

工學碩士 學位論文

指導教授 金 日 民

자바를 이용한 강의 저작 도구의 설계 및 구현

Design and Implementation of a Lecture Authoring Tool
using Java Language

2004年

漢城大學校 컴퓨터 신기술 大學院

컴퓨터 신기술 學科

컴퓨터 工學 專攻

張 恩 燮

工學碩士 學位論文

指導教授 金 日 民

자바를 이용한 강의 저작 도구의 설계 및 구현

Design and Implementation of a Lecture Authoring Tool
using Java Language

2003年 12月

위 論文을 工學碩士 學位論文으로 提出함

漢城大學校 컴퓨터 신기술 大學院

컴퓨터 신기술 學科

컴퓨터 工學 專攻

張 恩 燮

張恩燮의 工學碩士 學位論文을 認定함

2003年 12月 日

심사위원장 정 인 환 (인)

심 사 위 원 김 일 민 (인)

심 사 위 원 조 세 홍 (인)

국 문 요 약

인터넷의 등장과 통신기술의 발달로 인해 여러 분야에서 원격 교육이 시도되고 있다. 이러한 점을 통해 다양한 멀티미디어 기술을 활용한 교육 콘텐츠의 저작이 일반화되고 있다.

하지만 기존에 제안된 멀티미디어 콘텐츠를 이용한 원격교육 시스템들은 대부분의 기능상에 문제점을 가지고 있다. 또한, 원격 교육을 통한 강의를 진행하기 위해서 강의자는 관련 프로그래밍 기술 등을 따로 익혀야 하므로 상당한 컴퓨터 관련 지식이 요구되므로 가상강좌를 개설하기가 어려운 실정이다.

본 논문에서는 이러한 어려움을 해결하기 위하여 강의자와 학생의 학습을 효과적으로 지원하는 가상 강의 저작도구를 설계 및 구현하였다. 따라서 컴퓨터 관련 지식 및 프로그래밍 지식이 없는 강의자도 간단한 강의 관련 정보 입력과 사용자 작성 메뉴를 사용함으로써 가상 강의를 제작할 수 있는 점을 강조한다. 특히 이 강의저작도구는 JVM환경하에서 구현했으므로 개개의 컴퓨팅환경에 큰 영향을 받지 않는 점이 큰 장점이다.

이 시스템은 교수 강의 저작 도구를 구현하고, 수강생 개개의 학습이 가능하도록 저작된 강의를 재생함을 목적으로 하고 있다.

목 차

요 약

1. 서 론	1
2. 관련연구	3
2.1 원격교육	3
2.1.1 웹브라우저 의존 교육 서비스	4
2.1.1 웹브라우저와 플러그인 의존 교육 서비스	4
2.2 다중 스레딩(MultiThreading)	5
2.2.1 스레드 내에서의 변수의 스코프	8
2.2.2 효과적인 다중 스레딩	9
2.3 동기화(Synchronization)	10
2.3.1 세마포어(semaphores)	13
2.3.2 동기화 효율	14
2.4 입출력 스트림	15
2.4.1 문자 스트림과 바이트 스트림	17
2.5 JMF(Java Media Framework)	18
3. 시스템 분석 및 설계	20
3.1 시스템의 주요 기능 및 요구사항 분석	20
3.2 UML을 이용한 시스템 분석	21

3.2.1 Use Case 모델링	22
3.2.2 Sequence Diagram	25
3.2.3 Class Diagram	28
4. 시스템 구현	29
4.1 시스템의 구성	29
4.2 시스템의 동작	31
4.2.1 시스템의 초기화	32
4.2.2 강의 만들기	33
4.2.3 강의 노트 편집	34
4.3 강의노트 이미지에 쓰기	35
4.4 판서시 판서정보 저장	36
4.5 객체 저장 방식의 사용	38
4.6 저장 시간의 차별화	39
5. 결과 및 분석	41
6. 결론 및 향후 과제	43
참고문헌	45
ABSTRACT	46
부 록	48

그림 목차

그림 2.1. 단일 스레드.....	5
그림 2.2 다중 스레드.....	6
그림 2.3 스레드의 Life Cycle.....	7
그림 2.4 스레드의 동기화.....	10
그림 2.5 시간의 흐름에 따른 동기화 메소드.....	12
그림 2.6 동기화 블록.....	13
그림 2.7 읽기 스트림.....	15
그림 2.8 쓰기 스트림.....	16
그림 2.9 JMF 구조.....	18
그림 3.1 Use Case Diagram(강의 제작).....	23
그림 3.2 Use Case Diagram(강의 재생).....	24
그림 3.3 실행 초기화(Sequence Diagram).....	25
그림 3.4 강의 제작 Sequence Diagram.....	26
그림 3.5 강의 노트 편집 Sequence Diagram.....	27

그림 3.6 판서시 Class Diagram.....	28
그림 4.1 강의 저작 도구 시스템의 구성.....	29
그림 4.2 강의 만들기 폴더 구조.....	30
그림 4.3 시스템의 동작.....	31
그림 4.4 초기 실행 화면.....	32
그림 4.5 강의 저작 도구 초기화 상태.....	32
그림 4.6 강의 저작 도구를 이용한 판서.....	33
그림 4.7 강의 노트 활용.....	34
그림 4.8 강의 노트에 쓰기.....	35

표 목 차

표 2.1 하나의 메소드 안에서 스레드의 실행.....	11
표 3.1 각각의 ACTOR의 역할.....	24
표 5.1 도구적 측면 비교 분석.....	41
표 5.2 서비스 측면 비교 분석.....	42
표 5.3 데이터 전송 방법의 비교.....	42

1. 서 론

1986년 가상(virtual) 수업이라는 용어가 Hiltz에 의하여 처음 사용된 이후 90년대 중반 가상교육, 가상대학이라는 용어가 보편화되어 활용되고 있다. 지금까지 여러 사람들에 의하여 정의된 가상교육 또는 가상대학의 개념은 다음과 같다. "가상대학이란 정보통신기술을 이용한 가상의 공간 또는 사이버 공간에서 고등교육을 받을 수 있도록 구성된 새로운 교육체제이다." [10] 여기서 가상대학은 기존 방송 등의 매체를 이용한 고등교육 기관도 포함하는 개념이나, 특히 사이버 공간상에서의 가상교육을 강조하고 있다.

"가상교육이란 외형은 일반적인 교육과 다르지만, 일정한 교육과정을 이수하고, 학위를 받는 등 실질적으로는 기존 교육과 동일한 교육 체제를 의미한다." [12]

최근 급속한 정보통신의 발달은 사회 전체의 빠른 변화를 주도하고 있으며 교육 분야에 있어서도 원격강의라는 새로운 교육 형태를 가져왔으며 시간과 공간의 제약을 벗어나 각종 멀티미디어 자원을 이용하는 영역으로 확대해 가고 있다.

원격교육은 새로운 지식창조를 위한 인재양성을 위하여 정보인프라와 정보서비스의 활용이 중요시되는 형태로 변화되고 있다. 과거의 단순한 지식전달 학습형태에서 벗어나 멀티미디어 콘텐츠(multimedia contents)를 포함한 한 차원 높은 원격교육 시스템을 필요로 하고 있다. 미래의 원격교육(distance education)은 인터넷상에 분산되어 있는 멀티미디어 콘텐츠의 이용을 보다 쉽고, 안정된 방법으로 서비스를 제공해야 한다. 또한 사용자 인터페이스 측면에서 사용하기 쉽고, 사용자의 요구 사항에 적절히 원하는 정보를 선택적으로 제공할 수 있는 기능들을 포함하여야 한다. 그리고 시스템 측면에서는 기존의 시스템들과 쉽게 통합되거나 통합할 수

있는 인터페이스를 지원해야하며, 기존의 시스템보다 빠르거나 최소한 동등한 성능을 보장하여야 한다. 그리고 쉽게 강의를 제작 배포할 수 있어야 한다. 이러한 시스템을 구축하기 위해서는 인터넷 서비스인 웹(Web)을 이용하여 효과적으로 구축할 수 있다. 하지만 기존의 원격교육 시스템은 학습자의 교육만을 주안점으로 개발되어 인터넷을 활용한 원격강의가 이루어지고 있다. 또한 강의자는 사이버 강의를 개설하고 싶어도 관련 프로그래밍 기술을 익혀야 하므로 상당한 관련 지식이 요구되어 원격강의를 개설하기가 어려운 실정이다. 이러한 제반 문제들을 해결하고자 본 논문은 강의자와 학생의 학습을 효과적으로 지원하는 JAVA기반의 가상 강의 저작도구를 제안한다.

2. 관련연구

본 장의 구성은 다음과 같다. 2.1절에서는 원격 교육의 목적과 원격교육을 위한 효과적인 기술방법론을 명시하며 2.2, 2.3, 2.4절에서는 동기화된 멀티미디어 사용을 위한 자바 관련 기술을 이용하는 방법을 고찰한다. 또한 2.5절에서는 자바에서 멀티미디어기술을 이용하기 위한 JMF에 대해 고찰한다.

2.1 원격교육

원격교육은 교육자와 피교육자 사이에 다양한 교습매체의 활용과 1대 1 수업을 병행함으로써 교육의 질적 향상을 최대화하려는 제반 활동을 의미한다.[11] 원격교육은 초창기 통신(correspondence)교육에 기반을 두었지만 최근에는 위성, 압축 영상과 컴퓨터 등과 같은 정보통신 기술의 발달로 인해 인터넷과 멀티미디어 분야의 기술을 사용하여 교육매체로 이용하고 있다.

현재 급격히 발전하고 있는 기술과 이론들은 컴퓨터 분야뿐만 아니라 타학문 분야에서도 새롭고 흥미로운 환경을 구성할 수 있는 좋은 수단으로 대두되고 있다. 특히 교육자들은 현재 교육 분야가 당면하고 있는 문제점들 가운데 정규 교육 현장과 달리 학생이나 사회인들에 대한 재교육 문제와 교육 재정 부족에 따른 문제를 해소할 수 있는 유력한 대안으로 새로운 컴퓨터 기술을 꼽고 있다. 그 환경들 중 하나가 원격 교육이다. 컴퓨터를 이용한 원격교육 시스템은 1950년대 일리노이 대학의 Donald Botzer 박사에 의해 구상되어 1961년에 이르러 PLATO 시스템으로 개발된 이래 30년 동안 다각적인 연구가 이루어져 왔고, 이들 시스템들은 눈부신 하드웨어의 발전과 멀티미디어 기술의 개발, 그리고 인터넷으로 대표되는 정보통신 기술을 통해 다양하고 풍부한 학습내용을 체계화하여 교수학습활동에 적극적으로 활용되고 있다.[11]

2.1.1 웹브라우저 의존 교육 서비스

웹 브라우저 의존적 교육 서비스란 인터넷의 한 구성 요소인 웹의 문서 기술 언어인 HTML(Hyper Text Markup Language)과 스크립트 언어, 혹은 CGI(Common Gateway Interface)로 구성되어 인터넷 익스플로러(Internet Explorer), 넷스케이프(Netscape Navigator)와 같은 웹 브라우저를 통해 서비스가 제공되는 것을 말한다. 사용자와의 상호작용을 높이기 위해 CGI나 스크립트언어를 사용해서 사용자의 반응, 즉 폼 양식을 통해 내용을 입력하거나 항목을 선택하는 등의 반응을 받아들여 서버로 전송한 후 서버에서 내부적으로 처리한 후 웹 페이지로 가공하여 전송하는 역할을 한다.

2.1.2 웹브라우저와 플러그인 의존 교육 서비스

인터넷을 통해 사용되어 지던 기존의 다양한 서비스와 새롭게 개발된 서비스는 웹 브라우저로 통합되어지고 있다. 그러나 웹 브라우저에서 그 모든 다양한 프로그램을 지원하기는 사실상 어렵다. 그래서 웹브라우저와 인터넷응용프로그램을 연결할 수 있고 각각 따로 개발 할 수 있는 새로운 방식을 개발하게 되었다. 그 새로운 방식이 바로 플러그인(Plug-In)과 ActiveX이다.

웹 브라우저와 플러그인 의존 교육 서비스는 웹 상에서 특정 포맷형식의 자료를 다운 받아서 해당 플러그인을 통하여 서비스가 제공되는 것을 말한다. 플러그인 프로그램은 웹 브라우저에서 처리할 수 있는 형태의 데이터를 처리 할 수 있도록 만들어진 하나의 보조 프로그램이다. 브라우저에서 파일을 전송 받으면 자동적으로 보조 프로그램이 실행된다.

2.1 다중 스레딩(MultiThreading)

다중 스레딩은 자바 응용 프로그래밍에서 매우 중요한 부분을 차지한다. 클라이언트/서버 프로그래밍에서, 서버측에서는 네트워크상에 분산되어 있는 여러 클라이언트들로부터 동시에 들어오는 접속 요청을 정확하게 처리해 주는 것이 매우 중요하다. 이 경우 다중 스레딩을 사용하면 각각의 접속 요청마다 하나씩 스레드를 사용하여 처리하는 효과적이고 견고한 서버를 만들 수 있다. 또한 다중 스레딩을 사용하는 경우 한 클라이언트의 접속 요청을 처리하는 부분을 다른 클라이언트의 접속 요청을 처리하는 부분과 서로 분리시킬 수 있으므로, 접속이 실패하는 경우에도 애플리케이션이 전혀 영향을 받지 않도록 만들 수 있다.[3] 전통적인 프로그래밍 방식은 한 프로그램에서 하나의 제어 흐름만을 갖는 것이 일반적이었다. 이 제어 흐름은 프로그램의 시작 부분에서 시작되고 프로그램이 끝날 때까지 하나의 경로만을 따라서 이동한다. 그림 2.1은 하나의 스레드를 사용하는 애플리케이션을 보여주고 있다.

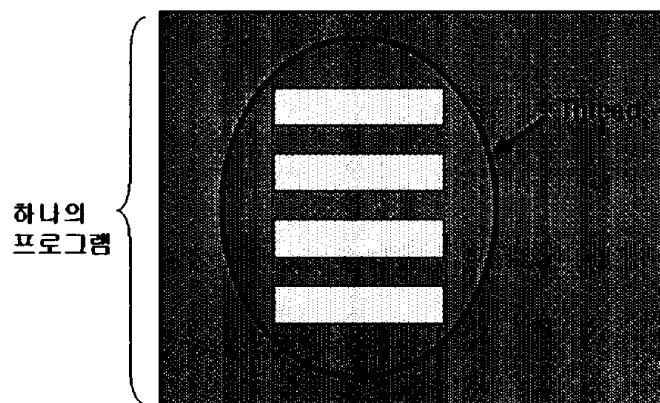


그림 2.1 단일 스레드

다중 스레딩(Multithreading)이란, 한 프로그램에서 이러한 제어 흐름을 동시에 여러 개를 생성하는 것을 뜻한다. 이것은 멀티태스킹(multitasking)과는 약간 다르다. 멀티태스킹이란, 동시에 여러 프로세스를 수행할 수 있다는 것을 의미하며, 이 경우 각각의 프로세스는 독자적인 데이터 공간을 가지게 된다. 그러나 다중 스레딩에서 각각의 스레드는 동일한 데이터 공간 내에서 동작하기 때문에, 만약 한 스레드가 전역 변수(global variable)의 값을 바꾸어 버린다면 다른 스레드들도 바뀐 변수값을 사용하게 될 것이다.[1] 그림 2.2는 여러 개의 스레드를 동시에 사용하는 다중 스레드 애플리케이션을 보여주고 있다.

