

CAI 지원을 위한 멀티미디어 편집 프레임워크

김 행 곤*, 신 호 준*, 최 봉 식*

* 대구효성가톨릭대학교 컴퓨터공학과

Multimedia Edit Framework for CAI

Haeng-Kon Kim*, Ho-Jun Shin*, Bong-Sik Choi*

* Software Engineering Lab., Dept. of Computer Engineering,
Catholic University of Taegu Hyosung

요 약

멀티미디어 학습 자료는 학습자에게 다양한 형태의 학습경험을 제공해 줌으로써 학습에 대한 강한 동기유발과 이를 통한 학습목표의 달성에 크게 기여하고 있는 바, 그 교육적 활용가치는 날로 증대되고 있다. 대부분의 교사들은 멀티미디어 편집 관련 애플리케이션을 사용하여 학습자료를 제작하고 있지만 일반적으로 멀티미디어 편집에 있어서, 관련된 주요 기능들이 서로 다른 애플리케이션에서 실행되기 때문에 사용자의 입장에서 여러 개의 애플리케이션을 핸들링해야 하는 불편함이 있다. 또한, 지금까지의 멀티미디어 관련 툴이나 애플리케이션은 사용자의 수준을 고려해보았을 때 상당히 복잡하거나 난해한 부분들이 많아 활용상의 한계를 가지고 있다.

본 논문에서는 이러한 문제점을 해결하기 위한 멀티미디어 편집 프레임워크의 개발을 목표로 연구범위를 동영상의 캡처와 편집으로 제한하고, 지금까지의 사례를 모델링하여 분석하고, 이를 기반으로 새로운 형태의 동영상 캡처와 편집을 위한 프레임워크를 설계하였다. 이를 통해 두 작업 사이의 공통적인 구성요소를 추출하여 단순화시킴으로써 애플리케이션에 쉽게 적용할 수 있고, 사용의 용이성과 이식성을 가질 수 있다.

1. 서론

정보화 사회에서의 학교 교육은 학습자로 하여금 확산적인 사고를 바탕으로 하는, 창의적 문제 해결능력의 신장에 그 목표를 두고 있다. 책을 넘기는 것보다 마우스를 클릭하는 것에 더 익숙한 학습자들의 다양한 요구를 충족시켜 주기 위해서는 교과서와 칠판, 단순한 텍스트와 그래픽을 중심으로 하는 전통적 교수-학습방법만으로는 부족하다는 것에 교사, 학생 모두가 공감하고 있다. 따라서, 학교에서 이루어지는 교수-학습은 다매체적 정보를 중심으로 하는 새로운 형태의 방법 개선이 이루어져야 할 것이다. 또한, 컴퓨터 보급의 확산으로 이를 활용한 수업도 오늘날의 교사가 지향해야 할 중요한 부분이며, 특히 컴퓨터의 단순한 활용에서 벗어나 멀티미디어 학습 정보에 기반하여 체계적으로 설계된 CAI(Computer Aided Instruction)시스템으로의 활용이 이루어진다면 상당한 교육적 성과를 거둘 수 있다[1]. 그러나, 지금까지의 멀티미디어 관련 툴이나 애플리케이션은 사용자의 수준을 고려해보았을 때 상당히 복잡한 부분들이 많아 활용상의 한계를 가지고 있다. 대부분의 교사들은 멀티미디어 편집 관련 애플리케이션을 사용하여 학습자료를 제작하고 있지만, 일반적으로 멀티미디어 편집에 있어서, 관련된 주요 기능들이 서로 다른 애플리케이션에서 실행되기 때문에 사용자의 입장에서 여러 개의 애플리케이션을 핸들링해야 하는 불편함이 있다.

본 논문에서는 이러한 문제점을 해결하기 위해 동영상, 오디오, 이미지, 텍스트를 고려한 멀티미디어 편집 프레임워크 구성을 목표로 모델기반의 프레임워크 구성 프로세스와 모델을 제시한다. 전체 프레임워크 중에서 동영상의 캡처와 편집부분을 지금까지의 사례를 통해 모델링하여 분석하고, 이를 기반으로 하는 새로운 형태의 동영상 캡처와 편집을

위한 프레임워크를 설계하고자 한다. 이를 통해 두 작업 사이의 공통적인 구성요소를 추출하여 단순화시킴으로서 애플리케이션에 쉽게 적용할 수 있고, 멀티미디어 정보를 이런 방식으로 분석, 설계하고 각각의 개발된 프레임워크를 통합함으로써 사용의 용이성과 이식성을 가질 수 있게 한다.

2. 관련 연구

2.1 멀티미디어(Multimedia)

2.1.1 CAI 시스템

교사가 학생에게 교재나 문제를 제시하여 그에 대한 학생의 반응을 살피고, 이를 평가해서 피드백시키는 교사의 활동을 컴퓨터의 대량 정보처리능력을 이용해 대행시키는 시스템으로 컴퓨터를 이용하여 학습의 효과를 높이기 위해 교육분야에 활용되면서 대화식 상호작용에 의해서 학습이 진행되는 체계적인 시스템으로 개발되었다[2].

