

CBD 기반의 CIS 구성에 관한 연구

김 행 곤[†] · 신 호 준^{††}

요 약

최근 표준화 컴포넌트 기반 방법론이 대두되면서, 객체지향에서 컴포넌트 지향의 시스템으로의 전환이 빠르게 진행되고 있다. 특히, 비즈니스 영역에서 컴포넌트 기반 시스템이 성공을 거두면서 컴포넌트의 중요성과 추진효과의 신뢰성이 증가하고 있다. 현재 방법론의 실무 적용 단계에 이르고 있으나, 아직도 분산환경과 프로세스 및 컴포넌트의 COTS(Commercial Off The Shelf)를 이루기 위해서는 기반 환경뿐만 아니라, 프로세스 관리와 이를 지원하는 도구 및 유지 보수 등 다양한 서비스의 통합이 요구되고 있다. 본 논문에서는 컴포넌트 기반 방법론을 지원하기 위해 제시된 상호 운용 정보시스템을 프레젠테이션 계층, 비즈니스 로직 계층, 데이터 제어 계층의 3계층으로 구성하여 잠재적인 사용자의 요구를 반영하고자 한다. 각각의 계층별 행위와 기능의 상세한 명세와 핵심적인 계층인 비즈니스 로직의 기능 및 관리되어야 할 정보를 기술한다. 상호 운용시스템을 구성함으로써, 비즈니스 영역에서의 컴포넌트 재사용성과 효율성 및 컴포넌트 개발과 조립에 대한 사용자의 결정에 도움을 줄 수 있다.

A Study on the Construction of CIS(Cooperative Information System) based on CBD

Haeng-Kon Kim[†] · Ho-Jun Shin^{††}

ABSTRACT

In recent years, we recognize a new paradigm in the development process : From object oriented development, to the development process which has focused on the use of standard components. In recent years a lot of research related to the component-based development has been done, especially in business domain. but still there are many open and unresolved problems in this area. such as established development process for a distributed environment, formal process, infrastructure for COTS, development and management tool considering maintenance to guarantee a proper treatment of components. It also required a service and an application integration for component. In this paper, we propose cooperative information systems (CIS) that supports component based development. It must address for component based systems to achieve their full potential. We identify a set of CIS organized 3-tier which is a presentation layer, business logic layer and data control layer. We also discuss the specific roles and activities for the layers. we also define the behavior and managed information for business logic layer as core level. As an illustration of the CIS, we present a successful considerations which is widely helpful to user when they make decision in component development and assemble. Also, we expect to component reusability and efficiency in business domain.

키워드 : 상호운용 정보시스템(Cooperative Information System), CBD(Component Based Development), 재사용

1. 서 론

현재 애플리케이션 서버 개발의 컴포넌트 기술과 표준들인 CCM(CORBA Component Model), COM+(Component Object Model)와 EJB(Enterprise JavaBean) 같은 컴포넌트 모델의 소개는 비즈니스 솔루션의 변형을 가져왔으며, 기업마다 컴포넌트 기반의 프로세스 및 개발지원을 위한 기술을 개발하고 있다. 현재 비즈니스 영역에서도 금

유부분은 컴포넌트 기반 프로세스의 적용과 파일럿 시스템의 개발로 인해 CBD(Component Based Development) 방법론의 산업화가 가속화되고 있다. 현재 방법론의 실무 적용 단계에 이르고 있으나, 아직도 분산환경과 프로세스 및 컴포넌트의 COTS를 이루기 위해서는 기반 환경뿐만 아니라, 프로세스 관리와 이를 지원하는 도구 및 유지보수 등 다양한 서비스의 통합이 요구되고 있다[1].

본 논문에서는 컴포넌트 개발과 개발된 컴포넌트의 조립, 기존 애플리케이션이나 시스템에 기능 추가를 위한 컴포넌트의 적용 및 필요한 컴포넌트 검색과 관리 등 컴포넌트 개발에서 유통에 이르는 행위와 관련된 사용자의 요구 사항을 반영한 서비스를 제공하기 위해 상호 운용 정보시

* 이 논문은 2001년 대구가톨릭대학교의 교비해외파견 연구 지원금에 의한 것임.

† 종신회원 : 대구가톨릭대학교 컴퓨터정보통신공학부 교수

†† 준회원 : 대구가톨릭대학교 대학원 전산통계학과
논문접수 : 2001년 10월 5일, 심사완료 : 2001년 12월 13일

시스템(CIS : Cooperative Information System)을 구성하고자 한다. CIS 애플리케이션 개발 환경은 새로운 컴포넌트 모델을 개발하고 다른 애플리케이션과 통신가능하고 투명한 방법으로 정보를 공유하기 위해 컴포넌트 아키텍처에 기반한다. 컴포넌트 개발에 대한 요구사항 도출 과정에서부터 개발된 컴포넌트를 적절한 시스템에 적용하는 과정까지 수반되는 처리를 통합된 형태로 제공하기 위한 서비스를 계층별로 구성한다. 이를 위해 비즈니스 영역에서 필요로 하는 컴포넌트의 생성과 관리 및 재사용을 위한 각각의 행위를 지원하기 위해서 계층별 기능에 대한 정의와 관련 기술의 요구사항을 추출한다. 또한, 추출된 산출물과 전체 시스템의 모델링을 통해 상호 운용 정보시스템을 구성하기 위한 컴포넌트 식별이 수반되어야 한다. 각 계층별 기능과 처리 절차의 정의를 통해 적절한 시스템을 구성함으로써, 비즈니스 영역에서의 컴포넌트 재사용성과 효율성을 기대한다.

