

PC 기반의 플루트 연주 자율학습 소프트웨어 개발

김재영*, 이정철*, 전희성*

Development of PC based flute performance learning software

Jae-Young Kim*, Jung-Chul Lee*, Hee-Sung Jun*

요약

음악교육은 창의력, 사회성, 학업 성취도를 향상시킨다. 효율적인 음악교육을 위해서 학습자의 학습 수준, 속도에 맞게 학습할 수 있는 학습도구의 개발에 대한 요구가 높아지고 있다.

본 논문에서는 콘텐츠의 형식을 지정하는 템플릿과 콘텐츠 내용을 기술하는 디스크립터 파일을 사용하여 일반 PC환경에서 PC MIDI를 이용한 자기주도 플루트 연주 학습 소프트웨어를 구현하는 방법을 제안한다. 제안된 방법은 사용자가 디스크립터 파일을 직접 변경할 수 있어서 악기 연주 학습자의 수준에 맞게 콘텐츠를 활용할 수 있는 장점이 있다. 본 논문에서 제안하는 방법을 이용하여 PC기반 플루트 연주 학습 소프트웨어를 구현함으로써 구현의 효율성을 보였으며 악기 초보자가 플루트 연주 학습에 편리하게 활용할 수 있는 가능성을 확인하였다.

▶ Keywords : 관악기, 자율학습 프로그램, PC MIDI

Abstract

The music education improves the creative talent, social skills and academic achievement of the students. For the efficient music education, it is requested to develop the collaborative educational learning tools, especially electronic collaborators suitable to the learner's study patterns and speed.

In this paper, we propose a new method to develop a PC-based self learning software for the flute performance using templates and descriptors to make the contents form and substance. Our proposed method can allow user to modify the descriptors to match the contents to his level. We implemented a PC-based self learning software for the flute performance compactly and a feasibility test showed the efficiency of our proposed method to construct a self learning tool to play the flute and the tool can be utilized for the beginner to learn playing flute.

▶ Keywords : Woodwind instrument, Self-study program, PC MIDI

• 제1저자 : 김재영 • 교신저자 : 이정철
• 투고일: 2013.1.3, 심사일 : 2013.1. 20, 게재확정일 : 2013. 1. 31.
* 울산대학교 전기공학부(School of Electrical Engineering, University of Ulsan)

I. 서론

최근 연구에 따르면 사회/경제적 직위가 학업 성취도와 상관도가 크며, 음악 교육을 받은 학생들의 수학, 읽기, 과학, 사회 등의 교과목 성취도가 훨씬 높다는 연구 결과가 있다 [1]. 또한, 음악 학습은 학생들의 창의성과 사회성을 향상시키는데 중요한 역할을 하고 있다. 따라서 음악학습의 효율을 향상시키기 위한 다양한 음악교육 방식 연구가 진행 중이며 그 중에서도 학습자 중심의 교육이 강조되고 있다[2]. 학습자 중심의 교육을 위해서는 학습내용을 제시하고 학생의 학습 수준을 평가하여 학생의 자기 수준을 인식 시켜주고, 자신의 학습 방향을 선택하여 수준에 맞게 학습할 수 있는 프로그램 개발이 요구된다[3]. 이러한 요구를 충족하기 위해서는 자율 학습이 가능한 전자악기와 음악 교육 소프트웨어가 필요하다.

교육용으로 사용할 수 있는 PC 기반 상용 음악 소프트웨어는 케이크워크(Cakewalk)의 소나(Sonar), Steinberg의 큐베이스(Cubase), Steinberg의 누엔도(Nuendo), MakeMusic의 피날레(Finale) 등이 있다[4-7]. 최근 누엔도를 활용하여 실제 중학교 음악 과목을 교육한 연구 사례도 있다[8]. 연구 결과에 따르면 컴퓨터 활용 수업으로 학생들의 참여율을 높일 수 있었으며 시청각적인 피드백으로 높은 학습율을 보였다. 그러나 이러한 소프트웨어들은 교육용으로 개발된 소프트웨어가 아니기 때문에 초보자가 사용하기에 다소 어려움이 있어 별도의 학습 지도가 필요하다. 음악 교육용으로 개발된 상용 해의 소프트웨어는 Practica Musica, Smart music 등이 있지만 이들 소프트웨어는 건반악기와 현악기 중 일부만 연주 학습을 지원하며 학습에 사용되는 가상 악기 모듈이 제한적이다[9,10]. 외장 전자악기와 연동할 수 있는 악기 연주학습 소프트웨어 개발 연구도 있었지만 관악기와 연동을 지원하지 않는다[11]. 하이퍼미디어 기반 목관악기 학습을 위한 CAI(Computer Assisted Instruction) 프로그램 개발 연구도 있었지만 HTML기반으로 제작된 텍스트, 그림, 동영상 정보를 단순히 선택하여 보고 듣기 위주이므로 악기 연주에 관한 학습내용이 제한적이다[12].

기존의 웹기반 자율 학습 콘텐츠 제작과 관련된 연구에서는 교수자는 과목별로 공부방을 만들 수 있고 문제은행을 등록 및 수정할 수 있으며, 학습자는 과목과 카테고리를 선택하여 해당 문제은행에서 출제된 문제를 풀고 채점된 결과를 확인할 수 있다[13]. 하지만 문제의 난이도나 다양성을 고려하지 않았고, 프로그램을 실행할 때마다 동일순서의 문제가 반복되어 출제된다. 그리고 텍스트, 파워포인트 및 동영상을 유

동적으로 사용할 수 있고 관련 문제들을 파일로 등록할 수 있도록 설계한 연구도 수행되었다[14]. 하지만 각 요소들의 배치가 고정적이며 콘텐츠 등록을 위한 파워포인트와 문항을 사전에 수작업으로 제작하여야 한다는 점에서 학습 콘텐츠 제작에 소요되는 시간이 길어는 단점이 있다.

