

CBD 아키텍처 기반 e-비즈니스 에이전트 프로토타이핑 시스템

신 호 준[†] · 김 행 곤^{††}

요 약

차세대 웹 애플리케이션은 거대하고 복잡하고 유연성을 요구한다. 에이전트 지향 시스템은 이러한 애플리케이션에 큰 가능성을 가지고 있으며, 에이전트는 동적으로 발견할 수 있고, e-서비스를 합성할 수 있고 상호작용을 증대할 수 있다. CBD(Component Based Development)를 통한 소프트웨어 에이전트의 개발은 프로젝트의 개발 속도를 증가시키며, 좀 더 나은 품질을 제공하며 낮은 개발비용으로 성공을 증명할 수 있다. 본 논문에서는 컴포넌트와 UML(Unified Modeling Language)을 사용한 소프트웨어 에이전트를 위한 체계적인 개발 프로세스를 제안한다. 일반 에이전트와 e-비즈니스 에이전트의 분류를 통해 식별된 관련 컴포넌트 계층에 대한 ebA-CBD 참조 아키텍처를 제시한다. 또한, UML로 사용한 의미적인 프레임워크에서 에이전트 지향 개념을 기반으로하여 기존의 에이전트 지향 소프트웨어 공학 방법론의 특징을 고려한 가이드라인으로 ebA-CBD 프로세스를 제안한다. 첫 번째로 에이전트 컴포넌트 명세를 개발하고, 목표, 역할, 상호작용과 아키텍처 모델을 작성한다. 제안된 프로세스에 따라 사례 연구로서 상품정보의 메일링 서비스인 e-CPIMAS(e-Commerce Product Information Mailing Agent System)를 개발한다. 끝으로 비즈니스 애플리케이션과 e-비즈니스 에이전트를 개발하기 위해 그 효율성, 재사용성, 생산성과 품질성 증가를 기대한다.

The e-Business Agent Prototyping System with Component Based Development Architecture

Ho-Jun Shin[†] · Haeng-Kon Kim^{††}

ABSTRACT

The next generation of web applications will need to be larger, more complex, and flexible. Agent-oriented systems have great potential for these e-commerce applications. Agents can dynamically discover and compose e-services and mediate interactions. Development of software agents with CBD (Component Based Development) has proved to be successful in increasing speed to market of development projects, lowering the development cost and providing better quality. In this thesis, we propose a systemic development process for software agents using component and UML (Unified Modeling Language). We suggest a ebA (e-business Agent) CBD reference architecture for layer the related components through identification and classification of general agent and e-business agent. We also propose the ebA-CBD process that is a guideline to consider the best features of existing agent oriented software engineering methodologies, while grounding agent-oriented concepts in the same underlying semantic framework used by UML. We first developed the agent components specification and modeled it with Goal, Role, Interaction, and Architecture Model. Based on this, we developed e-CPIMAS (e-Commerce Product Information Mailing Agent System) as a case study that provides the product information's mailing service according to proposed process formality. We finally describe how these concepts may assist in increasing the efficiency, reusability, productivity and quality to develop the business application and e-business agent.

키워드 : e-비즈니스 에이전트, CBD(Component Based Development), ebA-CBD(e-business Agent-CBD), ebA-CBD 아키텍처

1. 서 론

컴포넌트 기술은 plug-and-play 형태로 상업적인 인프라 구조 상에서 비즈니스 솔루션 개발에 적합한 기술로 인정하고 있다. IT 업계에서도 최우선 과제로 생각하고 있는 것은 비즈니스 개념을 적절히 표현하고 빠른 개발을 요구하

고 있으며, 이러한 e-비즈니스 영역에 에이전트의 활용은 간과 할 수 없는 부분이다. 따라서, 빠른 개발시간과 변경에 대한 높은 대체성과 유연성을 가진 컴포넌트 기술을 활용한 에이전트 서비스 제공은 매우 중요하다. 또한, 이러한 시스템 구축을 위한 기반 연구로써 표준화된 아키텍처 상에서 에이전트 모델의 정의가 고려되어야만 한다[1-3].

본 논문에서는 CBD 아키텍처를 연구하고 이를 통해 e-비즈니스영역에 적용가능한 에이전트를 컴포넌트로 구성할 수 있는 프로세스를 제안한다. e-비즈니스 에이전트를 식별하기 위해 기존의 일반적인 에이전트와 비즈니스 영역에서

※ 본 연구는 한국과학재단 지역대학우수과학자육성지원연구(R05-2002-000-011980) 지원으로 수행되었음.

† 준 회 원 : 대구가톨릭대학교 대학원 전산통계학과

†† 중 심 회 원 : 대구가톨릭대학교 컴퓨터공학과 교수
논문접수 : 2003년 7월 1일, 심사완료 : 2003년 8월 13일

사용되고 있는 에이전트를 분류하여 이를 e-비즈니스 에이전트 아키텍처로 구성한다. 구성된 아키텍처를 기반으로 e-비즈니스 컴포넌트 명세 기법에 대해 연구한다. 특히 e-비즈니스 에이전트의 내부구조를 모델링으로 표현함으로써 가시적으로 에이전트를 표현한다. 본 논문에서는 e-비즈니스 에이전트를 ebA(e-business Agent)로 명명하며, ebA 용어는 프로세스 내에서와 단계별 산출물의 명칭에 사용한다.

ebA-CBD 프로세스에는 e-비즈니스 시스템 및 에이전트 시스템의 요구사항 식별, e-비즈니스 에이전트의 모델링, 식별된 에이전트 기능을 통한 컴포넌트 개발 등의 세부적인 단계와 활동 등을 기술한다. 정의된 프로세스에 따라 전자상거래 영역에서 주로 사용되는 메일링 서비스를 에이전트 컴포넌트로 구성하는 것을 사례로 제시한다.

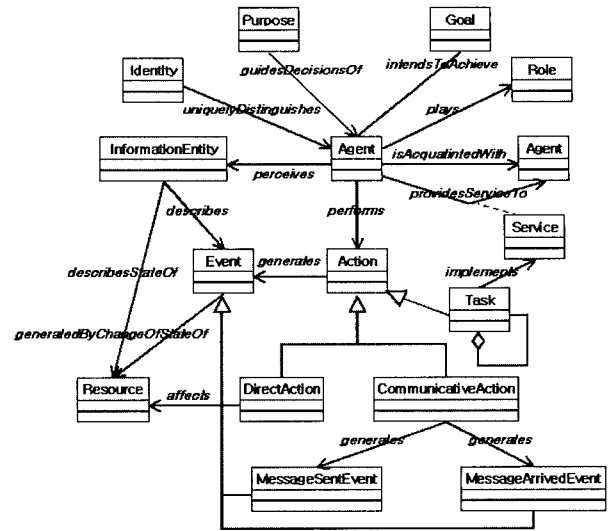
논문의 구성은 2장에서 에이전트의 정의와 특징에 대해 살펴보고, ABCD(Architecture Base Common Domain) 컴포넌트 아키텍처에 대해 고찰한다. 3장에서는 기존의 일반적인 에이전트와 e-비즈니스 영역에 적용 가능한 에이전트의 분류를 통해서, 컴포넌트 참조 아키텍처를 제시한다. 4장에서는 ebA-CBD에서 참조 아키텍처를 기반으로 에이전트를 효율적으로 개발하기 위한 컴포넌트 기반 개발 프로세스를 정의한다. 5장에서는 제안된 ebA-CBD 프로세스에 따라서 전자상거래에서 메일을 통한 상품 정보 알림 서비스를 사례 연구하며, 기존의 개발 방법론들을 비교, 평가한다. 끝으로, 6장에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

2.1 에이전트

에이전트는 일반적으로 사람을 대신해서 사람이 원하는 작업을 자동적으로 해결하여 주는 프로그램으로 해석될 수 있다. 이러한 에이전트는 자율성, 지능성, 이동성, 사회성 등의 특성을 추출해 낼 수 있다. 일반적인 에이전트 뿐만 아니라 e-비즈니스 영역의 에이전트는 다음과 같은 몇몇 특성의 조합을 가지고 있다[4].

