OPEN PRAXIS

INTERNATIONAL
COUNCIL FOR OPEN AND
DISTANCE EDUCATION

Designing an Interactive OER Course Development at Athabasca University Based on ODL Principles

Hongxin Yan & Sandra Law
Athabasca University (Canada)
hongxiny@athabascau.ca & slaw@athabascau.ca

Abstract

Failure rates in first year calculus courses are high in most post-secondary institutions across North America and other parts of the world. This Inukshuk-funded open education project involved the development of five stand-alone pre-calculus learning modules. The design and revision phases of this project occurred between the fall of 2007 and late spring of 2009. These modules were designed to support learners enrolled in first year calculus by providing just-in-time instruction in five areas: algebraic operations, factorization, polynomials and rational expressions, radical expressions, linear and quadratic equations. One of the major challenges of the project was developing dynamic activities that could support the display of a variety of mathematical formulas. To this end an open source Flash-based authoring tool was developed called the Athabasca University Tutor Authoring Tool (AUTAT). This paper explores the design and development of the AUTAT based on the needs assessment and design principles discussed.

Keywords: Authoring Tool; Flash-based; Interactivity; Pre-calculus; ODL Principles; OER

Resource - Project Course Site: http://ocw.lms.athabascau.ca/course/view.php?id=5

Introduction

The primary goal of the project was to improve completion in first-year calculus in the distance education context. Introductory calculus is a popular course at universities across Canada as it is a prerequisite for higher studies in engineering, life sciences, medicine, technology and economics. It also has one of the lowest completion rates of all courses offered at the introductory level. The modules developed for this project were intended as stand-alone, open educational learning activities for learners registered in calculus courses requiring a refresher in algebra. Member institutions of the Canadian Virtual University (CVU) assisted in the evaluation of the modules and of the content.

Needs Assessment

Completion rates in first-year calculus at Athabasca University between the years 1999 and 2008 were reviewed as part of background research for this project. A literature review showed that low student retention and high failure rates in introductory postsecondary mathematics courses were also seen in other jurisdictions such as the U.S.A (Hoyt & Sorenson, 2001), Australia (Wilson & MacGillivray, 2007), Ireland (Hourigan & O'Donoghue, 2007), Europe (Brandell, Hemmi & Thunberg, 2008), and Asia (Tang, Voon & Julaihi, 2009; Lak, 2005). Low completion rates in introductory mathematics courses at the postsecondary level have been attributed to a variety of causes:

- Inadequate student preparation at the secondary level for postsecondary level mathematics (Varsavsky, 2010).

- Focus on preparation for examinations rather than skill development.
- Lack of adequate support for at-risk students registered in postsecondary introductory mathematics courses.

Design Process

Selection of a development platform

During the initial stages of this project a variety of authoring platforms were evaluated based on their alignment with a set of design parameters:

- Not reliant on server application, stand-alone
- Supports display of mathematics
- Can be used by novices with no programming skills (e.g. simple interface, documentation)
- Available for public use, not a single implementation
- Utilizes non-commercial open source software
- Sustainable over time
- Allows for the addition of metadata

Three different development platforms were selected for further analysis: ASPIRE, ActiveMath, and Carnegie Mellon University's (CMU) Cognitive Tutor Authoring Tool (CTAT) Version 2.3 (Carnegie Mellon University, 2007). Only the CTAT was a stand-alone application, the other two were server-based. ActiveMath provided native support for mathematics display while the CTAT had no native support for mathematics. Two of the platforms (ASPIRE and CTAT's example tracing tutor) did not require content authors to have programming skills. There was evidence that ActiveMath and CTAT were more sustainable choices because they were actively being used to create courses. Two of the platforms were open source (ActiveMath and CTAT). Only ActiveMath had a built-in mechanism for adding metadata. While ActiveMath met many of the criteria needed for this project a major barrier was the fact it was server-based. This fact led the team working on the project to select the CTAT.

Online mathematics display

MathML was selected as the format for displaying mathematics because it had several advantages:

- More accessible to learners with disabilities than other formats (e.g. PDF, LaTeX, images).
- Expressions can be copied into equation editors and computer algebra systems that recognize MathML.
- Expressions render quickly unlike images.
- Native rendering is available in many browsers (e.g. Firefox, Safari, Chrome and Opera) while freely available players (e.g. MathPlayer) allow for display in others (e.g. Internet Explorer).

