

TOWARDS A PROPERLY WEB 2.0 WAY OF CREATING AND SHARING QUESTIONS

Steve Bennett, Vamsikrishna Nuthi

Towards a Properly Web 2.0 Way of Creating and Sharing Questions

Steve Bennett

Blended Learning Unit and Department of Computer Science
University of Hertfordshire
S.J.Bennett@herts.ac.uk

Vamsikrishna Nuthi

Department of Computer Science
University of Hertfordshire
V.Nuthi@herts.ac.uk

Abstract

The promise held by ideas like item banking and question repositories do not seem to have borne spectacular fruit in UK Higher Education. Projects have come and gone but a real culture of sharing of question content does not seem to have been established. This is in spite of the very real benefits such an approach ought to bring. This paper argues that the major hindrances to this are the complexity of existing xml standards, the difficulty of browsing of questions and the inability to embrace web 2.0 ideas. It offers some results from the JISC funded MCQFM project as pointing in the direction this activity needs to go in order to facilitate greater collaboration among the community. It also believes that the actions of question and test sharing are so different from that of test deployment that potentially we need two standards not one: one for the exchange of questions, and one for their deployment.

The Practice and Potential of Computer Aided Assessment

Probably the three most diffuse and well known standards for encoding question content are Questionmark's QML then IMS's QTI 1.2 and 2.1 Each of them establish a defined group of questions that can be asked. They are:

QTI 2.1	QTI 1.2	QML
1. Choice	1. Multiple choice	1. Drag-and-Drop:
2. Hotspot	2. True false	2. Fill-in-the-blank
3. Order	3. Multiple response	3. Hotspot
4. Select	4. Image hot spot	4. Knowledge Matrix
5. Point	5. Fill in the blank	5. Survey Matrix
6. Associate		6. Likert scale

7. Graphic Match	6. Select text	7. Matching
8. Graphic Order	7. Slide	8. Multiple choice
9. Inline Choice	8. Drag object, drag target	9. Multiple response
10. Graphic Associate	9. Order objects	10. Numeric questions
11. Text Entry	10. Match items	11. Pull-DownList
12. Graphic Gap Match	11. Connect points	12. Ranking
13. Extended Text		13. Select-a-blank
14. Position object		14. True/False
15. Hot Text		15. Wordresponse
16. Slider		16. Yes/No.
		17. Adobe Flash
		18. Adobe Captivate Simulations
		19. SpokenResponse
		20. Java

There may be good reasons for the different palette of questions each specification offers – but they are likely to be of a very recherché and exotic character probably uncongenial to the typical practitioner of objective testing who just wants to test his or her students.

But what are the motivation and status of these palettes? All the possible question types there could be? All that have been implemented so far? All that academics consider useful after a prolonged consultation? And what defines the differences? A slider question, which creates a number by the student moving a slider between maxima and minima doesn't really offer much over and above a numerical text entry question, other than perhaps constraining guesses from being utterly unreasonable. Many identical questions could be asked in more than one form. A multiple choice single answer question with two alternatives "true" or "false", could also be asked by offering a drop down list in a match exercise where the proposition would have to be matched by the words "true" or "false". If you were extremely silly you might ask the students to type in true or false into a box in a text entry question (though in language teaching you might, just, justify making students put words like vrai or faux into text boxes to answer questions).

At the 2002 CAA conference, Jane Peterson did indeed attempt the thankless task of trying to offer a more rigorous justification for the typology of questions. She made the point that when you tried to put the questions through different renderers, the stylistic differences that the questions seemed to encode, would not always be realised in the way the engine presented such questions to the user:

"JCloze is a name given to a fill in the blanks question type within Hot Potatoes (Half Baked Software, 2002). Another example is: drop down list in one engine can be identical to multiple choice or even select a phrase in others. Can select a phrase in fact be treated as a type when it could be presented as a multiple choice selection or as a hotspot? This is not a criticism of any particular engine but hopefully demonstrates the complex nature of any comparison between assessment engines. " (Paterson, 2002)

In an attempt to propose a more rigorous classification she wrote that question types should be classified along four axes.