- 자율성(Autonomous) : 목적에 따라 활동성을 시험적으로 초기화한다. 제어의 연결을 가지며 사용자의 행동에 따라 활동하며, 에이전트가 보내는 메시지에 독립적이다.
- 수용성(Adaptable) : 사용자의 커스터마이징나 새로운 기능의 다운로드를 통한 자신의 학습을 통해 에이전트의 행위를 변경한다.
- 지능성(Knowledgeable) : 지식베이스와 추론 능력을 가진 에이전트가 지식과 정보를 통해 사용자의 의도나 목적 등을 파악하고 이를 지원하기 위한 계획을 세우고 행동을 취할 수 있다.
- 이동성(Mobile) : 코드나 상태에 의해서 실행중인 작업에서 다른 작업으로 이동가능하며, 새로운 작업과 상태가



(그림 1) 에이전트 개념 모델

변경된 남은 작업에 대해서 계속적으로 실행 가능하다.

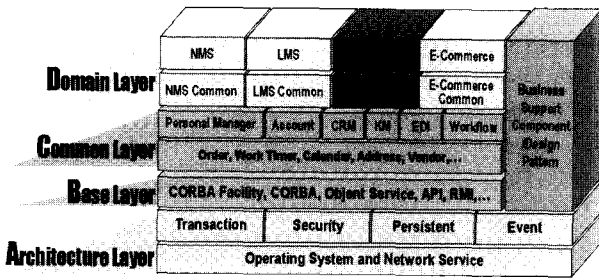
- 협력성(Collaborative) : 통신할 수 있고 하나의 작업을 공동 수행, 동적 수행 혹은 정적 다중 에이전트를 형성하기 위한 다른 에이전트와 협력하여 처리함으로써 문제 해결 능력을 향상시킨다.
- 지속성(Persistent) : 기반구조는 유효한 실행시간 동안 작업이 실패했을 경우 초과한 시간 동안에 상태를 자동적으로 유지해야한다.

에이전트는 환경의 변화에 대응하며, 다른 에이전트간의 상호 작용하는 소프트웨어 컴포넌트의 성격을 가지고 있다. 또한, 원자적인 자동성을 가진 엔티티로써 (그림 1)과 같은 개념적인 관련성을 가진다. 기능적인 능력은 에이전트의 서비스로써 나타나며, 이는 객체의 행위와 유사하다[5].

에이전트의 행위는 단지 내부적인 이벤트나 상호작용에 의해서뿐만 아니라 자신의 동기에 의해서도 지시된다. 동기는 목적으로 기술되는 속성으로 언급되며, 목적은 서비스 제공의 요청에 에이전트가 동의를 할 것인지에 영향과 서비스 제공 방법이 된다. 에이전트는 정보 엔티티를 인지하며, 이는 이벤트의 명세와 에이전트 환경에서의 엔티티 상태에 대한 정보로 분류된다. 자원은 에이전트에 의해 사용되는 데이터베이스나 내부적인 프로그램과 같이 자율적이지 않은 엔티티로 표현된다.

2.2 ABCD 컴포넌트 아키텍처

컴포넌트 기반의 비즈니스 솔루션을 위한 표준 모델로서 ABCD 컴포넌트 아키텍처는 멀티 벤더, 멀티 솔루션의 통합을 위해 컴포넌트의 영역과 추상성, 입자성을 기준으로 계층적 분류가 되어 있다. ABCD 컴포넌트 아키텍처 정의를 위한 원칙은 비즈니스를 위한 컴포넌트와 일반적인 시스템 요구 컴포넌트로 구분하고, 기반이 되는 필수 컴포넌트와 부가



(그림 2) ABCD 컴포넌트 아키텍처

적 선택적인 컴포넌트로 구분한다. (그림 2)는 ABCD 아키텍처로써 전체 4개의 계층으로 구성되며, 전체 계층은 수직적 관점으로 나누어지며, 각 계층은 수평적인 관련성을 가지고 있다[6, 7].

컴포넌트 아키텍처의 아키텍처 계층은 분산 컴퓨팅 환경에서 멀티 벤더/응용 구축을 위한 하부적인 물리적 플랫폼이며, 기초 계층은 분산 컴퓨팅을 위한 미들웨어적인 통합 API 및 기존의 분산 객체 서비스를 지원하며, 응용에 하부적인 기능 수행에 필요한 공통 컴포넌트들로 구성된다. 또한, 벤더 독립적, 장비 지향적 서비스 및 기존의 분산 환경 멀티 응용 컴포넌트를 이용한다. 공통 계층은 비즈니스 응용을 위한 실행성의 기능성 컴포넌트들로써 도메인 지향적인 컴포넌트들은 비즈니스의 기초 해결군이 된다. 또한, 확장적인 컴포넌트들 집합 및 작업 프로세스를 지원하며, 설계 패턴등을 포함한 확장, 변경을 통한 응용 전개가 가능하다. 도메인에 공통적이고 독립적인 작은 규모로 비즈니스 영역에서 사용가능하며, 영역의 논리수행에 핵심적 프로세스를 제공한다. 도메인 계층은 각 비즈니스 영역별 필요 컴포넌트들로 구성되며, 직접 활용되는 실행 모듈 형태로써 도메인 응용의 기본적인 빌딩 블록이 된다.

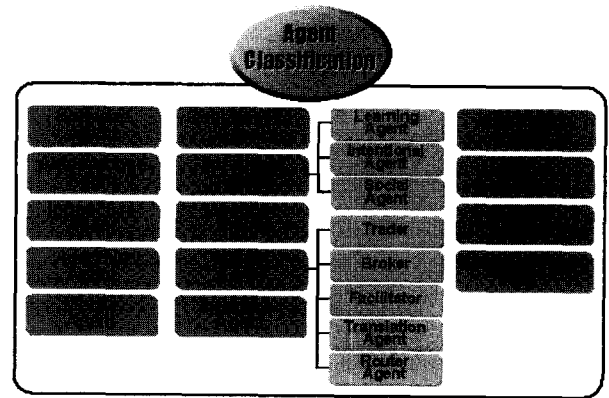
3. ebA-CBD 참조 아키텍처

컴포넌트 참조 아키텍처는 기존의 에이전트 기반 시스템에서 주요한 속성을 기준으로 분석하여 일반적인 에이전트를 분류한다. 또한, e-비즈니스 시스템에서 사용되거나 요구되는 에이전트를 영역별 속성 기준으로 분류함으로써 일반 에이전트와의 공통 영역을 식별한다. 식별된 공통 영역은 e-비즈니스 에이전트의 개발을 지원하기 위한 컴포넌트의 체계적인 참조 아키텍처로 구성된다. 이 아키텍처는 e-비즈니스 에이전트 시스템 개발에 적용가능한 컴포넌트의 분석, 설계를 위한 가이드라인으로 제시된다.