(1) 특징

학습자들로 하여금 의도된 수업 목표를 효율적으로 달성해 주도록 하는 효과적인 CAI시스템의 특성들을 요약하면 다음과 같다.

- 수업설계의 원리가 반영된 것
- 학습자 요구사항과 특성을 고려하여 상호 작용을 극대화하며, 학습에 대한 흥미와 긍정적인 강화를 제공할 것
- 컴퓨터의 기능을 폭넓게 활용할 수 있고, 현재의 활용환경에 적합할 것
- 적절한 심의를 받은 것

(2) 개발 절차

CAI시스템 개발은 일반적으로 요구사항 분석, 설계, 개발 및 적용의 순서로 이루어진다. 그리고 한 단계에서 다른 단계로 진행되는 통로가 반드시 평가 및 수정 단계를 거치도록 되어 있다. 즉 각 단계마다 지속적인 평가 및 수정의 과정을 거쳐야 하며, 이것에 따라 다음 단계로의 진행 여부를 판가름하게 된다. 또한, 종료라는 것이 없으며, 최초 버전이 완성된 후에도 실제 적용을 통한 평가와 수정 작업이 지속적으로 이루어져야 하기 때문에 정지점이 없다[3].

2.1.2 멀티미디어 저작도구

멀티미디어 저작도구는 컴퓨터 사용자들로 하여금 각종 멀티미디어 관련 자료를 손쉽게 만들어 서로 연결함으로써 다양한 형태로 출력할 수 있도록 해 주는 도구이다. 다양한 유형의 교수-학습용 멀티미디어 프로그램 개발을 위한 환경 조성 및 프로그램의 개발 기간과 비용을 절감시켜 질적으로 우수하고 다양한 교육용 프로그램 개발을 촉진시켜 주고 있다[4]. 현재 학교 현장에서 교사들이 비교적 많이 사용하고 있는 멀티미디어 저작도구를 인터페이스를 중심으로 그 특징과 구성, 문제점 등을 살펴보면 다음과 같다.

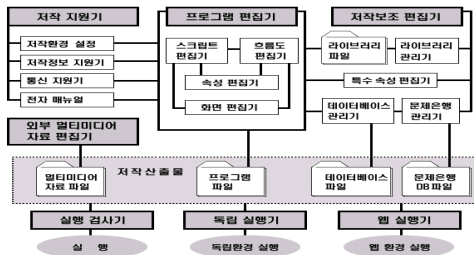
(1) 특징

멀티미디어 저작도구를 인터페이스를 중심으로 그 특징을 살펴보면 다음과 같다.

- 표준화된 사용법과 손쉬운 사용자 인터페이스로 프로그래밍에 관한 전문적인 지식이 없는 초보자도 일반적인 수준의 애플리케이션을 생성
- 프로그래밍 언어를 사용하지 않고 아이콘이나 스크립트를 사용하여 애플리케이션을 개발하는 방식
- 멀티미디어 관련 툴이 내장되어 있지 않고, 외부 미디어파일을 불러들여 적절한 위치나 시점에서 재생하는데 초점
- 마우스나 키보드 등의 이벤트 구동 방식으로 화면전환이 용이

(2) 시스템 구성

멀티미디어 저작도구 시스템은 일반적으로 다음(그림 1)과 같이 실제 프로그램을 작성하는 프로그램 편집기를 중심으로 각종 기능이 유기적으로 결합되어 있다.



(그림 1) 멀티미디어 저작도구의 시스템 구성도

이 외에도 저작환경을 설정하기 위한 저작지원기와 작성되는 프로그램의 내용에 따라 필요한 데이터 파일을 생성·관리하며 프로그램의 저작을 도와주는 저작보조편집기, 외부에서 제작한 멀티미디어 파일을 불러와 재생하는 외부 멀티미디어 자료 편집기, 완성된 프로그램을 각각의 환경에서 실행하기 위한 실행기 등으로 구성된다.

2.2 프레임워크(Framework)

프레임워크는 관계된 문제집합의 해결책에 대한 추상설계를 포함하고 있는 클래스들의 집합이라고 정의할 수 있다. 또한 애플리케이션 서브시스템 영역에 대한 책임들을 수행하기 위해 상호협력하는 객체들의 집합이라고 할 수 있다. 그러므로 추상적이고, 구체적으로 서로 협력하는 클래스들의 집합이고, 이는 소프트웨어의 특정 클래스에 대해 재사용 가능한 설계를 구성한다[5].

