Mathematics teaching & learning and XML

Peter James Rowlett
Nottingham Trent University

peter.rowlett@ntu.ac.uk

A previous project [1] (the author's MSc dissertation) investigated the potential for dynamic production of Mathematical Markup Language (MathML) [2] by creating an online assessment system using PHP.

The author has enrolled as a research student to continue this work and to investigate the application of other eXtensible Markup Language (XML) [3] technologies to mathematics teaching and learning. This document will outline some of the ideas connected with this new project.

XML

XML is used to encode information so that it is machine-readable. This could create a custom made XML language or use one that has been developed as an international standard; for example, MathML encodes mathematics for access by computers. Other XML standards mentioned in this document include:

- eXtensible HyperText Markup Language (XHTML) for webpages;
- eXtensible Stylesheet Language Formatting Objects (XSL-FO) for printed documents (ready conversion to PDF);
- Scalable Vector Graphics (SVG) for graphics and vector-based animations;
- eXtensible 3D (X3D) for 3D graphics;
- Digital Accessible Information SYstem (DAISY) for Digital Talking Books (DTB).

Using the XML language eXtensible Stylesheet Language Transformations (XSLT), a document in one XML format can be converted to another.

1. Mathematics Assessment

MathML-based assessment system: Rowlett [1] outlines the production of a MathML-based multiple-choice assessment system. A brief overview is given here, however, interested readers are encouraged to read that article.

The system takes question formats written using MathML, which contain the question statement, the correct answer and a number of distractors. These contain XML markers, which identify constants that might be pseudo-randomised and places where other functions might be included. For example, the product rule is encoded as the product of two markers where functions might be included. An XML control file contains limits for the pseudo-randomised numbers and suitable functions for inclusion. This control is needed to prevent generation of absurdly difficult or trivial questions.

The system creates questions according to the rules in the question formats and control file. Arbitrary constants are determined pseudo-randomly in the question statement and calculated in the multiple-choice answers. When the system combines question formats together it produces a composite correct answer and distractors. The system has some built in mathematical manipulation, sufficient to check that the multiple-choice answers offered are distinct.

For testing and evaluation purposes a set of very basic question formats on differentiation were created (see Fig 1) along with multiple-choice answers. These were combined to give the question formats listed in Fig 2, as well as their (composite) multiple-choice answers.

n	nx	x^n	nx^m
$\sin(f(x))$	$\cos(f(x))$	$e^{f(x)}$	$\ln(f(x))$
$u+v$	$u-v$	uv	$\frac{u}{v}$

Fig 1 Mathematics used in the test system – the rules for differentiating these expressions were encoded

These 8 questions, with their distractors and suitable limits for pseudo-randomised constants are capable of generating large numbers of questions (something in the region of 9 million distinct questions).

Of course, these eight questions are only a small subset of those that can be produced as combinations of the basic question formats in Fig 1. Using the existing system, many more basic question formats can be written, on many different topics. Potentially, any piece of mathematics that can be described in MathML (indeed, in XML) could be used as the basis for a question format.

Questions on topics such as basic differentiation, which can be written in several different contexts, could be generated with differing wordings. This would add to the variety of the test, or allow the same set of questions to be customised for different kinds of engineers, say.

MathML conversion to LaTeX, PDF, Mathematical Braille, etc. might be used to enhance the test output. Facilities to generate tests for printing out could be produced, either to assist lecturers in setting written tests or to provide non-computer based tests for students with disabilities who might benefit from such a format.

Other dynamically generated assessment: One form of dynamic element is the pseudo-randomised number, as used in the system described above. The process could also be used to include other dynamic elements, such as information from databases or input from the user.

Text and mathematics: The test system above demonstrates the potential for producing mathematics questions with pseudo-randomised numbers. Equally, questions could be written around data pulled live from a database, or previous user input could be used to create 'follow through' questions.

Diagrams: By generating SVG code, diagrams with dynamic elements could be created. So a diagram for a trigonometry question could have pseudo-randomised angles, say. Diagrams could be combined together to build up more complicated figures.

Perhaps more interestingly, one can use SVG to plot graphs to accompany questions, either from functions (possibly pseudo-randomised) or from dynamic data sources. Moreover, a question could ask the user to input a function and plot this for feedback (i.e. "incorrect: this function does not pass through the origin" and a graph illustrating this).

3D environments: Using the 3D graphics language X3D, one can create 3D environments with dynamic elements. These might vary the parameters of a simulation, the height of a slope, say. One could even imagine a 3D animation based on a pseudo-randomised equation of motion. Then an incorrect equation could create an animation demonstrating the "unnatural" behaviour generated for feedback, making a link between the abstract mathematics and the real-world scenario being modelled.

