동적 전자책을 위한 블록 조립식 사용자 피드백 에디터 설계 및 구현

최자령 · 윤지현 · 장미연 · 장수지 · 임순범

숙명여자대학교 IT공학과

Design and Implementation of User Feedback Block Editor for Dynamic E-Book

Ja-Ryoung Choi · Jihyun Yun · Miyeon Jang · Suji Jang · Soon-Bim Lim

Dept. of Information Technology Engineering, Sookmyung Women's University, Seoul, Korea

[요약]

최근 소셜 리딩과 같이 사용자 피드백이 활발해지면서 사용자 피드백 기반 전자책 콘텐츠를 제작에 대한 요구가 증가하였다. 그러나 사용자 피드백을 전자책에 반영하기 위해서는 직접코딩을 해야 하므로 코딩을 못하는 저자에게는 어려움이 있었다. 본 연구는 이러한 문제점을 해결하기 위한 블록 조립식 피드백 에디터 시스템을 개발하였다. 에디터는 입력·처리·출력모듈을 직접코딩으로 구현하였던 사용자 피드백을 영역지정, 컴포넌트 배치, 블록 편집 및 코드 생성·삽입 기능으로 반영할 수 있다. 본 시스템은 HTML, 자바스크립트, PHP, Codeigniter를 이용하여 구현하였다. 블록 편집 기능은 오픈 소스인 Blockly를 이용하여 블록을 제공, 조립할 수 있도록 하였다. 코드 생성 및 삽입 기능은 라이브러리 함수 코드를 삽입한다. 본 시스템을 통해 코딩을 못하는 일반 저자도 코딩 없이 쉽게 사용자 피드백 반영이 가능하도록 한다.

[Abstract]

Recently, as user feedback such as social reading become active, demand has been increased on e-book contents making which is based on user feedback. However, to reflect the user feedback onto the e-book, direct coding is required, which was difficult to the author who was not good at programming. To resolve this problem, Block assembly style feedback editor system, using Blockly was developed. This editor enables to reflect the user feedback by area allocation, component allocation, block editing, and code generating insertion, contrary to the existing way of programming realization in which direct coding was required for input, processing and output separately. This system was developed by using HTML, Javascript, PHP, and Codeigniter. Block editing is enabled to do provision and assembly of blocks by Blockly. The function of code generation & insertion allows to insert the Library function code. Through this system, the general users who are not capable of coding also can reflect feedback without doing actual coding.

Key word : User feedback, Block Coding, E-book, HTML, Editor

색인어 : 사용자 피드백, 블록 코딩, 전자책, HTML, 편집기

<http://dx.doi.org/10.9728/dcs.2017.18.1.63>

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 8 February 2017; Revised 25 February 2017

Accepted 27 February 2017

*Corresponding Author; Soon-bum Lim

Tel: +82-02-710-9424

E-mail: sbkim@sookmyung.ac.kr

I. 서 론

최근 IT 기술의 발전에 따라 전자책에서 1인 출판 활동이 확대되고 있는 추세이다. 1인 창작자인 경우 독자와의 소통이 중요하다. 이에 사용자 피드백을 반영하여 자동 갱신되는 동적 전자책이 요구되고 있다. 그러나 기존의 전자책 에디터인 경우 정적인 전자책 구현[1]에 초점을 맞춰져 있었다.

웹 페이지는 언제든지 갱신이 가능하지만 전자책은 출판 이후 수정을 할 수 없기에 전자책에 피드백을 넣기 위해서는 사용자 피드백 입력력을 미리 코딩을 해주어야 한다. 하지만 코딩을 못하는 일반 저자는 불가능하다. 따라서 본 논문은 코딩을 못하는 일반 저자도 동적으로 갱신되는 전자책 저작을 할 수 있도록 하는데 목적이 있으며 이를 바탕으로 블록 조립식 사용자 피드백 에디터를 개발하였다. 블록 조립식 사용자 피드백 에디터는 피드백 반영 단계인 입력, 처리, 출력 각 단계별 데이터 처리 모듈들을 직접 코딩대신 블록으로 편집으로 처리할 수 있도록 제공한다.

본 논문에서는 위의 문제점을 해결하는 블록 조립식 피드백 에디터 시스템을 Blockly를 이용하여 개발하였다. 에디터는 입력 처리 출력모듈을 직접코딩으로 구현하였던 사용자 피드백을 영역지정, 컴포넌트 배치, 블록 편집 및 코드 생성삽입 기능으로 반영할 수 있다.

