컴포넌트 지원을 위한 상호운용 정보시스템기반의 정보 통합

신 호 준° 이 수 기 김 행 곤
대구가톨릭대학교 컴퓨터공학과
g98521002@cuth.cataegu.ac.kr, sklee@bear.kyungil.ac.kr, hangkon@cuth.cataegu.ac.kr

Information Integration based on Cooperative Information System for Component

Ho-Jun Shin° Su-Ki Lee Haeng-Kon Kim
Software Engineering Lab, Dept. of Computer Science, Catholic University of Daegu

요 약

컴포넌트의 기반 개발은 저비용과 단기간에 개발되는 잠재적인 장점을 가지고 있으며, 대부분의 컴포넌트기반의 연구들은 설계와 통합 프로세스에 초점을 두고 있다. 또한, 방법론과 도구들은 컴포넌트 구축과 개발 프로세스를 지원하지만, 비동기적으로 상호작용하는 사용자를 위한 통합된 정보의 재공과 관리에 대한 연구는 미약하다.

따라서, 본 논문에서는 컴포넌트 생산자, 서비스업자, 소비자를 위해 요구되고 지원되어야할 컴포넌트 정보를 고려한 상호운용시스템에서의 정보 통합을 제안한다. in-house 형태의 컴포넌트 구축과 컴포넌트로 애플리케이션을 조립하는 개발자를 지원 가능하고, 컴포넌트 시장에서 구매하기 위한 가이드라인을 제시한다. 이러한 정보는 상호운용 정보 시스템 상에서 식별된 컴포넌트 사용자를 중심으로 통합되며, 결합과 추가적으로 요구되는 노력들을 감소시키는 고수준의 컴포넌트를 획득 가능하게 한다.

1. 서론

비즈니스 도메인을 중심으로 다양한 시스템에 지난 몇년동안 재사용에 관한 연구가 활발했으며, 현재 세계적인 소프트웨어 시장을 구성하기 위해 가장 적합한 기술로 컴포넌트를 인정하고 있다. 이는 독립적으로 개발되어진 컴포넌트를 레고 퍼즐의 조각처럼 자격을 제대로 갖춘 애플리케이션 내에 자연스럽게 통합되는 것을 지원한다. 또한, 애플리케이션의 기능 추가를 위한 부분이나 새로운 애플리케이션을 구성하기 위해서 조립할 수 있는 컴포넌트를 획득하기 위해서는 컴포넌트의 정보에 대한 이해는 필수적이다. 비즈니스 영역에서 컴포넌트에 대한 요구는 다양한 사용자들에게 발생하며, 각각 다른 형태로 컴포넌트와 관련된 정보를 필요로 한다. 즉, 컴포넌트 리파지토리를 중심으로 컴포넌트 개발자, 조립자, 일반 사용자, 컴포넌트 재공자 등이 각각의 목적에 따라 정보를 공유하며, 비동기적으로 상호운용된다. 이러한, 정보는 컴포넌트 생명주기상의 부산물과 유통을 위한 카탈로그, 관리를 위한 정보, 획득을 위한 정보 등으로 다양하다[1].

본 논문에서는 다양한 사용자들의 비동기적인 상호운용성을 재공할 수 있는 아키텍처를 제시하고, 이 아키텍처 상에서 공유되어야될 컴포넌트 정보를 통합된 형태로 재공하고자 한다. 이를 위해 컴포넌트의 정보를 요구하는 사용자들을 분석하며 각각의 사용자가 요구하는 정보를 파악하여 적절한 형태의 정보로 통합을 유도한다. 또한, 컴포넌트에 수반되는 정보를 분석하여 통합된 정보관리와 상호운용성을 지원할 수 있는 아키텍처에서 소비자와 재공자에게 요구되고 재공되는 다양한 컴포넌트의 정보를 고려한다. 이는 상호운용 정보 시스템 상에서 컴포넌트 정보를 공유하고 이를 통한 각각의 사용자를 충족시킬 수 있는 통합된 정보를 제공하고자 한다.