본 논문에서는 현재의 문제점을 해결하기 위해서 템플릿과 디스크립터 파일을 이용하는 학습 프로그램 개발 방식을 제안한다. 이 방식은 사용자가 직접 학습 콘텐츠와 문제 은행을 제작 및 편집을 가능하게 하며, 프로그램 사용자가 프로그램 내용을 추가 및 삭제할 수 있다. 그리고 플루트 가상악기 모듈을 구현하고 사용자 인터페이스를 단순하게 구현함으로써 초등학교 저학년도 쉽게 사용할 수 있는 플루트 연주 자율 학습 기능을 구현하였다. 학습을 위해 동영상 및 영상과 같은 멀티미디어, PC 내장형 MIDI, 연주 음향 효과를 통해 플루트 운지법과 연주법, 악보 읽는 법 등의 기능도 구현하였다. 또한 학습내용을 단계별로 구분하고 각 단계의 학습이 완료되면 바로 연습문제를 제시하여 배운 학습 내용을 즉각적으로 익힐 수 있도록 한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구에 사용된 플루트 연주 학습용 콘텐츠 제작 및 분석결과를 살펴보고, 3장에서는 플루트 연주 자율학습 프로그램 구현 방법을 설명한다. 4장에서는 실험과 평가를 통해 제안된 방법의 성능을 확인하고 5장에서 결론을 기술하였다.

II. 플루트 연주 학습 콘텐츠 제작 및 분석

본 연구에서는 템플릿과 디스크립터 파일을 이용한 플루트 연주 학습 소프트웨어를 개발하였다.

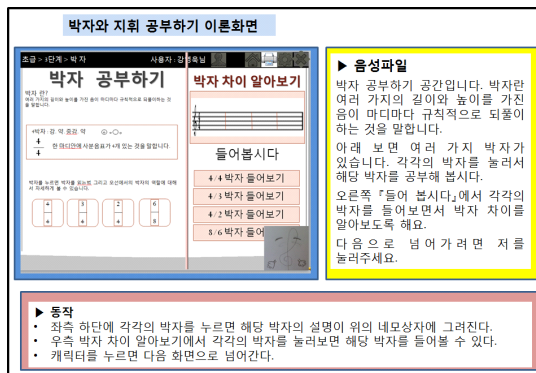


그림 1. 플루트 연주 학습용 콘텐츠 작성 예시
Fig. 1. A content example for learning of flute play

이를 위해 먼저 악기 연주 교육자의 도움을 받아 그림 1과 같이 학습 단계별로 초등학교 저학년용 플루트 연주 학습 콘텐츠를 문서 파일 형태로 제작하였다.

그리고 그림 2와 같이 소프트웨어 개발자는 제작된 콘텐츠들의 구성요소를 분석하여 컴포넌트들을 분류하고, 구성화면 내 컴포넌트들의 배치 패턴과 기능을 분석하여 유형별 템플릿을 작성하였다. 이후 학습용 콘텐츠의 형식을 나타내는 템플릿과 컴포넌트의 구체적인 정보를 텍스트 형식으로 표시하는 디스크립터 파일을 작성하였다.

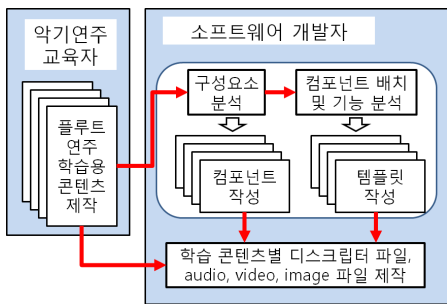


그림 2. 플루트 연주 학습용 콘텐츠 제작 및 분석 과정
Fig. 2. Contents production for learning of flute play and analysis procedure

그림 3은 플루트 연주 학습 소프트웨어가 템플릿과 디스크립터를 이용하여 학습 콘텐츠를 생성하는 과정을 보인다. 각 디스크립터 파일에는 템플릿 유형과 컴포넌트에 대한 속성들이 저장되어 있다. 소프트웨어는 명시된 템플릿을 구성하는 각 컴포넌트를 디스크립터에 저장된 내용으로 표현함으로써 학습 콘텐츠를 생성하게 된다.

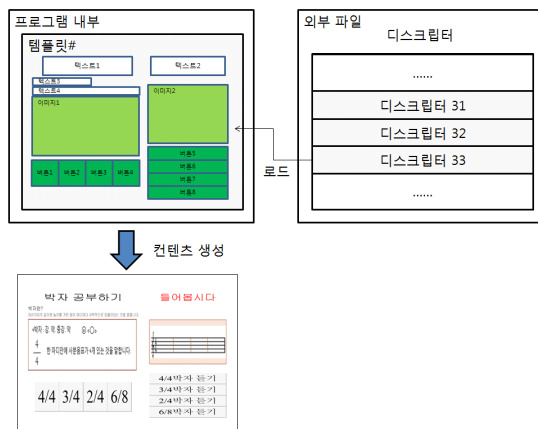


그림 3. 콘텐츠의 생성 과정
Fig. 3. Procedure of contents creation

본 논문에서 제안하는 방식으로 학습 소프트웨어를 제작할 경우, 콘텐츠 제작자는 제작하려는 콘텐츠에 적합한 템플릿을 선택함으로써 컴포넌트들의 배치를 고려하지 않고 콘텐츠의 내용 작성에 집중할 수 있으며 디스크립터 파일 작성 및 수정을 통해 플루트 악기 연주 학습자의 필요에 따라 독창적인 콘텐츠를 제작할 수 있어 자기주도적 학습의 학습전략 선택이 가능한 장점이 있다(3).