본 논문의 제 2장에서는 블록 조립식 사용자 피드백 에디터에 기반이 된 관련 연구와 기존 프로그램을 소개하고, 제 3장에서는 시스템 구성도를 설명한다. 제 4장에서는 블록조립식 사용자 피드백 에디터에 구현환경과 기능구현방법 및 결과 그리고 여러 피드백유형 샘플들을 설명한다. 제 5장에서는 결론을 맺는다.

II. 관련 프로그램 및 선행연구

2-1 Blockly

Blockly[2]는 구글에서 교육목적으로 만든 프로그램들 중 하나로 직접 코딩하지 않고 분류된 블록을 조립하면서 코딩하고 결과를 확인할 수 있다. 그리고 웹 상에서는 시각적 코드 에디터에서 블록으로 코딩한 코드를 5가지 프로그래밍언어(자바스크립트, PHP, Lua, Dart, 파이썬)로 변환된 것을 확인할 수 있다.

그림 1. Blockly 실행화면[2]

Fig. 1. Blockly Running screen[2]

그리고 블록 팩토리[3]에서는 사용자가 원하는 블록을 직접 만들어 색상, 모양 등을 지정할 수 있고 Blockly가 제공하는 5개의 언어 중 원하는 언어 코드로 해당 블록을 편집할 수 있다. 사용자 생성 블록을 만드는 과정도 블록 조립하는 방식으로 블록 코드를 코딩하지 않고 만들 수 있다.

그림 2. 블록 팩토리 실행화면[3]

Fig. 2. Block Factory Running screen[3]

웹 상에서 초보자를 위한 실력향상 과제들이 내장된 프로그램으로 블록 조립을 통해 바로 C 프로그램으로 바꿀 수 있는 시스템[4], 라인바이라인으로 스크래치, Blockly보다 직관적일 수 있도록 텍스트 코드를 블록으로 렌더링하는 프로그램[5], 블록 조립식으로 HTML 태그를 생성하는 프로그램[6]이 개발되었으며 텍스트 기반 코딩에서 벗어나 초보자들도 쉽게 비주얼 기반으로 코딩할 수 있도록 하는 Blockly 기반 학습 시스템을 통해 프로그래밍 초보자 교육을 위한 연구들이 진행되고 있다.

2-2 스크래치

스크래치[7]는 MIT미디어랩(Lifelong Kindergarten 그룹)과 구글이 함께 만든 프로젝트로 Blockly기반으로 만들어졌다. 이는 8세~16세를 대상으로 프로그래밍을 교육하기 위해 만들어졌으며 사용자는 동작, 형태, 소리 등의 다양한 스크립트별 다양한 블록을 조립하여 이야기, 게임, 애니메이션을 직접 만들 수 있다.

그림 3. 스크래치 실행화면[7]

Fig. 3. Scratch Running screen[7]

2-3 선행연구

사용자 피드백이 반영되는 동적인 전자책의 피드백 유형에는 사용자의 의견을 받을 수 있는 설문 유형과, 질문 및 답변 유형 그리고 사용자간의 의견을 나눌 수 있는 통신유형과 협업유형으로 나눌 수 있다.

이 중 사용자의 의견을 받을 수 있는 유형을 다시 4가지로 나누어 볼 수 있는데, 사용자 피드백이 반영된 전자책에 관한 소셜 리딩을 위한 사용자 피드백 기반 전자책 콘텐츠 갱신 기법 선행연구[8]에서 사용자피드백 유형을 자료보강형, 설문조사형, 의견취합형, 문제풀이형 4가지로 정의하였다.

사용자 피드백을 전자책에 활용하기 위해서는 저자가 원하는 특정 위치에서 사용자로부터 피드백을 입력 받고 가공하여 출력하기 위해 미리 정의한 피드백 클루 모델이 필요하다. 클루 모델은 입력을 정의하기 위한 입력 클루와 출력을 정의하기 위한 출력 클루로 나뉜다. 피드백 처리 엔진에는 독자의 피드백을 서버로 전송하는 전송 함수와 서버에서 피드백을 추출하는 출력 함수로 이루어져 있다. 전송 함수인 **Send** 함수가 호출되면 입력한 피드백 정보들이 저장되고 출력 함수인 **View** 함수는 서버에 피드백 내용을 요청하면 저자가 정의한 콘텐츠 영역에 피드백이 출력된다.