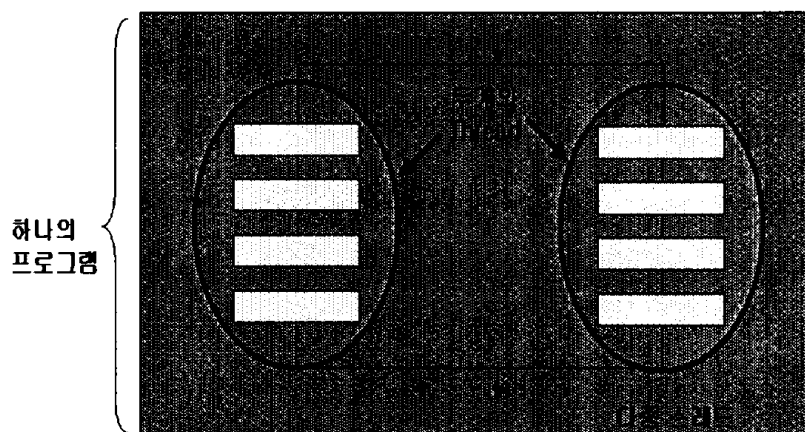


그림 2.2 다중 스레드

다중 프로세서 컴퓨터에서는 각각의 스레드가 서로 다른 CPU에서 동시에 수행될 수 있으며, 이 경우 서버의 작업처리 능력이 크게 향상된다. CPU가 하나밖에 없는 컴퓨터의 경우에도 다중 스레딩을 사용하면 비슷한 성능 향상을 기대할 수 있다.

자바는 언어 수준에서 스레드를 지원한다. 일반적으로 다른 언어에서는 스레드를 사용하기 위해서 보조 라이브러리를 사용해야 하지만, 자바에서는 언어 수준에서 스레드를 지원하기 때문에 스레드를 사용한 프로그래밍을 좀더 쉽게 할 수 있다.

자바에서 스레드를 사용하여 프로그램을 작성하기 위해서는 Thread 클래스를 사용해야 한다. Thread 객체 하나는 실제 프로그램에서의 스레드 하나와 일대일 대응이 되기 때문에, 자바 언어에서 스레드를 제어하기 위해서는 스레드와 대응되어 있는 Thread 객체의 해당 메소드를 호출해 주기만 하면 된다. 자바에서 스레드를 사용하기 위해서는 java.lang.Thread 클래스를 상속하여야 한다. 자바 프로그램에서 사용되는 스레드는 생성되어 소멸될 때까지 life cycle을 가지는데 상태의 이동은 그림의 화살표 방향으로만 이루어지며 모두 데몬 스레드인 thread scheduler의 관리를 받는다. 다음 그림은 자바 스레드의 life cycle을 나타낸 것이다.[1][9]

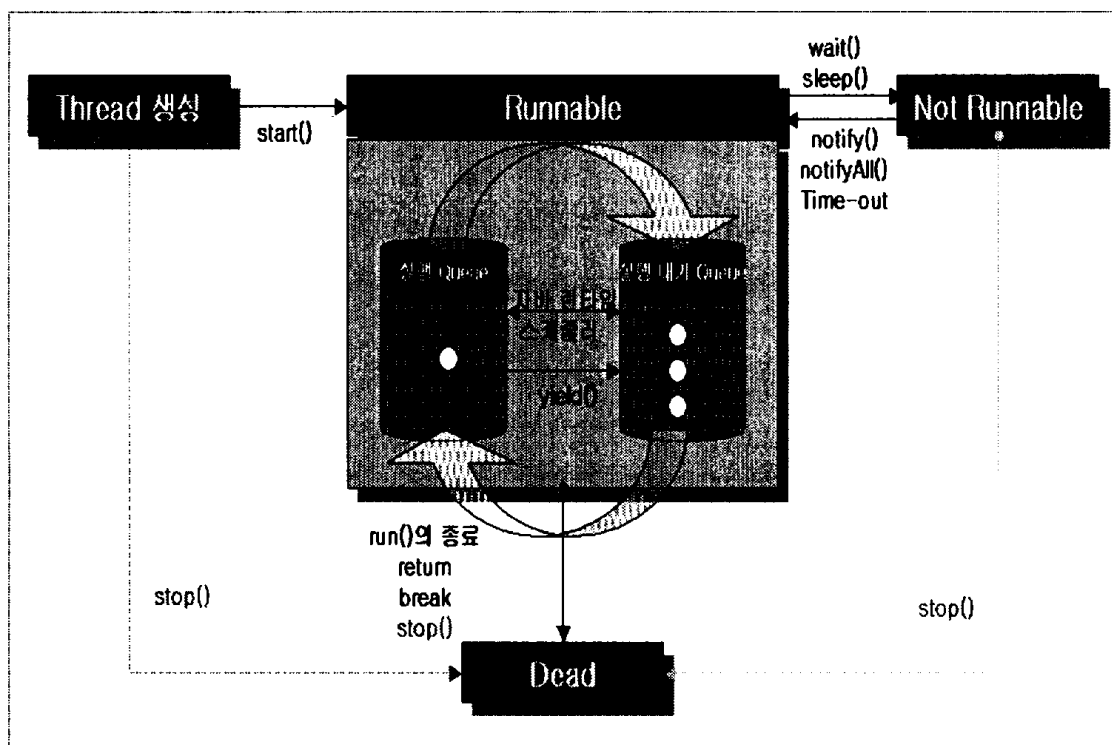


그림 2.3 스레드의 Life Cycle

2.1.1 스레드 내에서의 변수의 스코프

스레드가 수행되는 도중에 선언된 변수는 해당 스레드 내에서만 사용된다. 따라서 스레드가 수행되는 도중에 지역 변수값을 바꾼다 해도, 다른 스레드 지역 변수의 값에는 영향을 미치지 않는다. 여기에서 사용된 ‘지역 변수’라는 단어는 인스턴스 변수를 나타내는 것이 아니며, 메소드 내부에서 선언된 변수를 나타내는 것이다. 메소드 내에서 선언된 변수들은 스레드의 로컬 스택(local stack) 내의 메모리 공간을 할당받게 된다. [1][2][3]

정적인(static) 클래스 멤버 변수들은 모든 스레드에서 하나의 변수 공간을 공유하게 된다. 정적이지 않은(nonstatic) 인스턴스 변수들은 각각의 인스턴스마다 하나의 변수 공간을 가지며, 특정한 인스턴스에 접근하는 스레드들은 해당 인스턴스에 존재하는 하나의 변수 공간을 공유하게 된다. 또, 메소드 매개변수들과 지역 변수들은 하나의 스레드 내에서만 사용된다.

```
class Test{
    static int staticW;
    Vector instanceX;

    void aMethod(int paramY) {
        float localZ = staticW;
        instanceX.addElement(new Float(localZ));
    }
}
```

위 예제에서, aMethod() 메소드를 호출하는 모든 스레드는 자기 자신만의 paramY와 localZ 변수를 저장하는 공간을 할당하게 된다. 또, Test 클

래스로부터 생성된 하나의 인스턴스에서 aMethod() 메소드를 호출한 스레드들은 하나의 instaceX 인스턴스 변수를 공유하게 되며, staticW 정적 변수는 현재 실행중인 모든 스레드에서 공유될 것이다.

어떤 객체에 대한 참조자(reference)를 지역 변수로 선언한 경우, 참조자가 가리키는 객체가 로컬에서 생성된 객체일 것이라는 보장은 없다. 만약 로컬 참조자가 가리키는 객체가 여러 스레드에서 공유되는 인스턴스 변수인 경우, 참조자가 가리키는 객체의 값을 바꾸게 되면 그 객체를 참조하는 모든 스레드가 바뀐 객체를 사용하게 될 것이다.

2.1.3 효과적인 다중 스레딩

자바에서 스레드를 생성하는 것은 매우 느린 작업이며, 일부 자바 가상 머신은 스레드에 관련된 메모리 누수(memory leak) 때문에 고생하기도 한다. 따라서, 좀더 효과적인 스레딩을 위해서는 새로 스레드를 생성하는 것 보다, 이전에 생성했던 스레드 객체를 재사용하는 것이 좋다. 스레드를 재사용하는 방식을 사용하면 여러 가지 이점을 얻을 수 있다.

첫 번째 이점으로는 스레드를 재사용함으로써 가비지 컬렉터의 호출을 좀더 줄일 수 있다는 것이다. 스레드의 잦은 생성 소멸로 인해 생성될 가비지 컬렉션(Garbage Collection)의 대상을 줄임으로써 가비지 컬렉터의 호출을 줄이고 이로 인해 퍼포먼스에 악영향을 줄 수 있는 요소를 작게나마 사전에 예방한다는 것이다.[5]

두 번째 이점은 스레드를 새로 생성하지 않고 이미 생성된 스레드를 가져다가 쓰기 때문에 스레드를 새로 생성하는 것에 비해 속도가 빠르다는 것이다. 메모리상에 존재하는 스레드를 그냥 가져오는 것이 당연히 새로운 스레드를 메모리에 할당해서 가져다 쓰는 것보다 빠를 것이다. 앞에서도 언급했지만 스레드는 생성 시간이 결코 짧지 않다.

2.2 동기화(Synchronization)

동기화란, 여러 스레드가 동시에 작업할 때, 각 스레드의 작업 순서를 제어하기 위한 매커니즘이다. 동기화의 목적은 여러 개의 스레드가 하나의 자원을 액세스하려 할 때, 주어진 순간에는 오직 하나의 스레드만이 해당 자원을 액세스할 수 있도록 하는 것이다. 동기화를 사용하여 스레드 실행을 다루는 방법은 두 가지이다. 다음 그림 2.4는 스레드의 동기화를 나타낸다.[1][5][9]

- 메소드 수준에서 코드를 처리한다.(동기화 메소드 사용)
- 블록 수준에서 코드를 처리한다.(동기화 블록 사용)

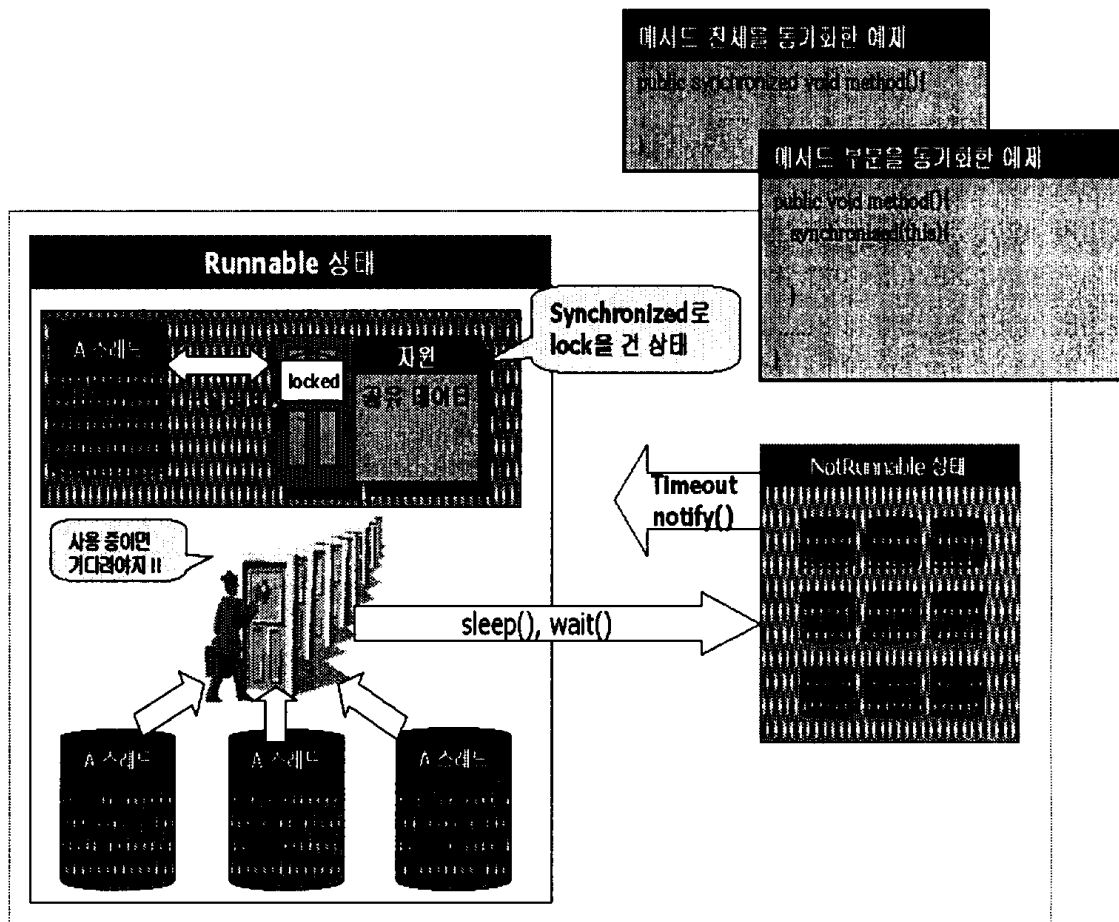


그림 2.4 스레드의 동기화

수행중인 쓰레드는 참조를 갖고 있는 어떤 객체에 대해서도 접근할 수 있다. 만일 두 개의 쓰레드가 같은 객체에 접근하고 각각 객체의 상태를 변경시키는 메소드를 호출할 때 무슨일이 발생할까? 다음 코드는 은행 계좌의 잔고를 예로 설명한 다중 스레드를 나타낸다.

```
int balance;

boolean withdraw(int amount) {
    if(balance - amount >= 0) {
        balance -= amount;
        return true;
    }
    return false;
}
```

일반적으로 위의 코드는 별로 문제가 없는 것처럼 보인다. 여기에서는 잔고(balance)가 양수일 경우에만 인출(with-drawal) 작업이 허용된다. 그러나 두 스레드가 동시에 이 메소드를 수행할 경우에는 큰 문제를 야기시킨다. 만약 1번 스레드가 잔고를 감소시키는 작업을 수행하는 도중에 2번 스레드도 잔고를 줄이는 작업을 수행하려고 한다면, 잔고(balance) 값은 음수가 될 수도 있다. 두 스레드는 하나의 메소드 안에서 매우 빠른 속도로 번갈아 실행된다. 따라서 이러한 위험 요소를 제거하기 위해서 동기화 작업이 반드시 필요하다. 다음 표 2.1은 두 스레드가 하나의 메소드 안에서 빠른 속도로 번갈아 실행됨을 보여주고 있다.