1. 학습 콘텐츠 분석

본 논문에서 플루트 연주 학습 소프트웨어의 구현을 위해 사용된 단계별 플루트 연주 학습용 콘텐츠들의 구성은 표 1과 같다. 각 단계에서는 플루트 연주에 필요한 기초적인 이론들과 연주법을 학습하고, 연습문제 풀이를 활용하여 학습 내용에 관한 이해도를 평가할 수 있게 하였다. 연습문제 풀이 과정에서 학습자가 문제의 정답을 맞히지 못할 경우 다음 과정으로 넘어가지 못하게 하여 해당 과정을 확실히 익힐 수 있도록 구현하였다.

표 1. 플루트 연주 학습 콘텐츠의 구성도
Table 1. structure of flute performance learning contents

플루트와 친해지기	
1단계	바른자세
	손모양 만들기
2단계	음이름과 플루트 키 누르기
	오선과 마디
3단계	음자리표
	음표
	쉼표
4단계	박자와 지휘
	큰보표
	반복기호

2. 컴포넌트

표 1의 콘텐츠들을 소프트웨어로 구현하기 위해서 먼저 콘텐츠의 구성요소를 분석하고, 소프트웨어 구현에 필요한 콘텐츠들의 컴포넌트들을 작성하였다. 컴포넌트의 종류는 표 2와 같이 버튼, 텍스트, 이미지, 가상 피아노, 가상 플루트, 동영상, 음성 등이 있다. 버튼은 문제선택, 화면이동 등과 같이 사용자 선택을 위해 사용된다. 텍스트는 학습 내용이나 문제의 내용, 힌트 등을 출력할 때 사용된다. 이미지는 플루트의 운지법이나 음표를 그리는 순서 등과 같이 학습자에게 사진이나 그림을 통해 내용을 전달할 때 사용된다. 동영상은 플루트의 연주방법, 박자의 지휘법 등을 학습자에게 동영상으로 보여줄 때 사용된다. 음성은 학습 내용에 대한 설명이나 문제를 푸는 방법 등을 말로 설명할 때 사용된다. 가상 피아노와 가상 플

루트는 학습자에게 플루트의 연주방법 교육, 악보를 읽고 연주하는 연습, 연습문제 출제 등에 사용된다.

표 2. 컴포넌트 구조체의 속성들
Table 2. Attributes of component structures

컴포넌트 이름	속성
버튼	이벤트 번호, 논리적 크기
텍스트	폰트, 글자색, 정렬, 논리적 크기
이미지	논리적 크기
동영상	논리적 크기
음성	없음
가상 피아노	논리적 크기
가상 플루트	논리적 크기
문제	컴포넌트 인덱스, 컴포넌트 종류, 문제 목록 문자열 벡터
문제 은행	출제 반복 횟수, 문제 구조체 벡터, 선택항 목록 문자열 벡터, 정답번호 정수형 벡터
문제 출제	문제 인덱스, 정답번호, 정답 선택지 인덱스, 오답 목록 정수형 벡터



그림 4. 템플릿의 대표적인 유형들
Fig. 4. Typical formats of template

3. 템플릿

템플릿은 컴포넌트의 조합에 따라 다양하게 구현되는데 그림 4와 같이 크게 4가지의 대표적인 유형으로 구분되며, 컴포넌트의 개수, 위치, 기능에 따라 세부적으로 나뉜다. 그림 4의 왼쪽 위에서부터 오른쪽 아래로 각각 학습형, 학습/실습형, 연습형, 문제형 템플릿의 예를 나타내었다.

학습형은 음성, 텍스트, 동영상, 이미지 컴포넌트를 이용하여 학습자에게 학습내용을 교육하는 형태의 템플릿이다. 학습/실습형은 가상 피아노, 가상 플루트, 이미지, 텍스트, 버튼 컴포넌트를 이용하여 학습자에게 학습내용을 교육하는 동시에 학습한 내용을 바로 실습할 수 있도록 하는 템플릿이다. 연습형은 가상 플루트와 악보를 이용하여 학습자가 배운 연주방법을 연습할 수 있는 템플릿이다. 문제형은 앞서 배운 내용을 익히기 위해 문제를 출제하는 템플릿으로서 텍스트, 버튼, 이미지, 가상 피아노, 가상 플루트 컴포넌트가 주로 사용된다.

4. 디스크립터 파일

디스크립터 파일은 학습자도 수정, 추가, 삭제가 가능하여야 하므로 일반 사용자에게 친숙한 텍스트 형식으로 작성하였다. 그림 5는 디스크립터 파일의 구조를 보인다. 첫 번째 줄에는 학습과정 경로, 두 번째 줄에는 템플릿 이름이 명시된다. 세 번째 줄부터는 템플릿에 채워질 컴포넌트들의 내용이 컴포넌트의 종류별로 명시된다.



그림 5. 디스크립터 파일의 구조
Fig. 5. Architecture of descriptor file

문제은행의 기술은 'question' 문자열의 다음 줄부터 시

작된다. 'repeat'문자열은 반복횟수를 정해주는 지정자로 뒤에 구분자를 붙인 다음 반복횟수를 숫자로 명시한다. 'problem'문자열은 문제를 정의하는 지정자로 다음 줄부터 문제가 기술된다. 문제는 컴포넌트 종류, 순번, 문제 목록 순으로 구분자로 구분하여 기술한다. 'selection'문자열은 선택항을 정의하는 지정자로 뒤에 구분자를 붙인 다음 선택항 목록을 기술한다. 'answer'문자열은 정답 번호를 정의하는 지정자로 구분자 뒤에 위에서 기술한 각 문제의 순서에 맞춰 정답 선택항의 인덱스 번호를 기술한다. 표 3은 컴포넌트들의 내용과 문제은행을 포함한 디스크립터 파일의 작성 예이다.