- What does the question hope to assess? - **OUTCOME**
- What input is the student required to take? - **ACTION**
- In what manner is this best achieved? - **STYLE**
- How is the question best presented? – **FORMAT**

(Paterson,2002)

She broke down action into two categories, selection and data entry – corresponding essentially to mouse action and to keyboard action

- **SELECTION**: Multiple choice, Hotspot, Drag and drop, Graphing, Mazes
- **DATA ENTRY**: Text entry, Numeric, Algebraic, Gap fill, Crossword

(Paterson,2002)

This does in fact seem very sensible (and also makes one think about the questions of the future when further technological feats are possible – for instance we might ask questions with gaze detection as the action done by the user – such that we can test if users are *looking at the right thing!*)

However, in an attempt to get true rigour she makes the proposal that a naming convention along the lines of IP Addresses might be used. Such as:

Thus a common type may be logged as *SZZ-HSZ-STM-AEZ* meaning a question type where the answer is **S**electe**S** on a **H**ot**S**pot with **S**ingle graphic of **T**ext and **M**ultiple correct answers to test the outcome skill of **A**pplication at an **E**asy level. (Paterson,2002)

At this point we can probably see that beneficial though such a classification scheme might be to the expert, it is likely to infuriate and bewilder the humble academic. Nonetheless one has to marvel at the way this argument was so taken to its limits by Paterson, and thus demonstrate how difficult it is to justify any question typology from a theoretical point of view alone.

Therefore perhaps we need to look at this from a different angle. Rather, what question types do academics actually use? Certainly there is plenty of data about, from institutional repositories of questions, which can tell us what kinds of questions academics write.

The University of Hertfordshire has a Questionmark Perception licence and has had one since 1999. Owing perhaps to local support issues, it probably hasn't had as much use as installations in similar institutions, but nonetheless possesses approximately 2400 questions covering a number of subjects, but primarily computer science, business and mathematics. The authors of questions for this tended to be a fairly hardy and self-selecting group and so the disposition of questions will probably reflect a bias towards more confident users of technology.

There is a slight complication here in that in order to calculate how many questions of each type there are in the database, a DOM parser of QML had to be used. QML however has the very non-standard ability to allow unescaped angle brackets (< and >) within its text elements – which makes such questions virtually unreadable by a DOM parser. As a consequence 353 of the 2323 questions in the database were unable to be read. However, we don't think the absence of those questions, other than excluding particularly computer science coding questions, is likely to skew the overall proportions of the various question types in the database. But all that taken into account- this was the overall proportion of question types in the database:

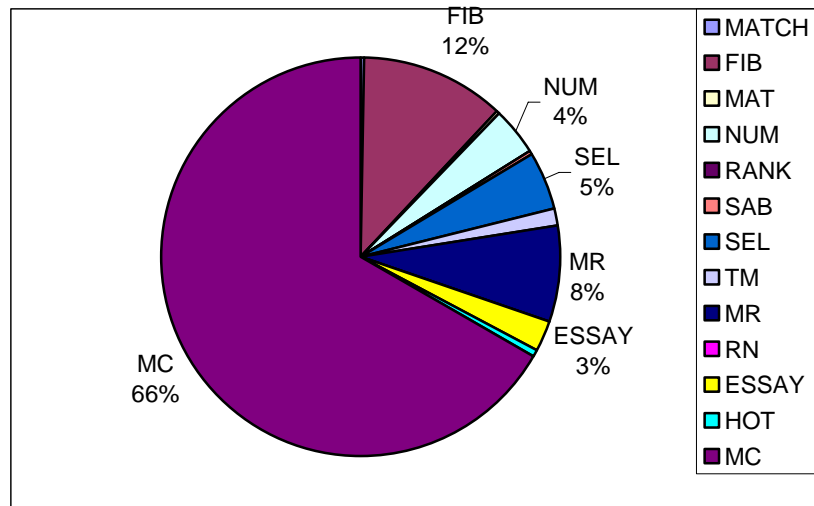


Figure 1: Proportions of Question Types in UH Database

Therefore almost exactly two thirds of all questions were multiple choice single answer questions. After that came *fill in the blanks* and *multiple response* questions. Numerical questions (which are really just fill in the blanks for numbers) and the SEL question (equivalent to the ORDER or MATCH question in QTI) also score in the four and the fives. The essay question we very rarely used and their number is probably artificially high here, in as much as they are used as much for surveys as for testing.