Thread 1	Thread 2	balance
if(15-10 >= 0)		15
	if(15-10 >= 0)	15
balance-=10		5
	balance-=10	-5

표 2.1 하나의 메소드 안에서 스레드의 실행

동기화된 블록이란, 한 순간에 오직 하나의 스레드만이 수행할 수 있는 영역을 말한다. 앞에서 살펴본 스레드 예제에서는 2개의 스레드가 동시에 withdraw() 메소드를 수행했다. 따라서 이 경우 balance 변수의 값을 바꾸는 부분을, 한 번에 하나의 스레드만이 수행할 수 있도록 해주어야만 한다. 자바에서는 코드의 특정 부분을 한 번에 한 스레드만이 접근할 수 있도록 하기 위해서 synchronized 문장을 사용한다.

```
synchronized(anObject) {
    //중요한 코드들
}
```

위와 같은 코드를 작성하면 자바 프로그램이 수행될 때, synchronized (anObject) {} 블록으로 둘러싸인 부분에는 한 번에 하나의 스레드밖에 들어갈 수 없다. 다음 그림 2.5는 시간의 흐름에 따른 동기화 메소드를 나타낸다.[5][9]

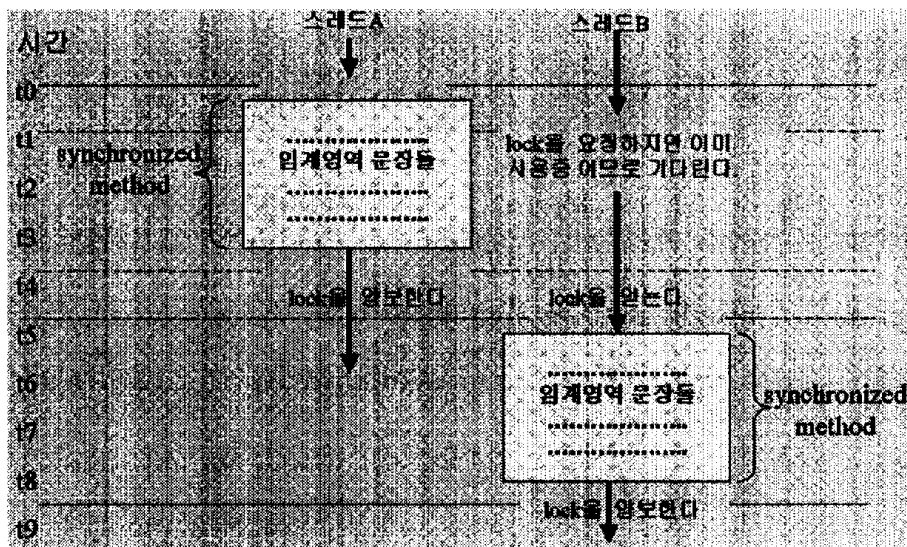


그림 2.5 시간의 흐름에 따른 동기화 메소드

2.2.1 세마포어(semaphores)

자바에서 모든 클래스와 객체는 자신과 관련된 세마포어(semaphore)를 가지고 있다. 세마포어란, 자바 객체가 아니라 특별한 형태의 시스템 객체이며, 세마포어는 실제 사용에 있어서는 프로세스 동기화에 관련하는 하나의 변수의 배열을 말합니다.[9] 가장 간단한 세마포어는 0과 1의 값만 사용하는 이진(바이너리)세마포어이다. 배열을 사용하는 세마포어는 보통 '일반적인' 세마포어라고 말한다. 이 객체는 '얻기(get)'와 '놓기(release)'라는 두 가지 기능을 가지고 있다. 한 순간에, 오직 하나의 스레드만이 세마포어를 얻을 수 있으며(get), 한 스레드가 세마포어를 가지고 있는 동안 세마포어를 얻으려고 시도한 다른 스레드들은 모두 대기 상태에 들어간다. 세마포어를 가지고 있는 스레드가 자신이 가지고 있던 세마포어를 놓으면(release) 대기 상태에 있던 스레드 중 하나가 세마포어를 얻게 되며, 세마포어를 얻으려고 시도했던 나머지 스레드들은 다시 대기 상태로 돌아간다. 이러한 매커니즘을 사용하여 특정 작업을 한 순간에 하나의 스레드(세마포어를 얻은)만이 수행하도록 할 수 있다. 자바에서는 세마포어의 본질과 동기화에 대한 구체적인 내용이 자바 프로그래머로부터 완전히 감추어진다. 따라서 자바 프로그래머가 세마포어의 구조에 대해서 전혀 모르더라도 스레드를 동기화하는 것이 가능하다. 코드의 일정 영역을 특정한 객체나 클래스를 사용하여 동기화시킬 수 있다.[7]

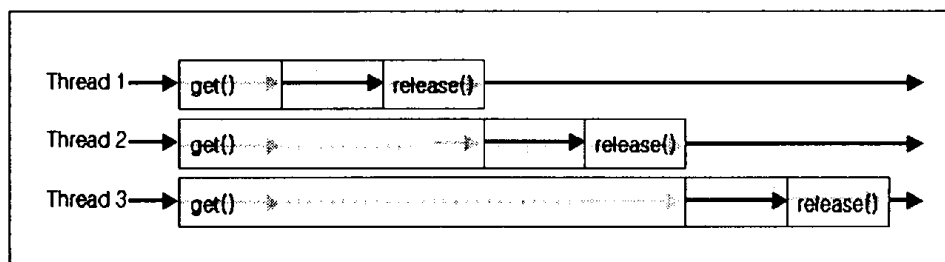


그림 2.6 동기화 블록

스레드가 동기화된 코드 블록에 진입하려고 할 때는 반드시 해당 클래스의 세마포어를 얻어야만 하며, 만약 다른 스레드가 이미 해당 클래스의 세마포어를 얻어야만 하며, 만약 다른 스레드가 이미 해당 클래스의 세마포어를 가지고 있는 경우에는 그 스레드가 세마포어를 놓을 때까지, 코드 영역에 진입하려고 하던 스레드는 블록킹된다. 이런 방식을 사용하면 코드의 특정 영역을 임의의 object 객체에 대한 크리티컬 섹션(critical section)으로 만들 수 있다. 이렇게 하기 위해서는 단지 해당 코드 영역을 `synchronized(object) { }` 블록으로 둘러싸 주기만 하면 된다.[1]

2.2.2 동기화 효율

멀티쓰레딩은 프로그램이 비동기적으로 실행되도록 하기 때문에, 멀티쓰레딩이 필요한 경우에는 강제로 동기화시켜주는 방법이 필요하게 된다. 예를 들어 서로 통신하고 관련된 데이터 구조를 공유하는 두 스레드를 사용하기 원한다면, 그것들이 서로 충돌하지 않는다는 사실을 보증하는 약간의 방법이 필요하다. 이것은 한 스레드가 데이터의 중간 정도를 읽는 순간에 다른 스레드가 그 데이터에 쓰는 것을 방지해야만 한다.

동기화 작업은 오직 하나의 스레드만이 CPU를 사용할 수 있도록 하기 때문에, 객체를 동기화하는 것은 프로그램의 성능을 저하시킬 수도 있다. 반면에, 이미 락(lock)을 얻은 객체에서 또 다시 동기화 작업을 수행하는 것은 거의 비용이 들지 않는다. 새로운 버전의 자바 가상 머신은 좀더 효과적인 동기화 매커니즘을 제공하지만, 프로그램의 효율을 좀더 높이기 위해서는 동기화 작업을 최소로 해야 한다. 물론, 대부분의 경우 동기화 작업은 매우 중요하며, 때로는 절대 생략해서는 안되는 경우도 있다. 그러나 불필요한 동기화 작업은 프로그램의 수행 성능을 떨어뜨리고, 재사용성(reusability)을 감소시키며, 프로그램이 교착 상태(deadlock)에 빠지게 할 수도 있다.

2.3 입출력 스트림

자바 프로그램은 스트림을 통해 입출력을 수행한다. 스트림은 정보를 생산하거나 소비하는 추상화된 개념이다. 스트림은 자바 입출력 시스템을 통해 물리적인 기기와 연결된다. 모든 스트림들은, 그 연결된 실제 기기들이 서로 다르더라도 동일한 방법으로 행동한다. 즉, 동일한 입출력 클래스와 메소드들은 모든 형태의 기기에 적용된다. 이는 디스크 파일, 키보드, 혹은 네트워크 소켓 등 많은 다양한 종류의 입력장치를 입력 스트림이 추상화 한다는 뜻이다. 비슷하게 출력 스트림도 콘솔, 디스크 파일, 네트워크 연결로 간주될 수 있을 것이다. 스트림은 키보드나 네트워크등과 같은 서로 다른 기기들의 차이점을 작성하는 코드의 모든 부분에서 이해하지 않고도 입출력을 다룰 수 있는 명확한 방법이다.

프로그램 상에서 데이터를 입력받기 위해서는 데이터를 제공하는 곳(파일, 메모리, 소켓)의 스트림을 열어야 하며 순차적으로 데이터를 입력받고 읽기가 종료되면 열었던 스트림을 닫는다. 보통 읽기 종료는 더 이상 읽어드릴 데이터가 없을 경우 수행한다.

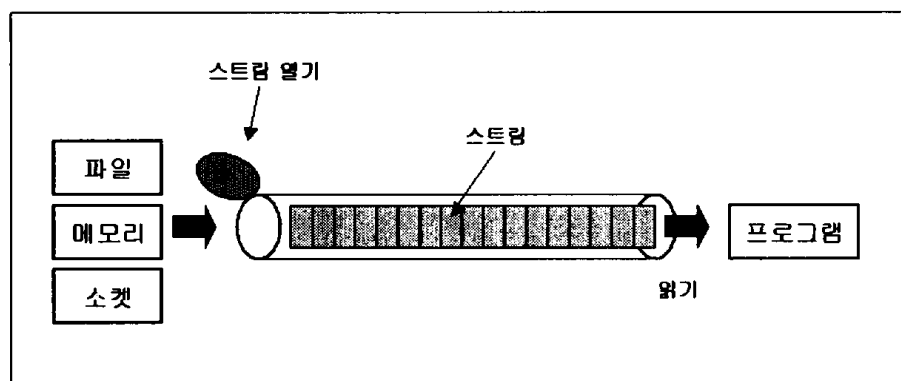


그림 2.7 읽기 스트림

또한 프로그램 상에서 데이터를 출력하기 위해서는 데이터를 쓸 곳(파일, 메모리, 소켓)의 스트림을 열고, 데이터를 그곳으로 보내고 순차적으로 데이터를 쓴 후 쓰기가 종료되면 열었던 스트림을 닫는다. 쓰기 종료는 더 이상 기록할 데이터가 없을 경우 수행한다.

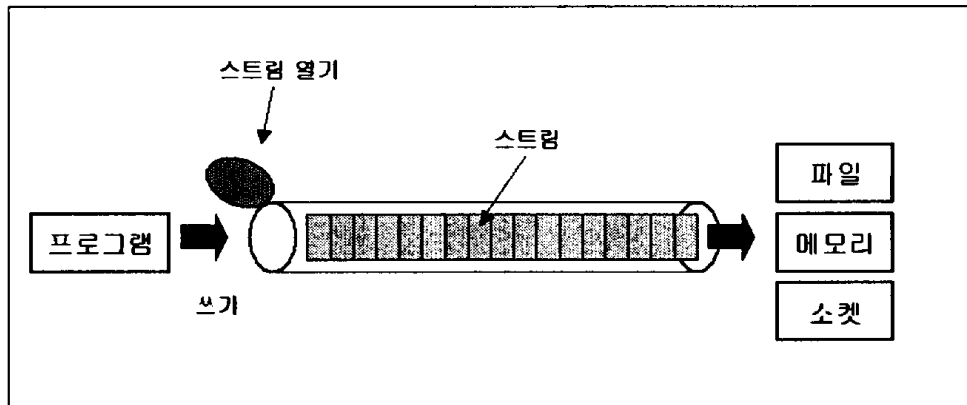


그림 2.8 쓰기 스트림

스트림이란 일련의 연속된 데이터의 흐름이다. 이런 데이터는 어떤 타입이든 상관없다. 데이터의 읽기와 쓰기에 관한 메커니즘들은 java.io 패키지에 포함된 스트림 클래스에 의해 제공된다. 자바 프로그램이 실행을 시작하면 다음과 같은 세 개의 스트림 객체가 자동으로 생성된다.

- System.in : 표준 입력 스트림 객체로서 키보드로부터 바이트를 입력받을 수 있다.
- System.out : 표준 출력 스트림 객체로서 데이터를 화면으로 출력할 수 있다.
- System.err : 에러 메시지를 화면으로 출력할 수 있도록 한다.

2.3.1 문자 스트림과 바이트 스트림

java.io 패키지에는 정보를 읽거나 쓰는데 필요한 많은 스트림 클래스와 메소드들을 정의해 놓고 있는데, 이러한 스트림 클래스들은 흘러 다니는 데이터 형식에 따라 크게 두 가지로 분류할 수 있다. 문자 데이터를 다루는 문자 스트림(character stream)과 바이트 데이터를 다루는 바이트 스트림(byte stream)이 있다.

텍스트 정보들을 읽거나 쓰기 위해서는 문자 스트림을 이용한다. 그 이유는, 자바는 16비트 유니코드에 있는 문자들을 지원하기 때문이다. 반면에 이미지나 사운드와 같은 바이너리 데이터를 다루기 위해서는 바이트 스트림을 이용해야 한다.

바이트 스트림은 2개의 클래스 계층구조를 사용하여 정의된다. 최상위에는 Input-Stream과 OutputStream이라는 두 추상 클래스가 있다. 이들 추상 클래스는 각각 몇 개의 구현된 하위 클래스들이 있어서 디스크 파일, 네트워크 연결, 심지어는 메모리 버퍼와 같은 다양한 기기들의 차이를 제어한다. Input-Stream과 Output-Stream 추상 클래스는 다른 스트림 클래스를 구현하는 핵심 메소드를 정의한다. 이들 중 가장 중요한 두가지는 read()와 write()로 데이터의 바이트값을 읽거나 쓰는 역할을 한다. 양 메소드 모두 추상 메소드로 선언되어 있으며 스트림 클래스들이 이 두 메소드를 오버라이드 한다.

문자 스트림은 2개의 클래스 계층을 사용하여 정의된다. 최상위에는 Reader와 Writer라는 2개의 추상 클래스가 있다. 이들 추상 클래스는 유니코드 문자 스트림을 처리한다. 자바에는 이들 각각에 대해 몇 가지 구체적인 하위클래스들이 있다. 이들 중 가장 중요한 메소드는 read()와 write()이며 데이터의 문자를 읽고 쓰는 역할을 한다. 이들 메소드는

write()이며 데이터의 파생된 스트림 클래스들에서 오버라이드된다.