표 3. 디스크립터 파일 내용의 예
Table 3. Example of the contents of descriptor file

1	초급\2단계\2번과 이름\문제-2\1
2	template_10
3	text
4	0\Arial\문제를 풀어봅시다
5	1\굵림\빈칸에 들어갈 정답을 아래에서 눌러주세요
6	question
7	repeat\5
8	problem
9	text\1\C\D\E\F\G\A\B
10	text\2\다\라\마\바\사\가\나
11	selection\도\레\미\파\술\라\시
12	
13	

III. 플루트 연주 학습 소프트웨어 개발

본 논문에서 구현한 소프트웨어의 흐름도는 그림 6과 같다. 소프트웨어는 객체지향 언어인 C++로 개발되었으며 MFC 라이브러리를 사용하였다. 먼저 메인 프레임 클래스는 CSplitterWnd를 이용하여 2개로 분할된 창을 구성하고 각 창에 콘텐츠 목록 클래스와 템플릿 뷰 클래스의 인스턴스를 생성하여 출력한다.

콘텐츠 목록 클래스의 인스턴스가 생성될 때 지정된 폴더 내에 존재하는 모든 디스크립터 파일들의 경로 정보와 각 디스크립터 파일에 저장되어 있는 학습과정 경로 정보를 읽어서 CTreeCtrl 클래스를 이용하여 콘텐츠 목록을 작성하여 창에 출력한다. 이후 사용자가 창에 출력된 콘텐츠 목록에서 원하는 학습단계 노드를 선택하면 해당 노드에 저장되어 있는 디스크립터 파일 경로를 템플릿 뷰로 전달한다.

템플릿 뷰는 학습과정 명을 보여주는 텍스트 컴포넌트와 템플릿을 보여주는 뷰 클래스, 그리고 학습과정 이동용 버튼 컴포넌트로 구성되어 있다. 템플릿 뷰는 전달받은 디스크립터 파일에서 지정하는 템플릿 클래스의 인스턴스를 생성한다.

템플릿 클래스는 컴포넌트 맵 클래스의 생성함수를 호출하여 템플릿에서 사용하는 컴포넌트 구조체들을 생성하며, 생성된 컴포넌트들로부터 사용자의 이벤트를 입력받고 템플릿에서 정의된 작업을 수행한다.

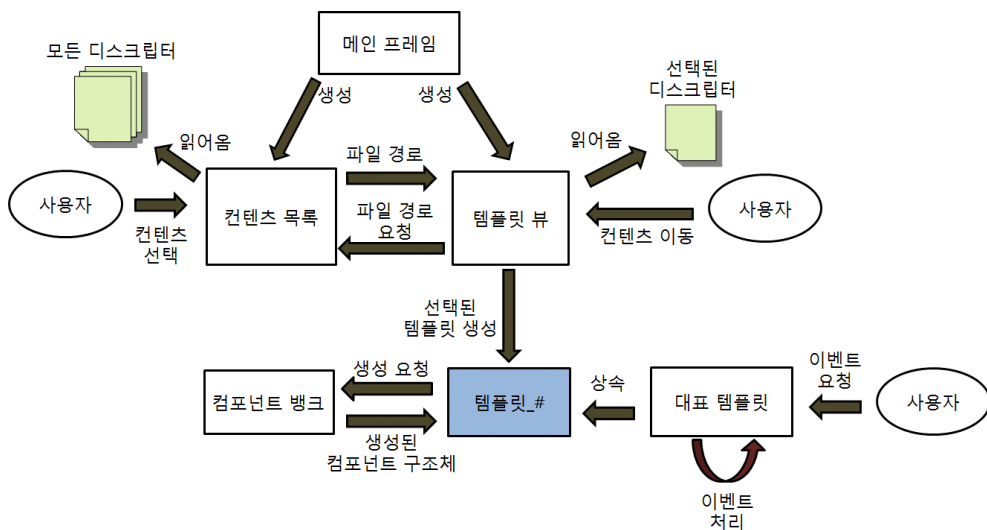


그림 6. 소프트웨어 흐름도
Fig. 6. Software flowchart

컴포넌트 백 클래스의 생성함수는 각 컴포넌트에 필요한 속성들을 컴포넌트 구조체로 생성하여 템플릿 클래스로 반환한다.

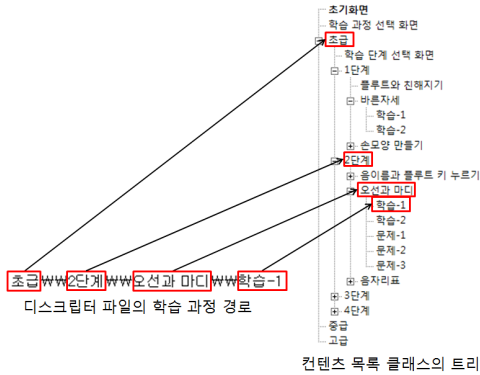


그림 7. 학습 과정 경로와 트리에 삽입된 노드들
Fig. 7. Learning procedure path and inserted tree nodes

1. 콘텐츠 목록 클래스

콘텐츠 목록 클래스는 생성될 때 지정된 폴더 내에 존재하는 모든 디스크립터 파일들을 열람하여 디스크립터 파일에 명시되어 있는 학습 과정 경로를 읽어와 콘텐츠 목록을 구성한다. 콘텐츠 목록은 트리형태로 구성되며 지식 노드들은 부모 노드의 세부 학습과정이 된다.

만일 학습 과정 경로에는 있지만 트리에는 없는 학습 과정이 있을 경우 부모 노드 위치를 찾아서 해당 위치에 새로운 세부 학습과정 노드를 삽입한다. 만약 학습 과정 경로의 모든 노드가 생성되어 더 이상 새로운 학습과정을 삽입할 노드의 위치를 찾을 수 없다면 현재 노드에 파일경로를 저장하고 마친다. 그림 7은 한 디스크립터의 학습 과정 경로와 각 학습과정을 통해 구성된 트리의 노드들을 표시한 것이다.

