

---

# UML을 이용한 화상 대화 시스템의 설계

장재명\* · 김윤호\*\*

## A Design of Video Conversation System Using the UML

Jae-Myoung Jang\* · Yun-Ho Kim\*\*

### 요 약

현재 소프트웨어 개발의 주요한 패러다임인 객체지향 설계를 많은 시스템들이 채택하고 있으나, 화상 대화를 대상으로 한 연구들에서 본격적인 객체지향 설계의 적용은 그리 많지 않은 실정이다. 따라서 본 논문에서는 화상 대화 시스템에 대한 객체지향 설계를 위한 UML을 이용한 체계적인 아키텍처의 설계를 제시한다. 화상 대화 시스템에 대하여 체계적인 기능/비기능적 요구사항을 포착하고, '4+1 View Model'을 적용하여 객체지향 설계를 함으로써, 컴포넌트들은 재사용성을 담보하며, 이후 필요에 따라 컴포넌트들을 추가하여 시스템을 확장하는 것이 가능하다. 또한, 본 논문에서 설계한 화상 대화 애플리케이션의 컴포넌트들은 다른 여러 화상 대화 시스템에 활용 가능하며, 웹 환경에서의 서비스로도 확장될 것으로 기대된다.

### ABSTRACT

Recently, the object-oriented design is the major paradigm for software development. Most systems are following this paradigm, but the past studies for a video conversation domain were not based on full-scale object-oriented design. Thus, this paper presents a systematical architecture design using UML for a video conversation system that is well-known and has high rate of usefulness. It analysis a video conversation system that has much demand of service as systematical functional/non-functional requirements, and the object-oriented design applying '4+1 View Model' guarantees the reusability of a component and makes it possible to extend a system by adding components as needed. Consequently, it is expected that the components of video conversation system designed by this paper will be useful the other video conversation systems and will be expanded to web environment.

### 키워드

객체지향 설계, UML, 4+1 View Model, 화상 대화

## 1. 서 론

현재 네트워크 기술과 더불어 멀티미디어 기술이 현저히 발달하고 일반 사용자들도 멀티미디어 서비스를 충분히 처리할 수 있는 컴퓨팅 환경을 가

지게 되면서 여러 화상 대화 시스템들이 활발히 개발·사용되고 있다. 화상대화 시스템의 예로서는 화상 회의와 원격 교육 등이 대표적이며, 인스턴트 메신저나 원격 감시 등에서 화상이나 음성을 처리하여 서비스하기도 한다. 미디어 처리 및 전송 기

---

\*안동대학교 컴퓨터공학과 석사과정

접수일자 : 2004. 12. 13

\*\*안동대학교 전자정보산업학부 부교수

술은 앞으로 더 많은 서비스 요구에 의해 다양한 시스템에 적용될 것으로 전망된다. 그러나 아직까지 화상 대화 시스템을 객체지향에 기반하여 체계적으로 분석하고 설계하는 본격적인 연구는 많이 이루어지지 않고 있다. 객체지향 설계를 기반으로 한 미디어 처리와 전송에 대한 연구로서는 김만수 [1]와 김종현[2] 등 소수에 불과하다. 따라서 화상 대화 시스템의 객체지향에 기반한 체계적인 분석 및 설계가 이루어진 연구의 필요성이 제기된다.

본 논문에서는 UML을 이용한 화상 대화 시스템의 설계를 제안한다. 화상 대화 시스템의 체계적인 요구사항 분석과 객체지향 설계를 위해 '4+1 View Model' 프레임워크를 사용하며, Java 프로그래밍 언어로 비즈니스 로직을 구현한다. 객체지향 설계에서 산출되는 컴포넌트들은 재사용성을 담보하며, 이후 필요에 따라 컴포넌트들을 추가하여 시스템을 확장하는 것이 가능하므로, 화상 대화 소프트웨어의 개발 기간 및 비용의 절감 효과를 기대할 수 있다.

## II. 화상 대화 시스템의 객체지향 모델링

객체지향 설계를 위해 제안된 아키텍처 모델들 중에서 '4+1 View Model'[3]은 객체지향 방법으로 개발하는 시스템의 아키텍처를 설계하는 체계적인 관점을 제공한다. 그림 1은 '4+1 View Model'의 구조와 각 뷰의 관계 등을 나타낸다.

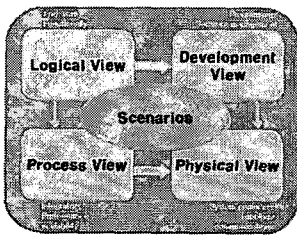


그림 1. '4+1 View Model'  
Fig. 1 '4+1 View Model'.

'4+1 View Model'은 논리 뷰 (Logical View), 처리 뷰 (Process View), 개발 뷰 (Development View), 물리 뷰 (Physical View)등 4개의 뷰로써 아키텍처의 해결영역을 표현하고, 시나리오 (Scenario)는 4개의 뷰를 보완하는 역할을 한다. 4+1 View Model에 기반한 설계의 모든 표기는 UML을 사용하며, 시나리오와 유스케이스 뷰는 같은 의미로 사용된다[4].

본 논문에서는 화상 대화 시스템의 아키텍처를

4+1 View Model에 의거하여 제시한다. 특히, 아키텍처를 분석·설계하기 위해 꼭 필요한 유스케이스 뷰와 논리 뷰, 개발 뷰에 대한 설계 내용을 중점적으로 제시하며, 문서화의 모든 표기는 UML 표기법을 따른다[5]~[8].

