

PDA 지원 에이전트 컴포넌트 개발에 대한 연구

김 행 곤[†]

요 약

최근 무선인터넷의 중요성이 강조되면서 모바일 단말 장치는 모바일 비즈니스 프로세스에서 중요한 역할을 담당한다. 특히, 모바일 단말 장치는 차세대 기술인 임베디드와 유비쿼터스 컴퓨팅의 주요 단말 장치이며 개발 대상이기도 하다. 또한, 모든 환경이 무선 인터넷 환경으로 이동되면서 무선 컴퓨팅 환경으로 도입하기 위한 서비스 기술에 많은 관심을 갖게 되었다. 그 중에서도 모바일 에이전트는 기존 인터넷 하부 구조상에 기능들을 전개하는 새로운 추상화 개념을 제공하며, 이러한 시스템은 융통성, 적용성, 확장성, 자율성 등을 요구하고 있다. 그러므로 이를 위해서는 새로운 소프트웨어 개발 방법론의 적용을 통해 해결하는 것이 중요하다.

따라서, 본 논문에서는 플러그 가능하고 독립성이 보장되는 조립 가능한 컴포넌트를 기반으로 에이전트 아키텍처 모델을 작성한다. 이러한 접근은 서블릿으로 컴포넌트 랩핑을 전개할 수 있으며, 이는 PDA 모바일 에이전트를 개발하기 위한 모델과 컴포넌트에 대한 연구로 행해진다.

키워드 : 에이전트, 에이전트 컴포넌트, 컴포넌트 개발 방법, PDA

A Study on the Agent Component Development Support to PDA

Haeng-Kon Kim[†]

ABSTRACT

In the focusing on the important of wireless internet, mobile terminal device plays a central role in tracking and coordinating terms in mobile business processing. Especially, mobile device has been considered as a key technology for embedded software and ubiquitous era. Because existing web environments is moving to wireless internet, the new concepts for wireless internet computing environments has gained increasing interest. Mobile agents provide a new abstraction for deploying over the existing infrastructures. Mobile application systems require the flexibility, adaptability, extensibility, and autonomous. New software developments methodology is required to meet the requirements.

In this paper, we present an agent architectures model that allows compassable components with pluggable and independable. Our approach involves wrapping components inside a servlet. We have used the model and components to develop the PDA mobile agent.

Key Words : Agent, Agent Component, CBD, PDA(Personal Digital Assistant)

1. 서 론

IT 산업에서 무선 인터넷의 중요성이 강조되면서 모바일 단말 장치에 대한 연구가 활발히 진행되고 있다. 특히, 무선 인터넷 환경으로 도입하기 위한 서비스 기술이 요구되는데, 특히 PDA 사용자의 능동적, 지능적인 서비스 요구사항을 지원하며 차세대 지능형 유무선 연동 서비스 시스템 설계를 위해 모바일 영역에 대한 기술이 요구된다[1-3].

에이전트지향 기술은 복잡한 소프트웨어 시스템을 분석, 설계하고 구축하는 흥미로운 새로운 수단을 나타낸다. 이는 소프트웨어공학에서 현재의 기술을 개선하고 응용범위를 확장할 수 있는 가능성을 가지고 있다.

또한, PDA 애플리케이션 시스템은 점점 더 융통성, 적용성, 확장성, 자율성 등을 요구하고 있으며, 이러한 요구사항은 새로운 소프트웨어 개발 방법론의 적용을 통해 해결하는 것이 중요하며 가속화되고 있다.

따라서 본 논문에서는 기능의 모듈성과 독립성이 보장되고 조립 가능한 컴포넌트를 기반으로 동적이고 복잡한 비즈니스 영역에 적용 가능한 에이전트 모델을 작성하고 작성된 모델을 기반으로 하는 PDA 지원 에이전트 컴포넌트 개발을 목표로 한다.

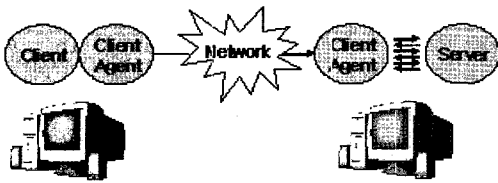
2. 관련 연구

2.1 PDA 에이전트 관련 기술

프로그램 자체가 네트워크를 돌아다니며 수행되는 프로그램인 PDA 에이전트라 한다. 기존의 메시지 전달 방식과는 달리 직접 작업을 수행할 수 있는 소프트웨어 객체를 전달

* 본 연구는 산학협동재단 연구비 지원사업에 의해 수행되었음.

† 종신회원 : 대구가톨릭대학교 컴퓨터정보통신공학부 교수
논문접수 : 2005년 8월 17일, 심사완료 : 2005년 12월 5일

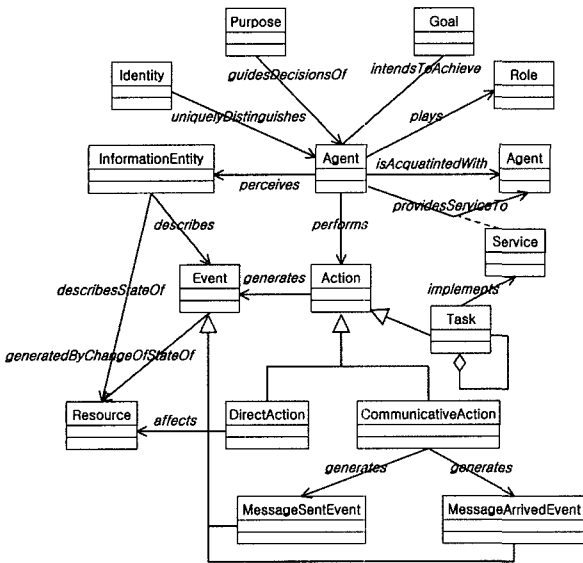


(그림 1) PDA 에이전트 모델

하는 방식을 이용하는데 RPC와 같은 방법에서처럼 지속적으로 많은 데이터를 주고받는 것이 아니라 프로그램 객체를 직접 이동시키기 때문에 성능이 그다지 좋지 않은 네트워크에서도 비용이나 자원을 보다 효율적으로 활용할 수 있다는 장점이 있다. 다음 (그림 1)과 같이 agent가 다른 호스트로 이동하여 그 곳에서 실행한 후 결과를 다시 가져오는 agent를 보여주는데 PDA agent에게 필요한 요소는 개발된 애플리케이션에서 행동하는 주체인 agent와 이러한 agent가 자리 잡고 작업을 수행할 가상 위치인 장소가 있어야 한다[4, 5].

2.2 에이전트 컴포넌트

PDA 에이전트는 개별 사용자의 필요를 충족하기 위하여 자율적으로 사용자를 대신하여 조건과 의사결정을 할 수 있는 도구이다. 특히, 모바일 인터넷과 웹이 확산됨에 따라 비즈니스 영역의 사용자들이 웹 안에 포함되어 있는 막대한 정보를 다루기가 점차 어려워지고 있으므로 향후 에이전트는 웹상의 정보를 이용하는데 있어 없어서는 안 될 중요한 기술이다. 또한, 에이전트는 자신의 도메인에 대한 확실한 지식을 가지고 있어야 하며, 행위를 이끄는 확실한 목적에 달성하고 유지하는 것에 대한 반응성을 가지고 있다. 또한, 환경과 사건에 대한 객체의 상태를 모니터링 가능해야 하며, 에이전트간의 상호작용성은 잘 정의된 의미론에 의해 통신 행위가 기술된다[6-8].



(그림 2) 에이전트 메타 모델

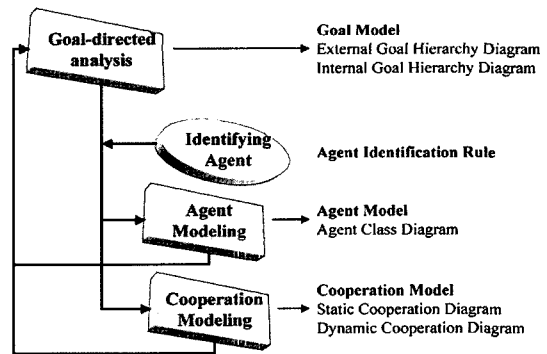
에이전트는 환경의 변화에 대응하며, 제반응과 사용에 따라 기반 환경과 다른 에이전트간의 상호 작용하는 시험적인 소프트웨어 컴포넌트의 성격을 가지고 있다. 또한, 사용가능한 몇몇 기능을 행할 능력을 가진 원자적인 자동성을 가진 엔티티로써, (그림 2)와 같이 개념적인 관련성을 나타내고 있다. 기능적인 능력은 agent의 서비스로써 나타나며, 이는 객체의 행위와 유사하다.

2.3 에이전트 식별 방법론

UML은 객체지향 시스템이나 모델 기반 개발에 유용한 표기법과 의미론을 제공하지만, 에이전트 영역의 의미적인 요소들을 포함하지 않으므로 에이전트와 에이전트 기반 시스템을 모델링하기에 불충분하다. 또한, 에이전트 기반 시스템 개발을 명세하기 위해 존재하는 정형적인 방법도 아직 고려되지 않고 있으며, 에이전트 기반 프로그래밍을 이끌기 위해 명세기술은 소프트웨어공학 전체 단계를 지원해야만 한다[1]. 에이전트 기반 시스템에 대한 표준화 연구로 FIPA (Foundation for Intelligent Physical Agents)와 OMG(Object Management Group)의 에이전트 그룹에서 UML의 확장에 대한 연구를 진행 중이다.

또한, AUML, Kinny의 AAIL(the Australian AI Institute), Wooldridge의 Gaia, Collinot의 Cassiopeia 방법론 등이 있다.

