

동영상 파일의 자막 기능 설계 및 구현

신 승 봉[†] · 홍 동 권^{††} · 황 인 재^{†††}

요 약

자막 기능은 청각 장애자를 위하여 외국에서 지상파 TV에서 처음 사용하기 시작하였으며 현재 국내에서도 방송을 하고 있다. 방송의 자막 기능은 청각 장애자에게만 도움을 주는 것이 아니라 해당 국가의 언어를 배우려는 모든 사람들에게 좋은 교육용 도구가 된다. 본 논문은 먼저 자막 기능을 제공하지 않는 인터넷상의 동영상 파일에 자막 기능을 제공하는 방법의 설계 및 구현 방법을 설명하고 기존의 동영상 재생 시스템이 제공하지 않는 반복 선택 및 집단 선택 등 어학 교육에 필요한 몇 가지 기능의 추가 방법을 소개한다. 구현된 시범 시스템은 현재 운용을 하면서 계속적인 자료의 추가와 시스템의 확장을 하고 있다.

Design and Implementation of Closed Captioned Video Files

Seung-Bong Shin[†] · Dong-Kweon Hong^{††} · In-Jae Hwang^{†††}

ABSTRACT

Foreign countries have been used closed captioning for people with hearing loss. Recently some TV programs in Korea also have started to use it. Closed captioned films and video tapes of foreign countries are especially useful for people who want to learn foreign languages as an effective educational tool. In this paper, we propose a method of implementing captioning functionality to an existing streaming software. In addition we added repetitive and group selection of video segments for the new interface of our prototype system. We keep adding additional data and extending functions of our prototype system.

1. 서 론

시청각을 이용한 교육의 효율성은 시각이나 청각 중 한 가지만을 이용하는 것보다 매우 효과적이므로 많은 분야에서 시청각을 이용한 교육 방법을 사용하고 있다. 지금까지의 동영상을 사용한 교육 시스템의 대부분은 제한된 공간에서 비디오 테이프를 사용하는 방법을 사용하고 있었다. 하지만 최근에는 공간에 제한을 받지 않는 인터넷의 장점을 최대한 살린 다양한 인터넷

교육 시스템들이 개발되어 여러 분야의 교육에 활용되고 있다. 인터넷에서의 동영상 재생 방법은 선택한 동영상을 전부 사용자에게 전송하는 방법을 사용하는 것이 아니라 부분적으로 전송된 데이터를 사용하여 동영상의 앞부분을 보여 주면서 다음 부분을 전송 받는 스트리밍 기법을 사용한다. 현재 널리 사용되는 상용 스트리밍 소프트웨어로는 Microsoft사의 넷쇼, Zing사의 스트림웬 등이 있다. 이들 소프트웨어에는 아직 오디오 동기를 이루는 자막 기능을 제공하는 것은 없다. 자막 기능은 원래 청각 장애자를 위해서 공중파 TV를 위해서 처음 사용되었다. 하지만 최근에는 인터넷을 통한 동영상의 활용이 증가하면서 인터넷상에서 활용

† 정 회 원 : 계명대학교 대학원 전자계산학과

†† 정 회 원 : 계명대학교 컴퓨터·전자공학부 교수

††† 종신회원 : 충북대학교 컴퓨터교육과 교수

논문접수 : 1999년 3월 8일, 심사완료 : 1999년 10월 27일

되는 동영상 파일에도 자막 기능이 필요하다는 점을 인식하고 청각 장애자를 위한 여러 가지 노력이 진행 중에 있다. 특히 외국어 자막 기능은 청각 장애자뿐만 아니라 외국어 교육용으로도 활용할 수 있다. 본 논문에서는 공간 제한이 없는 인터넷 환경에서 동영상 파일에 클로즈드 캡션을 지원하는 방법을 설계한다. 그리고 설계된 방법의 타당성을 검토하기 위하여 영어 교육용 동영상 데이터베이스 데모 시스템을 구현하고, 많은 인터넷 사용자들을 위한 실시간 영어 교육 시스템을 실제로 활용하면서 그 문제점을 보완하고자 한다.