3.1 분류

3.1.1 일반적인 에이전트의 분류

에이전트의 타입, 정의, 이름은 기존 에이전트 시스템에서 사용되어지며, 다양한 방법으로 표현된다. (그림 3)은 에



(그림 3) 일반 특성에 의한 에이전트 분류

이전트 영역에 시스템과 애플리케이션에서 식별되는 공통적인 타입과 이름을 최대한 반영하여 분류한 것이다.

지능 에이전트와 미들웨어 에이전트는 세부적인 에이전트를 포함하여 표현하며, 유사 기능과 명칭을 가지고 있는 에이전트는 최대한 중복을 피하기 위해 정련화 하였으며, 일반적인 에이전트는 다음과 같다.

소프트웨어 에이전트(software agent)는 기반 환경과 상호 작용할 수 있고 자치권이 있는 소프트웨어 실체로 정의된다. 에이전트가 외부의 제어로부터 독립성을 가지고 있을 때 그 자치권의 정도에 따라 자동 에이전트(autonomous agent)라 한다. 대화식 에이전트(interactive agent)는 등급에 따라 표현될 수 있는 다른 엔티티와 기반 환경에 따라 통신이 가능하다. 적응 에이전트(adaptive agent)는 환경에 대한 다른 에이전트에 반응하는 능력이 있을 경우 적응성으로 고려된다. 호스트 컴퓨터에 단일 처리를 가지는 고정적인 에이전트가 존재하고 이동 에이전트(mobile agent)는 실행을 재개할 수 있는 새로운 호스트에 그들의 코드를 통해 이동하고 획득된다. 조정 에이전트(coordinative agent)는 목적에 대한 많은 개별적인 행위를 동등하게 조절하기 위해 존재한다. 지능 에이전트(intelligent agent)는 계획, 목적, 가정, 의도 등의 지식에 의해 기술되며, 학습을 통한 인공지능 시스템에 많이 정의되고 있다. 래퍼 에이전트(wrapper agent)는 소프트웨어 명세에 의해 유일하게 식별되는 비에이전트(non-agent)의 시스템이나 서비스에 연결하기 위해 정의되는 에이전트이다. 미들 에이전트(middle agent)는 공통적이고 기본적인 에이전트 기능을 포함하는 에이전트화된 미들웨어 서비스를 제공한다. 인터페이스 에이전트(interface agent)는 사용자 자신의 작업을 수행하기 위해 자동성과 학습을 강조한다.

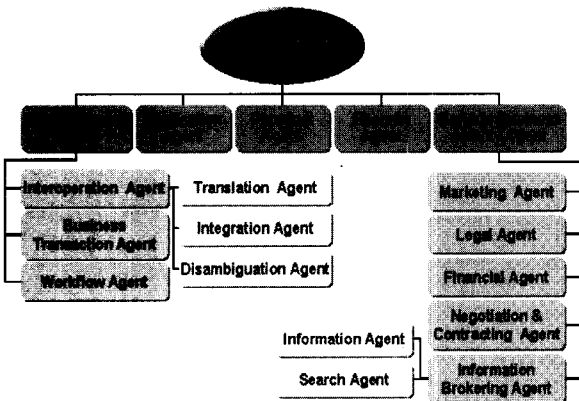
정보 에이전트(information agent)는 사용자 인터페이스 안에서 작동하고 그 사용자가 그 인터페이스를 동작시키고 그 하위에 있는 시스템을 조절하는 것을 도울 수 있는 에이전트이다. 스마트 에이전트(smart agent)는 외부 환경에 상호작용이나 재반응에 대한 학습이 가능하므로 실행성을

증가시키는 목적으로 현재 많은 연구가 진행되고 있다. 하이브리드 에이전트(hybrid agent)는 단일 에이전트 내에 둘 이상의 에이전트 개념을 조립하여 구성한 형태이다. 이종 에이전트(heterogeneous agent)는 두 개 이상의 다른 에이전트 분류에 의해 설치되고 통합되는 형태이며, 하이브리드 에이전트를 포함 가능하다.

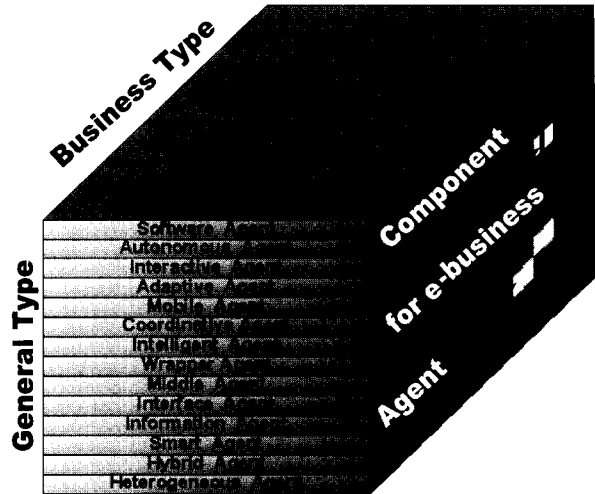
3.1.2 e-비즈니스 에이전트의 분류

현재 사용되는 에이전트의 애플리케이션 종류는 여전히 한계가 있지만, 에이전트에 대한 접근 방법이나 e-비즈니스 영역과 애플리케이션에 더 많이 적용되고 있다. e-비즈니스 영역에서 요구되고 운용되는 에이전트를 고려하여, 기능성이나 속성에 의존한 각각 다른 에이전트를 (그림 4)와 같이 분류하였다. 이는 5개의 기본 에이전트 타입과 세부적인 에이전트 타입으로 표현되며, 다음과 같다.

시스템 레벨 에이전트(system-level agent)는 분산 객체 구조의 상위에 존재하며, 에이전트 솔루션은 작업에 관련된 시스템을 구성하는 지원 기능과 분산 객체 기능의 확장으로써 전개된다. 애플리케이션 에이전트(application agent)는 B2B(Business To Business) 전자상거래 애플리케이션을 구성하는 응용 에이전트으로써 각각의 에이전트가 전문 영역에 특정화되어 있으며, 사용 가능한 정보에 대한 접근과 도메인 지식 자원을 제공한다. 개인 에이전트(personal agent)는 사용자 프로파일, 요청, 정보 수집의 관리와 조직화, 가시화를 지원하기 위해 사용자에게 직접적인 도움을 준다. 보안 에이전트(security agent)는 e-비즈니스 통신에서 요구되는 안전 서비스를 제공하며, 정보 보호 기능을 가지고 있다. 안전한 e-비즈니스를 위한 에이전트 지원은 인증, 위임, 비밀성, 자료의 완전성 그리고 부인 거절의 다섯 가지 별개의 범주로 나누어진다. 일반 비즈니스 행위 에이전트(general business activity agent)는 많은 상업 지원 활동을 수행하며, 비즈니스 업무에서 요구되는 정보를 발견하기 위해서 온라인 상에서 검색하는 에이전트나 정보 에이전트를 포함할 수 있다.