### 1. 시스템의 정의 및 범위

체계적인 설계를 위해서는 시스템을 정의하고 시스템의 목적을 결정해야 한다. 따라서 시스템에서 필요한 기능/비기능적 요구사항을 분석하여 시스템을 정의해야 한다. 기능/비기능적 요구사항 분석은 시스템을 정의하고 시스템의 행동 및 성능의 범위를 정할 수 있게 하기 때문에 아키텍처를 설계하는 데 있어 중요한 단계이다. 앞으로의 모든 설계에서 분석된 기능과 비기능이 모두 반영되고, 미디어를 통한 대화를 주목적으로 하여, 결과로 산출되는 시스템을 JVCS (JMF-based Video Conversation System)라 한다.

#### 1) JVCS의 기능적 요구사항

기능적 요구사항 분석은 JVCS가 해야 할 행위를 정의하며, 시스템의 범위 안에서 JVCS의 모든 기능적 요구사항을 식별한다. 모든 기능들은 유스케이스 (UseCase)를 식별하는 기반이 되며, 비기능적 요구사항 분석과 함께 JVCS를 정의한다. JVCS의 기능적 요구사항들을 표 1에서 정리한다.

#### 2) JVCS의 비기능적 요구사항

비기능적 요구사항은 JVCS의 품질을 결정한다. 따라서 JVCS의 비기능적 요구사항을 식별하고 품질속성에 따라 분류한다. 표 2에서 JVCS의 비기능적 요구사항을 정리한다.

### 2. 유스케이스 뷰 (UseCase View)

먼저 JVCS의 행위를 정의하기 위해서 4+1 View Model의 유스케이스 관점에서 시스템의 기능을 분석하여 기술한다[4]. 유스케이스 뷰는 유스케이스 다이어그램과 유스케이스 기술서 등으로 기술한다[8].

#### 1) 액터

액터는 시스템을 수행하는 역할을 나타내며, 유스케이스를 시작하는 역할을 한다. 액터는 시스템을 사용하는 사람이나 외부 시스템 등으로부터 선정된다[5],[6]. JVCS에서 선정된 액터들은 다음과 같다. 말하는 역할을 하고 시스템을 주도적으로 사용하는 역할의 Talker 액터, 말을 듣고 시스템으로부터의 결과를 받는 역할의 Listener 액터, Talker와 Listener의 대화를 중계하고 시스템을 제어하는

표 1. JVCS의 기능적 요구사항 분석  
Table 1. Functional requirements of JVCS

기능	설명
사용자 등록	JVCS를 사용하기 위하여 ID를 등록하고 사용자의 정보를 저장한다.
사용자 정보 변경	JVCS에 등록된 사용자의 정보를 변경하여 저장한다.
로그인	미디어로 대화하기 위하여 등록된 ID로 JVCS에 로그인한다.
대화 요청	상대에게 대화 시작을 요청한다.
대화 수락	상대로부터의 대화 요청을 수락한다.
대화 시작	상대와 미디어로 대화를 시작한다.
대화 종료	대화를 종료하고 로그인 상태를 유지한다.
상대 영상 저장	상대방으로부터 수신되는 영상을 로컬에 미디어 파일로 저장한다.
상대 음성 저장	상대방으로부터 수신되는 음성을 로컬에 미디어 파일로 저장한다.
저장 폴더 지정	미디어를 저장하기 위한 로컬 폴더를 지정한다.
미디어 파일 함께 보기	로컬의 미디어 파일을 상대방에게 전송하여 상대와 같은 미디어 파일을 재생한다.
파일 전송	로컬의 파일을 상대와 주고받는다.
장치 테스트	마이크와 카메라 등의 미디어 장치를 테스트한다.
통신모드 설정	사용 가능한 미디어 장치, 통신 상태 등을 고려하여 통신모드를 설정한다.
자신의 영상 on/off	자신의 영상 송신 여부를 결정한다.
자신의 음성 on/off	자신의 음성 송신 여부를 결정한다.
상대의 영상 on/off	상대 영상의 수신 여부를 결정한다.
상대의 음성 on/off	상대 음성의 수신 여부를 결정한다.
사용자 목록	대화했던 사용자들의 목록을 보여주고 대화 가능한 상대를 표시한다.
상대방 정보	현재 대화중인 상대의 정보를 표시한다.
미디어 재생	저장된 미디어를 포함하여 로컬의 미디어 파일을 재생한다.

다른 시스템이거나 사람의 역할을 하는 Manager 액터이다 (그림 2).

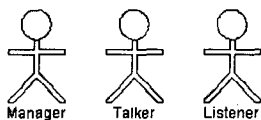


그림 2. 액터 다이어그램  
Fig 2. Actor diagram.

