

Java를 이용한 VOD용 GUI 설계 경험

김형중* 장순자** 이정수+ 박섭형**

* 강원대학교 제어계측공학과 khj@cc.kangwon.ac.kr

** 강원대학교 제어계측공학과 jsj@pine.kangwon.ac.kr

+ 한국통신 전송기술연구소 jslee@fruits.kotel.co.kr

++ 한국통신 통신망연구소 shpark@timpani.kotel.co.kr

1. 서론

인터넷 TV 또는 인터넷 VOD는 구현상 몇가지 취약점을 안고 있다. 그 가운데 가장 대표적인 약점이라면 브라우징 경험의 부족과 대역폭의 부족을 들 수 있다.

- 브라우징 경험의 부족: 시청자는 일반적으로 방송되는 프로그램을 채널만 바꾸면서 보면 되었지만, 인터넷에 접속하면 원하는 프로그램을 브라우징을 통해 찾아 나서야 한다. 따라서 시청자는 인터넷 환경에 두려움을 느끼지 않을 정도로 훈련되어야 한다.
- 대역폭의 부족: 인터넷은 대역폭의 한계로 멀티미디어 환경에 적합한 수준의 비디오와 오디오를 리얼타임으로 제공하는데 어려움이 있다. 국내의 상황을 보더라도 인트라넷은 100 Mbps 이상의 대역폭을 지닌 곳이 다소 있지만 대부분 캠퍼스나 대기업에 국한된다. 인터넷은 T1급인 1.544 Mbps의 회선을 가지고 있는데가 그리 많지 않다.

그런데 인터넷 TV나 인터넷 VOD와 경쟁하게 될 상품들이 속속 등장하고 있다. 게임 콘솔, 인터넷 셋탑박스, 인터넷 전용 TV, PC

TV, 그리고 네트워크 컴퓨터 NC까지 다양하다. 그래서 인터넷 TV와 인터넷 VOD가 경쟁력을 지니기 위해서는 다음과 같은 추가적인 서비스가 필요하다.

- 사용하기 쉽고 단순하며 염가인 이메일
- 온라인 게임
- 베틀시장 광고, 상가수첩, 세일 안내와 같은 지역정보
- 홈쇼핑, 홈뱅킹, 주식거래
- 뉴스

인터넷 VOD를 구현하기 위해서는 사용자가 인터넷에 대한 거부감을 느끼지 않도록 GUI를 설계하는 것이 무엇보다 중요하다. 또 실제 프로그래밍에서는 플랫폼에 독립인, 하나의 코드로 어떤 컴퓨터에서나 실행이 가능해야 한다. 인터넷이 급속히 확장하면서 인터넷을 지탱해온 IP 프로토콜이 어드레스 배정에 한계가 있게 될 것이라는 전제 아래 새로운 프로토콜 IPng를 도입하려고 한다. 이것은 개인용 휴대폰이나 가정의 냉장고와 같은 가전제품까지 인터넷에 접속하게 될 것이라는 전제가 바탕에 깔려있다. 조속한 시일 내에 인터넷의 대역폭이 획기적으로 증대되기를 기대하는 것은 무리이지만 새로운 기술로 부상하고 있는 인터넷과 VOD 기술을 접목시킴에 있어 우선 GUI의 설계와 인터넷

상에서의 VOD 시뮬레이션은 매우 중요한 의미를 지닌다.

본 논문에서는 DAVIC[1]에서 MHEG[5]과 Java를 표준으로 포함시키려고 하고 있다는 점에 착안하여 GUI를 Java로 구현했다. 물론 DAVIC에서 Java를 고려하고 있는 것은 이것이 플랫폼에 독립적으로 동작할 수 있다는 특성을 지니고 있기 때문이다. 따라서 Java의 호환성은 매우 중요하다. Java의 호환성을 높이기 위해 진행되고 있는 최근의 Java Beans의 동향을 논문의 전반부에서 살피고, 후반부에서는 Java를 이용해 GUI를 설계한 경험과, C로 작성된 MPEG-1 디코더를 Java에 임베드시켜 구현한 예를 설명한다. 그리고 인터넷에서 VOD를 구현하는데 필요한 다양한 통신 프로토콜에 대해 살펴본다.