에이전트 지향의 소프트웨어 모델링 방법에서는 앞서 네 가지 방법론과는 차이가 있다. 앞의 네 가지 방법론에서는 행위자 혹은 역할과 같은 객체 중심적 관점에서 에이전트 기반 시스템을 개발하나 이 방법에서는 앞의 식별 방법과는 달리 목표 중심의 관점에서 에이전트를 고려한다. 먼저 문제영역에서 시스템의 최상위 목표를 정의하고, 그 최상위 목표를 달성하기 위한 하위 목표로 나눈다. 나뉜 하위 목표는 다시 다른 하위 목표로 계속 나뉜다. 이렇게 나뉜 목표는 하나의 계층적 구조를 가지게 되며, 이러한 목표들은 다시 각각 분석되어 에이전트로 식별된다. (그림 3)은 Goal-directed agent-oriented modeling 방법론에서 작성되는 model과 agent 식별 과정을 제시한 것이다[1, 2, 4].



(그림 3) 목표 기반 에이전트 지향 모델링 방법론

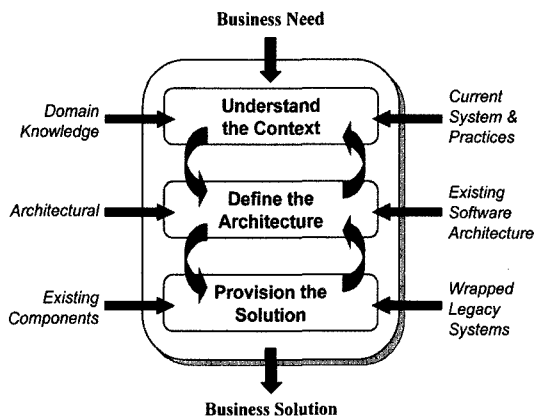
2.4 컴포넌트 기반 개발 방법론

컴포넌트 기반 방법론은 일반적으로 모델 기반으로 소프

트웨어 컴포넌트를 설계하고 구현하는데 솔루션을 제공할 수 있다. 현재 소프트웨어 시장에서 주목할 만한 컴포넌트 기반 개발 방법론으로는 Catalysis, Advisor, SELECT Perspective, RUP(Rational Unified Process), UML Components 등이 있으며, 그 중 Catalysis 방법론만 언급하기로 한다[6, 7].

Catalysis 방법론은 Desmond F. D'Souza와 Alan C. Wills가 이론적인 접근법을 제시하고 있으나 학문적이어서 실제화하기 위해서는 프로세스의 정리가 필요하다.

Catalysis 방법론은 (그림 4)에서와 같이 요구사항을 반영한 시스템이 만들어지기까지 Understand the Context, Define the Architecture, Provision the Solution으로 나누어 개발한다[3].



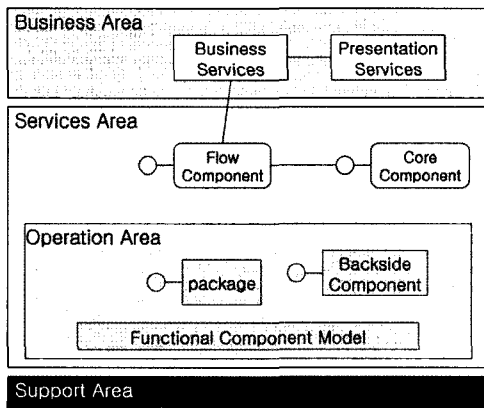
(그림 4) Catalysis 방법론

3. PDA 지원 에이전트 아키텍처

3.1 논리적 계층

웹 서비스 아키텍처를 컴포넌트 기반으로 구성하기 위해서 Business, Services, Operation, Support 4개의 논리적 계층 구조를 (그림 5)와 같이 구성한다.

Business Area는 사용자가 사용하는 단말장치와 외부 시스템이며 Presentation 부분이 포함된다.



(그림 5) 논리적 계층

Services Area는 웹 서비스가 이루어지는 부분으로 사용자 계층을 관리할 수 있는 비즈니스 로직을 포함하고, 비즈니스 계층에게 서비스를 제공한다. 이 계층에는 Facade component와 웹 스크립트가 위치한다. Facade component는 기능적으로 Core와 Flow로 분류할 수 있다. Core component는 이미 구축되어 있는 시스템에 웹 서비스가 가능하도록 하며, 자체에 비즈니스 로직을 포함할 수 있다. Flow component는 이러한 시스템 내부의 웹 서비스를 통합 관리 또는 흐름을 제어한다. 또한, 웹 스크립트를 사용하여 운영 계층의 컴포넌트의 데이터를 XML형식의 SOAP 메시지로 변환함으로써 웹 서비스의 기능을 수행할 수 있다.

Operation Area은 실제 비즈니스 로직을 포함하는 부분으로서 Backside component와 package component로 구성된다. component는 CBD 방법을 통해 개발된 공용 컴포넌트 또는 COTS(Commercial Off The Shelf) 컴포넌트를 말한다.

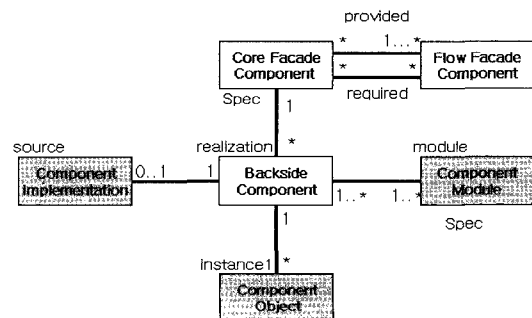
Support Area은 Operation Area의 비즈니스 로직이 요청하는 데이터에 대한 접근 또는 실제적인 데이터를 제공하는 계층으로 데이터를 생성하는 컴포넌트 또는 데이터베이스 등이 위치한다.

3.2 웹 서비스 Facade와 Backside component

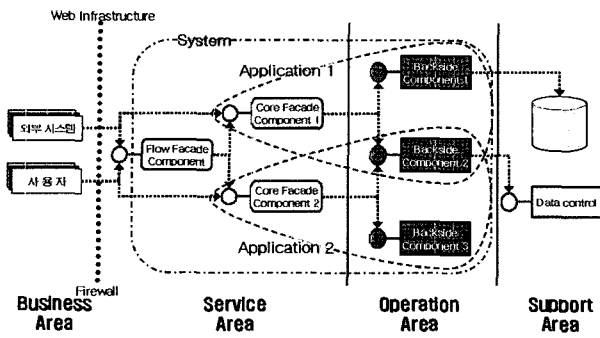
웹 서비스를 구성하기 위한 요소들을 식별하고 그들 간의 관련성을 파악하여 논리적 계층에 위치시킬 수 있다. 또한 이들 간의 관계를 계층 중심으로 정의함으로써 구현관점에서의 웹 서비스 위상의 정의가 가능하다. 웹 서비스의 논리적 계층을 바탕으로 (그림 6)와 같이 웹 서비스 컴포넌트 메타 모델을 제시한다. 웹 서비스는 하나의 Core Facade component와 다수의 Backside component로 구성이 되며, Backside component는 레거시로 존재하지 않아 소스로부터 구현된 component, 이미 모듈화된 component, 다른 component를 통합하여 인터페이스를 제공하는 component들이 대상이 될 수 있으며, 이들 Backside component는 하나의 Core Facade component를 통해 웹 서비스를 제공한다. Flow Facade는 다수의 Core Facade component간의 흐름, 즉 웹 서비스간의 흐름을 제어 하는 기능을 가지고 있다.

3.3 PDA지원 에이전트 기반 서비스 아키텍처

컴포넌트 기반의 웹 서비스 아키텍처는 웹 서비스의 논리



(그림 6) 웹 서비스 컴포넌트 메타 모델

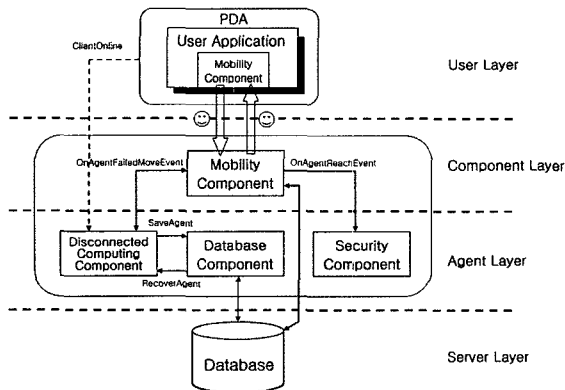


(그림 7) 컴포넌트 기반의 웹 서비스 아키텍처

적 계층구조를 기반으로 구성하여 (그림 7)과 같이 정의한다. 내부의 구조는 논리적 계층 구조와 매핑 된다. **Business Area**는 Business 또는 Presentation service를 제공하는 영역으로써 외부 시스템 또는 사용자가 이 영역에 위치한다. **Service Area**는 웹 서비스를 제공하는 영역으로써, 웹 서비스 기능을 제공하는 Core와 Flow Facade component가 위치한다. **Operation Area**는 실제 비즈니스가 이루어지는 영역으로서 레거시 시스템에서 비즈니스 로직을 제공하는 legacy component인 Backside component가 위치한다. **Support Area**는 Operation Area에서 사용하는 데이터 또는 데이터를 제공하는 컴포넌트가 위치한다. Business Area와 Service Area사이에는 웹 인프라가 위치한다. 시스템은 하나 이상의 어플리케이션으로 구성될 수 있다. 하나의 어플리케이션은 Facade와 Backside component로 구성되며 웹 서비스를 제공한다. 하나의 어플리케이션이 전체 시스템을 구성할 경우 서비스간의 흐름을 제어할 필요가 없으므로 Core Facade component를 통해 웹 서비스를 제공하며, Flow Facade component는 시스템의 내부가 2개 이상의 어플리케이션으로 구성되었을 때, 이들 간의 통합과 흐름을 제어를 제공한다.

