

오픈 소스 기반의 이러닝 평가문항 저작 도구를 위한 템플릿 유지 기법

한성재*, 최병욱**, 차재혁***

요약

특정 표준에 대한 사전 지식 없이 콘텐츠를 저작할 수 있는 템플릿 기반의 이러닝 저작 도구가 활발히 개발되고 있다. 그러나 일반적인 웹 기반 HTML 저작도구에서 템플릿을 사용하기 위해서는 중요한 극복 사항이 있다. 콘텐츠 내부에 템플릿이 삽입 되거나, 이를 편집 과정 중에 사용자의 의도치 않은 입력으로 템플릿 구조가 변형이 될 수 있다. 이것은 제한된 스키마를 준수해야하는 표준 콘텐츠 저작에 있어서 큰 위험 요소가 된다. 본 논문은 위지위 기반의 HTML 저작 도구에 삽입되는 템플릿 자원을 사용자가 임의로 변형하는 것을 막을 수 있는 효과적인 유지 기법을 제안한다. 우리는 웹 기반 오픈 소스 에디터인 tinyMCE에 IMS-QTI 표준 문항을 생성할 수 있는 템플릿 플러그인을 추가하였다. 그리고 사용자의 의도치 않은 입력 행위에 대응하여 템플릿의 구조가 지속적으로 유지 가능하게 하는 tinyMCE용 플러그인 모듈을 구현하였다. 제안하는 기법은 tinyMCE 이외에도 다양한 오픈 소스 기반의 편집 도구에 적용 가능하다는 장점이 있다.

키워드 : 이러닝, 저작 도구, 콘텐츠 템플릿, 오픈 소스, QTI

A method to maintain templates in open source-based authoring tool for e-learning assessment items

Sungjae Han*, Byung-Uk Choi**, Jaehyuk Cha***

Abstract

Existing tools using in the standard e-learning contents authoring were used a method to provide users contents template produced in advance. In order to use resources of the template in a common web-based authoring tool, there is problem to overcome. If the resource of template is inserted within the contents on the authoring tool, the deformation of the template by the user's input that may occur during the edit process cannot be controlled. In this paper, we propose an effective maintenance method to prevent deformation of the resource of template inserted into any WYSIWYG-based HTML authoring tool by user's discretion. We added a template plug-in that can create the IMS-QTI standard resource in tinyMCE the web-based open source editor of representative examples. And the plug-in for tinyMCE was realized as a module of directly respond to the action of limited user input. So, in response to the action of user's input, the structure of the template can be sustained possibly.

Keywords : e-Learning, authoring tool, content template, open source, QTI

1. 서론

※ 교신저자(Corresponding Author): Jaehyuk Cha
접수일:2014년 02월 11일, 수정일:2014년 02월 2일
완료일:2014년 02월 27일

* 한양대학교 전자컴퓨터통신공학과

Tel: +82-2-2220-4363, Fax: +82-2-2220-1723
email: sjhans@hanyang.ac.kr

** 한양대학교 컴퓨터공학과

현재까지 공개된 콘텐츠 저작 도구는 자유 저

*** 한양대학교 컴퓨터공학과

■ 본 논문은 중소기업청에서 지원하는 2012년도 산학연공동기술개발사업(No. 000-45394)의 연구수행으로 인한 결과물임을 밝힙니다.

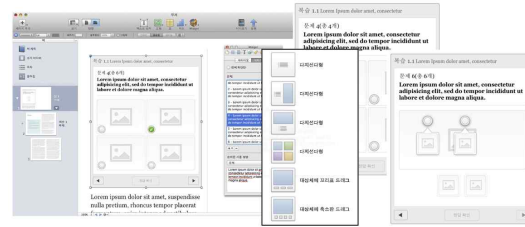
작 방식과 템플릿을 이용한 저작 방식으로 나누어진다. 자유 저작은 콘텐츠의 구조나 구성에 제한 없이 사용자가 전체 영역의 편집을 임의대로 진행한다. 이를 통해서 사용자는 자신의 의도에 따라 자유롭게 콘텐츠를 구성할 수 있다. 하지만 이 방식은 저작 도구의 사용법과 콘텐츠 저작에 대한 경험 및 사전 지식이 요구되기 때문에 복잡한 규약을 준수해야 하는 표준형 콘텐츠 저작에는 널리 사용되지 않는다.

템플릿을 이용한 콘텐츠 저작 방식은 사전에 미리 정의 되어있는 자원 및 구성 방안을 저작 도구 상에 호출하고 사용자는 적은 노력을 통해서 이를 재가공하여 콘텐츠를 생산한다. 따라서 많은 수의 콘텐츠를 저작하거나 반복적인 양식이 콘텐츠에 적용되어야 할 경우에는, 자유 저작 방식보다 템플릿을 이용한 저작 방식이 사용자에게 효과적이다. 그리고 사용자는 콘텐츠 저작 경험이 많지 않더라도 일정 수준 이상의 품질이 보장된 콘텐츠 생산이 가능하기 때문에, 대부분의 콘텐츠 편집 도구는 다양한 템플릿을 사용자에게 제공하고 있으며, 일부 벤더의 경우 별도의 템플릿을 상업적으로 판매하기도 한다[1, 2, 3].

현재 상용 콘텐츠 저작도구는 대부분 템플릿 기반의 편집 기능을 지원한다. 그리고 이러닝 분야에서도 템플릿을 기반으로 하는 다양한 콘텐츠 저작 활동이 이루어지고 있다[4]. 템플릿 기반의 저작 도구 중 대표적인 예시로 Apple사의 iBook Author가 있다. iBook Author는 누구나 손쉽게 전자책이나 이러닝 콘텐츠를 만들 수 있는 iOS 기반 어플리케이션이다[5]. 이 어플리케이션은 (그림 1)에서 볼 수 있듯이 콘텐츠 자원의 구성 형태를 사전에 정의 하여 사용자가 다양한 콘텐츠 템플릿을 사용할 수 있도록 하였다.

이외에도 다양한 콘텐츠 저작용 어플리케이션이 보급되고 있다. 하지만, 현재 웹 콘텐츠의 대표적인 생산 도구는 웹 기반 HTML 편집기이다[6]. 대부분의 사용자는 웹 기반 HTML 편집기를 이용하여 커뮤니티나 블로그용 콘텐츠를 생산한다. 웹 브라우저 상에서 동작하는 HTML 편집기는 접근성이 매우 우수하다. 그리고 HTML의 특성상 다양한 어플리케이션에서 손쉽게 저작이 가능하기 때문에 현재 대다수의 CMS(Content Management System)들이 웹 기반 HTML 편집기를 내장하고 있다. 그러나 브라우

(그림 1) iBook Author의 템플릿을 이용한 콘텐츠 저작 화면 예시



(Figure 1) An example of content authoring with templates in the iBook Author

저에서 동작하는 HTML 편집기는 어플리케이션의 특성상 사용자의 편집 행위에 대한 제한이 어렵다. 자바나 MFC 등의 언어로 구현된 저작 도구에 비해서, 제공되는 템플릿의 구조와 구성에 대한 사용자 변경 행위의 통제가 어렵기 때문에 매우 제한적인 요소에만 템플릿을 활용하였다[7].