(1) 역할

프레임워크는 설계를 추상화 클래스들로 분리하고, 그들의 책임과 협동관계, 쓰레드 제어관계를 정의함으로써, 프레임워크를 사용하여 구축된 애플리케이션의 아키텍처를 결정한다. 애플리케이션 영역에 공통된 설계결정들을 뽑아냄으로써 애플리케이션 설계자는 개발하려는 애플리케이션에서 새로 요구되는 곳에서만 집중할 수 있게 된다. 그러므로 프레임워크의 개발은 코드의 재사용을 넘어 설계의 재사용을 강조하고 있다.

최근에 주목받는 객체지향 멀티미디어 프로그래밍의 최대 장점은 소프트웨어 환경의 확장이라는 메카니즘을 가지고 있다는 것이다. 또한, 프레임워크나 클래스 계층은 이러한 변화에 대해 대처할 수 있게 해준다. 개발자들은 새로운 데이터의 표현에 있어서, 이러한 플랫폼의 변화된 환경을 이용하고, 받아들일 수 있는 또는 미래의 환경까지도 고려한 애플리케이션을 개발하기를 원하고 있지만, 현실적으로 어려운 부분들이 많다. 그러나 프레임워크는 최소한 이러한 변화를 개발환경으로 통합할 수 있는 메카니즘을 제공해 준다[6].

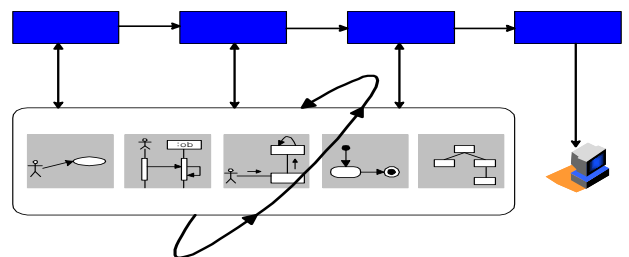
(2) 구축의 필요성

멀티미디어 애플리케이션 개발자들에게 가장 실제적인 어려움은 가장 안정적인 대상 플랫폼의 부족과, 애플리케이션의 실행을 위한 하드웨어와 그 기능이 급속하게 변화하고 있으며 크로스 플랫폼이 없어 이질 기종간의 호환성 문제가 발생한다는 것이다. 이러한 현상들을 묶어 “Platform Volatility”라고 하며 이러한 플랫폼의 변화가 애플리케이션의 개발을 어렵게 만들고 있다. 이를 해결하기 위한 방안으로 플랫폼이 급격히 변해도 연결가능하고 교체 가능한 프레임워크를 구축하는 것이다.

3. 멀티미디어 편집 프레임워크 구성

3.1 모델기반 프레임워크 개발 프로세스

멀티미디어 편집 프레임워크를 구성하기 위한 개발 방법으로 UML(Unified Modeling Language)을 이용한 모델기반의 개발 프로세스는 다음(그림 2)과 같다.



(그림 2) 프레임워크 구성을 위한 모델기반 프로세스

프레임워크 구축의 초기단계는 문제영역에서 도메인, 시스템, 사용자 요구사항을 추출하고 정형화시키는 단계로서 사용 시나리오를 작성할 수 있으며, 작성된 내용은 분석/설계 모델 구축에 초기 정보로 사용된다. 다음으로, 요구사항 명세를 바탕으로 분석을 하는 단계로서, 사용자 정의에 통한 사용자 요구사항을 파악하는 등의 모델 작성은 다음과 같은 다이어그램을 중심으로 작성된다.

또한, 작성된 분석 단계의 모델은 설계단계에서 반복적인 수정과 재작성을 통해 공통된 부분과 불일치되는 요소를 파악하여 설계모델을 작성한다. 이렇게 일련의 분석, 설계모델은 모델의 반복적, 단계적, 프로세스를 수행한다.

- Scenario : 사용자의 요구사항을 일련의 텍스트 형식으로 작성
- Use Case Diagram : 프로세스 모형화와 사용자 요구사항 분석을 통한 분석모델과 개념적 모델 제시
- Sequence Diagram : 사용자 요구사항의 시나리오 작성에 기반하여 객체들 간의 메시지 전달을 시간 지향적인 행위로 파악
- Collaboration Diagram : 객체들간의 상호작용을 모형화하는 것으로 메시지 지향적인 행위를 기술
- State Diagram : 시스템 내의 객체들의 행위를 이벤트 중심에 따라 취할 수 있는 상태를 기술
- Class Diagram : 시스템의 정적인 구조를 파악하기위해 연관, 상속, 집합 등의 연관관계를 나타냄으로써 개발 전체에 사용될 시스템 이름과 모델 개념을 정립하고 모형화

프레임워크 구성단계에서는 분석, 설계 모델을 기반으로 서로의 연관관계와 내부적으로 상세하게 구현되는 부분을 인터페이스를 뒤에 숨기고 서비스를 제공하며, 구축된 모델을 기반으로 캡슐화를 통해 최소한의 모듈성을 강조하며, 개발자의 요구에 맞게 애플리케이션을 구축할 수 있도록 서비스의 변경을 위한 추상화 정도를 결정하게 되며, 구성된 프레임워크는 기존이나 새로운 애플리케이션에 적용된다.