사용자가 콘텐츠 목록의 노드를 클릭할 경우 이벤트 처리 함수에서 노드에 저장되어 있는 디스크립터의 파일 경로를 템플릿 뷰 클래스로 전달하여 해당 콘텐츠가 출력되게 한다.

2. 템플릿 뷰 클래스

템플릿 뷰 클래스는 학습 과정 제목, 템플릿 뷰, 콘텐츠 이동 버튼으로 구성된다. 콘텐츠 목록 클래스로부터 디스크립터의 파일 경로를 전달 받으면 해당 디스크립터 파일의 내용을 문자열 배열로 모두 저장한다. 문자열 배열의 각 문자열은 디스크립터의 내용을 한줄씩 저장한다. 학습 과정 제목에는 디

스크립터의 학습 과정 경로를 읽어와 출력하고, 템플릿 뷰에서는 디스크립터의 템플릿 이름을 읽어와 해당 템플릿 클래스를 생성한다. 또한 생성된 템플릿 클래스에 디스크립터의 내용을 전달한다. 콘텐츠 이동 버튼은 현재 콘텐츠에서 이전 또는 다음 콘텐츠로 이동할 때 사용된다.

3 대표 템플릿 클래스와 유형별 템플릿 클래스

유형별 템플릿 클래스는 템플릿 종류에 따라 다르게 선언되어 있으며 모두 대표 템플릿을 상속받는다.

3.1 대표 템플릿 클래스

대표 템플릿 클래스에는 모든 템플릿 클래스에 공통으로 존재하는 변수와 함수들이 선언되어 있으며, 각 유형별 템플릿 클래스들은 대표 템플릿 클래스를 상속받아 사용한다. 벡터 변수에는 컴포넌트별로 생성된 컴포넌트 구조체가 저장되며 사용 용도별로 구분하여 선언되어 있다. 사용 용도에는 제시용, 질문용, 선택용이 있다. 제시용은 모든 컴포넌트로 사용될 수 있으며 단순히 컴포넌트의 내용을 디스크립터로부터 직접 채운다. 질문용과 선택용은 각각 문제와 선택지로 사용되며 디스크립터로부터 생성된 문제은행 구조체에서 선택적으로 내용이 채워진다. 질문용 벡터는 텍스트, 이미지, 가상 피아노, 가상 플루트가 있으며, 선택용 벡터는 버튼, 가상 피아노, 가상 플루트가 있다.

3.2 유형별 템플릿 클래스

유형별 템플릿 클래스에는 각 템플릿에서 사용되는 컴포넌트들의 속성이 매크로로 선언되어 있으며 초기화 함수에는 사용되는 각 컴포넌트의 생성함수를 호출한다. 또한 각 템플릿의 필요에 따라 서로 다른 이벤트 처리함수가 선언된다.

유형별 템플릿 클래스가 생성되는 과정은 그림 8과 같이 초기화 단계, 파싱 단계, 출력 단계로 구성된다. 초기화 단계에서는 컴포넌트 백 클래스의 생성함수를 통해 해당 템플릿에 필요한 컴포넌트 구조체들을 생성하여 벡터에 저장한다.

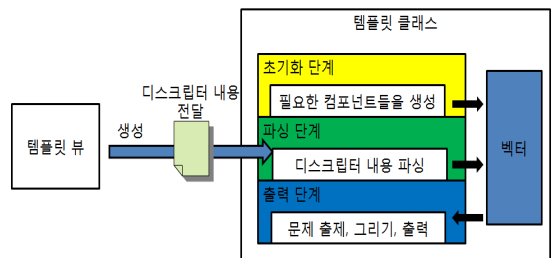


그림 8. 템플릿 클래스의 생성 과정
Fig. 8. Procedure of template class creation

파싱 단계에서는 각 컴포넌트에 해당하는 디스크립터 구문을 찾아서 디스크립터 내용을 인수로 하여 각 컴포넌트의 파싱함수를 호출한다. 각 컴포넌트의 파싱함수는 디스크립터의 내용 중 컴포넌트에 채워질 내용을 추출하여 컴포넌트들을 생성한다. 텍스트는 디스크립터내의 폰트와 내용, 버튼은 내용, 이미지와 동영상, 음성은 파일 경로를 생성에 사용한다. 출력 단계에서는 컴포넌트들을 현재 창 크기에 맞게 다시 그리며, 템플릿 내의 모든 컴포넌트들을 화면에 출력한다.

그림 9는 초기화 함수에서 생성함수를 호출하는 예를 보인다. 4개의 텍스트, 1개의 이미지, 2개의 영상을 생성하였고 ParseAll함수를 통해 디스크립터 내용을 파싱하였다. 마지막엔 Invalidate함수를 호출하여 템플릿을 출력하였다.

```
#define VIDEO0_SET 95,150,400,350
#define VIDEO1_SET 505,150,400,350
#define IMAGE0_SET 150,600,700,300
#define TEXT0_SET 220,50,560,100
#define TEXT1_SET 95,510,400,50
#define TEXT2_SET 505,510,400,50
#define TEXT3_SET 300,910,400,50

// CTemplate_6

// CTemplate_6 message handlers
int CTemplate_6::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CTemplates::OnCreate(lpCreateStruct) == -1)
        return -1;
    CreateVideo(VIDEO0_SET);
    CreateVideo(VIDEO1_SET);
    CreateImage(IMAGE0_SET);
    CreateText(TEXT0_SET);
    CreateText(TEXT1_SET);
    CreateText(TEXT2_SET);
    CreateText(TEXT3_SET);

    ParseAll();

    Invalidate();
    // TODO: Add your specialized creation code here

    return 0;
}
```

그림 9. 템플릿 클래스의 초기화 예

Fig. 9. Example of Initializing template class

4. 컴포넌트 बैं크 클래스

컴포넌트 बैं크 클래스에서는 각 컴포넌트와 연관되어 있는 속성들을 컴포넌트 구조체로 선언하고 컴포넌트 구조체를 생성하는 생성함수를 선언한다. 각 구조체는 기본적으로 컴포넌트를 포함하고 있으며 논리적 크기는 창 크기에 연동되는 컴포넌트의 크기를 말한다. 각 컴포넌트별 생성함수들은 표 2의 속성들을 인수로 입력받는다.

