# 정보자원관리 관점에서의 통합 통제 아키텍처 메타데이타모형

김정욱
동양공업전문대학 경영정보과 조교수
(jwkim@orient.dytc.ac.kr)
김영걸
한국과학기술원 테크노경영대학원 경영공학과 부교수
(domino2@unitel.co.kr)

이기종의 분산 컴퓨팅 환경에서의 정보자원관리가 점차 복잡해짐에 따라 이러한 정보자원들을 조정하고 통제하는 능력이 효과적인 정보자원관리의 주요성공요소로 간주되고 있다. 통합 통제 아키텍처를 활용함으로써 기업의 정보자원들을 좀더 효율적으로 관리 할 수 있고, 정보자원의 변동에 따른 영향분석을 수행할 수 있고, 주요 정보자원의 통제를 자동화함으로써 수작업을 경감할 수 있다. 이 논문에서는 통합 통제 아키텍처의 개념적 틀과 메타데이타 모형을 제시한다.

## I. 머리말

데이타 및 응용소프트웨어와 같은 개별 정보자원들이 이기종의 하드웨어와 네트워크의 복잡한 통신망으로 분산 배치됨에 따라 분산 컴퓨팅 환경하에서의 정보자원관리는 좀더 어려워지고 있다. 정보자원관리는 기업의 중요자원으로 하드웨어와 소프트웨어와 같은 정보생산체계를 인식한다는 사고로부터 출발한다. 일반적으로 정보자원관리 (IRM: Information Resource Mangement)는 현재의 정보기술에 적절한 경영정책과 경영기법을 효과적으로 결합하여 전사적인 정보자원의 활용을 최대화하는 활동으로 이해되어왔다. March와 Kim [1988]은 정보자원관리의 활동들을 "전략계획, 확장계획, 정보시스템개발, 프로젝트관리, 하드웨어와 소프트웨어 획득 및 데이타관리를 포함하는 다양한 활동"으로 정의했다.

정보자원관리의 활동을 성공적으로 수행하기 위해서는 목표되는 정보자원의 이해가 필요하다. 정보시스템 아키텍처는 정보시스템 관리자들에게 정보자원에 대한 그래프 모형 및 강력한 데이타 저장기능을 가진 메타데이타를 관리하는 능력을 제공한다. 정보시스템 아키텍처는 정보시스템 담당이사들이 선택한 10가지의 주요 정보시스템 관리 이유중 첫번째 이슈 [Niederman et al., 1991] 혹은 네번째 이슈 [Brancheu et al., 1996]로 등장하였다. Zachman[1987]은 3가지의 관점(소유자 관점, 설계자 관점, 구축자 관점) 및 3가지의 목표정보자원들(데이타, 응용소프트웨어, 네트워크)로 구성되는 매트릭스를 처음으로 소개하였다. 그이후, 많은 연구자들에 의해 집중 및 분산 컴퓨팅 환경을 포함하는 정보시스템 아키텍처의 구성및 구축과정이 수행되었다 [Sheer, 1992; Sowa와 Zachman, 1992; Umar,1993; Kim and Everest, 1994; Bauer et al., 1994; Everden, 1996].

그러나 대다수의 정보시스템 아키텍처 틀과 상업적 도구들은 데이타, 응용소프트웨어, 하드웨어 및

논문 접수일 : 98. 1 게재확정일 : 98. 3

네트워크의 개별 정보자원관리에 중점을 두고 있으나, 정보자원들을 통합된 개념으로 통제하는 개념이 부족하다. 이 논문에서는 이러한 기능을 수행하는 통합된 통제 아키텍처의 개념적 구성틀을 제시한 통제 아키텍처를 이용함으로써, 정보자원의 전사적 관리 및 정보자원의 변동에 따른 영향분석을 수행하고, 정보자원의 동적통제과정을 자동화함으로써 수작업을 경감할 수 있다. 또한 통합 통제아키텍처가 성공적으로 구축되면 이기종의 분산된 정보자원을 마치 동일한 플랫폼하에서의 집중컴퓨팅 환경에서와 같이 관리할 수 있게 한다.

이 논문의 구성은 다음과 같다. 첫째, 기존의 논문 연구에서는 정보시스템 아키텍처의 다양한 개념적 구성틀을 비교한다. 둘째, 정보시스템 아키텍처 구성틀중 적절한 대상을 선택하여 통제 관점에서 재정의 한다. 셋째, 통제아키텍처 구축을 위한 목적과 잇점등을 논의한다. 넷째, 통합 통제 아키텍처를 논리적 수준과 물리적 수준에서의 모형을 제시한다. 마지막으로 결론 및 향후 연구 방향을 논의한다.

## II. 문헌연구 및 재정의

### 2.1 정보시스템 아키텍처 구성틀에 관한 분석 및 평가

정보시스템 아키텍처의 주요 목적은 개별 정보자원의 계획, 설계, 구축 및 관리하는 방향을 제공하는 것이다. 〈표 1〉은 기존 정보시스템 아키텍처를 비교한다. Sheer[1991]는 새로운 전사적 모형의 분석 및 개발을 위하여 조직관점, 데이타관점, 기

〈표 1〉 정보시스템 아키텍처들의 비교

| | Sheer | Sowa와 Zachaman | Umar | Kim와 Everest | Bauer | Everden |
|---|---|---|---|---|---|---|
| 구성요소 | 데이타, 기능, 조직, 자원, 통제 | 데이타, 기능, 네트워크, 수행자 시간, 동기 | 데이타, 응용소프트웨어, 기술, 통제 | 데이타, 기능, 기술, 통제 | 네트워크, 운용체제, 응용소프트웨어 | 조직, 비지니스, 기술 |
| 상세화 범위 | 요구분석자, 설계자, 구축자 | 계획자, 소유자, 설계자, 구축자, 하위구축자 | 계획자, 소유자, 설계자 | 계획자, 소유자, 설계자 | 구축자 | 해체, 구성, 운영 |
| 설계수준 | 계획 설계 구축 | 계획 설계 구축 | 계획 | 계획 설계 | 구축 | 계획 설계 구축 |
| 통제종류 | 통제관점 | 없음 | 관리 및 지원 체제 | 상호참조 매트릭스 | 관리도구, 서비스와 에이전트 | 이력관리 |
| 통제개념 | 지원수준간의 통제관점 | 시간경과의 통제구조 | 관리 | 동적관점의 통제아키텍처 | 통합관리 | 시간경과의 변형 |
| 영향분석유무 | 무 | 무 | 무 | 유 | 무 | 무 |

능관점, 자원관점으로 구성되는 통합정보시스템 아키텍처를 제시한다. 그는 개별 정보자원 사이의 관계를 통제관점 중심으로 보여준다. 통제관점이 추구하는 목적은 개별 관리대상들을 통합하는 것이다. 통제관점은 조직, 정보객체, 기능, 정보기술의 통제정보를 위하여 위에서 언급한 4가지의 관점과 상호 교류한다. 그러나 통제관점은 정보자원의 동적인 상태 혹은 정보자원 변화에 따른 영향분석등을 수행하지 않는다.