(그림 4) e-비즈니스 특성에 의한 에이전트 분류



(그림 5) ebA-CBD 참조 아키텍처

3.2 e-비즈니스 에이전트 개발을 위한 컴포넌트 참조 아키텍처

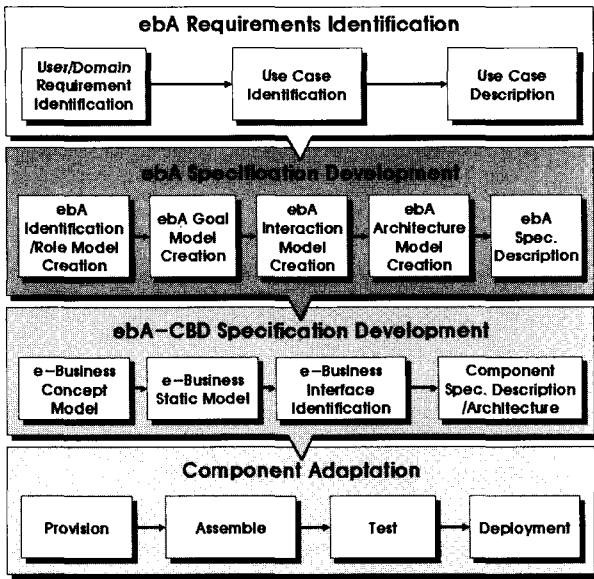
분류를 기반으로 e-비즈니스 에이전트를 위한 컴포넌트와 개발 프로세스의 메타 아키텍처로써 (그림 5)와 같이 구성된다. 이는 도메인 지향적인 컴포넌트 아키텍처이며, 14개의 일반적인 에이전트 타입과 11개의 세부적인 비즈니스 에이전트 타입으로 분리된 측면으로 구성되어 있다.

두 가지 영역은 상호 독립적이며 상호 참조의 성질을 가진다. 일반적인 에이전트 타입은 수평적인 특징을 가지며, 특정 e-비즈니스에 관련된 에이전트 타입은 수직적인 특징을 가진다. 또한, 일반 에이전트 타입은 에이전트 플랫폼, 프레임워크, 애플리케이션으로 대응된다. 에이전트 기반의 시스템이나 애플리케이션을 비즈니스 영역에 적용하거나 e-비즈니스 영역에서 에이전트 기술을 도입할 경우 참조 아키텍처를 통해 필요한 컴포넌트를 조립하여 구성한다.

개발된 컴포넌트는 참조 아키텍처에 따라 분류되며, 일반적인 에이전트의 타입과 비즈니스 속성에 따라서 하나의 영역에 배치된다. 또한, 에이전트 타입을 지향하는 시스템이나 비즈니스 영역에 필요한 에이전트를 응용하고자 할 경우, 이와 연관된 컴포넌트를 비즈니스 영역에서 식별하여 조립함으로써 시스템 구성이 가능하다.

4. ebA-CBD 프로세스

컴포넌트 기반으로 e-비즈니스 에이전트를 구성하는 것은 에이전트의 크기에 따라 하나의 컴포넌트나 그 이상의 컴포넌트로 구성될 수 있지만 컴포넌트로 구성하기 어려운 부분도 존재한다. 본 논문에서는 e-비즈니스 에이전트의 기능성에 초점을 두고 CBD 참조 모델을 기반으로 e-비즈니스 에이전트 도메인을 위한 컴포넌트 기반 개발 프로세스는 (그림 6)과 같다.



(그림 6) ebA-CBD 전체 프로세스

4.1 ebA 요구사항 식별

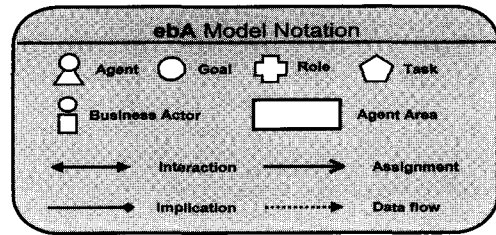
시스템 개발을 위해 관련된 특정 도메인 개념과 관련성을 보여야 하며, UML을 통한 e-비즈니스 도메인 분석을 기본으로 한다. e-비즈니스 도메인과 사용자, 시스템, 기반환경 등을 고려하여 문제 영역을 분석하고 요구사항 시나리오를 작성함으로써 개발 목표를 명확히 정의한다.

사용자의 관점에서 시스템이 어떻게 동작하는지 결정하여 전체 시스템 예측 활동은 스토리보딩 기법을 사용하여 요구사항 시나리오를 작성한다. 또한, 작성된 요구사항 시나리오를 기반으로 사용사례를 식별하고, UML의 유스케이스 다이어그램으로 작성한다. 이는 목표 시스템의 영역을 정의하고 전체 업무 영역 중에 어느 업무 기능을 책임질지 명확히 하는 것이며, 행위자와 기능간의 관계 및 부가적인 설명이 된다. 요구사항 식별단계에서는 후보가 될 수 있는 에이전트에 대한 이해와 에이전트를 기반으로 구성되는 기능적인 요소나 시스템 전체의 구성 파악이 주목적이다.

4.2 ebA 명세 개발

ebA-CBD 참조 아키텍처에서 에이전트의 영역을 식별하고 에이전트간의 연관성을 정의한다. 정의된 에이전트는 각각의 기능 및 특징을 중심으로 명세를 하게 된다. 이 명세 과정을 통해 에이전트를 개발하기 위한 적절한 컴포넌트의 구성이 가능하다.

사용자의 요구에 기반하여 ebA 명세와 4개의 모델을 생성한다. 이는 명세단계에 획득되며, 새로운 컴포넌트를 식별, 개발하기 위한 주요 자원으로 사용된다. ebA 모델은 더 높은 수준의 표현을 제공하기 위해 UML 메타모델로 확장된다. 본 절에서는 ebA 모델을 가시적으로 표현하기 위해서 새로운 구성요소로 작성되는 다이어그램을 기술한다.



(그림 7) ebA 모델 표기법

(그림 7)에 ebA 모델 개념과 관련성을 기호로 제시하였다. 기호는 UML에서는 언급되지 않지만 메타모델의 형태로 포함된 구성요소로서 각각의 관계와 용법은 다음과 같다.

- 상호작용 관계(Interaction) : 상호작용 모델을 작성할 때 뿐만 아니라, 다른 모델에서 두 개의 목표, 역할, 태스크 간의 메시지 교환이나 양방향 통신을 의미한다.
- 포함 관계(Implication) : 주로 목표 모델에 사용하게 되며, 하나 이상의 원소에 상태 속성을 가지고 있는 원소를 연결할 때 사용하게 된다.
- 할당 관계(Assignment) : AutonomousEntity의 속성을 가지고 있는 원소에 AutonomousEntity의 원소를 연결할 때 사용한다. AutonomousEntity로부터 화살표가 지정하는 다른 엔티티에 할당되는 것을 의미한다.
- 자료 흐름(Data Flow) : 생산되거나 소비되는 InformationEntity에 자료 생산 소비자를 연결하며, UML에서와 동일한 의미로 사용된다.

4.2.1 ebA 식별 및 역할 모델 생성

에이전트 역할에 초점을 두고 특징을 표현할 수 있도록 다이어그램을 생성한다. 행위규칙에 따라 동작하는 작업이나 자원의 제어, 어떤 이벤트에 반응하는지 등을 중심으로 식별하게 된다. 요구사항을 통해 얻어진 에이전트의 기본적인 요구사항을 통해 e-비즈니스 에이전트의 CBD 참조 모델에 어느 영역에 위치하는지를 식별하고 식별된 내용을 (그림 8)의 참조 모델 매트릭스를 이용하여 체크리스트를 작성한다. 또한, 작성된 매트릭스에 해당하는 에이전트의 이름을 명명하며, 이는 에이전트 명세 아키텍처 모델과 스펙 생성에서 목표 에이전트로 정의한다.

e-Business Agent Agent Type	System-Level Agent (S)			General Business Activity Agent (G)					System-Level Agent (S)	Security Agent (S)
	System Agent (S1)	Business Functional Agent (S2)	Business Agent (S3)	Business Agent (G1)	Business Agent (G2)	Business Agent (G3)	Business Agent (G4)	Business Agent (G5)		
Software Agent (SWA)										
Autonomous Agent (AUA)										
Interactive Agent (IA)										
Adaptive Agent (ADA)										
Mobile Agent (MA)										
Coordinable Agent (CA)										
Intelligent Agent (IA)										
Wrapper Agent (WA)										
Mobile Agent (MA)										
Intelligence Agent (IA)										
Information Agent (IA)										
Smart Agent (SA)										
Hybrid Agent (HA)										
Heterogeneous Agent (HA)										

(그림 8) ebA-CBD 참조 매트릭스

참조 모델 매트릭스에서 일반적인 에이전트 타입은 이름 기반 식별자와 e-비즈니스 에이전트 타입은 수치 기반 식별자를 정의하며, 조합되는 부분에 여러 개의 에이전트나 컴포넌트가 식별될 수 있으므로 순차코드를 고려한다. 두 코드의 조합은 요구되는 에이전트의 식별코드로 사용된다.