템플릿은 반복되는 생산 작업을 줄이기 위해 사용될 뿐만 아니라, 일반인들에게는 복잡하며 다루기 어려운 표준이 적용된 자원들을 생산하는 경우에 효과적으로 사용된다[8]. 최근 이러닝 분야에서는 콘텐츠의 상호운용성과 접근성을 위한 국제 표준 활동이 활발히 이루어지고 있다. 대표적인 사례가 평가 문항에 대한 콘텐츠 구성과 그에 제반되는 시스템에 관하여 세부 규격을 기술한 사실상의 국제 표준인 QTI(Question and Test Interoperability)이다. QTI는 학습 평가용 콘텐츠의 메타데이터를 규정하고 있다. 이 표준은 데이터의 구성 형태에 대하여 매우 엄격한 유효성 검사가 동반된다. XML과 XSD로 구성된 표준의 성격상 일반 콘텐츠 저작자는 그 내용에 대한 이해가 어렵고, 저작 과정이 익숙지 않기 때문에, 주로 템플릿 기반의 QTI 저작 도구를 활용하여 콘텐츠를 생산한다[9].

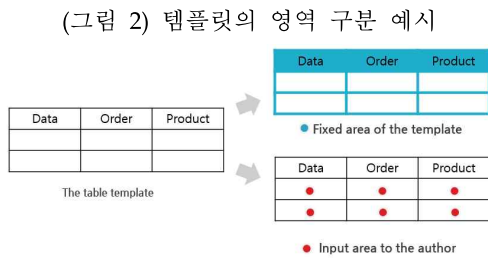
그동안 웹 기반 HTML 편집기에서 QTI 표준을 적용한 템플릿을 활용하는 많은 연구 사례가 있었다[6, 10]. 그러나 기존 연구에서는 문항 저작시 발생하는 템플릿 변형에는 대응을 하지 않았다. 이러한 콘텐츠 편집기가 실제 상용 서비스로 제시되기 위해서는 사용자에게 의한 템플릿의 구조 변경에 대해서 대응이 요구 된다. 따라서 본 논문은 특정 표준을 준수하는 템플릿이 웹

기본 HTML 편집기에서 그 형태와 구조가 지속적으로 유지될 수 있는 새로운 시스템 구성 방법을 제안한다.

2. 관련 연구

2.1 콘텐츠 저작에서 사용되는 템플릿

템플릿(Template)이란, 사용자의 콘텐츠 저작시 어려움과 번거로움을 줄이기 위해서 사전에 특정 양식에 따라 구성해 놓은 자원의 집합을 의미한다. 템플릿을 지원하는 편집기는 콘텐츠의 일부 혹은 전체의 구조와 형태를 사전에 생산하고, 사용자는 한정된 정보를 템플릿 내에 입력하여 콘텐츠를 저작한다. 템플릿은 콘텐츠의 성격에 따라서 그 구조 혹은 구성 변경의 허용 여부가 결정된다. 만약 구성 변경이 허용되지 않는다면 사용자는 템플릿 내의 데이터 입력 이외에는 다른 작업을 수행할 수 없다[1, 2].



(Figure 2) An example of template area divisions

(그림 2)는 3행 3열의 구조와 실선으로 이루어진 테이블 템플릿의 예시이다. 템플릿은 고유 영역과 사용자 입력 영역으로 나뉘어진다. 템플릿 고유 영역은 사전에 정의된 요소로써 편집이 허용되지 않는 자원들로 구성된다. (그림 2)는 3행 3열의 테이블에 별도의 열이 추가 혹은 삭제되지 않아야 한다는 가정에서 템플릿이 제공된다. 또한 첫 번째 행의 각 열에 기입된 단어도 고유 영역이다. 따라서 사용자는 원으로 표기된 텍스트를 삽입하는 행위가 허용된다. 필요에 따라서는 행이 추가될 수 있으나, 3열의 테이블 구조는 유지되어야 한다. 유사 예시로서 가장 대표적인 문서는 이력서와 업무 보고서이다. MS-Word와 한글 등의 문서 편집기는 사전에

문서 양식을 사용자에게 제공한다. 사용자는 템플릿으로써 제공된 문서 양식을 이용하여 보다 쉽게 콘텐츠를 저작할 수 있다. 제공되는 템플릿은 사용자의 의도에 의하여 그 구조나 구성이 얼마든지 수정 될 수 있다.

그러나 특정한 구조나 형태를 반드시 준수해야 하는 공문서의 경우는 템플릿의 변경을 허용해서는 안된다. 이력서 양식은 사용자가 얼마든지 그 형태를 변경할 수 있지만, 일부 기업이나 공공 기관에서는 이력서 자체의 구성 변경을 허용하지 않는 경우도 있다. 콘텐츠 구성 항목의 순서나 배치 그리고 문서 내의 자원 규격들이 법령이나 내규로써 제한 될 수 있다. 이처럼 구성 변경이 제한적인 템플릿을 제공하고자 할 경우에는 사용자의 편집 행위에 대하여 별도의 제한 수단이 필요하다. 또한 이러한 템플릿이 편집 과정에서 사용자에게 의하여 변형 된다면 사용자에게 변경 사실을 알려주거나 자동으로 교정 작업이 수행 될 수 있는 저작 도구가 필요하다.

2.2 IMS-QTI의 콘텐츠 템플릿

QTI(Question and Test Interoperability)는 이러닝 콘텐츠 중에서 평가 문항의 상호 운용성을 보장하기 위하여 IMS Global에서 제정한 사실상의 국제 표준 규격(de-facto standard)이다. QTI는 콘텐츠를 구성하는 자원과 이를 명시한 메타데이터의 세부적인 스키마를 정의한다. 또한 이러닝 시스템의 각 세부 영역에서 콘텐츠의 구체적인 운영 방법에 대해 명시하고 있다[9, 11, 12].

QTI 콘텐츠의 구조적인 내용 표현 방식은 XHTML과 유사하다. 실제로 XHTML에서 사용되는 객체와 속성 규칙을 QTI에서 대부분 그대로 모방하고 있다. 사용자가 시각적으로 확인 가능한 영역은 XHTML 유사 요소로 표현하고, 채점과 피드백 등의 영역은 독자적인 스키마로 정의한다.

QTI의 시각적 표현 중에서 가장 대표적인 요소는 사용자 인터랙션이다. QTI는 표 1과 같이 다양한 사용자 인터랙션 모델을 명시하였다. 기존에 공개된 QTI 기반의 콘텐츠 저작 도구는 평가 문항의 답안 선택 유형인 사용자 인터랙션 영역을 템플릿으로 제공한다[13, 14].