2장에서는 본 논문에서 다루어질 통합된 상호 운용환경과 관련된 애플리케이션 통합과 컴포넌트의 관리 및 재사용에 대해서 살펴보고, 3장에서는 CIS에 대한 정의와 계층별로 제공되는 서비스에 대해서 기술하며, 4장에서는 CIS의 핵심적인 비즈니스 로직부분의 상세한 구성내용을 살펴보고, 끝으로 결론 및 향후 연구방향을 제시한다.

2. 관련 연구

2.1 EAI(Enterprise Application Integration)

EAI는 공통의 미들웨어를 사용하여 애플리케이션들을 통합하여 비즈니스 솔루션을 만들어 내는 것이다. 또한, 커스텀 개발 및 패키지형 업무용 애플리케이션이 필요한 정보를 모두 이해하는 포맷과 문맥으로 변환할 수 있도록 지원하는 기술과 프로세스의 결합이라 할 수 있다. 상호 이질적인 애플리케이션과 데이터들을 일관성 있게 통합 관리하고, 막대한 시간과 비용을 소모하는 추가 프로그래밍이나 변환 작업 없이 기업내 모든 애플리케이션을 통합할 수 있는 장점이 있다[2].

업무처리 전반을 지원하는 다양한 애플리케이션을 통합된 형태로 지원하는 EAI 시스템 개발에서도, 컴포넌트 기반으로 시스템을 구축하고자 한다. 이를 실현하기 위해 다양한 비즈니스 영역에서 컴포넌트로 프로젝트를 패키징하는 기술을 개발하고 있으며, 컴포넌트 기술을 사용하여 비즈니스 솔루션을 개발하고, 고객을 지원하고자 한다.

IBM에서는 EAI 솔루션 컴포넌트를 이용하여 경쟁력 향상과 프로세스를 체계화하고 최적의 EAI 설계 및 구현을 위해 아래 (그림 1)과 같은 서비스 방법을 개발하였으며, 각 단계는 필요한 행위(Activity)와 작업(task) 및 산출물로 구성되어 있다[3].

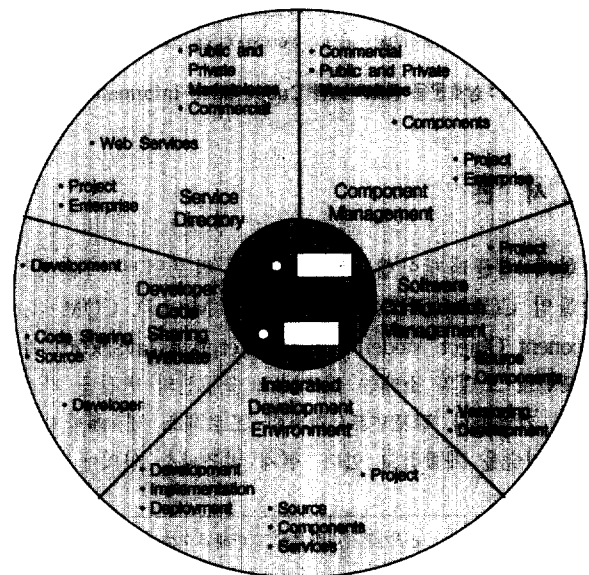


(그림 1) EAI 서비스 프로세스

2.2 컴포넌트 관리와 재사용

재사용의 주요한 형태로 컴포넌트가 가장 효율적인 자원으로 받아들여지고 있으며, 여러 도메인에서 적용가능한 컴포넌트들이 제공되고 있다. 소프트웨어 개발에서 중요한 요소로 고려되는 재사용성과 컴포넌트의 관련성은 다음 (그림 2)와 같이 컴포넌트 관리와 재사용의 관점에서 제공되는 솔루션을 고려할 수 있다[4].

- Developer Code Sharing Website : 광범위한 영역의 웹 사이트를 통한 개발자들의 코드 공유가 가능하며, 제일 일반적인 형태의 재사용 메카니즘으로 간주된다.
- Integrated Development Environment : 대형 소프트웨어 개발이 가능한 IDE는 브라우저와 카탈로그 기능을 통한 개발되는 소프트웨어의 관리를 지원하며, IDE 자체에 포함된 컴포넌트를 제공받을 수 있다. 하지만, Microsoft 환경에서 EJB를 지원하기 어려운 것처럼 IDE에 종속적인 기능에 제약이 따른다.
- Software Configuration Management : IDE에서 최소한의 개발환경을 제공하는 제약에 비해, 목적에 맞는 형태로 컴포넌트의 카탈로그를 정의하고 버전관리, 작업관리, 결함관리 등을 포함한다.
- Component Management : 개발된 컴포넌트의 in-house 형태의 전통적인 관리와 웹을 기반으로 배포를 목적으로 하는 관리로 나눌 수 있다. 이는 컴포넌트 시장을 형성하게 되며, 사용자 관리와 호스트 관리를 수반하게 된다.