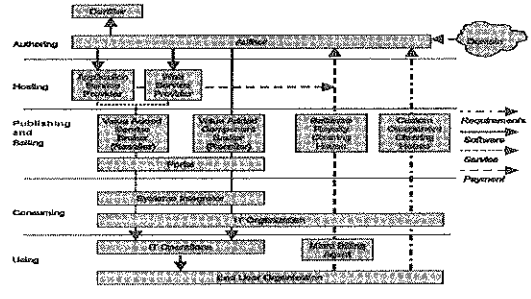
2. 관련 연구

2.1 Cooperative Computing

컴퓨터와 통신망을 이용하여 공동작업 및 상호운용성을 지원하는 기술로 일반적으로 상호운용 컴퓨팅, 그룹웨어 CSCW(Computer Supported Collaboration Work)라고 불린다. 이를 이용한 작업은 크게 두가지 측면으로 그 중요성을 파악 할 수 있는데, 첫째는 통신망 기반 구조 및 제공 능력에서 빠른 성장에 기반을 둔 기술이다. 단순한 기술이 아니라 다양한 사용자들 간의 커뮤니티 형성에 영향을 줄뿐만 아니라, 작업하는 방식과 조직의 구조에까지 영향을 미치는 종합적인 기술을 의미한다. 둘째는 비즈니스 전략으로써 새로운 기술의 도입을 통해 기존의 계층적 조직 형태에서 탈피한 새롭고 효율적 구조를 위한 엔지니어링이 불가피하게 되었다. 이는 직면하는 문제들이 다양한 선택을 제공하는 기반 환경을 제공하며, 그에 따르는 서비스와 아주 밀접한 연관성을 지니고 있다. 상호운용성을 제공하기 위한 기술 및 서비스는 전자 메일과 메세징, 문서관리 그리고, 공동 저작도으로 나눌 수 있다[2].

2.2 컴포넌트 공급체인

컴포넌트의 기술이 발전하면서 개발된 컴포넌트를 구매하고 적용할 수 있는 컴포넌트 시장이 형성되고 있다. 다음(그림 1)과 같이 관련된 몇몇 계층이 존재한다. 모든 컴포넌트가 명확하지는 않지만, 웹서비스는 각각의 관련된 계층을 통해서 전달될 것이다. 어떤 개발자는 판매자나 브로커를 통해 배포되는 컴포넌트의 보증을 원할 것이다. 그러나, 고객의 입무는 몇몇 관련된 계층을 통해서 업무를 진행한다[3].



(그림 1) 소프트웨어 컴포넌트 공급체인

- 컴포넌트 작성 : 컴포넌트를 설계하고 구축하며, 소프트웨어 컴포넌트가 적용되는 도메인에 적절한지 품질을 검증한다. 또한, 소프트웨어 컴포넌트 시장의 요구사항 변경을 기능과 가격 등을 통해 찾아내고 명세를 작성하며, 사용자의 요구사항을 검증한다.
- 호스팅 : 개발된 컴포넌트를 관리하며, 각기 다른 고객의 요구를 위한 웹 서비스를 재공하며, 물리적인 구현을 설립한다.
- 배포 및 판매 : 컴포넌트상의 정보를 웹서비스로 배포하며, 판매를 위한 컴포넌트의 가공을 지원한다. 또한, 컴포넌트와 개발자의 정보를 재공하며, 카탈로그를 생성하여 기능적, 비기능적인 정보를 통한 컴포넌트 선택을 도와준다.

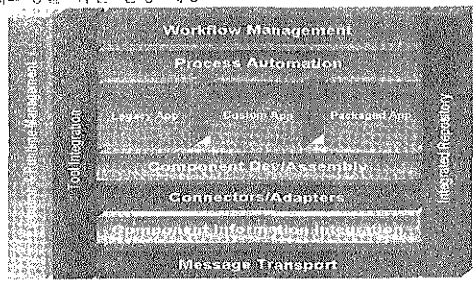
● 구매 및 사용 : 웹을 통한 검색과 컴포넌트를 구매하며, 구매한 컴포넌트를 사용자 애플리케이션에 통합을 유도한다.
● 사용 : 컴포넌트를 설치하며 애플리케이션이나 컨테이너에서 컴포넌트를 실행시킨다.