본 논문의 구성은 모두 5절로 되어 있으며 다음과 같다. 먼저 2절에서는 비디오 스트리밍 관련 기술, 캡션에 대한 정의와 표준에 대해 소개가 된다. 3절에서는 본 논문에서 제시하는 해결 방안에 대한 시스템 설계 및 구현을 언급하고 4절에서는 실제 데모 시스템 구현 후 시스템의 평가내용을 보여준다. 마지막으로 5절에서는 본 논문의 결론과 향후 연구 방향을 제시한다.

2. 관련 연구

2.1 스트림 방식

스트림 방식 서비스는 데이터를 저장 장소에 저장하지 않고 데이터 스트림을 이용하는 방법으로 다운로드 방식에 비해 데이터를 받는 즉시 재생할 수 있는 장점이 있다. 스트림 방식 서비스를 위해서는 데이터를 네트워크를 통해 재생할 수 있는 미디어 플레이어 필요하다. 스트림 방식 서비스에는 현재 개발되고 있는 것으로 외부 미디어 플레이어 방식과 플러그인 방식이 있다. 인터넷 통신을 위해 널리 사용되는 HTTP(HyperText Transfer Protocol)는 데이터 양이 큰 오디오나 비디오 검색을 위해서는 긴 지연 응답시간 때문에 사용되지 않는다. 이러한 이유로 대부분의 실시간 멀티미디어 웹 응용들은 (RealAudio, VDOLive, StreamWorks ...) HTTP 대신 UDP 기반의 프로토콜을 사용한다. 이러한 스트리밍 프로토콜의 목적은 가능한 한 클라이언트 쪽의 출력을 빨리 하는데 있다.

2.2 자막 처리 기법 캡션

캡션이란 TV 방송의 주사선에 알파벳 기호를 인코딩하고 재생시 필요에 따라 이를 디코딩 하여 TV 화면에 보이게 하는 메카니즘이다[15]. 캡션은 원래 청각

장애자들을 위해 배우의 대사 및 충돌음과 같은 효과음을 자막으로 표시한 것이다.

● 클로즈드(Closed) 캡션

비디오 신호 안에 숨겨져 있어 특별한 풀이장치(decoder) 없이는 볼 수가 없다. 텔레비전 화면에 투사되는 525개의 텔레비전 방송신호선(line)중 제21번째 선에 시간·주파수 정보 코드(closed caption)를 삽입하고 수신기에서 이것을 풀이(decode)하여 캡션으로 사용한다[8, 15]. 주로 대화나 동작을 자막으로 표시하는 것으로 시청자의 선택에 따라 활성/비활성이 가능하다.

● 오픈(open) 캡션

텔레비전이나 여러 가지 영상의 사라지지 않는 구성 요소로서 화면에 나타난다[8].

캡션은 원격 위성 강의라든가 원격 가상 교육을 통한 전문 교육 시스템에 활용이 가능하다. 현재 부분적으로 국내 방송에 클로즈드 캡션의 사용을 실시하고 있다. 또한 교육 전문 사이트에서는 비영어권 사람들의 영어 교육을 위해 사용할 수도 있다. 뿐만 아니라 청각 장애자들을 위한 시설에 사용하는 것은 이미 외국에서는 활용되고 있는 부분이다. 외국의 몇몇 국가에서는 TV에 캡션 디코드를 내장하는 것을 의무화하고 있다. 오픈 캡션은 우리들이 아주 밀접하게 접하고 있는 영화관에도 사용이 되고 있다. 외화의 자막으로 사용이 될 뿐만 아니라 비디오에도 사용이 된다. 이처럼 캡션의 사용 영역은 아주 크며 활용 가치도 아주 높아 TV에 사용되던 것이 이제는 인터넷을 타고 있으며 머지않아 위와 같은 서비스뿐만 아니라 실시간 화상 회의, 가상 대학에도 사용 가능성이 충분히 보여진다.