Sowa와 Zachman[1992]은 기 발표된 Zachman[1987]의 논문에 근거하여 정보시스템의 대상인 비지니스를 5가지 관점에서 관찰하는 정보시스템 아키텍처 구성틀을 제시하였으며, 특히 시간과 주기요소로 구성되는 기술 모형의 시간 차원으로서 통제구조를 언급되었다. 이들의 구조는 30개의 독립적 셀로 구성되기 때문에 완전히 구축하기에는 너무 범위가 넓고 복잡하다. 각각의 수준에 대해 모든 다이어그램과 메타 데이타를 만들고, 유지하기 위해서는 많은 노력이 필요하다. 이들의 아키텍처는 두가지의 제약조건 즉, 기능이 상호 비지니스 프로세스를 고려치 않고 단지 정보처리만을 수행하는 반면 네트워크는 클라이언트 혹은 서버와 같은 정보처리요소가 아닌 노드와 링크와 같은 네트워크 구성요소만을 정의한다. 감시 및 관리기능으로서 통제 아키텍처가 지원되지 않고, 정보자원들간의 영향분석 기능 역시 수행되지 않는다.

Umar[1993]는 전사적 시스템, 응용소프트웨어 및 플랫폼의 기능단계로부터의 분산 컴퓨팅 참조모형을 제시한다. 전사적 시스템은 비지니스 프로세스와 기능활동들로 표현된다. 응용소프트웨어는 사용자 데이타베이스, 데이타베이스 접속 및 처리 프로그램, 사용자 인터페이스로 구성된다. 플랫폼은 다른 아키텍처 구성 틀의 기술 아키텍처에 해당한다. 관리 및 지원은 분산 컴퓨팅 서비스를 관리하기 위한 기술 도구, 기법 및 조직관리와 연관되며 Kim과 Everest[1994]에 의해 제시된 통제아키텍처와 유사하다. Umar의 모형은 데이타베이스와 응용소프트웨어와 같은 다른 정보자원은 고려하지 않고 네트워크 관리기능에만 중점을 두고 있어서 통제아키텍처와 같이 자세한 정보를 제공하지는 않는다. 이 모형은 구축단계와 연결되지 않고 정보계획 단계만을 지원하므로 정보자원들간의 영향분석은 수행할 수 없다.

Kim과 Everest[1994]의 구성 틀은 다양한 정보자원 통합을 촉진하기 위하여 하나의 아키텍처와 다른 아키텍처간의 상호참조 매트릭스를 포함한다. 그들은 Zachman[1987]의 네트워크 흐름을 기술 아키텍처로 확장하고, 동적 정보자원관리를 지원하기 위한 정보 자원들의 시간경과에 따른 통제 아키텍처를 추가시켰다. 그러나 정보자원들간의 관계를 표현하기 위해 상호 참조 매트릭스를 사용했으나 통제 아키텍처의 자세한 구성요소들을 충분히 설명하지 못했다. 또한 통제 아키텍처의 논리적 수준만을 표현했기 때문에 통제 아키텍처의 구현기반이 미비하며, 논리적수준 및 물리적 수준 사이의 연결이 필요하다.

Bauer외 3인[1994]은 분산시스템을 위한 통합관리 아키텍처를 제시했다. 이것은 네트워크 서비스와 장치, 운영시스템 서비스와 자원, 사용자 응용소프트웨어를 포함하는 3가지 유형으로 정의했다. 주요구성요소들로는 관리도구, 관리서비스, 관리 에이젼트들이다. 관리서비스는 다시 형상시스템, 감시시스템, 통제시스템, 정보저장관리시스템의 4가지 하위시스템으로 구분했다. 그들의 아키텍처는 개별정보자원의 통합을 위한 논리적 모형을 정보관리자들에게 제공하지 않는다. 따라서 시스템

관리자가 환경변화에 쉽게 적응할 수 있도록 통제 아키텍처의 논리적 모형과 물리적 모형의 연결이 필요한데, 이 아키텍처 역시 개별 정보자원간의 관계를 통합 관리하지 않고, 단지 관리 서비스를 이용하여 정보자원을 개별적으로 통제한다.

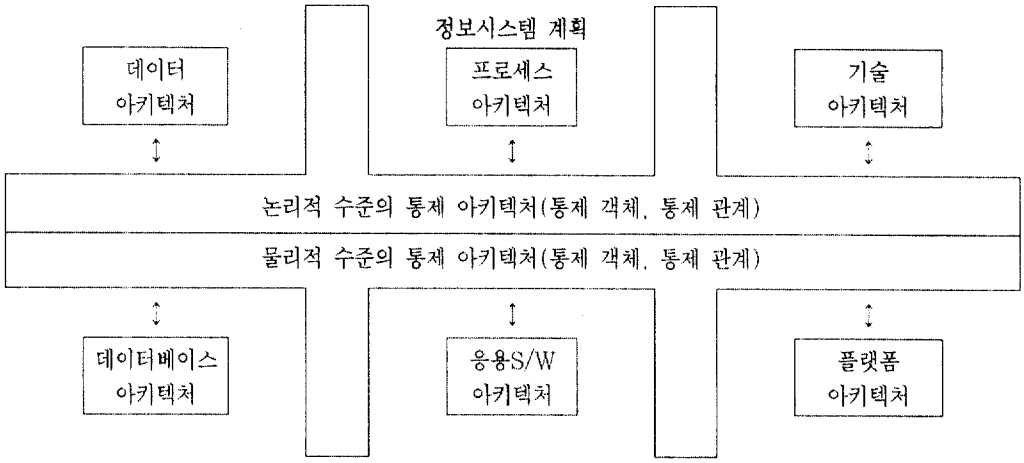
Everden[1996]은 Zachman[1987]의 아키텍처보다 보다 넓은 목적과 범위의 정보 구성틀을 제시하였다. Sowa와 Zachman[1992]의 정보시스템 아키텍처가 6개의 행과 5개의 열로 구성된 반면, Everden의 정보 구성틀은 정보유형에 따라 조직, 비지니스, 기술의 3가지 관점과 해체, 구성, 구축의 3가지 수준으로 정의된다. 각각의 관점은 10개의 행으로, 각각의 수준은 5개의 행으로 재분류된다. 위에서 언급한 아키텍처 관점에 해당하는 정보유형, 제약사항, 내용, 시간경과에 따른 변형, 소유권, 방법론, 이정표등의 6차원을 제시했다. 이 아키텍처 역시 시간경과에 대한 개별정보자원들의 이력번호를 표현하지만 정보자원들을 통합관리할 수 없기 때문에 영향분석이 가능하지 못하고 단지 6차원의 구성요소들의 개별적인 이력관리만을 수행한다.