독자의 피드백을 반영하여 책의 구조를 깨뜨리지 않고 사용자 피드백을 활용할 수 있지만 피드백 입력, 출력을 위한 영역 지정 방법과 필요로 하는 사용자 피드백을 **Send** 함수와 **View** 함수인 라이브러리로 직접 코딩을 하여 요청, 활용할 수 있어야 하므로 코딩을 못하는 일반 저자가 활용하기에는 어려움이 있다.

III. 시스템 설계

3-1 시스템 개요

본 논문에서 제안하는 블록조립식 사용자 피드백 전자책 에디터는 코딩을 못하는 일반 저자도 동적으로 갱신되는 전자책 저작을 할 수 있도록 하는 목적으로, 그림 4와 같은 전체시스템을 구성하고 있다. 그리고 다음과 같은 흐름에 따라 시스템이

진행된다.

저자는 피드백 편집기로 피드백 반영을 원하는 전자책(HTML)파일을 불러온다. 그리고 저자는 자신이 원하는 피드백의 모듈을 편집과정을 거쳐 만들게 된다. 이때, 편집과정을 거쳐 생성된 피드백 처리 코드는 전자책 내에 삽입되는데, 피드백 처리 부분이 템플릿 라이브러리형태로 삽입된다.

전자책에 피드백편집이 완료되고 저장되어 피드백 처리가 가능한 전자책을 독자가 보고 피드백 내용을 입력한 후 입력버튼을 누르면 이에 해당하는 템플릿라이브러리 함수가 입력된 피드백을 피드백 서버로 전송하고, 전송된 피드백 값을 저장한다. 그리고 출력에 해당하는 템플릿 라이브러리가 서버로 피드백 처리에 따른 값을 출력한다. 독자는 다음 출력물에 따라 피드백을 받아볼 수 있다.

3-2 피드백 처리 코드

그림 4. 전체 시스템 개요

Fig. 4. Whole System Overview

전자책에 피드백 처리 코드를 직접 코딩 할 경우 다음과 같은 과정을 거쳐야 한다. 전자책 내에 피드백을 넣고자 하는 부분에 원하는 컴포넌트에 해당하는 HTML코드를 코딩해야한다. 좋아요 유형 피드백을 넣고자 한다면 좋아요 이미지버튼에 해당하는 코드 그림 5와 출력 텍스트필드에 해당하는 코드 그림 6을 다음과 같이 작성해야 한다. 그리고 클릭 시 작동할 **Send**와 **View** 함수를 **onclick** 속성에 넣어주어야 한다.

```

<img src = "/img/input1.jpg" width =
"30" height = "30" data-clueid = "clue1"
onclick = "Send('img1');View('likebutton',
'', 'clue1');"/>
    
```

그림 5. 좋아요 이미지 버튼 코드

Fig. 5. Like image button Code

```
<input type = "text" id = "likebutton" size = "5"/> 명이 좋아합니다.
```

그림 6. 텍스트필드 코드
Fig. 6. Textfield Code

3-3 피드백 코드 에디터의 설계

피드백 에디터 구성도는 그림 7과 같으며 피드백 처리 기능이 없는 전자책에 피드백 편집 작업을 수행하면 에디터에서 영역 지정, 컴포넌트 배치, 블록 편집을 통해 코드 생성 및 삽입이 되고 저장하면 피드백 처리기능이 있는 전자책이 결과물로 나오는 흐름을 가지고 있다. 에디터의 피드백 반영 과정을 통해 각 과정별로 전자책에 코드가 자동 생성 및 삽입이 된다.

컴포넌트는 선행연구의 사용자 피드백 4가지유형 기반으로 모든 컴포넌트를 포함하며 입력컴포넌트와 출력컴포넌트로 분류하였다. 입력컴포넌트에는 좋아요 이미지버튼, 좋아요 텍스트버튼, 별점, 팝업형 텍스트 Comment버튼이 있고 출력 컴포넌트에는 텍스트필드, 팝업형 텍스트 입력 Answer, 텍스트 입력이 있다. 블록 편집 과정을 위한 블록 모양의 아이콘을 선택하면 블록편집이 팝업창으로 제공된다. 블록메뉴는 입력처리와 출력처리로 메뉴를 구성하였으며 처리블록들은 각 함수의 파라미터에 해당한다.

블록 코딩을 완료하면 배치한 입력처리, 출력처리에 해당하는 블록편집으로 생성된 View, Send함수가 생성 및 삽입된다.

