QuestionBuddy – A collaborative question search and play portal

Will M Davies, Hugh C Davis

QuestionBuddy – A Collaborative Question Search And Play Portal

Will M Davies, Hugh C Davis

Learning Technologies Group, ECS, University of Southampton, Southampton, UK, wmd04r|hcd@ecs.soton.ac.uk

Abstract

Generally itembanks are inaccessible to students. Current use of itembanks focus on the teacher as having responsibility to organise questions (place them in pools, associate them with course content) and make them available/deliver them to students. This limits students to the teachers perspective and to the questions that the teacher has made available. As the practice of itembanking increases it may be appropriate to encourage students to use questions from pools not directly prepared by their teacher. A mechanism for searching across itembanks and sharing recommendations with peers would be of help in facilitating this. We describe QuestionBuddy, a collaborative filter based question portal for students, built to study student usage of, and attitudes to, such a system.

Introduction

We introduce QuestionBuddy, our self assessment website for students of electronic and electrical engineering. The site allows students to search for questions from the (E3AN) itembank and gives feedback to their answers. Having attempted a question the student is then asked to rate it, on relevance to their current study. By comparing a student's rating profile with those of other users, recommendations for further study questions can be made. This is done by selecting additional items rated highly by users with a similar rating profile.

The reason for this work is to investigate ways of enabling users to find itembank content for their needs. It is assumed that searching across the item metadata alone will not always be able to offer a complete solution to satisfy users' search requirements. Factors contributing to this include varied granularity of metadata and possibly the users incomplete knowledge of the domain they are searching. It is intended that that this work will be able to contribute in the area of engaging students and assisting them in seeking feedback. The QuestionBuddy self-assessment process aims to help students to making informed choices when directing their study efforts. By using the system regularly, students will be able to get timely feedback on the effectiveness of their study. It is anticipated that lessons learnt from QuestionBuddy will be applicable to other sets of itembank users, such as teachers, that compile assessments. A call for an improvement in user/itembank interfaces can be seen in the Itembanks Infrastructure Study [IBIS], (Cross 2004).

QuestionBuddy has been built using the content from the E3AN itembank in combination with the APIS rendering engine, available at (APIS), for questions in IMS Question and Test Interoperability [QTI] format. To complement these, a custom webservice search interface has been added to E3AN and a collaborative filter has been constructed to make item recommendations to users of the site. To enable the APIS service to handle question rendering and response processing the original E3AN questions were converted to QTIv2 using the (PyAssess) conversion tool.

The Problem

Hidden Content

As the size and availability of learning object repositories and itembanks increase teachers are about to be swamped by yet another source of learning resources. Developing tools and techniques for finding and managing these resources is crucial. (Anderson, Ball et al. 2003) discuss the issues raised in searching for ever smaller learning objects with increasingly fine grained descriptions. (Lemire, Boley et al. 2005) identify problems in searching for learning objects over subjective metadata such as the IEEE Learning Object Metadata classification 'semantic density'.

Search systems in existing itembanks rely on author/librarian created metadata, pools of questions created by teachers and crude plain text searches. To return questions appropriate to the student's needs these techniques require significant input from the author/librarian/teacher in classifying the questions. When a question is used outside the context which it was created for it is likely that its description will need to be reconsidered. This is a significant problem for an itembank that is intended to be shared among a large number of institutions. It seems reasonable to consider search and retrieval issues relating to a single objective question as similar to those associated with a small learning object.

A Possible Solution

It is possible to gain knowledge about an item from its previous usage. In traditional models of summative assessment this usually means recording student scores and carrying out analysis of these scores. This can be used to identify questions that unfairly discriminate against certain students and also to identify discrepancies between the taught curriculum and the subject assessed. Having identified unfair questions it is then possible to remove them from future use. This analysis relies on results from a significant number of students. Rather than asking one expert whether, in their opinion, a question is biased, statistics make it possible to examine the results of a large number of students.

Extending this analysis to a formative assessment environment used by students from multiple institutions, with varying curricula, at different stages of their courses, appears fraught with statistical problems. The need for investigation in this area is stated in the IBIS report by (McAlpine and Cross 2004).

"As the analysis of student data is generally for summative purposes, a closer look must be taken at this use to facilitate formative use and empower students and their learning. Some of the key ways that this can be done is through helping students to make the correct choices in their learning by providing them with data which can assist them become more responsive and self-aware learners"

Collaborative filtering provides a way of making comparisons between similar users. In its simplest terms collaborative filtering ignores the all properties of an item except for the identities of the users that have interacted with it. The commonality between two items is measured by the intersection of the sets of people that have used them. For a class of 100 students on a course, they are no longer 100 individuals struggling in a sea of questions to find revision material, they can be empowered with the results of each other's efforts.Rather than relying purely on the use of protected term classifications and placing the sole burden of describing a question accurately with the author/librarian, we aim to augment an item's description with some notion of the context in which it is used.