Nonetheless certain biases shine through. One user, a mathematician, has a much greater reliance on *fill-in-the-blanks* questions than many other users. Here are his questions broken down:

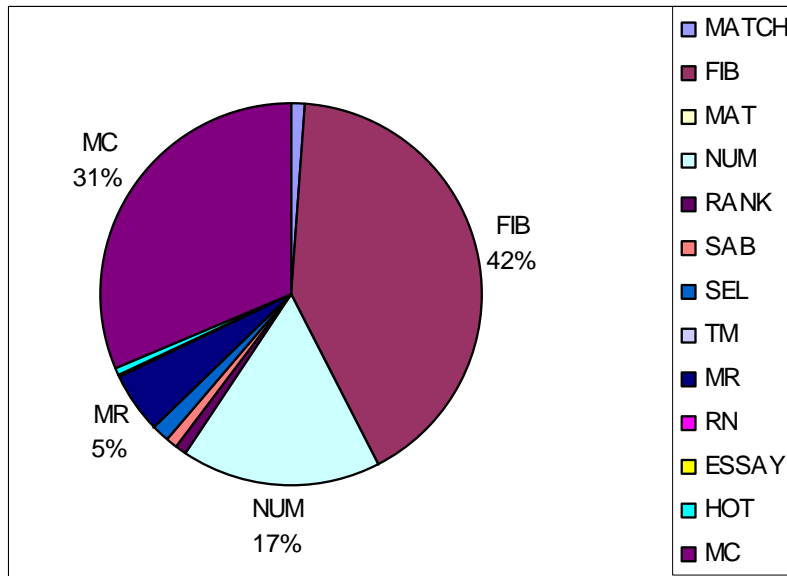


Figure 2: Proportion of Question Types for a Mathematician

This might be a very good reflection on the the culture of mathematics that objective testing so concentrates on getting the students to input real values into boxes, rather than recognize truth among distractions. This was out of 369 questions.

Finally looking at my own questions, we see the following distribution of questions:

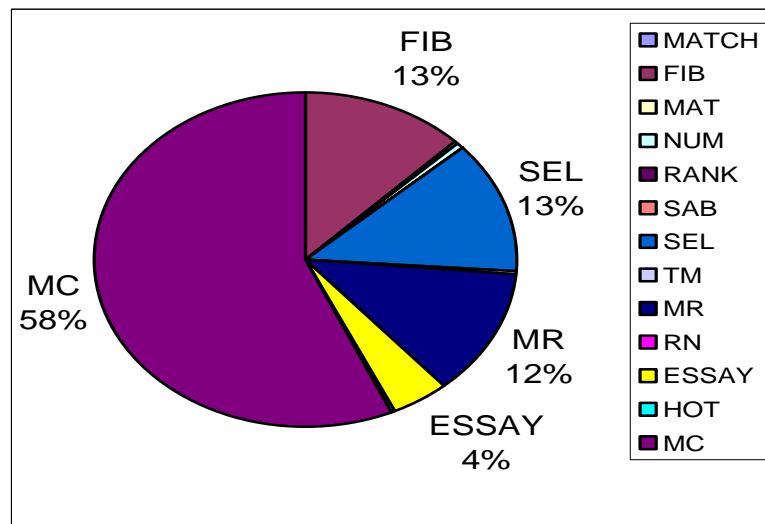


Figure 3: Proportion of Question Types for me!

