CASE STUDY

Developing a math e-learning question specification to facilitate sharing questions between different systems

Mitsuru Kawazoe, Faculty of Liberal Arts and Sciences, Osaka Prefecture University, Sakai, Japan. Email: kawazoe@las.osakafu-u.ac.jp
Kentaro Yoshitomi, Faculty of Liberal Arts and Sciences, Osaka Prefecture University, Sakai, Japan. Email: yositomi@las.osakafu-u.ac.jp
Yasuyuki Nakamura, Graduate School of Informatics, Nagoya University, Nagoya, Japan. Email: nakamura@nagoya-u.jp
Tetsuo Fukui, Mukogawa Women's University, Nishinomiya, Japan. Email: fukui@mukogawa-u.ac.jp
Shizuka Shirai, Osaka University, Toyonaka, Japan. Email: shirai@ime.cmc.osaka-u.ac.jp
Takahiro Nakahara, Sangensha LLC., Chitose, Japan. Email: nakahara@3strings.co.jp
Katsuya Kato, Cybernet Systems Co., Ltd., Tokyo, Japan. Email: katsuyak@cybernet.co.jp
Tetsuya Taniguchi, Nihon University, Tokyo, Japan. Email: taniguchi.tetsuya@nihon-u.ac.jp

Abstract

In recent years, e-assessment has become increasingly popular in mathematics education. However, there are several different systems, and hence, the contents need to be developed independently in each system. Sharing contents between different systems is important for the diffusion of math e-learning/assessment systems. This study focuses on sharing computational questions, which are the main contents in most systems. The structure of such questions seems essentially compatible between many systems. Based on this observation, a specification, namely mathematics e-learning question specification (MeLQS) is proposed and described as a common base for developing contents in computer-algebra-system-based mathematics e-learning/assessment systems. Furthermore, the development of authoring tools for MeLQS is reported.

Keywords: mathematics e-learning system, contents sharing.

1. Introduction

E-assessment has become increasingly popular in mathematics education. Several different and independently developed math e-learning/assessment systems have been reported—STACK (Sangwin, 2013; Stack.ed.ac.uk, 2018), Möbius Assessment (DigitalEd.com, 2018) (formerly Maple T.A.), Numbas (Numbas.org.uk, 2018), MATH ON WEB (Osakafu-u.ac.jp, 2018), WeBWorK (The Mathematical Association of America, 2018), and many others. In these math e-learning/assessment systems, questions need to be implemented independently in each system due to non-compatibility between the systems. Because the development of content is difficult, it is important to reduce the burden on content developers. There have been a few attempts to share e-learning contents. The Abacus project (2018) and ItemBank (Nakamura et al., 2014) aim to share questions used in STACK, and Möbius Cloud (DigitalEd.com, 2018) aims to share questions used in Möbius Assessment. These attempts also contribute to developing communities of content developers at different institutions. However, these are closed to specific systems.

Sharing content between different systems makes it possible to not only reduce the burden on contents developers but also connect the communities of content developers for different systems and enhance the diffusion of math e-learning/assessment systems. Therefore, computational questions in math e-learning/assessment systems developed with computer algebra systems

(CASs), such as solving systems of linear equations and solving differential equations, are focused on, because they are implemented in a similar way in each system.

One approach is to develop a tool for importing questions from one system to another. Kinnear and Sangwin (2016) developed a ‘MapleTA question importer’ which imports Maple T.A. questions to STACK. However, its functionality is restricted to ‘import questions placing corresponding fields in the correct places pending editing by an intelligent human’ (ibid.). Developing an import tool is difficult, even if it is a semiautomatic one, because translating codes between different systems is complex and difficult. In addition, with this approach, several tools for all pairs of systems must be developed.

Another approach is to make a common ‘question specification’ of math e-learning questions and share the data of questions, containing the idea of questions, marking principles, and algorithms needed for implementation, in a common format that does not depend on any specific e-learning/assessment system or any CAS language. With this approach, content developers do not need to learn codes of questions in another system in order to implement them in the target system. They only need to implement a question according to its question specification. Hence, knowledge of multiple CAS languages and multiple systems is not required.

This work shows that an approach that involves making a common question specification for math e-learning is possible and proposes such a question specification. The proposed specification is called the ‘mathematics e-learning question specification’ (MeLQS).