2. Java Beans와 호환성

인터넷에서 VOD를 구현하기 위해서는 무엇보다도 플랫폼 인디펜던스, 즉 어느 호스트에서나 프로그램이 실행될 수 있는 여건이 중요하다. Java는 크로스 플랫폼 프로그래밍 언어로 널리 쓰이고 있다. HTML 이후 가장 중요한 표준이 Java인데, 이 표준이 선한 회사에 의해 관리되고 있다는 점에서 우려를 표하는 목소리도 있다. 공식적으로 선 마이크로시스템스는 SPARC과 x86을 위한 Solaris 2.x, Microsoft의 Windows 95와 NT, 그리고 Apple의 MacOS만 지원한다고 밝히고 있다[6]. 따라서 PC에서 Windows 95와 NT를 이용한다면 인터넷 VOD를 구현하는데 호환성이 큰 문제가 되지 않는 것으로 보인다.

Java를 바탕으로 선 마이크로시스템스는 Java Beans (오브젝트 아키텍처), JavaOS (오퍼레이팅 시스템) 등을 구축하고 있다. Java API에는 다음과 같은 것들이 포함된다[8].

- Java Core API
- Java Beans API
- Java Media API

- Java Server API
- Java Management API
- Java Commerce API
- Java Enterprise API
- Java Security API
- Java Embedded API

Java Core API는 널리 사용되는 입출력 루틴, AWT (Abstract Windowing Toolkit), 네트워킹 라이브러리, 유틸리티 라이브러리, 애플릿 라이브러리로 구성된다[2,3].

Java Beans는 non-Java 오브젝트 시스템과의 인터опера비빌리티와 포터빌리티 메커니즘을 위한 API 모음이다. Java Beans API는 소프트웨어 컴포넌트를 위한 이식성이 있고 플랫폼에 무관한 API의 모음을 정의하고 있다. Java Beans 컴포넌트는 Microsoft의 OLE, COM, Active-X 아키텍처, OpenDoc, Netscape의 LiveConnect와 같은 기존의 컴포넌트 아키텍처에 플러그 인할 수 있다.

Java Media API는 멀티미디어 프로그래밍을 지원하는 도구로, Adobe Photoshop 또는 Macromedia Director와 같은 인기있는 패키지 처럼 뛰어난 응용프로그램을 개발하려는 프로그래머에게 도움을 줄 것이라고 한다. 금년 12월 JavaSoft는 Java Beans Developer Kit를 발표할 것이라고 하는데 여기에는 다음과 같은 것이 포함된다.

- Beans API를 지원
- Beans의 기능을 시험할 수 있는 테스트 컨테이너 ("beanbox")
- 예제 Beans
- 문서와 튜토리얼

Java Media API는 웹에 존재하는 풍부한 인터랙티브 미디어를 쉽게 그리고 융통성 있게 이용할 수 있도록 도와준다. 그런데 이것은 Java 2D, Java Media Framework, Java Share, Java Animation, Java Telephony, Java 3D로 구성된다.

Java 2D는 추상적인 이미지 모델을 제공하며, 1.0.2 AWT 패키지를 확장하는 것으로 라인 아트, 컬러, 변환, 합성 등을 포함한다.

Java Media Framework는 동기를 맞추기 위해 클럭을, 그리고 오디오, 비디오, MIDI를 연주하기 위해 미디어 플레이어를 보유한다.

Java Share는 응용프로그램을 여러 사용자가 공유할 수 있도록 지원한다.

Java Animation은 2D 오브젝트를 변환하고 움직이게 한다. 동기, 합성, 타이밍을 위해 Java Media Framework를 이용한다.