2.4 JMF(Java Media Framework)

JMF(Java Media Framework)는 썬마이크로 시스템즈와 IBM의 공동연구에 의해 만들어진 자바기반 멀티미디어 프로그래밍을 위한 API이다. 마이크로소프트사의 Video for windows, 또는 멀티미디어 재생, 녹화, 검색, 전송등의 기술을 자바기반으로 만들어 이용하기 위한 도구이다. 그리고 분산 컴퓨팅 환경에서 실시간 멀티미디어의 스트림 합성, 분배, 접합, 분리, 공유, 전송, 입출력, 저장, 동기 기능을 가지고 있다. 또한 분산된 다중 멀티미디어 객체의 시간정렬 및 합성을 통한 멀티미디어 시나리오 프리젠테이션 기능도 포함하고 있다. 그림 2.9는 JMF가 실시간 멀티미디어 환경에서 처리할 수 있는 다양한 기능을 가진 구조를 나타낸다.[8]

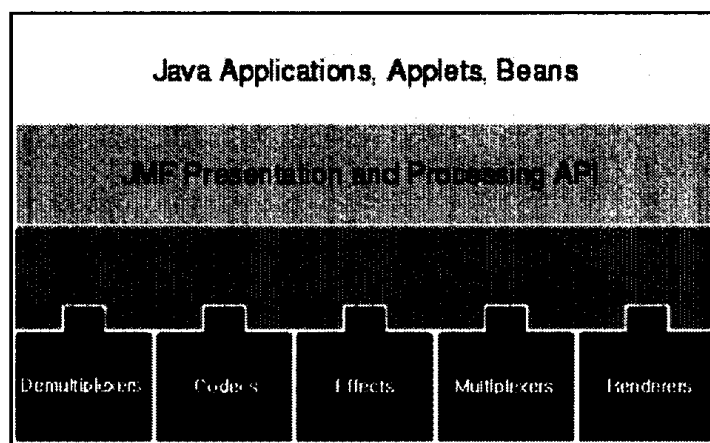


그림 2.9 JMF 구조

JMF는 오디오와 비디오의 타임기반 데이터 표현과 동기를 위한 기능을 가지고 있다. 이러한 기능을 수행하기 위해 3가지의 주요 실행기능을

가지고 있다. 멀티미디어를 실행하기 위한 플레이어(Player), 미디어 저장을 담당하는 캡처(Capture), 이들을 지원하기 위한 API(Conference API)로 구성되어 있다. 이러한 기능들을 통해서 멀티미디어의 압축된 스트리밍 표현, 제어, 동기, 타임기반의 미디어 저장을 자바기반의 플랫폼에서 실시할 수 있다.

JMF 에서 멀티미디어 데이터를 다루기 위해서는 Clock, TimeBase 클래스를 이용한다. 개념적으로는 TimeBaseTime과 MediaTime을 이용한다. TimeBaseTime은 TimeBase가 제공하는 시간으로서 우리가 일시 정지를 시키거나 멈출 수 없는 실제 시간의 흐름을 관리하는 부분이며 Media Time은 미디어 데이터의 시작과 끝을 관리하는 시간이다. 또 한가지 Duration 이라는 것은 MediaTime에서의 경과 시간을 의미한다. 실제로 멀티미디어 데이터가 플레이 될 때 MediaTime은 아래와 같은 식에 의해 계산될 수 있다.[8]

$$\text{MediaTime} = \text{MediaStartTime} + \text{Rate}(\text{TimeBaseTime} - \text{TimeBaseStartTime})$$

TimeBase의 인터페이스를 살펴보면 getNanoSecond() , getTime() 등 현재의 시간을 받는 함수만 제공할뿐, 시간을 멈추거나 정지시키는 함수는 제공하지 않는다.

3. 시스템 분석 및 설계

3.1 시스템의 주요 기능 및 요구사항 분석

본 시스템이 제안하는 강의 저작 도구 스템의 주된 기능은 강의자와 학습자간의 상호 작용을 위한 인터페이스를 제공함으로써 강의자는 별도의 프로그램 기술을 습득하지 않고 쉽게 강의를 제작하고 배포할 수 있는 기능을 제공하며, 학습자는 배포된 강의를 해당 사이트로 부터 다운로드하여 학습할 수 있는 환경을 제공한다. 본 교육시스템은 자바 언어를 기반으로 설계되었다. 따라서 시스템에 독립적인 바이트 코드(byte code)에 의해 어떠한 호스트에서도 실행이 가능하다. 상세한 사항은 다음과 같다.

- 사용자 환경(GUI)는 크게 강의를 제작할 수 있는 모듈과 강의를 재생할 수 있는 모듈로 구성한다.
- 강의자는 강의 노트를 제작하거나 로드 할 수 있는 환경이 제공되어야 한다.
- 강의자는 자신의 강의를 제작하기 위해 판서가 가능해야 한다.
- 강의자가 판서시 다양한 그래픽 툴을 지원해야 한다.
- 강의자가 강의 제작시 음성을 입력 및 저장할 수 있어야 한다.
- 제작된 강의는 배포시 압축 저장 되어야 한다.
- 강의 배포는 웹 사이트를 통해 배포가 가능하여야 한다.
- 배포된 강의의 재생은 다양한 플랫폼에서 지원되어야 한다.
- 강의 재생은 시간과 공간에 제약이 없어야 한다.
- 강의의 재생은 수강자가 쉽게 컨트롤 할수 있어야 한다.

3.2 UML(Unified Modeling Language)을 이용한 시스템 분석

UML(Unified Modeling Language)은 1997년 말에 OMG에서 국제표준으로 채택된 분석/설계단계에서 사용되는 모델링언어이다.[4] UML을 이용하면 소프트웨어 시스템의 모델링결과를 표현력이 강한 시각적 심벌로서 나타낼 수 있기에, 시스템개발에 참여하는 분석가, 개발자, 사용자 등이 함께 이해하고 모델링 결과를 공유할 수 있게된다. 또한, UML은 표현력이 매우 풍부한 언어로서, 그러한 시스템들을 개발하고 배치하는데 필요한 모든 관점을 다룬다.

UML은 여러 가지 다이어그램들을 제시함으로써 소프트웨어 개발과정의 산출물들을 시각적으로 제시하고 표현함으로써 개발 프로젝트를 성공적으로 수행할 가능성이 높아지게 되며, 사용자가 진정으로 원하는 시스템을 개발할 수 있게 된다.

UML이 제공하는 내용을 정리하면 다음과 같이 서술할 수 있다. 첫째, 광범위하고 다양한 모델링 이슈(Issue)들을 표현하기 위한 의미와 표현방법을 제공한다. 사용자의 업무현상을 모델로 표현하는데 있어서 제한이 많이 없게끔 확장성을 다양하게 제공한다.

둘째, 미래의 모델링 이슈들 예를 들면, 컴포넌트 기술, 분산 컴퓨팅, 프레임 워크, 실행성에 관련된 특징을 표현하는 의미를 제공한다.

셋째, 확장 메커니즘(Mechanism)을 이용함으로써 UML 언어에 대한 확장이 가능하다.

넷째, 여러 가지 도구들 사이의 모델교환을 촉진시킴으로써 의미(semantics)를 가진다. Code Generation이나 Document Generation, User Interface Generation을 수행하는 모든 도구(Tool)사이에는 당연히 호환성이 존재한다. 즉 각 도구에서 만들어지는 산출물들을 서로의 도구에서 이

용할 수가 있다.[4][6]

다섯째, 모델 제작물의 저장소(Repository)와 공통으로 사용할 수 있는 저장소에 대한 인터페이스(Interface)를 나타낼 수 있는 의미를 제공한다. 저장소에 대한 인터페이스가 가능해짐으로써 저장소에 들어있는 정보를 공유할 수가 있게 된다.

3.2.1 Use Case 모델링

Use Case는 사용자의 입장에서 바라본 시스템의 특성을 설명하는 것이다. 즉, 사용자들의 시스템에 대한 요구사항들을 정리해 놓은 것이라 할 수 있다. 본 논문에서 제시하는 사용자는 크게 두 부류로 나눌 수 있다. 첫 부류는 강의를 제작 및 배포하는 강의자이고, 두 번째는 강의를 시청 학습하는 학습자이다.

강의자는 강의제작 도구를 실행하여 강의 노트를 만들거나 준비된 강의 자료를 이용하여 음성을 입력하고 강의노트에 판서한다. 또한 제작된 강의를 배포한다. 학습자는 강의자에 의해 배포된 강의를 다운로드하여 강의를 재생 수강이 가능하도록 해야 한다.

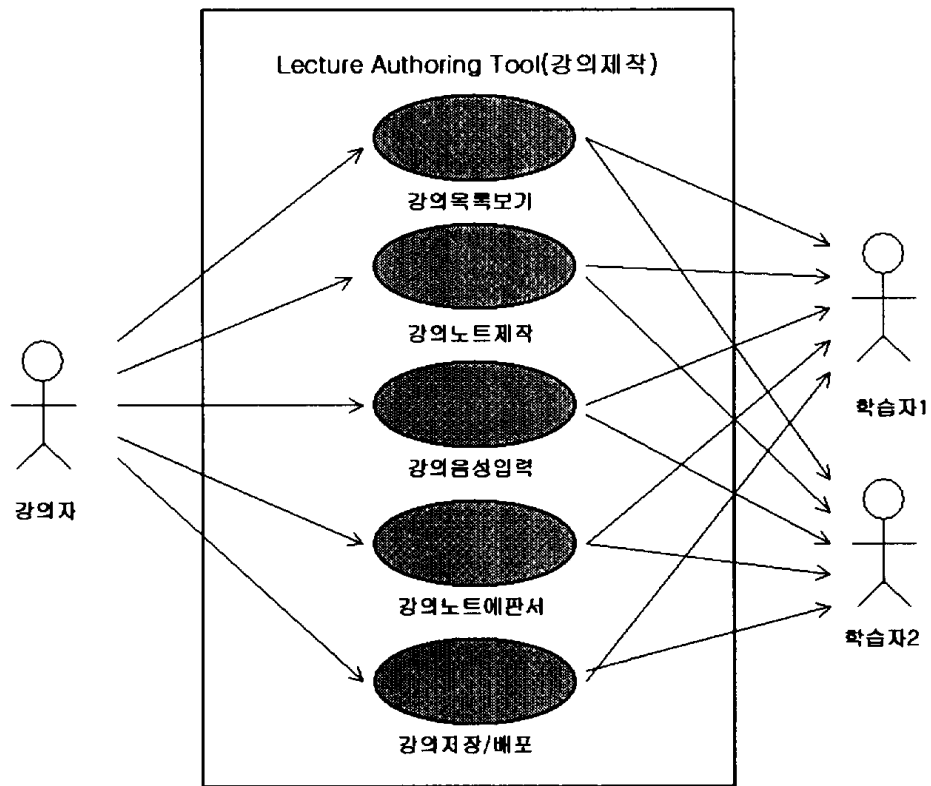


그림 3.1 강의 제작 도구의 Use Case Diagram(강의 제작)

학습자 중심의 Use Case 모델은 강의자 중심의 Use Case 모델과 유사하지만 강의 제작 및 배포 기능이 없고 강의자에 의해 배포된 강의를 다운로드 하여 재생만이 가능하도록 구성 하였다.

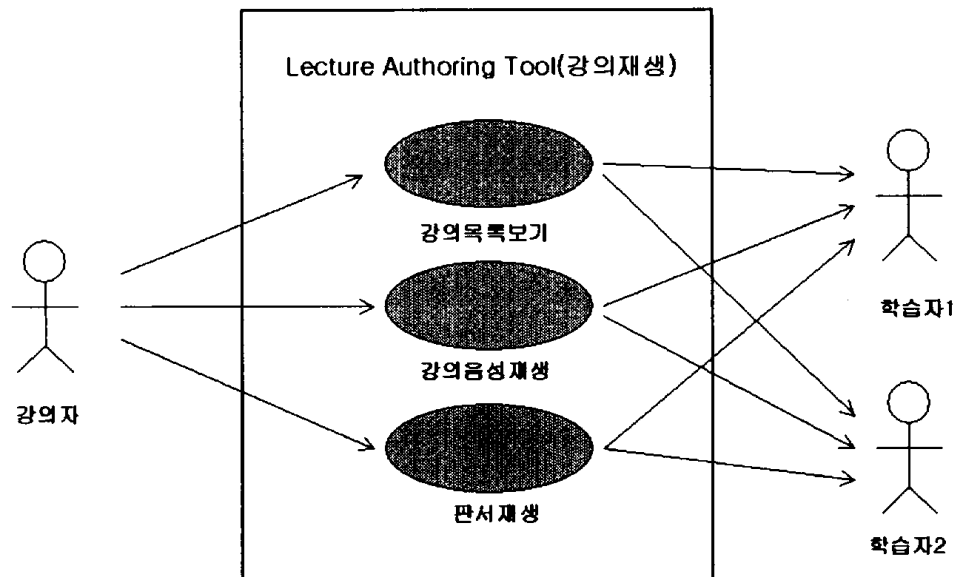


그림 3.2 강의 제작 도구의 Use Case Diagram2(강의 재생)

액 터	설 명
강의자	강의 저작도구를 이용하여 강의를 제작하는 강의자
학습자	제작된 강의를 재생하여 학습하는 학습자

표 3.1 각각 Actor의 역할

3.2.2 Sequence Diagram

- 실행 초기화 시퀀스 다이어그램

Sequence Diagram은 Use Case가 수행되는 절차를 객체간의 메시지로 표현한다. 본 논문에서 제안하는 강의저작 도구 시스템은 강의 저작용 모듈에 재생용 모듈을 포함하므로 핵심이되는 재생용 시퀀스 다이어그램만을 명시하도록 하였으며 그림 3.3은 강의저작용 시스템 실행시 초기화 유스케이스에 대한 시퀀스 다이어그램을 나타내고 있다.