3.4 PDA 지원 에이전트 기반 서비스 아키텍처

모바일 에이전트를 통한 서비스를 위해서 모바일 에이전트를 구현하고, 이를 위한 기반 컴퓨팅 환경을 제공하기 위해 모바일 에이전트를 구현한 컴포넌트기반 모바일 에이전

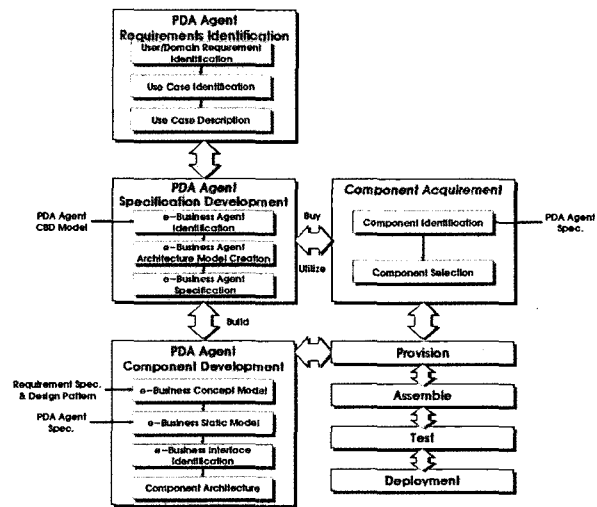


(그림 8) PDA 지원 에이전트 기반 서비스 아키텍처

트 컴퓨팅 아키텍처가 요구된다. 본 논문에서는 모바일 에이전트의 서비스를 위한 환경을 지원하고 개발이 용이하도록 컴포넌트 기반 개발을 고려한 4 계층의 아키텍처를 User Layer, Component Layer, Agent Layer, Server Layer와 같이 정의하였다(그림 8). 이는 사용자 측면과 에이전트의 기능을 구현하기 위한 측면 그리고 사용자 서비스를 지원하기 위한 데이터 접근 측면을 고려하였다. 특히 컴포넌트와 모바일 에이전트를 위해 Component Layer, Agent Layer를 각각 분리하여 에이전트의 기능은 Component Layer에서 구현되며, 에이전트의 관리는 Agent Layer에서 이루어진다.

4. PDA지원 에이전트 컴포넌트 개발 프로세스

PDA 에이전트를 위한 컴포넌트 기반 분석 및 설계는 (그림 9)와 같이 PDA 관점과 agent, component 관점이 PDA 도메인을 위한 프로세스에 포함된다. 사용자와 도메인 요구사항을 기반으로 PDA 에이전트 요구사항을 식별하여, 에이전트 명세와 컴포넌트의 개략 명세를 개발하며, 이를 통해 컴포넌트를 작성하거나, CBD 참조 모델을 기반으로 컴포넌트를 선택하여 조합함으로써 목표로 하는 시스템이나 소프트웨어를 구축한다.



(그림 9) PDA 지원 에이전트 컴포넌트 개발 프로세스

4.1 PDA Agent Requirements Identification

PDA 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 스토리보드 기법을 사용하여 요구사항 시나리오를 작성한다.

작성된 요구사항 시나리오를 기반으로 사용사례를 식별하고, UML의 Use case Diagram을 작성한다. 이는 목표 시스템의 영역을 정의하고 전체 업무 영역 중에 어느 업무 기능을 책임질지 명확히 하는 것이며, 행위자와 기능간의 관계

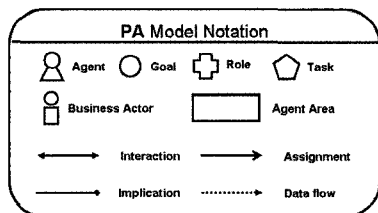
및 부가적인 설명이 된다. 요구사항 식별단계에서는 후보가 될 수 있는 에이전트에 대한 이해와 에이전트를 기반으로 구성되는 기능적인 요소나 시스템 전체의 구성 파악이 주목적이다. 여기서 에이전트는 단일의 에이전트이거나 집합적인 에이전트의 대표일 수 있다.

4.2 PDA Agent Specification Development

식별된 요구사항을 기반으로 개발될 에이전트에 대한 식별을 통해 에이전트 아키텍처 모델과 각각의 에이전트에 대한 명세 작성을 주요 단계로 구성된다. 이 단계를 통해 얻은 부산물은 구성될 컴포넌트의 식별 및 새로운 컴포넌트의 개발에 주요한 자원이 된다.

PDA 에이전트 명세 개발은 컴포넌트 참조 아키텍처를 기반으로 한다. 컴포넌트 참조 아키텍처에서 에이전트의 영역을 식별하고 식별된 참조 아키텍처 부분에 해당하는 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 기반을 둔 에이전트 명세는 기본 명세와 4-3개의 모델을 생성한다. 이는 명세단계에 획득되며, 새로운 컴포넌트를 식별하고 개발하기 위한 주요 자원으로 사용된다. PA(PDA Agent)모델은 더 높은 수준의 표현을 제공하기 위해 UML 메타모델로 확장된다. 본 절에서는 PA 모델을 가시적으로 표현하기 위해서 새로운 구성요소로 작성되는 다이어그램을 기술하며, (그림 10)에 PA 모델 개념과 관련성을 기호와 요약 정보로 제시하였다.



(그림 10) PA 모델 표기법

4.2.1 PA 식별 및 역할 모델 생성

각각의 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 PDA 에이전트의 CBD 참조 모델에 어느 영역에 위치하는지를 식별한다. 또한, 에이전트의 이름을 에이전트의 사용 사례 기술서를 참조하여 명명하며, 이는 에이전트 명세 아키텍처 모델과 스펙 생성에서 목표 에이전트로 정의한다.

4.2.2 PA 목표 모델 생성

본 모델은 에이전트간의 목표, 태스크, 상태와 의존성들을

표현한다. 목표와 태스크는 상태와 함께 연관되며 논리적인 의존성에 의해 연결될 수 있다. 상위의 목표는 하위의 서브 목표의 집합이 달성됨으로써 충족될 수 있으며, 이러한 하위 서브 목표를 식별하고 모델로 제시함으로써 에이전트 시스템을 이해하고 분석 가능하다. 또한, 태스크는 목표를 달성하기 위해 수행되며, activity diagram으로 표현 가능하다.

4.2.3 PA 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 PA 아키텍처 모델 생성

식별된 PDA 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 PA 아키텍처 모델로 제시한다. 에이전트 영역은 두 형태로 나뉘는데, 다른 에이전트와 통신하거나 가용성을 알리기 위한 영역과 사용자와 상호작용하기 위한 사용자 영역으로 나뉜다.

4.2.5 PA-명세 작성

작성된 PA 아키텍처 모델을 기반으로 각각의 에이전트 별로 에이전트의 특징, 공유되는 자원, 접근에 대한 정보, 모델 정보 등을 명세 한다. 이는 에이전트의 기능적, 비기능적 특징을 표현하는 것으로서 기능적인 명세는 UML의 클래스 다이어그램, 시퀀스 다이어그램을 이용하여 각 에이전트 마다의 속성과 행위를 기본적으로 정의한다. 또한, 정의된

<표 1> PA-명세 목록

Item	Description
Agent Name	식별된 PA 이름
e-business Type	PA-CBD 참조모델에서 PDA 에이전트 타입
General Agent type	PA-CBD 참조모델에서 일반적인 에이전트 타입
Identification Code	PA-CBD 참조 메트릭스에서 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

에이전트간의 비동기적으로 전송되는 메시지와 이벤트에 대한 부분을 표현하며, <표 1>은 PA-명세 항목과 해당하는 내용을 나타낸 것이다.

4.3 PDA Agent Component Development

본 논문에서는 기존에 제시되어진 컴포넌트 명세 기법에 새롭게 요구되는 명세 특성들을 포함하여 컴포넌트 명세를 작성한다. 이는 컴포넌트 개발을 위한 상세 명세로서 응용으로의 전개시 조립을 위한 명확한 의미적인 플러깅 지점을 확보하고 비즈니스 프로세스의 계층적 실현을 위해 명세 항목을 정의한다. 명세 항목은 컴포넌트 구현을 위해 명세된 것이며 식별될 수 있는 컴포넌트의 카테고리라 이름, 식별 코드 등을 작성한다. 명세는 컴포넌트 개발과 컴포넌트의 획득을 위해 작성되는 가이드라인의 역할을 한다. 구현단계에서는 Component Diagram 항목에 Invariant와 Variant, Exception으로 인터페이스를 명확히 한다. 또한, Usage Scenario 항목은 카테고리 내의 컴포넌트들의 이용 절차를 예시한 것으로 조립을 위한 확신된 가이드라인으로서 이용한다. Qualities Attribute에는 타입, 언어, 컨테이너, 데이터베이스, 미들웨어 등의 구현적 선택과 성능, 보안, 플랫폼 관점에서 준수해야 하는 요소들이 나열된다.

PDA 개념 모델, PDA 정적 모델, PDA 개념 모델과 PDA 인터페이스 식별과 컴포넌트 명세의 4개의 활동으로 나뉜다. 명세개발은 요구사항 단계의 Use Case 모델과 PA 모델과 명세를 중심으로 컴포넌트를 식별하고 설계하는 전반적인 과정을 포함한다. PA 에이전트에서 분석된 에이전트의 목표, 역할, 상호작용, 아키텍처 모델에서 속성들을 컴포넌트 개발을 위한 속성과 행위들로 매칭 시키기 위해 모델간 매칭을 시키고 인터페이스를 식별해내는 과정이 수반된다. 또한, 컴포넌트간의 의존성을 통해 에이전트를 구성할 수 있

<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 Architecture에 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 컴포넌트 간의 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 명세
Usage Scenario	컴포넌트 사용을 위한 인증된 시나리오

도록 컴포넌트 아키텍처도 정의해야한다.

