『学習院大学　経済論集』第48巻　第2号（2011年7月）

Ubiquitous e-Learning: Designing Web Systems for Economics and Business Mathematics

Paul Trafford and Yukari Shirota*

Keywords: e-learning, Web design, open standards, computer algebra systems, Web:VisualEconoMath, How2SolveIt, mathematics teaching, business mathematics, problem solving, heuristics

1. Introduction

In this paper we survey some of the developments with regards to technology-enhanced support of mathematics learning and teaching. Coverage is especially given on how to design systems that work effectively on the Web, particularly in the fields of economics and business studies. This is a rapidly evolving landscape in which general trends in e-learning have raised expectations of the electronic tools to go beyond merely providing mechanical functions: nowadays the teacher expects them to help organise their teaching and even enhance their teaching methods. For the student, there is increasing focus on the learner experience, which seeks a wider range and availability of e-learning support tools together with greater ease of use.

Advances in technology mean that options keep changing, setting many challenges for system developers, so this paper seeks to indicate the broad features in these changes, with particular regard to future planning. These general considerations are subsequently brought into focus in later sections of this paper by addressing the question of how e-learning systems may provide judicious support for the uniquely human abilities of solving problems. In this case visualisation is an excellent aid, but any tools provided need to draw on a wide range of online resources that are readily available so as to properly establish the learning context. Finally, we briefly introduce into this context two new systems for teaching mathematics: *VisualEconoMath*, an integrated environment comprising automated tools for deriving solutions to prescribed mathematical problems and *How2SolveIt*, which supports the quintessential human learning processes surrounding questions and answers, in this case between students and lecturers.

*） *Department of Management, Faculty of Economics, Gakushuin University, Japan*

1.1 The Evolving e-Learning Landscape

The importance of e-learning to the provision of a country's education has been repeatedly underlined in recent years. As technological advances provide ever more sophisticated options for learning, government and educational bodies are placing more importance on their strategic value in contributing to the economy. A recent publication in the UK (OLTF 2011) consulted widely with students, the public sector and private sector. Overall, it recommends more closely and flexibly supporting the learner and the choices they wish to make, and to do this in a cost-effective manner through collaboration and combining resources.

Technology has come to occupy a core position in education due increasingly to general trends in society, where the usage of technology has been driven by consumer behaviour. In particular, the growth in *personalisation* of technology – which owes a great deal to inventions like the Sony Walkman – is a facet of behaviour that affects study as well as leisure. Since students are major consumers of technology, who depend on computer hardware and software to support their learning, this raises expectations concerning personalisation in learning. This is in turn contributing to a growing phenomenon of 'disruptive innovation' – *disruptive* in the sense that the teaching methods need to be adjusted to accommodate a fundamental change in learning context.

1.2 Flexible Study Modes

Such contextual changes are borne out by surveys into student use of technology. At Oxford University, a student I.T. survey is carried out each year among incoming undergraduates. For those new students who responded to a survey in autumn 2010 it is reported that 85.5% owned laptops with 6.5% owning notebooks (Littlehales 2010). In addition, over 50% have some kind of smartphone, about one third of which were iPhones. Further, the survey found that the majority of respondents (60%) were aware of University podcasts, with 21% downloading content related to their subject matter.

The precise figures vary across campuses around the world, but almost everywhere the options are diversifying, which is commensurate with the way learning is taking place in different ways and styles. For example, traditional modes continue with knowledge absorption in the classroom, a process involving mainly listening, watching and recording. Yet, nowadays more consideration is being given also to learning away from the classroom, where there emerge qualities of recollection and reflection, which are essential to solving problems. Digital technology is a boon in that it has the capacity to support many of these scenarios. In particular, outside formal lectures, the flexible delivery of teaching materials is supported through podcasts. A recent JISC-funded project, *Listening for Impact: Rapid Analysis* is assessing the impact of Oxford University's podcasting collection. Its initial report summarised feedback gained in three main points (Wilson *et al* 2010: 8):

a) Engagement and interaction with the podcasting topic for private study
b) The podcasts influenced the listeners to explore the podcasts in other Oxford subjects
c) Re-use in teaching situations

It was remarked:

> A listener said that one of the advantages of learning through podcasts was the capability to learn at their own pace: "*I have quite a bit of work to do to understand some of the trickier derivations – fortunately, I have a 'pause' button and much more time than your students do.*"

Podcasting services, which complement the standard methods of teaching delivery, are thus excellent models for providing students with greater flexibility in studying at a pace that suits them.

1.3 Open Standards in e-Learning

It can be seen how rapidly the computing landscape is changing when considering specific cases. For example, specialist 3D graphics applications that used to run exclusively in laboratories on high performance workstations are now available not only on home computers, but as Web-enabled applications on a range of networked devices. This inevitably means new developments that target new platforms, but the effort has potentially significant rewards in terms of enabling greater ease and ubiquity in use.

In the UK, the strategic consideration of how to make the most effective use of digital technologies are undertaken especially by the aforementioned Joint Information Systems Committee (JISC), the national IT body serving Higher Education[1] At a broad architectural level, the JISC has consistently promoted open and distributed approaches based on open standards – at its most recent conference delegates have even received a 'virtual goody bag' extolling their benefits[2].

In such an environment, it is expected that learning, teaching and research can be carried out with greater flexibility in terms of time and location. This has had a major impact for software in terms of how it is developed and licensed. In continuing to expand access, educational institutions have in large measure proceeded to enable a larger proportion of the e-learning they use and develop through the Internet and especially the World Wide Web. It has created a natural gravitation towards open standards and from proprietary software applications designed to be used in a fixed and enclosed environment to more open source software that place no such restrictions.