### 2.2 정보시스템 아키텍처 구성틀의 재정의

이 논문에서는 통합통제 아키텍처들로서 표1에서 Kim과 Everest[1994] 정보시스템 아키텍처를 선택한다. 그림1에서 보여진바과 같이 정보시스템 아키텍처의 재정의된 구성틀은 개별 아키텍처의 통합을 위하여 통제아키텍처를 중심으로 특성화된다. 통제아키텍처는 논리적수준 및 물리적 수준등 2가지 수준으로 구분되며, 각각의 수준은 개별 정보자원을 관리하는 통제객체와 정보자원들간의 관계를 관리하는 통제관계의 2가지의 구조로 재분류된다. 통제관계는 데이타-데이타, 프로세스-프로세스, 기술-기술의 관계를 나타내는 동종의 정보자원들간의 관계와 데이타-프로세스, 데이타-기술, 프로세스-기술의 관계를 나타내는 이종의 정보자원들간의 관계가 존재한다.

이 논문에서는 Kim와 Everest[1994]에서와 같이 정보시스템 아키텍처의 논리적관점을 데이타 아키텍처, 프로세스 아키텍처, 기술 아키텍처를 다룬다. 데이타 아키텍처는 기업의 데이타 자산에 대한 전사적 시야를 제공하며 엔티티조건, 엔티티, 엔티티값, 속성, 범위, 주요 속성, 비주요 속성, 식별자와 같은 메타 데이타가 포함된다. 프로세스 아키텍처는 정보시스템의 자원대상이 되는 핵심 프로세스들에 대한 전사적 시야를 제공하며, 이들의 상호연관관계에 중점을 둔다. 따라서 기업목적, 기업전략, 비지니스문제, 비지니스 프로세스, 비지니스 기능, 비지니스 단위, 정보요구사항과 같은 메타데이타가 포함된다. 기술아키텍처는 전사적 플랫폼 모델링을 통해 구축되며 기업에 컴퓨터와 통신망에 대한 일관된 시야를 제공한다. 따라서 컴퓨팅 및 통신기술표준을 설정하고, 컴퓨터 하드웨어와 통신망이 개방시스템과의 통합과정을 보여준다.

정보시스템 아키텍처의 물리적관점은 데이타베이스 아키텍처, 응용소프트웨어 아키텍처, 플랫폼 아키텍처를 다룬다. 논리적관점이 비지니스 범위를 다루는 반면, 물리적 관점은 컴퓨터화된 범위를 좀 더 깊이 있게 취급하고 있다. 데이타베이스 아키텍처는 데이타 아키텍처의 물리적 수준으로 구축된 형태이며 데이타 테이블, 레코드, 필드, 데이타베이스키, 레코드값, 색인, 대상범위와 같은 데이타베이스의 실질적인 엔티티들로 구성되며, 시스템개발자와 데이타베이스 관리자에 의해 사용되고 있다. 응용 소프트웨어 아키텍처는 비지니스 프로세




〈그림 1〉 재정의된 정보시스템 아키텍처의 구성틀

스의 물리적 수준의 구축된 형태로서 정보시스템 응용소프트웨어, 모듈, 모듈 프로그램 라이브러리로 구성되고, 각각의 모듈은 다시 데이타모듈, 프로세스 모듈, 사용자 인터페이스 모듈로 세분화된다. 플랫폼 아키텍처는 기술아키텍처의 물리적 구성요소로 구체화시킨 형태로서 물리적네트워크, 네트워크 장치, 클라이언트, 서버, 시스템 소프트웨어로 구성된다.

## III. 통제 아키텍처의 개념, 목적 및 장점

### 3.1 통제 아키텍처의 개념

기존 문헌에 의하면 조직론에서 통제개념은 "행동과 결과를 감시하고, 평가하는 프로세스"〔Ouchi와 Maguire, 1975 ; Ouch, 1977 ; Eisenhardt, 1985〕, "정의된 기업목적에 대하여 피드백을 시험, 측정, 제공하는 사이버네틱 프로세스"〔Thompson,1967 ; Reeves와 Woodward, 1970〕와 같이 정의한다. 따라서 이 논문에서의 통제아키텍처는 "정의된 기업목적에 대하여 조직행동과 결과를 감시 및 평가하는 프로세스를 도와주는 아키텍처"로 정의할 수 있다.

통제아키텍처의 역할은 실시간으로 비행기트래픽을 관리하는 관제탑의 기능과 유사하다. 비행장 운행은 예상치 못한 사고를 미연에 방지하기 위하여 비행기의 이착륙 정보를 계속적으로 관리한다. 관제탑은 비행기 이착륙의 안전을 위해 비행기, 활주로, 기상조건의 세가지요소를 독립적으로 처리하기보다는 동시에 고려해야 한다. 통제아키텍처는 기업의 모든 정보자원을 감시 및 관리하는 통제패널 형태로 구축된다. 통제아키텍처는 정적통제 아키텍처와 동적통제 아키텍처와 같이 2가지의 구분될 수 있는데 정적통제 아키텍처는 정보자원계획 및 관리를 효율적으로 수행하고, 정보자원들의 상호의존성을 파악하기 위해 정보시스템 아키텍처간 관계를 표현한다〔Kim과 Everest, 1988; Lefkvovits, 1991〕. 동적통제 아키텍처는 이벤트/액션(event/

action)기법을 활용하여 데이타아키텍처, 프로세스 아키텍처, 기술아키텍처의 정보자원의 상태를 시간경과에 따라 자동적으로 감시 및 통제한다 〔Kim과 Everest, 1994; Bauer, 1994〕.

### 3.2 통제아키텍처의 장점

분산컴퓨팅 환경하에서, 정보자원들은 각각의 정보처리 장소에서 독립적으로 관리되기 때문에 잘못 활용될 수 있다. 만약 각각의 노드에서 표준화가 되지 않았다면, 전체적인 관리는 중복적 데이타베이스와 서로 일치되지 않는 응용소프트웨어로 집중컴퓨팅 환경하에서보다도 훨씬 어렵게 된다. 또한, 개별적 정보자원의 통합이 어렵기 때문에 시스템 성능하락 및 신뢰도 하락의 결과가 초래된다. 집중컴퓨팅 환경하에서의 통제능력은 시스템 신뢰도 증가 및 성능 향상이 필요하다. 노드상태의 변화는 통제화면 위에서 감시되고, 개별 정보자원 변동에 따른 영향이 분석될 수 있어야 한다. 통제아키텍처 구축으로 기대되는 장점은 첫째, 비지니스 사용자(논리적)수준과 정보시스템 사용자(물리적)수준에서 분산 정보자원에 대한 통합개념의 실시간 통제가 가능하고, 둘째, 어떤 정보처리 장소에서 시스템 성능이 하락될때 전체시스템 성능에 대한 영향이 즉시 파악될 수 있도록 영향분석이 가능하며, 셋째, 개별 정보자원의 효용도와 신뢰성을 최대화하기 위하여 시스템고장전에 시스템관리자에게 통보될 수 있으며, 넷째, 더 적은 시간으로 정보자원의 유지, 관리 및 재배치를 훨씬 더 자동화된 방법으로 할 수 있다.