기존에 제시되어진 컴포넌트 명세 기법에 새롭게 요구되는 명세 특성들을 포함하고 ABCD 아키텍처에 기반 한 특징을 포함하여 컴포넌트 명세를 작성한다. 이는 컴포넌트 개발을 위한 상세 명세로서 전개 시 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 것이지만, 기본 명세 단계에서는 음영으로 된 부분을 작성한다. 이는 식별될 수 있는 컴포넌트의 카테고리라 이름, 식별 코드 등을 작성한다. 명세는 컴포넌트 개발과 컴포넌트의 획득을 위해 작성되는 가이드라인의 역할을 한다. 구현단계에서는 컴포넌트 다이어그램 항목에 Invariant와 Variant, Exception으로 인터페이스를 명확히 한다. 또한, Usage Scenario 항목은 카테고리 내의 컴포넌트들의 이용 절차를 예시한 것으로 조립을 위한 확신된 가이드라인으로서 이용한다.

4.4 Component Acquirement

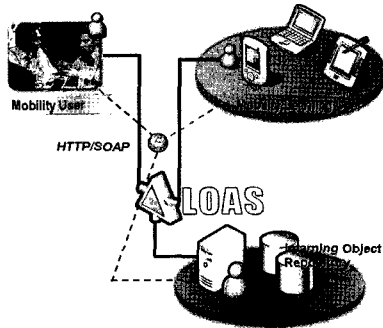
명세된 컴포넌트가 실행 환경에서 동작할 수 있도록 기반 환경에 기능을 포함한 프로그래밍이 요구된다. 적용 단계의 첫 번째인 준비과정에는 명세의 실체화 과정에서 분산 객체 환경에 컴포넌트를 등록하는 과정이 포함된다. 표준 컴포넌트 환경으로는 MS사의 COM+나 Enterprise Java Bean, CORBA 등이 있으며, 컴포넌트의 통합 환경 및 컨테이너의 역할을 하게 된다. 등록된 컴포넌트는 목표 시스템의 일부 혹은 전체로 구성하기 위해 조립을 하게 된다. 조립은 컴포넌트와 기존 소프트웨어와의 결합과 컴포넌트간의 결합을 통해 실행 시스템을 구성하고 사용자 인터페이스를 설계하여 애플리케이션을 만들어내는 과정이다. 또한, 사용자 인터페이스와 식별된 PDA 응용시스템에서의 에이전트 기능을 확인하고 비즈니스 로직을 정의해야 한다. 조립은 프로그래밍 레벨에서 표준 환경에서 제공하는 도구를 통해 쉽게 구축 가능하며, 부분적인 기능 추가나 애플리케이션 형태로 출력된다. 조립을 통해 구성된 애플리케이션은 테스트 후 배포된다.

5. PDA지원 에이전트 컴포넌트 개발

본 논문에서 구현하고자하는 PDA 서비스는 LOAS(Learning Object Access Service)로서 PDA 원격강의를 위해 저장된 학습 객체들의 히스토리관리 및 사용자별 권한 관리를 제공하는 서비스를 구현하고자 한다. 이는 교육영역에서 제출된 강의에 대한 학사관리 및 강사별 콘텐츠를 관리하는 영역뿐만 아니라 소프트웨어의 개발 시에도 소스의 버전관리 및 사용자별 접근 가능한 소스들을 제한을 하는 기능을 제공한다. 이는 모방일 원격강의에서 강의자뿐만 아니라 내부의 사용자들에게 강의 자원의 접근 권한을 할당함으로써, 모바일 원격강의의 자원관리 측면에서 필요한 사항이다.

LOAS에서의 사용자는 내부 사용자와 개발된 웹 서비스를 사용하기 위해 접근하는 외부 사용자 및 다른 웹 서비스

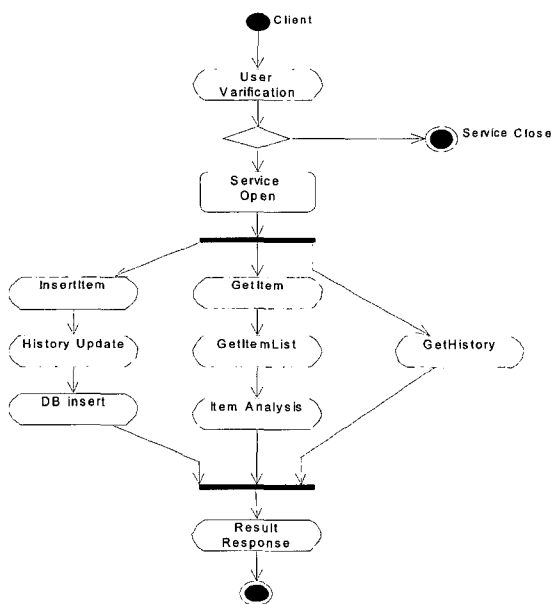
또는 LOAS를 사용한 어플리케이션 등이 비즈니스 영역에 해당한다. 웹 서버는 웹 서비스 정의 및 게시 그리고 XML을 핸들링 할 수 있어야 하며 작업의 흐름을 제어하는 기능을 제공한다. 이는 서비스 계층에 해당하며, 메일링 시스템과 통합 가능한 시스템으로 확장될 경우 Flow Facade 컴포넌트가 위치할 수 있다. 데이터 운영 시스템은 논리적 구조에서의 서비스 계층과 운영 계층을 함께 포함하는 부분이다. 서비스 계층에서의 Core Facade 컴포넌트는 운영 계층의 Backside 컴포넌트를 기반으로 웹 서비스를 제공하기 때문에 이 두 부분은 하나의 시스템에 위치하게 된다. 마지막으로 지원 계층에서는 데이터 운영 시스템이 사용하는 히스토리 정보 repository, 사용자정보 repository, 관리될 자원의 repository 등이 위치한다(그림 11).



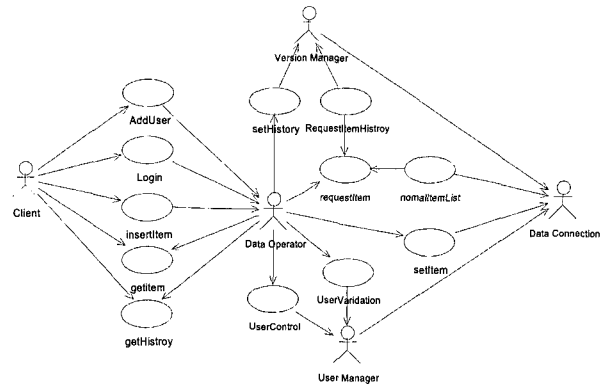
(그림 11) LOAS 웹 서비스의 구조

5.1 LOAS 요구사항 분석

문제 정의 단계에서의 시나리오를 바탕으로 (그림 12)와 같이 Client가 사용하는 기능들의 흐름을 제시하는 서비스 워크플로우 모델을 생성한다.



(그림 12) 서비스 Workflow 모델



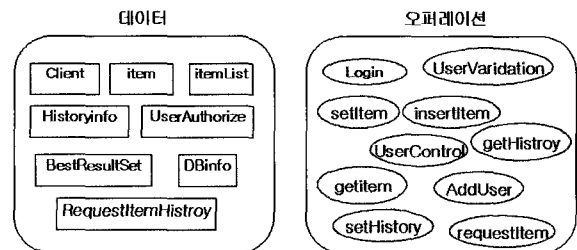
(그림 13) 유즈 케이스 모델

또한, (그림 13)은 제공되는 유즈 케이스 모델로서 서비스 지향 컴포넌트 작업 흐름에서 제공되는 흐름을 중심으로 사용되는 자원과 프로세스를 중심으로 식별된다.

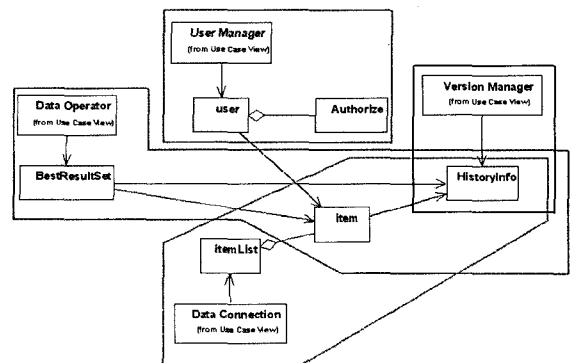
5.2 LOAS 서비스 지향 설계

5.2.1 LOAS 서비스 지향 컴포넌트 식별

LOAS의 유즈 케이스 모델로부터 컴포넌트를 식별하기 위해 각 유즈 케이스들에서 사용되거나 제공하는 기능들을 식별하여 데이터 및 오퍼레이션 요소로 식별한다(그림 14). 시스템을 사용하는 Client도 데이터 요소로 식별을 한다.

