Serving Mathematics in a distributed e-learning environment

This article describes the work of the recent serving mathematics in a distributed e-learning environment project. The project, which ran from September 2004 to the end of March 2005, was funded by the JISC and enabled an unusual level of collaboration between colleagues to develop the mathematical e-learning tools described in this article.

1. Introduction

The project was led and managed by Gustav Delius at the University of York, together with a small team of developers. At Imperial College, were Phil Ramsden, Daniel May, and Jakub Kahovec. The University of Sheffield participated through the work of Neil Strickland, and Manolis Mavrikis from the University of Edinburgh was also part of the team. James Blowey is from the University of Durham, and I represented the MSOR Network, based in the School of Mathematics at the University of Birmingham. This is a large, diverse and distributed group of individuals and, as shall be described below, the project itself was a multi-part endeavour to bring together existing projects on which team members had worked. These are as follows:

- METRIC: a self-testing and mathematical learning system based on client-side marking implemented in Java;
- AiM: a self-testing and assessment system based on server-side marking using the computer algebra system Maple, see [5], [6];
- STACK: a system similar to AiM using the open-source computer algebra system Maxima;
- WaLLiS: an interactive learning environment that focuses on providing adaptive and intelligent feedback, see [7]; and
- Moodle: a user-friendly and modular Virtual Learning Environment (VLE).

Before commenting on the project itself attention should be drawn to the website which, at Gustav’s suggestion, was built using the Virtual Learning Environment Moodle (http://www.moodle.org/). This site played a number of important roles for the project, and worked so well for the team in doing this that a short digression is in order to describe it.

http://maths.york.ac.uk/serving_maths/

One purpose of having this site was to make the project open to others at the outset, with guest access also available. A description of the project, the evolving discussions, links to test sites being developed and to other relevant projects were, and indeed still are, available online. There was great encouragement to others in the field to join the website and participate in the online discussions. The extent to which this happened, and the support from others with experience in e-learning was surprising. By the end of the project over 120 other colleagues had registered with the site as participants, and many of these contributed to the project through their suggestions. This was a great strength of the project, and the core team certainly benefited from this method of working. In addition to the public materials, the team naturally also used the site to host private discussions between members. Since the core team were geographically distributed, draft documents could be edited online by a number of people, with confidence that they were working on the definitive version. Again, although very simple this did work well and
avoided circulating email around a large group and collating responses. This is mentioned since setting up a virtual learning environment to help with collaborative working seemed like a strange idea, and to be honest I initially doubted it would work. Other collaborative projects with geographically distributed partners would do well to consider this approach in the future. The website will continue for the foreseeable future, and participation is still encouraged even though this project has officially ended.

2. Project Background

The MSOR community have a long track record in using e-learning and e-assessment, and the CAA series edited and hosted by the Network has documented much of this activity since 2001 [1]. Rather than develop yet another new system, all the project members were committed to finding common ground and trying to work together to adapt existing systems to integrate more closely. This is as difficult as it sounds given the well-developed existing mathematical e-learning tools and systems, written by the various members of the project team. A goal was to make them interoperable with other components of online learning environments, as well as with one another. There are interoperability standards to allow online questions to be exchanged and reused on different systems, such as those of QTI [2]. However, such standards only really allow a question author to create questions of a general type, such as multiple-choice, textual or numerical input. These are less interesting in mathematics, when questions can be distorted by providing the student with a list of potential answers from which to make a selection. Further, they often do not allow one to include written mathematics, or to generate such mathematical expressions from randomly generated parameter sets. Just as seriously, these do not allow an author to create questions that assess students’ answers in the form of such expressions. All the problems encountered by the general systems have been seriously addressed in various ways during previous projects by the current participating partners, and the existing software has been proven at several institutions. One aim was to provide convenient and easy-to-use interfaces with these to help colleagues create their own computer aided assessment questions for mathematics. Another was to integrate them as seamlessly as possible into their existing teaching and campus systems, such as VLEs.

The approach in the project was to develop various Web Services. The philosophy behind this is for users to collect together the relevant services into useful packages, rather than to develop their own independently. Users do not interact directly with these services, but rather applications make use of them. For example, an institution would set up a web service to authenticate access for their staff and students. The library might offer a service to access details of its holdings. These two services might be combined in an application to access a particular borrower’s records. Clearly the library are not interested in students’ records, and the Student Records Department are not interested in full details of the Library’s holdings. But some information needs to be exchanged between the two. By offering such services functions can more cleanly be separated out, duplication removed, and it is suggested that better applications more rapidly developed. The Athens system, for example, is an authentication service facilitating access to a number of quite different online sites. Full details of the JISC’s e-learning framework (ELF) can be found on their website [3]. The approach in the project was to provide mathematical web services. After all, the team do not have the energy (or inclination) to duplicate all the functions of a VLE, student records system and other online utilities. The interest of the current team members is in the learning and teaching of mathematics, with the specialised assessment and e-learning tools together with other associated support services. The services developed during the project are explained in more detail in subsequent sections.