이와 더불어 문제, 문제은행, 문제 출제 구조체와 생성함수도 선언되어 있다. 문제 구조체는 문제가 출제될 컴포넌트의 타입, 문제용 컴포넌트의 인덱스, 출제될 문제 목록을 저장하는 문자열 벡터를 포함한다. 문제은행 구조체는 출제될 수 있는 문제들을 저장하는 구조체로 출제 반복 횟수, 문제 구조체

의 벡터, 선택항 목록을 저장하는 문자열 벡터, 그리고 각 문제에 대한 정답번호를 저장하는 정수형 벡터로 구성된다. 마지막으로 문제 출제 구조체는 문제은행에서 출제될 하나의 문제를 저장하는 구조체로 문제 인덱스, 정답번호, 정답 선택지 인덱스, 오답 목록을 저장하는 정수형 벡터로 구성된다.

5. 가상 플루트 클래스

플루트의 구조는 그림 10과 같으며 마우스 피스 홀을 부는 방향과 눌려진 키들의 조합에 따라 플루트의 연주가 달라진다. 본 논문에서 구현한 가상 플루트는 음이름과 옥타브 위치값이 입력되면 마우스 피스 홀을 부는 방향과 각 키의 누름 여부를 표시하고 PC MIDI를 이용하여 플루트 연주가 가능하도록 구현하였다.

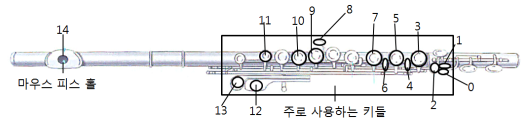


그림 10. 플루트의 구조
Fig. 10. Structure of flute

음이름과 옥타브 위치값에 따른 가상 플루트 제어 정보는 그림 11과 같이 16비트로 구성되며 lookup table로 작성하였다. 비트 15는 사용하지 않고, 비트 14는 마우스 피스 홀을 부는 방향을 나타내며 1이면 위, 0이면 정면으로 부는 것을 의미한다. 비트 13부터 비트 0은 각 키의 누름 여부를 나타내며 1이 누름을 의미한다. 그림 9의 숫자들은 비트의 자리를 의미하며 숫자와 연결된 선은 각 비트가 의미하는 부분을 가리킨다.

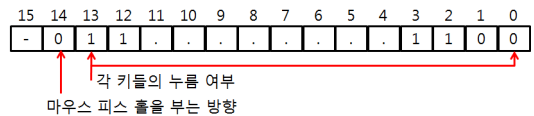


그림 11. 가상 플루트의 제어 정보
Fig. 11. Control information for the virtual flute

플루트는 3개의 옥타브 영역을 연주할 수 있으므로 컴퓨터 키보드를 가상 플루트의 입력수단으로 사용할 경우 Ctrl, Shift 키를 이용하여 구현할 수 있지만 사용이 편리하지는 않다[4,7,12]. 본 연구에서는 가상 피아노의 건반과 마우스를 이용한 입력방식을 구현함으로써 사용자 편의성을 향상시켰다. 마우스로부터의 입력 처리는 마우스의 버튼이 눌려진 상태와 마우스 커서의 위치를 이용하여 누른 건반을 검출한다.

건반 입력 이벤트가 발생하면 템플릿 클래스의 PressedNote 함수 혹은 ReleaseNote 함수를 호출하여 현재 누른 건반 혹은 선택해제된 건반의 번호를 템플릿 클래스로 전달한다.

6. MIDI 클래스

MIDI 제어모듈에서는 가상 플루트로부터 입력된 건반의 번호를 입력받아 눌러진 건반번호에 해당되는 음표번호와 채널, 속도, 음표 on 정보로 MidiOutShortMsg를 생성하고 PC의 MIDI를 제어하여 플루트 음을 출력한다[15]. 이전에 눌러진 건반이 선택해제되면 음표번호와 채널, 속도, 음표 off 정보로 MidiOutShortMsg를 생성하고 PC의 MIDI를 제어하여 플루트 음을 출력을 중지한다.

7. 문제 출제

문제 출제는 선택지의 개수를 인수로 갖는 GenQuest 함수를 사용하여 출제한다. GenQuest 함수는 문제은행 구조체로부터 표 3의 문제 출제 구조체를 생성하는 함수이다. 하나의 문제를 임의로 선택하여 해당 문제의 정답번호와 함께 문제 출제 구조체의 문제 인덱스, 정답번호에 저장한다. 또한 문제 출제 구조체의 정답 선택지 인덱스를 해당 템플릿의 선택지 범위 내에서 임의로 정한다. 문제 출제 구조체의 오답 선택지들은 문제은행 구조체의 선택항들 중 임의로 선택하여 채운다. 문제은행 구조체에 저장되어 있는 반복횟수만큼 중복되지 않는 문제들을 출제한다.

