

Cover Sheet	JISC Invitation to Tender Framework and Tools – Toolkit Project
-------------	--

Name of Institution/Organisation: Open University
Name of Partners (if any) None
Name of Proposed Project: Translation of mathematical content as a web service
Full Contact Details for Primary Contact: Name: Dr Jonathan Fine Position: Technical Developer Email: J.Fine@open.ac.uk Address: LTS Strategic The Open University Milton Keynes MK7 6AA Tel No: +44 1908 659891 Fax No: +44 1908 653639
Length of Project: 7 months
Project Start and End Dates: 1 December 2006 to 30 June 2007
Total Funding Requested from JISC: £22,979
Outline Project Description This is technical project to solve some problems that stand in the way of handling mathematics in electronic media. It opens up pedagogic possibilities. At present, the problems are being solved in a patchwork of piecemeal ways. This project helps unify print and electronic approaches. There are three major standards for mathematical content, namely the open source program TeX for mathematical typography, and LaTeX and MathML for plain text representation. Also, there are many graphics standards in which mathematical formulae can be represented. This project will provide a web service that translates mathematical content from a substantial and useful subset of LaTeX to MathML and vice versa, and to several graphics formats. The new software that does this will be packaged for deployment on other web servers. Together, this will make it easier for other systems to cope with mathematical content. TeX will be used for mathematical typography (selection of font and size for characters and symbols, and their placement and spacing) and for the parsing of LaTeX input. Existing open source tools will be used to generate graphics from TeX's typeset output. The project makes an important technical innovation, which is to 'harden' TeX so that it is sufficiently secure for deployment on a mainstream web server. It also relies on two existing innovations. The first is to use TeX to parse LaTeX encoded data, without also executing it. The second it to run TeX as a daemon. These innovations are of value in their own right, and this project can be thought of as a demonstrator for them.

Introduction

Mathematical content presents particular difficulties in both print and electronic media. Accented and non-latin characters used to cause electronic difficulties, until Unicode replaced the previous 8-bit (256 possible values) character sets. Now it is possible to cut and paste Greek, for example, from a web page and into a Word document. But not mathematics.

The project will contribute to the program by easing the problems caused by mathematical content in electronic media. We expect its contribution to grow over time, as its key tools become more widely accepted.

An ideal solution would

1. Accept LaTeX input
2. Use TeX typography
3. Convert to and from MathML
4. Provided graphics outputs in multiple formats
5. Be efficient, and support interactive (instant preview or wysiwyg) use
6. Run everywhere, from web server to stand-alone PC
7. Handle the full range of published mathematical constructions
8. Integrate with other applications, e.g. that need to edit or display mathematical content

The Higher Education community is a long way from having an ideal solution. There is at present a patchwork of partial solutions. The most widespread solution, for print, is to use LaTeX in the traditional way. This fails, for electronic media, on 3, 5, 6 and 8. It is also very closely tied to using LaTeX not only for the mathematics but also the text. Our proposal will address these points but will leave further work on 6 and 7 to realise the ideal solution.

There is, of course, pressure to remedy the deficits in any partial solution. This can lead to unanticipated and uncomfortable programming requirements, or worse development dead ends.

This project will start on 1 December 2006, and finish before the end of June 2007.

Related work

Mimetex (<http://www.forkosh.com/mimetex.html>) is a fairly simple C program that implements an approximation to a subset of LaTeX syntax and TeX mathematical typography. It can be readily deployed on a web server to provide safe, reliable, efficient translation of mathematics into PNG bitmap graphics. Mimetex, in its external interface, is a close approximation to the deliverable this project will provide.

AsciiMath (<http://www1.chapman.edu/~jipsen/mathml/asciimathdemo.html>) depends on the browser having MathML capabilities. It translates a LaTeX-like syntax to MathML, which is then passed to the browser for rendering. It provides a very nice stand-alone instant preview environment. AsciiMath is written in JavaScript.

MathPlayer (<http://www.dessci.com/en/products/mathplayer/>) is a freely available (but not open source) MathML plugin for Internet Explorer. It allows IE to run AsciiMath. It is from DesignScience, who also sell the MathType plugin for Microsoft Word.
(<http://www.dessci.com/en/products/mathtype/default.htm>)

JsMath (<http://www.math.union.edu/~dpvc/jsMath/>) is in broad terms similar to AsciiMath. However, instead of relying on the browser having MathML support, it implements in JavaScript a fairly good approximation to TeX's mathematical typography algorithms. It also implements an approximation to the LaTeX syntax.