(그림 2) 컴포넌트 관리와 재사용성

- **Service Directory** : 서비스 재사용 측면에서 디렉토리를 사용하는 솔루션으로써, 일반적으로 UDDI (Universal Description Discovery and Integration)는 카탈로그를 위한 인프라를 제공하고, SOAP(Simple Object Access Protocol)을 기반으로 하는 웹 서비스를 제공하게 된다. 수직적인 도메인에서 특정 서비스를 제공받기 위해 등록된 개발자나 고객 등에게 웹서비스 메커니즘을 제공한다.

3. CIS의 계층별 서비스

현재 조직과 지역에 상관없이 개방적인 방식의 재사용성을 강조하고, 기존에 존재하는 시스템을 활용할 수 있는 기술과 새로운 공급 사슬을 통한 외부적인 서비스들을 확대할 수 있도록 비즈니스 영역이 바뀌고 있다. 특히, 컴포넌트를 기반으로 하는 기술과 환경들은 하나의 독립된 형태가 아니라 상호 연관된 서비스로써 다양한 사용자의 요구사항을 만족시켜야하며, 개발에 따르는 비용과 시간, 컴포넌트가 가지는 재사용성을 최대로 제공하기 위해서는 유기적인 형태로 통합되어야 한다. 또한, 컴포넌트를 지향하는 방법론을 위한 개발 환경과 이를 관리하고 지원하기 위한 리파지토리, 상용 서비스들이 일원화되고, 자동화된 형태로 서비스가 제공되어야 하며, 이를 지원하는 통합환경과 상호 운영 정보 시스템이 구성되어야 한다.

따라서, 컴포넌트를 중심으로 공유되어야 할 정보와 이를 지원하기 위한 서비스에 대한 분석을 통해 통합된 계층을 형성해야 하며, 다양한 사용자를 지향하는 프로세스의 통합과 컴포넌트 지향 프로세스를 지원할 수 있는 도구들의 통합이 이루어져야 한다. 이러한 서비스를 유기적, 효율적으로 관리하기 위한 리파지토리 시스템과 유통 시스템의 연관성을 정의하며, 통합된 아키텍처에서 컴포넌트 및 애플리케이션

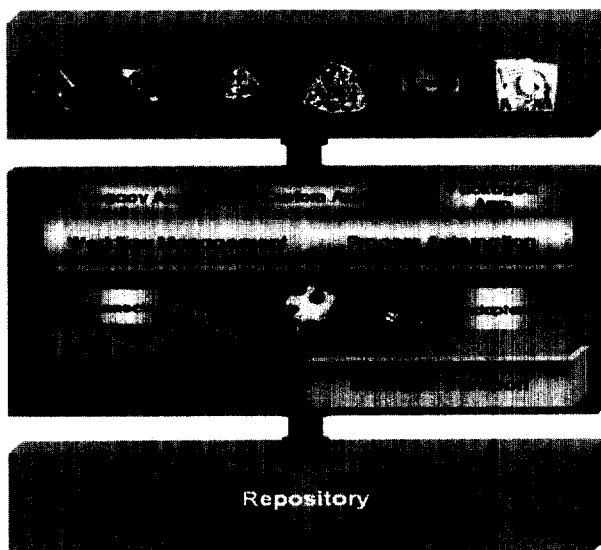
개발자와 컴포넌트 단순 사용자의 요구사항을 만족해야 할 것이다. 이를 지원하는 관리자 및 실행환경 등을 고려하여 다음 (그림 3)과 같이 구성할 수 있다.

CIS는 컴포넌트의 개발과 적용을 위한 다양한 사용자 중심의 프레젠테이션 계층과 외부 사용자에게 서비스 제공과 컴포넌트 기반 개발 지원을 위한 핵심적인 비즈니스 로직 계층과 통합된 정보의 관리와 컴포넌트에 수반되는 다양한 산출물을 관리하는 데이터 제어계층의 3-tier 환경을 모델로 하고 있다.