3. 동기화 된 클로즈드 캡션

최근 여러 곳에서 캡션에 관한 문제 해결책을 발표하고 있다. 대표적으로 Microsoft사에서 1997년 6월에 발표한 SAMI(Synchronized Accessible Media Interchange Format)기술은 멀티미디어 소스파일(오디오, 비디오나 애니메이션)에 시간적으로 동기화 된 캡션 파일을 새로 만들고 DirectShow 미디어 플레이

이를 통해 클로즈드 캡션을 제공하는 계획을 발표했다[14]. 하지만 본 논문은 SAMI 방식과는 달리 기존에 활용중인 여러 가지 스트리밍 소프트웨어를 그대로 사용하면서 클로즈드 캡션을 지원하는 방법에 대해서 연구를 하고 그 방법을 구현하였다.

3.1 캡션 구현 방법

본 논문에서 살펴본 클로즈드 캡션 정보에 대한 몇 가지 해결 방안은 다음과 같으며 캡션 정보에 대한 가장 큰 관건은 오디오와 동기화가 적절히 이루어져야 한다는 것이다. 먼저 자막을 제공하는 방법을 살펴보면 다음과 같다.

- 첫째로 비디오 정보에 직접 캡션 내용을 삽입하여 스트리밍하는 방법이 있다. 이 방법은 100%의 캡션 동기화를 보장하지만 오픈 캡션이 되어 캡션의 활성/비활성의 기능을 제공할 수 없다. 본 논문에서 제시하는 동영상의 클로즈드 캡션 제공 방법은 아니다.
- 둘째, 자막 내용 파일을 따로 작성하여 시간 이벤트에 따라 웹 브라우저의 플러그인이 자막 파일에서 해당하는 자막 정보를 얻어와서 자막을 출력하는 방법이 있다. 이 방법은 자막 내용을 서버로부터 지속적인 통신으로 필요시마다 정보를 가져와야 한다. 인터넷의 불안정한 트래픽을 고려한다면 현실적으로는 적합한 방법이 아니다.
- 셋째, 동영상 파일의 특정한 트랙내에 클로즈드 캡션을 삽입하여 비디오 정보의 시간 이벤트에 따라 메시지를 통해 클로즈드 캡션 정보를 출력하거나 처리를 하는 방법이 있다. 이 방법은 스트리밍 제품이 제공을 해주어야 하는 부분이다. 많은 스트리밍 소프트웨어들이 비록 동기화된 방식은 아니지만 텍스트를 동영상 화면에 제공하는 방법을 이미 가지고 있으므로 클로즈드 캡션을 쉽게 구현할 수 있다.

본 논문에서는 위의 방법 중 3번째 방식을 이용한다. 구현 대상으로는 여러 가지가 있지만 VXtreme이 3번째 방식을 구현하는 쉬운 메커니즘을 제공하므로 [23] 기본 문제 해결 방안으로 VXtreme 서버를 사용한다.

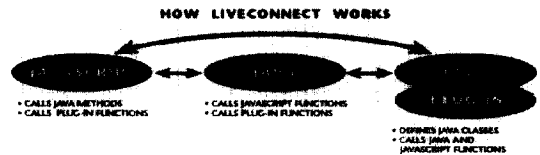
3.2 클로즈드 캡션기능의 구현

3.2.1 CaptionView 클래스의 등록

클로즈드 캡션기능을 지원하기 위해서 구현된 CaptionView라는 클래스를 클라이언트에 먼저 설치를 한다. 이것은 자바 애플릿에서 비디오의 스트리밍을 조작하기 위해서는 클라이언트의 특정 포트를 제어하여야 하기 때문에 서버로부터 내려 받기를 할 경우 보안에 문제가 생겨 포트의 접근이 되지 않으므로 클라이언트에 직접 설치를 하여야 한다. 설치는 스트리밍 클라이언트가 등록되어 있는 디렉토리에 설치되어야 하므로 본 논문에서는 MS C++ 5.0을 이용하여 레지스트리 정보를 읽어들이어 스트리밍 클라이언트의 설치 위치를 파악한 후 클라이언트의 특정 디렉토리에 CaptionView 클래스 파일을 등록하는 프로그램을 MS C++ 5.0을 이용하여 구성한 뒤 이 프로그램을 이용하여 등록이 되도록 하였다.

3.2.2 Java, JavaScript, Plug-In의 동작

동일한 HTML 문서 안에서의 자바스크립트와 자바 애플릿과 플러그인과의 통신 방향은 (그림 1)과 같이 이루어진다[17, 18, 19].