4.2.2 ebA 목표 모델 생성

에이전트간의 목표, 테스크, 상태와 의존성들을 표현한다. 목표와 테스크는 상태와 함께 연관되며 논리적인 의존성에 의해 연결될 수 있다. 상위의 목표는 하위의 서브목표의 집합이 달성됨으로써 충족될 수 있으며, 이러한 서브 목표를 식별하고 모델로 제시함으로써 에이전트 시스템을 이해하고 분석 가능하다. 또한, 테스크는 목표를 달성하기 위해 수행되며, UML 액티비티 다이어그램으로 표현 가능하다.

4.2.3 ebA 상호작용 모델 생성

설계단계에서 어떻게 통신하는지를 상세하게 모두 나타내는 것으로 어느 에이전트와 어떻게, 언제 통신이 이루어지는지를 에이전트와 역할을 중심으로 표현한다. 상호작용 모델은 새로운 상호작용이 발견되기 전까지 몇몇의 상호작용을 통해서 정련화된다. 이 모델에서는 상호작용을 중심으로 각각의 참여로 달성되고 제공되는 트리거 조건과 정보처럼 선택적인 정보에 상호작용 가능한 initiator, responder, motivator를 표현한다.

4.2.4 ebA 아키텍처 모델 생성

식별된 e-비즈니스 에이전트는 각각의 기능에 따라 에이전트 타입에 대응되며, 구별될 수 있는 특징을 가진다. 동일한 에이전트 타입은 목적에 따라 다른 에이전트와 협동이나 협상 등의 기능을 수행한다. 이러한 다른 에이전트와의 관계 및 상호 작용하는 동적인 성질을 명세하기 위한 부분이 에이전트 아키텍처 모델을 생성하는 것이다.

본 논문에서는 비즈니스 사용자와의 관계 및 식별된 에이전트를 에이전트 영역에 정의하고 각 에이전트간의 관계를 ebA 아키텍처 모델로 제시한다. 에이전트 영역은 두 형태로 나뉘는데, 다른 에이전트와 통신하거나 가용성을 알리기 위한 영역과 사용자와 상호작용하기 위한 사용자 영역으로 나뉜다.

4.2.5 ebA-명세 작성

작성된 ebA 아키텍처 모델을 기반으로 각각의 에이전트 별로 에이전트의 특징, 공유되는 자원, 접근에 대한 정보, 모델 정보 등을 명세한다. 이는 에이전트의 기능적, 비기능적인 특징을 표현하는 것으로서 기능적인 명세는 UML의 클래스 다이어그램, 시퀀스 다이어그램을 이용하여 각 에이전트마다의 속성과 행위를 기본적으로 정의한다. 또한, 정의된 에이전트간의 비동기적으로 전송되는 메시지와 이벤트에 대한 부분을 표현하며, <표 1>은 ebA-명세 항목과 해당하는 내용을 나타낸 것이다.

<표 1> ebA-명세 목록

Item	Description
Agent Name	식별된 ebA 이름
e-business Type	ebA-CBD 참조모델에서 e-비즈니스 에이전트 타입
General Agent type	ebA-CBD 참조모델에서 일반적인 에이전트 타입
Identification Code	ebA-CBD 참조 매트릭스에서 식별 코드
Access Information	에이전트가 접근, 저장, 갱신할 정보
Produce Information	에이전트가 동작하는 동안 생성하는 파일/정보
Related Agent	협상이나 협동하거나 메시지를 주고받을 에이전트
Information Diagram	에이전트의 속성 및 행위 표현
Operation Diagram	에이전트간의 행위적인 순서 표현

4.3 ebA-CBD 명세 개발

명세개발은 요구사항 단계의 Use Case 모델과 ebA 모델과 명세를 중심으로 컴포넌트를 식별하고 설계하는 전반적인 과정을 포함한다. ebA 에이전트에서 분석된 에이전트의 모델에서 속성들을 컴포넌트 개발을 위한 속성과 행위들로 매칭시키고 인터페이스를 식별해내는 과정이 수반된다. 또한, 컴포넌트간의 의존성을 통해 에이전트를 구성할 수 있도록 컴포넌트 아키텍처도 정의한다.

기존에 제시되어진 컴포넌트 명세 기법에 새롭게 요구되는 명세 특성들을 포함하고 ABCD 아키텍처에 기반한 특징을 포함하여 컴포넌트 명세를 작성한다. 이는 컴포넌트 개발을 위한 상세 명세로서 전개시 조립을 위한 명확한 의미적인 플러깅 지점을 확보하고 비즈니스 프로세스의 계층적 실현을 위해 <표 2>와 같은 명세 항목으로 정의한다. 제시된 명세 항목은 컴포넌트 구현을 위해 명세된 것이지

<표 2> 컴포넌트 명세 목록

Item	Description
Category	컴포넌트가 속한 비즈니스 도메인의 세부 기능적인 분류에 따른 컴포넌트 군
Component Diagram	인터페이스를 포함한 컴포넌트의 정의
Component Name	식별된 컴포넌트 이름
Classification Code	ABCD 아키텍처에 기반한 컴포넌트의 식별 코드
Short Description	컴포넌트 전반에 대한 기능, 동기, 동작 과정, 제약사항 등에 대한 서술
Glossary	컴포넌트 명세에 사용된 용어의 의미 설명
Component Context Diagram	컴포넌트의 주요 기능을 서술
Component Interaction Diagram	컴포넌트 실행시 관련된 컴포넌트들간의 관계성을 표현
Component Sequence Diagram	컴포넌트 자체의 행위적 순서 표현
Component Relationship Diagram	컴포넌트가 제공하고 제공받는 인터페이스와 다른 컴포넌트간의 관계를 표현
Interface Description	공급하고 요구하는 인터페이스의 의미적 서술, pre/post-condition, 입력/출력 결과 명시
Usage Scenario	컴포넌트 사용을 위한 인증된 시나리오

만, 기본 명세 단계에서는 음영으로 된 부분을 작성한다. 이는 식별될 수 있는 컴포넌트의 카테고리과 이름, 식별 코드 등을 작성한다. 명세는 컴포넌트 개발과 컴포넌트의 획득을 위해 작성되는 가이드라인의 역할을 한다.

5. 사례 연구 및 평가

앞장에서 제시한 ebA-CBD 프로세스를 기반으로 전자상거래 분야에 상품정보를 전송하는 e-메일링 서비스에서 에이전트를 컴포넌트로 개발하는 과정을 사례로 제시한다. 각각의 단계별로 산출물을 제시하고, 컴포넌트 개발 과정의 ebA와 ebA-CBD 명세를 작성한다. 또한, e-비즈니스 영역에서 에이전트가 운영되는 것을 프로토타입으로 나타냈다.