This project differs from the others in that it uses TeX for its mathematical typography and for parsing LaTeX. This gives it a very high degree of compatibility with TeX and LaTeX as they are used for the

production of print materials. It also means that it is much easier to extend it so that it supports the full range of LaTeX syntax and mathematical constructions.

Finally, we mention **DOMÉ**. This is part of the Serving Mathematics project, funded by JISC funded in 2004-5. The purpose the DOMÉ component is to provide a web service interface to open source translators of mathematical content.

The DOMÉ component of the Serving Mathematics project plan clearly describes the importance in a Virtual Learning Environment of translation of mathematical content. It also makes the case for this to be made available as a web service.

http://www.jisc.ac.uk/uploaded_documents/Serving_Maths_Project_Plan_R1.doc

This project differs from the DOMÉ in that it provides a new translator of mathematical content, rather than an interface to existing translators. This project will, where appropriate build upon and reuse the work done by DOMÉ.

Project Description

The project will provide a toolkit for translation of mathematical content. It will accept large subsets of both LaTeX and MathML as input. Its possible outputs are LaTeX, MathML, and graphics in bitmap and outline standards, such as PNG and SVG. The mathematical content in the OU's entry level suite for mathematics will be in the admissible input subset.

The main deliverable will be open source software, packaged for installation on Linux and Unix web servers, that will make this translation available as a web service. In addition, this project will until July 2008 make the translation publicly available as a web service.

The key features of this translation software are

- It will use TeX for mathematical typography
- It will use TeX to parse LaTeX input
- TeX will be hardened to meet web server security constraints
- It will run TeX as a daemon, to remove its long startup time
- It will translate LaTeX to and from MathML
- It will translate TeX's output to PNG, SVG, EPS and PDF using existing open source programs

Standards used

Name of standard	Version	Notes
TeX	3.14	TeX is the de facto standard for mathematical typography, to which all other systems aspire.
MathML	2.0	MathML is the W3C standard for mathematics as XML.
LaTeX	2e	LaTeX is the most widely used input syntax for encoding mathematics as plain text. It is the syntax that most other systems emulate.
PNG	1.0	PNG is a W3C recommendation for bitmap graphics.
SVG	1.1	SVG is a W3C recommendation for vector graphics.
EPS	2.0	EPS is a vector and bitmap graphics standard widely used in print production workflows
PDF	1.6	PDF is an open file format created and controlled by Adobe for

		paginated documents.
--	--	----------------------

As conversion between and use of standards is the crux of this project, some further comments are provided.

TeX is a 'document compiler' written by the esteemed computer scientist Don Knuth, so that his multi-volume series on The Art of Computer Programming would be typeset to the highest standards. Don Knuth consulted with typographers, and studied mathematical typography, when writing TeX. As a result, the quality of its mathematical output has not been surpassed since its introduction in 1982. TeX is remarkably bug-free, and produces effectively identical output on all platforms.

LaTeX is a front end to TeX, written in TeX's macro expansion language. It translates the user's input into the primitive typesetting instructions provided by TeX. LaTeX became stable in 1993. The majority of academic mathematics and physics papers and books are written using LaTeX. LaTeX plays a role similar to style sheets in HTML and XSLT in XML, although as a language it is very different from both.

MathML was introduced by the World Wide Web Consortium (W3C) in 1997, to allow mathematics to be placed on web pages. It was adopted by some computer algebra systems, particularly Mathematica. In 2002 Design Science introduced its MathPlayer MathML plug-in for Internet Explorer, and in 2005 Firefox was released with built in MathML support.

Technologies and Design Choices

TeX will be used for mathematical typography. It is both the best and most widely used system for typesetting mathematics. However, TeX will be run as a daemon rather than as a batch process. This is similar, for example, to the `mod_perl` extension to Apache. It allows TeX to process document snippets quickly, by avoiding startup costs. This work has already been developed, as part of the TeX daemon project. <http://sourceforge.net/projects/texd>

Used in the ordinary way, TeX makes no distinction between program code and document data. This gives it enormous flexibility, but exposes the host server to denial of service and other attacks. The project will use the `tex2tok` package to avoid this problem, by enforcing a distinction between code and data. <http://www.ctan.org/tex-archive/help/Catalogue/entries/tex2tok.html>

A key part of the project will be hardening TeX so that it can be run securely on untrusted data. A small part of the system will, out of necessity, be written in the TeX macro programming language.