그림 7. 피드백 에디터 구성도
Fig. 7. The structure of our feedback editor

IV. 블록 조립식 피드백 에디터 구현

4-1 시스템 구성 및 구현환경

본 시스템의 구성 및 구현환경은 그림 8과 같으며, HTML5와 블록편집에 사용된 오픈소스 Blockly, 소스 확인 및 코드삽입

에 사용된 오픈소스 Ckeditor, Send, View 라이브러리 함수와 관련해서 자바스크립트를 주로 이용하였다. 서버언어로는 PHP를 사용하였으며 웹 프레임워크인 Codeigniter를 이용하였다.

그림 8. 시스템 구성도
Fig. 8. System Configuration

4-2 영역지정 및 컴포넌트 배치 기능 구현

저자는 파일을 선택하여 불러온 다음 자신이 원하는 위치에 영역을 지정할 수 있다. 영역지정 및 컴포넌트에서 에디터부분은 Ckeditor를 이용하였다. 드롭다운 메뉴로 영역지정, 입력컴포넌트만 활성화되어 있으며 출력컴포넌트는 비활성화되어 숨김 처리되어 있다. 그림 9를 보면 좋아요 유형으로 선택한 영역에 영역지정을 하면 <div></div>태그가 생성되고 점선으로 영역이 표시된다. 영역 내에 하나의 입력컴포넌트와 출력컴포넌트를 지정할 수 있다. 입력컴포넌트 선택 시 나머지 입력컴포넌트들은 비활성화 된다. 그리고 해당하는 출력컴포넌트를 선택하여 배치한다. 배치된 컴포넌트를 삭제하려면 컴포넌트 우측에 위치한 '-'모양의 버튼을 클릭한다.

그림 9. 피드백 영역지정 및 컴포넌트 배치
Fig. 9. Feedback Area and Component allocation

4-3 블록 편집 기능 구현

블록편집 화면은 컴포넌트블록과 처리블록메뉴, 자바스크립트코드 변환되는 영역, 자바스크립트코드변환 버튼, 블록코딩완료버튼을 포함한다. 입력컴포넌트에 따라 그에 해당하는 블록편집화면이 등장하도록 구현하였다. 오픈 소스 Blockly를 이용하였으며 블록팩토리로 블록의 모양과 색상을 정의하였다.

사용자는 처리블록메뉴에서 처리블록을 확인하고 마우스로 드래그하여 해당하는 컴포넌트블록에 가져다 놓으면 레고블록을 맞추듯 블록조립이 된다. 처리블록은 사용자가 잘못된 블록조립을 하는 위험을 방지하기 위해 올바른 조립이 아닐 경우 블록킹을 시켜 블록이 뿔겨지면서 조립이 불가능하도록 구현하였다.

4-4 피드백 코드 생성 및 삽입 기능 구현

그림 10은 텍스트유형에 해당하는 블록조립과정으로 블록조립을 하고 조립된 출력 컴포넌트 블록의 파라미터 값에 알맞은 값을 설정해준다. 주황색 박스에는 출력되는 피드백이 정렬되는 방법을 최신순, 과거순, 랜덤순 중에 선택할 수 있고, 분홍색 박스부분에는 출력되는 피드백의 개수를 직접 입력하여 설정할 수 있다. 블록조립을 완료 후 위쪽 하늘색 박스부분으로 표시된 자바스크립트코드 변환 버튼을 클릭하면 코드변환 캔버스에 블록조립으로 생성된 Send와 View라이브러리 함수가 등장한다.

그림 10. 피드백 코드 생성
Fig. 10. Generation of Feedback Code

블록코딩완료버튼을 클릭하고 소스 보기를 선택하면 그림 11의 표시한 부분에서처럼 Send와 View함수코드가 삽입된 결과를 확인할 수 있다. 모든 피드백 과정을 마친 후 저장하기를 통해 피드백처리 과정이 완료된다.

그림 11. 피드백 코드 삽입
Fig. 11. Insertion of Feedback Code

V. 피드백 유형별 구현결과 및 사용성 평가

5-1 피드백 유형별 구현결과

그림 12 좋아요 유형은 독자로부터 좋아요 피드백을 받아 좋아요 합계를 출력한다. 입력 컴포넌트로 좋아요 이미지 버튼이 사용 되었으며 출력 컴포넌트는 텍스트 필드가 사용되었다. 저자가 블록 편집 후 전자책이 저장되면 독자가 좋아요 이미지 버튼을 누르는 만큼 좋아요 수가 증가한다.