‘Choice’는 일반적인 객관식 유형을 의미한다.

‘Order’는 화면에 표시된 선택 항목을 평가 응시자가 임의로 순서를 지정하여 재배열하는 유형이다. ‘Associate’와 ‘Match’ 그리고 ‘Gap Match’는 연관성이나 동질성이 있는 선택 항목들을 서로 연결해주는 인터랙션을 제공한다. ‘Text Entry’와 ‘Extended Text’는 주관식 유형으로써 정해진 질문에 대하여 사용자가 단답형 혹은 서술형으로 답안을 작성한다. ‘Slider’는 수량이나 수치를 답안으로 제출 할 때, 숫자를 직접 기입하지 않고 사용자가 슬라이드 바를 마우스로 드래그 하여 수량을 표기하는 방식이다. 이 유형은 ‘Text Entry’나 ‘Choice’로 대체될 수 있지만 사용자에게 보다 다양한 문항 인터페이스를 제공하기 위한 장치로써 활용되고 있다. ‘Inline Choice’는 한정된 특정 위치에서 펼쳐지는 리스트 형태의 목록 중에서 사용자가 원하는 답안을 선택하는 유형이다. ‘Hot Text’는 문단이나 문장 내에서 골고루 분산되어있는 텍스트 중에 일부를 선택한다. 이외에도 이미지나 아이콘을 이용한 다수의 인터랙션 유형이 존재한다. ‘Hotspot’과 ‘Position Object’는 특정 이미지 상에서 아이콘을 특정 위치로 배치하여 답안을 작성하는 유형이다. 그리고 ‘Select Point’는 이미지 상의 특정 지점을 마우스로 선택한다. 단일 이미지 내의 특정 두 좌표를 이어서 관계를 표시하는 ‘Graphic Associate’, 특정 두 좌표에 아이콘 객체를 각각 배치하여 관계를 표시하는 ‘Graphic Gap Match’ 방식도 존재한다. QTI는 이러한 응답 유형 외에도 파일 업로드와 사용자 정의형 응답 유형 등이 존재한다. 또한 한 개의 평가 문항에는 다수의 인터랙션을 함께 사용할 수 있으며, 인터랙션의 수에 따라서 Simple Item과 Composite Item으로 분류 된다.

QTI 콘텐츠를 위한 저작도구는 사용자 응답 유형에 대한 양식을 템플릿으로써 제공한다. (그림 3)은 앞서 명시된 인터랙션 중에서 ‘Choice’ 템플릿을 지원하는 저작 도구의 예시이다. 저작자는 제시된 템플릿의 특정 필드에 문항과 관련된 정보를 기입함으로써 평가 문항을 저작할 수 있다. 예시에서는 HTML 편집 영역이 존재하지만, 이는 콘텐츠의 한정된 요소만을 위해 존재한다. 템플릿은 HTML 편집기 외에 별도의 필드 형태의 입력 영역으로 표시된다. HTML 편집기 내에 템플릿이 존재할 경우 사용자의 입력에 따

<표 1> IMS-QTI 문항 인터랙션 예시

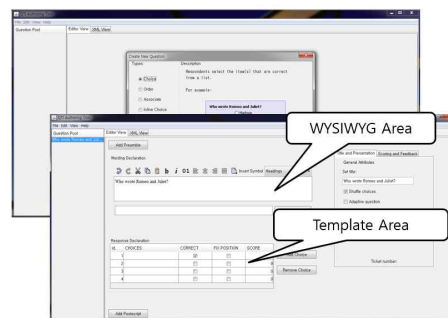
Choice	Order
Match	Inline Choice
Extended Text	Hot Text
Select Point	Gap Match
Graphic	Graphic
Associate	Gap Match
Slider	Associate
Text Entry	Hotspot
Graphic Order	Position Object

<Table 1> Some interaction types of IMS-QTI items

라 템플릿이 변형 될 우려가 있기 때문에 기존 저작 도구들은 템플릿을 별도의 위치에 분리하여 제공하였다[15].

저작 도구를 통해 생성되는 콘텐츠의 모든 메타데이터는 IMS-QTI에서 사전에 정의한 XML 스키마를 준수해야 한다[16]. 저작도구의 템플릿은 사용자가 복잡하고 어려운 스키마를 사전에 알지 못하더라도, 표준 규격에 맞는 평가 문항을 저작할 수 있도록 효과적인 기반 환경을 제공한다. 그리고 평가 문항의 응답 유형에 따라서 복잡한 프로그램 모듈이 콘텐츠에 삽입되는 경우가 있다. ‘Hotspot’이나 ‘Graphic Order’와 같이 사용자가 특정 객체를 조정하는 응답 유형에 사용되는 템플릿은 별도의 객체 제어용 모듈이 함께 내장된다. 기존의 템플릿들은 이와 같은 기능을 모두 제공하기 때문에 사용자의 콘텐츠 저작에 필요한 노력과 사전 요구 지식을 최소화할 수 있다.

(그림 3) QTI 문항 저작도구의 분리된 영역



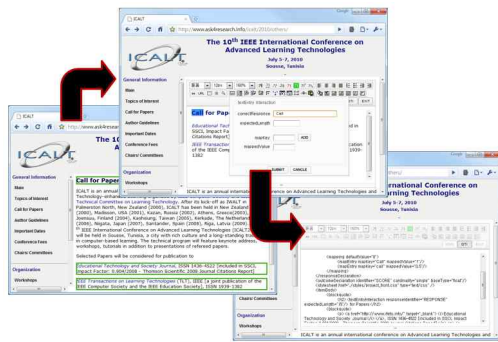
(Figure 3) Separate areas in a authoring tool for QTI items

3. 템플릿 구조 유지 기법

3.1 템플릿 변형 사례의 분석

본 연구는 콘텐츠 저작 과정에서 발생 가능한 예외 사항을 분석하기 위하여 실제 사용자들의 저작 과정을 관찰하였다. 관찰 대상은 컴퓨터 전공의 학부생 및 대학원생 16명이며, 템플릿을 이용한 콘텐츠 저작 과정 중에서 그 구조가 변형되는 상황을 기록하고 분류하였다. 실험에 사용한 콘텐츠 저작 도구는 QTI 평가 문항을 생성할 수 있는 dinsEditor를 사용하였다[10]. dinsEditor는 QTI와 호환 가능한 템플릿의 사용이 가능한 저작 도구로써 웹 기반 HTML 편집 도구이다. 이 저작 도구는 브라우저 확장 (Browser Extension)의 형태로 제공되는 독특한 구조를 갖고 있으며, 브라우저를 이용하여 사용자가 방문 중인 웹 페이지 내에 일시적으로 저작 도구를 동적으로 삽입한다[10, 17]. (그림 4)는 dinsEditor를 이용하여 사용자가 방문한 웹 사이트에 저작도구를 직접 삽입하여 사이트 내의 자원을 이용해서 평가 문항 콘텐츠를 동적으로 생성하는 과정을 나타낸다.