There are some drawbacks of using MathML such as the lack of universal browser support. However, this problem can be resolved by using MathJax along with MathML.

Set of design principles

Stand-alone

One of the main considerations in the design of these modules was that they should be developed in a flexible format that could be utilized as stand-alone learning objects or incorporated into course materials in a learning management system (LMS).

Adaptable content

The goal was to make the different components (assessment, instructional, practice) of the module as re-usable as possible. For example, a decision was made not to link between components in a module or between modules, as this would reduce the independence of modules and their components.

Opportunities for practice

The modules were designed with the intention of allowing learners to gain some content knowledge and to test their understanding through practice.

Timely feedback

Given that these modules were designed to provide timely feedback to students already registered in introductory calculus courses, the development team felt that it was necessary to provide learners with the ability to assess knowledge or skill deficits with pre-tests, and then allow the learner to choose the modules/learning activities that would best rectify those deficits. The design of feedback was modeled on the recommendations of Piccinin (2003). This author argued that feedback should be timely (Chickering & Gamson, 1987), provide specific suggestions on how to improve performance, and confirm what learners have done right so they can repeat effective practice.

Development of the OER Course

AUTAT (AU Tutor Authoring Tool) Design and Development

Why a New Tool Was Needed

As discussed early, the CTAT had many desirable features: stand-alone, open source, and a built-in mechanism for providing feedback. However, it did not support the creation of complex mathematical expressions with MathML, a mathematical display format that was used in this project. In the end, the developer concluded that a new tool was needed in order to achieve the learning goals mandated by the project.

An OER API Used for the New Tool Development

One of the major technical challenges associated with the development of the tool was the need to support display of mathematical expressions. Many different kinds of expressions were used in the modules: quadratic equations, long division, square roots, etc. As MathML was to be used, a MathML parser (Sorenson, 2009) was needed to generate these expressions efficiently and satisfactorily. The option of developing a custom parser was explored but not pursued because of the time that would have been required to develop it. Fortunately, an open source MathML parser capable of generating math expressions was found through an Internet search. The API was written in ActionScript3 for Flash. Testing showed that this API worked very well for almost any type of algebraic symbol. It was estimated that the API could reduce development time by two thirds. This showed that it was technically and financially feasible to create a new Flash-based authoring tool.

The New Tool Interface Design Ideas

Since the Cognitive Tutoring Authoring Tool (CTAT) had many desirable features, the new tool was based on the CTAT user interface design. The Hint button is located in the top, right-hand corner of the tool. Learners click on the button to view contextual hints that are intended to assist them in

solving the problems posed. Immediate feedback opens at the bottom of the screen once the user has selected an answer. Because it was inspired by CTAT, this new tool was named AUTAT–Athabasca University Tutor Authoring Tool.

AUTAT Development Framework

In order to ensure optimal re-usability of components and efficiency of the development process, it was necessary to separate content from presentation when using the AUTAT. This approach also meant that the tool had to be suitable for non-expert users with no programming skills. The presentation component was programmed in Flash with ActionScript, and the data contents were created as XML. In this project, the MathML is the actual data saved in the XML file. Data was read by Flash at the AUTAT initialization stage when running, and then the MathML was parsed as mathematical expression to the Flash stage. Figure 1 shows the framework for the tool development.

AUTAT Interface and Features

Exercises can be created by the AUTAT. Figure 2 shows the Exercise interface. The interface elements include:

- Title (e.g. "Exercises" in this case).
- *Navigation Bar:*
  The navigation bar is automatically generated on the top-right of the screen based on the questions created in the XML file.
- *Question body:*
  The question number, question statement, and answer options are dynamically read from the XML file.
- *Hint Button:*
  Learners can select the Hint button to obtain help to solve the question.
- *Feedback:*
  Once the user selects an answer option, feedback displays. Most questions provide detailed feedback.