2) 유스케이스

표 2. JVCS의 비기능적 요구사항 분석  
Table 2. Non-functional requirements of JVCS

품질 속성	설명
사용성 (Usability)	사용자의 편리한 요청을 위하여 모든 기능은 버튼과 메뉴로 구성해야 한다.
	GUI에서 주 대화 상대의 영상을 가장 크게 하고, 주 대화 상대와 자신의 영상은 영역을 분리한다. JVCS는 사용자의 영상과 음성으로 대화하며, 텍스트는 포함하지 않는다.
기능성 (Functionality)	JVCS는 화자와 청자를 구분하며, 관리자 시스템이 화자와 청자를 연결해주고 관리한다.
	JVCS는 사용자 정보 및 대화 정보 등을 DB에 저장하고 사용하며, DB는 관리자 시스템에서 관리한다.
	JVCS는 영상의 전송 포맷을 H.263으로 한다. JVCS는 음성의 전송 포맷을 MPEGULTRA로 한다.
신뢰성 (Reliability)	대화가 아닌 다른 기능 수행 중에도 끊어짐이 없도록 사용자의 영상과 음성의 전송 및 처리를 최우선으로 한다.
	JVCS 사용 중에 일어나는 오류에 대해서 사용자에게 알리고 초기화한다.
이식성 (Portability)	JVCS는 현재 엔터프라이즈 기반 서비스나 모바일, 임베디드 기반 서비스는 지원하지 않는다.
	JVCS는 PC를 기반으로 설계되며, 모든 자원은 PC에서 가용한 자원으로 설정한다. 다만 플랫폼에는 종속적이지 않고, 통신망 구성에 대해서도 관여하지 않는다.
	JVCS에서 미디어의 처리와 전송은 JMF와 JMF에서 지원하는 RTP를 사용한다.
	JVCS의 모든 구현은 Java 프로그래밍 언어로 작성한다.
효율성 (Efficiency)	JVCS에서 대화 시작 시, 대화 요청과 응답은 2초 이내에 이루어져야 하며, 대화 종료 후에 JVCS는 로그인 상태로 초기화되어야 한다.
	대화하는 영상과 음성의 최대 전송 지연시간은 1초 이내여야 한다.
	JVCS에서 최대 동시 대화 인원은 8명으로 제한한다. 사용자의 환경에 따라 통신모드가 변경될 수 있어야 한다.
유지보수성 (Maintainability)	JVCS는 화상 대화를 대상 도메인으로 하지만 향후 다른 서비스를 주목적으로 하는 서비스에 확장될 수 있다. 따라서 확장되더라도 성능 저하가 없어야 한다. JVCS의 각 컴포넌트는 필요에 따라 기능과 명세가 동일한 컴포넌트로 교체될 수 있어야 한다.

유스케이스는 액터에게 의미있는 결과를 제공하기 위해 시스템이 수행해야 하는 일련의 활동이며, 시스템의 행동을 정의하기 위해 사용된다[4]~[6]. JVCS의 기능 및 역할을 분석하기 위해서 유스케이스를 식별해야 하며, JVCS의 유스케이스 패키지들

은 다음과 같다. 사용자 정보를 관리하는 사용자 정보 패키지, 미디어 장치의 테스트, 사용자 환경에 따른 통신모드 설정 등을 수행하는 환경 설정 패키지, JVCS에서 핵심적인 기능을 하는 대화 제어 패키지, 그밖에 파일의 송수신을 담당하는 파일 전송 패키지, 미디어를 조정하거나 저장할 수 있는 미디어 제어 패키지가 있다. 그림 3은 분석된 5개의 유스케이스 패키지들을 보인다.

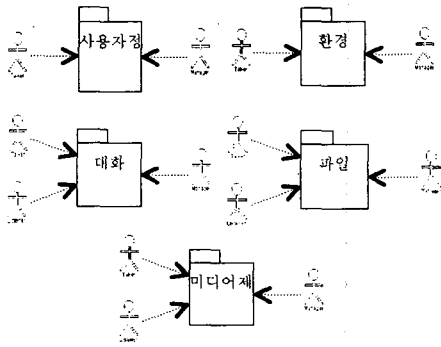


그림 3. JVCS의 총괄적 유스케이스 다이어그램  
Fig 3. High-Level UseCase diagram of JVCS.

그림 3의 패키지들에 대하여 정련된 결과를 그림 4에서 보인다.

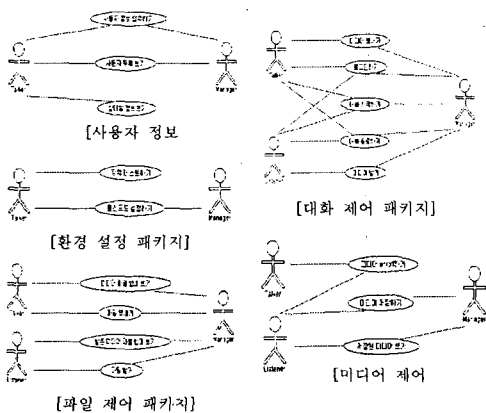


그림 4. JVCS의 세부적 유스케이스 다이어그램  
Fig 4. Detail UseCase diagram of JVCS.

본 연구에서 분석된 유스케이스는 5개의 패키지의 17개 유스케이스이며, 지금까지 분석된 유스케이스의 목록은 간략한 설명과 함께 표 3에 나타내고, 그림 5는 본 연구에서 분석된 JVCS의 유스케이스를 다이어그램으로 나타내었다.