Q1 - 3

$(ax+b)^m$

$2 \le |a| \le 10$, $2 \le |b| \le 10$, $1 \le n \le 3$

$Q1: 2 \le m \le 5$

$Q2: m = \frac{1}{2}$

$Q3: m = -\frac{1}{2}$

Q4

$\ln(ax^n + b)$

$2 \le |a| \le 10$, $2 \le |b| \le 10$, $1 \le n \le 3$

Q5

$e^{nx}\sin(mx)$ or $e^{nx}\cos(mx)$

$1 \le |n| \le 4$, $1 \le |m| \le 4$

Q6

$e^{\frac{1}{n}x}\sin\left(\frac{1}{m}x\right)$ or $e^{\frac{1}{n}x}\cos\left(\frac{1}{m}x\right)$

$2 \le |n| \le 4$, $2 \le |m| \le 4$

Q7

$\frac{a+bx}{c+dx}$

$1 \le |a| \le 5$, $1 \le |b| \le 5$, $1 \le |c| \le 5$, $1 \le |d| \le 5$

Q8

$\sin(a+bx^n)$ or $\cos(a+bx^n)$

$1 \le |a| \le 5$, $1 \le |b| \le 5$, $2 \le |n| \le 3$

Fig 2 Specifications for the eight questions used in the test

2. Accessible mathematics materials

The more one learns about creating accessible learning experiences the more apparent it becomes that in order to cater effectively for the needs of all students (with disabilities or otherwise) it is necessary to provide a wide range of learning experiences. For example, students might be given an algorithm or process to learn. Some will be able to follow this from a printed page of notes; some might find it easier to have notes marked up (underlined, encircled) in a particular way. Some students will learn more effectively from a diagrammatic approach to the learning, yet more will appreciate an animation to explain the concepts. Some students will not be able to learn without the chance to practice the algorithm or process (perhaps a great number of times). And students might have even more particular needs or preferences.

A student with a learning difficulty might find they can only learn from one of these methods. Consequently it is necessary for a truly inclusive learning environment to provide all methods, to accommodate all learning styles.

Lecture notes: Lecture notes encoded in a custom XML format could be converted to other XML languages using XSLT. Suitably encoded lecture notes could be output as XHTML for webpages or as XSL-FO to be converted live to PDF for download and printing. These could even be encoded in the DAISY 3 format to produce a structured digital ''text-only'' document for use with a braille display or speech synthesizer.

In a piece of mathematical working, some students might benefit from having no "logical leaps", by having each step very clearly explained. Other students might prefer not to have every step explicitly spelled out for them, requiring a lower level of detail. Yet more students might learn concepts best if they have been left to discover them on their own, so working should be left absent, with a note like "derivation left as an exercise for the reader."

The XSLT language allows the choice of which parts of the source file to convert. Then lecture notes might have variable levels of detail ("derivation left as an exercise", partial explanation and full derivation, say) accessed according to the student's preferences. In addition, a lecturer might maintain a single file containing lecture notes for a course from which the XSLT can extract any single lecture's notes, slides for each lecture or a set of revision notes for the full course.

By dynamically generating Cascading Style Sheets (CSS), the style of a web page can be varied. Then a user could specify their preferences for viewing lecture notes through some web interface. One student might wish theorems to be displayed with a blue border, another might prefer red. A student might require large print or high contrast notes. By dynamically generating the style sheet all such preferences can be accommodated.

As well as having lecture notes outputted as words, the document structure could be accessed to produce a "spidergram" diagram. This could help students with a particularly visual learning style.

A lecturer would need only a single file containing lecture notes, and from this they could extract overheads for lectures; students could access the lecture notes, greatly customised to their requirements, and revision notes. The benefits of producing all of these options from a single source file are clear.

Algorithms or processes: A system can be envisaged in which an algorithm or process in mathematics can be encoded in a custom-made XML language, with set ways to encode, say, the cancelling out of a fraction or cancelling out of brackets. Then one could design wordings, diagrammatic elements, animations, etc. to fit with this language. This custom XML code could be converted to other XML languages:

- The wordings could be used to create written notes (could be webpage, print or Digital Talking Book formats).
- A base of diagrammatic elements could be used to create flow charts or diagrams in the SVG format.
- Simple animations could be combined to create an animated movie of an algorithm, in SVG.
- The XML language that encodes the algorithm could also be associated with commands to a Computer Algebra System (CAS), potentially creating an interactive assessment tool.

This could illustrate the algorithm's application to a relatively arbitrary mathematical expression. All of these outputs are from a single information source. Standard algorithms and processes in mathematics could be encoded and the written notes, diagrams, animations and even the self-assessment exercises could all be generated from this. The overhead on staff time creating the online learning materials would be drastically lower than if all had to be created separately. If and when the materials must be updated, they need only be updated once, and the notes, diagrams, animations and even the assessment system could all be updated.

3. Other XML implementations

The ideas in this document relate to a few implementations based on the XML specification. A part of the project will be concerned with investigating a wider range of XML applications and assessing their potential relevance to mathematics teaching and learning. For example, the Resource Document Framework (RDF) specification allows the encoding of logical associations in a machine-understandable way. The application of this technology to mathematical proof teaching will be explored.

This project is to be completed on a part-time, (currently) self-funded basis. Because of these restraints this work will proceed at a relatively slow pace, and can afford to exercise patience. Whatever the next 5 years bring in XML can be examined and assessed.

References

[1] Rowlett, P.J. 'Pseudo-randomised CAA by "preprocessing" MathML'. *In*: Beevers, C., ed. *Maths CAA Series* [online], September. Maths, Stats and OR Network, 2004. Available at: http://mathstore.ac.uk/articles/maths-caa-series/sep2004/
[2] MathML: http://www.w3.org/Math/
[3] XML: http://www.w3.org/XML/