Features:

- *Question types:*
  Multiple-choice, true/false, and matching questions are supported by the AUTAT.
- *Number of questions:*
  The number of questions that display in an activity can be customized through the data file.
- *Images and Video support:*
  The tool also supports images and video. In the question body, images and video can be embedded.
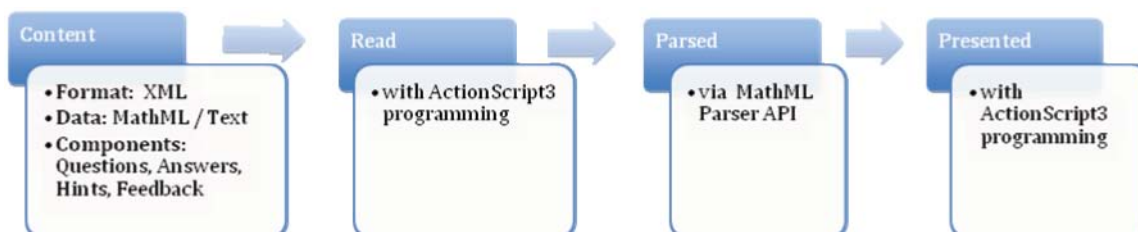
Figure 1: AUTAT Development Framework

Figure 2: AUTA Exercise Interface
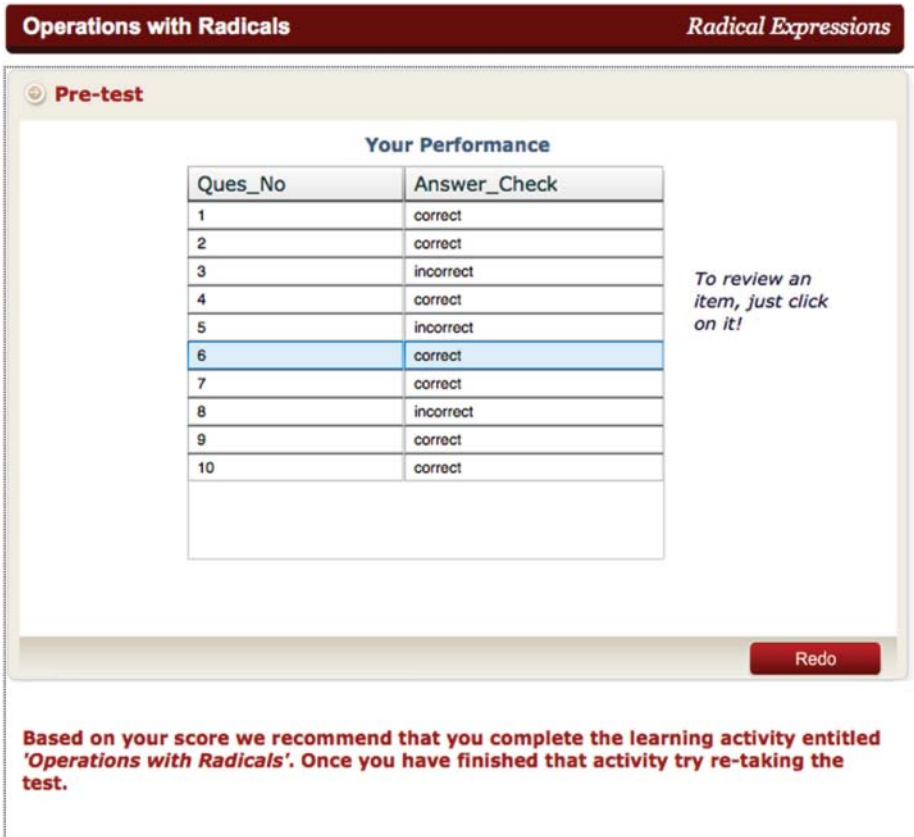
Another Two Activity Types Created by AUTAT

Besides the Exercises, AUTAT also creates another two activities – Assessment and Tutorials. They both have similar interfaces and features to the Exercises, but also have their own functionalities as listed below.

Assessment (pre-test, post-test)

- *Randomization*
  Assessments consist of questions selected randomly from the question bank. This allows for diversity in the questions displayed in the assessments and allows students to re-take a test a few times.
- *Summative Feedback:*
  Results of the assessment are summarized in a table. Learners can re-attempt the assessment as many times as they want (see Figure 3).
- Unique feedback is provided based on the overall score.

Tutorial

- The tutorial structure shown in Figure 4 requires learners to work through a question or problem in a step-by-step manner.
- The tutorial activity provides the branching feature for learners to explore different solution paths.

Figure 3: Assessment Summative Feedback

AUTAT Content Editor

Open educational resources (OERs) have the potential to reduce course development costs. The intention was to design the AUTAT in a way that would allow content to be created by content experts with no programming expertise.