We think that it is useful to design MeLQS along with a process of developing questions in math e-learning/assessment systems. Each question is produced from an idea of a mathematics teacher. First, a mathematics teacher who would like to create a question for an online test acquires an idea based on his/her knowledge of mathematics teaching. The idea includes the following: the types of questions that need to be presented to students; methods to verify the correctness of students’ answers; the types of errors often found in students’ answers; and the types of feedback that need to be provided to students. Then, the teacher’s idea is formulated as a set of algorithms on marking, identifying errors, and so on. Finally, the question is implemented in a target system by coding algorithms. Thus, a process of developing questions can be divided into the following three phases: 1) describing a teacher’s idea, 2) algorithmizing the teacher’s idea, and 3) implementing algorithms. Based on these three phases, we propose MeLQS as a pair of two specifications, concept design and implementation specification. The teacher’s idea in the first phase is described in a concept design, and its algorithmization in the second phase is described in an implementation specification.

The outline of this work is as follows. Section 2 focuses on the three systems used in Japanese universities, and it is observed that these three systems have a common data structure for questions. In Section 3, the MeLQS is proposed based on the observations in Section 2. Finally, conclusions are provided in Section 4.

2. Observation

This section focuses on a data structure of questions in the following three systems used in Japanese universities: STACK, Möbius Assessment, and MATH ON WEB.

2.1. Question data in STACK

STACK is an online assessment system for mathematics and STEM that works as one of the question types for Moodle and ILIAS; it can also be integrated more widely via LTI. Maxima (Maxima.sourceforge.net, 2018) is used for the manipulation of symbolic and numerical expressions in STACK. The question data of STACK consist of question variables (potentially random variables that can be used to generate a question), question text (a statement of the question students actually see), general feedback (a general feedback text, usually a worked solution, that is shown to all students after they attempt the question), inputs (areas where students input answers and some input types—algebraic input, numerical input, matrix, and so on are allowed) to be placed in a question text, and potential response trees that are the algorithms that establish the mathematical properties of the students' answers and generate feedback with scores.

2.2. Question data in Möbius Assessment

Möbius Assessment is an online testing and assessment system designed especially for courses involving mathematics, and it can be incorporated into virtually any learning management system, such as Moodle. It consists of a wide variety of question types to assess the mathematical ability of students. The system makes it possible to generate mathematical questions algorithmically as well as evaluate the students' responses flexibly with a grading code programmable by Maple, a computer algebra system. When a Maple-graded question is used, students' responses are graded with Maple (Maplesoft.com, 2018). The dataset basically consists of five primary items: question text (a mathematical problem statement), response area (a field for students to enter their own answers), grading code (a field to write Maple code to grade the student response), algorithm (a field to write algorithmic code to generate formulae, matrices, and plots), and feedback (a field to write a comment for the question itself). Compared with the potential response tree (PRT) of STACK, Möbius Assessment does not have an identical capability to the PRT but can have multiple parts, which could be subquestions or messages (feedback), in a single question with adaptive sections (separators to divide questions). Each part is selectively displayable only with the students' responses: correct or incorrect answers. When a question is generated online, its algorithm code is executed once. Then, the algorithmic elements, such as its answer and mathematical expressions, are referred from the other items, such as question text and grading code.

2.3. Question data in MATH ON WEB

MATH ON WEB is a web site that offers two mathematics e-learning/assessment systems: WMLS and WASM (Kawazoe and Yoshitomi, 2017). These two systems have been developed with web*Mathematica* (Wolfram Research, Inc., 2018). Although it has different types of contents, questions in the drill section of WMLS and questions in WASM have almost the same structures as the ones in STACK and Möbius Assessment. The question data comprises the following: a problem template (a question text with places where mathematical expressions are inserted); problem parameters (mathematical expressions given as a list or as a *Mathematica* program generating mathematical expressions) to be inserted in a question text; an answer column (HTML form with variables setting); an answer analyser (a marking algorithm given as a *Mathematica* program) that marks students' answers and also identifies the type of error if students fail to solve the question; a list of feedback messages corresponding to the output of the marking algorithm; and a problem example with its model answer and a guide for how to solve (PDF). Marking algorithms are written as the which statement of *Mathematica*, and marking is performed by a first-match method.

2.4. A common data structure of STACK, Möbius Assessment, and MATH ON WEB

From the observation of the above three systems, a common structure of question data can be observed, which consists of the following data.