The Python objected oriented scripting language (<http://www.python.org>) will be used. It is flexible and reliable, with extensive pre-existing libraries for network and web-services capabilities. It is already widely used on web-servers.

The following open source programs will be used to translated TeX's typeset output (in the dvi format) into various graphics formats.

- PostScript and EPS <http://www.radicaleye.com/dvips.html>
- PDF <http://gaspra.kettering.edu/dvipdfm/>
- SVG <http://dvisvg.sourceforge.net/>
- PNG and GIF <http://sourceforge.net/projects/dvipng/>

Quality Assurance

The software will be developed using either CVS or Subversion version control, and be most likely hosted on SourceForge.

Test data will be taken from the mathematical content in the three courses (MU120, MST121 and MS221) in the Open University's mathematics entry suite. The project plan allocates 5 days for the collection of this data and the analysis of test results. This test data will be used for both system and unit tests.

Coding will follow the Style Guide for Python Code <http://www.python.org/dev/peps/pep-0008/>, and documentation will use the extensive tools provided with Python <http://docs.python.org/doc/doc.html>

Work plan

The lead developer (68 days over 7 months) works in Strategic subunit of Learning and Teaching Solutions (LTS). LTS Strategic is responsible for development and support of software such as the Virtual Learning Environment. Work on web services and Unix administration (18 days over 7 months) will be done by other staff in LTS Strategic and in Academic and Administrative Computing Services (AACs). Here, a day is understood to be 1/20 of a full-time month.

The work divides into five phases, as below. For each phase, we indicate the major milestones and deliverables, and also portion (**N**) of the 18 days (Unix and web services support) needed. By publish we mean make available as a web service. There are 220 productive working days in a year.

A – Setting up – December 2006

Web server set up and running. Project established on Sourceforge. (2)

B – Hardening and publishing – 2 January to 16 February 2007

TeX hardened. Publish parsing LaTeX into tokens (tex2tok). Publish basic TeX typesetting, as PNG bitmap graphics. Security audit of project software installed on web server. (6)

C – Basic Mathematics – 19 February to 30 March 2007

Equations such as example 1 in the Appendix. Publish server statistics, and log errors. (3)

D – Further Mathematics – April and May 2007

Equations such as examples 2 and 3 in the Appendix. Additional functionality, driven by test data. (3)

E – Conclusion – June 2007

Publish conversion of dvi to EPS, SVG and PDF. Security audit of project software on web server. (4)

Time estimates for the 75 days of lead developer time (and the **phase** of the project) are

- Harden TeX for mathematical typography (**A, B** 15 days)
- Harden TeX for parsing LaTeX (**A** 5 days)
- Construction of parse trees from LaTeX data (**C** 5 days)
- Basic MathML input and output (**C** 5 days)
- System integration of above components (**C** 5 days)
- Back end conversion of dvi to graphics formats (**C, E** 4 days)
- Supply of test data, and analysis of results (**D** 4 days, institutional contribution)
- Adding functionality, driven by test data (**D** 20 days)
- Administration, dissemination, contingency (**A-D, E** 5 days)

Dissemination

A primary means of dissemination will be a web server, running until at least July 2008, from which the translations of mathematical formulae will be publicly available as a web service.

We will ask the Open University's Centre for Open Learning in Mathematics, Science and Technology (COLMSCT, <http://cetl.open.ac.uk/colmsct/>) and the Open University led Physics Innovation Centre of Excellence in Teaching and Learning (PiCETL, <http://cetl.open.ac.uk/picetl/>) to help and to advise with the dissemination of the results of the project.

We will also contact the partners of the JISC-funded Serving Mathematics project for help and advice in engaging the wider Higher Education community.

Risk analysis

Risk	Probability (1-5)	Severity (1-5)	Score (P x S)	Action to Prevent/Manage Risk
Staffing	1	5	5	If lead developer becomes

				unavailable, then the project cannot proceed. The lead developer is highly motivated to do this project.
Organisational	2	2	4	The 20 days Unix and web server staffing will be required at unpredictable times
Technical – TeX	0	5	0	TeX runs reliably as a daemon. This has been thoroughly tested. If not, it would be a severe problem
Technical – performance	3	2	6	If performance turns out to be unsatisfactory, we can live with it, and recode bottlenecks in C later.
External suppliers	1	1	1	No external suppliers
Legal	2	1	2	If test data cannot be cleared, freely available alternatives will be used.