Java Telephony는 전화와 컴퓨터를 통합하는데 사용된다. 여기에는 전화를 위한 기본적인 서비스들인 통화, 텔레컨퍼런싱, 호출자 식별, DTMF 인코딩/디코딩 등이 포함된다.

Java 3D는 Java 2D와 같이 3D영상을 지원한다.

Java Beans API는 코드와 Java 클래스에 대한 인터페이스를 제공하여 새로운 소프트웨어 환경에 이식할 수 있게 지원한다. 물론 이러한 도구가 이미 완성된 것은 아니고 금년과 내년엔 걸쳐 출시될 것으로 보인다.

API의 집합에서 한 오브젝트가 기술될 수 있는데 각 소프트웨어 컴포넌트가 다른 것과 동적으로 접속되어 새로운 유용한 응용을 생성하게 될 때 COM (component object model)은 그런 시스템을 지칭하는 용어이다. 컴포넌트들은 컨테이너 안에 놓게 된다. 컨테이너도 역시 컴포넌트가 될 수 있다. 많은 종류의 COM이 있지만 대부분은 다음과 같은 기본 특성을 공유한다 [8].

- 컴포넌트 인터페이스 퍼블리싱과 디렉토리 시스템
- 컴포넌트 레이아웃 기능
- 이벤트를 처리하는 능력
- 컴포넌트 오브젝트 지속성 또는 적절한 시간마다 컴포넌트의 정보를 저장하기
- 응용프로그램에 컴포넌트를 링크시키는 것을 지원하기

Java Beans 프레임워크에서 생성된 개별적인 컴포넌트는 단순히 빈스 beans로 알려진다. 빈스는 큰 것도 있고, 작은 것도 있다. 이 빈스는 Java Beans API, Java VM과 표준 라이브러리와 같은 환경이라면 어디에서나 동작한다. 그리고 Java Beans는 IDL, RMI[10]를 이용한다. 그래서 같은 컴퓨터 내에서는 물론이고 다른 컴퓨터에서도 빈스는 다른 빈스에게 데이터를 보내거나 다른 빈스의 메소드를 실행시킬 수 있게 된다.

실제로 COM은 여러 종류가 있다. 예를 들면 Microsoft의 OLE, COM, DCOM은 물론이고 OpenDoc, Netscape의 LiveConnect 등이 있다. 개념은 비슷하지만 구현이 다르다.

Java Enterprise API의 핵심은 JDBC[7], Java IDL[10], Java RMI[10]이다.

Java Embedded API는 전화, 공장 자동화용 컨트롤러, 가전제품, 기타 미래형 휴대용 소형 통신기거나 가전제품 등에 사용될 내장 회로나 시스템을 위한 특별한 Java 환경을 제공한다. 따라서 Java Core API와 완전한 호환성은 보장할 수 없지만 부분적으로 Java 환경을 제공하기 위해 정의된다. 개별 환경에 적합한 네트워킹이나 GUI와 같은 특별한 영역에 대한 확장이 정의될 것이다. 따라서 기본적으로 다음과 같은 부분만 포함될 것으로 보인다.

- java.lang
- java.util
- java.io

Java API는 TCP/IP 소켓을 통해 통신할 수 있는 라이브러리를 제공한다. Java 애플릿 프로그래밍은 상당히 제한적이다. 대부분의 브라우저를 근간으로 삼는 보안관리자가 클라이언트에 파일을 쓰거나 읽는데 많은 제약을 가한다. 클라이언트-서버 환경은 이와 같은 데이터 저장 제약을 극복하는데 도움이 된다[4]. 이러한 클라이언트-서버 환경을 구

축하는대는 스레드, 소켓, Java I/O를 사용한다.

3. Java를 이용한 VOD GUI 설계

인터넷 VOD에서 GUI는 사용자와 셋톱, TV 또는 컴퓨터가 만나는 접점이다. 따라서 사용자가 인터넷 VOD에 대한 거부감이 없도록 GUI가 설계되어야 한다. 또 VOD 이외의 서비스도 제공되어야 한다. 이 논문이 목표로 삼고 있는 것은 아래와 같이 크게 두가지이다.