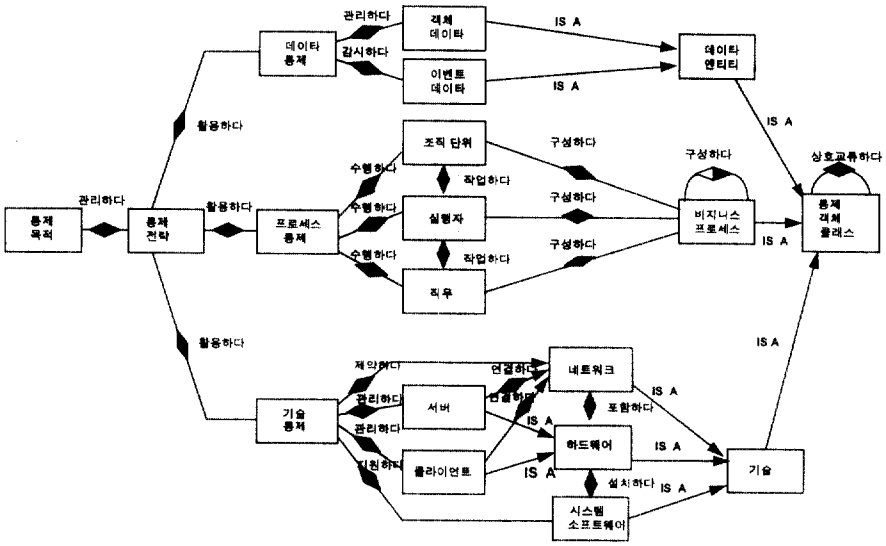
## Ⅳ. 통합 통제아키텍처의 메타데이타 모형

앞에서 언급했듯이 통제 아키텍처의 메타모형은 개별 정보자원을 감시 및 관리하는 통제객체 아키텍처와, 동종 혹은 이종의 정보자원들간 영향분석에 중점을 둔 통제관계 아키텍처의 두가지로 분류된다. 이하에서는 통제객체 아키텍처와 통제관계 아키텍처의 논리적 및 물리적 관점을 독립적으로 다루게 될 것이며, Teorey ,Yang 과 Fry〔1986〕의 확장된 ER모델방법으로 통제 아키텍처의 메타데이타 모델을 표현할 것이다.

### 4.1 논리적 관점에서의 통제객체 아키텍처

통제객체 아키텍처의 한 축은 기업목적, 기업전략 및 통제유형을 나타내며, 다른 한 축은 통제대상인 목표정보자원으로 구성된다. 그림 2는 오른쪽의 목표 정보자원에 대하여 왼쪽의 어떤 통제방법으로 관리하는 것이 적절한가를 모형화하여 보여준다. 통제객체는 통제 아키텍처의 목표 정보자원인 데이타 엔티티, 비지니스 프로세스 및 기술을 하위유형으로 한다. 데이타 엔티티는 다시 객체 데이타와 이벤트 데이타를 분류되는데, 고객 혹은 제품같은 객체 데이타는 시간경과에 따라 안정적이고, 고객출현 혹은 주문상태와 같은 이벤트 데이타는 시간경과에 따라 빈번하게 변하고, 불안정한 특성을 지닌 데이타이다. 객체 데이타와 이벤트 데이타는 데이타 통제도구와 DBMS, 데이타사전, 정규화, 재조직화와 같은 기법에 의해 감시 및 관리 된다.

비지니스 프로세스는 대부분이 서로 상호작용하며 하나 이상의 하위 비지니스 프로세스들로 구성된다. 가장 낮은 수준의 비지니스 프로세스는 다른




〈그림 2〉 논리적 관점에서의 통제 객체 아키텍처

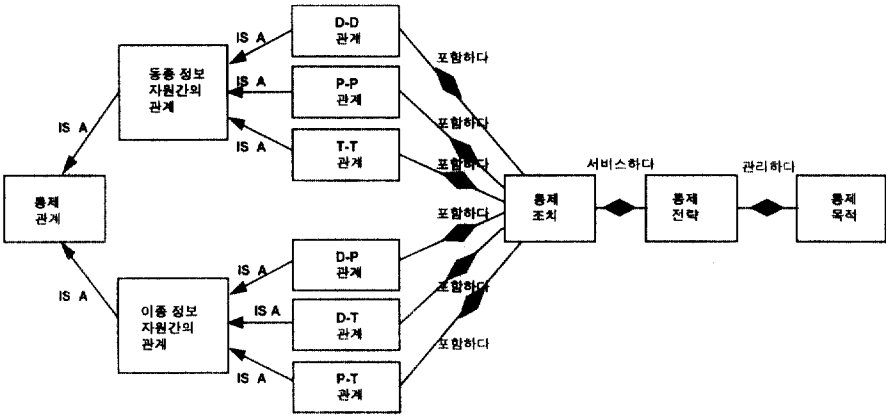
기능부서에서 수행되는 다양한 프로세스 실행자(고객, 소유자, 관리자, 임원등)들에 의해 한개 이상의 직무가 수행된다. 기능부서에 대한 프로세스 통제는 부서의 직무를 감시하고, 비지니스 프로세스를 실행하기 위한 회사 내부의 규칙과 절차에 따른다. 프로세스 실행자에 대한 프로세스 통제는 조직의 성과뿐만아니라 적합한 프로세스역활도 포함된다. 직무에대한 프로세스 통제사례로는 개별 직무의 성과뿐만아니라 직무조정의 효율성 및 효과성 등이 있다.

기술엔티티는 네트워크, 하드웨어 및 시스템 소프트웨어의 하위유형을 가진다. 네트워크는 클라이언트 혹은 서버같은 하드웨어 구성요소를 포함하고, 각각은 다시 하나이상의 네트워크에 포함된다. 동종의 시스템 소프트웨어 역시 하나이상의 네트워크에 설치될 수 있는 반면 각각의 네트워크는 한개 이상의 시스템 소프트웨어에 설치된다. 또한 제약요소 혹은 점검표 형태의 기술 통제하에서 운영된다. 여기서 기술통제의 제약요소는 네트워크를 적절하게 관리하는 기준 혹은 임계값이다. 클라이언트와 서버는 구체화된 성능한계치와 수용능력내에서 운영되며 기술 통제메카니즘으로 감시된다. 운영체제, DBMS, 네트워크 관리시스템과 같은 시스템 소프트웨어는 표준화, 이력관리, 형상관리와 같은 기술통제에 의해 지원된다. 동시에 시스템 소프트웨어는 다른 네트워크를 관리하기 위한 무결성 및 성능검사와 같은 기술통제를 제공한다.