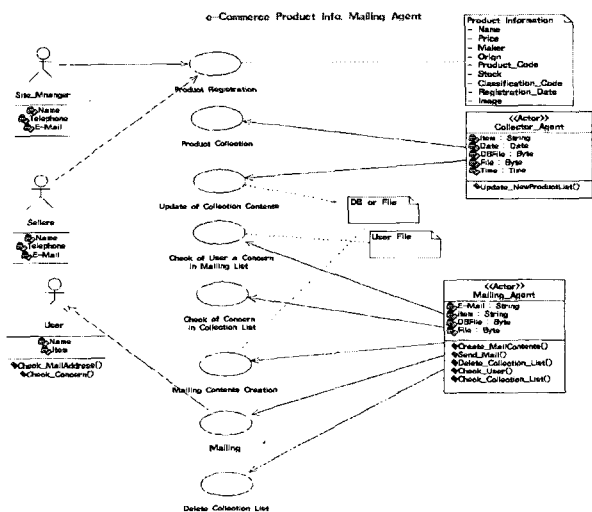
5.1 e-CPIMAS의 ebA 요구사항 식별

e-비즈니스 영역에서 B2B나 B2C(Business To Customer) 형태의 비즈니스 활동이 많이 이루어진다. 전자상거래 사이트에서 새로 등록된 신상품에 대한 정보를 관리자가 작성하여, 메일링 리스트에 등록된 사용자에게 보내는 서비스를 에이전트 통해 하고자 한다. 에이전트를 고려하는 시스템을 e-Commerce Product Information Mailing Agent System (이하 e-CPIMAS)이라 명명한다.

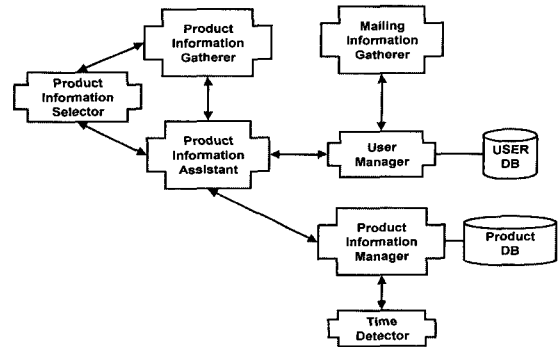
(그림 9)는 사용자 및 시스템 요구사항을 기반으로 식별된 액터와 유스케이스이다. 핸들링 되는 데이터베이스와 관련 정보를 메모를 사용하여 이해를 용이하게 하였다. 에이전트로 식별된 것은 액터의 스테레오 타입으로 제시하였다.

5.2 e-CPIMAS의 ebA 명세 개발

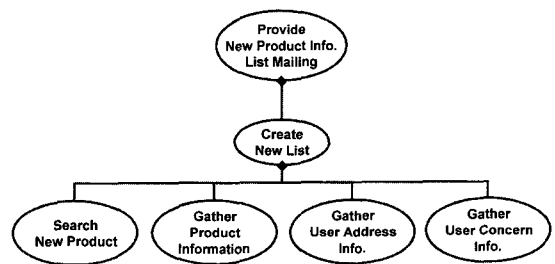
ebA 식별 및 역할 모델 생성을 위해서 유스케이스 다이어그램으로 식별된 에이전트의 기본적인 사용사례를 중심으로 ebA-CBD 매트릭스를 통해 소프트웨어 에이전트 형



(그림 9) e-CPIMAS 유스케이스 다이어그램



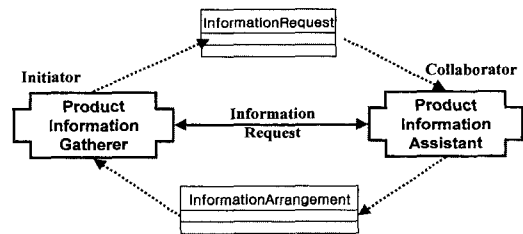
(그림 10) e-CPIMAS 역할 모델



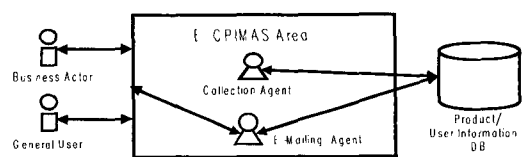
(그림 11) e-CPIMAS 목표 모델

태의 개인 에이전트인 수집 에이전트를 식별하고, 정보 브로킹 기능을 수행하는 정보에이전트인 e-메일링 에이전트를 식별하였다. 두 개의 에이전트를 중심으로 전체 시스템의 일부로서 에이전트를 운영할 수 있도록 구성한다.

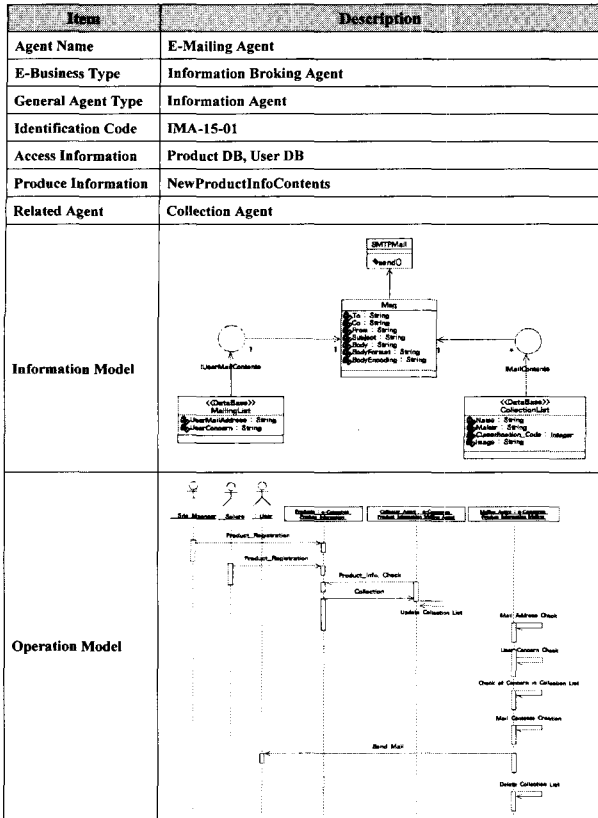
(그림 10)은 메일과 상품 정보를 획득하기 위한 역할 모델이다. 여러 개의 역할들은 다른 역할들과 상호작용하면서 최종 목표를 달성하게 된다. (그림 11)은 새로 등록된 상품 정보를 메일로 보내는 것을 최상의 목표로 이를 해결하기 위해 요구되는 sub-목표들로 구성된 모델을 제시하였다. 상품 정보를 수집하기 위해 정보를 요청하는 초기 실행을 담당하는 역할과 정보를 제공해 주는 상품 정보 보조자간의 관계를 보여주는 상호작용 모델은 (그림 12)와 같다.