(그림 1) JavaScript와 Java Applet과 Plug-In과의 통신 방향

다른 프레임으로 나누어진 경우 서로 다른 프레임간의 자바 애플릿과 자바스크립트는 서로 직접적인 통신이 불가능하므로 프레임간의 통신에 좀더 복잡한 방법을 사용한다. 우선 자바스크립트를 이용한 접근이 가능하다. 이것은 현재 프레임의 자바스크립트로 다른 프레임의 자바스크립트를 호출하고 호출 당한 자바스크립트가 다시 애플릿을 제어하는 방법이다. 또 다른 방법은 브라우저의 윈도우 객체를 이용하여 접근하는 것인데 이것은 현재 프레임의 자바 애플릿이 직접 윈도우 객체의 참조를 얻어서 현재 프레임의 자바스크립트를 호출하여 이 자바스크립트가 다시 다른 프레임의 자바스크립트와의 통신으로 애플릿과의 통신을 성립시

킨다. 본 논문에서 구현한 시스템의 전체적인 구성은 자바스크립트, 자바 애플릿 그리고 플러그인의 통신방법을 이용하여 (그림 2)와 같은 프레임 구성으로 화면을 디자인하였다. (그림 2)에서 두 가지 애플릿이 존재하는데 CaptionView와 CommandApp라는 애플릿이다. CaptionView는 화면에 클로드드 캡션을 출력하기 위해 내부의 또 다른 애플릿인 Annotation과의 통신으로 클로드드 캡션을 만들어 출력을 하게되고 비디오의 재생이 끝났을 때 끝남을 알리기 위해 브라우저 윈도 객체를 이용한 프레임간의 통신으로 CommandApp에게 메시지를 전달한다. CommandApp의 역할은 비디오 재생, 정지, 반복 등을 제어하고 CaptionView에게 명령을 내리는 역할을 한다.

(그림 2) 전체 프레임 구성화면

3.2.3 캡션 클래스

VXtreme의 제품은 비디오, 오디오, ticker, flipper, toc(Table of Content) 등으로 5가지의 트랙을 지원하며 toc를 제외한 나머지는 모두 서로 다른 파일로 저장되고 스트리밍 된다. 본 논문에서는 이중에서 ticker 트랙의 이용으로 시간적으로 동기화 된 캡션을 가능하게 한다. VXtreme의 동작 원리는 애플릿 간의 통신으로 이루어진다. Annotation이라는 애플릿이 서버로부터의 데이터를 계속 요청하고 전송 받는다. 이렇게 전송 받은 데이터를 ticker, flipper 등의 애플릿으로 메시지 통신을 통해 값을 넘겨준다. 값을 받은 ticker나 flipper는 받은 값을 처리하게 된다. 여기서 ticker에게 전송되는 메시지가 바로 저작 도구를 이용하여 메시지를 넣은 값이 된다. 클로드드 캡션을 구현하기 위해 이 메시지를 가로채어 사용한다. Annotation은 ticker에게 받은 메시지를 setMsg() member function을 통해 통신을 하므로 ticker 클래스를 상속받

아 메시지를 가로채기 위한 public member function인 setMsg()를 이용한다. 여기서 member function인 setMsg()는 넷스케이프의 자바 콘솔창과 ticker.class파일의 디버깅으로 찾아내었다.

```
public void setMsg(String s)
{
    //중략
    message = s; // 메시지를 처리하기 위해 받는다.
    //중략
    getConString(); // 메시지를 처리한다.
}
```