그림 12. 좋아요 유형
Fig. 12. Like Type

그림 13 평점 유형은 독자로부터 별 점을 입력 받아 평점을 출력한다. 입력컴포넌트는 별 점이 사용되며 출력컴포넌트로 텍스트 필드가 사용된다. 1부터 5까지의 별 점이 제공되며 평점이 출력된다.

그림 13. 평점 유형
Fig. 13. Grade Point Type

그림 14 텍스트 유형은 독자로부터 직접 텍스트 피드백을 받는다. 입력컴포넌트로 텍스트가 사용되며 출력 컴포넌트는 텍스트 입력이 사용된다. 이 유형은 독자로부터 텍스트 피드백을 입력 받아 텍스트 입력에 출력된다.

그림 14. 텍스트 유형
Fig. 14. Text Type

그림 15 Comment 팝업형 텍스트 입력 유형은 텍스트 피드백을 팝업으로 받아 최신순으로 하나의 텍스트 피드백이 출력된다. 입력 컴포넌트로 팝업형 텍스트가 사용되며 출력 컴포넌트로 텍스트 입력이 사용된다. 이 유형은 독자로부터 텍스트 피드백을 팝업으로 받아 텍스트 입력에 출력된다.

그림 15. Comment 팝업형 텍스트 입력 유형
Fig. 15. Comment Popup Text Input Type

그림 16 Answer 팝업형 텍스트 출력 유형은 한 개의 최신 텍스트 피드백을 받아 팝업으로 출력된다. 입력 컴포넌트로 텍스트가 사용되며 출력 컴포넌트로 팝업형 텍스트 입력이 사용된다. 이 유형은 독자로부터 받은 텍스트 피드백이 팝업형 텍스트 입력으로 출력된다.

그림 16. Answer 팝업형 텍스트 출력 유형
Fig. 16. Answer Popup Text Input Type

5-2 사용성 평가

본 프로젝트의 목적이 제대로 이루어졌는지 확인하기 위해 코딩을 못하는 사용자가 블록에디터를 이용하여 피드백 편집을 할 수 있는가와 사용자가 블록에디터를 이용함에 있어 어려움이 없는가에 대한 사항을 평가하였다.

평가 대상자는 12명으로 구성하여 코딩을 못하는 사용자 6명, 코딩을 할 수 있는 사용자 6명으로 사용자그룹을 나누었다. 데이터는 성공률, 각 작업의 소요시간, 오류횟수를 수집하였다. 5가지 피드백 유형을 이용하여 각 피드백 유형을 컴포넌트 배치만을 요구한 간단 형 샘플과 사용자의 복잡한 편집과정을 요구한 복잡 형으로 만들어 총 10가지의 샘플을 제작하였다. 10가지 샘플은 주된 기능인 블록편집의 복잡도 기준으로 <표1>과 같이 상, 중, 하로 분류하였다.

표 1. 사용성 평가 샘플

Table. 1. Usability Evaluation Sample

A(하)	B(중)	C(상)
좋아요 유형(간단)	텍스트 유형(간단)	텍스트 유형(복잡)
좋아요 유형(복잡)	Comment 유형(간단)	Comment 유형(복잡)
별점 유형(간단)	Answer 유형(간단)	Answer 유형(복잡)
별점 유형(복잡)		

사용자 한 명 당 난이도 별 샘플을 무작위로 2개씩 총 6개의 샘플을 수행하도록 하였고 6명으로 이루어진 한 그룹 내에서도 샘플 수행 순서를 (A->B->C), (B->C->A), (C->A->B)로 나눠 각 순서 당 2명씩 배정하였다. 코딩을 못하는 사용자와 코딩을 할 수 있는 사용자가 블록에디터를 이용해 작업을 하는데 얼마나 차이가 있는지 알아보고자 각각 샘플 난이도별 작업 소요시간을 비교하였다. A샘플 1.6배, B샘플 1.5배 C샘플 1.8배로 모든 샘플의 소요시간이 2배 이상 차이 나지 않는 것으로 확인하였다. 따라서 그룹 간 소요시간의 차이는 있을 뿐 코딩을 못하는 사용자도 블록에디터를 활용하여 동적으로 갱신되는 전자책 저작을 할 수 있다는 결론을 얻었다.

그림 17. 그룹 간 작업 소요시간 비교

Fig. 17. Compare Working Time Between Groups

블록에디터 작업 횟수 증가에 따라 사용자의 숙련도가 증가하는지 확인하기 위해 사용자가 수행한 6개의 샘플에 대하여 각 그룹의 회차별 평균 소요시간을 측정하였다. 1차에서는 2.4 배로 6차는 1.2배로 점점 차이가 줄어들었다. 회차가 진행될수록 두 그룹의 소요시간이 차이가 크게 줄어들기 때문에 블록에디터를 사용할수록 숙련도가 증가함을 알 수 있다. 따라서 코딩을 못하는 사용자도 코딩을 할 수 있는 사용자만큼 블록에디터를 능숙하게 사용할 수 있다.