Since the content is saved in XML file format, users can create XML content with text editors, such as Notepad or Dreamweaver. However, XML editors are not easy for non-experts to use. Thanks to contributions of a Flash designer, an AUTAT content editor was designed that works for non-experts. This tool allows questions to be easily added, modified and deleted in the design mode. The preview function provided by this editor further enhances data entry efficiency (see the interface in Figure 5).

The new interface makes use of another open source software - FMath Editor (Alexandru, 2011). The FMath Editor can edit equations or mathematical symbols on web pages using MathML, LaTeX or OMML (Microsoft Word) standards. By embedding this editor in the AUTAT, even novices can use this tool to create their own question banks.

Sharing the AUTAT with the OER community

This project benefited from open source/education movement in a few different ways:

- AUTAT was modeled on open source software (e.g. CTAT interface).
- Display of mathematical expressions was made possible by an open source API (i.e. MathML parser API).
- FMath editor gave non-experts a chance to develop their own content using the AUTAT.

Figure 4: Tutorial Example

Figure 5: AUTAT Content Editor

Athabasca University wanted to give something back to the open educational community because it had benefited from that movement. To that end the AUTAT and pre-calculus modules were made available on the following OERs: Curriki, WikiEducator, Merlot and the Athabasca University Open Courseware Website.

Module Structure Developed in the Course

This OER course was named Math Support for Calculus. Five modules were developed in this course:

1. Algebraic Operations
2. Factorization
3. Polynomials and Rational Expressions
4. Radical Expressions
5. Linear and Quadratic Equations

Figure 6 describes the structure of the modules. The colored boxes are the final components that can be accessed by learners. The blue boxes denote static contents, and the yellow dynamic contents.

Evaluation

The evaluation of the modules was undertaken by two different groups of reviewers who work at institutions that are part of the Canadian Virtual University (CVU): six instructional designers and two content experts. The reviewers were provided with a set of criteria upon which they were asked to evaluate the modules. The criteria included: ease of use, organization, effectiveness of instruction, value of the feedback provided to students, and learner support features. Reviewer comments are summarized in Table 1.

Conclusion

Completion rates in first year calculus courses are low in many jurisdictions around the globe. The modules developed for this project were intended as just-in-time learning activities for students
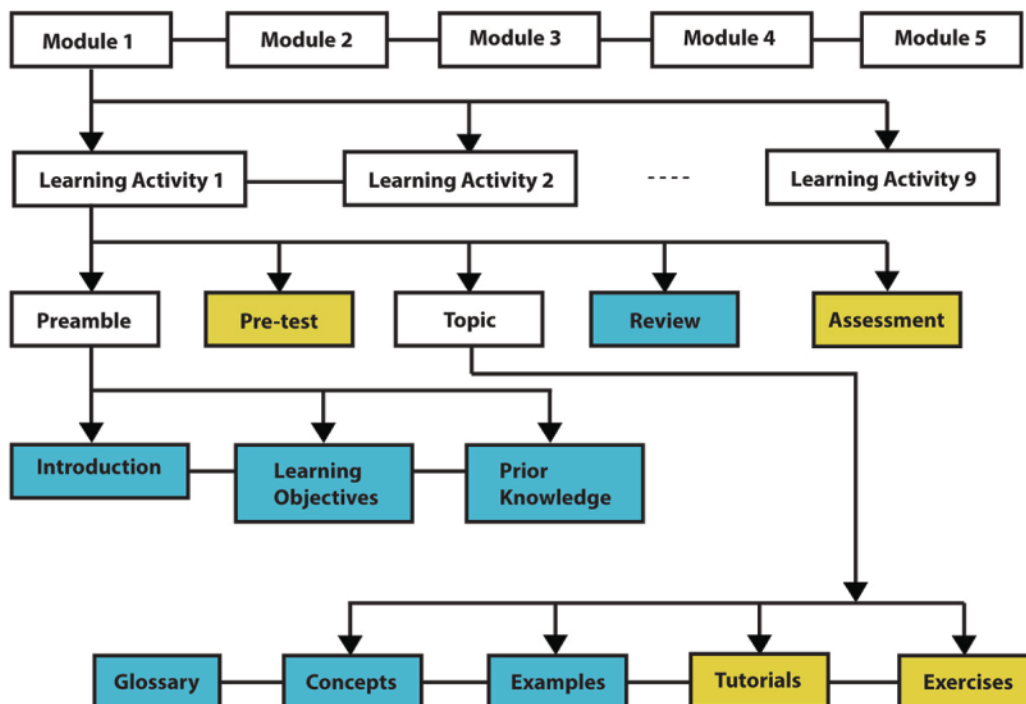