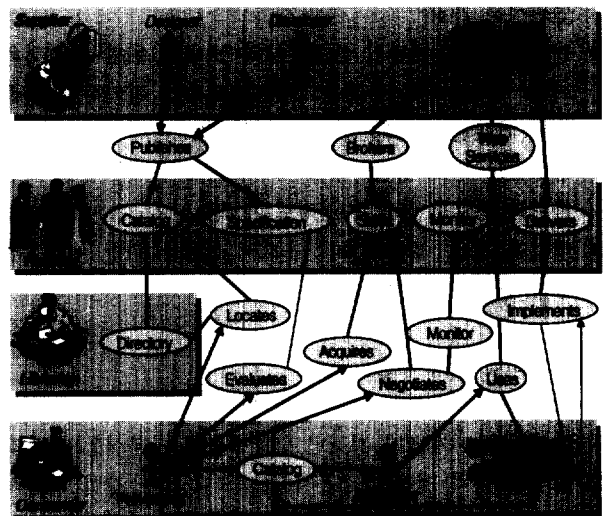
3.1 프레젠테이션 계층

통합된 아키텍처 상에서 발생하는 구현물과 관련된 정보는 다양한 사용자가 일반적으로 저장된 정보를 검색하고 유도함으로써 이용된다. 또한, 컴포넌트의 개발과 이미 개발되어진 컴포넌트의 획득, 이를 지원하는 서비스 단계에서 정보는 비동기적인 형태로 사용자에게 획득된다. 또한, 생명주기의 각 단계에서 이루어지는 다른 타입의 컴포넌트 정보는 다른 레벨에서 요구되어진다. 컴포넌트의 개발과 이를 기반으로 하는 애플리케이션의 구현을 위한 개발 프로세스에서 수행되는 행위는 생명주기에서 크게 컴포넌트를 제공하는 측면과 이를 사용하는 측면 즉, 생산자와 소비자로서 식별 할 수 있다.

이는 컴포넌트 개발을 위한 명세의 생성, 개발 그리고 배포의 제공자 행위와 요구되는 컴포넌트의 목록 작성 및 위치 식별, 컴포넌트의 평가, 구현 및 구현된 산물의 사용단계로 구성된다. 컴포넌트 생명주기의 생산자, 소비자 측면과 수반되는 행위적인 측면은 비즈니스 영역과 연관지을 경우에 세분화된 사용자와 행위들을 식별 할 수 있다. 이는 서비스를 제공하는 측면을 포함하여 다음 (그림 4)와 같이 나타낼 수 있다.



(그림 3) CIS 구성



(그림 4) 연관된 사용자 및 행위

- **컴포넌트 제공자** : 애플리케이션에 적용될 컴포넌트의 구현을 위해 설계와 구축 및 기본적인 테스트를 수행함.
- **컴포넌트 소매업자** : 개발된 컴포넌트를 배포하기 위해 자체 혹은 의뢰를 통한 테스트와 효율적인 분배를 위한 분류 및 카탈로그의 작성과 공유 리퍼지토리에 컴포넌트의 등록, 갱신 및 관리를 함.
- **컴포넌트 라이브러리언** : 컴포넌트 유통업자와 컴포넌트 개발자들의 위치 파악을 함.
- **컴포넌트 소비자** : 전체 조직이나 개인적인 프로젝트를 위해 요구되는 컴포넌트의 식별과 획득 작업을 하며, 컴포넌트 구현에 의해 제공되는 서비스사용 및 기존 혹은 새로운 시스템을 위한 컴포넌트 조립을 수행함.

3.2 비즈니스 로직 계층

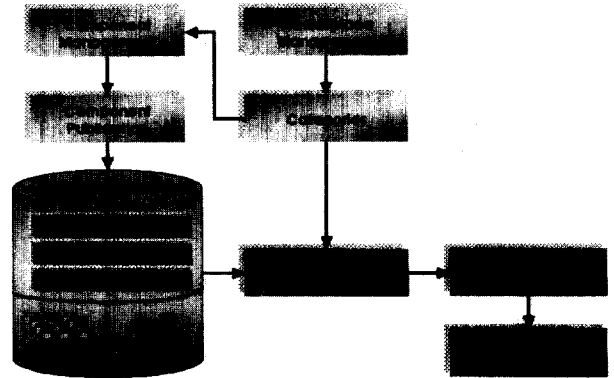
시스템 내외적으로 제공되는 핵심적인 기능을 수행하는 부분으로써 대상 애플리케이션과 비즈니스 영역의 작업흐름을 통해 개발된 컴포넌트와 적용가능한 서비스를 지원하며, 적절한 컴포넌트의 메타정보를 정의함으로써 컴포넌트 기반 개발이 지원 가능하다. 세부적인 내용과 추가적으로 고려되어야할 비즈니스 로직부분의 기능들은 다음과 같다.

- **기존, 고객, 패키지화된 애플리케이션** : 개발되거나 적용될 기존의 애플리케이션과 사용자 애플리케이션이 위치하며, 컴포넌트의 패키지화된 애플리케이션이 위치
- **작업흐름 관리** : 비즈니스 프로세스 내에 컴포넌트의 개발, 획득, 적용을 위한 작업흐름 개발과 실행을 관리함.
- **프로세스 자동화** : 컴포넌트 개발과 컴포넌트 기반의 애플리케이션을 구성하기 위한 시스템과의 프로세스의 흐름과 자료 처리를 수행함.
- **컴포넌트 개발/조립** : 분석, 설계된 컴포넌트를 개발하여 조립을 하기 위한 계층으로 틀과 컴포넌트 실행 환경과 연관성을 가짐
- **연결/어댑터** : 개발된 컴포넌트의 조립을 위한 커넥터와 기존 시스템과의 연결을 위한 어댑터를 정의하며 이를 통해 컴포넌트 간의 연결성을 확보함.
- **컴포넌트 정보 통합** : 컴포넌트의 배포와 재사용을 위한 관련된 정보를 통합된 형태로 제공함.
- **메시지 전송** : 컴포넌트의 유도과 사용을 위한 메시지전송과 관리 서비스를 제공함.