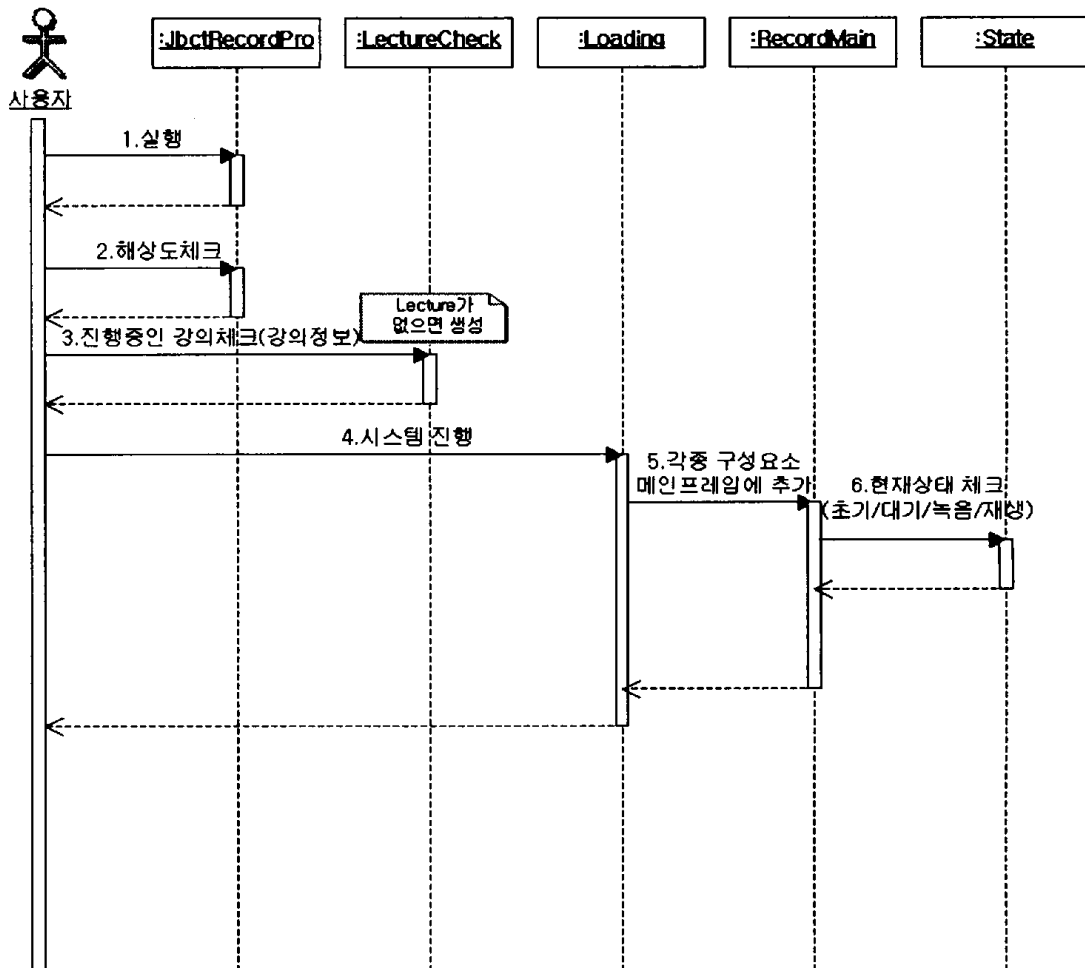


그림 3.3 실행초기화 Sequence Diagram

- 강의 만들기 시퀀스 다이어그램

강의 만들기 시퀀스 다이어그램은 크게 두 가지의 액션으로 구분할 수 있다. 하나는 강의를 처음 생성하기 위해 새로운 강의명과 강의 챕터를 입력해서 생성하는 방법, 두 번째는 이미 만들어진 강의의 챕터를 추가하거나 새로운 강의 노트를 불러와서 생성하는 방법이 있다. 다음 그림 3.4는 강의 만들기 시퀀스 다이어그램을 나타낸다.

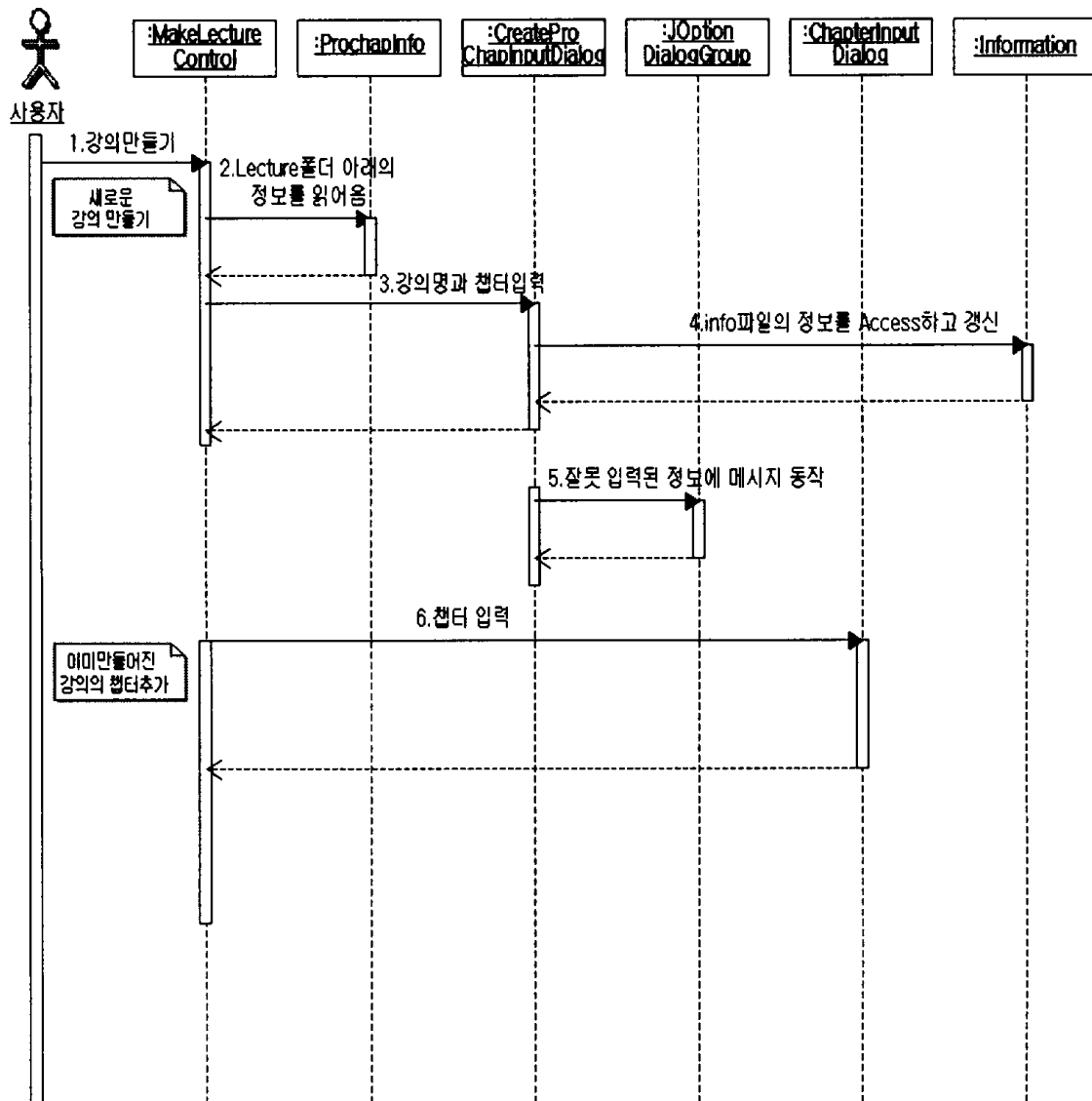


그림 3.4 강의 제작 Sequence Diagram

- 강의노트 편집 시퀀스 다이어그램

강의 노트의 편집 기능은 최초 강의 노트를 캡처하거나 기존 캡처된 이미지를 로드하여 강의 노트 이미지를 편집할 수 있는 기능이 추가되어 있다. 주요 기능으로는 원본이미지 로드, 이미지 확대 및 축소, 이미지 크기의 최적화, 이미지 캡처, 원본이미지 갱신 기능의 각종 이벤트를 포함하고 있다.

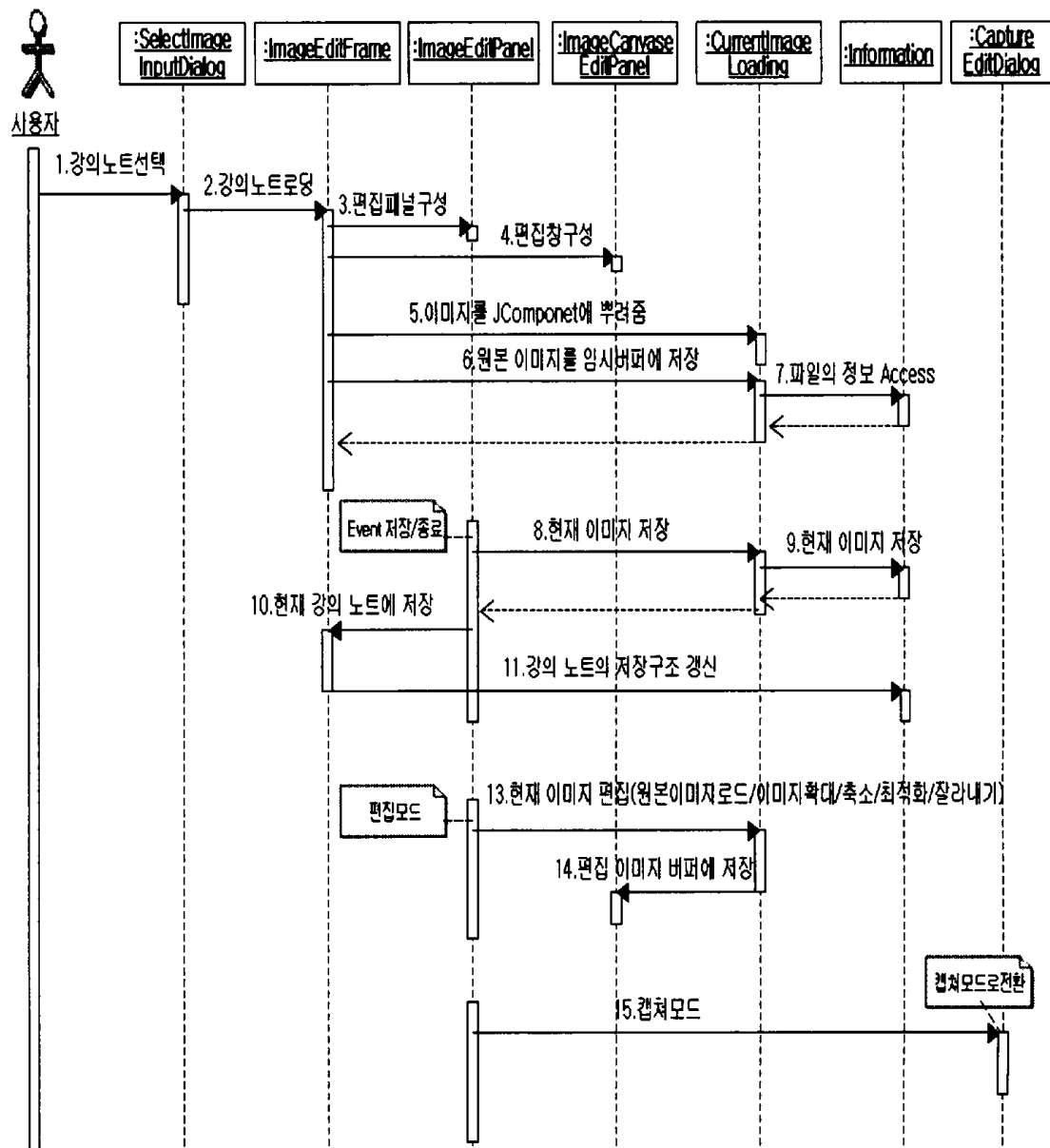


그림 3.5 강의 노트 편집 Sequence Diagram

3.2.3 Class Diagram

Class Diagram은 시스템을 이루는 클래스들과 각 클래스들간의 관계를 나타낸 것이다. 그림3.6은 본 논문에서 제안하는 강의 저작 도구를 사용하여 판서시 주요 클래스들을 클래스 다이어그램으로 나타낸 것이다. 자바 언어는 다중 상속을 지원하지 않기 때문에 추상클래스와 인터페이스를 사용하여 다형성 구조를 표현하는데 본 논문에서 제안하는 구조 또한 추상클래스와 인터페이스를 사용하여 다형성을 표현하였다.

AbstractShape 클래스는 추상 클래스로 선언하였으며 구체적인 작업은 하위 클래스에서 draw 메소드가 호출되어 draw하게 된다. AbstractShape 클래스의 서브 클래스들은 LineFigure, CircleFigure, RectangleFigure, TextFigure, PointFigure 클래스들로 구성하였다. 또한 AbstractShape 클래스의 draw 메소드를 정의한 DrawShape 인터페이스와 저장 구조를 반환하는 ShowInfo 인터페이스를 implements하고 있다.

그리기 개체중 선/원/사각형/텍스트는 마우스가 release되었을 시, 그리기 정보를 저장하고 연필 개체는 모두 이어진 선으로 저장해야 하므로 별도의 PointGroupFigure 클래스를 두어 Vector에 저장하도록 구현 하였다.

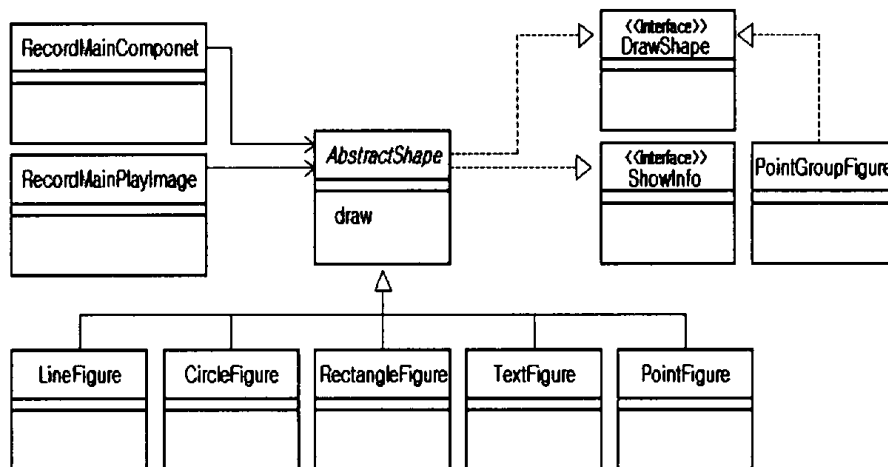


그림 3.6 판서시 Class Diagram

본 논문에서 제안하는 시스템의 저장형태는 아래와 같은 각각의 4개의 파일로 구성 저장된다.

- ① 교재 페이지인 이미지 파일(.jpg)
- ② MIC음성을 저장하는 파일(.wav)
- ③ 판서를 저장하는 파일(.bod)
- ④ 만들어진 강의노트에 음성정보와 판서정보가

포함되어 행위에 대한 동기화를 해 주는 파일(jbct.info)

다음 그림 2.1은 파일 “알기쉬운자바”라는 과목의 “1장 입출력”의 강의 노트들을 작성하고 강의 저작 도구를 이용하여 만들어진 강의파일의 구조를 나타낸다.