3. Remote Question Protocol

Imagine that an academic is interested in using computer aided assessment or interactive e-learning in an Institution with a centrally supported VLE, on which students are registered. This probably has a quiz module of some kind, but almost certainly lacks the kinds of highly mathematical questions which are interesting and useful in mathematics learning and teaching. There are a number of options. One is to ignore the VLE and set up a stand-alone system. In doing this there is none of the central support an institutional VLE enjoys, and so there will be overheads with administration, such as student records. Another option is to accept what the VLE provides, and so not have access to potentially very useful mathematical tools. Alternatively an attempt could be made to integrate new question types into the VLE itself, which may fall foul of the licence agreement. This is the scenario the Remote Question Protocol (RQP) seeks to address by allowing individual questions to be slotted into another application, such as a VLE. Essentially the RQP provides questions as a web service. The random generation, marking, feedback and other processes which relate to the specific question are performed by a remote and dedicated server. The client, in this case the VLE itself, now simply packages together certain information relating to these questions. Since the approach to this problem was a rather abstract
one, it actually has nothing to do with mathematics questions. Indeed, other groups with interesting subject specific questions could offer to serve them using this RQP. By including the RQP question type in the VLE quiz, this would immediately allow all such questions to be included. This solves the problem not only for mathematical questions, but more widely, and hopefully with a degree of future-proofing to question types not yet developed. Well, this is the idea, but it hinges on both the VLE and the subject specific question system moving far enough to meet in the middle. By using a protocol, such as the RQP, the VLE does not need to be adapted for every new question type making implementation a much more realistic proposition. Initially, a quiz module for the VLE Moodle was developed, and STACK and AiM offer questions remotely. During the project, this aspect attracted attention from computer scientists willing to offer their questions, and other clients who wanted to include the authored questions into their quizzes using the RQP. This work is continuing under future projects, and developments will certainly continue to be posted on the site. This is inevitably quite technical, and the challenge is making the user interface as simple and transparent as possible.

4. MathQTI

In a sense the remote question protocol avoids the issue of interoperability: you do not need to get other people’s questions working on your server, you just use a remote server which already copes with them. However, sharing and exchanging questions that are authored in the same programming language, and having the ability to modify them with common authoring tools, is an important topic if real collaboration is to be sustained. The present project was used to develop a common specification to address the particular needs of Mathematics. The main issue is to have a mathematically solid infrastructure to support response processing of student’s answers that are given in the form of mathematical expressions. These are not multiple choice questions, or similar, which distort questions by asking the student to make selections from lists of teacher provided answers. All the systems in the project had taken this approach, but used a variety of techniques to process students’ answers.

After a series of face-to-face workshops and online discussions using the VLE’s forum, a formal XML specification was defined [9]. The language was termed MathQTI to signify that it is an extension of IMS Question and Test Interoperability (QTI) specification [2]. This is quite naturally based upon the experiences and some technical aspects of the projects involved in the consortium and particularly previous work described in Mavrikis [8]. In doing this the team have attempted to respect the QTI format, which reflects the work of a number of experienced CAA groups developed over many years. The team preferred to adapt and extend, rather than redevelop, so QTI constitutes a valuable starting point. Furthermore, building upon an existing standard ensures that when the extensions are not used MathQTI files default to valid QTI files, and so can be used in a QTI compliant system. An authoring tool for MathQTI would then be useful for QTI items as well, for example.

5. Authoring tools

It is important, for a variety of reasons, that authors can create assessment questions easily for themselves. Basic tools for this purpose exist for the respective existing systems, but these are not always user friendly and are designed primarily for the system developer. In AiM, for example, the question author needs to write short Maple code fragments to implement sophisticated feedback. In Metric question authors need to become computer programmers to implement their own questions. This is something which the team wished to avoid. Hence, the team developed an authoring tool to provide a simpler interface. This has a “Wizzard”-like feel and was focused initially towards the Metric system.

6. Display of mathematics

In order for any on-line learning material or tool to be useful for mathematics and science teaching it needs a way to allow the user to enter and view mathematical formulae. This is still a problem for web-based systems. It could be said that the promise of MathML has not been delivered, since a robust and practical tool set is hard to assemble, and native support is currently poor in the web browsers used by the majority of students. There are many work-around solutions, such as a plug-in to display MathML, but these are unsatisfactory. One reason why MathML is important is that it is a documented standard, and so allows individuals with special needs, such as a visually impaired user for example, to choose display options. Performing the display task is an ideal candidate for a dedicated service, rather than requiring each individual e-learning tool to re-implement this. The service is called SMITE, and demonstrations are available online. The service translates expressions from one format, such as LaTeX, into the most appropriate format for the user, e.g. HTML or MathML.

7. Conclusions

The project finished at the end of March. Hence it is too soon to judge “success”, other than by asking, did the
team produce a tonne of coal? That is, where is the software developed? Does it work? A longer term perspective is needed, however, to see whether the team really did form lasting collaborations, and move closer in the thinking about mathematical computer aided assessment. My sense is that this was achieved, that the project was a valuable focus for activities. Time will tell how this progresses over the coming years. The software is open-source, which means that anyone is free to download, install and look at the code. It can be freely modified if there are things you would like to change. Of course, the code does have a licence but the team chose licences which are designed to protect your rights to use and distribute the software freely. Details of the licences are available online, with a discussion of the implications of various open source versus commercial licences discussed in Zymaris [4].

References