This was out of 619 questions – though a substantial proportion of them are likely to be duplicates of other questions, since the interface to the system at times makes it easier to re-upload a whole sequence of questions from a test, when you want to just edit 4 or 5 of those questions. However I don't believe this will skew the overall proportions. What accounts for my own preferences (and the large amount of SEL or order questions) is that for the last 7 or 8 years I have used a tool for writing such questions using a text editor, which,

made more robust and with more format translation capabilities, became MCQFM.

MCQFM is a tool coming from a JISC funded project which essentially is a method of writing questions without modal editors. It can be found at <http://smirkboard.herts.ac.uk:8080/mcqfm> it uses an extremely simple notation of headers, newlines and @ symbols to delineate the parts of a question such that it remains humanly readable. Here is what its interface looks like:

The interface is divided into two main sections. On the left, a large text area is titled "Enter Questions Below". On the right, a sidebar titled "Click below to insert sample questions" contains several buttons: "Multiple Choice Question", "Multiple Response Question", "Cloze Question", "Order Question", and "Match Question". Below these buttons, a section titled "Then select the quiz type from the dropdown menu below and press submit" contains a dropdown menu and a "Submit" button. The dropdown menu is currently open, showing the following options: "Produce HTML Quiz!", "Produce QT1 zip for R2Q2!", "Produce QT1 Jorum zip!", "Produce QT1 quiz!", and "Produce QML Quiz!".

Figure 4: The MCQFM Authoring Interface

The system allows users to enter text into a text box, and then output the results either as a simple html+javascript page – ideal for formative assessment since it will work on any webserver without requiring any server programming. Or it could be a zip-of-zips ideal for the old r2q2 server, a Jorum style zip (which also works in ASDEL), a stream of xml containing QTI items, or a stream of QML. There also exists another page where this can be done in reverse. Questions encoded in QML can be transformed back into simple text, or converted into simple html or turned into QTI. The simple text itself looks as follows:

MCQ

What is the capital of China?|Beijing is "northern" (bei) "capital" jing

Shanghai
@Beijing
Kunming

CLOZE

Put in the correct answer here|Ni hao

The Mandarin for hello is @ni@ @hao@

MCQ

Which of these territories belong to the People's Republic of China?Only Taiwan is outside the PRC

Taiwan
@Mainland China
@Hong Kong
@Macau

ORDER

Where do they speak what?|Its mandarin in the two main cities

Beijing@Mandarin
Hong Kong@Cantonese

ORDER

Put these chinese cities in order by number of inhabitants|Shanghai is not the capital is the commercial hub and has 20 million inhabitants

Shanghai
Beijing
Xi'an

The major design principle was that all questions should be readable at a glance, and be capable of authoring using only a text editor. Because of this restriction, it meant that a number of the question types we have seen before were not capable of being authored in MCQFM – but, as we have seen from the data above, this might be a very minor impediment. If the University's of

Hertfordshire's data is anything to go by, it would be possible to author 91% of the institution's questions in MCQFM – and if you subtract “essay” type questions from that number, which were used much more in surveys than in tests, then it reaches approximately 95%. Moreover, such a system is likely to significantly enhance the percentage of questions which are difficult to author in conventional authoring tools (such as order questions) – since the question template makes that so easy.

Having such a radical simplification to the palette of possible questions generates some gains and some losses.

Losses	Gains
<ul style="list-style-type: none"> – Control over the ordering of initial states of ordering/matching questions – Control over what allowances to make for misspellings or capitalisations in cloze questions – Control over scoring <ul style="list-style-type: none"> – <i>(However, this is a mixed blessing. Specifying the scoring of a question actually limits its potential for reuse. Therefore forcing a default score for a question makes it level with other questions allowing for easier incorporation into other tests.)</i> – No metadata fields 	<ul style="list-style-type: none"> – Immediate legibility of the question – I do not have to take it to evaluate it – I do not need a specialised editor to write it – No metadata fields! <ul style="list-style-type: none"> – In practice what seriously do I need to know about a question in order to use it? Just what it says probably, probably its subject, and maybe who wrote it (so I can find other questions by him/her if I like this question) and what level class (year 1/2/3) they wrote it for

Is this therefore an argument for junking the majority of the QTI 2.1 standard? For the purposes of sharing, probably yes. There are many question types in that standard which are just not worth implementing on an economic level. What is the point in training people to write questions, or writing assessment engines to render questions, when there will be so few of those question types used? Or at least the proportion of these in the overall listings of questions so miniscule?