3. 상호운용 정보시스템

컴포넌트를 기반으로하는 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 통합되어야한다. 또한, 컴포넌트를 지향하는 방법론을 위한 툴들과 리파지토리, 상용 서버스들이 하나의 선상에서 자동화된 형태로 서비스가 제공되어야 하며, 이를 지원하는 통합환경과 상호운영 정보 시스템이 구성되어야 한다.

따라서, 컴포넌트를 중심으로 공유되어야할 정보에 대한 분석을 통해 정보의 통합이 이루어져야하며 다양한 사용자를 지향하는 프로세스의 통합과 컴포넌트 지향 프로세스를 지원할 수 있는 도구들의 통합이 이루어져야한다. 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 관리자와 실행환경 등을 고려하여 다음 (그림 2)와 같이 구성할 수 있다.

● 작업흐름 관리 : 비즈니스 프로세스 내에 컴포넌트의 개발, 획득, 적용을 위한 작업흐름 개발과 실행의 관리
● 프로세스 자동화 : 컴포넌트 개발과 컴포넌트 기반의 애플리케이션을 구성하기위한 시스템간과의 프로세스의 흐름과 자료의 처리
● 기존, 고객, 패케이지화된 애플리케이션 : 개발되거나 적용될 기존의 애플리케이션과 사용자 애플리케이션이 위치하며, 컴포넌트의 패키지화된 애플리케이션이 위치
● 컴포넌트 개발/조립 : 컴포넌트 개발과 조립을 하기 위한 계층으로 툴과 실행 환경과 연관성을 가짐
● 연결/어댑터 : 개발된 컴포넌트의 조립을 위한 컨넥터와 기존의 시스템과의 연결을 위한 어댑터 정의
● 컴포넌트 정보 통합 : 컴포넌트의 배포와 재사용을 위한 관련된 정보를 통합된 형태로 제공
● 메시지 전송 : 컴포넌트의 유도와 사용을 위한 메시지전송과 관리
● 통합된 리파지토리 : 컴포넌트의 설계, 구현물과 배포를 위한 모든 정보와 자원의 저장 및 관리
● 툴 통합 : 컴포넌트의 전체 생명주기를 지원하기 위한 툴의 지원
● 실행 그리고 실시간 환경 : 구현된 컴포넌트나 애플리케이션의 실행과 테스팅을 위한 환경 제공

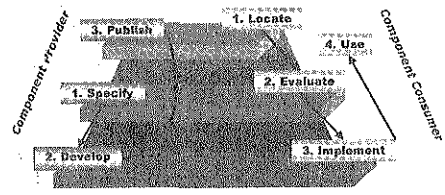


(그림 2) 상호운용정보시스템 아키텍처

4. 컴포넌트 정보 통합

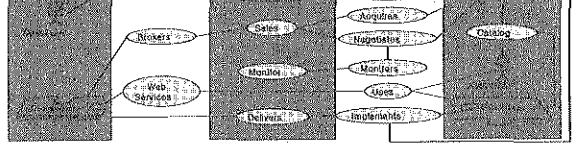
4.1 사용자 식별

통합된 아키텍처 상에서 발생되는 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 이끌어지는 다른 타입의 컴포넌트 정보는 다른 레벨에서 요구되어진다. 다음(그림 3)은 컴포넌트의 개발과 이를 기반으로 하는 애플리케이션의 구현을 위한 개발 프로세스에서 수행되는 행위를 레벨형태로 나타낸 것이다. 생명주기에서 크게 컴포넌트를 제공하는 측면과 이를 사용하는 측면 즉, 생산자와 소비자로써 식별 할 수 있다.



