. Open group, "preface" the open group architecture framework(TOGAF) version 8.0, 2002
3. Henk Jonkers et al., "Concepts for Modelling Enterprise Architectures", [http://www.serc.nl/lac/2003/academic\\_papers/H.Jonkers.pdf](http://www.serc.nl/lac/2003/academic_papers/H.Jonkers.pdf), 2003
4. MetaCase, "Domain Specific Modeling - 10 Times faster than UML", <http://www.metacase.com/papers/>, 2004
5. Supply Chain Council (SCC), "Supply Chain Operations Reference Model - SCOR version 6.0", Supply Chain Council, Inc., 2003
6. Members of the TopicMaps. Org Authoring Group, "XML Topic Maps (XTM) 1.0", <http://topicmaps.-org/xtm/>, 2001
7. John Zachman, "The Zachman Framework : A Primer for Enterprise Engineering and Manufacturing", electronic book <http://www.zachmaninternational.com>, 2003
8. John Zachman, "Concepts of the Framework for Enterprise Architecture", Zachman International, Los Angeles, 1996
9. DoD Architecture Framework working group, "DoD Architecture Framework : Volume I : Definitions and guidelines", 2003
10. The chief information officer council, "FEAF version 1.1", 1999
11. Philip Allega, "Modeling the Enterprise Solution Architecture", [www.metagroup](http://www.metagroup).

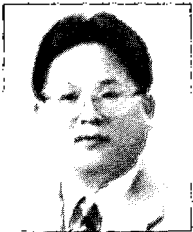
com , META Practice, 2002

- 12. Kal Ahmed (Techquila), "Introducing Topic Maps. A Powerful, Subject-Oriented Approach to Structuring Sets of Information." XML Journal Volume 03, Issue 10, 2003
- 13. Clark A, Evans A and Kent S. "Aspect-Oriented Metamodelling", The Computer Journal , Volume 46, Issue 5, pp. 566-577, 2003

### 저 자 소 개



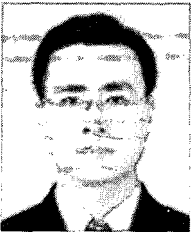
채희권 (E-mail : hkchae@postech.ac.kr)  
 1986. 3 ~ 1990. 2 연세대학교 기계공학과(학사)  
 1990. 3 ~ 1992. 2 연세대학교 기계공학과(석사)  
 1992. 2 ~ 2000. 2 삼성전자  
 2000. 2 ~ 현재 포항공과대학교 산업공학과 박사과정  
 관심 분야 Semantic Modeling, Enterprise Architecture, Business Process Management, Web service, Product Modeling



김광수 (E-mail : kskim@postech.ac.kr)  
 1973. 3 ~ 1977. 2 서울대학교 공과대학 산업공학과(학사)  
 1977. 3 ~ 1979. 8 서울대학교 공과대학 산업공학과(석사)  
 1977. 1 ~ 1980. 7 한국섬유기술연구소연구원  
 1980. 8 ~ 1982. 7 Kansas Stac Univ. 연구원  
 1982. 9 ~ 1985. 8 Univ. of Central Florida 산업공학과 박사  
 1985. 8 ~ 1988. 2 Rochester Institute of Technology 조교수  
 1988. 2 ~ 현재 포항공과대학교 부교수/정교수  
 관심 분야 Enterprise Architecture, Web Services, Semantics, Business Process Management



김철한 (E-mail : chkim@dju.ac.kr)  
 1980. 3 ~ 1984. 2 한양대학교 공과대학 정밀기계공학과(학사)  
 1984. 3 ~ 1986. 2 한양대학교 공과대학 정밀기계공학과(석사)  
 1987.10 ~ 1989. 2 금성소프트웨어 공장자동화실  
 1989. 3 ~ 1993. 8 포항공과대학교 산업공학과 박사  
 1993. 7 ~ 1995. 8 삼성 SDS SM전략실  
 1999. 7 ~ 2000. 6 영국 MSIRI (Loughborough 대학) 객원연구원  
 1995. 8 ~ 현재 대전대학교 인터넷정보공학과 부교수  
 2002. 8 ~ 현재 ebXML 전문위원회 위원  
 2004. 8 ~ 현재 전자거래진흥원 / 비즈니스 프로세스 워킹그룹 위원장  
 관심 분야 Virtual Enterprise, Enterprise Architecture, Web Services,  
 Business Process Management



최영환 (E-mail : yhchoi@cebt.re.kr)  
 1986. 3 ~ 1990. 2 서울대학교 공과대학 산업공학과(학사)  
 1990. 3 ~ 1992. 2 포항공과대학교 산업공학과(석사)  
 1992. 3 ~ 1999. 1 대우중공업 산업차량 연구개발실  
 1999. 3 ~ 2005. 2 포항공과대학교 산업공학과 박사  
 2005. 3 ~ 현재 서울대학교 컴퓨터연구소 선임연구원  
 관심 분야 Enterprise Architectures and Frameworks,  
 Business Process Modeling and Management,  
 Semantic Web Services, Topic Maps Applications,  
 Product Information Model