- Question text with places where mathematical expressions are inserted
- Mathematical expressions inserted into the question text (given as a list or as a CAS program generating them)

- Answer box (given as an HTML form, including variable setting)
- Marking algorithm (given as a CAS program)
- Fixed feedback message shown to all students
- List of feedback messages for adaptive feedback according to the result of marking

Because Möbius Assessment does not have a function to give a different feedback message depending on the result of marking, the last data were regarded as optional. Each of the data is named differently in each system. The results of the observation are summarized in Table 1.

Table 4. Correspondence between data of each system

Item	STACK	Möbius Assessment	MATH ON WEB
Question text	Question text	Question text	Problem template
Mathematical expressions	Question variables	Algorithmic variables	Problem parameters
Answer box	Inputs	Response area	Answer column
Marking algorithm	Potential response trees	Grading code (available only in Maple-graded question)	Answer analyser
Fixed feedback message	General feedback	Feedback	Model answer of a problem example, and a guide for how to solve
Adaptive feedback messages	Feedback defined in potential response trees	Available but limited inside the sequence of question text with adaptive sections by correct or incorrect answers	A list of feedback messages with respect to the result of answer analyser

3. MeLQS

With the observation presented in the previous section, a common question specification (MeLQS) is proposed that can be used at the design stage of questions for different CAS-based math e-learning systems. In previous work (Nakamura et al., 2018), only a brief idea of MeLQS was presented, and a prototype of an authoring tool for concept design that is a part of MeLQS was shown. Here, the details of MeLQS and authoring tools are described. They have been updated from the ones reported previously.

3.1. Overview of MeLQS

MeLQS is designed as a pair of two specification: concept design and implementation specification. Concept design is a specification describing a teacher's idea. Its objective is to share the idea of questions from the viewpoint of mathematics teachers. Hence, it is described with the mathematics teachers' language, not with programming language. Implementation specification is a specification describing the algorithmization of the teacher's idea. The objective of implementation specification is to provide sufficient information needed in implementing questions to programmers or engineers who are engaged in authoring questions for online tests. In the implementation specification, algorithms for implementing questions are provided, but their descriptions do not depend on a specific math e-learning system. These specifications are described in the following sections.

3.2. Concept Design

The concept design aligns with the phase of describing a teacher's idea. The concept design consists of data corresponding to the items in Table 1 together with some supplemental data (subject, course, learning unit, title, the aim of the question, answer example, and remark). The question text, mathematical expressions, and answer box in Table 1 are compiled into 'Question' in the concept design. A 'question' is described as a sample question with examples of mathematical expressions and an answer box. The marking algorithm and adaptive feedback messages are compiled into 'Check List'. A check list is the most important data in the concept design, and it describes how to assess students' answers based on the teachers' knowledge of students' errors. The fixed feedback message in Table 1 is optional data which can be attached to 'Answer Example' in the concept design. Sample data of concept design are shown below.

- *Subject/course/learning unit*: mathematics(college)/linear algebra/Euclidean vector space
- *Title*: Length of vectors
- *The aim of the question*: To assess students' understanding of the definition of length of vectors in the numerical vector space
- *Question*:

Find the length $\|\boldsymbol{a}\|$ of the vector $\boldsymbol{a} = \begin{pmatrix} 1 \\ 2 \\ -1 \\ 3 \end{pmatrix}$ with respect to the dot product. [Answer] $\|\boldsymbol{a}\| =$ _______

- *Answer example*: $\|\boldsymbol{a}\| = \sqrt{15}$
- *Check list*:

No	Condition	Feedback	Score (%)	Remark
1	The input answer is the same as the correct answer.	Correct!	100	
2	The input answer is a negative number.	The length cannot be negative.	0	
3	The input answer is an imaginary number.	Your answer is an imaginary number. The length cannot be an imaginary number.	0	
4	The input answer is a sum of squares of the coordinates. (Taking a square root is forgotten.)	Recall the definition of the length.	0	

- *Remark*: no comment.

An authoring tool has been developed for concept design (Figure 1). With the authoring tool, a teacher can input data step-by-step. Mathematical equations can be described in TeX format. The next version of the authoring tool will include a math input interface MathTOUCH (Shirai and Fukui, 2017), which would help users in inputting mathematical equations.