3.1.1 분석 모델 작성

멀티미디어 편집 프레임워크를 구성하기 위해서 본 절에서는 동영상 부분만을 고려하여 기존의 동영상 생성과 편집 부분을 각각 일련의 모델로써 분석하도록 한다.

(1) 멀티미디어 편집 애플리케이션 사용자례 명세

기존의 멀티미디어 편집 애플리케이션에서 동영상의 캡처와 편집에 관련된 사용자례는 각각 다음(그림 3-a, 3-b)과 같이 작성할 수 있다.

동영상 캡처는 컴퓨터상의 영상 캡처 카드와 A/V 단자를 통해 연결되어있는 VTR에 해당 비디오 테이프를 넣거나, 해당 TV 채널을 선택한 후, 캡처 프로그램을 활성화시킨다. 메뉴에서 미리보기 버튼을 클릭하고, 저장할 파일명을 입력한다. 영상이 나타나는 창이 활성화되었을 때 VTR의 재생버튼을 클릭한 후 재생되는 영상을 보고, 필요한 부분은 녹화버튼을 클릭하여 지정한 파일로 저장한다.

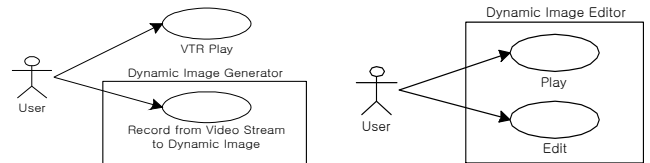
(그림 3-a) 동영상 캡처 사용자례 명세

동영상 편집은 캡처 프로그램 내에서는 파일의 편집이 불가능하므로 저장된 동영상 파일 중 편집을 원하는 대상 파일을 편집 프로그램으로 불러낸다. 플레이어 창(Player Window)에 대상 파일을 활성화시킨 다음, 편집을 원하는 부분의 시작과 끝부분을 표시한다. 이때, 시작과 끝 부분을 정확히 표시하기 위하여 프레임의 진행속도를 조절하여 전진, 또는 후퇴의 방법으로 맞추어 준다. 편집을 원하는 부분이 정확히 맞추어졌을 때 레코더 창의 녹화버튼을 클릭 하면 레코더 창 쪽으로 해당부분만(편집을 원하는 부분만) 녹화된다. 현재의 편집부분의 앞, 중간, 뒷부분에 또 다른 편집된 부분을 첨가할 수도 있다. 편집이 끝난 동영상은 레코더 창에서 일단 재생하여보고 이상이 없으면 파일 메뉴에서 저장을 선택하여 저장한다.

(그림 3-b) 동영상 편집 사용자례 명세

(2) Use Case Diagram

다음(그림 4-a)은 멀티미디어 편집 애플리케이션중 비디오 스트림을 입력받아 동영상 파일로 저장하는 Use Case Diagram을 나타낸 것이며, 디지털 파일로 저장된 동영상을 사용자가 원하는 형태로 편집하는 것은 다음(그림 4-b)과 같다.

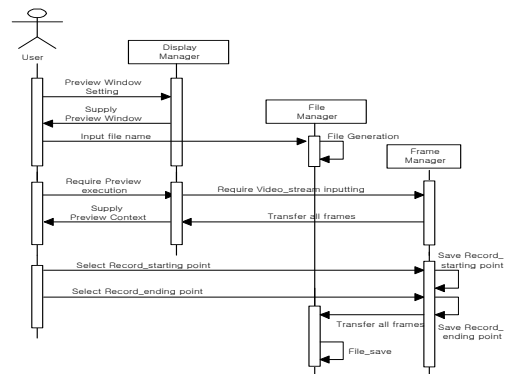


(그림 4-a) 캡처 Use Case Diagram (그림 4-b) 편집 Use Case Diagram

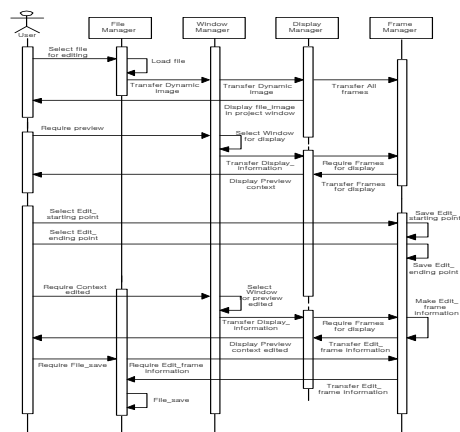
(3) Sequence Diagram

다음(그림 5-a)은 동영상 파일의 생성을 위한 프레임워크를 구성하는 Display Manager, File Manager, Frame Manager를 중심으로 사용자가 비디오 스트림으로부터 동영상을 캡처하여 파일을 생성하는 과정을 시간의 흐름에 따라 나타낸 것이다.