Collaborative Filtering Overview

Successful collaborative filter systems include those used at (Amazon.com; Last.fm; MovieLens). Simplistically, a collaborative filter works by comparing two user's ratings of some material and calculating the similarity or distance between these users. If two users have a high degree of similarity then it is assumed that they will appreciate recommendations of items that they have not rated, but have been rated highly by the other. For an in depth review of collaborative filtering systems and techniques see (Adomavicius and Tuzhilin 2005), many of the systems they discuss take a hybrid approach of combining collaborative and content-based recommendations.

One collaborative filter that is likely to be familiar to many people is that used by Amazon.com. By making comparisons between different users' purchase and rating profiles Amazon is able to suggest items for purchase. The usefulness of these recommendations varies, one reason for this is that the system does not record the context in which a purchase is made. A good example of this is that by buying gifts for several very different people a customer can end up getting recommendations from several conflicting stereotypes. Amazon has recognised this and now includes a link with each item, 'why was this recommended to me' that allows users to remove items from their rating profile. (Linden, Smith et al. 2003) describes the specific filtering algorithm used by Amazon.

Collaborative filters are well established technology but they have not, until now, been used for question material. In the education domain, (Downes, Fournier et al. 2004) discuss Sifter, an experimental learning object recommender developed in Canada. The filtering system behind this, RACOFI, is now being used to power (InDiscover) a music recommender system. Sifter asked users to rate content along up to 15 dimensions including 'level of interaction' to 'ability to motivate'. One of RACOFI's strengths is its ability to filter efficiently over a large number of dimensions. The intended Sifter user group were developers responsible for assembling learning objects to build coherent courses.

There is a trade-off when creating a collaborative filter, between obtaining a sufficiently rich user model, and overloading the user by asking them too many questions about themselves and the content they are using, (Swearingen and Sinha 2002). In the context of Sifter, asking developers using a learning object to apply ratings in 15 dimensions seems acceptable. Asking for a similar level of detail from a student taking a two minute multiple choice question creates a burden that the student is unlikely to tolerate.

QuestionBuddy – The User Experience

Students are expected to come to QuestionBuddy having already studied a subject but wanting to confirm their understanding. After logging in they are presented with a personal summary page, shown in Figure 1, this presents some recommended questions. These recommendations are created both by analysing previous subject interests and also by the collaborative filter.