(그림 3) setMsg() member function

메시지를 setMsg()를 통해 받은 후에 그 메시지를 다시 처리하기 위해 변수 "message"에 값을 넘겨준다. 변수 "message"에 입력된 메시지를 가공하고 분석해서 다음 명령을 처리하기 위해 (그림 4)의 getConString()

```
public void getConString()
{
    String token;
    int i=0;
    // String Tokenize를 수행한다.
    StringTokenizer st = new StringTokenizer(message, "!");
    // String이 입력될 변수를 초기화한다.
    init_String();
    // 이전 메시지와 비교
    if(p_message.equals(message)) {
        fflag=false;
        return;
    }
    fflag = true;
    p_message = message; // 이전 메시지를 기억한다.
    // 메시지의 끝임을 비교한다. 비교키는 '-'
    if(message.equals("-"))
    {
        // 메시지의 끝임을 확인하고 Window의 reference를 이용하여
        // 'setCon'이라는 JavaScript를 작동시켜 다음 동작을 수행한다.
        putmsg[0] = "stop";
        win.call("setCon", putmsg);
        System.out.println("=====>STOP");
        message = "";

        putmsg[0] = "play";
        win.call("setCon", putmsg);
        System.out.println("=====>PLAY");
        message = "";

        System.exit(1);
        return;
    }

    // 잘려진 Token을 str 변수에 입력하여 출력을 준비한다.
    while(st.hasMoreTokens()){
        token = st.nextToken();
        str[i] = token;
        i=i+1;
    }
}
```

(그림 4) getConString() member function

을 호출한다. 캡션을 보여줄 라인을 구현하기 위해 getConString을 이용하여 입력받은 문자열로부터 파싱을 한다. getConString()에서는 5라인까지 고정되어 있다. 이 논문에서는 화면출력시에 적당한 라인수를 5줄이라 가정한다.

3.2.4 자막의 활성화/비활성

자막의 활성화/비활성 기능은 전송되어 오는 메시지는 계속 받으나 화면에 출력할 때 글자의 색깔을 바탕의 색과 같은 색으로 출력함으로써 가능하다. 자바스크립트는 CaptionView의 public member function인 setShow()와 setHide()를 이용하여 클로즈드 캡션의 활성화/비활성을 선택 가능하게 한다. 이때 다른 프레임에서 특정한 프레임으로의 메시지 통신을 위해서는 넷스케이프의 LiveConnect 기법이 필요하다. 본 논문에서의 자바 애플릿과 자바스크립트 간의 통신은 넷스케이프의 LiveConnect 기법을 이용한 서로간의 메시지 통신으로 해결하고 있다[17].

(그림 5)와 같이 Netscape의 LiveConnect를 이용한 메시지 통신은 애플릿이나 자바스크립트의 "name"을 이용하여 접근할 수 있다. 위에서는 ticker라는 것이 애플릿의 이름이 되고 setShow()나 setHide()는 ticker라는 이름의 애플릿의 공개된 member function이 되는 것이다. 이런 방식으로 애플릿이나 자바스크립트에 직접 접근할 수 있다. 그리고 CaptionView에서 공개된 member function은 (그림 6)과 같다.

```
<form name="myform">
Caption :
<input type="button"
value="On" onclick="parent.frames[0].document.ticker.setShow()">
<input type="button"
value="Off" onclick="parent.frames[0].document.ticker.setHide()">
</form>
```

(그림 5) JavaScript와 Java Applet과의 LiveConnect

```
public void setHide() {
    setForeground(new Color(0,0,0));
}

public void setShow() {
    setForeground(new Color(255,255,255));
}
```

(그림 6) 활성화/비활성을 위한 function

3.3 선택적 보기, 반복기능

선택적 보기란 검색결과로 얻어진 동영상 리스트 중 여러 개의 동영상상을 선택하여 연속하여 선택된 동영상을 보는 것이다. 반복이란 비디오의 반복 횟수 선택으로 선택된 비디오의 반복회수만큼 출력하는 것이다. 현재 본 논문에서 구현된 것은 선택적 보기 가능한 비디오의 수는 5개이며 선택된 비디오에 대하여 각각 최대 10번까지 반복하여 재생할 수 있다. 선택적 보기와 반복 기능의 구현은 선택된 비디오에 대하여 반복의 횟수와 함께 배열로 저장되어 되고 저장된 배열을 하나씩 읽어 오면서 비디오의 재생과 반복을 수행한다. CaptionView에서 비디오의 마지막임을 CommandApp에서 알려 오면 CommandApp에서는 다음 배열의 비디오와 반복 횟수로 다시 비디오의 재생을 시작한다. (그림 7)에서 보면 "st" 라는 배열이 선택된 비디오의 리스트를 저장해 놓았다. 이 배열에서 배열의 인자를 증가시킴으로써 선택된 다음 비디오의 정보를 얻을 수 있다.