Whilst open source software carries risks, especially in terms of support and maintenance, as more educational establishments collaborate in the development of Internet-based applications, such software gains further momentum. This is particularly of relevance in e-Mathematics, where there are some major initiatives providing applications that merit serious consideration in comparison with established commercial products. So in planning future systems, designs can be made more sustainable and robust

1） Joint Information Systems Committee http://www.jisc.ac.uk/
2） *The benefits of open standards: Virtual Goody Bag*, JISC Conference 2011
http://www.jisc.ac.uk/events/2011/03/jisc11/programme/2openstandards.aspx

if they ensure a degree of modularity so that these options may – if deemed appropriate – readily form part of an overall e-learning solution.

2. Developments in e-Math systems and the Web

In this section we present an overview of existing software-based systems that offer support for teaching Mathematics, particularly for students of business and economics, as might be conducted according to text books such as Shirota (2009b). We start with general-purpose computer systems and then unfold the discussion by concentrating on design considerations for the Web. To ensure the right focus, we list a broad selection of typical requirements, set against the background of the emerging trends we have just depicted. We indicate the extent to which technical solutions exist and where further development is still required.

2.1 The Early Origins of Computer Algebra Systems

The foundations of e-Mathematics implementations depend on systems to carry out computation. The most popular class of systems for this purpose are termed *computer algebra systems (CAS)*, which have a long history. One of the earliest was Macsyma, which was a symbolic computation system developed by MIT in the late 1960s and has formed the foundations of many systems in existence today[3]. Maxima[4], which is a popular descendant of Macsyma, was established as a branch in 1982; Leydold and Petry (2010) have provided a detailed manual for use explicitly in the field of economics with examples including price discrimination, profit maximisation, and the Cobb-Douglas Production function. Over the decades many systems have been developed, a mixture of free[5] and commercial packages. Among the commercial offerings is Maple[6], which was started in 1980, and Mathematica[7], which came along in 1988. Another package, designed primarily as a numerical computing environment, but which also provides symbolic manipulation through an optional toolbox, is MATLAB, which has an even longer history[8].

It's notable that all these named systems predate the Web and were designed for specialist laboratories as mentioned above. All of them have required some adaptation and extension in order to work on the

---

3) An early research publication on Macsyma is: Martin, W.A. and Fateman, R.J. *The MACSYMA system in SYMSAC '71: Proceedings of the second ACM symposium on Symbolic and algebraic manipulation*, ACM New York, NY, USA, 1971

4) *Maxima: A Computer Algebra System, SourceForge* home page http://maxima.sourceforge.net/

5) For a succinct summary of developments in open source systems, including a detailed list, see: Joyner, D. *OSCAS – Maxima*. ACM Communications in Computer Algebra, Vol 40, No. 3/4, September/December 2006.

6) The date is narrowed to December 1980 in: Char B.W., Gedders, K.O., Gentleman, W. M. and Gonnet, G.H. *The design of Maple: A compact, portable, and powerful computer algebra system*. In. *Computer Algebra*. Lecture Notes in Computer Science, 1983, Volume 162/1983, 101-115

7) Wolfram, *S. Mathematica: a system for doing mathematics by computer*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1988

8) Cleve Moler recounts his personal experiences of over 40 years in The Origins of MATLAB, MATLAB Notes & News, December, 2004. Available at: http://www.mathworks.com/company/newsletters/news_notes/clevescorner/dec04.html

Web, but these are large systems, so it is not a straightforward task to re-design and re-engineer them. A common solution is to implement a client-server architecture in which the CAS runs server processes that listen for queries from Web clients. This input is usually transmitted in the form of expressions such as formulae and equations; outputs are subsequently returned, either as expressions (strings) or other outputs such as images. Examples include Maple TA[9], MapleNet[10] and webMathematica[11]; Maxima also offers various Web interfaces through its related projects[12].

A key advantage of this server-oriented setup is that a Web application can be situated in a particular context – such as business mathematics – and make use of specific CAS functionality without its complex general-purpose interface. Furthermore, whilst CAS often assume a significant amount of technical knowledge, Web applications can reduce the learning curve by offering suitable interfaces. In general, they can offer a greater diversity and richness of activities similar to the way the Web has really expanded general computer user. An apt illustration in the field of economics has been provided by Shirota (2006b) who has used an intelligent Web agent linked to Maple outputs to form a discourse from solution plans and guidance plans around various mathematical problems. A distinctive feature of the agent is its linkage with speech synthesis to deliver the narrative, thus enhancing the sensory impact.

Whilst these arrangements have undoubtedly brought CAS online, there remains the question of how the requirements for implementing Web-oriented mathematics teaching software more generally might differ if we step back from the established CAS and reconsider the evolving picture.