(그림 4) dinsEditor의 사용자 인터페이스



(Figure 4) User interface of dinsEditor [10]

현재 오픈 소스로 배포되고 있는 많은 HTML 콘텐츠 편집기는 템플릿과 일반 자원에 대한 다양한 편집 방식이 존재한다. 따라서 발생 가능한 모든 사용자의 입력과 템플릿의 변형 사례에 대해서 수집 및 분석 과정이 요구된다.

우리는 사용자의 변형 사례를 의도하지 않은 변형과 의도된 변형으로 나누어 분류하였다. 의도되지 않은 변형은 사용자의 입력 과정 중에서

자연스럽게 발생한 상황으로써 편집이나 키 입력 실수에 의하여 템플릿 내의 자원이 영역 밖으로 이동되거나 삭제되는 등의 사례를 통칭한다. 의도된 변형은 스트레스 테스트의 일종으로 무리한 키 입력과 같은 방법을 활용하여 저작 도구 내에서의 상황 대처가 얼마나 효과적으로 진행되는지를 판단한다. 템플릿을 이용한 저작 과정에서 발생하는 모든 변형 상황에 대해서 사용자와 관찰자가 직접 세부 목록을 작성하였다.

<표 2>는 주관식 유형의 템플릿이 사용자의 입력에 의하여 그 구조가 변경된 사례이다. 예시 이미지 내에서 점선 영역이 템플릿이고, 'Korea'라는 텍스트가 저작자가 입력한 답안이다. 2-1은 사용자가 템플릿을 삽입하여 문항을 완성한 정상적인 상태이다. 2-2는 템플릿 내에 삽입된 텍스트의 일부를 백스페이스키를 이용하여 삭제하였을 경우 템플릿으로 감싸진 객체가 삭제되는 사례이다. 2-3은 WYSIWYG 방식의 HTML 편집기에서 제공하는 외부의 서식을 삽입 할 경우 템플릿이 두 개로 분리된 예시로써, 이는 분리된 객체가 서로 중복되는 식별자를 갖는 문제점이 함께 발생하였다. 마지막으로 예시 2-4는 저작자가 마우스로 객체 일부를 선택하여 다른 곳으로 이동하거나 콘텐츠 내에서 강제 개행 문자를 입력 한 경우, 템플릿이 분리되는 사례를 나타낸다. 이 경우도 2-3과 마찬가지로 식별자가 중복되는 문제가 발생한다.

<표 2> 주관식 템플릿의 변형 사례

Situations in a authoring tool	
2-1 Default	
2-2 Delete	
2-3 Duplicate1	
2-4 Duplicate2	

<Table 2> Examples of broken templates in a subjective question

상기 예시는 사전에 정의된 템플릿이 사용자의 행위에 따라 의도치 않게 변형된 사례 중 일부이다. 입력 장치의 구분과 템플릿의 유형 그리고, 저작도구에서 다루지는 HTML 객체에 따라 다양한 템플릿 변형 사례가 사용자 관찰을 통해 도출되었다.

마우스 드래그를 이용하여 템플릿의 전체 혹은 일부 요소를 이동할 경우 발생할 수 있는 예외 사항 목록은 다음 <표 3>과 같다. 템플릿의 오류를 사용자가 직관적으로 확인할 수 있는 예시 항목은 <표 3>의 {1, 2, 3, 4, 6, 10}이다. 이외의 항목은 HTML 소스 코드 상에서 내부적인 변화가 이루어지기 때문에 콘텐츠 저작자는 오류를 인지하는 것이 어렵다. 특히 {7, 8, 9}번에서 발생하는 식별자 중복 문제는 유효성 오류를 발생시킨다. 따라서 콘텐츠의 저장이나 배포 이전의 단계에서 식별자 교정 작업이 선행되어야 한다.

<표 3> 마우스 드래그로 인한 템플릿의 변형 예시

Examples of broken templates on DOM (Document Object Model)	
1	A template was left alone without internal elements.
2	A template tag was left without a unique attributes such as " <i>identifier</i> ".
3	An error occurred in debug mode.
4	The duplication of objects was deactivated.
5	An unique attribute was removed when it was duplicated.
6	A template was divided. And its unique attribute was removed.
7	Unique attributes in some templates were overlapped. And each templates shared same elements.
8	Unique attributes in some templates were overlapped. And unwilled elements were modified.
9	Unique attributes were overlapped.
10	An object was inserted in templates without intended duplications.

<Table 3> Examples of broken templates in using the mouse drag

모든 실험자 대상자는 구글 크롬(Chrome)과 인터넷 익스플로러 그리고 파이어폭스 브라우저 중 자신에게 가장 익숙한 브라우저를 선택하여 콘텐츠 저작을 수행하였다[18, 19, 20]. 이 과정에서 사용자의 브라우저에 따라서 변형 예시에 차이가 있다는 사실이 발견되었다. 이 차이는 웹 에디터의 기본적인 크로스 브라우징 문제로 판단되었다. 이것은 템플릿이나 플러그인 모듈 상의 문제가 아닌, 저작 도구 자체의 특성에서 비롯된 문제이기에 본 논문에서는 해당되는 항목을 다루지 않는다. 이 항목은 모두 클립보드를 경유하는 콘텐츠 삽입과 개행 문자로 인한 다른 객체의 이동에 관한 사항들이다.

3.2 템플릿 구조 유지 기법의 제시

본 연구에서는 사전에 분류된 변형 사례에 대응할 수 있는 사전 방지와 사후 조정 기법을 제시한다. 사전 방지는 특정한 키의 입력을 무효화함으로써 콘텐츠의 변형을 원천적으로 예방한다. 사용자의 편집 행위를 식별하기 위해서 커서와 마우스 포인터의 위치 그리고 현재 저작도구에서 편집중인 HTML 자원의 노드를 실시간으로 추적한다. 사전 방지 기법은 특정 HTML 노드 상에서 사용자의 제한적인 입력이 발생할 경우 이 입력 자체를 무효화하기 위해서 이벤트 바인딩을 일시적으로 해제한 뒤 다시 복원하는 방법을 취하였다. 그러나 템플릿 이외의 자원 편집에 영향을 줄 수 있기 때문에 적용 범위는 다소 제한적이다. 사용자의 입력 이벤트를 감지하여 사전 방지로 대응하지 못한 템플릿 변화는 사후 방지 영역에서 다룬다. 사후 방지는 템플릿의 구조나 구성이 변형 된 이후에 브라우저의 클립보드 영역의 데이터를 직접 접근하여 입력 행위의 실행 취소(Undo / Backward)나 코드 교정을 수행한다. 사용자의 입력에 따라 콘텐츠는 지속적으로 변경되는데, 사후 방지는 이러한 업데이트마다 실시간으로 내부 요소를 점검하여 변경된 요소를 감지한다.