A counterargument to this is that the low proportion of say *hot-spot* questions is not down to the innate difficulty of authoring this type of question, but rather the lack of training and support that academics have such that they are unwilling to experiment with any but the most ordinary of question types. This has some merit, but I believe my earlier argument still holds. Namely that the inherent problem with a question of this type is that *it is very difficult to present it in a way both readable to the human and the computer, and writable without helper applications*. If I wish to get a question of this type moderated in time for a test, I will have to generate the graphic, with the hotspot area superimposed over it and printed on paper. Also a question of this type is not

really browse-able (I can't cast a glance over it easily and decide if I want to use that question in my test – I have to really, *take* it).

Therefore, I would argue, for the goal of sharing, the more exotic end of the question typology is not all that helpful. Conversely, however, the scoring and branching capabilities of the QTI assessment standard are *extremely* useful. While sophistication is not important for question authoring, it is important for deployment and testing. There will be all kinds of tweaks that we will need to do in the concrete reality of courses and classrooms.

Principal among these is scoring. The score of a question only has meaning in the context of other questions. And also there might be departmental policies (negative marking *can't* be used, or even negative marking *must* be used!). There might be biases arising from how risk averse the students' area – and therefore the scoring might need to be further tweaked. Then there are questions of branching.

Here is a personal example. I teach a course called *Mobile Standards Interfaces and Applications*. It is taught primarily via student led seminars which were in essence illustrated summaries of research papers. To ensure the students have attended the classes and taken in what was expressed in the classes, I run an objective test at the end of the course covering all the seminars the students gave – meaning they were quizzed on 12 research papers – which went extremely well. In the second iteration however, I had 28 students. In this occasion it seemed too much to ask the students to revise 28 papers. I therefore gave them 7 blocks of 10 questions – each representing a particular topic – and containing questions on about 4 papers each. The students then took the test with their scores from their highest 3 representing their score for this assignment.

It is these highly practical, highly context dependent deployments of online testing which will lead to all manner of differing practice. One of my colleagues in Pharmacology, Dale Peterson gives test containing entirely multiple response questions of five true/false propositions each and scores as follows:

- 0 for 5 wrong or unattempted
- 0 for 4 wrong or unattempted
- 0 for 3 wrong or unattempted
- 1 for 2 wrong or unattempted
- 3 for 1 wrong or unattempted
- 5 for all right

This method (as opposed to simple negative marking of incorrect alternatives chosen) corresponded to very precise issues of comparability with exams, departmental and subject specific values, and the desire not to over-penalize a student for non attempting some questions.

An assessment engine which cannot deal with complexities of deployment as seen in the examples above, is not likely to get very far. But equally, a sharing engine which contains detail like this within its repository, is likely to put off rather than engage other academics.

At this point we might wonder what the characteristics affordances of a much simpler question authoring system, and a fully web 2.0 question discovery and sharing system, might be?

At the level of creation it will mean easier authoring – and moreover, easier transportation to other authors who might wish to see it. This is not a trivial thing. If I want true peer assessment of my questions – the most likely way to do it is in the body of an email. If I have to open a specific tool to see the questions then this is likely to hinder peer comments.

Leading on from this there is moderation. Questions which are readable on paper will be much easier to *evaluate*, *amend* and *annotate* and ultimately *moderate*. However, at-a-glance visualisation leads to much more important gains – namely the creation of more tangible and direct manipulation user interfaces where questions may be dragged around a semantic space in order to group them in palpable way.