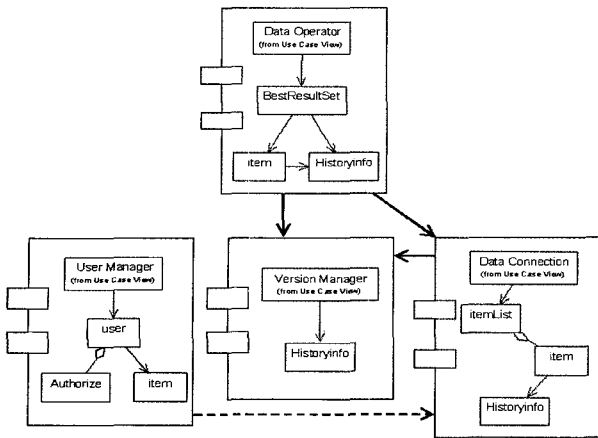


(그림 14) 데이터 및 오퍼레이션 요소 식별

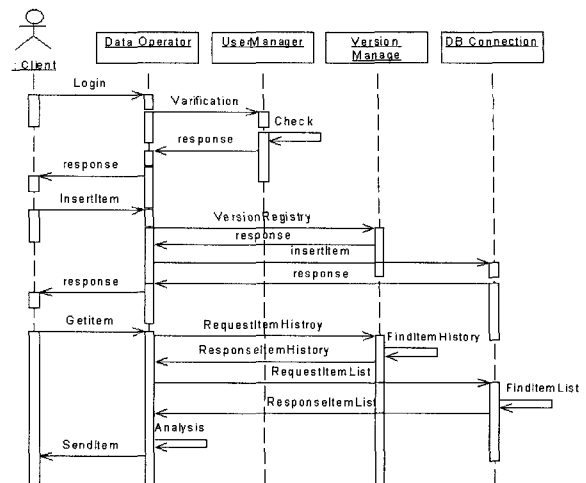


(그림 15) 데이터 요소의 관계 및 그룹

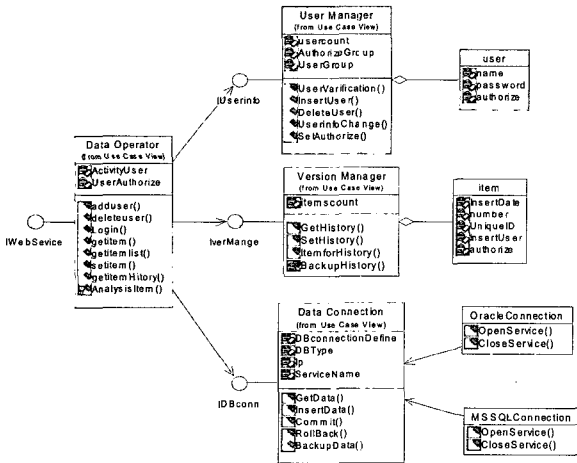
앞에서 식별된 데이터 요소를 데이터들의 관계를 기준으로 핵심 데이터 요소를 식별하고 이를 그룹화하여 (그림 15)와 같이 나타낸다.



(그림 16) 초기 아키텍처



(그림 18) 인터페이스 명세



(그림 17) 서비스 개념 모델

사용자관리 데이터 요소인 UserManager는 User 데이터 요소를 관리하며 알맞은 권한을 획득하여 가질 수 있다. 이처럼 데이터는 서로간의 의존성을 통해 그룹화 한다. 이는 (그림 16)과 같이 초기 아키텍처를 구성하는데 사용된다.

서비스 개념 모델로서 서비스가 제공할 기능 즉, 오퍼레이션 요소와 초기아키텍처를 사용하여 생성된 서비스 개념 모델인 Class Diagram을 (그림 17)에서 생성한다. 데이터 요소인 User는 데이터 클래스로 이는 이를 활용하는 기능인 오퍼레이션 요소들은 UserManager 클래스에서의 메서드로 제공이 되며, 관련된 이들을 UserManager 컴포넌트로 식별할 수 있다.

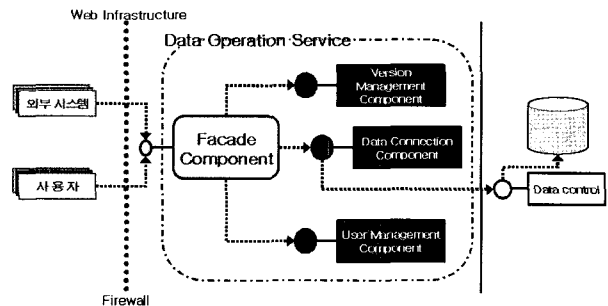
5.2.2 LOAS 서비스 지향 컴포넌트 상호작용 분석

인터페이스를 명세하는 단계로써 인터페이스 명세란, 컴포넌트들 간의 상호작용을 나타낸 것으로, 서비스 지향 컴포넌트 식별 단계에서 식별된 컴포넌트와 내부 클래스의 인터페이스를 식별하고, 서비스 작업 흐름 모델의 서비스가 가지는 기능의 절차를 실제화한다. 이는 (그림 18)과 같이 Sequence Diagram을 사용하여 생성한다. 사용자는 웹 서비

스기능을 제공하는 Facade component의 인터페이스를 통하여 서비스 요청을 한다. Facade component는 일반적인 인터페이스를 사용하여 내부 컴포넌트와 상호 작용을 한다.

5.3 LOAS 서비스 지향 컴포넌트 스펙 작성

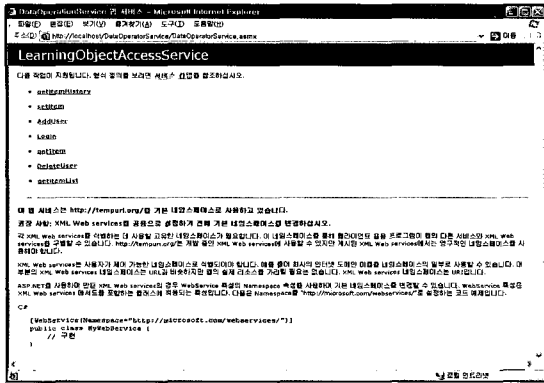
컴포넌트 식별 단계와 상호 작용분석 단계에서 생성된 모델을 정련화하고 컴포넌트의 명세를 작성하는 단계로써, 컴포넌트의 배포 또는 재사용을 위한 정보를 작성한다. (그림 19)는 서비스 지향 설계 단계에서 생성된 모델로부터 정련화된 서비스 지향 아키텍처이다. Facade component는 웹 서비스 컴포넌트로서 이에 대한 명세는 웹 서비스 표준인 WSDL(Web Service Description Language)로 작성되며, 이는 서비스 조립 단계에서 정련하여 완성하게 된다.



(그림 19) 서비스 지향 컴포넌트 아키텍처

5.4 실행 예

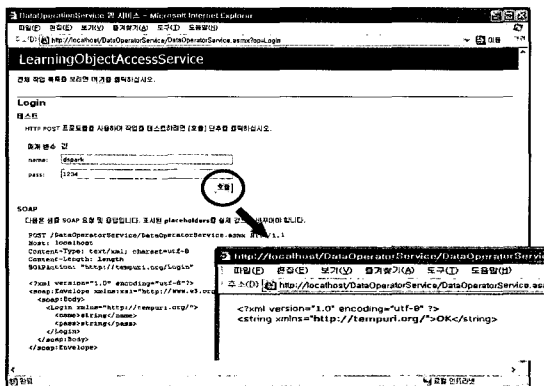
LOAS의 요구사항 분석을 통해 식별된 컴포넌트를 Facade 컴포넌트로 구현하고 기존의 개발된 Backside 컴포넌트를 조합하여 컴포넌트 기반 웹 서비스 아키텍처에 배치시켰다. 이 LOAS의 실행 상태를 웹 브라우저를 통해 확인하며 서비스 요청과 응답메시지를 확인함으로써 웹 서비스의 동작을 보여준다. 웹 서비스는 웹을 기반으로 하기 때문에 구현된 결과를 웹 브라우저를 통해 볼 수 있다.



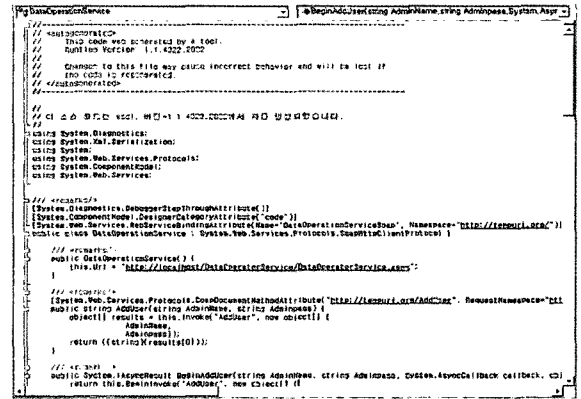
(그림 20) LOAS 실행 예



(그림 23) WSDL로 실행 코드를 생성



(그림 21) LOAS 웹 서비스 호출 테스트와 결과



(그림 24) 생성된 실행코드

(그림 20)은 LOAS 시스템의 실행 예이다. (그림 21)은 웹 서비스의 인터페이스를 통해 서비스 호출 테스트와 서비스 요청에 대한 결과를 나타내는 응답 메시지를 보여준다. 웹 서비스의 사용을 위해 사용자 인증을 하는 화면과 인증 결과를 나타낸다.

5.5 LOAS의 재사용

(그림 22)는 웹 서비스 설계 단계에서 작성된 서비스 지향 컴포넌트 스펙을 정련한 WSDL문서이다. 스펙 단계에서 생성된 컴포넌트 인터페이스 명세인 WSDL에 서비스를 제



(그림 22) LOAS 서비스 지향 컴포넌트 스펙

공하기위에 시스템이 배치된 위치 및 접근방법을 추가하여 정련된 WSDL문서이다.

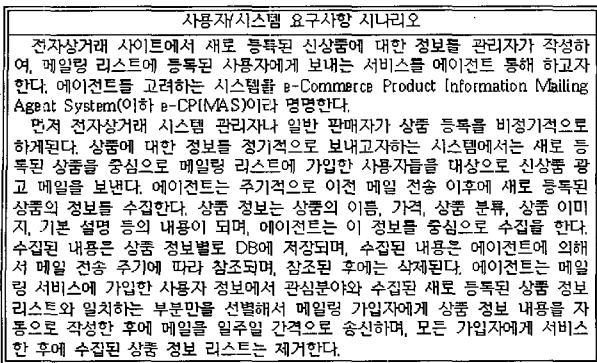
(그림 23)은 WSDL문서로부터 클라이언트가 실행코드를 생성하는 화면이다. 클라이언트는 생성된 클래스 파일을 첨부함으로써 다른 시스템에서 제공하는 웹 서비스 인터페이스를 자기 로컬에 위치하고 있는 것처럼 사용할 수 있다.

(그림 24)는 (그림 23)에서 생성된 실행코드를 에디터를 통해 소스를 열어본 화면이다.