(그림 12) e-CPIMAS 상호작용 모델



(그림 13) e-CPIMAS 아키텍처 모델



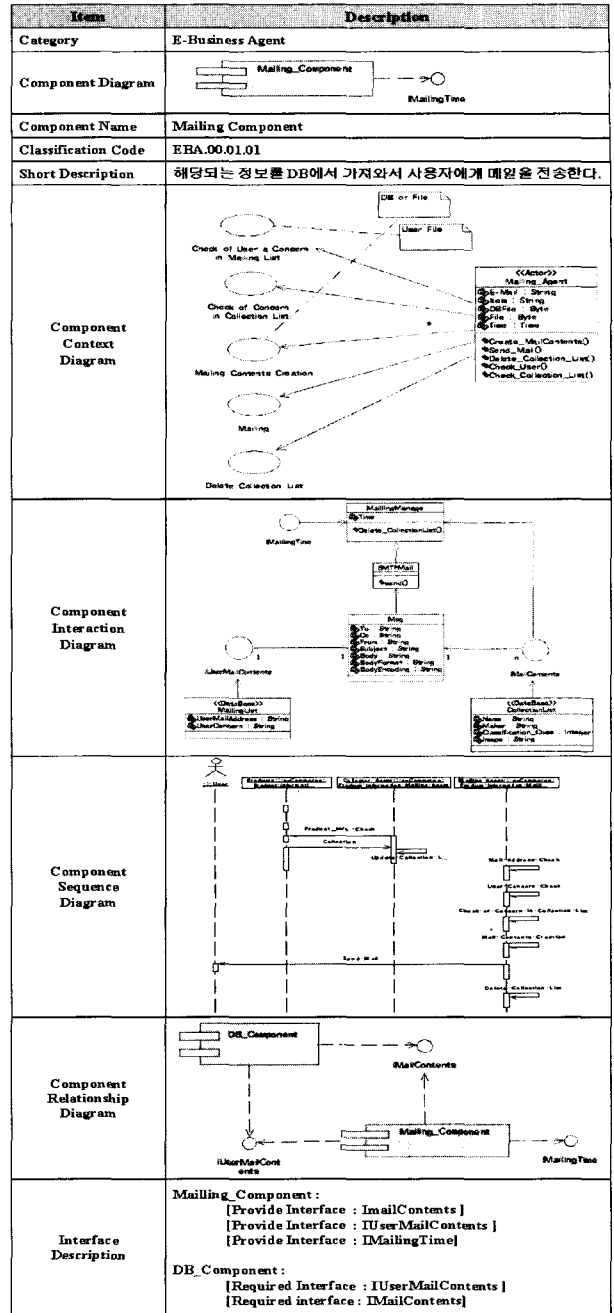
(그림 14) e-CPIMAS에서 e-메일링 에이전트 명세

(그림 13)은 아키텍처 모델로 e-CPIMAS의 에이전트 영역에 존재하고 있는 에이전트와 외부 시스템이나 사용자의 관계를 나타낸 모델링이다. (그림 14)는 작성된 모델을 기반으로 최종적으로 작성된 에이전트 시스템의 명세이다. 전체 e-CPIMAS에서 e-메일링 에이전트의 내용만 명세한 것으로써 일반적인 항목들과 정보 모델, 행위 모델로 에이전트의 기본 속성과 기능을 제시하였다. 행위 모델은 다른 에이전트와의 관계를 시퀀스 다이어그램으로 표현하였다.

5.3 e-CPIMAS의 ebA-CBD 명세 개발

앞 절에서 식별된 에이전트 기능과 모델을 기반으로 컴포넌트로 구성하기 위한 모델을 작성한다. 정적 모델과 인터페이스 식별을 통해 컴포넌트를 작성하고 기능적, 비기능적인 요소를 중심으로 ebA-CBD 명세를 개발한다.

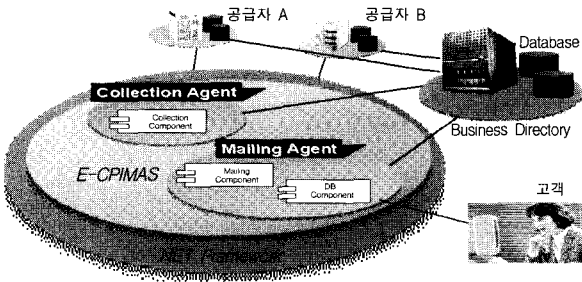
컴포넌트는 수집 에이전트를 위한 Gathering_Component와 Retrieval_Component로 구성되며, e-메일링 에이전트는 DB_Connector와 Mailing_Component 두 개로 구성된다. 특히, 메일링 컴포넌트는 (그림 15)에 컴포넌트 명세를 제시하였다. 명세에서 컴포넌트 상호작용을 나타낸 부분은 Mailing_Component의 내부 클래스와 외부 컴포넌트인 DB_Component와 연결된 것을 나타내었다. 인터페이스 명세는 Provide와 Required Interface를 제시하고 간략히 명세하였다.



(그림 15) e-CPIMAS에서 메일링 컴포넌트 명세

5.4 e-CPIMAS 프로토타이핑

에이전트의 기능을 하는 컴포넌트의 실행을 프로토타입 형식으로 제시하고자 한다. 개발 환경 및 운영환경은 (그림 16)과 같이 .Net 프레임워크이며, C#.Net으로 개발하였다. 컴포넌트는 .Net 프레임워크 상에서 구동하며, 컴포넌트의 주요 기능은 에이전트 기능을 한다. (그림 17)은 수집 에이전트가 주기적으로 새로 등록된 상품 정보를 수집하는 과정과 사용자의 메일 정보 및 관심 분야 정보를 획득하고, 획득된 정보와 수집 에이전트에 의해서 수집된 상품 리스트에서 관심분야를 매칭하는 것을 나타낸다.

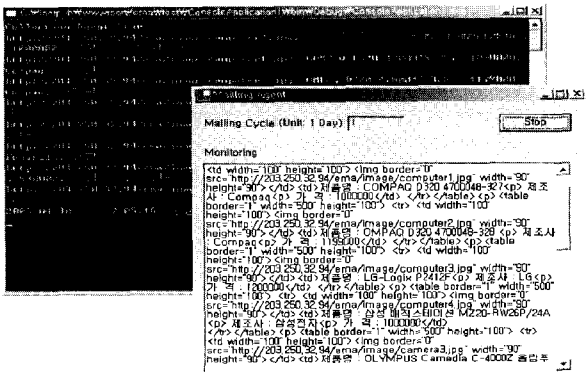


(그림 16) e-CPIMAS의 운영 환경

5.5 평가

기존의 에이전트를 개발하기 위해서 컴포넌트 기반 방법론을 고려한 것은 아직 없으나 기존의 방법론과 본 논문에서 제시한 방법과 단순 비교를 하고자 한다. 일반적으로 기존의 컴포넌트 기반 방법론의 에이전트 영역을 고려하는지의 여부와 에이전트를 위한 분석, 설계 방법을 제공하는지의 여부를 중심으로 비교하고자 한다. 비교 대상으로는 컴포넌트 개발 방법으로 이미 공신력을 가지고 있는 RUP(Rational Unified Process)[8], Catalysis[9], UML Components [10]를 대상으로 한다. 또한, 에이전트 영역에서 컴포넌트 기반 개발을 고려하는지의 여부도 고려하고자 한다. <표 3>은 ebA-CBD 방법과 비교한 것을 나타낸다.

에이전트 영역에서 에이전트 식별방법으로 AUML(Agent Unified Modeling Language)[11]과 Kinny의 AAI(the Australian AI Institute)[12] 방법론과 Wooldridge의 Gaia[13]를 고려하였다. 소프트웨어 공학적인 접근 방법으로 에이전트를 식별하는 방법은 제공하지만, 컴포넌트 기반 개발을 고려하지 않는다. 또한, 객체중심의 관점에서 에이전트 기반 시스템을 접근하며, 문제영역에서 행위자 혹은 역할들을 찾아내고 분석하여 에이전트를 식별하는 것이 주요 프로세스이다. 또한, 표준 모델링 방법을 적용하는 부분이 일부분이므로, 모델간 호환성이 떨어진다. 에이전트를 컴포넌트화할 수 있는 모델이나 방법을 제공하지 않는다. 그러나, 에이전트 식별이 용이함으로 컴포넌트 기반 개발에서 분석 단계에 도입이 가능하다.