(그림 3) 컴포넌트 생명주기의 레벨

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 4) 상호운용되는 사용자와 행위

● 컴포넌트 제공자 : 컴포넌트 구현을 위한 설계와 구축
● 컴포넌트 소메업자 : 개발된 컴포넌트의 등록과 배포
● 컴포넌트 라이브러리언 : 컴포넌트 유통업자와 컴포넌트 개발자들의 위치 파악
● 컴포넌트 소비자 : 전체 조직이나 개인적인 프로젝트를 위한 컴포넌트의 식별과 획득하며, 컴포넌트 구현에 의해 제공되는 서비스사용

4.2 컴포넌트 정보 통합

식별된 컴포넌트의 생산자, 소비자와 서비스를 위한 브로커는 각각의 행위를 가지게 되며, 수반되는 정보를 요구하고 제공하게 된다. 이러한 정보는 생산자 입장에서는 설계, 구현 정보로써 배포할 경우에 작성된다. 또한, 소비자는 생산자가 제공하는 정보 외에도 상업적으로 컴포넌트를 유통하기위해 브로커에 의해 작성된 정보 또한 제공받는다. 그리고, 각각의 사용자의 행위적인 측면에 따라서 요구되는 정보는 다양하게 제시될 수 있다. 또한, 컴포넌트를 관리하기위한 정보와 메시지는 다양한 형태로 존재하게되며, 상호 운용시스템상에서 통합되어야만 한다. 다음 <표 1>은 행위에 따라 요구되는 컴포넌트의 정보와 영역을 제시한 것으로 컴포넌트의 개발과 배포뿐만 아니라 전체 행위를 고려한 것이다.

<표 1> 컴포넌트 정보의 요구사항

활 동	전형적인 질문 요구	정보 영역
식 별	유효성/표준 지원/업무 내에 플러그/품질 레벨	업무영역/표준증적/분류기술
설 계	제공하는 서비스/실행하는 방법/의존성	UML 분석, 설계
구 축	제공하기위해 요구되는 서비스/기술 프레임워크	UML 분석, 설계/기술
평 가	하는 일/동작 방법/의존성/	UML 분석, 설계/
	업무와 기술 환경에서의 작업/가능성/지원 방법	기술/표준 증적/보증
관계획득	가격/지불방법/제품 방법/서비스 레벨의 특의	상거래
구 현	컴포넌트 사용 방법/사용된 기술/의존성	객체와 컴포넌트/기술
구 축	제공되는 서비스/호출 방법/사용 모니터 방법	객체와 컴포넌트/기술
실 행	구현 환경	특정 플랫폼

행위적인 측면과 사용자의 측면을 고려할 경우 컴포넌트 기반 개발의 시각을 가진 대부분의 사람은 다른 자원에서 다양한 컴포넌트들을 선택 할 것이다. 그 기대는 이미 구현된 in-house 컴포넌트가 특정 요구에 부합되지 않는다면 적당한 후보의 위치를 알아내기 위해 웹을 통해 빠르게 찾는다. <표 2>은 컴포넌트의 목록상의 정보로서 각각 다른 방법으로 탐색하게 되며 사용되는 각각의 카탈로그는 다음과 같이 정의 할 수 있다.

<표 2> 컴포넌트 카탈로그 정보

타입	정보	주요활동	영역	공급자	소비자
산업 카탈로그	고수준:명세/소스 /인증/표준 승낙	획득을위한 위치파악	이해/다중벤더 /다중업자/ 컴포넌트 모든 타입	유통업자 라이브러리언	준비자
유통 카탈로그	고수준:명세/상업 /표준 승낙	판매/획득	단일벤더	유통업자	준비자
기업 카탈로그	고수준:명세 /표준 승낙	사용을 위한 위치파악	조직	제공자	어셈블러
명세	상세:명세/모델	평가/사용 /설계/구축	컴포넌트	개발자	준비자 어셈블러
인터페이스	상세:명세/모델 /구현	평가/사용 /개발/구축	인터페이스	개발자	어셈블러 운영시스템

산업 카탈로그는 다수의 유통업자와 개발자들에 위치하는 컴포넌트가 가능하다. 주요한 활동은 소비자들을 올바른 방향으로 지시하며 전체 컴포넌트 시장의 가시성을 보증함으로 획득가능한 잠재적인 컴포넌트를 발견할 수 있도록 도와준다. 제공되는 정보는 키워드 검색이 가능하다. 또한, 유통 카탈로그는 산업 디렉토리와 유사하며, 특별한 유통업자를 통해 가능한 컴포넌트의 목록을 제공한다. 그러나, 이는 발생되는 동적인 처리가 가능한 더 상업적인 정보를 제공한다. 그리고 기업 카탈로그는 조직내에서 동적으로 개발된 컴포넌트, 획득과 개발된 in-house 툴 다와 애플리케이션에 조합을 용이하게하는 컴포넌트의 목록을 만든다. 마지막으로 명세와 인터페이스는 컴포넌트의 상세한 명세를 제공하며, 평가 가능한 유통업자들의 카탈로그와 재사용 가능한 업체 카탈로그의 확장으로써 제공되어진다.