A small experiment we did after MCQFM was to write a sequence of 16 questions covering 4 questions each in the fields of football, tennis, cricket and chess. These were in QML. We then developed an interface where these questions appearing in a semantic space could be moved around that space and then deposited into boxes where the questions could be differentiated. To render the questions in more simple text only format, the MCQFM web service was used. Once the topics had been fully differentiated, they could then be exported back out as html, qml or qti files, topic by topic – again using the MCQFM web service.

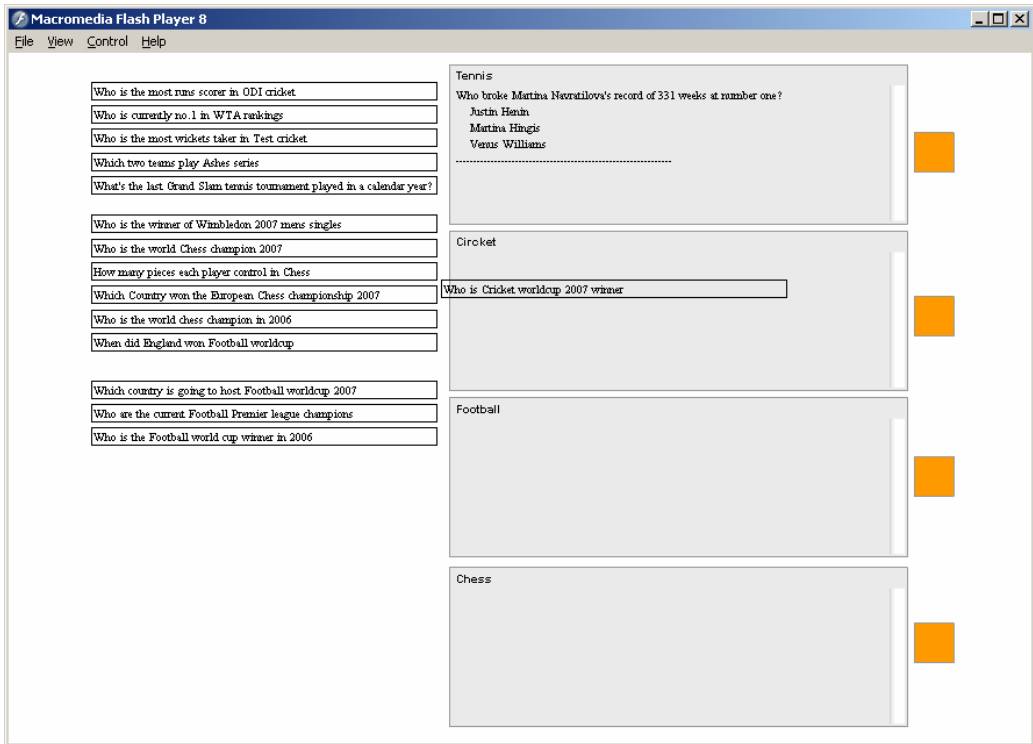


Figure 6: The initial screen where users drag questions into topic boxes

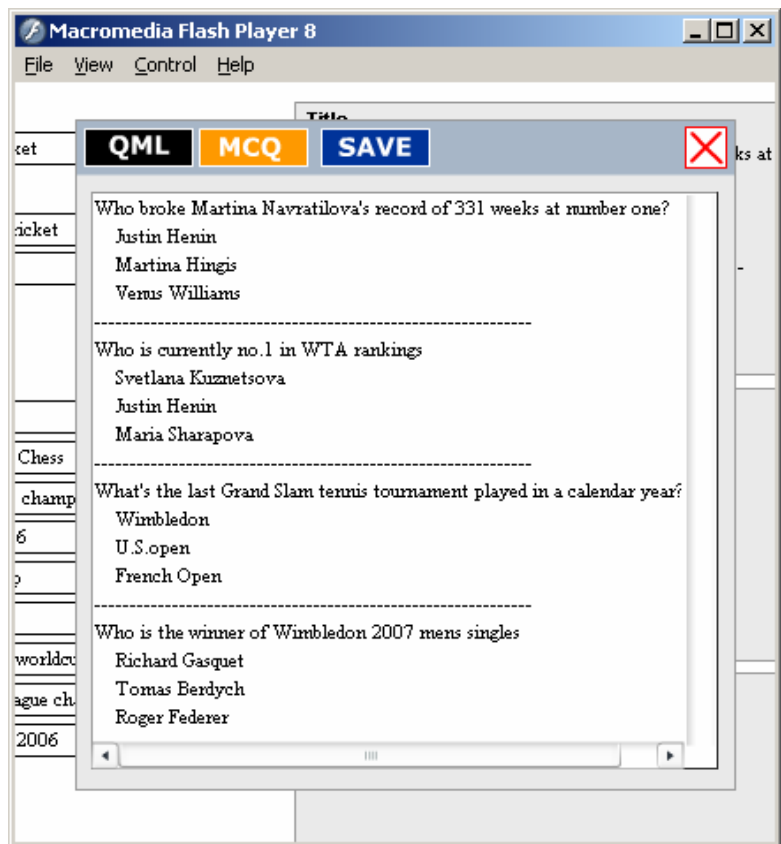


Figure 7: Prototype Topic Export Facility

This would contrast with the highly modal interfaces available in tools like Questionmark Perception. In version 3 at least of the client software, any

restructuring of banks of questions into new topics and hierarchies requires the use of a tree metaphor, and the copying and pasting questions (visualized through a sub string of the question descriptor) following the folder metaphors of *Outlook* or *Windows Explorer*.

What might lead on from this is vastly more easily visualized subject domains where questions might be able to be moved to specific points, potentially superimposed on images such as mind maps or even UML diagrams. Question similarity through contiguity will thereby become another at-a-glance organizing principle. Such visualisation techniques are common at Digg:



Figure 8: The Digg Swarm Interface

Another example is Grokker: have a look at its visualisation of a search for "VLE" (yielding up Yahoo! And Wikipedia results)