2.2 Web Software for Mathematics Teaching: Broadening the Requirements

As well as providing enhancements to CAS in terms of the existing learning context, Web-based systems can be designed to support mathematics pedagogy in new ways. The hypermedia basis of the Web, which can connect to potentially any kind of resource, allows us to specify components that provide broader coverage of the learning cycle. A sample list might include:

(i) The easy creation, editing and storage of mathematical content
(ii) Facilities for inputting and manipulating mathematical notation
(iii) Tools for computing solutions
(iv) Facilities to develop and work through instructional narratives
(v) Automated assessments of student submissions
(vi) Guidance for solving problems

The first half consists of purely technical items, whilst the second half consist more of explicit pedagogical requirements. We explore both of these in the following subsections, but we note here that

9) Maple T.A. *Testing, Evaluation and Grading Software* http://www.maplesoft.com/products/mapleta/
10) Maplesoft: *MapleNet* http://www.maplesoft.com/products/maplenet/
11) Wolfram webMathematica http://www.wolfram.com/products/webmathematica/
12) Related Projects // Maxima CAS http://maxima.sourceforge.net/relatedprojects.html

the multitude of elements presents a considerable challenge in terms of overall instructional design. It may even be argued that there is a need for a “grammar of interactivity” in onscreen communication analogous to a grammar of language for verbal communication (Lester 2000).

2.3 Mathematical Input

We start by considering the second item – the long-standing issue of mathematical input and display for the Web. Many of the early efforts to display mathematical content consisted of typesetting in TeX or LaTeX and then converting the formulae and equations into images. These are still useful tools, especially when provided as a Web service, for example TeX2PNG[13)] a tool for converting a LaTeX formula into PNG graphic image.

However, few students can be expected to master LaTeX or any other mathematical typesetting language and even for academic staff it can be an inconvenience especially when mathematics is not their specialism. Many of them expect user-friendly interfaces, particularly visual, with push buttons and WYSIWYG controls that can generate these formulae as easily as a word processor. There have been numerous solutions available for quite some while, but there is a considerable amount of flux. For example, WebEQ was a popular plug-in, implemented as a Java Web applet that was integrated in several major learning management systems (LMS), including WebCT (Moss 2002). It was originally developed in the mid 1990s by the Geometry Center at the University of Illinois at Urbana Champaign[14)], became a commercial product, was taken over by Design Science, and has since been discontinued[15)].

Yet Design Science has played a major role over an extended period in developing applications that involve creating and processing mathematical content for presentation, so a brief examination of how its approach has evolved can be an informative yardstick for developments in this area. Its most widely deployed product is the well-established equation editor in Microsoft Word, which is a simplified version of MathType. A new product is MathFlow, which extends various document authoring tools, including XML editors such as oXygen, with user-friendly graphical input facilities. Of general significance is that all these products are based on MathML[16)], an international standard format for mathematical content that encodes semantics as well as structure.

Perhaps of broadest significance is the work that targets Web browsers, especially in relation to accessibility. Many countries publish legislation that stipulates that educational resources should cater for people with disabilities, especially the visually impaired – for example, in the UK there is the Special Educational Needs and Disability Act 2001 (DfES 2001). In order to implement this online, it is good practice to adhere to guidelines published by the World Wide Web consortium, in the form of the Web

---

13) Nawouak.net: TeX2PNG: http://www.nawouak.net/?cat=informatics.tex2png+lang=en

14) WebEQ: http://www.geom.umn.edu/software/WebEQ

15) Design Science: WebEQ End of Life notice: http://www.dessci.com/en/support/webeq/eol.htm

16) *Mathematical Markup Language (MathML) Version 3.0*, W3C Recommendation 21 October 2010 http://www.w3.org/ TR/ MathML/

Content Accessibility Guidelines[17]. National bodies may publish further guidelines also, such as the Japanese Industry Standards policy for Web Content (JIS 2004).

Among the most common visual disabilities are dyslexia and partial vision. Content delivered should thus pay careful attention to the presentation, including the layout, size of fonts and colour schemes. Where sight is impaired further, as in blindness, there is a need to deliver audio alternatives (Soiffer 2009). Technical solutions usually involve some form of screen readers that operate in several ways. Perhaps the most common solution is a tool that runs on the operating system, such as JAWS[18] for Windows and VoiceOver[19] for the Apple MacIntosh. However, there are platform-independent Web-oriented solutions: for instance, the ATBar[20], a continuation of the JISC TechDIS Accessibility Toolbar project, uses JavaScript to provide extra navigation options that can invoke a remote speech synthesis service to read the page.

By implication the display of mathematical equations using MathML is thus preferable to images. Design Science's effective replacement for WebEQ is MathPlayer, which is a browser-based plug-in to render MathML. However it is more restrictive in deployment compared with WebEQ as it is designed only for Internet Explorer for Windows[21], though there are ways to make documents work in Mozilla-based browsers[22]. With the crop of new mobile devices and browsers, there is a pressing need to develop for these platforms also. Design Science has thus joined in the MathJAX project – "an open source JavaScript display engine for mathematics that works in all modern browsers"[23]. Their move into open source with its much broader organisational involvement as well as the adoption of open standards exemplifies the general trends indicated in section 1.3.

There are many other competing products, including the LiveMath suite[24], which provides the MathEQ Expression Editor and the LiveMath plug-in, with unusual capabilities for CAS interaction, particularly in carrying substitutions in equations. However, like MathPlayer and other proprietary plug-ins, code needs continual updating to target the rapidly evolving Web browsers. In terms of open standards, ASCIIMathML[25] is a JavaScript library that converts simple "calculator-style" syntax into MathML and graphics; and the WIRIS plugin provides mathematical editing for the general purpose

---

17) World Wide Web Consortium (W3C) Web Accessibility Initiative. *Web Content Accessibility Guidelines (WCAG) Overview* http://www.w3.org/WAI/intro/wcag

18) *JAWS for Windows Screen Reading Software* http://www.freedomscientific.com/products/fs/jaws-product-page.asp

19) *Apple – Accessibility – VoiceOver* http://www.apple.com/accessibility/voiceover/