전사적 통제전략은 데이타통제, 프로세스통제 및 기술통제의 도구와 기법같은 통제메카니즘을 활용한다. 이러한 통제전략은 전사적인 통제 목적으로부터 도출되어 지원된다.

### 4.2 논리적 관점에서의 통제관계 아키텍처

통제관계는 하나의 정보자원과 다른 정보자원과의 관련성을 구체화시킨다. 통제관계 아키텍처는




〈그림 3〉 논리적 관점에서의 통제 관계 아키텍처

〈그림 3〉과 같이 두가지의 축으로 구성되는데, 한 축은 정보 자원사이의 관계를, 또다른 한축은 목표 정보자원을 관리하는 통제목적, 통제전략 및 통제 활동등을 표현한다. 통제관계는 동일 정보자원 및 이기종 정보자원 관계의 2가지의 하위유형을 가진다. 동종의 정보자원관계는 데이타-데이타, 프로세스-프로세스 및 기술-기술 관계의 3가지로 구성된다. 여기서 데이타-데이타 관계는 데이타 자원간의 관계로서, 예를 들면 고객들의 주문이 취소되면 관련된 주문 항목이 자동적으로 연쇄 취소되거나 혹은 참조 무결성을 보장하기 위해 취소가 요구되어야 한다. 또 다른 예로서, 주요 키가 취소되면, 해당하는 필드값이 엔티티 무결성을 보장하기 위해 취소가 요구되어야 한다. 프로세스-프로세스관계는 비지니스 프로세스들간의 관계로서 프로세스의 실행이 연속적인 경우는 "연속적", 프로세스흐름이 조건 변수에 따라 변화하면 "선택적", 동일한 프로세스가 반복적으로 실행되면 "반복적"으로 표현된다. 기술-기술관계는 클라이언트및 서버와 같은 기술 구성요소간의 관계로서 예를 들어, 만약 클라이언트가 서버에게 어떤 조치를 요구하면 서버는 클라이언트에 "응답"한다.

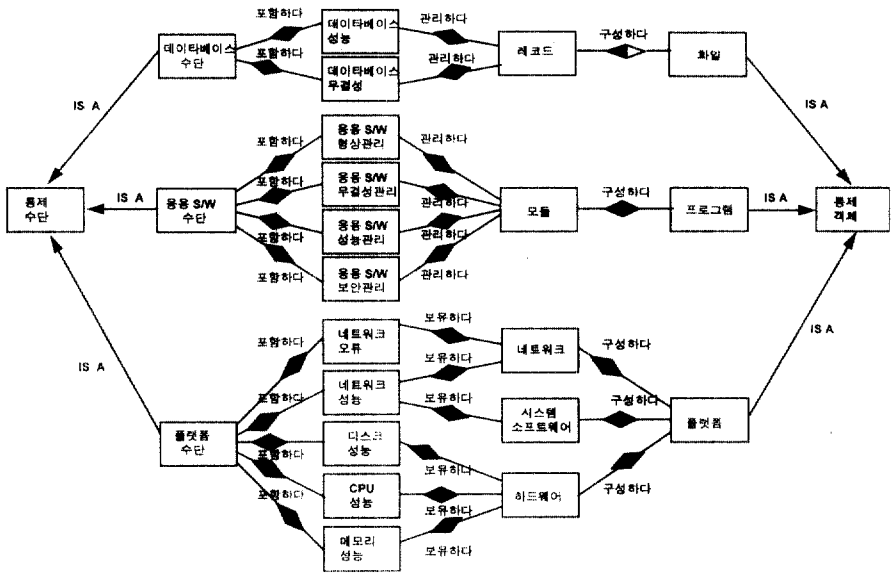
이기종 정보자원간의 관계는 데이타-프로세스, 데이타-기술, 프로세스-기술 관계의 3가지의 하위유형을 가진다. 데이타-프로세스 관계는 데이타와 프로세스간의 관계로서 "만들기(C: Create), 읽기(R: Read), 수정하기(U: Update) 및 삭제하기(D: Delete)"[Martin, 1990]와 같은 4가지 명령어를 사용하여 데이타 아키텍처와 프로세스 아키텍처 사이의 상호관계를 표현한다. 여기서 "만들기"는 지정된 프로세스가 데이타클래스를 창출해내는 반면, "R", "U", "D"는 각각 읽고(Read), 수정하고(Update), 삭제하는(Delete) 조치를 의미한다. 데이타-기술관계는 데이타와 기술사이의 관계로서 어떤 데이타가 해당하는 클라이언트 혹은 서버에 어떻게 저장되고, 어느정도 상호 의존되는가를 보여준다. "단독(Unique)" 데이타베이스는 집중서버에 단지 하나만의 복사본이 저장되고, "복제(Replicated)" 데이타베이스는 하나의 데이타베이스가 복수의 클라이언트 혹은 서버에 복사본이 저장되고, "분할(Partitioned)" 데이타베이스는 하나의 데이타베이스가 분할되어 다른 서버 혹은 클라

이언트에 저장됨을 의미한다. 네트워크 트래픽이 많고, 적고의 결정기준은 하드웨어와 응용소프트웨어종류에 따라 다르다. 프로세스-기술관계는 비지니스 프로세스와 기술사이의 관계로서, 어떤 프로세스를 실행하기 위해 해당하는 어떤 기술요소를 정보자원으로 활용하는가를 표현한다. "저장(Store)"은 프로세스가 해당하는 기술요소에 저장되는 것을, "실행(Execute)"은 목표 프로세스가 프로세서 혹은 시스템 소프트웨어와 같은 기술요소에서 어떻게 처리되는가를 보여주고, "저장 및 실행(Store and Execute)"은 기술 아키텍처의 요소가 해당되는 어떤 프로세스에 저장되고 동시에 실행되는가를 제시해 준다.

각각의 관계는 저장과 감시기능과 같은 하나 이상의 통제 조치를 포함하며, 각각의 통제조치는 하나이상의 관계에 포함된다. 데이타-데이타 관계의 하나의 예로서는 참조무결성을 위한 통제조치는 엔티티간(예: 고객과 주문, 주문과 대금 청구 등) 상호의존하는 모든 데이타 엔티티에 적용된다. 각각의 통제조치는 하나이상의 통제전략으로 관리되는데 그 하나의 예로서, 데이타베이스 참조무결성을 위한 통제조치는 "연쇄삭제(cascade delete)"와 같은 구체적인 통제전략에 의해 도출된다. 각각의 통제전략은 하나이상의 통제 목적을 성취하기 위한 정책 혹은 절차이다. 또 다른 예로서 "95%의 데이타 무결성 유지"의 통제목적은 "연쇄삭제"의 통제전략을 요구한다.