표 3. JVCS의 유스케이스 목록  
Table 3. UseCase list of JVCS

이 름	목 적
사용자 정보 입력하기	Talker가 JVCS에 로그인하기 위해서 사용자의 정보를 DB에 저장한다.
로그인하기	Talker가 JVCS를 사용하기 위해서 대화 준비 상태가 된다. 자신의 상태를 DB에 갱신한다.
대화 시작하기	Talker와 Listener가 화상대화를 시작하기 위해 요청과 수락을 통해 스트림이 연결되어 대화가 가능하게 된다.
대화 종료하기	대화를 종료하고 Talker와 Listener와의 연결을 해제한다.
미디어 보내기	Talker가 대화하고자 하는 형태의 미디어를 Listener에게 보낸다.
미디어 받기	Talker로부터 미디어를 받아서 Listener에게 표현한다.
미디어 저장하기	Talker로부터 받고 있는 미디어를 Listener가 파일로 저장한다.
미디어 파일 함께 보기	Talker가 자신의 미디어 파일을 Listener와 함께 보고 듣기 위하여 미디어 파일을 재생하고 Manager를 통하여 Listener에게 보낸다.
받은 미디어 파일 함께 보기	Talker로부터 전송된 미디어 파일을 재생하여 Talker와 Listener가 같은 미디어 파일을 보고 듣는다.
파일 보내기	Talker가 자신의 파일을 Listener에게 전송한다.
파일 받기	Talker로부터 전송된 파일을 Listener가 자신의 로컬에 저장한다.
장치 테스트하기	자신의 Media Device들을 테스트하여 동작 여부를 확인한다.
통신모드 설정하기	V+A, A only, V only 모드를 설정하여 전송의 효율을 높인다.
미디어 on/off하기	Listener는 Talker로부터 전송되는 미디어를 받지 않을 수 있다.
사용자 목록보기	연결 히스토리와 연결 가능한 사용자의 목록을 보여준다.
상대방 정보 보기	현재 대화중인 상대방의 정보를 보여준다.
저장된 미디어 보기	Listener가 저장한 미디어(파일전송 포함)의 목록을 보여주며, 선택 시 재생한다.

### 3) 유스케이스 기술

분석된 유스케이스들의 행동을 설명하기 위하여 유스케이스 기술서를 작성한다. 유스케이스 기술서에는 유스케이스에 참여하는 참여액터 (Participating Actor)를 기술하고, 유스케이스가 수행되기 전에 반드시 참이어야 하는 사전조건 (Entry Condition)과 유스케이스가 끝난 후 반드시 참이어야 하는 사후조건 (Exit Condition)이 포함된다. 또한, 유스케이스에서 수행되는 액터의 행위와 시스

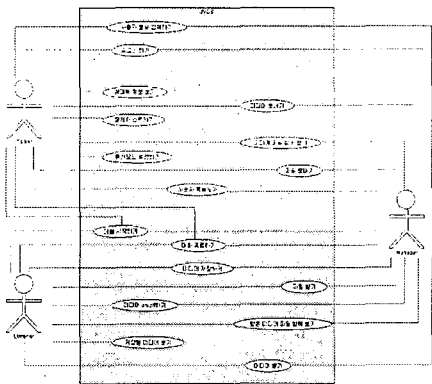


그림 5. JVC의 유스케이스 다이어그램

템 행위를 순차적으로 기술하는 사건 흐름 (Flow of Event) 기술이 주요한 점으로 포함되며, 사건 흐름은 순차 다이어그램 (Sequence diagram)으로 표현된다. 마지막으로 각 유스케이스에서 비기능적인 요구사항과 성능에 관련된 요구사항을 나타내는 품질요구사항 (Quality Requirement)을 기술한다[6],[8]. 본 논문에서는 지면 관계상 식별된 17개의 유스케이스 중에서 하나의 유스케이스에 대한 기술서만을 보인다.

‘미디어 저장하기’ 유스케이스는 Listener에게 표현되고 있는 미디어를 Listener측의 로컬에 미디어 타입에 맞는 파일로 저장한다. 미디어 저장은 Listener의 저장 종료 요청이 있을 때까지 연속적으로 이루어진다. ‘미디어 저장하기’ 유스케이스에

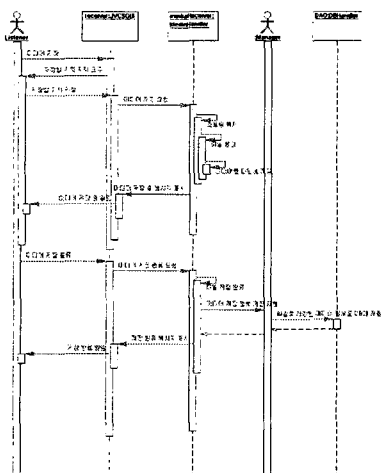


그림 6. ‘미디어 저장하기’의 순차 다이어그램  
Fig 6. Sequence diagram of ‘saving media’ UseCase.