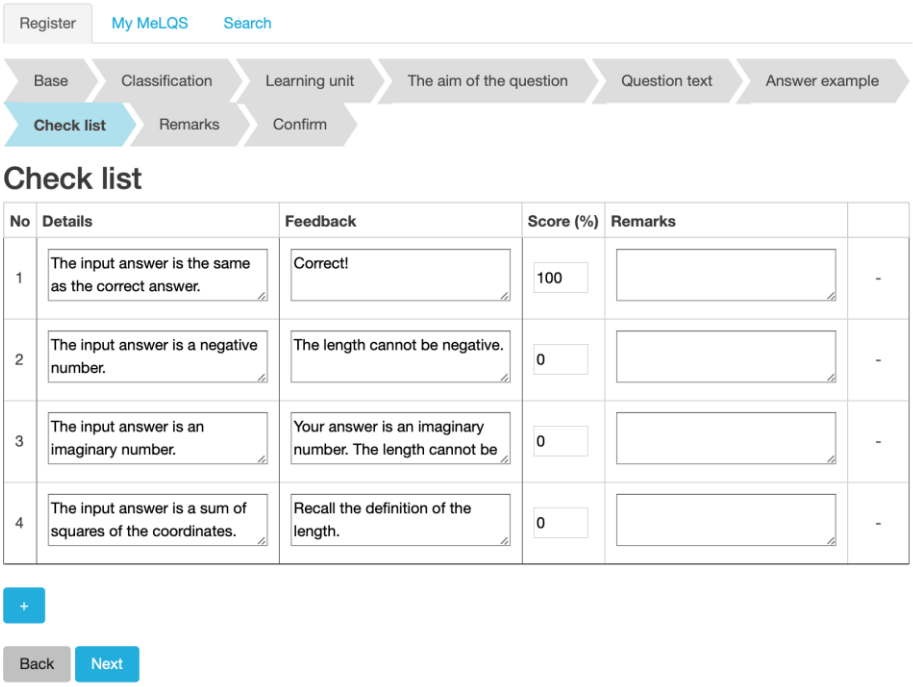

Figure 1. Screenshot of the MeLQS authoring tool

3.3. Implementation Specification

The development of implementation specification is ongoing work. Implementation specification aligns with the phase of algorithmizing the teacher's idea. The main contents of the implementation specification are the following three algorithms.

- [Algorithm 1] Algorithm for generating mathematical expressions embedded in a question text (it can be replaced by a random choice from a list of selected mathematical expressions)
- [Algorithm 2] Algorithm for generating an answer example
- [Algorithm 3] Algorithm for marking students' answers (and identifying students' errors)

Implementation specification consists of the above algorithms, a pointer to a concept design, and a question text with explicit insert positions of mathematical expressions generated by Algorithm 1.

Implementation specification is created from concept design data, but it does not necessarily have to be created by the same person who created the corresponding concept design. Anyone who can understand the concept design and can algorithmize it may create an implementation specification.

An authoring tool for implementation specification is still in progress. The user interface of the tool will be developed with a block programming interface like Blockly (Google, 2018), and the data will be stored as XML. A sample of a marking algorithm described with Blockly is shown in Figure 2. In Figure 2, 'input_answer', 'mark', 'feedback', and 'correct_answer' are variables for a student's answer, score, feedback message, and correct answer, respectively.

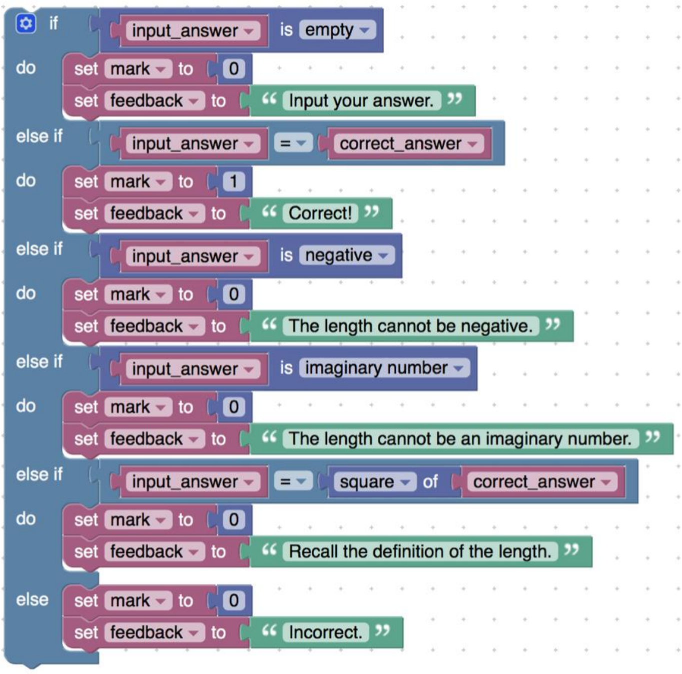

Figure 2. Sample of a marking algorithm described in implementation specification

To make descriptions of algorithms simple, each algorithm needs to be described with blocks representing mathematical functions or mathematical procedures. A sufficient number of blocks of mathematical functions/procedures will be given as preset blocks, but custom blocks could be created if they are needed.