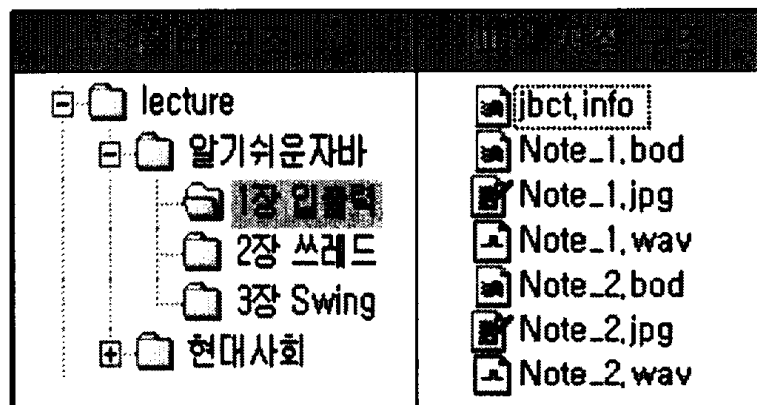


그림 4.2 강의 만들기 폴더 구조

4.2 시스템의 동작

시스템의 각 요소들의 동작 과정은 다음과 같다.

- (1) 강의제작 프로그램을 이용하여 강의 과목과 챕터를 생성한다.
- (2) 강의 저작 도구를 이용하여 강의 자료를 캡처하고 챕터의 페이지별 이미지를 편집하여 강의노트 저장한다.
- (3) 위에서 만든 강의노트 위에, 강의제작 프로그램을 이용해서 마이크로 음성을 녹음하거나 마우스 또는 디지털타저를 사용하여 판서한다.
- (4) 산출물은 jbct.info 파일(강의제작 컨트롤 파일) 하나에 강의노트 하나당 한 개씩의 강의노트(.jpg)파일, 판서정보파일(.bod), 음성녹음파일(.wav)이 들어간다.
- (5) 강의자는 위에서 작성된 파일을 zip파일로 압축해서 학습자에게 배포한다.
- (6) 학습자는 배포된 파일을 다운로드하여 재생용 프로그램을 이용해서 학습을 한다. 일시정지시 판서시간을 정지하고, 음성 또한 음성 데이터라인을 정지시킨다.

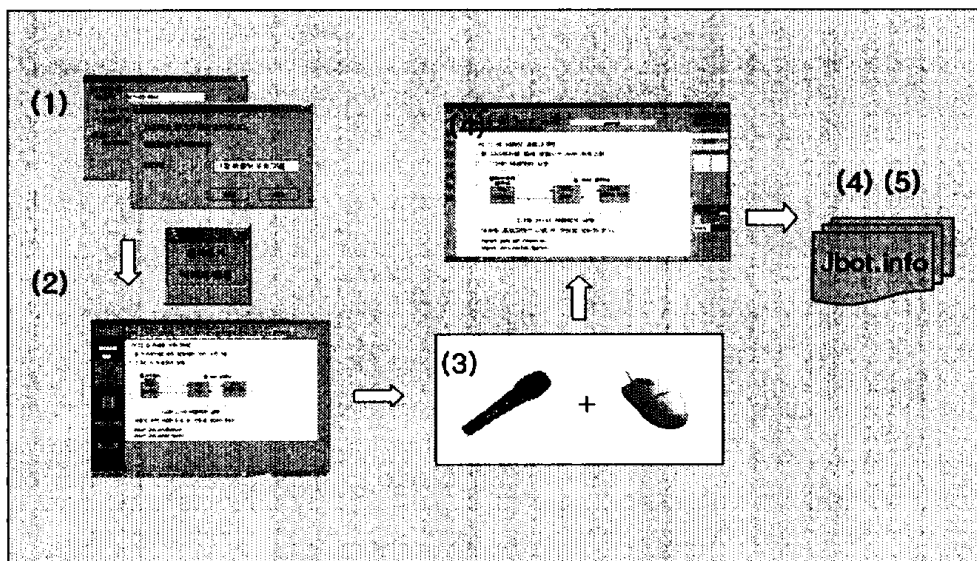


그림 4.3 시스템의 동작

4.2.1 시스템의 초기화

강의자가 강의를 강의를 제작하기 위해 강의 저작 도구 시스템을 실행시키면 화면 해상도를 체크하고 설정된 해상도에 따라 경고 메시지를 띄우고 종료하거나 실행창을 열게 된다. 다음 그림은 초기 실행시 해상도를 체크하고 시스템을 로딩하는 화면을 나타낸다.



그림 4.4 초기 실행 화면

최초 강의 저작 도구의 실행시 저장된 강의가 없기 때문에 설치된 디렉토리에 lecture폴더를 생성하고 로딩창이 실행되는 동안 강의 저작 시스템의 메인 폼에 붙는 Panel이 포함될 때 마다 Progress바를 진행시킨다. Progress바가 100% 진행후 메인창을 띄워 초기화 시킨다.

다음 그림은 강의저작 시스템을 실행시켜 초기화 된 상태를 나타낸다.



그림 4.5 강의 저작 도구 초기화 상태

4.2.2 강의 만들기

새로운 강의를 만들거나 기존의 만들어진 강의에 강의노트를 추가, 혹은 같은 강의 과목에 새로운 챕터를 생성하고 강의를 녹음할 수 있는 기능이다. 강의 노트를 만드는 방법은 두 가지를 제공하는데 한가지는 현재 폴더의 이미지를 가져와서 강의를 녹음 및 판서하는 방법이며, 두 번째는 강의 저작 도구 시스템에서 화면 캡처를 통해서 강의 노트를 만들고 녹음 및 판서하는 방법이다. 다음 그림은 강의를 “알기쉬운 자바”라는 과목으로 만들어진 강의를 실행후 챕터의 두 번째 강의노트 화면에 판서와 음성을 녹음하는 상태를 나타낸다.

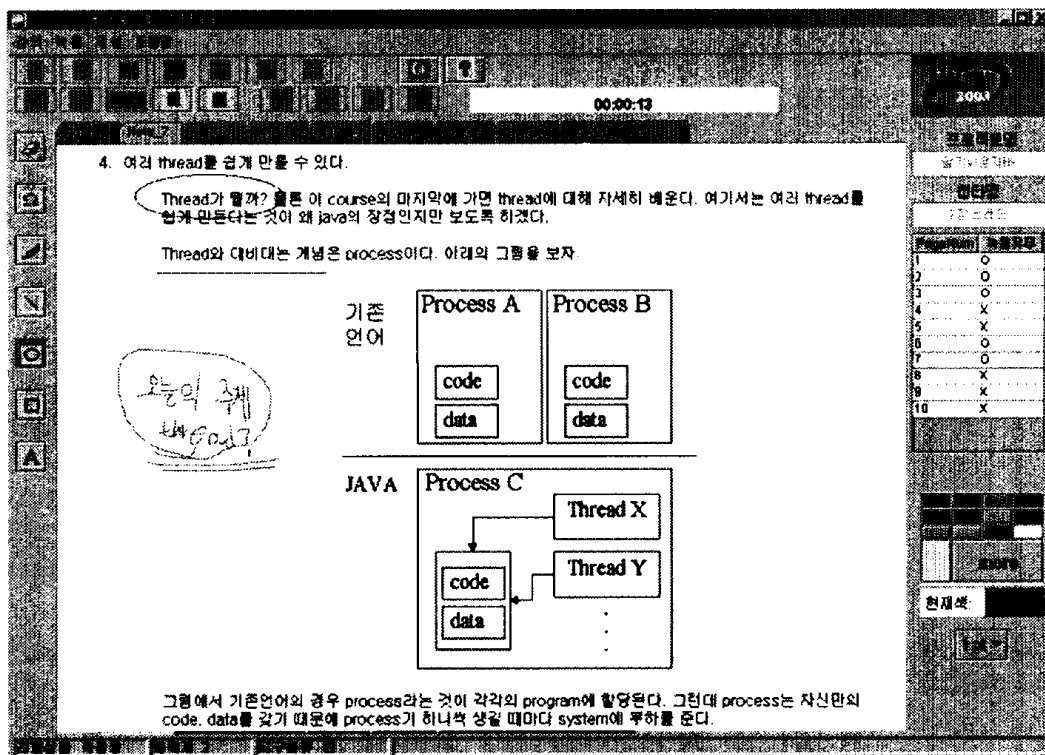


그림 4.6 강의 저작 도구를 이용한 판서

4.2.3 강의 노트 편집

강의 노트 편집 메뉴는 강의를 만들기 위해 한글97이나 파워포인트 등 별도로 만들어진 문서를 로드 하거나 인터넷 브라우저에서 불러온 웹문서를 로드하여 캡처 작업을 거쳐 강의 노트를 생성하게 된다. 기존의 가상 강좌 도구들은 별도의 강의노트를 작성해야 하는 문제점이 있었으나 본 논문에서 제안하는 강의 저작 도구 시스템은 화면을 캡처하는 방식으로 강의 노트를 작성하기 때문에 이기종 문서간의 호환성을 극복하였으며 강의제작시 강의자는 본인이 작업하기 쉬운 프로그램을 사용하여 강의 노트로 활용할 수 있다. 그림 4.7은 여러 문서를 활용한 강의 노트 작성을 나타낸다.

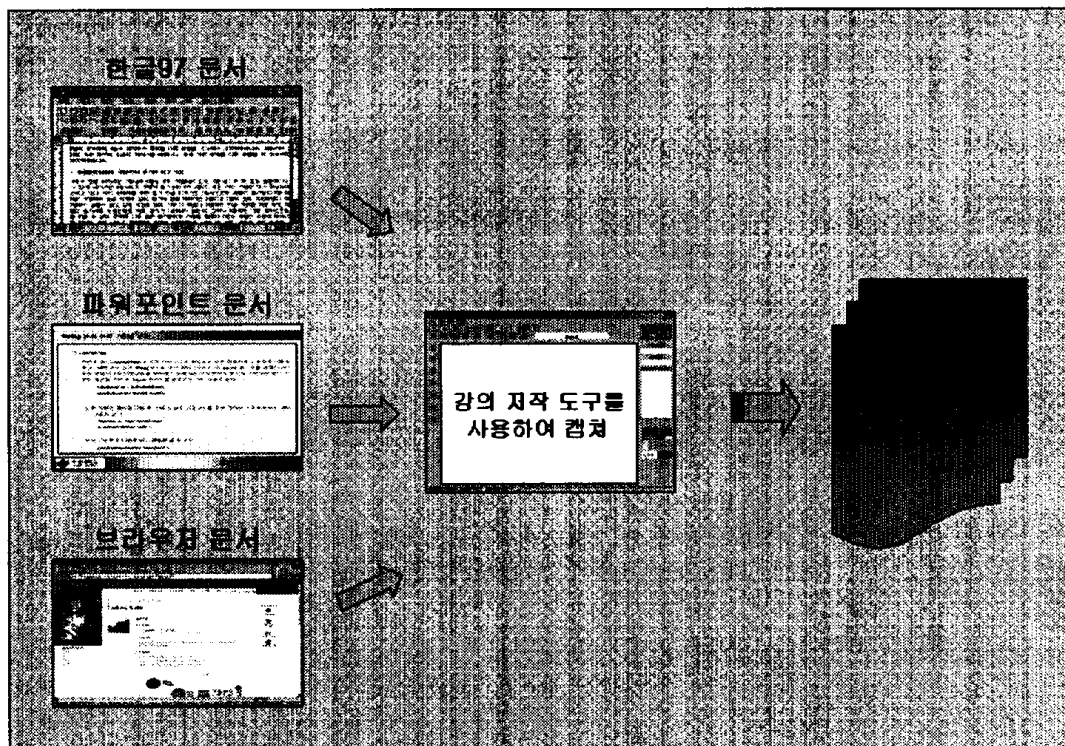


그림 4.7 강의 노트 활용

4.3 강의노트 이미지에 쓰기

강의제작을 위한 캔버스의 구조는 강의노트 이미지 위에 투명 판넬을 넣고 투명 판넬에 판서정보(연필/선/사각형/원/텍스트)를 그리는 방식이다. 하지만 투명 판넬에 계속 그림을 그릴 경우 Repaint시 느려지는 현상이 벌어지고 또한 그려지는 판서정보가 끊어지는 현상이 벌어진다. 그러므로 도형을 그리고 나서 마우스 Release가 되었을 시 해당 도형을 투명 판넬에 그리는 것이 아니라 이미지에 그려, 투명 판넬에서의 부하를 줄였다. 단, 선/원/사각형/텍스트 그리기의 경우만 마우스가 release되었을 시에 강의노트 이미지로 해당 판서정보가 이동한다.

연필 그리기의 경우에는 마우스를 떼지 않고 계속 그리기를 할 경우, repaint하는 정보가 많으므로, 투명캔버스에서 끊어져서 그려지거나 느려지는 결과를 가져온다. 그러므로 연필 그리기 시에는 투명 캔버스에서 마우스가 release가 되었을 시가 아니라 drag를 하였을 시, 그 때마다 강의노트 이미지로 판서정보가 이동하여 그려진다. 그림 3.3은 강의노트 이미지에 쓰기를 할 때 도형이 그려지는 상태를 나타낸다.

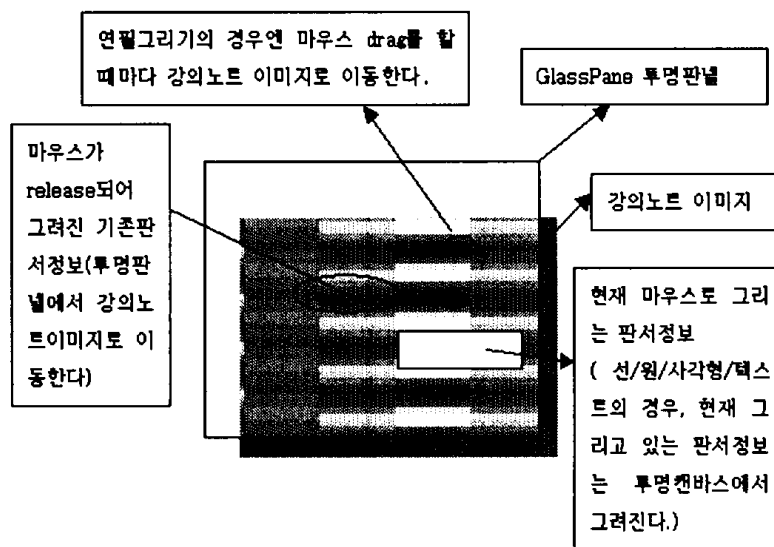


그림 4.8 강의 노트에 쓰기

4.4 판서시 판서정보 저장

강의 저작 도구 시스템은 강의를 저작시 그림 도구를 이용하여 판서를 하게되는데 본 시스템에서는 강의 저작시 성능을 향상시킬 수 있는 방법으로 Clipping 방식을 이용하여 판서를 저장하게 된다.

Repaint시 판서정보(연필/선/원/텍스트/사각형 그린 정보)만 그리는 것이 아니라 판넬의 모든 영역을 그려버리기 때문에 시스템 성능을 약화시킨다. 이를 해결하기 위해 해당영역만을 repaint시키는 Clipping 방법을 사용하였다.