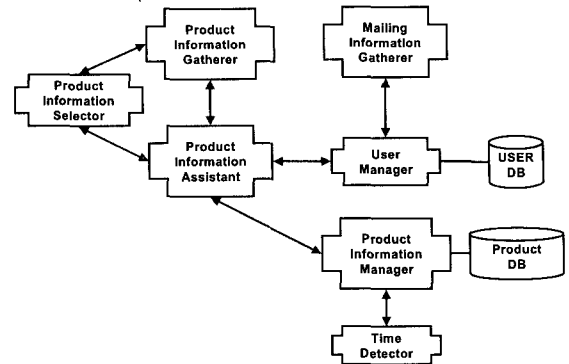
5.5.1 e-CPIMAS의 PA 요구사항 식별

PDA 영역에서 B2B나 B2C(Business To Customer) 형태의 비즈니스 활동이 많이 이루어진다. 고객이나 기업은 상품을 판매하고 일반 사용자들은 구매 활동을 하게 된다. 일반적인 전자상거래 사이트에서 상품에 대한 정보 제공뿐만 아니라 광고를 위해서 메일을 보내는 경우가 빈번하다. 이러한 영역에서 (그림 25)와 같이 시나리오를 기반으로 사용자 및 시스템의 요구사항을 식별하고자 한다.

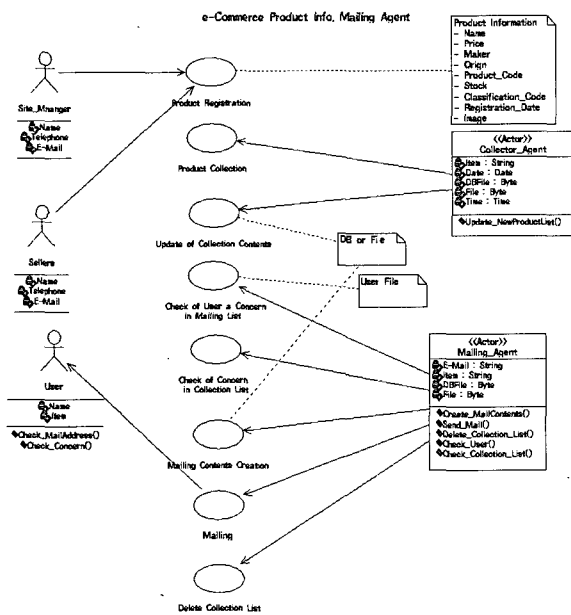
사용자 시나리오를 기반으로 식별된 액터와 유즈 케이스는 (그림 26)과 같다. 액터의 속성과 행위를 초기에 정의해 주었으며, 핸들링 되는 데이터베이스와 관련 정보를 메모하여 이해를 용이하게 하였다. 에이전트로 식별된 것은 액터의 스테레오 타입으로 제시하였다.



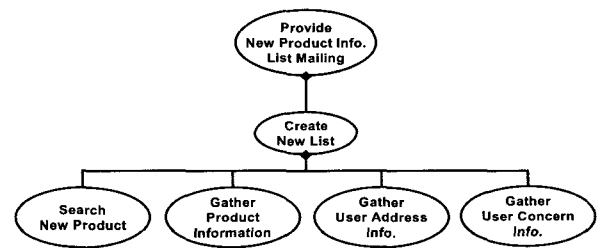
(그림 25) e-CPIMAS 시나리오



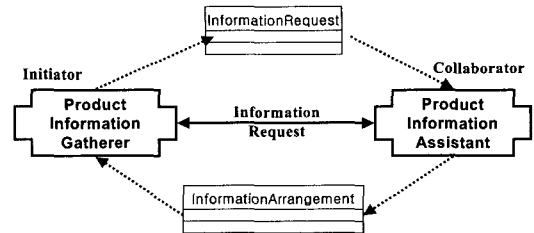
(그림 27) e-CPIMAS 역할 모델



(그림 26) e-CPIMAS 유즈 케이스 다이어그램



(그림 28) e-CPIMAS 목표 모델



(그림 29) e-CPIMAS 상호작용 모델

5.5.2 e-CPIMAS의 PA 명세 개발

PA 식별 및 역할 모델 생성을 위해서 유즈 케이스 다이어그램으로 식별된 에이전트의 기본적인 사용사례를 중심으로 PA-CBD 매트릭스를 통해 소프트웨어 에이전트 형태의 개인 에이전트인 수집 에이전트를 식별하였고, 정보 브로킹 기능을 수행하는 정보에이전트인 e-메일링 에이전트를 식별하였다. 두 개의 에이전트를 중심으로 전체 시스템의 일부로써 에이전트를 운영할 수 있도록 구성한다. 또한, 이 에이전트의 모델 작성을 통해 에이전트 자체의 기능을 식별하고 이를 컴포넌트화한다.

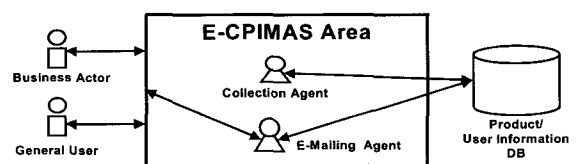
(그림 27)은 메일과 상품 정보를 획득하기 위한 역할 모델이다. 여러 개의 역할들은 다른 역할들과 상호작용하면서 최종 목표를 달성하게 된다.

(그림 28)은 새로 등록된 상품정보를 메일로 보내는 것을 최상의 목표로 이를 해결하기 위해 요구되는 sub-목표들로 구성된 모델을 제시하였다. 목표들은 이전 모델인 역할에 의해서 달성된다.

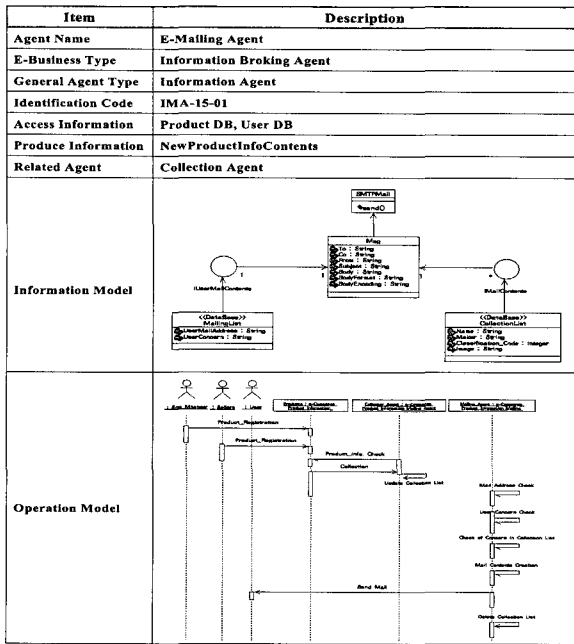
상품 정보를 수집하기 위해 정보를 요청하는 초기 실행을 담당하는 역할과 정보를 제공해 주는 상품 정보 보조자간의 관계를 보여주는 상호작용 모델을 (그림 29)와 같이 표현하였다.

(그림 30)은 아키텍처 모델로 e-CPIMAS의 에이전트 영역에 존재하고 있는 에이전트와 외부 시스템이나 사용자와의 관계를 나타낸 모델링이다. 아키텍처 모델은 에이전트간, 외부시스템간의 관계를 식별함으로써 시스템 일부로 사용될 에이전트의 연관성을 가시적으로 제공한다.

(그림 31)은 작성된 모델을 기반으로 최종적으로 작성된 에이전트 시스템의 명세이다. 전체 e-CPIMAS에서 e-메일링 에이전트의 내용만 명세한 것으로써 일반적인 항목들과



(그림 30) e-CPIMAS 아키텍처 모델



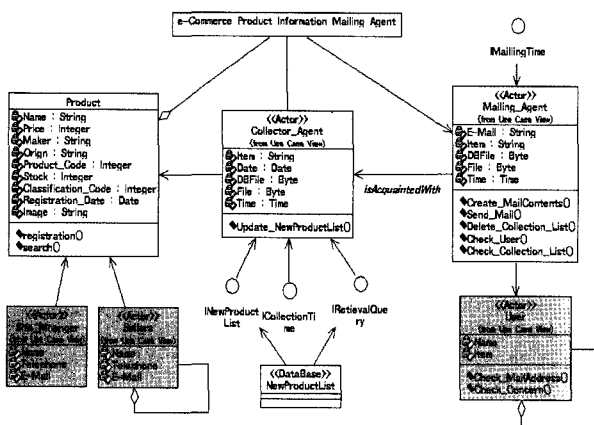
(그림 31) e-CPIMAS에서 e-메일링 에이전트 명세

정보 모델, 행위 모델로 에이전트의 기본 속성과 기능을 제시하였다. 행위 모델은 다른 에이전트와의 관계성도 함께 시퀀스 다이어그램으로 표현하였다.

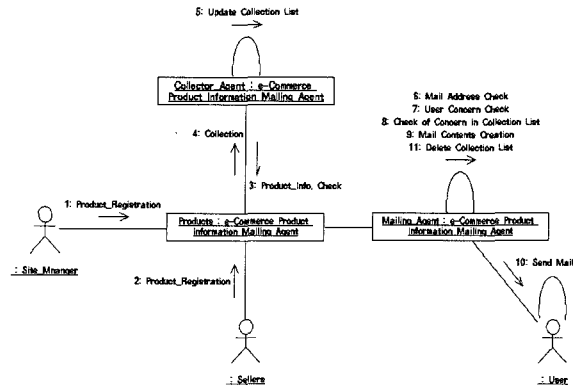
5.5.3 e-CPIMAS의 PA-CBD 명세 개발

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

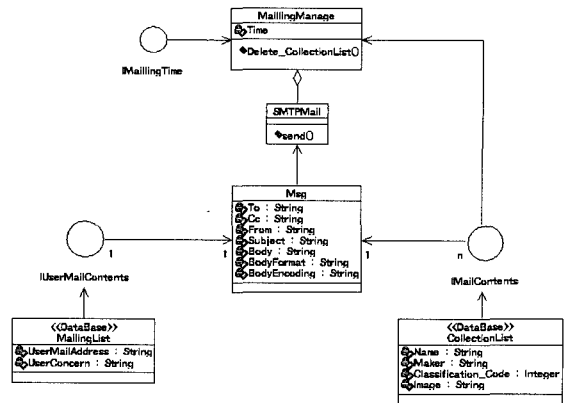
에이전트와 다른 행위자를 중심으로 최소의 클래스를 작성하고 기본적인 인터페이스를 식별 및 제공한 모델이 (그림 32)이다. 또한, 정보에 대한 접근을 위해 데이터베이스도 표현을 하고 있다. 추가적인 메시지의 흐름을 제공하고 인터페이스 식별을 용이하도록 (그림 33)과 같이 협력 다이어그램을 제시하였다.