- 사용자에게 친숙한 GUI 설계
- GUI를 통해 VOD에서 발생할 수 있는 상황을 시뮬레이션하기

사용자에게 친숙한 GUI란 매우 모호한 기준으로 주관적이기 때문에, 흔히 볼 수 있는 것처럼 영화의 타이틀, 장르, 주연배우나 조연배우, 감독, 제작사, 제작연도, 수상경력, 주제가 제목 등 다양한 메뉴를 제공하고 있다. 그러나 이러한 메뉴는 사용자가 특별히 자신의 기준에 의해 선택하기를 원할 때 이용하도록 한다. 따라서 인터넷 VOD는 간단히 인터넷 TV처럼 컴퓨터에 접속하면 보통 TV를 켜올 때와 같이 일단 어떤 종류의 영화가 방영되도록 설계한다.

그리고 시청자는 채널을 통해 영화를 선택하는데 익숙해져 있기 때문에 마우스를 클릭해서 채널을 선택할 수 있도록 한다. GUI 설계의 기본 개념은 기존의 TV에 익숙해진 카우치 포테이토 couch potato를 염두에 둔다는 것이다. 여기서 사용자가 특별히 VOD의 기능을 원할 때 VOD 모드로 진입하도록 시스템을 설계했다.

시청자가 VOD 모드에서 프로그램을 선택할 경우 LI 게이트웨이가 기본 메뉴와 정보를 제공한다. 만일 채널의 대역폭이 충분하지 않을 경우, 영화의 질이 떨어지더라도 영화를 볼 것이지를 묻는다. 이 과정에서 서

버와 시청자 사이에 QOS에 대한 협상이 이루어진다. 그리고 사전 예약을 받아 정해진 시간에 서비스를 제공하기 위한 예약 메뉴를 제공한다. 이때 서버와 네트워크 사이에 자원 예약이 필요하다. 그러나 RSVP와 같은 자원예약 프로토콜이 구현되어 있지 않기 때문에 개념적으로만 이 메뉴를 이용한다. 또 정확한 시간을 제공하기 위해 NTP를 이용한다. NTP는 두가지 면에서 중요하다. 하나는 컴퓨터 자체의 클럭을 믿을 수 없는 경우이다. 수명이 정해진 세어웨어가 수명을 다했을 때 이를 살리기 위해 날짜나 시간을 변경하는 경우처럼 인위적인 상황에서 시간이 틀리게 하더라도 시청자에게 약속된 시간에 영화를 제공하기 위해 NTP가 필요하다. 또 요금계산에서도 정확한 시간정보가 필요하기 때문에 시청자와의 분쟁을 피하기 위해 NTP가 필요하다. 둘째는 MPEG을 구현할 때, 오디오와 비디오를 정확히 동기시키기 위해서는 밀리세컨드 정도의 정확도를 갖는 시계가 필요하다. 이때 NTP는 원하는 정도의 타이밍 정보를 제공한다.

VOD에서 MPEG을 구현하는데 소프트웨어 디코더를 이용했다. Java로 쉽게 디코더를 구현하기 위해 인터넷에서 C 소스 코드를 구한 다음 이것을 Java에 임베드시켰다. 소프트웨어 MPEG 디코더 대신 하드웨어 디코더를 사용할 수 있지만 추가 경비가 부담이 될 수 있다.

그런데 MPEG은 정보전송량이 엄청나기 때문에 리얼타임 구현에 특별히 신경을 써야 한다. 현재의 인터넷은 신뢰성이 높고 효율적인 비디오 전송을 지원하지 못한다. 간단히 말해 TCP는 통신량을 과다하게 유발하며, UDP는 신뢰성이 낮다. 따라서 인터넷에서는 QOS의 보장을 기대할 수 없다. 그래서 IETF를 중심으로 다양한 제안을 검토하고 있고 그 가운데 하나가 RTP이며 또 다른 하나가 IP 멀티캐스트를 이용하는 것으로 RFC 1112가 이에 관한 규정을 정의하고 있다[11]. 현재 MBone이 멀티캐스트를 이용하고 있다.