사전 방지와 사후 조정을 수행하는 시스템 모듈은 저작 도구 내에 위치하며, 전자는 에디터의 외부 스크립트로 참조되어 사용자의 입력을 직접적으로 제어하고, 후자는 에디터의 플러그인으로써 사용자의 입력 이후의 변화를 감지하는 역할을 수행한다.

우리는 (그림 5)와 같이 사전 방지와 사후 조정을 처리하기 위한 모듈을 플러그인 아키텍처에 따라서 오픈 소스 기반 HTML 에디터에 삽입하는 방식을 함께 제안한다.

3.2.1 오픈 소스 HTML 에디터의 구성

웹 환경에서 동작하는 HTML 에디터는 DOM(Document Object Model)을 바탕으로 사용자의 콘텐츠 편집 과정을 웹 페이지의 한정된 영역에 출력한다. 우선 사용자가 입력한 모든 내용을 DOM Parser로 분석 후 분류한다. 이 과정을 거친 후 저작 도구 화면상에는 CSS Converter를 이용하여 실제 HTML 요소로 렌더링된 객체가 표시된다. 편집 과정에서 발생하는 모든 사용자가 입력한 콘텐츠 요소의 삽입이거나 요소에 대한 제어 행위이다. 각 객체에 대한 접근은 이벤트 감시자(Event Listener)가 제어한다. 그리고 이 모듈은 하나의 저작 도구 내에서 다수의 위치나 플러그인에 동시에 존재할 수 있다. 감시 영역의 범위에 따라서 모듈의 구성이 다르고, 감시 요소에 대한 실제 교정의 수행 여부에도 차이가 있다.

그리고 오픈 소스 기반의 HTML 에디터에 플러그인 조정(Plugin Handler) 모듈과 사전 방지(Beforehand Protection) 모듈이 추가 된다. 플러그인 조정 모듈은 저작 도구의 기본 모듈 이외에 추가로 삽입되는 플러그인들을 제어한다. 오픈 소스 기반의 HTML 에디터는 어플리케이션

의 확장성을 보장하기 위해서 플러그인 조정 모듈을 공통적으로 지원한다. 사전 방지 모듈은 에디터 내부에 자바스크립트 모듈로써 직접 참조된다. 템플릿의 구조를 변형할 수 있는 사용자의 행위를 직접적으로 제한하기 위해서는 에디터의 이벤트 모듈과 연계되어야 하기 때문이다. 따라서 사전 방지 모듈은 플러그인 모듈보다는 우선적인 이벤트 감지 권한 및 수정 권한을 갖게 된다.

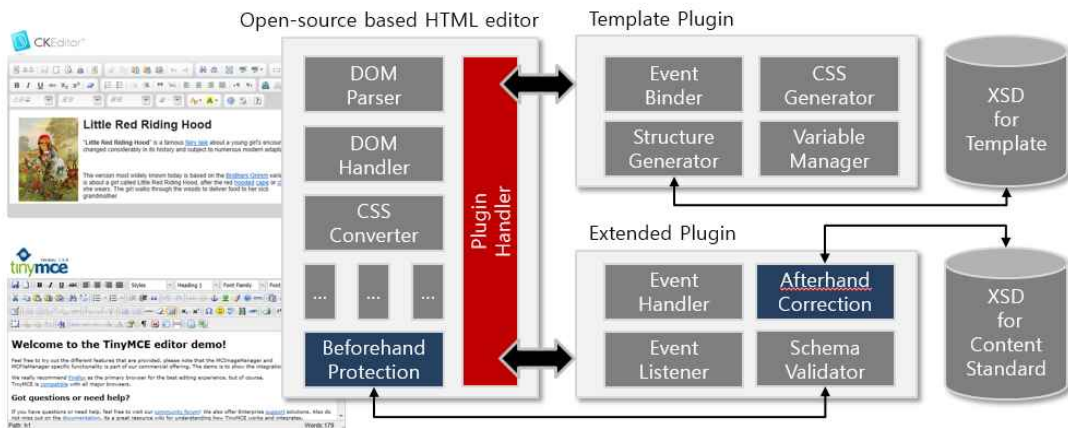
3.2.2 템플릿 플러그인 (Template Plugin)

템플릿 플러그인은 콘텐츠의 일부 혹은 전체 영역에 사용될 수 있는 한정된 자원과 그 구성 형태를 사전에 정의하고, 이를 저작 도구 상에서 사용자가 임의로 호출할 수 있는 외부 플러그인이다. 이 플러그인은 이벤트 할당자(Event Binder)와 디자인 요소 생성자(CSS Generator), 구조 생성자(Structure Generator) 그리고 변수 관리자(Variable Manager)로 구성된다.

구조 생성자는 템플릿을 구성하는 HTML 요소를 구조적으로 배치하고, 사용자의 요청이 있을 때 이에 대한 요소 정보를 별도의 저장소에서 가져온다. 저작 도구 상에서는 HTML 기반의 객체 집합이 동적으로 삽입되어 사용자에게 제공된다.

디자인 요소 생성자는 구조 생성자를 통해 삽입된 객체가 화면에 출력될 때 시각적 인식이 가능한 형태로 표현 요소를 변경하는 역할을 수

(그림 5) 오픈 소스 에디터 기반의 어플리케이션 구성도



(Figure 5) Proposed application architecture with open-source based HTML editor

행 한다. 저작도구는 각 HTML요소에 대한 CSS를 사전에 지정하고 있으나, 템플릿 플러그인은 저작도구의 확장 모듈이기 때문에 별도의 CSS 적용이 가능하다.

변수 관리자는 사용자가 이용하는 각 템플릿의 식별자와 그 내부 변수들을 조정한다. 단일 콘텐츠는 여러 개의 템플릿이 동시에 존재할 수 있다. 따라서 사용자와 콘텐츠간의 인터랙션을 관리하기 위해서는 각 템플릿의 식별자 구분이 필수적이다. 변수 관리자는 식별자를 이용하여 각 구성 요소와 연관되는 모든 콘텐츠의 속성 변수를 조정하는 역할을 수행한다.

이벤트 할당자는 저작 도구 상에 삽입된 템플릿에서 사용자의 입력 혹은 수정이 가능한 영역을 구분하여 독립적인 이벤트를 부여하는 역할을 수행한다. 사용자는 하나의 콘텐츠를 저작하는 과정에서 다수의 템플릿을 함께 사용하는 경우가 존재한다. 이때 하나의 템플릿에 대한 사용자 입력 행위가 다른 템플릿 이벤트에 영향을 주지 않도록 조정하는 기능이 필수적이다. 따라서 이벤트 할당자는 각 템플릿 고유의 인터랙션 기능이 유지되도록 조정한다.