3.3 데이터 제어 계층

사용자에게 제시되어야될 내용과 핵심적인 비즈니스 로직을 지원하고 개발된 컴포넌트와 수반되는 부산물의 관리를 위해 데이터 제어 계층은 다음 (그림 5)과 같이 리퍼지토리와 관리 영역으로 설명할 수 있다. 리퍼지토리에 컴포넌트의 카탈로그 정보와 컴포넌트 분석, 설계 모델, 문서

자료 및 바이너리 형태의 실행가능한 컴포넌트와 소스코드가 저장되고 이를 관리하기 위한 도구가 위치한다.



(그림 5) 컴포넌트 데이터 제어

- **Component Management** : 산출물간에 공유되어야 할 정보와 유지관리 해야할 구성요소를 포함하고 있다. 기본적으로 가격 정보와 컴포넌트의 이해를 위한 기능적인 명세, 기술 문서, 도움말, 사용자 가이드와 같은 다양한 자원 및 인스턴 가능한 컴포넌트 바이너리 파일을 배포하기 위한 준비단계로써 관리되어진다.
- **Component Publication** : 컴포넌트 시장에서 제공될 카탈로그와 그 내용 관리가 주요 목적이다. 컴포넌트 출판에서는 사용자에게 활성화 될 컴포넌트 카탈로그와 준비작업을 위한 카탈로그로 나누어 관리하며, 이는 변경에 융통성 있게 대처하는 장점을 제공한다.
- **Marketplace Management** : 컴포넌트 분류를 위해 사용되는 카테고리의 관리를 위한 부분으로 계층적인 구조로써 시장 그룹을 가지며, 정보통합 부분에서 제시되는 다양한 비기능적인 요소에 기반하여 구성된다. 또한, 서비스에 접근하는 사용자에게 인증을 허가하고 올바른 접근을 유도하도록 제시되며 유지 보수된다.
- **Component Search** : 컴포넌트 유통사이트에서나 조직내부의 저장소에서 요구되는 컴포넌트의 탐색과 평가를 지원하기 위한 서비스이다. 이는 카테고리를 기반으로 패킷 검색, 디렉토리 검색, 키워드 검색 방식 등으로 사용자가 요구하는 컴포넌트를 유도 가능하며 검색 기준에 대한 부분도 제시된다.
- **Component Purchase** : 컴포넌트 유통사이트에서 사용자가 요구하는 컴포넌트의 주문과 구매가 이루어지는 부분이며, 사용자의 계정 관리뿐만 아니라, 구매에 따르는 정보를 모니터링하고 사용 보고서를 제공한다.
- **License Management** : 구매에 의해 획득된 컴포넌트에 대한 계정을 제어하고 관리하게된다. 특정 라이선스에 대한 제어와 개인과 단체에 대한 라이선스 정책을 적용하여 인증서 발급도 가능하다.

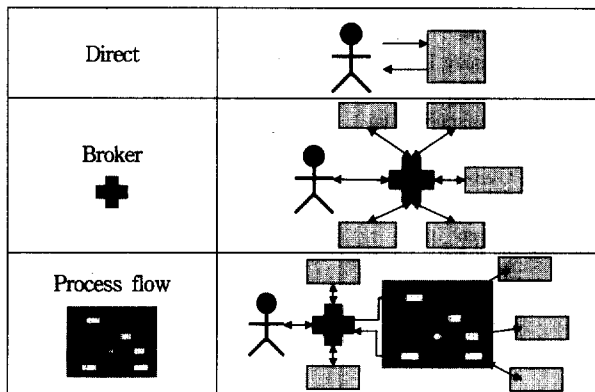
4. CIS 핵심 계층 정의

4.1 워크플로우 관리

통합된 환경에서는 컴포넌트 개발과 배포 등의 각각의 서비스를 서로 연결해 줌으로써 개발과 재사용에 대한 많은 시간을 줄일 수 있으며, 워크플로우 기술과 자동화 메커니즘은 커스텀 인터페이스 방식에 관계없이, 좀더 효과적으로 프로세스를 관리하고, 소프트웨어를 개발하고 검색하는 시간을 단축시키는 주요 수단이 된다. 즉, 생산성 수준 향상, 사용자 만족, 컴포넌트 적용 시간 감축, 수익성 증가 등을 그 효과로 꼽을 수 있다. 비즈니스 프로세스의 흐름을 파악하여 애플리케이션 개발에 컴포넌트를 구성해야 할 내용과 수반되는 정보를 공통 언어와 일련의 운영 규칙들로 상호 운용되는 사용자들간의 워크플로우 솔루션의 상호 운용성을 향상시키는 이점을 제공한다. 공통으로 사용하는 동적인 비즈니스 개발 언어의 사용을 촉진시키고 비즈니스 프로세스 사양을 더 많이 공유할 수 있다. CIS에서 비즈니스 영역에서의 워크플로우 관리는 다음과 같은 이점을 가진다.