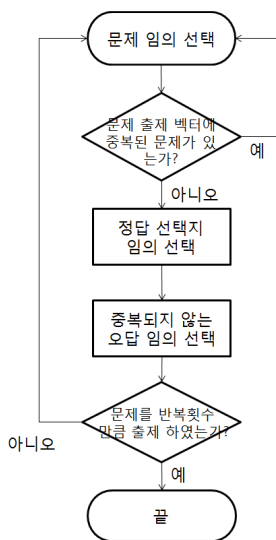


그림 12. 문제 출제 과정
Fig. 12. Procedure of setting questions

출제된 문제들은 문제 출제 벡터에 저장된다. 그림 12는 문제 출제의 순서도를 보인다. 선택지 컴포넌트에서 클릭 이벤트가 발생하면 출제 문제의 정답 선택지 인덱스와 버튼의 경우 이벤트 번호, 가상 악기의 경우 가상 악기 클래스로부터 전달받은 눌러진 건반번호와 정답 노트와 비교하여 같으면 정답, 다르면 오답으로 처리한다. 정답일 경우 다음 문제로 이동하거나 다음 학습 내용으로 이동하고, 오답일 경우 해당 문제를 다시 풀 수 있도록 기회를 제공한다.

III. 실험 및 결과

본 논문에서 제안한 디스크립터-템플릿 구현 방법을 활용하여 총 400여 페이지의 플루트 연주학습 콘텐츠를 소프트웨어로 구현하였다. 소프트웨어에 구현된 총 템플릿 수는 29개이며 작성한 디스크립터 파일 수는 233개이다. 프로그램의 크기는 249KB이고, 디스크립터 파일들의 총 크기는 42.6KB이다. 그림 13은 템플릿별로 사용된 디스크립터 수를 차트로 그린 것이다. 17번 템플릿이 174개로 가장 많은 디스크립터가 사용하였고, 7번 템플릿이 18개로 두 번째로 가장 많이 사용된 템플릿이다. 17번 템플릿은 악보를 보고 플루트를 연주하는 연습하기 형식의 콘텐츠에 사용되었고, 7번 템플릿은 플루트의 운지법을 연습하는 형식의 콘텐츠에 사용되었다. 실험 결과를 통해 플루트 연주연습과 관련된 콘텐츠 구현에서 개발 비용을 현저히 줄일 수 있다는 사실과 적은 수의 템플릿과 디스크립터의 조합으로 더욱 많은 학습 콘텐츠를 구현할 수 있음을 확인하였다.

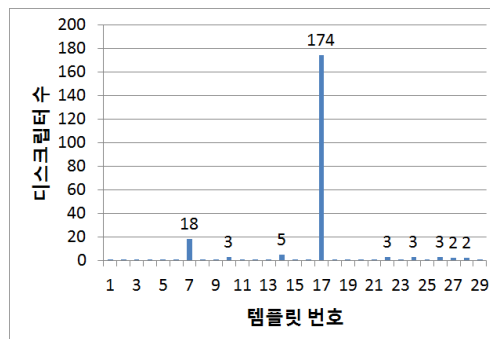


그림 13. 템플릿별 디스크립터 수
Fig. 13. The number of descriptors for each template

본 논문에서 제안한 디스크립터와 템플릿을 이용한 콘텐츠 생성을 직접 구현하여 그 결과를 분석하였다. 표 3의 디스크립터와 그림 4의 4지선다 문제형 템플릿을 이용하여 생성한

콘텐츠 결과들을 그림 14에 나타내었다. 그림 14의 각 그림들은 표 3의 디스크립터 내에 기술되어 있는 문제은행으로부터 출제된 서로 다른 문제들을 보인 것이다. 생성된 콘텐츠에서 문제의 정답을 선택하였을 때 정답임을 사용자에게 알리고 다음 문제가 출제되는 것을 확인하였고, 디스크립터의 문제은행 구문에 기술되어있는 반복횟수만큼 문제를 모두 풀었을 때 다음 콘텐츠로 이동하는 것을 확인하였다. 또한 오답을 선택하였을 경우 오답임을 사용자에게 알리고 문제를 다시 풀 수 있도록 기회를 제공하는 것을 확인하였다.



그림 14. 하나의 디스크립터에서 출제된 4개의 서로 다른 문제들
Fig. 14. 4 examples of question set by one descriptor

학습형, 학습/실습형, 연습형 템플릿을 이용하는 디스크립터를 이용하여 구현된 콘텐츠 생성 예를 그림 15에 보였다. 학습형에서는 텍스트와 이미지 컴포넌트가 사용되었고, 학습/실습형에서는 텍스트와 이미지, 버튼, 동영상상이 사용되었다.



그림 15. 학습형, 학습/실습형, 연습형 템플릿으로 생성된 콘텐츠들
Fig. 15. Example of contents using study, study/practice, exercise templates

연습형에서는 텍스트, 이미지, 가상 피아노가 사용되었다. 가상 악기가 있는 템플릿을 사용하는 '플루트와 친해지기' 콘텐츠의 구현 결과를 그림 16에 보였다. 가상 피아노는 옥타브 수는 3, 오른쪽 건반을 사용하고 시작 옥타브 위치는 5옥타브이다. 가상 피아노의 파란색 건반은 키보드나 마우스의 입력으로 현재 선택된 건반을 나타내는데, 그림 16은 옥타브 위치 6인 '솔' 건반이 눌리진 상태임을 보이고 있다. 가상 플루트의 회색영역은 운지 영역을 나타낸다. 가상 피아노의 건반을 눌렀을 때 플루트의 음이 스피커를 통해 올바르게 출력되는 것을 확인하였다. 그리고 가상 플루트의 마우스 피스 홀에 표시된 초록색 화살표 방향과 플루트 키들의 누름 정보가 정확하게 표시되는 것을 확인하였다.



그림 16. '플루트와 친해지기' 콘텐츠
Fig. 16. Example of "Being familiar with flute" contents

본 논문에서 개발한 학습 소프트웨어의 학습효율을 검증하기 위해 플루트에 대한 기초지식이 없는 대학생 10명을 대상으로 플루트 연주 학습 실험을 실시하였다. 비교 실험을 위해 무작위로 5명씩 A, B그룹으로 구분한 뒤, A그룹은 플루트 교본 책자를 사용하여 운지법을 학습하였고, B그룹은 본 논문에서 제안한 학습 소프트웨어를 사용하여 학습하였다. 그리고 각 학습자가 플루트 불기를 익힌 뒤, 운지학습 시작부터 동요인 "종이비행기"를 플루트로 완주할 수 있을 때까지의 소요시간을 측정한 소요시간을 표 4에 나타내었다.