표 4. ‘미디어 저장하기’ 유스케이스의 기술  
Table 4. Description of ‘saving media’ UseCase

참여 액터	Listener, Manager	
사전 조건	<ul style="list-style-type: none"> <li>• Talker와 Listener는 대화중이다.</li> <li>• Listener가 Talker의 미디어를 저장하고자 한다.</li> </ul>	
사후 조건	<ul style="list-style-type: none"> <li>• 지정한 위치에 미디어 파일이 저장되거나,</li> <li>• Listener는 저장장치의 용량이 모자라다는 메시지를 받는다.</li> </ul>	
사건 흐름	<p><b>액터 행위</b></p> <ol style="list-style-type: none"> <li>1. Listener는 해당 미디어 저장 메뉴를 선택한다.</li> <li>3. Listener는 저장될 위치를 입력한다.</li> <li>5. Listener는 미디어 저장 종료 메뉴를 선택한다.</li> <li>7. Manager는 미디어 저장 정보를 DB에 저장할 것을 명령한다.</li> </ol>	<p><b>시스템 행위</b></p> <ol style="list-style-type: none"> <li>2. JVC는 파일이 저장될 위치 입력 다이얼로그를 표시한다.</li> <li>4.1 JVC는 스트림을 복제한다.</li> <li>4.2 JVC는 저장될 위치에 선택된 미디어 타입의 파일을 만든다.</li> <li>4.3 JVC는 복제된 스트림을 파일에 저장한다.</li> <li>4.4 JVC는 미디어 저장 중이라는 메시지를 표시한다.</li> <li>6.1 JVC는 파일과 파일에 저장되는 스트림을 닫는다.</li> <li>6.2 JVC는 Manager에게 미디어 저장 정보 저장을 요청한다.</li> <li>8.1 JVC는 정보를 DB에 저장한다.</li> <li>8.2 JVC는 저장 완료 메시지를 Listener에게 표시한다.</li> </ol>
성능 요구 조건	<ul style="list-style-type: none"> <li>• Listener의 미디어 표현에는 영향을 미치지 않도록 스트림을 복제하여 저장해야 한다.</li> <li>• 저장 중에 파일의 용량과 저장장치의 용량을 체크하여야 한다.</li> <li>• 저장 완료 후 복제된 스트림은 자동으로 삭제되어야 한다.</li> <li>• 시스템이 종료되거나 대화가 종료되면 저장도 완료되어야 한다.</li> </ul>	

서 주요한 행동들은 Manager의 관여 없이 수행되며, 단지 DB에 정보를 저장하기 위하여 Manager와 교신하게 된다. ‘미디어 저장하기’ 유스케이스는 표 4에 상세하게 기술하였으며, 사건 흐름을 두 번의 요청으로 나누어서 그림 6에 순차 다이어그램으로 나타냈다.

3. 논리 뷰 (Logical View)

논리 뷰는 중요한 설계 구성요소들과 이 구성요소들 사이의 관계를 정의하고, 문제영역을 객체와 클래스로 추상화하는 방법과 시스템이 어떻게 기능적 요구사항들을 만족시키는지를 표현한다. 또한 논리 뷰는 시스템을 구성하는 구성요소를 찾아

내고 공통 매커니즘을 보여준다[3],[4]. 본 논문에서는 클래스 정의와 클래스 다이어그램, 데이터 모델로 논리 뷰를 기술한다.

1) 클래스 정의

클래스 정의는 논리적 클래스와 실제적 클래스로 정의된다. 논리적인 클래스는 유스케이스 분석을 통해서 분류되며, 유스케이스 내부의 행동들과 그 행동들의 관계를 분석하기 위해서 개념적인 클래스들을 생성하게 된다. 논리적인 클래스들은 비즈니스 객체 모델 (Business Object Diagram)로 표현되며, 실제적인 클래스의 모델이 된다. JVCES의 유스케이스 분석을 기초로 식별된 논리적인 클래스들은 사용자와 교신하는 사용자 인터페이스와 미디어를 표현하는 플레이어, 시스템 정보와 사용자 정보를 나타내는 데이터 객체, 메시지와 스트림, 데이터를 처리하는 핸들러, Manager와 교신할 수 있도록 해주는 중계자 인터페이스로 구성된다. 논리적인 클래스 정의와 클래스 간의 관계를 표현한 비즈니스 객체 다이어그램은 그림 7과 같다.

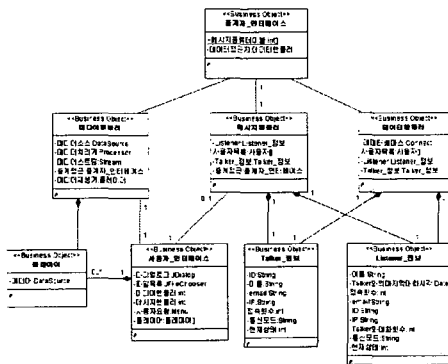


그림 7. JVCES의 비즈니스 객체 다이어그램  
Fig 7. Business object diagram of JVCES.

지금까지 분석된 유스케이스와 비즈니스 객체 모델 등을 통하여 실제 구현 가능한 클래스들을 생성한다. 먼저, 분석된 모델들에서 클래스 및 사용자의 관계에서 공통적인 기능을 하는 클래스들의 패키지를 추출한다. 다음으로, GUI를 구성하며 미디어를 표현하는 Presentation 패키지와 시스템의 전반적인 제어를 담당하고, 사용자 간의 중계를 담당하는 Management 패키지, 사용자의 요청을 처리하고 Presentation 패키지를 제어하는 Control 패키지, 사용자의 데이터를 유지하는 Data 패키지로 시스템을 구성한다. 그림 8은 역할이 구분된 JVCES의 패키지들을 나타내고 있다.

Presentation 패키지는 사용자와의 인터페이스

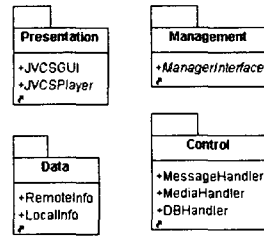


그림 8. JVCES의 클래스 패키지  
Fig 8. Class Packages of JVCES.