Clipping이란 컴퓨터그래픽에서 사용하는 용어로 자르기, 컴퓨터 그래픽에서 화면 표시 장치의 경계 밖의 이미지 부분을 제거하는 과정을 나타내며, 이는 세계 좌표계 상에서의 윈도우에 대해 행해질 수도 있고, 장치 좌표계 상에서의 뷰포트에 대해 이루어질 수도 있다.

위에서 말한 바와 같이 판서정보가 사용자에게 의해서 그려지고 repaint 되면 투명 판넬에서 이미지로 이동하여 그려진다. 즉, Clipping은 사용자의 마우스가 release되었을 시 이미지에 그려지면서 repaint시 자신의 해당영역만을 그리게 된다. 다음은 LineFigure Class 에서 Clipping 구조를 나타낸다.

```
public void draw(Graphics2D g2d, Color color,int lineDepth)
{
    Line2D line = new Line2D.Float( x1, y1, x2, y2 );
    g2d.setPaint(color);
    g2d.setStroke(new BasicStroke(lineDepth));
    itempX1 = x1;
    itempX2 = x2;
    itempY1 = y1;
    itempY2 = y2;
```

(다음페이지↓)

```

if(itempX1>itempX2)
{
    int itemp = itempX1;
    itempX1 = itempX2;
    itempX2 = itemp;
}
if(itempY1>itempY2)
{
    int itemp = itempY1;
    itempY1 = itempY2;
    itempY2 = itemp;
}
Rectangle rectBounds =
    new Rectangle(itempX1-2, itempY1-2, itempX2-itempX1+4, itempY2-itempY1+4);
g2d.setClip(rectBounds);
//Clipping 영역을 지정
g2d.draw(line);
this.color = color;
this.lineDepth = lineDepth;
}

```

4.5 객체 저장 방식의 사용

저장방식에는 여러 가지가 있으나 본 논문에서는 객체 저장방식을 사용하였다. 객체로 저장시에는 아직 할당받지 않은 ID를 생성하고 이 정보를 읽을 수 있도록 하였으며 판서정보 저장시 상태코드와 시간에 따라 좌표, 색상, 선두께에 관한 정보가 저장되게 된다. 다음은 판서정보를 객체로 저장하는 클래스를 나타낸다.

```
public class RecordStruct implements Serializable{
    static final long serialVersionUID = -2454869208241388472L;
    public int stateCode; // 상태코드
    /*
        * stateCode
        * 0. 아무것도 선택하지 않음
        * 1. 마우스 Pressed 시 (마우스 클릭시, 드래그시)
        * 2. 마우스 Released 시(마우스 릴리즈시)
        * 3. All Clear 모두 지우기
        * 4. Undo 되돌리기
        * 5. 연필 그리기 선택시
        * 6. 선 그리기 선택시
        * 7. 원 그리기 선택시
        * 8. 사각형 그리기 선택시
        * 9. 텍스트 넣기 선택시
    */
    public long recordedTime; //녹음시간
    // 좌표
    public int x1, y1, x2, y2;
    // 색상
    public int red, green, blue;
    // 선두께
    public int dept;
    public String data;
    기타 매소드...
}
```

4.6 저장 시간의 차별화

본 논문에서 제안하는 강의 저작도구를 이용하여 판서시 객체를 저장하는 시기를 결정하는 방법은 프로세서의 성능에 가장 큰 문제점으로 부각되었다. 판서를 할 경우 모든 판서 도구의 저장하는 시기를 동일하게 설정하였을 경우 불필요한 정보들이 저장되어 프로세서의 성능을 저하시키는 원인으로 작용하였으며 이를 해결하기 위해 객체가 저장되는 시간을 차별화 하여 CPU점유율을 낮추고 성능을 향상시켰다.

- 선/원/사각형/텍스트의 경우 초당 10프레임(100ms)
- 연필 툴의 경우 초당 20프레임(50ms)

```
//선,원,사각형,텍스트 그리기의 Timer
public class DrawTimer implements ActionListener{
    Timer timerClock ;
    private final static int timeRepaint = 100;
    public DrawTimer
        (RecordMainComponent recordMainComponent) {
        timerClock = new Timer( timeRepaint, this);
        this.recordMainComponent = recordMainComponent;
    }
    public void start() { //Timer의 시작
        timerClock.start();
    }
    public void stop() { //Timer의 종료
        timerClock.stop();
    }
    public void actionPerformed(ActionEvent ev) {
        if( recordMainComponent.isMousePress == true ) {
            recordMainComponent.repaint();
        }
    }
}
```

//연필 그리기의 Timer

```
public class PointTimer implements ActionListener{
    Timer timerClock ;
    private final static int timeRepaint = 50;
    private RecordMainComponent recordMainComponent;

    public PointTimer(RecordMainComponent
recordMainComponent){
        timerClock = new Timer( timeRepaint, this);
        this.recordMainComponent = recordMainComponent;
    }
    public void start() {
        timerClock.start();
    }
    public void stop() {
        timerClock.stop();
    }
    public void actionPerformed(ActionEvent ev) {
        if( recordMainComponent.isMousePress == true ) {
            recordMainComponent.pointDraw();
            recordMainComponent.repaint();
        }
    }
}
```

5. 결과 및 분석

본 논문의 시스템과 다른 시스템과의 비교는 서비스방식에서의 비교와 도구적 측면에서 평가한다. 현재의 원격강의 도구는 표준화 방식이 제시되고 있진 않은 시점에서, 직접적인 비교는 어렵다. 그래서 본 논문에서는 원격교육을 제공하는 방식간의 장단점 비교를 통해서 본 시스템을 비교 평가하고자 한다.

방식 \ 비교	장 점	단 점
웹브라우저 방식	1. 웹 기반의 편리한 사용자 인터페이스	1. 멀티미디어 이용 한계 2. 웹 관련 프로그램 기술 필요
플러그인 방식	1. 폭넓은 멀티미디어 학습자료 사용 2. 웹 상에서 특정 포맷 형식의 자료를 다운, 플러그인을 통하여 서비스	1. 부가적인 프로그램 설치 필요 2. 플랫폼 의존적 개발 부담
본 논문 제안 방식 (웹브라우저 방식 + 플러그인 방식)	1. 웹 기반의 편리한 인터페이스 2. 다양한 멀티미디어 학습자료 사용 4. 플랫폼 의존적 개발 부담 없음 5. 기존 강의 자료 이용	1. 부가적인 프로그램 설치 필요 (JVM)

표 5.1 도구적 측면 비교 분석

표 5.1은 원격교육의 도구적 측면에서 웹브라우저 방식과 웹브라우저와 플러그인 방식, 그리고 본 논문에서 제시하는 방식간의 장단점을 비교 분석한 도표이다.

원격강의를 제공하는 방법에는 도구 측면과 이를 서비스하는 서비스방법을 통해서 비교한다. 먼저 인터넷상에서 웹브라우저의 사용은 많은 장점을 내포하고 있는 반면, 웹브라우저와 플러그인 형식의 도구인 경우 나름대로의 장단점도 가지고 있으며, 단점에 대한 문제점 해결도 충분히 극복할 수 있다. 본 논문에서 제시하는 시스템은 2가지 방식의 장점을 대부분 수용하고 있다.

원격강의를 제공하는 서비스 방식에는 면대면 효과 높은 실시간 방식인 RT(Realtime Tele teaching)방식과 자유로운 학습이 가능한 NRT(Non Realtime Tele teaching)방식이 있다.

본 논문에서 제안하는 방식은 RT방식의 실시간 통신 기능과 NRT방식의 셀프 학습의 장점을 수용하고 있다. 표 5.2 는 RT방식과 NRT방식 그리고 본 논문에서 제안하는 시스템간의 장단점을 비교한 도표이다.

방식 \ 비교	장 점	단 점
RT방식	1. 면대면 효과 높음 2. 통신상에서 대용량자료 전달 기능 3. 실시간 통신가능	1. 통신/서버 유지부담 가중
NRT방식	1. 셀프 학습, 통신/서버 유지부담 최소 2. 많은 멀티미디어 학습자료 사용	1. 통신상에서 대용량자료 전달 부담 2. 면대면 효과 낮음
본 논문 제안 방식	1. 셀프학습 2. 통신상에서 대용량자료 전달 가능 3. 통신/서버 유지부담 최소 4. 많은 멀티미디어 학습자료 사용	1. 면대면 효과 낮음

표 5.2 서비스 측면 비교 분석

저장 스트림 방식은 NRT방식에서 주로 사용하며, 라이브 스트림방식은 RT방식에서 주로 사용하는 전송 방식이다. 본 논문에서 제안하는 방식은 저장 스트림 방식을 이용한 원격교육 방식을 채택하고 있다. 표3은 원격 강의 데이터의 실시간 전송방식의 비교를 나타낸다.

방식 \ 비교	특 징
저장 스트림	실시간 데이터를 서버에 저장(RT방식)
라이브 스트림	실시간 데이터가 동적으로 발생(NRT방식)
본 논문의 제안 방식	실시간 데이터를 서버에 저장

표 5.3 데이터 전송 방법의 비교

6. 결론 및 향후 과제

인터넷의 등장과 통신기술의 발달로 인해 여러 분야에서 원격 교육에 관한 연구가 활발하게 진행되어 왔다. 또한 멀티미디어 기술을 활용한 콘텐츠의 저작이 일반화되고 있다. 이러한 콘텐츠는 전통적인 교실 수업과 같은 학습 효과를 얻기 위해 멀티미디어 콘텐츠를 도입하는 노력을 기울이고 있지만 아직 기능상에 많은 제약이 있다.

현재 원격교육은 학생과 강사들이 물리적으로 함께 모이는 학교 캠퍼스의 시간적, 공간적 제약을 극복하기 위하여 가상환경에서만 교육이 이루어지는 가상학교 설립이 목적이다.

본 논문은 최종적으로 가상대학의 설립을 목표로 하는 연구의 일환으로써, 원격교육에 최대 교육 효과를 볼 수 있는 멀티미디어 원격강의 저작 도구를 제안하고 구현하여 분석하였다.

기존의 교육시스템들은 강의자가 프로그램 제반 기술을 익혀 강의를 제작해야 하는 많은 단점을 가지고 있었다. 본 논문에서는 이러한 단점을 해결하기 위해 JVM 환경 하에서 구현된 강의 저작도구 시스템을 제안하였다.

본 연구 결과 인터넷상에서 본 논문이 제시하는 교육 시스템을 사용하면 플랫폼에 상관없이 언제 어디서나 웹을 통해서 쉽게 원격강의에 접할 수 있으며 실행 환경의 범용성을 보장한다. 또한 앞으로 다가올 가상 교육 시대에 큰 도움을 줄 수 있다.

그러나, 본 논문은 원격교육의 기본적인 기능만을 구현하였기 때문에, 실질적인 교육적 효과를 위해서는 추가 보완해야 할 사항이 많이 노출되었다.

첫째, 유닉스 및 PDA와 같은 모바일(Mobile) 시스템에서도 강의 재생이 가능하도록 기능을 추가하여 시각적 효과와 더불어 멀티미디어 환경에 부합하게끔 기능을 추가 시켜야 한다.

둘째, 위에서 제시된 각 모듈의 인터페이스를 웹으로 통합해 시스템 관리가 용이함과 동시에 사용자 인터페이스를 보다 쉽고 간편하게 구성해야 한다.

셋째, 강의자와 학습자간의 대화를 통해 상호간의 커뮤니케이션 활동과 적절한 시기의 피드백이 가능하도록 이에 대한 연구가 계속 되어야 할 것이다. 또한 본 연구의 기본적인 기능 외에 다른 기능을 추가로 사용할 수 있게끔 한다면 보다 유용한 원격 교육 시스템으로 사용 될 수 있다고 생각한다.

참고문헌

- [1] 김일민, 알기 쉬운 중급 자바, 홍릉 출판사, 2001.
- [2] Joshua Bloch, Effective Java Programming Language Guid,
Daewoong Media, 2003.
- [3] Peter Hanggar, Practical Java Programming Language Guide,
Addison-Wesley, 2000.
- [4] Paul R. Reed, Developing Applications with JAVA and UML,
Scitech Media, 2003.
- [5] Merlin Hughes와 3인, JAVA NetWorking, Manning, 2000.
- [6] Hans-Erik Eriksson and Magnus Penker, UML ToolKit,
Wiley Computer Publishing, 2000.
- [7] Alejandro Terrazas/John Ostuni/Michael Barlow, Java Media APIs,
SAMS, 2002.
- [8] Sun Microsystems, Java Media Framework Programmer/s Guide,
<http://java.sun.com/products/java-media/jmf/2.1.1/>, 2003.
- [9] Cornell. Horstmann, Core Java 2, England and Korea publishing
Company, 2001.
- [10] 황대준 외 6인, “21세기형 첨단학교·가상대학 설립운영에 관한 연
구”, 교육부 교육정책과제 연구보고서. 1997.
- [11] 임경철, “인트라넷을 활용한 멀티미디어 원격학습시스템(Distance
Learning System)구현”, 한국정보처리학회, 정보처리 제4권 제3호,
1997.
- [12] 박재윤, “교육정보화를 위한 법제개선연구, 교육부 교육정책과제
연구보고서, 1997.

ABSTRACT

Design and Implementation of a Lecture Authoring Tool using Java Language

Jang, Eun-Seop
Computer New Technology
of the Graduated School
Hansung University
advised by Kim, il-min

With the development of internet and communication technology, remote education has been actively in practiced. As a result, remote education contents, which take advantage of various multimedia technologies, are commonly deployed.

However, many existing remote education systems, which exploited multimedia technology, have functional problems. Since a lecturer in remote education should have some knowledge of programming skill and computer science, it is difficult for professors to open classes of remote education.

In this paper, we designed and implemented a lecture-authoring tool that solves the previous problems. With this tool, lecturers who have no experience of programming and knowledge of computer science may be able to build remote lectures easily. Since this lecture-authoring tool is implemented on JVM(Java Virtual Machine),

it is not influenced by heterogeneous computing environments.

The object of the system, which is designed and implemented in this paper, is to embody an authoring tool for lectures, and replay the lectures for students to study individually.

[부록 1] 강의 저작 도구 시스템의 사용자 메뉴

본 논문에서 제안하는 강의 저작 도구 시스템은 가상대학의 설립을 목표로 하는 연구의 일환으로써, 원격교육에 관한 정보를 공유하고 더 나아가 발전된 원격교육을 활성화하는 방안으로 추가 보완되어야 한다. 본 논문에서 제시하는 자료를 다음 사이트에 게시하였다.