표 4. 각 그룹 학습자별 학습 소요 시간
Table 4. Flute learning time for each group member

	A그룹	B그룹
학습자 1	10분 41초	09분 22초
학습자 2	16분 37초	08분 21초
학습자 3	15분 45초	09분 35초
학습자 4	10분 50초	08분 25초
학습자 5	14분 20초	12분 50초
평균	13분 39초	09분 43초

A, B그룹의 평균 소요시간은 각각 13분 39초, 9분 43초이다. 실험결과를 통해 본 논문에서 제안한 학습 소프트웨어를 사용한 B그룹이 A그룹보다 평균 4분 더 빠르게 플루트로 동요를 완주할 수 있었음을 확인하였다. 초등/중등/고등학교에서 음악교육을 받은 대학생에 비해 초등학교 저학년 학생들의 경우에는 학습 소요 시간의 차이가 더욱 크게 나타날 것이라 예상된다.

V. 결론

본 논문에서는 관악기인 플루트를 대상으로 연주 자율학습 소프트웨어를 제안하였다. 교육 콘텐츠들을 디스크립터와 템플릿으로 설계함으로써 개발 비용의 절감과 학습 지도자의 독립적인 프로그램 제어를 가능하게 하였다. 또한 여러 컴포넌트들을 구현하고 이를 활용하여 템플릿을 구현함으로써 사용하기에 쉬운 인터페이스를 제작하였다. 연습문제 출제는 문제은행에서 출제하는 방법으로 설계하여 학습 지도자가 직접 문제를 만들 수 있도록 하였고 하나의 문제은행으로 다양한 문제들을 생성할 수 있도록 임의 선택 출제방식으로 구현하였다.

실험에서 실제 구현한 소프트웨어의 효율성과 인터페이스의 편리함, 문제은행을 통해 출제된 연습문제의 다양성을 확인하였고 악기 연주 학습 소프트웨어의 가능성을 확인하였다.

본 논문에서 제시한 방법은 그 가능성을 제시하기 위한 것으로 아직 보완이 필요하다. 향후 외장형 전자 플루트와 연동을 위한 연구를 통해 플루트 연주 학습 프로그램의 효율성을 향상시킬 예정이다.

VI. 감사의 글

이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2012-0004681)

참고문헌

- [1] Peter Miksza, "Music Participation and Socioeconomic Status as Correlates of Change: A Longitudinal Analysis of Academic Achievement," Bulletin of the Council for Research in Music Education, No. 172, pp. 41-58, 2007
- [2] C.J. Bonk, and K.S. King, "Electronic Collaborators: Learner Centered technologies for Literacy, Apprenticeship, and Discourse," Lawrence Erlbaum Associates Publishers, pp. 25-50, 1998.
- [3] P.S. Jang, "Relationships among Learner's Satisfaction, Self-Directedness and Flow Experience in Computer e-Learning," Journal of The Korean Society of Computer and Information, vol.17, no.2, p69-77, Feb. 2012.
- [4] Cakewalk SONAR, <http://www.cakewalk.com/>
- [5] Steinberg Cubase 7, <http://www.cubase.com/>
- [6] Steinberg Nuendo, <http://www.steinberg.net/>
- [7] Finale&SmartMusic, <http://www.makemusic.com/>
- [8] J.S. Park, "Study on the teaching of creation by using the Nuendo music program," Mater Thesis, The graduate school of education Chonnam National University, 2011.
- [9] Practica Music, Ars-nova software, <http://www.ars-nova.com/>
- [10] Smart music, <http://www.smartmusic.com/>
- [11] G.J. Lim and J.C. Lee, "An Electronic Keyboard Instrument Using PC MIDI and USB Interface," Journal of The Korean Society of Computer and Information, vol.16, no.11, p85-93, Nov. 2011.
- [12] D.S. Kim, "A study on the development of CAI program for learning woodwind instruments," Mater Thesis, The graduate school of education Catholic University of Daegu, 2004.
- [13] S.G. Park, H.S. Jang, J.K. kwan and K.H. Park, "The Design and Implementation of a Web-Based contents Management system for Self-Directed Learning Method," Summer Conference of The Korean Society of Computer and Information, vol. 18, no. 2, p503-506, Jul. 2010.
- [14] M.H. Kim, "Web-based E-learning System Supporting an Effective Self-directed Learning Environment," Journal of the Korea Contents Association, vol. 11, no. 9, p524-535, Sep. 2011.
- [15] P. Messic, "Maximum MIDI: Music Applications in C++," Manning Publications, 1998.

저 자 소 개



김 재 영
 2006~현재: 울산대학교
 컴퓨터정보통신공학부
 석사과정
 관심분야: 디지털영상처리
 Email : k jy7097@naver.com



이 정 철
 1984년 : 서울대학교
 전자공학과 공학사
 1988년 : 서울대학교
 전자공학과 공학석사
 1998년 : 서울대학교
 전자공학과 공학박사
 현 재: 울산대학교
 전기공학부 부교수
 관심분야 : 디지털신호처리,
 음성신호처리, 음성합성
 Email : junglee@ulsan.ac.kr



전 희 성
 1981년 : 서울대학교
 전기공학과 공학사
 1983년 : 서울대학교
 제어공학과 공학석사
 1992년 : 미국 Rutgers - The State
 University of New Jersey
 컴퓨터공학과 공학박사
 현 재: 울산대학교
 전기공학부 교수
 관심분야 : 디지털영상처리
 컴퓨터비전, 컴퓨터그래픽스
 Email : hsjun@ulsan.ac.kr