(그림 32) e-CPIMAS을 위한 정적 모델



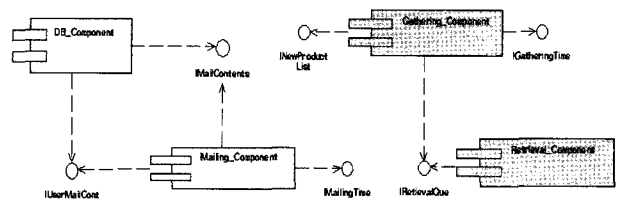
(그림 33) e-CPIMAS 상호작용을 나타낸 협력 다이어그램



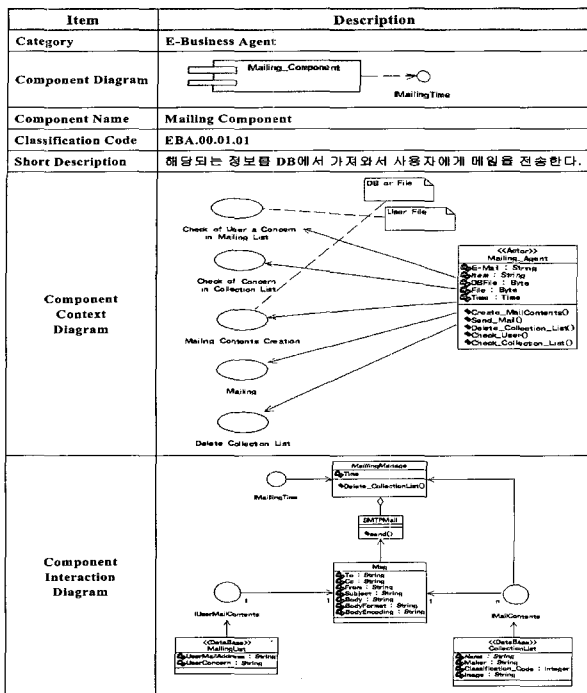
(그림 34) e-CPIMAS에서 e-메일링 에이전트 인터페이스 식별

(그림 34)는 e-CPIMAS의 에이전트 중에서 e-메일링 에이전트를 컴포넌트로 개발하기 위해 잘 정의된 인터페이스를 식별하여 외부로 나타내고, 제공받을 인터페이스와 연관된 클래스를 모델로 작성하였다.

식별된 인터페이스 클래스들을 중심으로 e-CPIMAS를 구성하기 위한 최종적인 컴포넌트를 (그림 35)에 제시한다. 제시된 컴포넌트는 총 4개이며, 수집 에이전트를 위한 Gathering_Component와 Retrieval_Component로 구성되며, e-메일링 에이전트는 DB_Connector와 Mailing_Component 두 개로 구성된다. 특히, 메일링 컴포넌트는 (그림 36)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 35) e-CPIMAS의 전체 컴포넌트 다이어그램



(그림 36) e-CPIMAS에서 메일링 컴포넌트 명세

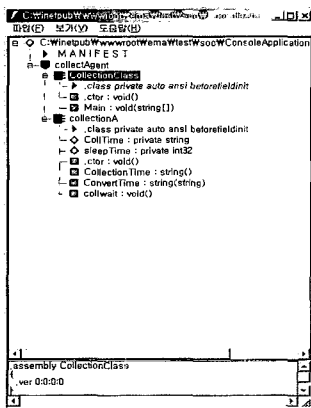
5.5.4 e-CPIMAS의 컴포넌트 게시 및 조립

생성된 e-CPIMAS의 컴포넌트 명세를 실제화하기 위해 본 논문에서는 COM+환경을 기반으로 .Net 프레임워크 상에서 실행 가능한 컴포넌트로 구현하였다.

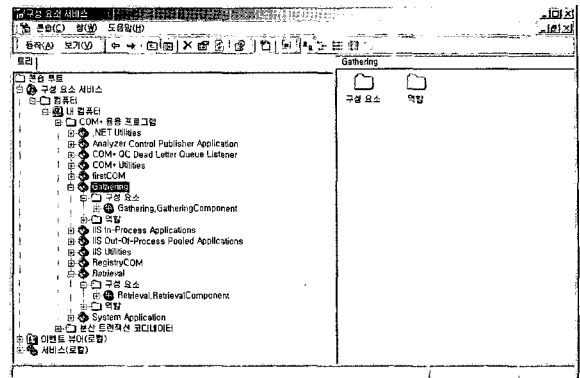
(그림 37)은 수집 에이전트를 구성하기 위해 실제화된 컴포넌트를 제시하였다.

(그림 38)은 실제화된 컴포넌트를 실행 환경에 등록하고 배포하기 위한 선행 과정으로, COM+ 컴포넌트 환경에 수집 컴포넌트를 등록하는 예이다. 또한, 분산 환경에서 공유 가능하도록 게시된 수집 컴포넌트는 (그림 39)와 같다.

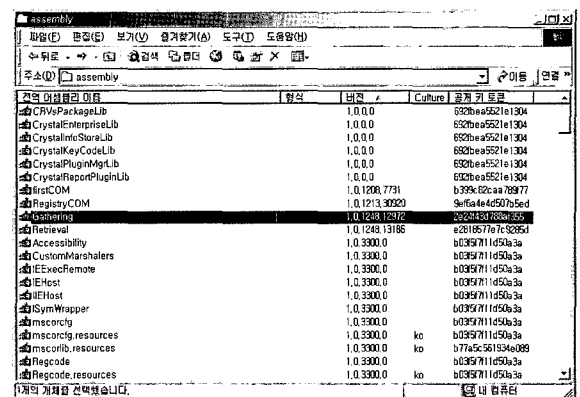
게시된 컴포넌트는 e-CPIMAS의 수집 에이전트를 구성하기 위해서 (그림 40)과 같이 배포된 컴포넌트에서 수집 컴포넌트와 검색 컴포넌트를 선택한 후 (그림 41)와 같이 조립한다.



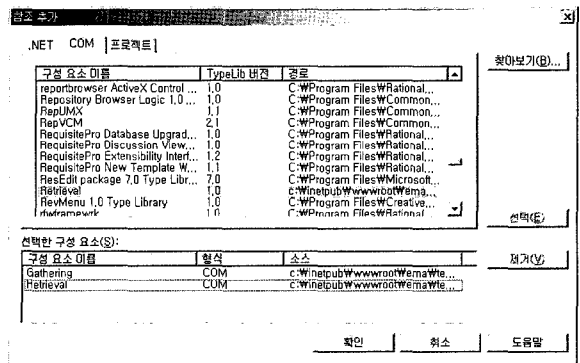
(그림 37) 실제화된 수집 에이전트



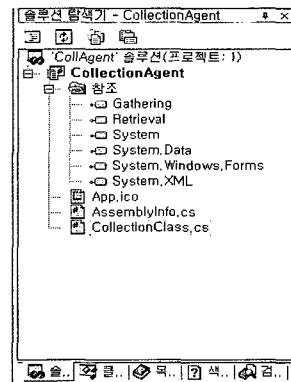
(그림 38) COM+ 컴포넌트 환경에 등록된 수집 컴포넌트



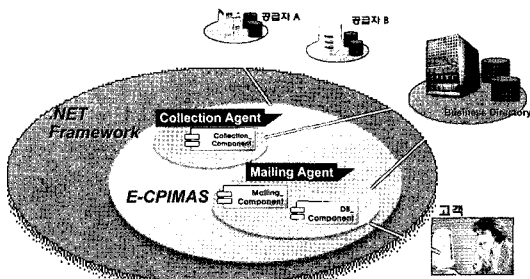
(그림 39) 공유 가능하도록 게시된 수집 컴포넌트



(그림 40) 배포된 컴포넌트 검색



(그림 41) 수집 및 검색 컴포넌트 조립



(그림 42) e-CPIMAS의 운영 환경

5.5.5 e-CPIMAS의 실행

PA-CBD 개발 프로세스에 따라 분석, 설계된 컴포넌트를 실제로 구현 및 배포함으로써 PDA 영역에 컴포넌트 기반 개발을 제시하였다. 배포 및 조립된 e-CPIMAS의 수집 에이전트와 e-메일링 에이전트는 컴포넌트로 구성된 에이전트로서 가시적으로 사용자에게 보이지 않는 내부적인 로직을 가진 실행 가능한 모듈이다. 따라서 본 논문에서는 에이전트의 기능을 하는 컴포넌트의 실행을 임의의 사용자 인터페이스를 사용하여 제시하고자 한다. 개발 환경 및 운영환경은 (그림 42)에서와 같이 .Net 프레임워크이며, C# .Net으로 개발하였다. 컴포넌트는 .Net 프레임워크 상에서 구동하며, 컴포넌트의 주요 기능은 에이전트 기능을 한다.

(그림 45)는 새로 등록된 상품 정보를 사용자에게 전송하기 위한 메일링 에이전트 시작 화면이다. 에이전트가 시작되면 사용자의 메일 정보 및 관심 분야 정보를 획득하고, 판매자나 사이트 관리자는 상품을 등록한다. 등록된 내용은 DB에 저장되고 저장된 내용은 수집 에이전트에 의해 주기적으로 참조되며, 전체 정보 중에서 사용자에게 필요한 정보만 선

택되게 된다. (그림 44)는 수집 에이전트가 주기적으로 새로 등록된 상품 정보를 수집하는 과정을 모니터링한 화면이다.