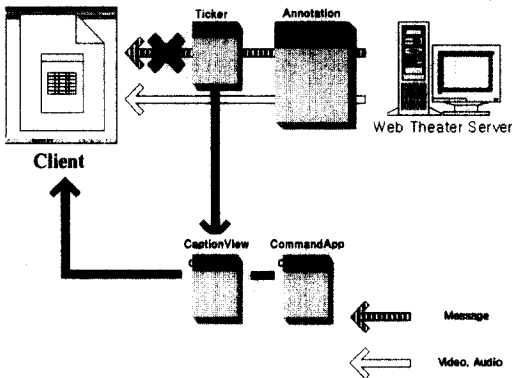
```
public void getVideoList()
{
    // videoList는 전체 선택된 비디오의 갯수
    // p_videoList는 비디오의 증가치

    // 만약 비디오의 리스트가 끝이 났다면
    if(p_videoList == videoList || st[p_videoList].equals(""))
    {
        System.out.println("END STREAMING");
        setCommand("stop"); // stop을 CommandApp 에게 알린다.
        return;
    }
    String fullUrl = st[p_videoList];
    System.out.println(repeatNum + " === "+fullUrl);
    // 정상적으로 비디오의 다음 정보를 선택했다면 비디오의 증가치를 증가
    p_videoList++;
    setRepeat(p_repeatNum); // 선택된 비디오의 반복치를 init한다.
    try {
        // 선택된 비디오를 화면에 출력
        getAppletContext().showDocument(new URL(fullUrl), "videoFrame");
    } catch (MalformedURLException e) {
        System.out.println("MalforedURLErrorException: " + e);
    }
    return;
}
```

(그림 7) getVideoList()

문제 해결 방법을 그림으로 도시하면 (그림 8)과 같다. 이 시스템의 흐름은 기존의 스티리밍 소프트웨어인 Web Theater는 전혀 손대지 않은 상태에서 Ticker라는 이름

의 애플릿을 본 논문에서 구현한 CaptionView라는 애플릿으로 변경하여 사용하면 가능하다. 기존의 다른 웹 페이지에서는 전혀 영향을 받지 않는다. Ticker라는 이름의 애플릿은 기존에는 Annotation이라는 애플릿과의 메시지 통신을 통하여 브라우저에 결과를 출력한다. 하지만 여기서 Ticker라는 이름의 애플릿을 CaptionView로 재설정하여 CaptionView에서 필요한 작업을 거친 가공의 데이터를 브라우저에 출력하게 된다. 이때 CaptionView의 클로즈드 캡션처리, 선택적 보기, 반복 등의 기능을 처리하기 위해 CommandApp와의 메시지 통신으로 마지막 처리를 거친 최종의 데이터가 브라우저의 화면에 클로즈드 캡션으로 출력되고 기능을 수행한다. 이로써 클로즈드 캡션의 실질적인 구현이 가능하다.



(그림 8) 전체 시스템의 구성 및 제안 시스템 구성도

4. 평 가

먼저 자바 애플릿에는 클라이언트의 특정 포트에 대한 보안문제로 클라이언트에 소켓 접속이 불가능하므로 CaptionView라는 클래스 파일을 특정한 디렉토리에 직접 설치할 하여야 하는 문제점이 뒤따른다. 그리고 비디오의 처리와 자바 애플릿의 사용으로 브라우저에서의 속도 문제를 들 수 있는데 클라이언트 시스템 평가결과 Pentium 233Mhz이상과 RAM 48M이상의 시스템에서 무리 없이 동작하였다. 비디오의 크기와 품질을 위해서 본 논문에서는 320×240의 화면 크기를 사용하였다. 동일한 서버넷이나 라우터에서는 어느 정도 품질을 보장하지만 외부망의 접근에서는 이와 같은 비디오의 처리를 위해서는 보다 높은 사양의 컴퓨터가

필요하며, 그렇지 않은 경우 화면 사이즈를 160×140 정도의 작은 사이즈로 구성하여야 한다. 상용 제품 중의 하나인 VXtreme 제품으로 서비스되고 있는 사이트는 160×140정도 이하의 아주 작은 크기의 비디오를 서비스하고 있는 실정이다.