20) ATBar http://access.ecs.soton.ac.uk/ToolBar/

21) Design Science: *MathPlayer Installation check* http://www.dessci.com/en/products/mathplayer/check.htm

22) Design Science: *Creating MathML Web pages* http://www.dessci.com/en/products/mathplayer/author/creatingpages.htm#InteroperabilityConsiderations

23) MathJAX. http://www.mathjax.org/

24) LiveMath Software Products http://www.livemath.com/

25) ASCIIMathML: Math on the Web for everyone http://www1.chapman.edu/~jipsen/mathml/asciimath.html

WYSIWYG HTML editor, Xinha[26]. We also mention that there are other technological solutions such as digital pens with built-in computing, as described in Sastri (2009), but research is still in its early stages and Sastri concludes that further research is still needed to address the human interface issues. Even so, these technologies will continue to improve and should remain under consideration.

2.4 Working Online and Offline

The mechanics of handling mathematical notation for the Web needs to be deployed flexibly and repeatedly in an overall e-learning context. We note that the Web makes it easy to offer multimedia audio and video, which is especially useful when there is limited face to face contact. The client-server arrangement works very well in many cases to incorporate mathematical equations in dynamic environments, perhaps most successfully in classrooms and workshops, when there is a reliable network connection. However, outside the classroom, such connections are not always available and in fact many Web applications do not necessarily require subsequent access to the Internet once they have been downloaded and installed on a host computer. This is apparent when considering the 'app' model for handheld devices in which applications are downloaded once from an online store and then may be run *offline*, entirely independently of the Internet. It has proved very popular and it is a similar to podcasting in which users subscribe to particular channels and opt to download media that they're interested in. As indicated in the OxTALENT survey cited above, this mechanism is familiar to most students and one that can be used for educational materials.

The validity of the offline mode in mathematics teaching is amply illustrated by Brown and Sastry (2007), who survey many of the ingredients involved in delivering online mathematical materials which can be studied at one's own pace. They describe how they introduced at Cranfield University an interactive and personalised e-learning mathematics package in which content has been already prepared or pre-packaged, so there is no requirement for interaction with a server using scripts. The content includes interactive instructional materials, exercises, and electronic self- assessments. The presentation is varied, comprising text, graphics and embedded video clips. A great deal of this content is stored as Adobe Portable Document Format (PDF), *a de facto* standard which can consolidate many kinds of materials and media, for example allowing Flash to be embedded. The mathematics package provides a convenient series of self-contained digital workbooks that can be used without an Internet connection. However, PDF is relatively weak on structuring content and information exchange: although it is an open format, the specification is complex and, unlike XML, the PDF code is not human readable (Bragdon *et al*, 2008).

2.5 Standards-based cross-platform support for CAS

In the last few years, especially since the advent of the 'smartphone', devices have multiplied in shape and form, offering a great range of screen sizes and resolutions (larger as well as smaller) and additional user interfaces, especially the use of touch. As already intimated in the discussions about accessibility, this has increased the importance of the design for a wider range of user interfaces and user agents,

26) WIRIS plugin for Xinha http://www.wiris.com/plugins/docs/xinha

including Web browsers. One other particular development we note here is the *e-book reader* (or *e-reader*). PDF usage has been prescient in this regard, anticipating many of the requirements. However, early incarnations of dedicated readers have been designed to substitute for static content – books and newspapers. Multimedia support is limited: for instance, as of writing Kindle devices do not play video due to hardware restrictions in the display.

As addressing such diversification in hardware becomes impractical on a device by device basis, it makes sense to refer to Web standards and interoperability, which applies not only to the 'front end' but permeates the entire e-learning environment. Recently tools have emerged that deliver traditional CAS functionality more oriented towards using Web standards such as JavaScript and CSS for presentation, and open standards for communication. For example, WebMathematics Interactive 2 (WMI2) (Kovács 2011), has adopted a client-server architecture and is built entirely on open standards and open source software, the whole package available on SourceForge[27]. It runs with almost full functionality on recent versions of the major browsers. Making it openly available has also helped with internationalisation – interfaces are available in 10 languages, including Hungarian, German and English

In WMI, the Maxima CAS carries out the computation, PHP scripts process Web requests that are communicated by the Web client in XML using Ajax (Garrett 2005). The use of Ajax means that it can execute dynamic updates to the output without refreshing the whole Web page. This is particularly useful for carrying out transformations on graphs, for which buttons are provided to execute functions such as zoom and rotation. The main distinguishing feature of WMI2 is its calculator-style interface, similar in concept to a graphing calculator, offering a large selection of Maxima's functionality without requiring knowledge of its input methods[28]. The interface can, if desired, be bypassed by an appropriately crafted URL, though there is not yet any formally defined Web service. There are also issues in terms of the outputs delivered to the Web: currently WMI translates Maxima output into images, in a similar manner to TeX2PNG. Whilst there are options to save content as Web pages in workbook format, this content is not delivered as text, so the semantic information is lost and there are issues concerning accessibility. However, the software could be extended to ensure the server also deliver textual outputs that can be used by screen readers, possibly as MathML.