를 담당하는 JVCESGUI와 미디어를 재생하는 JVCES-Player로 구성된다. JVCESGUI는 사용자에게 표현되는 모든 GUI를 관리하며, 모든 요청은 메뉴와 버튼을 통하여 이루어진다. 전달된 사용자의 요청을 MessageHandler나 MediaHandler에게 전달하며, Handler로부터 받은 메시지를 사용자에게 표시한다. 표 5는 사용자와의 정보 교환을 위해서 정의된 다이얼로그들이다. 이 다이얼로그들을 통하여 사용자는 시스템의 메시지를 볼 수 있고, 필요한 정보를 입력하게 된다.

Management 패키지는 시스템을 전반적으로 제어하고, 대화를 중계하는 등 Manager 액터와 같은 기능을 한다. Manager는 액터이므로 중계자 인터페이스를 두어 Manager 역할을 하게 한다.

Control 패키지는 사용자의 요청을 Manager에게 전달하고 Manager로부터 행동을 명령받아 Presentation 패키지를 제어하게 된다. 제어는 메시지와 미디어, DB 관련 부분으로 나누어지게 되며, 각각의 제어에 Handler를 만들고 역할을 분담시킨다. JVCES에서는 제어 메시지와 스트림 데이터 전송을 분리한다. 따라서, 제어 메시지는 MessageHandler가 처리하고, 스트림 데이터 전송은 MediaHandler가 전담한다. DBHandler는 DB와 정보를 교환하는 부분을 전담하여 처리한다.

마지막으로 Data 패키지는 Talker와 Listener의 정보를 저장하고 유지하는데 목적이 있다. UserInfo는 사용자의 ID, 이름, email, IP, 접속 횟수, 현재 상태, 통신모드, Talker와의 대화 횟수, Talker와의 마지막 대화시각 등의 정보를 포함한다. UserInfo를 배열이나 벡터 (Vector) 등을 사용하여 사용자 목록을 구성할 수 있으며, Talker의 정보를 유지하는 LocalInfo와 Listener의 정보를 유지하는 RemoteInfo가 UserInfo를 상속한다.

실제적 클래스는 구현에 적용 가능하도록 상세하게 정의해야 한다. 따라서 공통적인 부분들을 추출하여 상위 클래스를 정의하고, 상세 부분들은 개별 클래스로 나누어 정의하여 클래스 간의 계층 구조를 정의한다. 또한 각각의 필요한 기능을 하는

표 5. JVCSGUI에서 사용되는 다이얼로그 목록  
Table 5. Used dialog List in JVCSGUI

이름	설명	종류
상대의 파일 재생	상대방과 함께 재생되는 미디어를 표현하는 다이얼로그	JDialog
자신의 파일 재생	자신의 로컬 파일을 재생해주는 다이얼로그	JDialog
통신모드 설정	미디어 목록을 보여주고 통신모드를 설정하는 다이얼로그	JDialog
장치 테스트	미디어 장치를 테스트하고 결과를 저장하는 다이얼로그	JDialog
상대방 정보	대화중인 상대방의 정보를 보여주는 다이얼로그	JDialog
자신의 정보	자신의 정보를 보여주고 수정할 수 있는 다이얼로그	JDialog
사용자 목록	자신이 대화했던 사용자의 목록과 상태를 보여주고 대화를 시작하게 하는 다이얼로그	JDialog
미디어 저장	미디어가 저장될 파일을 지정하는 파일 다이얼로그	JFileChooser
저장된 미디어 목록	저장된 미디어의 목록을 보여주고 재생할 수 있는 다이얼로그	JDialog
로컬 파일 선택	자신의 파일을 선택할 수 있는 파일 다이얼로그	JFileChooser
수신할 파일 선택	상대가 파일 전송을 요청할 때, 전송할 파일에 대한 정보를 보여주고, 수신할 파일을 선택할 수 있는 다이얼로그	JDialog
파일 전송 상태 정보	전송중인 파일에 대한 전송 상태 정보를 보여주는 다이얼로그	JDialog
메시지 표현	시스템이 사용자에게 메시지를 전달하는데 사용되는 선택 다이얼로그. 확인, 수락, 취소 등을 선택한다.	JOptionPane

헬퍼 클래스 (Helper Class)들을 정의해야 하고, 클래스들 간의 관계를 명확하게 표현하여야 한다. 클래스들 간의 구체적인 관계를 그림 9에서 클래스 다이어그램으로 표현하였다.

### 2) 데이터 모델

데이터 모델은 시스템에서 사용되는 데이터의 구조를 정의하고 실제 사용 가능한 데이터 테이블을 생성한다. JVCS의 데이터 모델은 다음과 같은 테이블들을 정의한다. 사용자 테이블 (User)은 사용자의 기본 정보를 유지 관리하는 테이블이며, 사용자의 ID를 주요키로 하고, 이름, 비밀번호, 전자우편, 접속 횟수 등의 정보를 유지한다. 접속상태 테이블 (State)은 사용자의 현재 상태 정보를 유지하는 테이블로써 사용자 정보 테이블의 ID를 외래키로 사용하며, 현재 통신모드, 접속 상태, 접속 IP 등의 정보를 포함한다.