5. 결론 및 향후 연구 방향

최근 수년 동안 데이터베이스를 사용한 응용 분야는 급속히 변화되고 있다. 상대적으로 작은 크기의 정형적인 문자 및 숫자의 저장과 잦은 데이터의 변경 및 검색이 주연산인 온라인 트랜잭션 업무에서 오디오, 비디오, 이미지 등과 같이 비교적 대용량의 멀티미디어를 저장하고 검색하는 온라인 검색 시스템으로 변화하고 있다. 응용 분야의 다양화와 함께 데이터베이스 기술도 기존의 관계형 모델에서 여러 가지 멀티미디어를 효과적으로 지원하는 객체지향 또는 객체관계형으로 변천되고 있다. 새로운 데이터 타입의 지원이 손쉬운 객체지향 기법의 도입과 함께 멀티미디어의 특성에 맞는 데이터 타입의 생성과 연산이 새롭게 만들어지고 있으며, 인터넷 환경에서 검색된 멀티미디어 데이터를 사용자 공간에서 효과적으로 보여주는 방법들이 계속 연구되고 있다.

본 논문에서는 동영상 데이터베이스를 이용한 원격 교육 시스템의 구축 사례로 클로즈드 캡션을 제공하는 원격 교육 시스템을 구축하였다. 특히 동영상 세그먼트를 사용자에게 디스플레이 할 때 대사 내용을 동영상과 동기화시키면서 문자로 보여줄 수 있는 방법, 사용자가 문자 디스플레이를 선택 할 수 있는 기능 첨가, 사용자가 볼 영화를 한꺼번에 여러 개 선택, 또는 반복 선택 할 수 있도록 하는 첨단 기능을 구현한 시스템을 보여주고 있다. 본 논문의 연구내용은 실제 INFORMIX UNIVERSAL SERVER와 VXtreme 스트리밍 소프트웨어를 이용한 1차 버전 시스템을 구축하여 홈페이지 <http://dblab.keimyung.ac.kr/~sbsin>에 저장하고 논문에서 연구된 방법을 실제 적용하여 연구 결과의 효용성을 보였다. 멀티미디어 데이터베이스에서 동영상 데이터를 저장하는 방법 이외에도 사용자에게 효과적으로 전달하는 방법의 일환으로 연구된 이 논문의 내용은 현재 Microsoft와 같은 기업들이 인터넷에

서의 동영상 사용이 수년 안에 활성화 될 것으로 예견하고 많은 노력을 기울이고 있는 분야로 국내에서도 많은 노력을 기울여야 할 분야이다.

본 논문의 내용은 단일 웹 서버와 단일 스트리밍 서버 환경에서 수행된 내용이다. 본 논문의 연구 결과로 구성된 데모 시스템의 활용을 전국으로 확대해서 모든 사용자가 불편 없이 사용하려면 한쪽으로 몰리는 부하 편중과 네트워크의 트래픽이 가장 큰 장애가 될 것이다. 현재 개발하고 있는 2차 버전의 시스템은 여러 개의 스트리밍 서버와 지역별로 여러 개의 다른 데이터베이스를 효과적으로 유지하는 방법을 포함하고 있다. 사용자들의 사용 형태, 부하의 정도를 추적하여 데이터의 분산, 부하의 분산을 스스로 이루어내는 능동 기능을 가진 시스템의 기능을 구현하여 국내의 모든 인터넷 사용자가 원활히 사용할 수 있는 시스템의 구현 방법에 대한 연구는 상당히 의미있는 것으로 판단이 된다.