3.2.3 확장 플러그인 (Extended Plugin)

확장 플러그인은 본 논문에서 제안하는 사후 조정 모듈이 포함된 주요 컴포넌트이다. 주요 구성 요소는 이벤트 조정자(Event Handler)와 이벤트 감시자(Event Listener) 그리고 스키마 검사기(Schema Validator)이다.

이벤트 감시자(Event Listener)는 사용자의 키보드나 마우스 입력 행위를 감지한다. 저작 도구를 통해 일어나는 모든 템플릿 변형은 사용자의 입력행위에 의하여 발생한다. 따라서 사용자의 입력 이벤트가 발생과 동시에 유효성 검사기를 통해 템플릿 구조의 변형 여부를 판단하게 된다.

이벤트 조정자(Event Handler)는 사용자가 실행한 다른 외부 이벤트나 키 입력에 대하여 실행 취소(Undo / Backward) 작업을 수행한다. 템플릿 변형이 발생할 경우 가장 효과적인 대응 방법은 실행 취소이다. 변형에 대한 조정을 HTML의 객체 단위로 수행하게 되면 시스템의 성능이 저하 될 우려가 있다. 따라서 우리는 코드 계층의 직접적인 조정이 아닌, 사용자의 입력과 그와 관계된 이벤트 자체를 되돌려 취소하는 메커니즘을 도입하였다. 따라서 콘텐츠 저작 과

정에서 템플릿의 변형이 일어날 경우에 언제든지 이전 작업으로 되돌리는 것이 가능하다.

스키마 검사기는 콘텐츠 메타데이터의 표준 유효성 검사를 수행한다. 참조되는 스키마는 XHTML을 기반으로 하며 저작되는 콘텐츠의 유형에 따라 다양한 참조 모델이 존재 가능하다

4. 어플리케이션의 구현 및 적용

본 연구에서 제안한 기법을 실제 저작 도구에 적용하기 위하여, 오픈 소스 기반 HTML 편집기인 tinyMCE(ver 3.5.8)을 수정 및 확장하였다 [21]. 본 연구에서는 콘텐츠의 저작만을 다루기 때문에 저장과 배포 등의 기능은 포함하지 않고, 템플릿 삽입과 템플릿의 구조 유지를 위한 플러그인의 개발과 동작 확인을 목표로 한다.

tinyMCE의 편집 영역에 삽입되는 템플릿은 평가 문항의 답안 작성용 응답 유형이다. 콘텐츠의 상호운용성을 보장하면서 메타데이터와 스키마가 정형화되어 있는 IMS-QTI의 일부 모델을 참조하였다. 템플릿에 사용된 응답 유형은 주관식인 'Text Entry'와 객관식인 'Choice'이다[22].

우리가 제안하는 저작 도구의 사용자 인터페이스는 (그림 6)과 같다. 사용자는 콘텐츠를 저작하는 과정 중에 플러그인을 실행한다. 플러그인은 사용자의 요청에 따라서 독립된 다이얼로그 창을 표시하여 최종 삽입 여부를 확인한다. 그리고 템플릿의 삽입과 동시에 Interaction Properties 레이어에 응답 유형의 속성을 지정하는 블록이 동적으로 삽입된다. 각 필드의 값은 콘텐츠의 소스 코드에 삽입되는 객체와 속성 값이며 메타데이터로 저장된다.

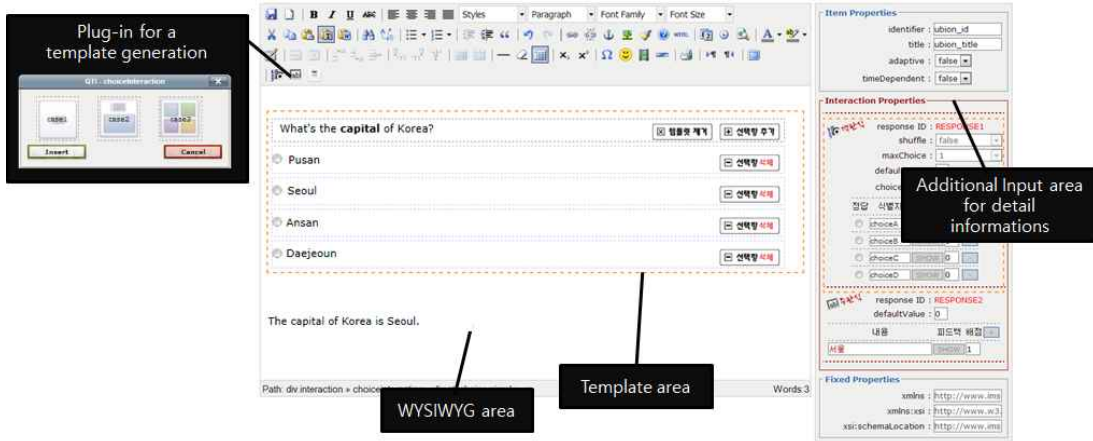
'Choice' 응답 유형에서 템플릿의 세부 기제 속성은 {identifier, shuffle, maxChoice, defaultValue}이다. identifier는 템플릿으로 삽입되는 인터랙션 고유의 식별자이다. 복합 유형의 평가 문항은 여러 개의 응답 유형을 포함할 수 있기 때문에 identifier를 지정한다. shuffle은 사용자가 실제 평가 문항을 이용할 때 선택항들을 순서와 관계없이 뒤섞을지의 여부를 결정하는 속성이다. maxChoice는 사용자가 답안으로 제출할 수 있는 최대 선택항목의 개수를 한정하며, defaultValue는 해당 응답 유형의 기본 채점 값

이다. 그리고 템플릿의 편집영역에서 삽입한 각 선택항들의 식별자와 배점을 부여함으로써 실제 평가 응시자가 제출한 선택항 단위로 피드백을 부여할 수 있는 기능을 함께 제공한다. 본 연구에서 구현한 시스템의 주관식 템플릿인 'Text Entry'는 세부 선택항이 존재하지 않기 때문에 identifier와 defaultValue 그리고 feedback의 입력을 허용한다.