- 컴포넌트 사용자(소비자, 개발자, 유통업자, 관리자 등) 간의 정보 교환 및 비즈니스 프로세스의 관리, 조정
- 컴포넌트 개발자와 파트너간의 공유가능한 작업 프로세스 자동화 지원가능
- 프로세스의 능률을 높이고 문서 작업과 검색 비용을 감소
- 공통의 지식기반, 프로세스, XML 형태의 비즈니스 정보 교환 언어 공유
- 비즈니스 프로세스뿐만 아니라 컴포넌트 작업량 예측 및 계획, 전반적인 운영상의 능률화를 위한 관리 및 제어를 향상시켜주는 통계 포착

CIS에서 애플리케이션간의 비즈니스 흐름의 형태를 다음(그림 6)과 같이 개괄적으로 세 가지 형태로 정의한다.



(그림 6) 비즈니스 흐름의 형태

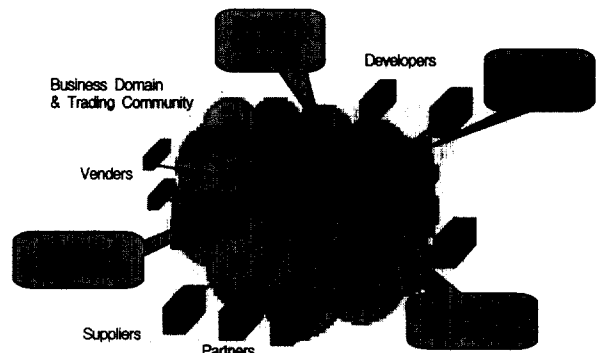
- **Direct** : 첨가적인 소프트웨어나 기능 없이 직접적으로 연결 가능한 작업흐름을 가진 형태로써 애플리케이션의 구성이 쉬우며, 서로간의 메시지를 통해 변경과 이해가 가능하다.
- **Broker** : 단지 브로커를 통해 메시지의 전송이나 수신이 가능한 작업흐름을 가진 것으로써 정해진 규칙이나 명세에 기반한 메시지 전송이 가능하다. 브로커는 작업들간의 정보의 흐름을 지원하며, 비즈니스 프로세스 측면에서 설명되어지는 정보흐름은 작업 행위의 변경에 충분히 적용 가능하다.
- **Process Flow** : Direct와 Broker의 혼합된 형태로써 작업이 다른 조직과 연계가 되고 단일 상태의 독립성을 가질 경우를 지원하기 위한 형태이다. 비즈니스 흐름의 다양한 상황에서 어떤 하나의 작업을 오랜 시간 기다리게 할 수 있는 경우가 생길 수 있다. 이럴 경우 먼저 처리되어야 할 경우와 또 다른 작업흐름을 거쳐야 할 경우 생길 때 적절히 구성될 수 있는 작업흐름의 형태이다.

4.2 프로세스 자동화

컴포넌트의 개발을 위한 프로세스와 컴포넌트를 기반으로 애플리케이션의 개발과 구성을 위한 프로세스는 분리된 형태로 존재하는 것이 아니라 병렬적으로 밀접한 연관성을 가지고 실행된다. 이러한 컴포넌트 방법론에는 컴포넌트 계획, 요구사항 분석, 컴포넌트 설계, 구현, 컴포넌트 시험 및 검증, 애플리케이션으로의 통합에 이르기까지, 전체 프로세스를 정의하고 각 단계별 상세한 지침이 제공되고 있으며, 지원하기 위한 도구도 개발 중에 있다.

이러한, 프로세스와 도구는 하나의 통합환경에서 지원되는 것이 효율적이며, 개발하려는 애플리케이션에 적합한 컴포넌트들을 저장소에서, 검색, 추출, 통합, 커스터마이징 기법들이 제공되어야 한다. 또한, 개발된 컴포넌트와 애플리케이션의 시험과 유지보수에 이르기까지 프로세스와 지원 도구 및 애플리케이션은 상호 운용되어야한다.

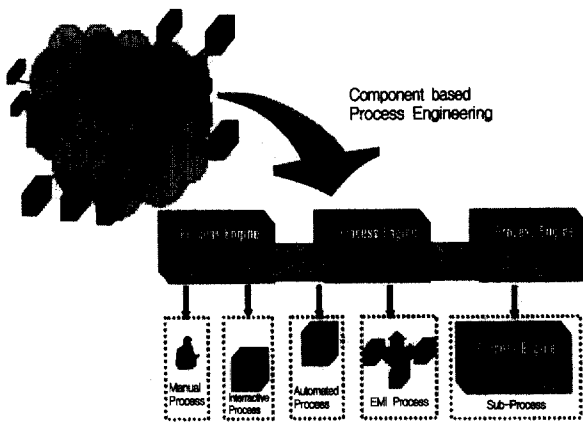
다음(그림 7)은 컴포넌트 개발자, 벤더, 컴포넌트 제공자, 고객 및 파트너 등 다양한 사용자 관점에서 도입하는



(그림 7) 다양한 사용자 관점의 프로세스

개발 프로세스, 애플리케이션 구축 프로세스, 컴포넌트 테스트와 판매를 위한 처리 절차들을 도식화하였다.