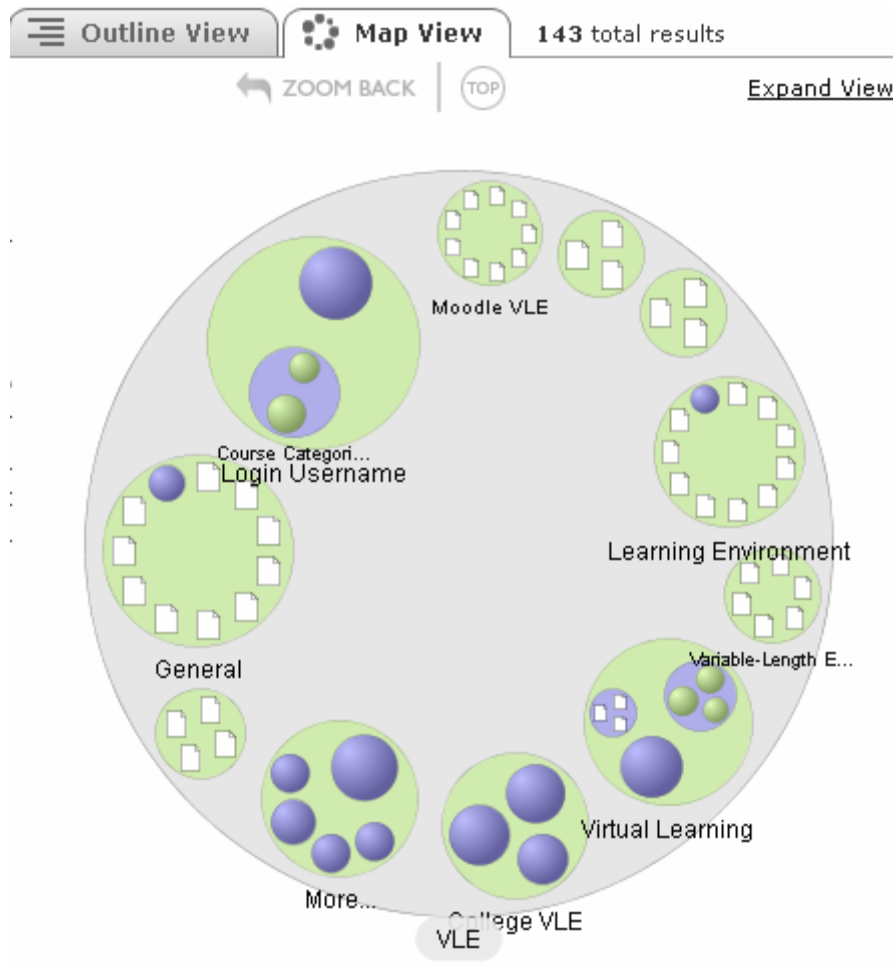


Figure 9: The Grokker Map Interface

It seems to me apparent that these kinds of interfaces, which allow for a much more agile practice of question bank organisation, need to be in place to really facilitate discovery of elsewhere authored materials.

The final hindrance might be said to be a Web 1.0 way of thinking. In this paradigm, the quality assurance of shared materials is established through procedures of peer-evaluation. Authors are recruited, templates handed out, questions submitted, ticked for approval by their peers then submitted to the database. However, as we have seen from web 2.0 initiatives like Wikipedia, quality is not to be achieved through procedural control, but is an *emergent* quality of the system. As Eric Raymond's puts it (explaining the success of Open Source projects like the Linux operating system), "given enough eyeballs, all bugs are shallow". In that paradigm, authoring is not about going through complex approval processes, but rather through the practice of "release early, release often" – the imperfect material is submitted to the system, and through the voluntary contributions of its community, the flaws ironed out, the bugs corrected, the typos transcended. (Raymond 1999)

Concluding, it is probably still worthwhile to think of ways to establish large item banks of questions to be shared between institutions in higher education. Anyone who has ever incorporated Open Course Ware material into their courses (as I have) can testify to how pleasing that is, both to use – and also

be able to edit and repurpose – materials produced to a high standard in an institution like your own. The same should be true for question and assessment creation. My hypothesis to try and achieve this in practice would be:

- Ditch the QTI and QML standards for question sharing. Just use a plain text subset of the 5 most popular question types (you'll thereby cover 95% of all the questions that are ever likely to be written)