(그림 7)은 본 연구에서 제안하는 템플릿 유지 기법의 적용 여부에 따라서, 사용자의 입력이

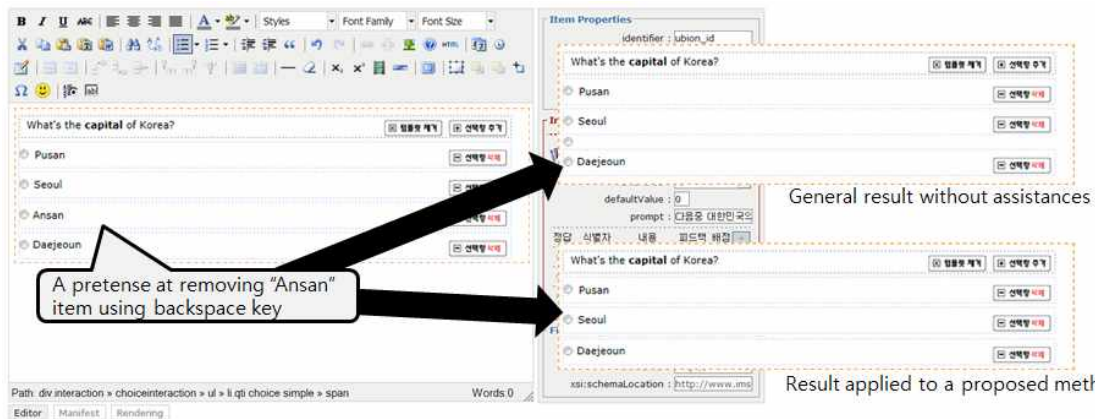
템플릿에 미치는 영향을 보여준다. 예시에서는 사용자가 'Ansan'이라는 선택항에 대하여 임의로 백스페이스키를 이용해 삭제를 시도하였다. 템플릿 구조 유지 기법이 적용되지 않은 경우에는 'Ansan'이라는 텍스트의 삭제를 시도하더라도, 해당 텍스트를 감싸는 태그(Tag)가 삭제되지 않고 잔여 코드로 남게 되는 사례가 발생한다. 이 경우에는 예시 화면처럼 값이 비어있는 (Empty) 선택항이 부분적으로 화면에 표시된다. 객관식의 선택항에 입력되는 데이터는 널(Null)

(그림 6) QTI 기반의 문항 템플릿을 제공하는 평가 문항 저작 도구



(Figure 6) Proposed authoring tool for QTI-compliant item templates

(그림 7) 템플릿 구조 유지 기법의 적용 유무에 대한 차이



(Figure 7) The result comparison of a user behavior according to existence of the template maintenance method

값이 허용되지 않기 때문에, 예시 중 전자의 경우처럼 잔여 태그가 남아서는 안 된다. 반면에 템플릿 구조 유지 기법이 적용된 경우에는 텍스트가 삭제될 때 동시에 이를 감싸는 태그가 완전히 제거됨으로써 템플릿의 구조가 완전하게 유지됨을 확인할 수 있다.

본 논문에서 구현한 모듈은 사용자의 입력 행위를 지속적으로 관찰하고, 콘텐츠의 업데이트가 발생할 경우 각 자원과 템플릿의 유효성 검사를 진행한다. 만약 유효성 검사에 오류가 있는 요소가 발견되면, 요소의 삭제나 교정, 혹은 작업 취소(Undo)를 수행함으로써 템플릿의 원래 구조를 유지한다.

5. 실험

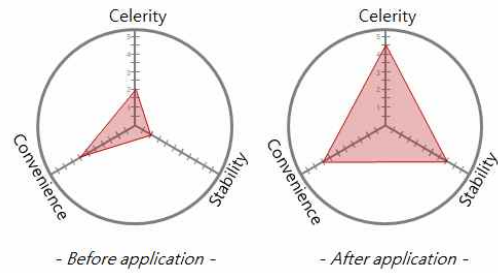
현재까지 QTI를 지원하는 WYSIWYG 기반의 완전한 평가 문항 저작 도구는 발표되지 않았다. 따라서 우리는 자체 개발 저작 도구만을 이용하여 사용자 실험을 진행하였다. 실험 대상자는 평가 문항 저작 경험과 QTI 문항 구조에 대한 지식을 갖고 있는 학부생 및 대학원생 16명이 대상이다. 학생 중에 문항 저작과 콘텐츠 표준에 대한 지식을 갖고 있는 학생이 많지 않기 때문에 소수의 실험 대상자만이 모집 가능하였다. 우리는 실험 대상자들에게 저작 도구를 제공하고, 주관식과 객관식 각각 20문항씩을 만들게 하였다.

이 과정 중에서 실험 대상자들은 우리가 제안한 콘텐츠 유지 기술이 적용되지 않은 저작 도구를 먼저 사용하여 문항을 만든 뒤, 제안하는 기법이 적용된 저작도구를 이용하여 앞 과정과 동일한 문항의 저작을 시도하였다. 그리고 사용자들에게 문항 저작의 신속성(celerity)과 편의성(convenience) 그리고 안정성(stability)에 대하여 각각 리커트 척도(Likert scale) 방식으로 설문조사를 진행하였다[23]. (그림 8)은 제안하는 기법이 적용되지 않은 저작도구와 적용이 된 저작도구를 사용했을 때의 사용자 비교 평가 결과를 나타내고 있으며 전체 실험자의 평균 점수이다.

대부분의 실험 대상자들은 템플릿 유지 기법이 적용되지 않은 저작 도구를 사용할 경우 신속성과 안정성(stability)에 높은 점수를 주지 않

았다. 이것은 사용자들이 템플릿 구조 변형에 효과적으로 대응하지 못하였기 때문에 문항을 완성하는데 많은 시간이 소모되었고, 또한 문항이 완성되었다 하더라도 실험 대상자가 시각적으로 확인하기 어려운 오류와 변형이 템플릿에 발생하였기 때문이다.

(그림 8) 제안하는 기법이 적용되기 전과 적용된 후의 평가 문항 저작 도구에 대한 사용자 평가의 변화



(Figure 8) The outcome variance of questionnaires according to application of the template maintenance method

그러나 본 논문에서 제안하는 기법을 적용한 경우 문항 저작 과정에서 발생하는 템플릿 변형 문제가 실시간으로 해결되었기 때문에 사용자들이 더욱 빠르고 안정적으로 문항 저작 과정을 수행할 수 있었다.

그리고 편의성 항목은 사용자로부터 두 실험 모두 비슷한 점수를 받았다. 이것은 대부분의 실험 대상자들이 오류에 대응하는 템플릿 유지 보다는 템플릿 사용 자체에 대한 편의성을 위주로 평가하였기 때문이라고 판단된다.

6. 결론 및 향후 연구

본 논문은 오픈 소스 기반의 HTML 편집기에서 템플릿이 변형되는 것을 방지하는 기법을 제안하였다. 제안하는 기법이 효과적으로 동작하는지 확인하기 위해서 우리는 직접 오픈 소스 에디터인 tinyMCE에 저작자가 평가 문항의 국제 표준인 IMS-QTI에 대응되는 템플릿 자원을 사용할 수 있도록 플러그인을 추가하였다. 또한 저작도구에서 동적으로 삽입된 템플릿 자원이 구

조나 구성의 변형 일어날 경우, 이에 대하여 효과적으로 대응할 수 있도록 별도의 확장 모듈을 추가하였다. 그리고 우리는 사용자 대상 실험을 통하여, 제안하는 기법이 적용된 저작 도구의 성능 향상 여부를 직접 확인하였다.