이러한 프로세스는 자동화된 제어로 정보의 공유와 관리를 통해 통합된 계층을 구성하고 다른 계층과 쉽게 정보를 전달하도록 한다. 다음 (그림 8)은 기존의 비즈니스 프로세스를 문서화하고, 기존 프로세스에서 새롭게 요구되거나 관련성을 가지지만 누락된 프로세스를 재정의하는 단계를 거쳐 제공되는 자동화된 프로세스의 모델을 제시하고 있다. 프로세스 운영의 가이드 라인의 제시와 상호 운용성을 제공하는 프로세스와 EAI형태의 프로세스 등으로 확장 가능하다.



(그림 8) 프로세스 자동화를 위한 서비스

4.3 컴포넌트 정보통합 및 메시지 전송

컴포넌트 사용자와 행위를 기반으로 식별된 정보들은 통합된 형태로 각각 제공하기 위해서 일련의 명세 형식으로 제공되어야 한다. 또한, 이러한 정보는 상호 운용 시스템에서 생산자, 브로커, 소비자 측면에서 제공되고 획득 될 수 있다.

다음 <표 1>은 개발자와 판매자 및 소비자를 고려하여 통합된 형태로 나타낸 것이다. 개발을 담당하는 생산자가 제공하는 컴포넌트 설계 정보와 수행하는 기능을 기술한 사용사례 모델, 객체모델, 컴포넌트 모델과 인터페이스 기능 정보를 제공한다. 또한, 운용 환경과 관련 도구 및 기술 언어 등의 컴포넌트의 비기능적 정보를 제공하며, 브로커는 생산자가 제공하는 정보와 함께 상업적인 정보를 포함해서 각각 생산자 정보와 컴포넌트 정보를 제공한다. 이들 정보들은 소비자에게 컴포넌트를 획득하기 위한 중요한 선택기준 및 재사용의 근거가 된다.

또한, 생성된 컴포넌트 정보의 효율적인 관리와 통합된 형태로의 접근을 쉽게 하기 위해 XML(eXtended Markup Language)형태로 간략하게 나타내면 (그림 9)와 같다. 이는 개발될 CIS에서 표준화된 정보를 일관된 형태로 상호전달 가능하다. 또한, 공유되는 자원으로 컴포넌트의 정보는 웹

<표 1> 통합 정보

생산자 정보	벤더명	객체 모델 정보	클래스 다이어그램
	주소		클래스 명
	협력 업체		클래스 설명
	전화번호		클래스 속성
	FAX 번호		클래스 오퍼레이션
	메일 주소		타 클래스와의 관계
	홈페이지 URL		컴포넌트 다이어그램
컴포넌트 정보	분류 번호	컴포넌트 모델 정보	컴포넌트 명
	이름		컴포넌트 설명
	품질 정보		구성 클래스
	버전	인터페이스 기능 정보	순차 다이어그램
	제조일		인터페이스 이름
	인증번호		기능 설명
	개발자명		파라미터
구성컴포넌트명	비기능적인 정보	불변성	
사용사례 모델 정보		사용사례 다이어그램	사건 조건
		행위자명	사후 조건
		행위자 설명	플랫폼
		타 행위자와의 관계	미들웨어
		사용사례명	데이터 베이스
		사용사례 설명	개발 툴
	사용사례 순차 다이어그램	기술 언어	
	이용 도구		
	가격		
	라이선스		

기반이나 분산환경에서 표준화된 형태의 통합이 수반되어야 하며, 이러한 정보의 교류는 SOAP을 통한 웹서비스 접근과 XML 형태의 문서로 전달된다.

```

<?xml version = "1.0" ?>
<Component>
  <Supplier_Info>
    <Vender_Name/>
    <Address/>
    <Cowork/>
    <Tel/>
    <Fax/>
    <E_Mail/>
    <Home_URL/>
  </Supplier_Info>
  <Component_Info>
    <Classification_Num/>
    <Name/>
    <Qualify_Info/>
    <Version/>
    <Ensurance_Num/>
    <Quality_Num/>
    <Author_Name/>
    <Component_Unit_Name/>
  </Component_Info>
  <Usecase_Model_Info>
    <Usecase_Diagram/>
    <Actor>
      <Name/>
      <Description/>
      <Flow_Event/>
    <Main Actor = "" System = ""/>
  </Usecase_Model_Info>
</Component>
  
```

```

<Replacement Actor = "" System = ""/>
  <Exception Actor = "" System = ""/>
  </Flow_Event>
</Actor>
</Usecase_Model_Info>
<Object_Model_Info>
  <Class_Diagram/>
  <Class Description = "" Attribute = "" Operation = ""
    Relation = ""/>
</Object_Model_Info>
<Component_Model_Info>
  <Component_Diagram/>
  <Specification>
    <Component_Name/>
    <Description/>
    <Member_Class/>
    <Sequence_Diagram/>
  </Specification>
</Component_Model_Info>
<Interface_Info>
  <Operation_Name Description = "" Return_Type = ""
    Parameter = "" Invariants = "" Pre_Condition = ""
    Post_Condition = ""/>
</Interface_Info>
<Nonfunctional_Info>
  <Platform/>
  <Middleware/>
  <Database/>
  <Develop_Tool/>
  <Language/>
  <Container/>
  <Cost/>
  <License/>
</Nonfunctional_Info>
</Component>

```

(그림 9) XML 형태의 컴포넌트 통합 정보

5. 결론 및 향후 연구

컴포넌트 기반 방법론이 대두되면서, 객체지향에서 컴포넌트 지향의 시스템으로의 전환이 빠르게 진행되고 있다. 특히, 비즈니스 영역에서 파일럿 시스템이 성공을 거두면서 컴포넌트의 중요성과 추진효과의 신뢰성이 증가하고 있다. 현재 방법론의 실무 적용 단계에 이르고 있으나, 아직도 분산환경과 프로세스 및 컴포넌트의 COTS를 이루기 위해서는 기반 환경뿐만 아니라, 프로세스 관리와 이를 지원하는 도구 및 유지 보수등 다양한 서비스의 통합이 요구되고 있다.