- Save them in databases with simple metadata such as author, subject and a few tags. Don't bother with anything beyond that.
- Allow them to be visualised in creative ways as exemplified in tools like the Digg Swarm tool, or the Grokker mapping tool
- Allow users to share their own creative organisations of questions just as del.icio.us users can share their ontologies of bookmarks
- (This is the hardest part) – allow true test engines such as Perception and ADSEL to annotate questions with difficulty scores which come from their usage in tests.

Of course I cannot guarantee that even implementing these things will result in a big question sharing culture to come about, but it might break down at least some of the barriers encountered so far.

References

Jane S Peterson. (2002). What's in a Name? - A New Hierarchy for Question Types. In: Danson M *CAA Conference Proceedings*. Loughborough: University of Loughborough. 285-292. See also: http://www.caaconference.com/pastConferences/2002/proceedings/paterson_j1.pdf

QuestionMark. (1995-2008). *Question Types*. Available: http://www.questionmark.com/us/perception/authoring_windows_qm_qtypes.aspx. Last accessed 07 March 2008 For QTI Question Types see:

IMS Consortium. (2008). *IMS Question & Test Interoperability Specification*. Available: <http://www.imsglobal.org/question/>. Last accessed 07 March 2008.

Digg Labs. (2004-2008). *Digg Labs*. Available: <http://labs.digg.com/swarm/>. Last accessed 07 March 2008. Grokker Interface

Grokker. (2006). *Grokker Enterprise Search Management*. Available: <http://live.grokker.com/>. Last accessed 07 March 2008.

Eric S. Raymond (1999). [*The Cathedral & the Bazaar*](#). O'Reilly. ISBN 1-56592-724-9. see also: <http://www.catb.org/esr/writings/cathedral-bazaar/cathedral-bazaar/ar01s04.html>