그림 18. 작업 횟수에 따른 소요시간

Fig. 18. Number of Hours Required by Work Count

VI. 결론 및 향후 연구

본 시스템은 영역지정 및 컴포넌트 배치, 블록편집, 코드 생성 및 삽입의 과정을 통해 직접코딩하지 않고도 쉽게 전자책에 피드백을 추가하고 이를 처리할 수 있도록 하여 코딩을 못하는 일반 저자도 피드백을 반영할 수 있는 전자책을 만들 수 있도록 하였으며, 기존 관련연구[1]의 4가지 유형의 컴포넌트들이 처리되는 것을 확인하여 앞으로 코딩을 할 수 있는 여부에 관계없이 일반 저자들도 사용자 피드백처리가 가능한 전자책을 저작할 수 있다는 가능성을 확인하였다.

본 시스템은 4가지 유형의 컴포넌트로 이용할 수 있는 초보 자용이다. 앞으로 본 시스템을 더 다양한 유형들의 입출력 컴포넌트들을 추가하여 전문가용으로 발전시킨다면 좀 더 사용자 피드백 전자책 제작에 활용도 높은 에디터가 될 수 있을 것이다.

참고문헌

- [1] Hyun-Suk Moon, "The Implementation of an EPUB3.0-Based Fixed Layout e-Book", *Journal of Digital Contents Society*, Vol. 16, No. 1, pp. 105-112, 2015.
- [2] Google Blockly. Available: <https://developers.google.com/blockly/>.
- [3] Google Blockly, Block Factory. Available: <https://blockly-demo.appspot.com/static/demos/blockfactory/index.html>.
- [4] Sang Cho, "Blockly webc Programming Convergent Learning System", Computer and Information Engineering Department, Cheongju University, 2015.
- [5] Bau, D. A. Droplet, "A Blocks-Based Editor for Text Code", *Journal of Computer Science in Colleges*, 2015
- [6] Aggarwal S, Bau D. A, Bau D. A "Blocks-Based Editor for HTML Code", IEEE Blocks and Beyond Workshop, 2015.
- [7] Scratch. Available: <https://scratch.mit.edu/about>

- [8] Ja-Ryoung Choi, Jungsoo Hwang, Soon-Bum Lim, "Design of User Feedback Interface for Dynamic Updating of E-Book Content.", *International Journal of Multimedia and Ubiquitous Engineering*, Vol. 10, No. 11, pp. 359-368, 2015.

최 자 령 (Ja-Ryoung Choi)

2009년 숙명여자대학교 멀티미디어학과 (학사)
2009년~현재 숙명여자대학교 IT공학과 (석박통합 과정)

※ 관심분야: 전자출판(전자책, 디지털 교과서), UI/UX, 웹/모바일 콘텐츠, 소셜 미디어, 클라우드소싱 등

윤 지 현 (Ji-Hyun Yun)

2012년~현재 숙명여자대학교 IT공학과 (학사)

※ 관심분야: 전자책, HTML, UI, 웹 등

장 미 연 (Mi-Yeon Jang)

2012년~현재 숙명여자대학교 IT공학과 (학사)

※ 관심분야: 웹, 전자책, 미디어 콘텐츠, UI/UX 등

장 수 지 (Su-ji Jang)

2013년~현재 숙명여자대학교 IT공학과 (학사)

※ 관심분야: 전자책, UI/UX, HTML, 디지털콘텐츠 등

임 순 범 (Soon-Bum Lim)

1982년 서울대학교 계산통계학과 (학사)
1983년 한국과학기술원 전산학과 (석사)
1992년 한국과학기술원 전산학과 (박사)

1989~1992년 (주)휴먼컴퓨터 창업 (연구소장)

1992~1997년 (주)삼보컴퓨터 프린터개발부 부장

1997~2001년 건국대학교 컴퓨터학과 교수

2001년~현재 숙명여자대학교 IT공학과 교수

2006년 University of Colorado 방문교수

※ 관심분야: 컴퓨터 그래픽스, 웹/모바일 멀티미디어 응용, 디지털 방송, 전자출판(폰트, 전자책, XML 문서), User Interface 등