Interactive E-Textbook Platform Based on Block Editing Model in Crowdsourcing E-Learning Environments

Ja-Ryoung Choi, Sookmyung Women’s University, South Korea
Soon-Bum Lim, Sookmyung Women’s University, South Korea

ABSTRACT

Instructors can now work with students to create various textbooks based on crowdsourcing. In particular, as feedback provided by students is essential for determining the quality and direction of classes, instructors should interact with students who are currently participating in classes by exchanging feedback. This paper proposes a block editing model that can reflect student feedback. The block editing model is an interactive e-textbook editing model that is dynamically updated based on the feedback provided by students in real time without modifying the structure of digital textbooks. In particular, in order for even non-developers who do not know web programming languages to be able to produce interactive digital textbooks easily, the authors developed an editor that could help implement them based on Blockly, a visual programming language. This paper enables instructors to improve the direction and quality of classes depending on the learning achievement of students and understanding based on feedback information provided by students and feedback analysis.

KEYWORDS

Block Programming Editor, Digital Material, E-Learning, Interaction, Student Feedback

1. INTRODUCTION

As online learning environments have become more popular with the advancement of web technologies, web-based learning materials are being used in many classes. Recently, as the voluntary participation of the general public has increased owing to the characteristics of social media such as participation, sharing, and openness, crowdsourcing, which was first identified by Jeff Howe, is being used in education. Crowdsourcing has an advantage of being able to solve problems in a short period of time through collective intelligence because content can be created by the participation of many non-specific people (Nkoana, 2016; Karataev E, 2016; Weld, 2012). Even for educational content, higher-quality textbooks can be produced through collective intelligence rather than by instructors on their own. This approach to creation of learning content based on crowdsourcing can help improve the quality of education.

The ADDIE model, which is well known in the existing teaching system design process, includes analysis, design, development, implementation, and evaluation (Molenda, 2003). Among them, the
evaluation step is of particular importance. Textbooks lead classes in the right direction, and the quality of classes can be improved only through the suitability and efficiency of textbooks, class evaluation, and feedback. In all schools, evaluations are conducted to evaluate the quality of classes after their completion at the end of the semester. Therefore, the results of the evaluations are used only to improve future classes as they are difficult to apply directly to current classes. The paradigm of education should be shifted to the direction where students actively express their opinions and ask questions in classes, rather than passively attend classes. This student-centered approach can help improve creative thinking, promote active participation in classes, and actively generate student feedback to steer these classes in a positive direction.

The study to collect and analyze feedback was carried out through interfaces regardless of textbooks. The NB system developed a method to manage queries and answers next to textbooks in an online learning environment (Zyto, 2012). This is a learning system in which students will view PDF textbooks and ask questions if they need clarification, and the tutors and students answer the questions. The web-based annotation sharing system enables discussion in class as well as adding notes into class materials. The Mudslide system receives feedback from students by letting them mark what they do not understand in textbooks after class (Glassman, 2015). In other words, after listening to online lectures, students select the parts that they do not understand, write down reasons for not understanding them, and provide them to teachers. The system then collects feedback given by the students and provides it to the teachers by displaying it as histograms. Through the Mudslide system, teachers can confirm which parts are difficult for students to understand and can improve them in the subsequent classes. The Marky system receives feedback by marking the parts regarding which authors want to receive feedback from users directly (Pérez-Pérez, 2015). For the purpose of sharing web-based document annotations, authors can write documents by specifying and referencing the format and time of feedback to be received from readers. As in the above examples, existing studies focused mainly on interfaces for receiving feedback easily, not on directly reflecting feedback. This study focuses more on updating digital textbooks dynamically by reflecting feedback on textbooks in real time through active communication with students.

In addition, basic web programming knowledge is required to create digital textbooks capable of immediate interaction between instructors and students. It is difficult for general instructors to produce interactive digital textbooks. (Papert, 1980) identified three problems for existing programming languages when applying to children: (i) Existing programming languages were too difficult to use, especially on programming syntax (ii) Activities within the programming, such as generating lists of numbers or making line drawings, were not connected to children’s interests or experiences (iii) Without proper guidance, it requires deeper exploration when things went “wrong”. Many attempts were proposed to introduce programming to children and novice programmers (i.e. Flash/ActionScript). To resolve such problem, a visual programming language and editor called Blockly (Fraser, 2015) was developed to make programming easier for novice level programmers. Blockly can create visual block programming languages and editors by providing a client-side JavaScript library. Unlike existing text-based programming languages, it employs a block assembly method so that it is easy for novices to develop simple programs in an interactive environment. In particular, it has an advantage whereby non-developers can use it simply by viewing the shapes and descriptions of blocks without having to learn the syntax of a programming language. Scratch (Resnick, 2009; Maloney, 2010) is another system for block-based programming environment, which is an extension of Blockly project. Scratch targeted people who hadn’t previously identified themselves as programmers, and gained a huge traction. Age group of 8 and 16 has the largest proportion within the Scratch users, and the semi-gamified platform on the web encouraged them to share their creation. Alice (Kelleher, 2006) is a block-based programming tool to build interactive narratives or 3D application, and the survey-based study found that it has a measurable positive effect on performance and retention on the education environment. TinyInventor (Hansen, 2011) applied block programming on the field of networking.
Related Content

Linked Data: The Story so Far
www.irma-international.org/chapter/linked-data-story-far/55046

RIKEN MetaDatabase: A Database Platform for Health Care and Life Sciences as a Microcosm of Linked Open Data Cloud
www.irma-international.org/article/riken-metadatabase/193932

Visual Ontology Modeling and the Semantic Web
www.irma-international.org/chapter/visual-ontology-modeling-semantic-web/44932

A Theory of Social Agentivity and its Integration into the Descriptive Ontology for Linguistic and Cognitive Engineering
www.irma-international.org/chapter/theory-social-agentivity-its-integration/76170

Modeling for Learning
www.irma-international.org/chapter/modeling-learning/44938