멀티캐스팅은 서버측에서 동일한 영상을 뿌릴 때 대역폭을 줄이기 위해 사용될 수 있다. 따라서 보통은 소프트웨어 MPEG 인코더들이 UDP를 사용한다. 서버측에서는 소프트웨어 디코더 StreamWorks와 같은 것이 멀티캐스팅을 할 수 있지만 UDP를 쓰기도 한다. 그런데 인터넷이 QOS를 보증하지 못하기 때문에 이에 대한 충분한 연구가 필요하다.

서버 소프트웨어는 사용자의 직접적인 요구에 따라 CGI를 통한 HTML 요구에 의해 비디오 스트림을 보낸다. 서버는 보통 스파스 MPEG 스트림 (56-128 kbps 비디오와 8.5 kbps 오디오 스트림)을 보낸다. 보통 인터넷 서버는 StreamWorks를 이용하고 있다. 본 논문에서는 자체적으로 MPEG 디코더를 Java에서 구현했다.

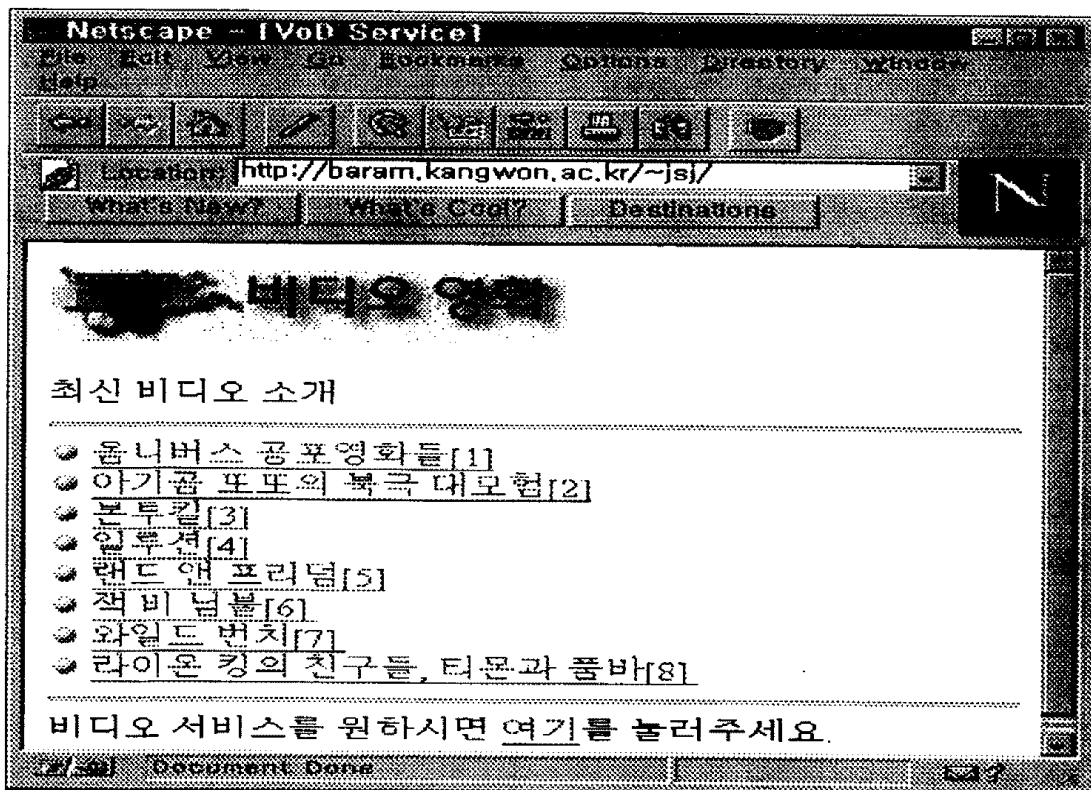
본 논문에서는 VOD 환경을 Sun SPARC에서 구현했고 오퍼레이팅 시스템은 Solaris 2.4이다. 영화목록과 같은 기본적인 메뉴는 클라이언트측이 보유하고 있지만, 영화에 대

한 설명과 같은 정보는 서버에서 오브젝트로 만들어 제공할 수 있게 했다. 서버는 시청자의 요구를 분석해서 인기도를 분석할 수 있게 했다. 그림 1은 구현한 클라이언트 화면 하나를 캡처한 것이다.