미디어 저장 정보 테이블 (SaveList)은 사용자가 저장한 상대방의 미디어 목록을 저장하는데 사용

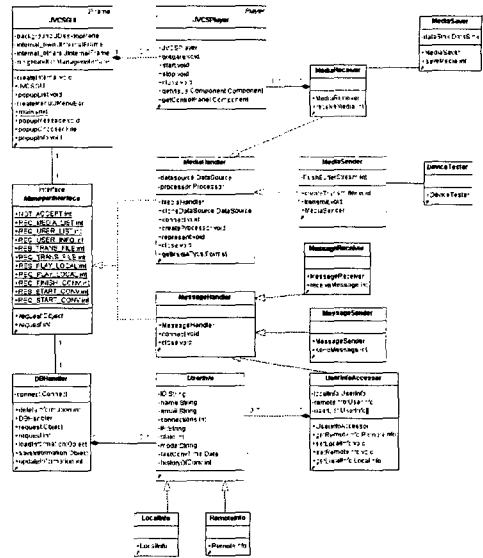


그림 9. JVCS의 구체적 클래스 다이어그램  
Fig 9. Concrete class Diagram of JVCS.

된다. 저장번호를 주키로 하고 사용자 정보 테이블의 ID를 외래키로 사용하며, 저장 일자, 저장된 파일이름, 저장된 위치, 저장한 상대방 ID 등의 정보를 테이블에 저장한다. 대화 내역 테이블 (History)은 사용자가 대화한 상대와 시각을 저장하는 테이블이다. 주요키는 내역 번호이며, User의 ID를 외래키로 사용하고, 대화한 상대방의 ID와 대화 시작과 종료 시각 정보를 저장한다. 그림 10에서 데이터 테이블간의 관계와 상세한 정보를 보인다.

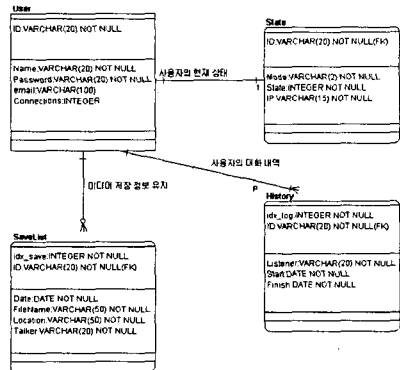


그림 10. JVCS의 데이터 모델  
Fig 10. Data model of JVCS.

### 4. 개발 뷰 (Development View)

개발 뷰는 논리적 관점에서 분석된 구성요소들을 개발 관점 (Development View)에서 모듈화 한

다. 개발 뷰는 모듈들이 모여서 시스템을 구성하는 방식을 설명하며, 모듈을 나누는 원칙을 설명한다. 개발 뷰는 논리 뷰에 나왔던 구성요소들을 모듈로 어떻게 묶어서 어디에 배치할지 다룬다[3],[4]. 본 논문에서는 컴포넌트 다이어그램으로 개발 뷰를 기술한다.

JVCS는 4개의 계층으로 구성된다. 가장 하단에는 시스템에 필요한 정보를 유지·관리하는 Data 계층이 있다. Data 계층 위에 있는 Business 계층은 시스템의 핵심 로직인 미디어와 제어 메시지 처리 컴포넌트들이 위치하게 된다. 그 위에 Management 계층이 있으며 이 계층은 Business 계층, Data 계층, Presentation 계층 사이에서 시스템을 관리하게 된다. Presentation 계층은 애플리케이션과 사용자 인터페이스로 구성된다. 이 계층에서는 사용자의 요청을 받거나 사용자에게 정보를 표시한다. JVCS의 기반 아키텍처를 구성하는 네 개의 계층을 그림 11에 나타내었다.

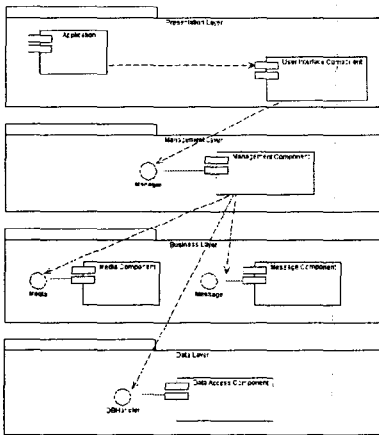


그림 11. JVCS의 컴포넌트 계층도  
Fig 11. Component layer diagram of JVCS.

JVCS는 대부분의 기능이 사용자의 요청에 의해 시작되며, 사용자의 요청은 Presentation 계층의 GUI 메뉴나 버튼을 이용하여 이루어진다. 사용자의 요청은 Presentation 계층을 통해 Management 계층에서 요청의 종류에 따라 알맞은 처리가 이루어지며, 각각의 처리를 Business 계층에서 담당하게 된다. Business 계층에서는 미디어에 관련된 부분과 제어 메시지를 처리하는 부분으로 나뉘게 된다. Data 계층에서는 DB와 관련된 부분을 담당하여 처리한다.

### III. 구현

JVCS 시스템의 구현은 Java 프로그래밍 언어로 구현되었으며, 미디어 전송 및 처리를 위해 JMF (Java Media Framework) [10]를 사용하고, 네트워크 구조는 클라이언트/서버 아키텍처를 이용하였다. 미디어 처리 및 전송을 위해 사용한 JMF는 Java 기반 미디어 프레임워크로서 다양한 플랫폼에 적용이 가능한 미디어 API이며, 비디오 및 오디오의 캡처, 저장, 전송, 스트리밍 등에 이용되는 기술이다. JMF에서 지원하는 RTP (Realtime Transfer Protocol) [11],[12]을 통해서 영상/음성 정보를 실시간으로 전송할 수 있다. 본 논문에서 구현된 JVCS의 전체적인 구성은 그림 12와 같다.