동영상 파일의 편집을 위한 프레임워크를 구성하는 Display Manager, File Manager, Frame Manager, Window Manager를 중심으로, 사용자가 이미 저장된 동영상 파일을 편집기로 불러내고, 편집할 부분(context)을 버퍼에 저장하여 cut, paste, copy 등으로 편집하여 새로운 동영상 파일을 생성하는 과정을 시간의 흐름에 따라 나타낸 것은 다음(그림 5-b)과 같다.



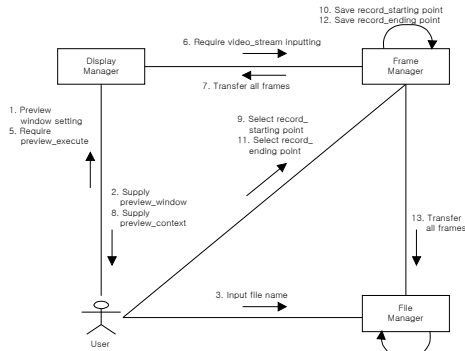
(그림 5-a) 캡처 Sequence Diagram



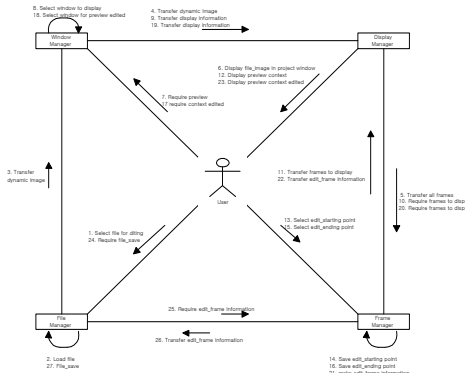
(그림 5-b) 편집 Sequence Diagram

(4) Collaboration Diagram

다음(그림 6-a)은 구성 요소 사이의 행위를 표현하는 Collaboration Diagram으로써, 비디오 스트림을 입력받아 동영상 파일을 생성하는 구성 요소간의 상호작용을 나타낸 것이다. 또한, 다음(그림 6-b)은 저장된 동영상 파일을 불러내어 사용자가 원하는 형태로 편집하여 새로운 동영상 파일을 생성하는 구성 요소간의 상호작용을 나타낸 것이다.



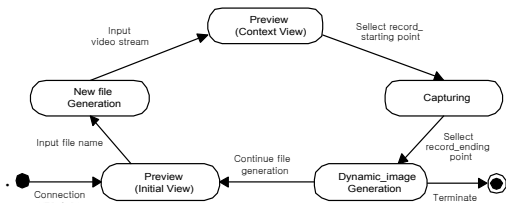
(그림 6-a) 캡처 Collaboration Diagram



(그림 6-b) 편집 Collaboration Diagram

(5) State Diagram

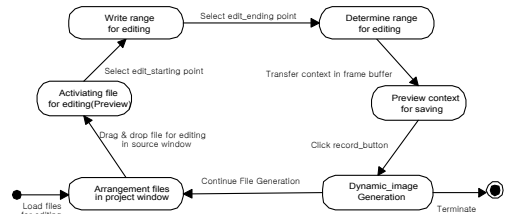
구성요소의 설명을 보완하며 구성요소가 가질 수 있는 모든 가능한 상태를 단지 여러 개의 잘 정의된 상태와 행위가 서로 다른 상태에 의해 어떤 영향을 미치고 변화를 주는지를 나타내는 것으로 다음(그림 7-a)은 동영상의 캡처를 위한 State Diagram이다. 여기에서 캡처작업은 사용자가 녹화의 시작포인트를 선택할 때부터 시작되며, 종료 포인트를 선택했을 때 끝나게 된다.



(그림 7-a) 캡처 State Diagram

다음(그림 7-b)은 동영상의 편집을 위한 State Diagram으로 (그림 7-a)와 가장 큰 차이점은 편집 대상 프레임의 시작

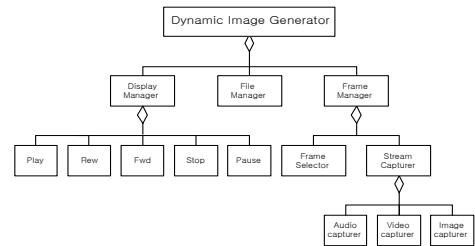
포인트와 종료 포인트를 선택하는 것은 편집의 범위를 결정하는 것이지, 그 자체가 녹화와 저장, 새로운 동영상 파일의 생성을 의미하지는 않는다.