In his paper Kovács (2011) also surveys some of the other CAS projects designed expressly for the Web and under active development. These include Sage[29], which aims to provide an open source alternative to commercial products such as Maple and Mathematica, implemented in Python as a programmable workbook environment and adopting a Web-based graphical interface since 2007[30]. Another is GeoGebra, which has gradually been expanding its functionality – GeoGebra[31] is rooted in

27) SourceForge: *WebMathematics Interactive*, http://sourceforge.net/projects/wmi/
28) An online demonstration of WMI2 is available at: http://matek.hu/
29) *Sage: Open Source Mathematics Software*: http://www.sagemath.org/
30) *William Stein introduces Sage* (video). http://www.sagemath.org/help-video.html
31) GeoGebra home page http://www.geogebra.org

geometry, but extends to the the fields of algebra and calculus. Thus it makes extensive use of visualization and helps to foster experimental and problem-oriented learning of mathematics (Hohenwarter and Preiner 2007). GeoGebra has an extensive community of users and some experiences of its use in economics have been reported in Chryssa and Tyrovouzis (2010).

GeoGebra has been developed as Java applets, but currently requires a standard Java runtime environment, which is not supported on most handheld devices. Even though this requirement is gradually being relaxed, it will not run on iPads and iPhones as iOS does not support Java. Furthermore, its CAS capabilities are limited as the current release (version 3.x) depends on MathPiper[32)] for its CAS, with functionality comparable with scientific calculators. However, development is active to allow Maxima as the CAS; also one of the nascent projects, GeoGebraMobile[33)], aims to translate the Java code into JavaScript. This presents a new set of challenges, especially in the field of graphing – efforts to render 3D graphs are only at very early stages[34)].

2.6 Delivering Video and Animations through the Web

The creation and consumption of video as a multimedia format has grown very rapidly – as we have already seen, they are becoming ubiquitous through podcasts. They also have considerable instructional potential, especially for demonstrating visualizations, so we summarise some of the main developments in this area, focusing on the media format.

The most popular format is Adobe Flash. Support for Flash playback on desktop and laptops is fairly universal, but the picture for mobile devices – smartphones and tablet PCs – varies: for instance, on Google Android, only versions from 2.2 (Froyo) onwards officially support Flash 10.1[35)]. As with Java, for devices running iOS, there is no application that runs on the machines natively, so Flash cannot work off-line. Once again limitations arise associated with proprietary formats.

If Web standards continue to be a central focus, then the long-term solution lies in the emerging HTML5 draft standard[36)]. HTML5 provides support for video[37)]: in the above case it may be possible to carry out transcoding beforehand and then deliver that as an embedding in HTML5, using standards based codecs such as H.264/MPEG-4, thus allowing offline viewing. Furthermore, HTML5 also provides the ‘canvas’ element to support graphical objects and animations, including gestures commonly used in touch screen devices. Although it has yet to be formally ratified, code for rendering HTML5 is

32） MathPiper: http://www.mathpiper.org/

33） GeoGebraMobile wiki http://www.geogebra.org/trac/wiki/GeoGebraMobile

34） See, for example, Dean McNamee *Pre3D, a JavaScript 3d rendering engine.* http://deanm.github.com/pre3d/ which has been used for a 3D graphing calculator: http://www.graphycalc.com/

35） Adobe Press Release: *Adobe Announces Availability of Flash Player 10.1 for Mobile,* 22 June 2010 http://www.adobe.com/aboutadobe/pressroom/pressreleases/201006/06222010FlashPlayerAvailability.html

36） World Wide Web Consortium *HTML5: A vocabulary and associated APIs from HTML and XHTML.* http://www.w3.org/TR/html5/

37） World Wide Web Consortium *The Video element- HTML5*: http://www.w3.org/TR/html5/video.html already being optimised in many Web browsers. This is a major improvement over exporting CAS animations as animated GIFs, which has hitherto been the limited choice available. Furthermore, the support for mobile devices is already prominent with a selection of developer application frameworks such as PhoneGap[38], Sproutcore[39] and Sencha Touch[40].

In summary, it is likely that HTML5 will feature strongly in future Web-based CAS; it is already in the sights of GeoGebra and WMI2.

3. How to Solve Mathematics Problems through e-Learning

As we have seen, the technological components to support mathematics teaching are progressing rapidly, but many constraints remain. We highlight some of these issues by focusing in this section on the requirements for e-learning systems to try and tackle the knottier problem of helping students make breakthroughs in conceptual understanding.

3.1 Requirements for a Business Mathematics e-Learning System

The visualization of data offers a powerful aid to understanding in the field of economics. As has been argued in Shirota (2009: 69), the visual approach exceeds the purely algebraic approach because it enables speedy modelling and prompt analysis of the target problem. Computer-generated models may provide a useful interface to explore in graphical form economic relationships expressed as variables, formulae and equations. When deploying these visualizations for Web browsers, a number of different requirements arise reflecting the diversity of scenarios in which such models may be used. In the case of visualisation of financial mathematics, there are several that are important to consider:

(1) The demonstration of key concepts by the teacher.
(2) Formal learning by students in classrooms (where the teacher is available).
(3) Self-study by students away from the classrooms.
(4) Question and answer feedback between student and lecturer.

Here we choose to focus mainly on the latter two points as we examine the area of assessment (thereby treating points 5 and 6 of the teaching list of section 2.2).

3.2 e-Assessment

There is a growing body of Internet tools able to provide electronic or automated assistance to deliver feedback to the student in the context of both formative and summative assessment. In our case, we are particularly interested in *formative assessment*, a reflective process intended to promote student

38) *PhoneGap* http://www.phonegap.com/
39) *SproutCore* http://www.sproutcore.com/
40) *Sencha Touch Mobile JavaScript Framework* http://www.sencha.com/products/touch/

attainment over a period of time. Student-teacher exchange is crucial to such assessment and involves judicious feedback.