참 고 문 헌

- [1] 박근한, 오황석, 김길윤, 전준현, 이홍구, "인터넷 상에서의 요구형 멀티미디어 서비스 시스템 구현", 정보과학회논문지 96.12 제2권 제4호, pp.390-398.
- [2] 박지숙, 송병호, 이석호, "멀티미디어 데이터 표현을 위한 시간적 모델화", 정보과학회지 논문지 93.7 Vol.20 , pp.957-966.
- [3] 신명기, "Web and Real-Time Multimedia," 제5회 WWW Workshop, pp.239-249, 1997.
- [4] Blam Entertainment Group, "DVD Technical Information Summary," <http://www.blamld.com/Universal/DVD/DVDTechnicalSummary.htm>, 1998.
- [5] David Bolnick, "SAMI," Microsoft, 1997.
- [6] David W. Brubeck, Lawrence A. Rowe "Hierarchical Storage Management in a Distributed Video-On-Demand System," IEEE Multimedia, Vol.3, No.3, pp.37-47, 1996.
- [7] Eric M. Hoffert, Joshua B. Weisberg, "New Media Data Management," http://www.magnifi.com/wpaper/wpaper_html/magnifi_wp2.htm, 1997.
- [8] Gary D. Robson, "Inside Captioning," CyberDaw Publishing, 1998.
- [9] Graham Hamilton, Rick Cattell, Maydene Fisher, "JDBC Database Access with Java," Addison Wesley, 1997.
- [10] INFORMIX, "INFORMIX Universal Server Manual," INFORMIX, 1997.
- [11] Jan Ozer, "Audio, Video LIVE from the web," PC magazine, pp.100-136, March, 1996.
- [12] Ketan Mayer-Patel, David Simpson, David Wu, Lawrence A. Rowe "Synchronized Continuous Media Playback Through the World Wide Web," EECS of University of Berkeley, 1996.
- [13] Microsoft, "Microsoft NetShow," <http://www.microsoft.com/netshow/>.
- [14] Microsoft, "Synchronized Accessible Media Interchange Captioning & Audio Description," Microsoft Corporation, Nov 18, 1998.
- [15] National Captioning Institute, "New Captioned Home Videos From MCI," <http://www.us.net/nci/interna.htm>.
- [16] National Center for Accessible Media, "Captioning QuickTime Movie Clips," <http://www.wgbh.org/wgbh/pages/ncam/qtcaptionhowto.html>, 1997.
- [17] Netscape, "Technical Manuals, Java, JavaScript, Plug-in," DevEdge Online Document, Sep., 1998.
- [18] Netscape, "JSObject Class," JDK Documentation, 1996-1997.
- [19] Netscape, "LiveConnect/applet API packages," JavaScript Documentation.
- [20] Peter Gutmann, "Closed Video Captioning Rules," Pepper & Corazzini, L.L.P., 1997.
- [21] SunWorld, "최신 객체 지향 데이터베이스", SunWorld, Nov., 1996.
- [22] Thomas Boutell, "World Wide Web(WWW) Frequently Asked Questions," Nov., 1994.
- [23] Vxtreme, "Vxtreme Web Theater Server, Author, Capture, Client Guide," Vxtreme Help Documentation, 1997.
- [24] W3C, "Synchronized Multimedia Activity," W3C Consortium, Nov, 1998.



신 승 봉

e-mail : sbshin@dblab.keimyung.ac.kr

1997년 계명대학교 통계학과
졸업(학사)

1999년 계명대학교 대학원 전자계
산학과(공학석사)

1999년~현재 주식회사 대동

관심분야 : 데이터베이스, 인터넷 응용



황 인 재

e-mail : lh@ghost.chungbuk.ac.kr

1986년 충북대학교 컴퓨터공학과
졸업(공학사)

1981년 University of Florida.
Computer & Information
Sciences 졸업(공학석사)

University of Florida. Computer & Information
Sciences 졸업(공학박사)

1986년~1987년 한국 전자통신연구소 연구원

1995년~현재 충북대학교 컴퓨터교육과 조교수

관심분야 : 병렬처리, 병렬컴퓨터 구조, 병렬 알고리즘



홍 동 권

e-mail : dkhong@knucc.keimyung.ac.kr

1985년 경북대학교 전자공학과
졸업(학사)

1992년 University of Florida
전자계산학과 졸업(석사)

1995년 University of Florida

전자계산학과 졸업(박사)

1985년~1990년 한국전자통신연구원

1996년~1997년 한국전자통신연구원

1997년~현재 계명대학교 컴퓨터·전자공학부 전임강사

관심분야 : 능동 실시간 데이터베이스, 병렬 처리, 성능
평가, 시뮬레이션, 멀티미디어 처리