(그림 7-b) 편집 State Diagram

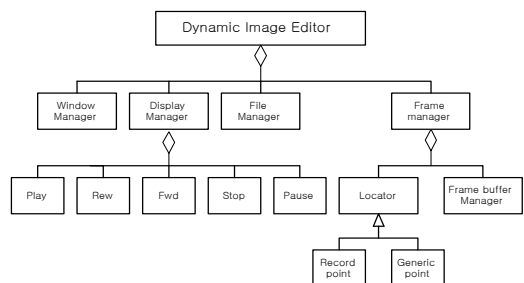
(6) Class Diagram

동영상 편집 프레임워크를 설계하기 위한 주요한 정보로 사용되는 다이어그램으로써 시스템에서 다루어지는 구성요소들의 정적인 구조를 연관, 종속, 특수화된 또는 패키지화된 여러 가지 방법으로 다른 것들과 각각 관계를 가지는 것을 표현하고 있다. 다음(그림 8-a)은 동영상 파일의 생성을 위한 Class Diagram으로 특히, Stream Capturer 클래스에 비디오 스트림으로부터 비디오, 오디오, 이미지(정지 화상)를 별도로 캡처 하는 서브클래스를 둔다.



(그림 8-a) 캡처 Class Diagram

다음(그림 8-b)은 동영상 파일의 편집을 위한 Class Diagram으로 특히, 편집 범위를 지정하기 위한 Locator 클래스에 프레임 단위로 편집의 시작과 끝부분을 정확하게 지정하기 위한 Record Point 클래스와 편집대상파일의 프리뷰(preview)를 위한 Generic Point 클래스를 둔다.



(그림 8-b) 편집 Class Diagram

3.1.2 설계

(1) 멀티미디어 프레임워크 사용사례 명세

기존의 편집 애플리케이션을 분석한 모델을 바탕으로 구성할 프레임워크를 설계하기 위해서 우선, 동영상의 캡처와 편집을 동일한 프레임워크 내에서 구성할 수 있는 사용사례를 다음(그림 9)과 같이 요구사항을 포함하여 명세할 수 있다.

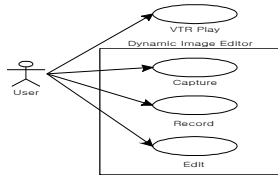
동영상 캡처작업은 비디오 스트림의 컴퓨터 입력을 의미하는 것으로, 컴퓨터상의 영상 캡처 카드와 A/V 단자를 통해 연결되어있는 VTR에 해당 비디오 테이프를 넣고 동영상 편집 프로그램을 실행시킨다. 메뉴에서 캡처를 선택하고, 캡처환경을 설정한다. VTR의 재생 버튼을 누른 후 소스 창(Source Window)에 프리뷰(Preview)되는 영상을 보고 필요한 부분을 캡처한다. 이 때 캡처한 부분은 일단 프레임 버퍼에 저장하여 비디오 스트림의 입력을 마친다. 이때, 프레임버퍼에 임시로 저장된 스트림을 동영상 파일로 저장하기 전에 시작과 끝 부분을 정확히 표시하기 위하여 프레임의 진행속도를 조절하여 전진, 또는 후퇴의 방법으로 맞추어 준다. 정확하게 저장범위가 결정되었을 때 저장을 선택하고 파일명을 입력하여 저장소에 저장한다.

동영상 편집작업은 저장된 동영상 파일 중 편집을 원하는 대상 파일을 프로젝트 창(Project Window)로 불러내고, 소스 창으로 드래그하여 대상 파일을 활성화시킨다. 파일을 프리뷰하여 편집을 원하는 부분의 시작과 끝 부분을 표시한다. 이때, 시작과 끝 부분을 정확히 표시하기 위하여 프레임의 진행속도를 조절하여 전진, 또는 후퇴의 방법으로 맞추어 준다. 편집을 원하는 범위가 정확히 맞추어졌을 때 레코더 창(Recorder Window)의 녹화버튼을 클릭하면 지정한 범위의 내용이 프레임버퍼에 일시적으로 저장된다. 현재 편집부분의 앞, 중간, 뒷부분에 또 다른 편집된 부분을 추가할 수도 있다. 편집이 끝난 동영상은 레코더 창에서 프리뷰 해 보고 이상이 없으면 파일 메뉴에서 저장을 선택하여 저장한다.

(그림 9) 멀티미디어 프레임워크 사용사례 명세

(2) Use Case Diagram

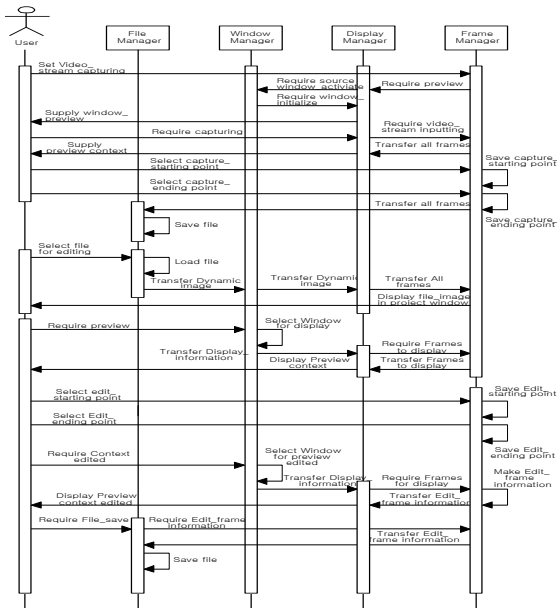
다음(그림 10)은 멀티미디어 편집 애플리케이션 중 비디오 스트림을 입력받아 파일로 저장하거나, 이미 저장된 파일을 불러내어 편집하는 Use Case Diagram이다.