There are several e-learning environments with assessment tools for mathematics, but only a few that incorporate a CAS. One is Assessment in Mathematics (AiM)[41)], an open source project developed at York University in the UK designed to provide feedback tailored to each response (Walker and Delius 2004). A key strength is its ability to invoke Maple to check student inputs to a variety of questions by recognising equivalent answers. Technically, AiM is similar to WMI2 in following the client-server approach, though Java servlets provide the interface between the Web browser and the CAS instead of PHP.

Its successor, STACK[42)], uses Maxima in the setting of mathematical quizzes and is also integrated into Moodle.

In setting the system in the study context, the authors make an important observation about how students may deepen their learning through reflection:

> Students should still be encouraged to take problems home and carefully work out their solutions, rather than attempt quiz questions that can be easily be dealt with while sitting at a computer. The key is to employ technology to motivate students to work on challenging problems, so that they reflect on their solutions and the approaches they have selected.
>
> (Walker and Delius, 2004:5)

Similarly, Kovács states: “Students have the time and opportunities to solve problems using a wide range of tools, including textbooks, the internet and computer software” and goes on to note that his students’ feedback identified this as “an important element for them in examination preparation and the more user friendly the software was, the more willing they were to use it at home” (2011: 2).

At the heart of the pedagogic consideration is the process of reflection. To what extent can technology support this process? How much human intervention is still needed? As noted by Sastri (2009) “*timely intelligent* feedback … still remains a big challenge” (our italics). In his brief survey of the literature, Sastri notes that the feedback given by teachers to their students is valuable input into a system’s design: “The responses collected from a cohort provide a rich repository which can then be further analyzed for providing more precise and targeted feedback and design remedial measures” (page 107). This then raises the question of how such feedback is generated and how best to communicate that feedback in ways students find most helpful.

If student submissions can be accurately marked – perhaps even through handwriting recognition systems – then one approach is to determine the common points of weakness and direct students to

41） *AiM: Assessment in Mathematics,* SourceForge project http://sourceforge.net/projects/aimmath/

42） STACK: System for Teaching and Assessment using a Computer algebra Kernel http://stack.bham.ac.uk materials accordingly. However, that may not go far enough to effectively assist the student in getting to know how to solve the problems; there may be a limit to how far drill practice improves the performance. Consideration of this question in light of experiences about the application of heuristics has prompted the authors here to suggest that it may be best aided by the human touch.

3.3 Introducing How2SolveIt

Whilst various software tools exist for handling mathematical formulae, there has been a general lack of support for students in finding strategies for solutions. The work of Shirota (2006b) describes e-Math Interaction Agent 2, an intelligent agent, a key feature of which is the simplicity of its solution plans and presentation definitions. It goes further and presents *guidance plans*. Yet students might need further help where they make mistakes and where they have problems in understanding.

A well-established technique to aid solution in the mathematical domain is to use heuristics as described by Pólya (2004). It has been shown that software-based systems can be useful in supporting students in such reasoning: Shirota (2009) has illustrated its application to business mathematics by the provision of a deduction system on rule databases that supports the student in developing their ability to carry out inference. However, for a student to properly understand the process they must possess sufficient information at each step. This may require feedback at key steps to help them gain the insight necessary to solve the problem, particularly which heuristic to apply. As machines are generally unable to make this selection, it requires human intervention. This is the motivation behind How2SolveIt, a question and answer system between a student and a lecturer. The requirements have recently been described by Trafford and Shirota (2011), so we only highlight here a couple of key features in the assessment.

In this system, students submit handwritten scripts to be marked and include remarks where they have difficulty in producing answers. The lecturer marks these scripts also by writing in freehand and then digitises the annotated paper. Next, to help guide the student more closely, they record a screencast narration in which they navigate the student script, draw attention to lecturer annotations, and point to supporting materials – relevant formulae, solution plans and video demonstrations (for which hyperlinks are created). The lecturer uses their expertise and experience to make these vital selections according to how they assess the individual student and the cohort as a whole.

Thus in this system much of the content generation and overall workflow is automated, but at key points in the assessment the personal contribution of the teacher or lecturer is captured in a naturalistic way both visually and in audio. One rationale for this is that students respond more to humans than computers, perhaps picking up tacit signals as much as the explicit instruction.

3.4 Content Management

The multitude of source materials and different kinds of activities and pathways that make use of them require careful organisation and are best served through an integrated e-learning environment tailored to the needs of teaching maths for economics. However, as the online engagement with mathematics increases, it raises another burgeoning issue: irrespective of whether the setup is server-oriented or client-based, there is a need for content management in order for such systems to scale. Data needs to be stored, edited, exchanged and presented in multiple contexts.

These considerations typically raise tensions between efficient systems that may be thin on learning versus inefficient systems that are rich in pedagogic value. Hand-crafted content that is specially programmed and variously authored from mixed sources may be a delight for the student but costly and impractical to prepare and administer, whereas the deployment of a single CAS may be limited to certain kinds of laboratory activities. The storage format also needs consideration – MathML may be a recognised standard, but it is complex, so may not be the most suitable even for internal communication, whereas custom XML schemas can be designed to allow easier conversion, with the option of MathML output. Such schemas allow simpler editing interfaces, particularly based on XForms[43)], the next generation of Web forms, for which prototypes already exist in various programming languages, including Java (Raudjärv 2010). However, the structuring and semantics are weak when compared with MathML.