4. 향후 과제

DAVIC에서 CORBA[9]를 표준으로 제안했기 때문에 본 연구팀은 현재 Java와 Orbix를 연계시키는 방법과 Java만을 이용해서 VOD 환경을 구현하는 방법을 병행하고 있다. Java Beans API가 확정되면 이것을 이용해 VOD 환경을 구현하는 방법도 시도할 수 있다. 특히 MPEG 디코더에서 VCR 기능을 구현할 수 있도록 지원하는 연구가 가장 큰 과제라고 할 수 있다.

무엇보다 중요한 것은 리얼타임과 QOS의 보장인데 이 분야는 아직 만족할만한 성과가



[그림 1] 서버에서의 VoD 클라이언트 화면

없다. 앞으로 이 분야의 표준이 정해지면 그 결과를 수용하는 연구가 필요하다.

약어표

후사

이 연구는 한국통신 장기기초연구의 일환으로 이루어졌음.

참고문헌

- [1]. Digital Audio-Visual Council, DAVIC 1.0 Specification, December 1995.
- [2]. J. Gosling, F. Yellin, *et al.*, *The Java Application Programming Interface, Vol.1: Core Packages*, Addison -Wesley, 1996.
- [3]. J. Gosling, F. Yellin, *et al.*, *The Java Application Programming Interface, Vol.2: Window Toolkit and Applets*, Addison -Wesley, 1996.
- [4]. R. Kadel, "Generic client-server classes," *JavaWorld*, vol. 1, no. 7, September 1996.
- [5]. T. Meyer-Boudnik, W. Effelsberg, "MHEG explained," *IEEE Multimedia*, vol. 2, no. 1, pp. 26-38, 1995.
- [6]. J. Mitchell, "Is Java really cross-platform ?," *JavaWorld*, vol. 1, no. 7, September 1996.
- [7]. R. Shah, "Integrating databases with Java via JDBC," *Java World*, vol. 1, no. 3, May 1996.
- [8]. R. Shah, "A portable hill of beans," *Java World*, vol. 1, no. 7, September 1996.
- [9]. J. Siegel, *CORBA: Fundamentals and Programming*, John Wiley and Sons, 1996.
- [10]. B. Sommers, "Distributing Java: Remote objects for Java," *JavaWorld*, vol. 1, no. 4, June 1996.
- [11]. A. S. Tanenbaum, *Computer Networks*, 3rd. ed., Prentice-Hall, 1996.

API	Application Programming Interface
AWT	Abstract Windowing Toolkit
CGI	Common Gateway Interface
COM	Componet Object Model
DAVIC	Digital Audio-Visual Council
DTMF	Dual Tone Multi-Frequency
GUI	Graphic User Interface
HTML	Hypertext Markup Language
IDL	Interface Description Language
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPng	Internet Protocol next generation
JDBC	Java Database Connectivity
JDK	Java Development Kit
MHEG	Multimedia Hypermedia Expert Group
MPEG	Motion Picture Experts Group
MIDI	Musical Instrument Digital Interface
NTP	Network Time Protocol
OLE	Object Linking and Embedding
ORB	Object Request Broker
QOS	Quality of Service
RFC	Request for Comment
RMI	Remote Method Invocation
ROJ	Remote Object for Java
RPC	Remote Procedure Call
RSVP	Resource Reservation Protocol
RTP	Real-time Transport Protocol
TCP	Transport Control Protocol
UDP	User Datagram Protocol
VM	Virtual Machine
VOD	Video on Demand