(그림 10) Use Case Diagram

(3) Sequence Diagram

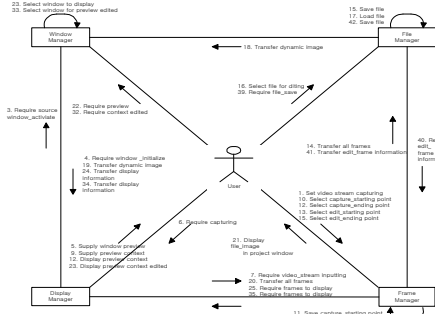
다음(그림 11)은 동영상 캡처 작업 후에 바로 편집작업이 이루어진다는 가정하에서의 Sequence Diagram이다.



(그림 11) Sequence Diagram

(4) Collaboration Diagram

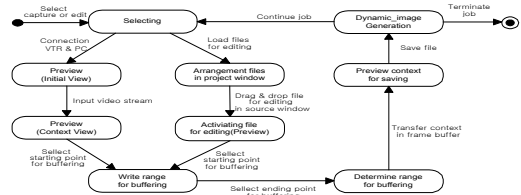
다음(그림 12)은 비디오 스트림을 입력받아 동영상 파일을 생성하고 편집하는 구성 요소간의 상호작용을 나타낸 Collaboration Diagram이다.



(그림 12) Collaboration Diagram

(5) State Diagram

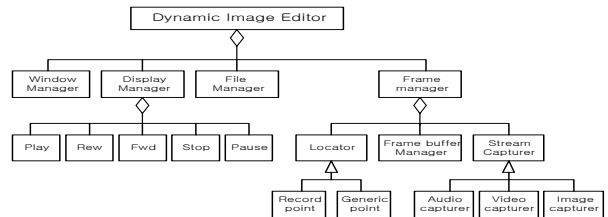
다음(그림 13)은 State Diagram으로 가장 큰 특징은 시작 부분에서 동영상의 캡처와 편집을 선택하며, 하나의 동영상 파일이 생성된 후에 작업을 계속할 것인지를 질의한다. 또한, 파일로 저장할 때에는 선택한 컨텍스트를 프레임 버퍼에 일시적으로 저장, 프리뷰하여 그 내용을 확인하고 난 뒤에 이루어진다.



(그림 13) State Diagram

(6) Class Diagram

다음(그림 14)은 본 논문에서 제안하고 있는 동영상 파일의 생성과 편집을 위한 Class Diagram으로 특히, 비디오 스트림으로부터 비디오, 오디오, 이미지(정지 화상)를 캡처하여 프레임 버퍼에 일시적으로 저장하기 위한 Frame Buffer Manager 클래스를 가지며, Locator 클래스가 있어 캡처 작업 시에도 시작과 끝 부분을 정확하게 지정할 수 있다.

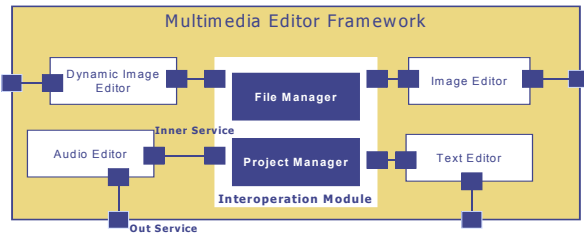


(그림 14) Class Diagram

3.1.3 프레임워크 구성

앞에서 구성된 동영상 설계모델을 프레임워크로 구성하기 위해서는 프레임워크 외부로 서비스가 되어야할 부분과 내부적으로 각각의 구성요소와 메시지를 주고받고 상호연

되는 부분을 고려하여야하며, 모든 요소가 공통적으로 수행해야할 부분에 각 요소들은 허브형태로 통합되며, 외부적으로는 내부의 상세한 설계와 구현은 외부 서비스 인터페이스로 감추고 다음(그림 15)과 같이 프레임워크를 구성할 수 있다.



(그림 15) 프레임워크 구성

4. 프로토타입 및 평가

4.1 프로토타입

개발될 멀티미디어 프레임워크는 공통된 작업 공간에서 각각의 멀티미디어 요소를 편집할 수 있으며, 동영상은 편집을 위한 창과 미리보는 창의 기능 차이는 없다. 소스창과 캡처창은 작업이 이루어지는 것에 따라 바뀔 수 있으며, 전체 진행사항과 제어를 위한 툴은 각각의 멀티미디어에 관련된 탭창에 위치한다. 다음(그림 16)은 전체 프레임워크와 동영상 파일을 위치시킨 것을 볼 수 있다.