Figure 6: Course Modules

Table 1: Summary of instructional designer and content expert comments on modules

	Rating	
Item	Functions well	Needs improvement
Feedback	Immediate and provides clear direction to students as well as supporting independent learning	• Improve quality in tutorials • Disable hint button when no hint is used
Ease of use	Easy to navigate	Inconsistency in numbering steps in tutorials
Learning objectives	Clearly stated	
Sustainability		Sustainability of technology (e.g. Flash)
Instruction	• Good ideas on difference and sum of cube in factorization module • Glossary good resource	• Difficulty level too high for some exercises
Learner support	Technical support good	• Provide legend to explain specific mathematical symbols • Provide concrete examples to illustrate proofs and theorems
Organization	• Well organized • Step by step setup good way to provide worked examples	Needs master table of contents
Assessments	• Difficulty level appropriate for most part	Reduce number of questions in pre-test

enrolled in introductory calculus. Specifically, they were designed to address gaps in student understanding that content experts had identified as being particularly problematic (e.g. algebraic reasoning). After the modules were completed they were placed in a live calculus course. There is no definitive data at this point on whether these modules have impacted student outcomes. With improvements in learning analytics it is hoped that more data can be collected about whether students are accessing these modules and how students are using the modules in the context of introductory calculus. Work that was done on these modules has informed the design and development of mathematics courses, for example, in the use of MathML to display mathematical expressions.

The design requirements of the project necessitated the development of a new tool, the AUTAT. The AUTAT has been made available to the general public by submitting it as well as its documentation to a variety of open education sites like Curriki, Merlot, WikiEducator, and the AU Open Courseware (OCW) site. Modifications have been made to the AUTAT to improve its usability for people who are not technical experts.

Acknowledgments

References

Alexandru, I. (2011). *FMath Editor*, Available from http://www.fmath.info/editor/download.jsp

Brandell, G., Hemmi, K. & Thunberg, H. (2008), The widening gap – a Swedish perspective, *Mathematics Education Research Journal*, *20*(2), 38–56.

Carnegie Mellon University. (2007) Cognitive Tutor Authoring Tool (CTAT) (Version 2.3). Available from http://ctat.pact.cs.cmu.edu/index.php?id=software

Chickering, A. W. & Gamson, Z. F. (1987). Seven principles for good practice in undergraduate education. *AAHE Bulletin*, *39*(7), 3–7.

Tang, H. E., Voon, L. L. & Julaihi, N. H. (2009). A Case Study of 'High-Failure Rate' Mathematics Courses and its Contributing Factors on UiTM Sarawak Diploma Students. Paper presented at the *Conference on Scientific and Social Research*. 14–15 March 2009.

Hourigan, M. & O'Donoghue, J. (2007). Mathematical under-preparedness: the influence of the pre-tertiary mathematics experience on students' ability to make a successful transition to tertiary level mathematics courses in Ireland. *International Journal of Mathematical Education in Science and Technology*, *38*(4), 461–476. http://dx.doi.org/10.1080/00207390601129279

Hoyt, J. E. & Sorenson, C. T. (2001). High school preparation, placement testing, and college remediation, *Journal of Developmental Education*, *25*, 26–34.

Lak, H. S. (2005). The gap between secondary school and university mathematics. *International Journal of Mathematical Education in Science and Technology*, *36*, 161–174.

Piccinin, S. (2003). *Feedback: key to learning.* London, Canada: Society for Teaching and Learning in Higher Education.

Sorenson, J. (2009). *MathML Parser in Actionscript3*. Available from http://www.sevenson.com.au/actionscript/mathml/

Varsavksy, C. (2010). Chances of success in and engagement with mathematics for students who enter university with a weak mathematics background. *International Journal of Mathematical Education in Science and Technology*, *41*(8), 1037–1049. http://dx.doi.org/10.1080/0020739X.2010.493238

Wilson, T. M. & MacGillivray, H. L. (2007). Counting on the basics: Mathematical skills among tertiary entrants. *International Journal of Mathematical Education in Science and Technology*, *38*(1), 19–41. http://dx.doi.org/10.1080/00207390600819029