현재 무들(Moodle), 사카이(Sakai) 그리고 블랙보드(Blackboard)와 같은 다양한 LMS (Learning Management System)에서 오픈 소스 기반의 HTML 편집기가 사용되고 있다. 제안하는 방법은 자바스크립트로 구현되기 때문에 어떠한 오픈 소스 기반 HTML 에디터에서도 제한 없이 적용 될 수 있다는 장점을 갖는다. 또한 참조 데이터베이스의 변경을 통해 콘텐츠 템플릿의 확장이 용이하다.

본 연구에서는 사용자를 대상으로 저작 행위를 관찰하고, 변형 사례 목록을 수집하였다. 하지만 유지 기법의 효과를 정확하게 측정하기 위해서는 보다 많은 사용자를 대상으로 지속적인 관찰 연구가 필요하다. 그리고 향후에는 템플릿 구조가 변형될 수 있는 다양한 테스트 케이스를 설계하여 시스템의 안정성을 높일 필요가 있다. 또한 tinyMCE 이외의 다른 오픈소스 에디터에서도 실제 적용 가능한지의 여부를 직접 구현을 통해서 확인 할 예정이다.

References

- [1] P. Szekely, "Template-based mapping of application data interactive displays," UIST'90 Proceedings of the 3rd annual ACM SIGGRAPH symposium on User interface software and technology, pp. 1-9, 1990.
- [2] S. Hui, J. Liu, "An Online Template-Based Authoring System for E-Learning," ICWL 2005, LNCS 3583, pp. 38-48, 2005.
- [3] Y. S. Choi, "Developing SCORM Sequencing Model and Sample Contents," Journal of Digital Contents Society, Vol. 10 No. 2, pp. 259-268, 2009.
- [4] M. Dontcheva, S. M. Drucker, D. Salesin, M. F. Cohen, "Relations, Cards, and Search Templates : User-Guided Web Data Integration and Layout," UIST'07 Proceedings of the 20th annual ACM symposium on User interface software and technology, pp. 61 - 70, 2007.
- [5] Apple, iBook Author, <http://www.apple.com/kr/ibooks-author/>
- [6] H. R. Hong, S. J. Han, B. U. Choi, J. H. Cha, "An approach to maintain the template structure for WY SIWYG-based HTML editor," Proc. of the 36th KII SE Fall Conference, pp. 58-60, 2012.11.
- [7] D. D. Chamberlin, "An Adaptation of Dataflow Methods for WISYWYG Document Processing", DOCPR OCS '88 Proceedings of the ACM conference on Document processing systems, pp. 101-109, 1988.12.
- [8] M. Johnson, J. Karat, C. M. Karat, "Usable Policy Template Authoring for Iterative policy Refinement," POLICY'10 Proceedings of the 2010 IEEE International Symposium on Policies for Distributed Systems and Networks, pp. 18-21, 2010.
- [9] P. Zhang, G. Wills, L. Gilbert, "IMS QTIEngine on Android to Support Mobile Learning and Assessment," 2010 International Computer Assisted Assessment (CAA) Conference, Southampton, pp. 20 - 21, 2010.
- [10] S. J. Han, J. J. Kim, Y. S. Lee, J. H. Cha, B. U. Choi, "dinsEditor: A Browser Extension for QTI-Compliant Assessment Item," Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference, pp. 231 - 232, 2010.
- [11] G. Wills, H. Davis, L. Gilbert, J. Hare, Y. Howard, S. Jeyes, D. Millard, R. Sherratt, "Delivery of QTIv2 Question Types," Assessment & Evaluation in Higher Education, pp. 353-366. 2009.
- [12] G. Wills, J. Hare, J. Kajaba, D. Argles, L. Gilbert, D. Millard, "Assessment Delivery Engine for QTIv2 Tests," International CAA Conference, 2008.
- [13] G. Wills, H. Davis, C. Swapna, G. Lester, Howard, Y. Jam, E. R. Jeyes, J. Steve, D. Millard, S. Robert, W. Gavin, "R2Q2: Rendering and Responses Processing for QTIv2 Question Types,"

10th International CAA Conference, Loughborough University, UK, 2006.

[14] M. Johnson, J. Karat, C. M. Karat, K. Grueneberg, "Usable Policy Template Authoring for Iterative Policy Refinement," 2010 IEEE International Symposium on Policies for Distributed Systems and Networks, pp. 18-21, 2010.

[15] Learning Technology Research Group in Kingston University, Aqurate, <http://aqurate.kingston.ac.uk/>

[16] IMS QTI - 'choiceInteraction' Example <http://www.imsglobal.org/question/quiv2p1pd2/examples/items/choice.xml/>

[17] Chrome extension, Google, <http://developer.chrome.com/extensions/>

[18] Microsoft, Internet Explorer, <http://www.ie9.com/>

[19] Mozilla, Firefox, <http://www.mozilla.or.kr/ko/>

[20] Google, Google Chrome, <https://www.google.co.kr/>

[21] Moxiecode Systems AB, TinyMCE, <http://www.tinymce.com/>

[22] IMS Question and Test Interoperability, <http://www.imsglobal.org/question/>

[23] R. Likert, "A technique for the measurement of attitudes," Archives of Psychology, 140, pp. 1-55. 1932.

한 성 재



2006년 : 한국산업기술대학교
(공학사 - 컴퓨터공학)
2008년 : 한양대학교 대학원
(공학석사 -
전자컴퓨터통신공학)

2008년~ 현재 : 한양대학교 대학원
(박사과정- 전자컴퓨터통신공학)

관심분야 : 웹, 이러닝, 디지털 콘텐츠, HCI

최 병 옥



1973년 : 한양대학교 (공학사
전기공학)
1978년 : 게이오대학교 대학원
(공학석사-전기공학)

1981년 : 게이오대학교 대학원 (공학박사-전기공학)

1981년~2002년: 한양대학교 전자전기컴퓨터공학부
교수

2002년~2008년: 한양대학교 정보통신대학 교수

2009년~ 현재 : 한양대학교 컴퓨터공학부 교수

관심분야 : 인공지능, 멀티미디어, 디지털 콘텐츠

차 재 혁



1987년 : 서울대학교
(계산통계학 - 이학사)
1991년 : 서울대학교 대학원
(공학석사-컴퓨터공학)
1997년 : 서울대학교 대학원
(공학박사-컴퓨터공학)

1998년~2001년: 한양대학교 컴퓨터교육과 조교수

2002년~ 현재 : 한양대학교 컴퓨터공학부 교수

관심분야 : 데이터베이스, 파일시스템, 이러닝,
플래시 메모리, 콘텐츠 분석