(그림 16) 멀티미디어 편집 프레임워크의 프로토타입

4.2 평가

본 절에서는 도메인 분석의 대상이 된 기존의 멀티미디어 관련 애플리케이션과 구성된 프레임워크와의 상대적인 비교를 통하여 <표 1>과 같이 평가한다.

<표 1> 다른 시스템과의 비교

구분 \ 툴	VicoMpeg	Ifilmedit	Adobe premiere	멀티미디어 편집프레임워크
용도	동영상 캡처용	동영상 편집용	동영상 편집용	멀티미디어 편집용
캡처능력	비디오카드	유일	-	일부 제한
	오디오별도	○	×	○
	비디오별도	○	×	○
	정지화상	○	×	○
동영상 편집	동영상	○	×	○
	일반편집	×	○	○
특수편집	×	×	○	×
시스템 구성	단순	단순	복잡	단순
사용자 인터페이스	단순	단순	복잡	단순
재사용 단위	도구	도구	도구	프레임워크
개발시간	장	장	장	중

기존의 멀티미디어 편집 시스템은 동영상 편집과 관련하여 비교해 보았을 때 캡처부분과 편집이 서로 다른 프로그램내에서 이루어지는 불편함이 있었지만, 본 논문에서 제시하는 프레임워크는 두가지 작업을 모두 한 개의 애플리케이션 내에서 할 수 있도록 설계하였다. 향후 과제로 제시할 이미지, 오디오, 텍스트의 재생과 편집 또한, 이 통합 프레임워크 내에서 이루어짐으로써 여러 개의 프로그램을 사용해야하는 번거로움과 복잡함으로부터 벗어날 수 있다.

또한, 멀티미디어 편집 중 동영상의 캡처와 편집 프레임워크는 모델에 기반한 절차로서 사용자와 소프트웨어, 시스템에 대한 요구사항을 고려하여 UML을 이용하여 분석, 설계 모델을 만든다. 또한 체계적인 개발 프로세스를 통해서 애플리케이션을 개발함으로써 기존의 애플리케이션에 보다 더 향상된 유지보수성을 기대할 수 있으며, 유사한 시스템 개발에 본 프레임워크를 재사용 할 수 있다.

5. 결론 및 향후연구

일반적으로 멀티미디어 편집에 있어서, 관련된 주요 기능들이 서로 다른 애플리케이션에서 실행되도록 되어있기 때문에 사용자의 입장에서는 여러 개의 애플리케이션을 핸들링해야 하는 불편함이 있다. 또한, 지금까지의 멀티미디어 관련 툴이나 애플리케이션은 사용자의 수준을 고려해보았을 때 상당히 복잡하거나 난해한 부분들이 많아 활용상의 한계를 가지고 있다.

본 논문에서는 이러한 문제점을 해결하기 위한 멀티미디어 편집 툴의 개발을 목표로 우선 연구범위를 동영상의 캡처와 편집으로 제한하고, 지금까지의 사례를 모델링하고, 이를 기반으로 하는 새로운 형태의 동영상 캡처와 편집을 위한 프레임워크를 설계하였다. 지금까지는 하나의 애플리케이션 내에서의 동영상 캡처와 편집이 이루어진다 하더라도 완전히 분리된 작업으로 인해 매우 복잡하였으나, 본 논문에서는 두 작업 사이의 공통적인 구성요소를 추출하여 단순화시킴으로서 애플리케이션에 쉽게 적용할 수 있고, 사용의 용이성과 이식성을 가질 수 있다. 향후 오디오, 이미지, 텍스트의 편집기를 추가하여 멀티미디어 편집과 관련된 대부분의 요구사항을 충족시켜주기 위한 멀티미디어 통합 프레임워크 개발에 대한 연구가 요구된다.

【참고 문헌】

- [1] 김세환, "상황적 시뮬레이션을 적용한 하이퍼미디어의 설계 및 구현", 한국교원대학교 대학원 석사학위논문, 1996.
- [2] 박말련, "CAI 개발 지원을 위한 객체지향 프레임워크 설계 및 구현", 대구효성가톨릭대학교 교육대학원 석사학위논문, 1999.
- [3] 교육부, "코스웨어 개발요원 연수교재", 1993.
- [4] 한국교육학술정보원, "교육용 멀티미디어 저작도구 PASS2000", 1999.
- [5] 배재민, "설계패턴과 Hotspot을 기반으로 한 객체지향 영역 프레임워크의 개발 방법", 중앙대학교 대학원 석사학위논문, 1998.
- [6] Oscar Nierstrasz & Dennis Tsichritzis, Object_Oriented Software Composition, Prentice Hall, 1995.
- [7] Artnix, "VicoMpeg", Available by web server from <http://www.artnix.co.kr/kor>
- [8] Adobe, "Adobe Premiere", Available by web server from <http://www.adobe.com/products/premiere>