(그림 17) 수집 에이전트와 메일링 에이전트의 실행 화면

<표 3> 기존의 개발 방법론과 ebA-CBD 방법의 비교

평가 항목	RUP	Catalysis	UML Components	AUML	AAI	Gaia	ebA-CBD
적용 목적	객체지향 및 컴포넌트 기반 시스템	컴포넌트 기반 개발	모델 기반 컴포넌트 개발	멀티 에이전트 시스템 모델링	에이전트 시스템	객체 중심의 에이전트 개발	e-비즈니스 에이전트 컴포넌트 개발
에이전트 분석, 설계 방법 제공	△	△	△	○	○	○	○
에이전트 식별 방법 제공	△	△	△	○	○	○	○
에이전트 모델링 제공	×	×	×	○	△	△	○
컴포넌트 기반 개발 제공	○	○	○	×	×	×	○
확장성	○	○	○	×	△	△	△
모델 재사용성	○	○	○	○	△	△	△
모델 유지보수 용이성	○	○	○	△	×	×	△

○ 제공함 △ 부분 제공 × 제공하지 않음

본 논문에서 제안하는 ebA-CBD는 에이전트형 컴포넌트 개발이 가능하고, 이를 통한 기업이나 조직의 비즈니스 기능의 확대와 통합을 빠르고 효과적인 방법을 제공한다. 또한, 에이전트를 통한 유연한 서비스를 제공하고, 컴포넌트 기술은 개발의 효율성을 보장함으로써 다른 방법론이 제공하지 않는 최신의 요구사항을 해결할 수 있다.

6. 결론

비즈니스 영역에 적용가능한 에이전트를 컴포넌트 기반으로 개발하기 위해 ebA-CBD 프로세스를 제안하고, 전자상거래 영역에서 주로 사용되는 메일링 서비스를 에이전트 컴포넌트로 구성하는 것을 사례로 제시하였다. 체계적인 프로세스 도입과 e-비즈니스 에이전트의 ebA-CBD 참조 모델을 기반함으로써 분석된 에이전트 영역의 컴포넌트 식별을 쉽게 하고, 개발에 효율성을 제공한다. 또한, 명세를 제공함으로써 적용가능한 컴포넌트 선택의 가이드라인이 되고, 컴포넌트 생성을 위한 기반 모델로써 재사용 된다.

ebA-CBD 프로세스의 제안에 따른 기대효과는 컴포넌트의 재사용을 통하여 e-비즈니스 영역에서 목표로 하는 소프트웨어 개발 기간을 크게 단축시키며, 그에 따른 개발비용은 감소된다. 또한, 테스트와 검증된 컴포넌트를 사용함으로써 개발된 에이전트 모델 상에서의 소프트웨어 품질을 크게 향상시키는 효과를 가져온다.

참고 문헌

- [1] Lawrence Wilkes, Understanding Component Based Development, Addison-Wesley, 2000.
- [2] Jun Han, "Characterization of Components," Proceeding of 1998 International Workshop on Component-Based Software Engineering, pp.65-70, Apr., 1998.
- [3] Nicholas R. Jennings, "On agent-based software engineering," International Journal of Artificial Intelligence, Vol.117,

No.2, pp.277-296, 2003.

[4] Mike P. Papazoglou, "Agent-Oriented Technology in support of E-Business," *Communications of the ACM*, Vol.44, No.4, pp.71-77, Apr., 2001.

[5] Richard Evans, "MESSAGE : Methodology for Engineering Systems of Software Agent," EURESCOMP Project P907 Publication, Sep., 2001.

[6] 김행곤, 한은주, 신호준, "영역별 컴포넌트 분류 방법에 관한 연구", ETRI 최종연구보고서, 1999.

[7] 김행곤, 차정은, 김지영, 신호준, "컴포넌트 저장소 형상관리 시스템에 관한 연구", ETRI 최종연구보고서, 2000.

[8] Philippe Kruchten, "The Rational Unified Process : An Introduction 2nd Edition," Addison-Wesley, 2000.

[9] Desmond F. D'Souza and Alan C. Wills, *Objects, Components and Frameworks with UML*, Addison-Wesley, 1998.

[10] John Cheesman and John Daniels, *UML Components*, Addison-Wesley, 2001.

[11] Bernhard Bauer, Jorg P. Muller and James Odell, "Agent UML : A Formalism for Specifying Multiagent Interaction," *Proceeding of 2000 Agent-Oriented Software Engineering*, pp.91-103, May, 2001.

[12] David Kinny, Michael Georgeff and Anand Rao, "A Methodology and Modelling technique for systems of BDI agents," *Proceedings of 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, Springer-Verlag : Berlin, Germany, pp.56-71, 1996.

[13] Michael Wooldridge, Nicholas R. Jennings and David Kinny, "A Methodology for Agent-Oriented Analysis and Design," *Proceeding of International Conference on Autonomous AGENTS '99*, pp.69-76, Mar., 1999.

[14] Martin L. Griss, Gilda Pour, "Accelerating Development with Agent Components," *IEEE Computer*, Vol.34, No.5, pp.37-43, May, 2001.

[15] Divine Ndumu and Hyacinth Nwana, "Research and Development Challenges for agent-based systems," *IEEE Proceeding on Software Engineering*, Vol.144, No.1, pp.2-10, Feb., 1997.

[16] Mike Wooldridge, "Agent-Based Software Engineering," *IEEE Proceeding on Software Engineering*, Vol.144, No.1, pp.26-37, Feb., 1997.

[17] James Odell, H. Van Dyke Parunak and Bernhard Bauer, "Extending UML for Agents," *Proceeding of Agent-Oriented Information Systems Workshop at the 17th International Conference on Artificial Intelligence*, Vol.11, No.3, pp.303-328, 2001.

[18] Hideki Hara, Shigeru Fujita and Kenji Sugawara, "Reusable Software Components based on an Agent Model," *Proceeding of 7th International Conference on Parallel and Distributed Systems Workshops*, pp.447-452, Jul., 2000.

[19] Agent Platform Special Interest Group, "Agent Technology green paper," *OMG Document agent/00-0-01 Version 1.0*, at URL : <http://www.objs.com/agent/index.html>, 2000.

[20] Kutluhan Erol, Jun Lang and Renato Levy, "Designing Agents from Reusable Components," *Proceeding of International Conference on Autonomous Agents '00*, pp.76-77, Jun., 2000.



김 행 곤

e-mail : hangkon@cu.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

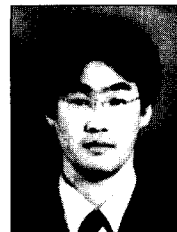
1988년~1989년 AT&T 객원 연구원

1990년~2000년 대구효성가톨릭대학교 컴퓨터공학과 부교수

2001년~2002년 Central Michigan University 교환교수

2000년~현재 대구가톨릭대학교 컴퓨터공학과 교수

관심분야 : CBSE, AOSE, 소프트웨어 재공학, CASE, 유지보수
자동화 틀, 요구공학 및 도메인 공학



신 호 준

e-mail : component@cu.ac.kr

1998년 경일대학교 전자계산학과(공학사)

2000년 대구가톨릭대학교 전산통계학과
(이학석사)

2003년 대구가톨릭대학교 전산통계학과
(이학박사)

관심분야 : CBSE, AOSE, 재공학, 분석·설계 모델링