### 4.3 물리적 관점에서의 통제객체 아키텍처

〈그림 4〉와 같이, 통제객체 아키텍처의 물리적 관점은 두축으로 구성되는데 왼쪽의 한축은 통제수단을 표현하고 있는 반면, 오른쪽의 다른 한 축은 목표 정보자원들을 (응용소프트웨어, 데이타베이스, 플랫폼) 모형화 한다. 논리적 관점과 물리적 관점의 주요한 차이는 전자가 좀더 추상적이고, 일




〈그림 4〉 물리적 관점에서의 통제 관계 아키텍처

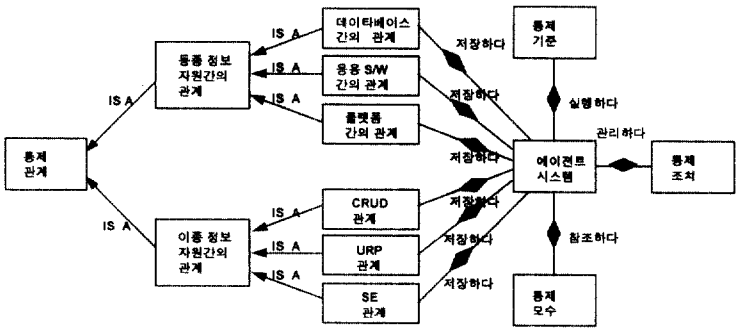
반적인 반면 후자는 좀더 구체적이고 전산화되는 수준에 있다. 만약 통제 객체와 통제 정책이 정해졌다면, 통제수단은 확정적 요소로서 표현되어 데이타베이스 수단, 응용 소프트웨어 수단 및 플랫폼 수단의 3가지 하위유형을 가진다.

예를들어, 시스템 성능개선을 위한 데이타베이스 수단은 최대 프로세스, 블럭크기, 테이블공간, 세그먼트 크기, 캐쉬, 플래그멘테이션 크기, 잠금기능, 데이타할당, 통계, 테이블 분리등 데이타베이스의 환경 모수를 포함한다. 데이타베이스 수단의 주요 범주는 데이타베이스 무결성으로 각각의 데이타베이스 수단은 엔티티 무결성 혹은 참조무결성과 같은 하나이상의 데이타베이스 무결성과 관련를 맺고 있다.

응용 소프트웨어 수단은 응용 소프트웨어 요소, 응용 소프트웨어 이력, 이력관리체제, 모수, 응용 소프트웨어간의 상호의존성과 같은 하나이상의 응용 소프트웨어 형상과 관련된다. 응용 소프트웨어 성능 향상을 위하여 응답시간, SQL사용, 접속 우선순위, 감사궤적, 사용자 권한 등의 응용소프트웨어 수단을 사용한다. 또한 응용 소프트웨어 무결성을 위하여 데이타 효율성등이 포함되고 응용 소프트웨어 보안을 위해서는 경고 메시지 정보제공과 같은 메카니즘과 사용함으로써 오류발견, 오류분리 및 오류해결과 관련된 수단이 포함된다. 형상관리, 무결성관리, 성능관리 및 보안관리의 통제 수단은 응용소프트웨어의 기본 블럭요소로서 프로그램 모듈이 구축된다. 플랫폼 수단은 네트워크 오류, 네트워크 성능, 디스크 성능 및 기억용량의 성능들을 감시하고 통제하는 수단들을 포함한다. 여기서 네트워크 오류방지를 위하여 오류확인, 오류진단, 오류경고 및 오류해결등의 활동을 포함하고, 네트워크 성능 향상을 위하여 네트워크 트래픽, 네트워크 부하, 사용자수, 평균 응답시간과 같은 플랫폼 수단을 사용한다. 플랫폼 수단은 디스크 성능 (예:활용도, 접속률, 입출력 속도 등), 중앙처리장치 성능 (예:평균 활용도, 최대 활용도, 병렬 처리 능력 등), 기억장치 성능 (예: 가용기억용량, 페이징/스위핑율, 기억용량 크기 등)을 관리하기 위해 사용된다. 네트워크, 시스템 소프트웨어, 하드웨어 구성요소들의 다양한 구성요소들로 플랫폼을 형성하며, 물리적 통제 객체의 값은 논리적 수준의 기술 아키텍처 표준에 의해 좌우된다.

### 4.4 물리적 관점에서의 통제 관계 아키텍처

물리적 관점에서의 통제 관계는 논리적 수준보다 하위 수준을 표현하며 〈그림 3〉과 같이 동일한 목표관계를 다룬다. 이 수준에서는 에이전트 기반의 통제 시스템과 같은 구축도구가 동종의 정보자원간 혹은 이종의 정보자원간의 관계를 감시 및 처리하기 위해 사용된다. 이러한 시스템을 통해 통제 데이타가 주기적으로 비교되고, 미리 정해진 임계값을 점검하고, 필요하다면 이벤트 혹은 경고 메세지를 만들어 낸다. 〈그림 5〉에서의 왼쪽은 〈그림 3〉과 동일하고, 오른쪽의 통제 기준은 에이전트 시스템을 위한 표준 값들이 설정된다. 예를들어 클라이언트의 응답시간은 최종 사용자 요구사항에 부합하기 위한 중요한 통제기준으로 정의된다. 통제모수는 에이전트 시스템에 의해 사용되는데 예로서, 클라이언트의 응답시간이 5초 소요된다는 사실은 클라이언트와 서버간의 통제 관계를 구축하기 위해 필요하다. 각각의 에이전트 시스템은 하나이상의 통제 조치를 수행하는데 예로서, 응답시간이 5초이상이면, 에이전트 시스템은 운용자에게 경고 신호를 송부한다.




〈그림 5〉 물리적 관점에서의 통제 관계 아키텍처

## V. 현행 상용화 제품분석

이 장에서는 통제 아키텍처를 지원할 수 있는 세 종류의 상용화된 도구 (COMPTUERWARE-Ecotools, BMC-Patrol, CA-Unicenter)를 〈표 2〉와같이 비교 분석한다. 세종류의 제품 모두는 국제표준기구 (ISO)의 네트워크 관리 표준인 SNMP (Simple Network Management Protocol)로 부터 도출되어 데이타, 응용 소프트웨어, 네트워크 및 프로세서 등의 전사적 정보 자원의 통제를 좀 더 향상시키기 위한 정보자원 관리기능이 제공된다. 세종류의 제품 모두 통합된 정보자원 관리 (IRM) 기능을 제공하지만 일부의 정보자원 기능만이 수행되고, 이 논문에서 제시된 전체적인 정보자원에 관한 통합 통제기능은 수행하지 못한다.