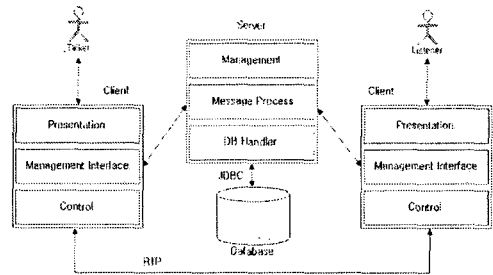


그림 12. JVCS의 시스템 구성도  
Fig 12. System structure of JVCS.

Server는 Manager 역할을 하는 시스템이며, Client는 Server를 통하여 다른 Client와 연결할 수 있고, DB에 접근할 수 있다. 사용자는 Server에 로그인하여 상태를 DB에 저장하며, 다른 사용자와 연결하여 미디어로 대화할 수 있다. 연결과 해제, 정보 저장과 로드는 Server를 통하여 이루어지며, 스트림 연결만 Server가 중계하고 스트림 송수신은 Server가 제어하지 않고 Client 사이에서 RTP로 제어된다.

Presentation뷰는 사용자와의 교신을 담당한다. 사용자로부터 요청을 받고, 미디어를 표현한다. 미디어 표현은 자신의 영상 영역을 구분하고, 특정



그림 13. JVCS의 GUI 설계  
Fig 13. GUI design for JVCS.



사용자의 영역을 크게 하여 표현하며, 크기를 조절할 수 있다. 사용자의 요청은 모두 메뉴로 처리한다. 그림 13에서 JVCS의 GUI를 보인다.

#### IV. 결 론

본 논문에서는 화상 대화 시스템의 기능/비기능적 요구사항을 체계적으로 분석하고, UML을 사용하여 '4+1 View Model'에 기반한 객체지향 설계를 제안하였다. 또한 Java 프로그래밍 언어로 비즈니스 로직을 구현하고, 미디어 처리를 위해 JMF를 이용함으로써 순수 Java 기술로 구현되어 플랫폼 독립성을 확보하였다. 또한, 객체지향적으로 설계되어 컴포넌트들이 확장되어 재사용될 수 있는 장점이 있다. 따라서 본 논문에서 설계한 화상 대화 시스템의 컴포넌트들은 원격 교육 등의 화상 대화 시스템뿐만 아니라 인스턴트 메신저나 원격 감시 등에 적용이 가능하다.

#### 참고문헌

[1] 김만수, 정목동, "CORBA/JMF 기반 오디오/비디오 스트림 시스템의 설계 및 구현," 멀티미디어학회 논문지, 4권 4호, 2001.  
 [2] 김종현, 노영욱, 정기동, "CORBA 기반의 분산 오디오/비디오 스트림 서비스 프레임워크의 설계 및 구현," 정보처리학회논문지A, 제9-A권, 제2호, 2002.  
 [3] Philippe Kruchten, "Architectural Blueprints—The '4+1' View Model of Software Architecture," IEEE Software 12 (6), pp.42-55, 1995.  
 [4] Philippe Kruchten, The Rational Unified Process: An Introduction, 3rd Ed., Addison Wesley, 2003.  
 [5] Object Management Group, Inc., OMG Unified Modeling Language, <http://www.omg.org/technology/documents>, 2003.  
 [6] Martin Fowler, Kendall Scott, UML Distilled, 3rd Ed., Addison Wesley, 2003.  
 [7] Terry Quatrani, Visual Modeling with Ra-

tional Rose 2000 and UML, Addison Wesley, 1999.

[8] Geri Schneider, Jason P.Winters, Applying Use Cases, 2nd Ed., Addison Wesley, 2001.  
 [9] James Rumbaugh, Ivar Jacobson, Grady Booch, The Unified Modeling Language Reference Manual, Addison Wesley, 1999.  
 [10] Sun Microsystems Inc., Java Media Framework API, <http://java.sun.com/products/java-media/jmf>  
 [11] RFC 1889 RTP: A Transport Protocol for Real-Time Applications, 1996.  
 [12] RFC 1890 RTP: RIP Profile for Audio and Video Conferences with Minimal Control, 1996.

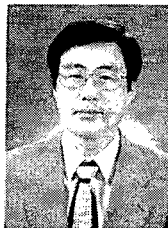
#### 저자소개

##### 장재명(Jae-myung Jang)



2003. 2. 안동대학교 컴퓨터공학과 공학사  
 2005. 2. 안동대학교 대학원 컴퓨터공학과 공학석사  
 2005. 3. - 현재 (주)IDS 근무  
 ※관심분야 : Media Framework, 분산 컴퓨팅, 객체지향 분석/설계/프로그래밍

##### 김윤호(Yun-Ho Kim)



1983. 2. 경북대학교 전자공학과 공학사  
 1993. 2. 경북대학교 컴퓨터공학과 공학석사  
 1997. 2. 경북대학교 컴퓨터공학과 공학박사  
 1997.8.- 현재 안동대학교 전자정보산업학부 부교수  
 ※관심분야 : 인터넷 컴퓨팅, 객체지향 분석/설계/프로그래밍, 분산객체시스템, 병렬처리