There is a need to find some middle ground – the provision of an integrated environment that still offers custom functionality with significant pedagogic merit. Accordingly, the client-server architecture and CAS merit further consideration, especially when set against developments in Web standards. This may help us to establish an appropriate supporting role.

In assessing the design for an overall system that incorporates both a variety of tools specific to economics and business mathematics, along with the specific problem-solving facilities, the authors have found that from the users' perspective there are essentially three different but interdependent areas – for authoring, content management and Web delivery. The basic structure of the system specification is thus conceived as follows:

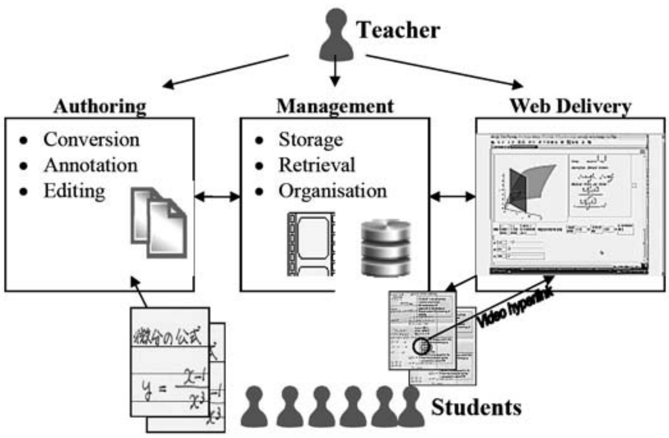

Figure 1: How2SolveIt System Architecture

43) World Wide Web Consortium: The Forms Working Group http://www.w3.org/MarkUp/Forms/

It is intended that the components of How2SolveIt will make use of World Wide Web Consortium standards or drafts, as described above, including (X)HTML and CSS for the presentation of Web pages, XML for data storage and export, together with XSLT specifications for transforming XML to various Web outputs. Implicit in this view are the accompanying resources drawn from Web:VisualEconoMath, an e-learning environment tailored to the study of economics consisting of a suite of tools, many of which are client-server scripts that make calls to Maxima. It is envisaged that content management will be implemented by a relational database management system (RDBMS), which is ideally suited for the storage, querying, and updating of data relating to the math word problems. Content may be exported to XML formats as needed.

For students reviewing their annotated scripts, the workflow and user interface design need to be highly polished. Templates with simple layouts containing intuitive navigation may be used for a consistent look and feel. The designs should work on all Web-enabled devices that can play multimedia – audio and video. Reflecting the podcasting mode of delivery, students may be able to 'subscribe' to certain newsfeeds alerting them to availability of their annotated scripts and subsequently download the associated content.

3.5 Organisational Considerations

There are also organisational issues that surround software development, which need to be considered, especially in relation to the longer term objectives of the software being developed. Is the e-learning system to be established in a group or a department? Where teaching resources are limited, an institution may look to centralise its computing systems. This is commonly reflected – in the UK at least – in Higher Education establishments choosing to commission a centrally hosted learning management system (or virtual learning environment). This then can impact on the development of a particular product – its future may to a significant extend depend on how well it can fit in the wider context. This is why a modular architecture may be preferable.

Looking further afield, it may be hoped that the mathematics e-learning system is adopted by other institutions. The planning for this is substantially more difficult. By way of illustration, WME: Web-based Mathematics Education (Wang *et al* 2003,2008) shows the long-term investment and considerable planning by Kent State University in a sophisticated environment that involves all stakeholders – learners and teachers in the processes of learning mathematics. Client delivery makes use of Web standards – outputs are delivered typically in HTML and JavaScript, using XSLT to process educational content authored in a custom markup, Mathematics Education Markup Language (MeML). However, there is little sign of WME having been deployed elsewhere, so the framework does not have much evidence of portability. It may be observed that computation is carried out by individual WME services, services that could be alternatively carried out by dedicated and more powerful CAS. The precise reasons for fruition may be difficult to determine, but it is usually essential to cultivate a vibrant community around the software.

4. Conclusions

As the use of the Web for e-learning continues to expand, it is important to recognize the limits as well as the potential. If the target environment were to remain fixed then it would be a relatively straightforward matter to plan the development of e-learning software. However, as this survey has indicated, there have been major changes that affect the learning landscape, due substantially to the emergence of mobile computing.

The variety of processes and activities that contribute to the mathematics learning process gives rise to a multitude of software pieces. It makes sense to present them as an integrated learning environment. However, it is evident that an all-purpose multi-function environment may not be adequate in giving students the help they most need. There may be limits to what automated systems can achieve by themselves in addressing the more subtle requirements of helping students realize breakthroughs in conceptual understanding. This is where the emphasis can usefully shift to providing automated support for uniquely human facets of communication and instruction. It is the rationale behind the concept of the new student-teacher question and answer system, How2SolveIt. It is hoped that the development of such systems will be steps towards showing how e-learning may truly enhance traditional pedagogy.

References

(Hyperlinks were checked on 24 April 2011.)