산업 디렉토리와 유통 카탈로그는 웹페이지 형태로 제공되며, 목록의 수가 다루기 힘들 때 데이터베이스 형태로 처리되며, 기업 카탈로그는 툴이나 리파지토리 기반으로 어디에서나 제공된다. 이는 툴과 리파지토리가 기반되기 때문에 기업 카탈로그는 명세의 요구사항을 더 지원하게 되며, 그래픽한 모델의 링크나 표현이 가능하다. 정보의 단계는 온라인 상의 유통 카탈로그가 드물며, 많지 않은 비즈니스 컴포넌트가 이러한 메커니즘을 통해서 현재 판매되는 부분을 반영한다.

툴과 리파지토리 업체간의 협상과 대화를 통한 통합이 어뤄져야 하며, 유통 카탈로그는 유통과 기업 카탈로그간의 수직적인 통합이 증가될 IDE(Integrated Development Enviroment) 개발자와 통합될 컴포넌트 관리 툴과 리파지토리 형태로 서비스해야한다. 공급체인을 통한 저장, 교환, 컴포넌트 정보의 표현에서의 요구사항 또한, 하나의 작업흐름 형태로 통합해야한다. 상업적인 컴포넌트의 상세한 명세는 이미 구현된 in-house로써 단지 기대되는 구매자에게 보여질 것이다. 소스에 관심이 없는 모든 컴포넌트는 동일한 사용자 인터페이스를 통해 동일한 레벨에서 동일한 방법으로 보여져야 한다.

4.3 컴포넌트 정보 명세

컴포넌트 사용자와 행위를 기반으로 식별된 정보들은 통합된 형태로 각각 제공하기 위해서 일련의 명세 형식으로 제공되어야 한다. 또한, 이러한 정보는 상호운용시스템에서 생산자, 브로커, 소비자 측면에서 제공되고 획득 될 수 있다. 생산자가 제공하는 컴포넌트 설계와 개발 정보 및 기능적, 비기능적 정보를 제공해야하며, 브로커는 생산자가 제공하는 정보와 함께 상업적인 정보를 포함해서 제공해야한다. 이 정보들은 소비자에게 컴포넌트를 획득하기위한 중요한 선택기준 및 재사용의 근거가 된다. 다음 <표 3>는 이러한 정보를 개발자와 판매자 소비자를 고려하여 통합된 형태로 나타낸것이다.

<표 3> 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 모델 정보	컴포넌트 다이어그램
컴포넌트 정보	툴팁 번호		컴포넌트 명
	이름		컴포넌트 설명
	일반 정보		구성 클래스
	버전		순차 다이아그램
	제조일	인터페이스 기능 정보	인터페이스 이름
	인증 번호		기능 설명
	개발자 명		파라미터
	구성컴포넌트 명		불변성
사용사례 모델 정보	사용사례 다이어그램		사전 조건
	행위자 명		사후 조건
	행위자 설명	비기능적인 정보	플랫폼
	타 행위자와의 관계		미들웨어
	사용사례 명		데이터 베이스
	사용사례 설명		개발 툴
	사용사례 순차 다이어그램		기술 언어
			이행 도구
			가격
			라이센스

또한, 생성된 컴포넌트 정보의 효율적인 관리와 통합된 형태로의 접근을 쉽게 하기 위해 XML형태로 나타내면 (그림 5)과 같다.