### 5.1 COMPUWARE-Ecotools

Ecotools은 클라이언트/서버 컴퓨팅 환경에서 데이타베이스, 응용소프트웨어, 하드웨어 플랫폼 관리를 감시하고, 자동화하는 제품이다. Ecotools의 관리 아키텍처는 어떤 모수가 한계치 이상으로 이르렀을때는 시스템 관리자에 의해 제시된 구체적인 컴퓨터, 네트워크, 데이타 베이스의 모수들을 동적으로 감시하는 이벤트-액션 메카니즘을 이용한다. 만약 시스템 이상이 있거나 시스템 책임자에게 전자메일을 보내거나, 관리통제판에 경고음을 울리거나 혹은 문제를 해결할 수 있는 사전 프로그램을 가동한다. Ecotools의 특징은 첫째, 시스템 관리자가 클라이언트/서버 환경하에서의 변경을 추적하거나 계속적으로 감시하도록 형상관리 기능이 있고, 둘째, 잠재적 문제 해결을 위해 시스템을 계속적으로 감시하고 자동적으로 예방조치 혹은 수정조치를 취하는 오류관리 기능이 있고, 셋째, 컴퓨팅 환경의 모든 관점으로부터 실시간으로 적합한 데이타를 수집 혹은 저장하는 성능관리기능과 상세한 분석을 위한 그래프형태로 표현하는 기능, 넷째, 시스템의 가용성 혹은 성능에 미치는 이벤트 추적을 수작업이 아닌 자동화된 수정조치 기능등이 있다.

통합 통제아키텍처와의 관계 : Ecotools은 RDBMS와 네트워크를 사용한 분산 데이타 자원을 관리한다. Ecotools의 하나의 모듈인 EcoPMON 역시 사용자들이 관계형 데이타 베이스를 접속할

수 있는 SQL 명령문을 수행하는 프로세스 감시기능을 제공한다. Ecotools은 추가적인 SNMP 플랫폼으로 패킷 개수및 빈도의 확인기능을 제공함으로써 기술자원을 관리한다. 〈그림 2〉와같이 Ecotools에 의해 수행하는 다양한 기능들은 데이타 통제, 프로세스 통제, 기술 통제로 분류될 수 있다. 이 기능들은 데이타 통제 수단, 프로세스 통제수단 및 기술 통제수단을 포함하는 개별적 정보자원 관리 조치들을 포함한다. Ecotools이 분산 아키텍처의 개별적 정보자원의 데이타 수집 및 이벤트 감시 기능을 제공하지만, 정보자원 변경에 따른 통제 관계의 영향 분석 기능을 제공하지 못한다.

### 5.2 BMC-Patrol

Patrol은 Ecotools과 유사하게 분산 컴퓨팅 환경하에서 데이타 베이스와 응용 소프트웨어를 자동적으로 감시 및 관리한다. Patrol의 아키텍처는 네트워크에 대한 지속적인 수요 혹은 생산 시스템에 대한 지속적인 감시를 제공한다. Patrol은 수작업의 수정조치를 수행하는 관리 통제장치, 서버, 데이타 베이스, 응용 소프트웨어 및 다른 정보 자원을 추적하는 관리 에이젠트와 가용한 전문지식을 저장해 놓은 지식 모듈이 있다.

통합 통제 아키텍처와의 관계 : 통제 장치와 에이젼트는 〈그림 4〉와 같이 다양한 플랫폼 조치에 대한 이벤트/수정 조치를 수행한다. 지식 모듈은 성능 향상을 위한 데이타 베이스 수단과 플랫폼 수단을 지원하고, 데이타 베이스와 플랫폼의 무결성을 보장한다. Ecotools의 SQL 명령어가 수행시간을 포괄하지는 않지만 독립 모듈의 통제기능을 수행한다. Patrol은 통제관계 아키텍처를 지원할 뿐만 아니라 개별적 정보 자원의 관리기능도 제공한다. Patrol이 〈그림 5〉와 같이 에이젼트 시스템을 이용한 통제 모수와 통제 기준을 고려하더라도 통제관계를 나타내는 구조에 현행의 통제 시스템이 포함되지 않아 영향 분석을 수행하지 못한다.

### 5.3 CA-Unicenter

CA-Unicenter는 이기종 시스템간의 공통 관리 모드를 제공하는데, 이것은 분산화된 지역 시스템 관리에 매우 중요하다. 이것은 집중 컴퓨터 시스템이 위치한 장소의 시스템 관리자에 의해 수행되는 계정, 생산, 보안, 성능및 저장관리 기능을 제공한다 [Jones, 1995]. CA-Unicenter는 5가지의 주요 영역에서 근본적인 요구사항을 만족시키는 통합 시스템이다. 첫째, 보안, 통제 및 감사 (SCA) 기능은 시스템 접근 타당성, 자원 및 설비 접속 통제, 사용자 등록, 사용자와 자원의 감사성, 시스템의 무결성을 다룬다. 둘째, 자동 저장 관리 기능(ASM)은 백업, 저장, 재저장, 복구, 운영, 이동, 감시및 보류 기능을 포함하는 데이타 수명주기의 확장된 관리 요구사항을 다룬다. 셋째, 자동화 생산통제(APC) 기능은 부하균형, 배치 작업관리, 통제 장치관리 및 보고서 관리의 영역을 다룬다. 넷째, 성능관리 및 계정관리(PMA)는 데이타 센타가 시스템 성능을 향상시킬 수 있는 자원 계정, 요금 및 시스템 성능 감시기능을 수행한다. 다섯째, 데이타 센터관리기능은 (DCA) 도움말 및 문제관리 영역을 다룬다.

통합 통제아키텍처와의 관계 : Unicenter는 오류관리, 성능관리, 계정 및 보안관리, 형상관리 및 운용관리등 대부분의 SNMP 표준기능등을 수행한다. PMA와 DCA는 〈그림 4〉와 같은 데이타 베이스 수단을 수행한다. SCA, ASM 및 APC 기능은 플랫폼 수단을 제공하는 반면 PMA의 한 부분은

〈표 2〉 상용화 제품의 비교분석

| | COMPUWARE-<br>Ecotools | BMC-<br>Patrol | CA-<br>Unicenter |
|---|---|---|---|
| 특징 및<br>기능 | - 이벤트/액션 메커니즘<br>- 형상관리,오류관리<br>성능관리, 수정조치- | - 관리 에이전트<br>- 지식모듈 내장 | - 보안통제 및 감시(SCA)<br>- 자동저장관리기능(ASM)<br>- 자동화생산통제기능(APC)<br>- 성능및계정관리기능(PMA)<br>- 데이터센터관리기능(DCA) |
| 통제아키텍처와<br>의 관계 | -프로세스감시기능<br>-개별정보자원(데이터,<br>프로세스, 기술) 통제 기능<br>-정보자원간 영향분석<br>기능 미수행<br>-통제관계아키텍처 미지원 | -독립모듈의 통제 기능<br>-개별 정보자원 관리 기능<br>-정보자원간 영향<br>분석기능 미수행<br>-통제관계 아키텍처 미지원 | -부분적 개별정보자원관리 기능 제공<br>-영향분석기능의 미수행<br>-통제관계 아키텍처의 미지원 |

부분적인 응용 소프트웨어 수단을 제공한다. 그러나, Unicenter는 네트워크 성능 및 데이타 베이스에 설계 노력을 집중시켰기 때문에 그림 4에 나타난 바와 같은 응용 소프트웨어 수단을 전적으로 지원하지는 않는다. 그것은 〈그림 5〉와 같은 통제 관계 아키텍처 역시 제공되지 않는다. 그러므로 Unicenter는 영향분석 기능이 제공되지 않는다.