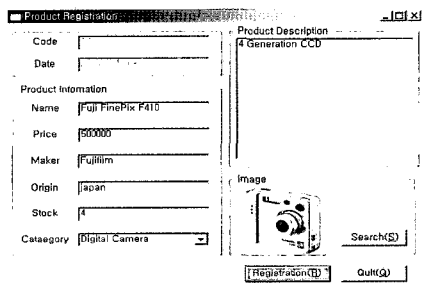
수집 에이전트에 의해서 수집된 상품 리스트에서 관심분야를 매칭 한다. 매칭된 데이터는 일정한 레이아웃에 따라 사용자에게 메일로 보내는 역할을 한다. (그림 46)은 사용자 매칭 과정을 모니터링 하는 것을 나타내었다.

6. 평 가

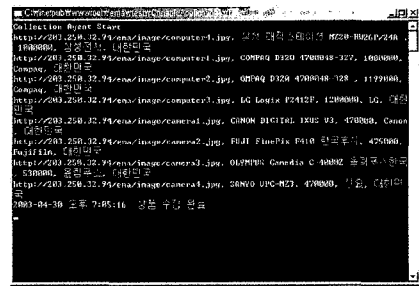
컴포넌트를 기반으로 웹 서비스를 개발을 고려한 개발 프로세스는 없으나, 객체 지향 개발 방법론과 관련연구에서 제시되는 웹 서비스 개발 프로세스를 기준으로 본 논문에서 제시하는 PDA 지원 agent 컴포넌트 기반의 웹 서비스 개발 프로세스를 비교하고자 한다.

각각의 방식으로 LOAS를 구현했을 경우를 정상적 방법으로 평가된 내용을 (그림 47)과 같이 제시한다. 비교 대상으로는 분석, 설계 모델을 제공하는지의 여부와 인터페이스 통합 기능을 가지고 있는지에 대한 여부를 조사한다. 데이터 접속 수는 개발된 시스템이 3-tier로 구현이 되어 있을 경우 서버와 데이터 베이스간의 연결설정이 이루어진 수를 기준으로 평가한다. 구현된 시스템 기능을 다른 시스템에서 사용하고자 할 때 운영체제나 개발 언어의 의존도를 중심으로 파악하였다. 개발 시간은 실제 구현시간을 중심으로 파악하였다.

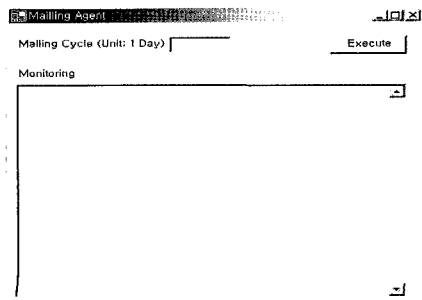
배포 용이성은 다른 시스템에서 LOAS 시스템을 사용하고자 할 때, 얼마나 쉽게 사용 가능한가를 중심으로 평가한다. 웹 서비스를 이용하는 클라이언트를 개발할 경우 WSDL 명세 문서를 통해 서비스의 종류와 서비스 제공 위치를 알 수 있으며 .Net과 같은 개발 툴에서 제공되는 도구를 통해



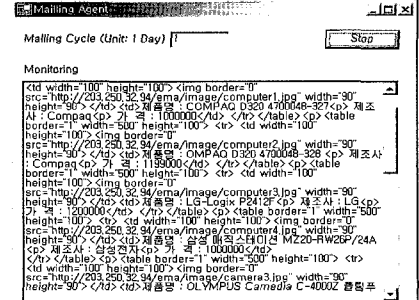
(그림 43) 상품 등록 화면



(그림 44) 수집 에이전트의 상품 정보 수집 화면



(그림 45) 메일링 에이전트의 시작 화면



(그림 46) 메일링 에이전트의 실행 화면

웹 서비스 접근 모듈을 획득하여 사용할 수 있다.

대체성은 새로운 기능이 추가되거나 기능이 변경되었을 경우 웹 서비스를 사용하고 있는 클라이언트에 주는 영향을 통해 대체성을 평가한다. 유지보수성은 기능이 수정되거나 변경되었을 때 얼마나 쉽게 적용할 수 있는가에 대해 평가한다. 재사용성은 개발된 시스템의 재사용이 얼마나 용이한지를 평가한 항목이다. 즉, 얼마나 쉽게 다른 응용 시스템에 적용이 가능한가를 평가하는 항목이다. 배포의 용이성에서도 설명했듯이 웹 서비스는 표준화된 WSDL 명세를 사용함으로써, 다른 시스템에서 재사용이 용이하다. 마지막으로 구현의 용이성은 개발 시간과도 관련이 있으며, 구현시 구현틀 등과 같이 구현하기 얼마나 편리한가를 중심으로 평가한 항목이다. 본 논문에서 제시한 PDA 지원 agent component 기반 웹 서비스 개발에서는 기존 개발된 컴포넌트 재사용을 기본으로 실제 개발되는 부분은 웹 서비스 기능을 가진 특정 컴포넌트만이 개발 대상이 되므로 개발 시간이 적고, 이미 테스트된 컴포넌트를 사용함으로써 안정성을 확보하고, 이에 따른 테스트 시간을 줄일 수 있다.

구분	기존의 기능구현방식	기존의 웹서비스개발	
분석, 설계 모델 제공	사용자 선택	없음	제공
인터페이스 통일 기능	없음	없음	있음
데이터 검색수	N개	1개	1개
플랫폼	의존	독립	독립
개발 시간	보통	높음	빠름
배포 용이성	❌	⊙	●
대체성	❌	△	●
재사용성	△	⊙	●
유지보수성	△	❌	●
구현의 용이성	⊙	❌	●

(비례: ● 아주 좋음 ○ 좋음 △보통 ❌ 나쁨)

(그림 47) 연구개발된 LOAS 시스템 평가

7. 결론 및 향후 연구

IT 환경의 변화에 따라, 기존의 단일 컴퓨터 시스템과 웹을 통한 교육 시스템들이 소형화되고 휴대하기 쉬운 모바일 환경에 적응해야하는 상황이다. 그러나 PDA와 같은 이동 디바이스는 낮은 대역폭과 데이터 교환 시 고비용등으로 모바일 환경에서에서의 서비스가 원활이 되지 않는다.

따라서 본 논문에서는 기능의 모듈성과 독립성이 보장되고 조립 가능한 컴포넌트를 기반으로 동적이고 복잡한 비즈니스 영역에 적용 가능한 에이전트 모델을 작성하고 작성된 모델을 기반으로 하는 PDA 지원 에이전트 컴포넌트 개발을 하였다.

이를 통해, 체계적인 프로세스 도입과 PDA 에이전트의 PA-CBD 참조 모델을 기반을 둬서 분석된 에이전트 영역의 컴포넌트 식별을 쉽게 하고, 개발에 효율성을 제공한다. 또한, 명세를 제공함으로써 적용 가능한 컴포넌트 선택의 가이드라인이 되고, 컴포넌트 생성을 위한 기반 모델로써 재사용 된다.

참고 문헌

- [1] 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.
- [2] Kenha Park, Jintae Kim and Sooyong Park, "Goal based agent-oriented software modeling", Proceeding of the Seventh Asia-Pacific Software Engineering Conference (APSEC'00), pp.320-324, Dec., 2000.
- [3] Desmond F. D'Souza and Alan C. Wills, Objects, Components, and Frameworks with UML, Addison-Wesley, 1998.
- [4] Tan-Hsu Tan and Tsung-Yu Liu, "The MOBILE-Based Interactive Learning Environment (MOBILE) and A Case Study for Assisting Elementary School English Learning," Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT'04), pp.530-534, August 2004.
- [5] Raul Silaghi and Alfred Strohmeier, "Integrating CBSE, SoC, MDA, and AOP in a Software Development Method," Proceeding of Seventh International Enterprise Distributed Object Computing Conference, pp.136-146, Sep., 2003.
- [6] Jian Yang, "Service oriented computing: Web service componentization," Communications of the ACM, Vol.46, No.10, pp.35-40, Oct., 2003.
- [7] OMG, OMG Unified Modeling Language Specification Version 1.5, <http://www.omg.org/uml>, 2003.
- [8] 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.
- [9] 김행곤, 최하정, 한은주, "웹 환경에서의 e-business 컴포넌트에 관한 연구," 한국정보처리학회 소프트웨어공학회지, 제4권, 제1호, pp.60-66, 2001. 3.
- [10] 김행곤, 차정은, 김병준, "아키텍처 기반의 컴포넌트 저장소 개발을 위한 메타 데이터분류 및 프로토타이핑," 한국정보처리학회 산학연 소프트웨어공학기술 학술대회논문집, 제4권, 제2호, 2000. 4.
- [11] 김행곤, 차정은, 김지영, 신호준, "컴포넌트 저장소 형상관리 시스템에 관한 연구," ETRI 최종연구보고서, 2000. 12.
- [12] 김행곤, 최하정, 한은주, "분산 컴포넌트 명세에 기반 한 비즈니스 컴포넌트 구축에 관한 연구," 한국정보과학회 추계 학술대회, 제27권, 제2호, pp.321-324, 2000. 10.



김행곤

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

2001년~2002년 Central Michigan University 교환교수

1990년~2000년 대구효성가톨릭대학교 컴퓨터공학과 부교수

2000년~현재 대구가톨릭대학교 컴퓨터정보통신공학부 교수

관심분야: 컴포넌트기반 소프트웨어공학, 임베디드 소프트웨어 개발 방법론 및 툴, 프로덕트라인 공학