1. Ancsin, G., Hohenwarter, M. and Kovács, Z. *Geogebra goes Mobile*, whitepaper,22-10-2010. Available from: http://www.geogebra.org/trac/wiki/GeoGebraMobile
2. Bragdon M., Burk, A., Charlong, L., Nugent J. *Practice and Preservation — Format Issues.* Chapter 30. Part IV: Methodologies. In Schreibman, S. and Siemens R. (eds.) *A Companion to Digital Literary Studies,* Blackwell 2008.
3. Brown, I., Sastry, V. *Digital Technologies and Multimedia Content for Developing e-Learning Materials for Maths*, Proceedings of the 2nd International Conference on e-Learning (ICEL), Columbia University, 2007, pp. 63-74
4. Chryssa, C. and Tyrovouzis, P. *Exploring the effectiveness of open source dynamic mathematics software in teaching economics: The case of Geogebra*, 18th European Economics Education Conference, The Association of European Economics Education, Annual Conference, 25-28 August 2010, Manchester http://www.economicseducation.eu/index.php?page=presentations-conference-2010-2
5. DfES. *Special Educational Needs and Disability Act 2001*, Department for Education and Skills, HMSO, London, UK http://www.legislation.gov.uk/ukpga/2001/10/contents
6. Garrett, J.J. *Ajax: A New Approach to Web Applications*. AdaptivePath.com, 2005. http://www.adaptivepath.com/ideas/e000385
7. Hohenwarter, M. and Preiner, J. *Dynamic Mathematics with GeoGebra* Journal of Online Mathematics and its Applications, Volume 7, 2007 http://www.maa.org/joma/Volume7/Hohenwarter/index.html
8. JIS 2004. *Japanese Industrial Standard for Web accessibility*, JIS X 8341-3:2004

9. Kovács, Z. *WMI2, the student's on-line symbolic calculator*, International Journal of Mathematical Education in Science and Technology, iFirst, Taylor and Francis, 2011, 1 – 11. Available online at: http://dx.doi.org/10.1080/0020739X.2010.550940
10. Lester, J. *Designing Interactive Mathematics*, Proceedings of the Asian Technology Conference in Mathematics, Chiang Mai, Thailand, December 2000.
11. Leydold, J. and Petry, M. *Introduction to Maxima for Economics*. Institute for Statistics and Mathematics, WUWien, November 30, 2010
12. Littlehales, J. *Report on 2010 Freshers Survey*, Internal Report, Oxford University Computing Services, 2010. Some text extracts are available at: http://blogs.oucs.ox.ac.uk/oxtalent/2010/11/08/freshers-survey/
13. Moss, W.F. *Mathematics in WebCT IV: MathML*. Clemenson University, v2. September 2002. http://www.math.clemson.edu/~bmoss/math_on_web/math4.htm
14. OLTF, *Collaborate to compete: Seizing the opportunity of online learning for UK higher education*. Online Learning Task Force Report to HEFCE, January 2011. http://www.hefce.ac.uk/pubs/hefce/2011/11_01/
15. Polya, G., *How to Solve It: A New Aspect of Mathematical Method* 2004: Princeton Science Library.
16. Raudjärv, R., *Dynamic XML Schema Based Web Forms in Java*, Master's thesis, Faculty Of Mathematics And Computer Science, Institute Of Computer Science, University Of Tartu, Estonia. 2010. Software and thesis available from: http://code.google.com/p/xsd-web-forms/
17. Sastri, V.V.S.S. *On E-Assessment*, Proceedings of the Defence Academies & Colleges International Conference: Network Centric Learning: Towards Authentic ePractices. Stellenbosch University, 25 – 27 March 2009, pp. 103-109
18. Shirota, Y. *Methodologies for the Automation of Web-Based Interactive Courseware for Mathematics*, Gakushuin Economic Papers, Vol. 43 (No. 1) 2006.
19. Shirota, Y. (2006b) *Efficient Method for Generation of Web-Based Courseware for Mathematics*, Information Bulletin of the Union of National Economic Associations of Japan, 27:2, 2006, pp.71 – 81. Available at: http://www.gakushuin.ac.jp/univ/eco/gakkai/pdf_files/keizai_ronsyuu/contents/4301/4301shirota-B.pdf
20. Shirota, Y., *Instruction Methods for Solving Word Problems in Mathematics Education*. Gakushuin Economics Papers, 46:1, 2009, pp. 69－81.
21. Shirota, Y. (2009b) *Introduction to mathematics for students of economics and management studies*, Kyoritsu Publishing, 2009
22. Shirota, Y., Hashimoto, T., and Kuboayama, T. *Concept Model for Solving Bond Mathematics Problems*, Public Lecture, Faculty of Economics, Islamic University of Indonesia, March 2011.
23. Soiffer, N.. *The Path to Accessible Math*, The Guelph Accessibility Conference, University of Geulph, 27 May 2009.
24. Trafford, P., Shirota Y. *Requirements for Business Mathematics e-Learning System: How2SolveIt*, Japan Society of Business Mathematics annual conference, 2011 (forthcoming)
25. Wang, P.S., Kajler, N., Zhou, Y., Zou, X. WME: *Towards a Web for Mathematics Education*. ISSAC '03 Proceedings of the 2003 international symposium on Symbolic and algebraic computation, Association for Computing Machinery, New York.
26. Wang, P., Mikusa, M., Al-shomrani, S., Lai, X., Zou, X., Zeller D. *WME: a Web-based Mathematics Education System for Teaching and Learning*, ICME 11 – TSG 22 Theme 3 the 11the International

Congress on Mathematical Education, Monterrey, Mexico, July 6 - 13, 2008.

27. Walker R., Delius G., *Integrating on-line assessment with class-based teaching and Learning: a preliminary study of the AiM Marking System*, Proceedings of the 8th Computer-Assisted Assessment (CAA) Conference, Loughborough, 2004
28. Wilson, R., Marshall C., Geng, F. *Listening for Impact: Rapid Analysis Report*, University of Oxford, Report to JISC, 7th Dec 2010