<?xml version="1.0" ?>
<Component>
<Supplier_Info>
<Vender_Name/>
<Address/>
<Country/>
<Tel/>
<Fax/>
<E_Mail/>
<Home_URL/>
</Supplier_Info>
<Component_Info>
<Classification_Num/>
<Name/>
<Quality_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=""/>
<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="" Returen_Type="" Parameter="" Invariants="" Pre_Condition="" Post_Condition=""/>
</Interface_Info>
<Nonfunctional_Info>
<Platform/>
<Middleware/>
<Database/>
<Develop_Tool/>
<Language/>
<Complexity/>
<Cost/>
<License/>
</Nonfunctional_Info>
</Component>

(그림 5) XML 형태의 컴포넌트 통합 정보

5. 결론 및 향후 연구

소프트웨어 컴포넌트 시장은 웹을 기반으로 시장을 형성하고 있으며, 위치한 컴포넌트에 의해 제공되는 서비스는 구매와 사용, 조정, 실시간에 사용되기위한 서비스를 고려해야 한다. 이러한 요구사항을 달성하기 위해서 정보의 자유로운 교환을 위한 산업계의 표준을 설립하고, 공급 체인을 제공하는 다른 B2B에서 일어나는 동일한 상호운용성을 제공해야 한다. 이는 표준을 이끄는 밴더, 가능한 컴포넌트를 선택하기위한 자유로운 정보의 제공에 있다. 따라서, 컴포넌트의 개발에서 적용에 이르는 과정에서 수반되는 정보의 관리와 전체 공급체인을 형성하는 정보를 제공함으로써, 다양한 사용자의 요구사항을 반영하고 컴포넌트의 등록, 검색 및 적용에 이르는 시간을 줄일 수 있다. 또한, 통합된 환경에서 컴포넌트 관련 서비스를 제공할 수 있으며, 분산된 작업자들간의 상호운용성을 제공할 수 있다. 또한, 컴포넌트의 개발에서 재사용까지에 걸리는 시간과 소프트웨어의 개발 기간을 크게 단축하고, 결과적으로 개발비용을 감소한다. 그리고 통합된 환경을 제공함으로써 실행성의 보장을 가져올 수 있으며, 컴포넌트 개발 기술력을 향상, 컴포넌트 통합환경에 의한 소프트웨어 재사용 마인드 확산을 기대한다.

향후 연구로는 제안한 통합환경을 기반으로 서비상의 애플리케이션 통합과 리파지토리 기반의 컴포넌트 공급 체인을 지원하는 작업흐름 모델 생성, 상호운용 도구의 개발 등 많은 연구 과제를 남기고 있다.

[참고 문헌]

[1] 김행곤 외3인, "컴포넌트 메타데이터의 정의를 통한 효율적인 검색 방법에 관한 연구", 2000가을학술발표논문집 제 27권 2호, 2000.
[2] Mark Levitt, "Collaborative Computing", IDC, 1997.
[3] Lawrence Wilkes, "Business Integration", CBDi Forum Report Produced for IBM, 2000.
[4] David Hollingsworth, "The Workflow Reference Model", Workflow Management Coalition Document, 1995.
[5] Jacobson, Griss, Jonsson, Software Reuse, Addison-Wesley, 1999.
[6] M. Shaw and D. Garlan, Software Architecture : Perspectives on an Emerging Discipline, Prentice Hall, 1996.
[7] Desmond F. D'Souza, Alan c. Wills, Objects, Components, and Frameworks with UML, Addison-Wesley, 1998.
[8] Oren Gampel, Alex Gregor, Saniya Ben Hassen, IBM Component Broker Connector Overview, IBM Book EZ30RG02, 1998.
[9] Tim Barrett, Technical Reference Architecture for Component Based Development and Enterprise Application Integration, ComCor I.T Solutions, 2000.
[10] Tim Barrett, Enterprise Integration Middleware Positioning, ComCor I.T Solutions, 2000.
[11] Butler Group, Forté Intelligent Business Backbone, 17th Amsterdam CBDi Forum, 1999.
[12] Christian Zeidler and Bernhard Malle, Integration Techniques Approaches, Proceeding of Tools USA 97, 1997.
[13] David S. Linthicum, Process Automation and EAI, EAI Journal, 2000.
[14] Toacy C. de Oliveira, Ivan Mathias Filho, A Framework Based Approach for Workfolw Software Development, IASTED2001, 2000.