## V. 결론 및 향후 연구 방향

이 논문에서는 데이타, 응용 소프트웨어, 하드웨어, 네트워크 등과 같은 다양한 정보자원들을 관리하기 위하여 통합된 통제 아키텍처의 기본적인 구성틀이 제시되었다. 통합 통제 아키텍처는 전사적인 정보시스템 아키텍처를 개발 및 관리하기 위한 활발한 연구흐름에 기반을 두고 있다. 기존의 정보시스템 아키텍처와 다른 통합통제 아키텍처의 특징은 정보시스템 아키텍처를 논리적 수준과 물리적 수준의 2가지 수준으로 분류한 점과 집중 컴퓨팅 환경처럼 다른 정보자원을 관리 및 지원하는 메카니즘을 채택했다는 점이다. 데이타, 프로세스, 기술등의 논리적 비지니스 수준과 데이타베이스, 응용소프트웨어, 플랫폼의 물리적 수준을 통합함으로써, 전산화된 정보자원뿐 아니라 상위의 관련 비지니스 자원을 효과적으로 관리할 수 있다. 시간경과에 따른 정보자원간 관계를 통제함으로써, 통합 통제 아키텍처는 정보자원의 "동적이고 통합된" 통제를 제공한다. 향후 통합 통제 아키텍처는 지능형 에이전트 서비스로 확장될 것이고, 이 논문에서 언급된 통합된 통제 아키텍처의 특성과 능력을 보여주기 위해 프로토타입 시스템을 구축해야 할 것이다. 현재의 상용화된 시스템 관리 도구들은 통합 통제기능을 제공하지 못하므로 향후에 통합 통제 아키텍처의 구성틀은 지능형의 에이전트 서비스로 확대하여 실험적인 프로토타입 시스템이 이 논문에서 제시된 통합 통제 아키텍처의 기능 및 특성이

구현할 수 있도록 구축되어야 할 것이다.

## 참 고 문 헌


King, W.R (1978), "Strategic Planning for Management Information Systems," *MIS Quarterly*, Vol.2, No.1, 27-3

Sowa, J.F and. Zachman, J.A (1992), "Extending and Formalizing the Framework for Information Systems Architecture," *IBM Systems Journal*, Vol.31, No.3, 590-616

Bauer, M. A., Finnigan, P. J., Hong, J.W., Rolia, J.A., Teorey, T.J. and Winters, G.A. (1994), "Reference Architecture for Distributed Systems Management", *IBM Systems Journal*, Vol.33, No.3, 428-444

Brancheau, J.C., Janz, B. D. ad Wetherbe, J. C. (1996), "Key Issues in Information System Management : 1994-95 SIM Delphi Results," *MIS Querterly*, 225-242

Date, C.J. (1995) *An Introduction to Database Systems*, Addison-Wesly, Sixth Edition,

Eisenhardt, K.M. (1985), "Control : Organizational Economic Approaches," *Management Science*, Vol. 31, No 2, 134-149

Everden, R. (1996), "The Information Framework", *IBM Systems Journal*, Vol.35, No.1, 37-68

Flamholtz, E.G., Dasand, T.K., Tsui, A.S. (1985), "Toward an Integrative Framework of Oganizational Control," Accounting Organizations and Society, Vol. 10, No. 1, 35-40

Frank, M. (1993), "Modeling Transaction Integrity", *DBMS*, pp.62-67

IBM Corporation, (1978) " *Business Systems Planning : Information Systems Planning Guide*," Publication No. GE 20-0527-4, Armonk, New York,

Jones, K. (1995), "Managing Diverse Environment with CA-UNICENTER for UNIX", *International Journal of Network Management*, 138-156

Kim, Y. and Everest, G. C. (1994), " Building an IS Architecture : Collective Wisdom from the Field," *Information and Management*, Vol. 26, No.1, 1-11

Lefkovits, H.C. (1991) *IBM's Repository Manager/ MVS Concepts, Facilities, and Capabilities*, QED Technical Publishing Group

March, S. T. and Kim, Y. (1988) "Information Resource Management : A Matadata Perspective," *Journal of Management Information Systems*, Vol.5, No.3, 5-18

Martin, J. (1990) Information Engineering , Vol. II : Planning and Analysis, Englewood Cliffs

Myers, G.J. (1979) *The Art of Software Testing*, Wiley

Niederman, F., Brancheau, J. C. and Wetherbe, J. C. (1991), "Information System Management Issues in the 1990s," *MIS Quarterly*, 475-500

Ouchi, W.G. (1977), "The Relationship between Organizational Structure and Organizational Control," *Administration Science Quarterly*, Vol.22, No.1, 95-113

Ouchi, W. G. and Maguire, M. A. (1975), "Organizational Control : Two Fictions," *Administration Science Querterly*, Vol. 20 559-569

Reeves, T.K. and Woodward, J. (1970), "*The Study of Managerial Control*," *Industrial Organizations : Behavior and Control*", Oxford University, London, England, 37-56

Scheer, A. W. (1992) *Architecture of Integrated Information Systems : Foundations of Enterprise Modeling*, Springer-Verlag

Sowa, J.F. and Zachman, J. A. (1992), "Extending and Formalizing the Framework for Information

Systems Architecture," *IBM Systems Journal*, ol. 31, No.3, 590-616

Teory, T. J., Yang, D. and Fry, J. P. (1986), "A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model", *Computing Surveys*, Vol.18. No.2, 197-222

Thompson, J.P. (1967) *Organizations in Action*, MaGraw-Hill, New-York

Umar, A. (1993) *Distributed Computing and Client-Server Systems*, Prentice-Hall

Weber (1947), *The Theory of Social and Economic Organization*, Free, Glencoe, IL.

Zachman, J.A. (1987), "A Framework for Information Systems Architecture," *IBM Systems Journal*, Vol.26, No.3, 276-292

# Metadata Model of Integrated Control Architecture : Information Resource Management Pespective

Jeong-Wook Kim* · Young-Gul Kim**

Abstract

Since management of information resources is getting more complicated in the distributed, heterogeneous computing environment, the capability of monitoring and controlling the dispersed information resources is perceived as a critical success factor for the effective enterprise-wide information resource management. Integrated Control Architecture(ICA) provides that capability. Utilizing such architecture, we can manage corporate information resources more efficiently, perform impact analysis for changes in information resources, and alleviate the human effort by automating the monitoring of critical information resources. In this paper, we propose a conceptual framework and metadata model of ICA.

* Department of Management Information Systems, Dongyang Technical College
** Graduate School of Management, Korea Advanced Institute of Science and Technology