[<http://webmail.hansung.ac.kr/~ikim/cbt/>]

• 강의 저작 도구를 이용한 강의 저작

(1) 강의 저작 도구 실행

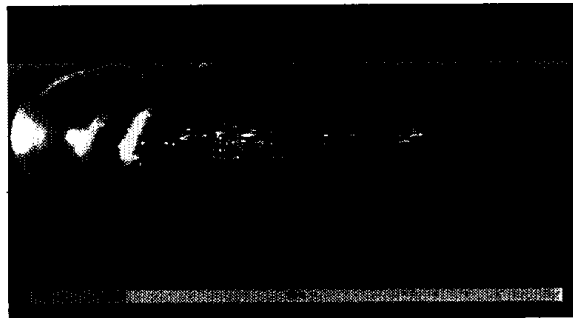


그림 4.4 초기 실행 화면

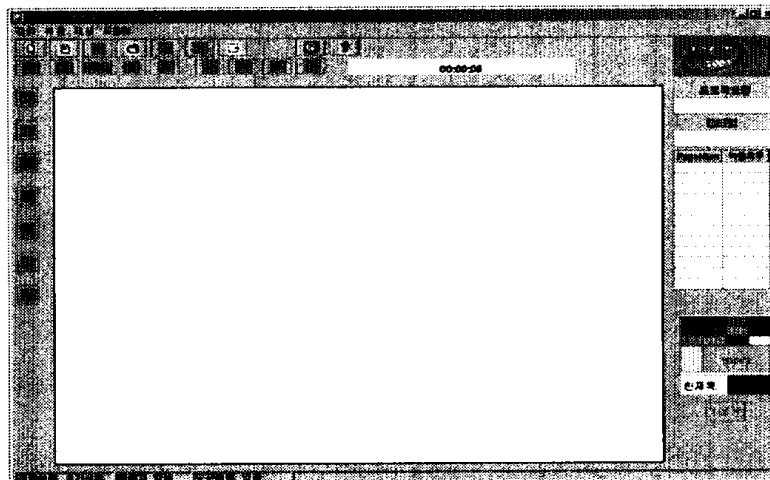
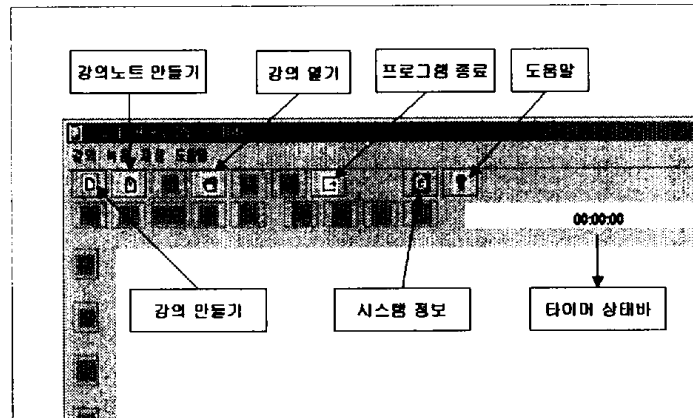
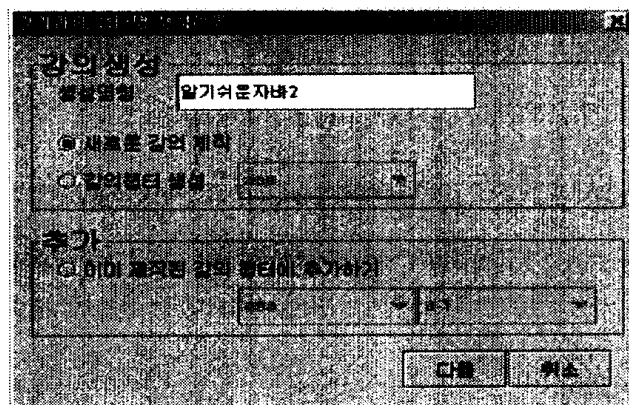


그림 4.5 강의 저작 도구 초기화 상태

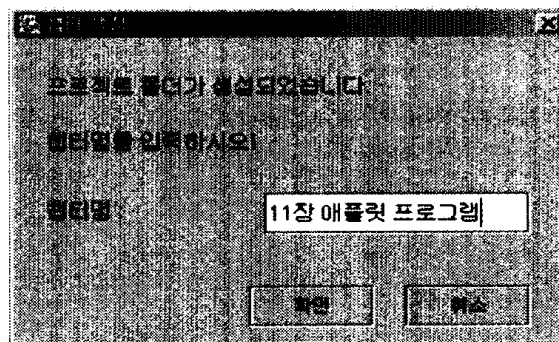
(2) 초기 메뉴 설명



(3) 강의 목록 및 챕터 생성하기

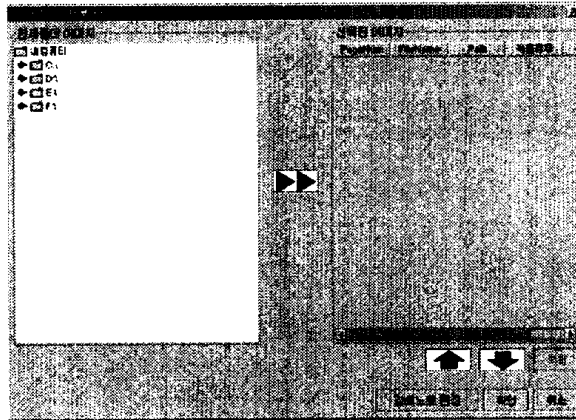


(4) 강의 노트 만들기

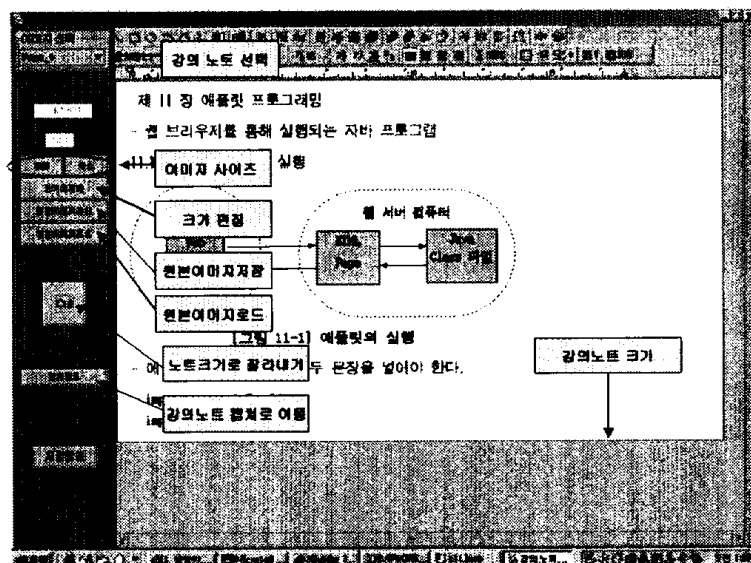


(5) 강의 노트 챕터의 각 페이지 편집

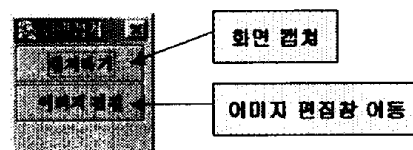
① 강의 노트 이미지 불러오기 윈도우에서 “강의노트편집”을 선택



② 강의 노트 편집 윈도우에서 캡처모드 선택

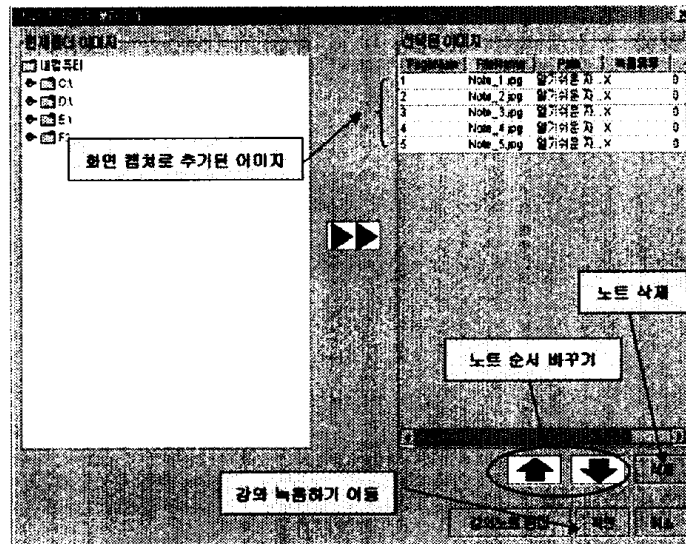


③ 강의 노트 캡처

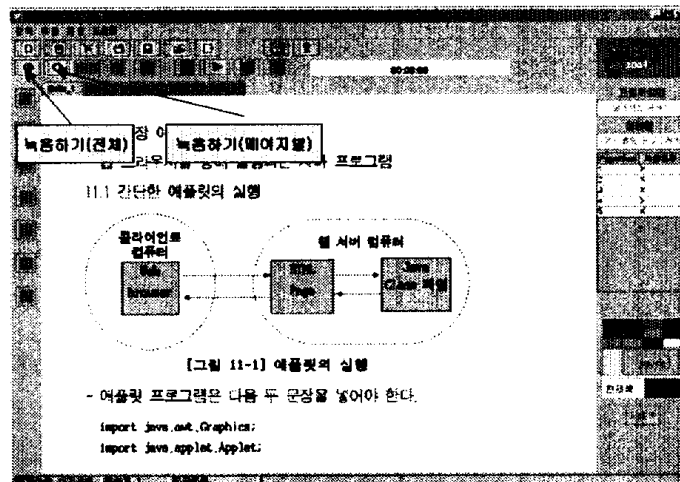


(5) 강의 노트에 음성 녹음 및 판서

- ① 이미지가 추가된 강의노트 이미지 불러오기 윈도우에서 “확인” 버튼을 클릭하고 녹음하기 윈도우로 이동한다.



- ② 추가된 강의노트 이미지에 판서와 음성 녹음을 시작한다.



(6) 강의 노트 저장 / 배포

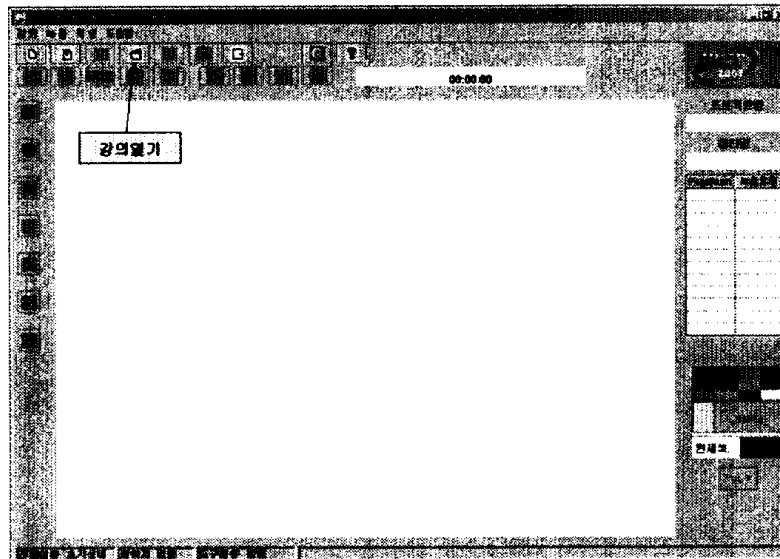
- ① 저장하기 메뉴를 선택하여 모든 강의를 저장
- ② 만들어진 강의를 “압축 저장하기”를 선택하여 압축 저장하고 배포하여 학습하도록 한다.

- 강의 저작 도구를 이용한 강의 재생 및 학습

- (1) 강의 저작 도구 실행

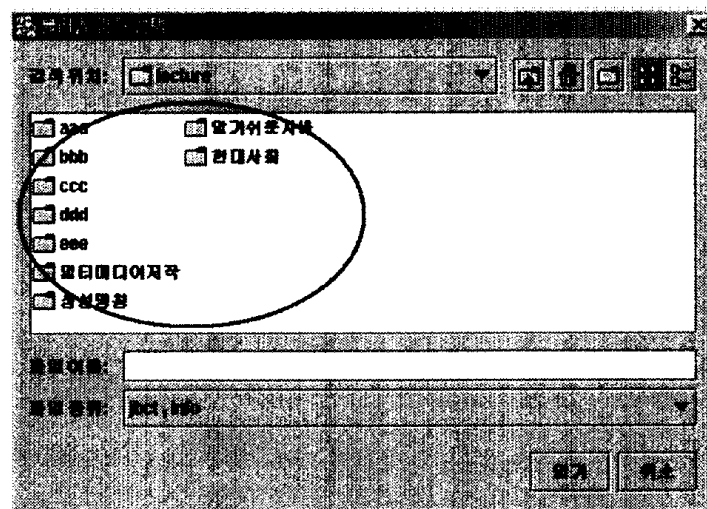
- (2) 강의 불러오기

- ① 강의를 불러오기 위해 강의 열기를 선택한다.



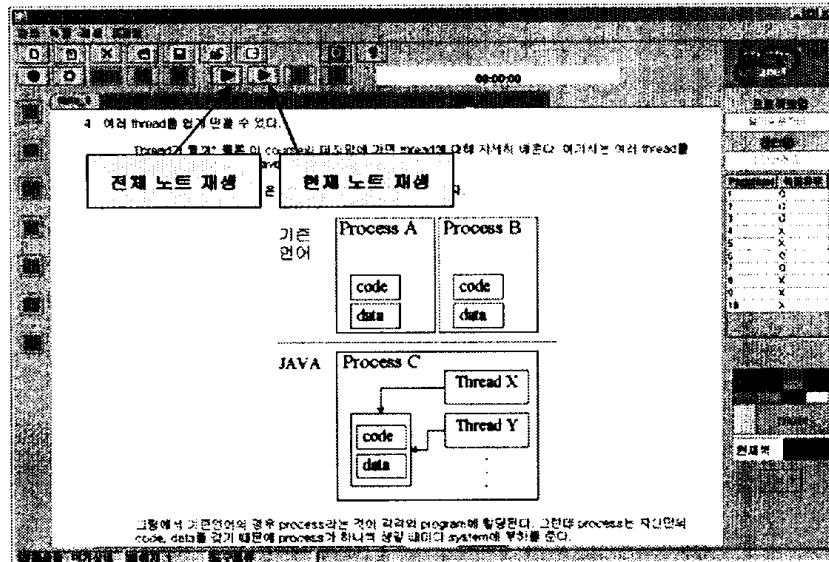
- ② 만들어진 강의 목록을 확인하고 재생할 강의를 선택한다.

만들어진 강의는 강의 목록 중 해당 과목의 챕터를 선택하고
jbct.info파일을 열어 실행시킨다.



(2) 강의 재생

- ① 불러온 강의를 실행시키기 위해 전체노트 재생 버튼이나 현재 노트 재생 버튼을 선택하여 강의를 재생한다.



(2) 강의 멈춤

- ① 실행중인 강의를 정지시키기 위해 일시정지 버튼이나 재생 중지 버튼을 선택하여 강의를 정지시킨다.