Budget

Direct Staff Costs = £15,456, being
68 days lead software developer
18 days Unix and web server support

Direct Non Staff Costs = £2,000, being
Dissemination: 18 months dedicated web hosting = £1,000
Travel and other dissemination = £1,000

Total Direct Costs = £17,456
Indirect Costs = £12,365
Total Costs = £29,820

Institutional Contribution = £6,841, being
4 days lead software developer
50% indirect costs

Total cost to JISC = £22,979

Benefits to the Open University

The first objective of the University's **eLearning Policy** is to 'construct an OU-wide framework for the development of eLearning materials, services and support'. For this framework to be OU-wide we need to include in it eLearning for disciplines that rely heavily on mathematical notation. This project will help us deliver this objective.

There are two action points in the **eLearning action plan** that relate specifically to this. The first is to 'Roll out 2007 policy for student choice in submitting electronic Tutor Marked Assignments. Currently most courses in Mathematics, and some courses in Science, are excluded from this due to difficulties in students producing and tutors annotating mathematical notation.

The second action point addresses this. It is to 'pursue projects in Maths and Computing and COLMCST (CETL) looking for tools for on-screen production of mathematics notation, graphs and diagrams in student assignments.' A successful outcome to this project would provide some workarounds to the current difficulties with the eLearning action plan.

OpenLearn (<http://oci.open.ac.uk>), supported by The William and Flora Hewlett Foundation, will make educational resources freely available on the Internet, with state of the art learning support and collaboration tools to connect students and educators. Many of the mathematics courses published in this way will have been authored in LaTeX. This project will ease their translation into web pages.

Key Personnel

The lead software developer for this project is Dr Jonathan Fine, who is a Technical Developer in LTS Strategic. Dr Fine joined the Open University in 2003 to support and develop the TeX system used for the production of Mathematics and upper-level Physics course materials. At that time, most TeX work was done on a long obsolete Alpha/VMS server, using a proprietary printer driver. Dr Fine moved this production, without disruption, onto an Intel/Windows platform. He has over 15 years experience of using and developing TeX, and 5 years experience of Python. Prior to 2003 he worked for Cambridge University Press, first on developing and supporting the in-house typesetting system, and then on setting up an XML workflow for books and journals.

In March 2006 Dr Fine organised, as part of the OU's Learning and Teaching Conference, a workshop session on Mathematical Content and Distance Learning.

In June 2006 Dr Fine planned a session, as part of the Mathematics and Computing one-day workshop on the OU's Virtual Learning Environment, on existing internet resources for the teaching of mathematics.

AACS and LTS Strategic in the OU have extensive experience in developing and deploying web-services for distance learning. Up to 20 days time will be available from these sources to provide Unix and similar support for this project. It is not possible at this time to identify the precise needs or the names of staff members. This information will be provided as part of the project plan.

Appendix: Examples of mathematical content

These examples have been produced using the Word plug-in MathType. A PDF version of this file is being made available, in case they do not display for you. (This is an example of the difficulties caused by mathematical content.)

A typical page on an entry level mathematics course might have 10 or 20 equations on it, of this size. In higher level mathematics course, most of a page might be occupied with equations.

1. Equation of a unit circle: $x^2 + y^2 = 1$

The LaTeX encoding of this equation is: $x^2 + y^2 = 1$

2. Solution to quadratic equation: $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

The LaTeX encoding of this equation is: $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

The MathML encoding of the right hand side of this equation is

```
<math xmlns='http://www.w3.org/1998/Math/MathML'>
  <mfrac><mrow><mrow><mo>-</mo><mi>b</mi></mrow><mo>&pm;</mo><msqrt>
  <mrow><msup><mi>b</mi><mn>2</mn></msup><mo>-</mo><mrow>
  <mn>4</mn><mo>&it;</mo><mi>a</mi><mo>&it;</mo><mi>c</mi>
  </mrow></mrow></msqrt></mrow>
  <mrow><mn>2</mn><mo>&it;</mo><mi>a</mi></mrow></mfrac>
</math>
```

3. Definition of derivative of a function: $f'(x) = \lim_{h \rightarrow 0} \left(\frac{f(x+h) - f(x)}{h} \right)$

The LaTeX encoding of this equation is:

```
f'(x) = \lim_{h \to 0} \left( \frac{f(x+h)-f(x)}{h} \right)
```