본 논문에서는 컴포넌트 기반 방법론을 지원하기 위해 제시된 상호 운용 정보시스템의 아키텍처를 프레젠테이션, 비즈니스 로직, 데이터 제어 계층으로 제시하였고, 각 계층별 상세한 명세와 이들 서비스를 제공하기 위한 수반되는 기능과 역할에 대해서 논의하였다. 또한, 핵심 계층인 비즈니스 로직부분의 각 구성요소의 정의를 통해 컴포넌트 기

반 방법론과 상호 운용시스템과의 방법을 제시하였다. 제시된 시스템은 비즈니스 영역에서 요구하는 컴포넌트의 획득과 이를 위한 개발을 지원가능하며, 컴포넌트의 재사용성과 효율성과 전체 작업 시간을 줄이는 장점을 기대한다.

향후 연구로는 추출된 산출물과 전체 시스템의 모델링을 통해 상호 운용 정보시스템을 구성하기 위한 컴포넌트 식별과 식별된 컴포넌트를 계층별로 요구사항에 적절하게 매칭하여 효율적인 시스템을 구성함으로써, 비즈니스 영역에서의 컴포넌트 재사용성과 효율성을 고려하고자 한다.

참고 문헌

- [1] Clemens Szyperski, Component Software : Beyond Object-Oriented Programming, Addison-Wesley, January 1998.
- [2] Katy Ring, "Enterprise Application Integration," OVUM Report, 2000.
- [3] David S. Linthicum, "Application Servers and EAI," EAI Journal July/August, 2000.
- [4] CBDi Forum, "Component Management and Reuse : The Framework," Interact, 2001.
- [5] Oren Gampel, Alex Gregor, Saniya Ben Hassen, BM Component Broker Connector Overview, IBM Book EZ30RG02, 1998.
- [6] Tim Barrett, Technical Reference Architecture for Component Based Development and Enterprise Application Integration, ComCor I.T Solutions, 2000.
- [7] Toacy C. de Oliveira, Ivan Mathias Filho, "A Framework Based Approach for Workflow Software Development," IASTED2001, 2000.
- [8] Desmond F. D'Souza, Alan c. Wills, Objects, Components, and Frameworks with UML, Addison-Wesley, 1998.
- [9] Jacobson, Griss, Jonsson, Software Reuse, Addison-Wesley, 1999.
- [10] Ho-Jun Shin, Sung-Won Kim, Rhan Jung, "A Study on the Information Integration in CIS(Cooperative Information System) based on Component," Proceedings of the ACIS 2nd International Conference on Software Engineering, Artificial Intelligence, Networking & Parallel/Distributed Computing, pp.185-189, 2001.
- [11] 김행곤 외, "아키텍처 기반의 컴포넌트 저장소 개발을 위한 메타 데이터분류 및 프로토타이핑", KISP 산학연 소프트웨어 공학기술 학술대회논문집, Vol.4, No.2, 2000.
- [12] 김행곤 외, "CBD 기반의 컴포넌트 저장소 프로토타이핑 시스템", 한국정보처리학회 추계학술대회, Vol.7, No.2, 2000.
- [13] 김행곤 외, "분산 컴포넌트 명세에 기반한 비즈니스 컴포넌트 구축에 관한 연구", 한국정보과학회 추계학술대회, Vol.27, No.2, 2000.

김 행 곤

e-mail : hangkon@cuth.cataegu.ac.kr

1985년 중앙대학교 전자계산학과(공학사)

1987년 중앙대학교 대학원 전자계산학과
(공학석사)

1991년 중앙대학교 대학원 전자계산학과
(공학박사)

1978년~1979년 미 항공우주국 객원 연구원

1987년~1989년 한국전기통신공사 전임연구원

1988년~1989년 AT&T 객원 연구원

1990년~현재 대구가톨릭대학교 컴퓨터공학과 부교수

2001년~현재 Central Michigan University 교환교수

관심분야 : CBSE, 소프트웨어 제공학, CASE, 유지보수 자동화
툴, 사용자 인터페이스, 요구공학 및 도메인 공학

신 호 준

e-mail : g98521002@cuth.cataegu.ac.kr

1998년 경일대학교 전자계산학과(공학사)

2000년 대구효성가톨릭대학교 전산통계학과
(이학석사)

2000년~현재 대구가톨릭대학교 전산통계
학과 박사과정

관심분야 : CBSE, 소프트웨어 제공학, 도메인 공학, 분석·설계
모델링