3.4. How to implement questions based on implementation specification

Finally, we mention the implementation of questions based on the implementation specification, which is ongoing work. Questions can be implemented in each system by writing codes based on the data described in the implementation specification. Implementation is performed by a person who has the skill to write codes for a target system. Implementation is intended to be performed manually at this moment. Developing an automatic export tool from the implementation specification, although it might be partially automatic, is a future task. To develop such a tool, libraries for each target system must be developed, consisting of codes of the target system representing mathematical blocks used in algorithms in the implementation specification.

4. Conclusion

In this work, with the observation of the common data structure of three math e-learning systems, MeLQS, which can be used in different CAS-based math e-learning systems, was proposed. MeLQS consists of two types of specification: concept design and implementation specification. An authoring tool for concept design was developed. An authoring tool for implementation specification is still in progress, but the idea of using a block programming interface in describing algorithms is outlined. Implementation based on the implementation specification is supposed to be performed manually at this moment. An automatic export from the implementation specification is a future task.

5. Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP16H03067.

6. References

Abacus.aalto.fi, 2018. *Abacus project*. Available at: https://abacus.aalto.fi. [Accessed 9 November 2018].

DigitalEd.com, 2018. *MÖBIUS ASSESSMENT – ONLINE ASSESSMENT SYSTEM FOR STEM COURSES.* Available at: https://www.digitaled.com/products/assessment/ [Accessed 20 November 2018].

Google.com, 2018. *Blockly*. Available at: https://developers.google.com/blockly/ [Accessed 12 November 2018].

Kawazoe, M. and Yoshitomi, K., 2017. E-learning/e-assessment systems based on webMathematica for university mathematics education. *MSOR Connections*, 15(2), pp.17–24.

Kinnear, G. and Sangwin, C., 2016. MapleTA question importer. Available at: https://github.com/maths/moodle-qformat_mapleta [Accessed 9 November 2018].

Maplesoft.com, 2018. Maple for Education & Research: The Essential Tool for Mathematics. Available at: https://www.maplesoft.com/MapleEducation/ [Accessed 28 November 2018].

Maxima.sourceforge.net, 2018. Maxima, a computer algebra system. Available at: http://maxima.sourceforge.net [Accessed 28 November 2018].

Nakamura, Y., Taniguchi, T. and Nakahara, T., 2014. Item bank system for the mathematics e-learning system STACK. *Electronic Journal of Mathematics & Technology*, 8(5), pp.355–362.

Nakamura, Y., Yoshitomi, K., Kawazoe, M., Fukui, T., Shirai, S., Nakahara, T., Kato, K. and Taniguchi, T., 2018. Effective use of math e-learning with questions specification. In: J. Silverman and V. Hoyos, eds. *Distance Learning, E-Learning and Blended Learning in Mathematics Education*, ICME-13 Monographs. Springer.

Numbas.org.uk., 2018. Numbas: Really versatile maths e-assessment. Available at: https://www.numbas.org.uk [Accessed 12 November 2018].

Osakafu-u.ac.jp, 2018. *MATH ON WEB: Learning college mathematics by webMathematica*. Available at: http://www.las.osakafu-u.ac.jp/lecture/math/MathOnWeb/ [Accessed 12 November 2018].

Sangwin, C., 2013. *Computer aided assessment of mathematics*. Oxford: Oxford University Press.

Shirai, S. and Fukui, T., 2017. MathTOUCH: Mathematical input interface for e-assessment systems. *MSOR Connections*, 15(2), pp.70–75.

Stack.ed.ac.uk, (2018). STACK. Available at: http://www.stack.ed.ac.uk [Accessed 28 November 2018].

The Mathematical Association of America, 2018. *WeBWorK*. Available at: http://webwork.maa.org [Accessed 12 November 2018].

Wolfram Research, Inc., 2018. Mathematica, Version 11.3. Champaign, IL.