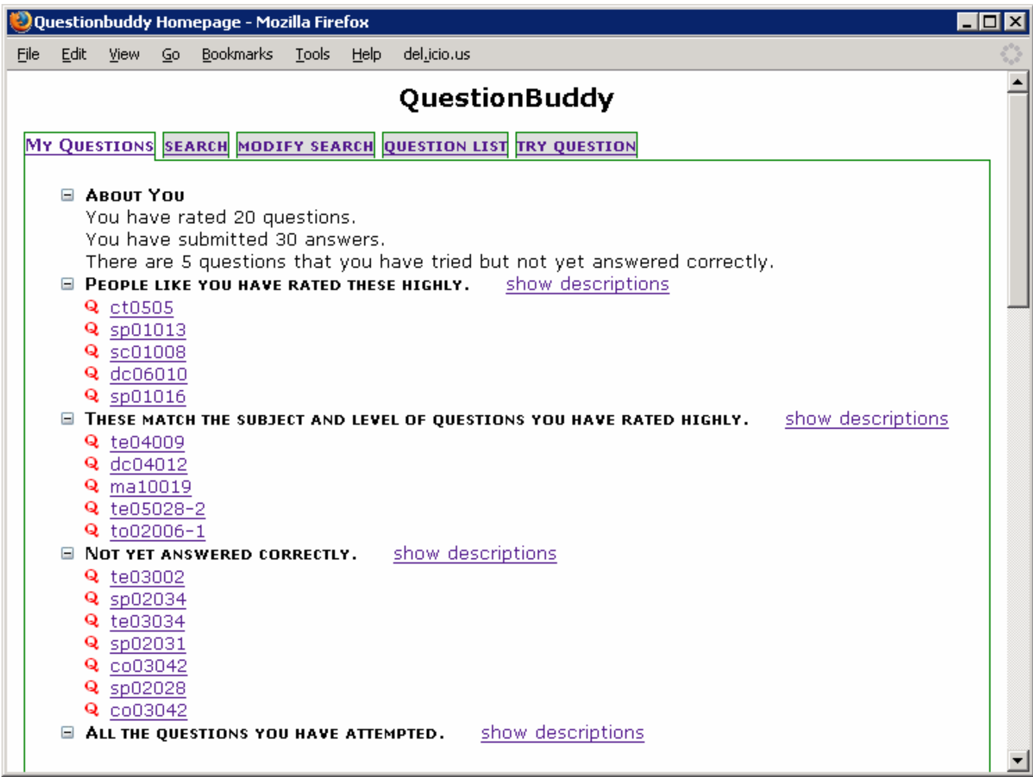

Figure 1

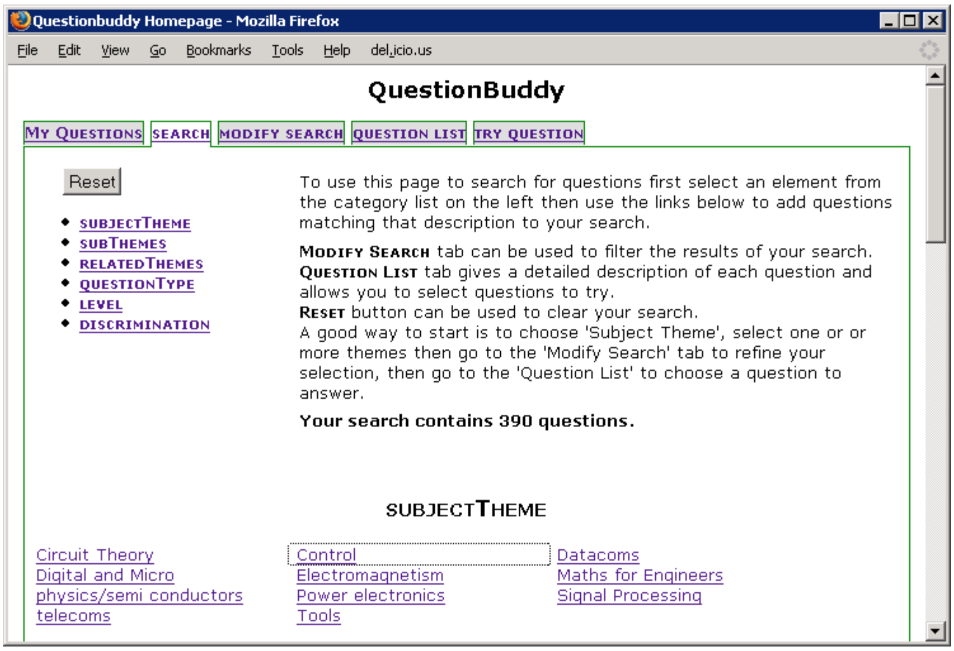

Figure 2

For a student new to the system this page will not be able to make recommendations, they will need to use the search page, Figure 2. The *search* page presents closed lists of categories from which users can select the questions that interest them. The number of hits in their search is updated and displayed as the scope of the search is increased.

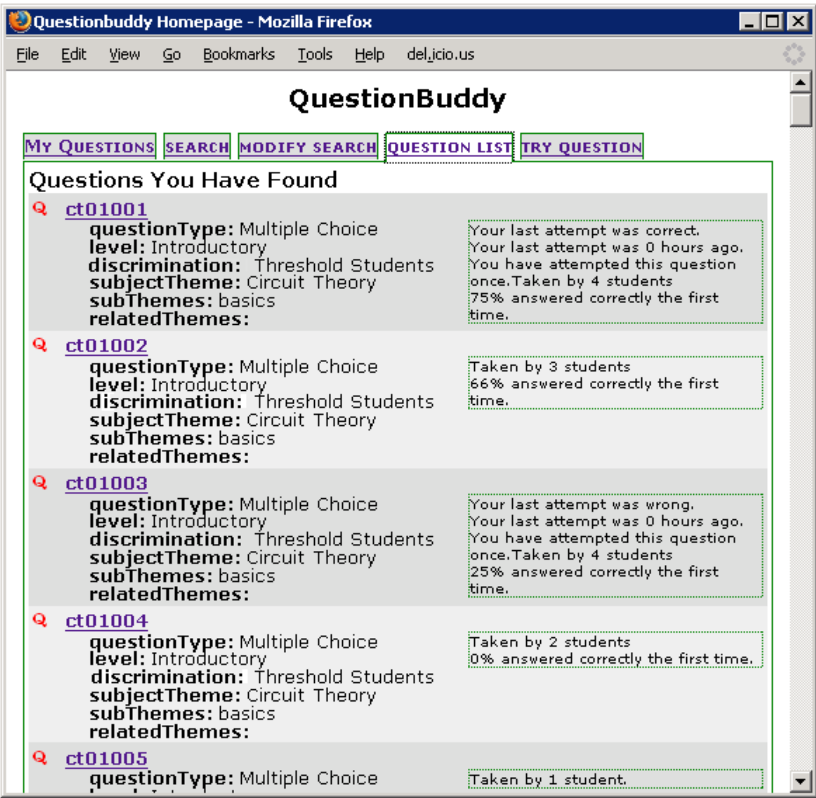

Figure 3

If desired the *modify search*, not shown, page can then be used to filter the search for example by only including questions that are multiple choice. The student may also choose to restrict the difficulty, discrimination or sub-theme of the results to reduce the number of returns to a manageable number.

Navigating to the *question list* page, Figure 3, displays the results of the students search. Each item is described using the available metadata and also some statistics concerning its previous use. This description is one of the areas of the system that needs further investigation. Important design questions are, what, of the information presented, is useful, and, what other information could be shown to help users.

Selecting a question from the list takes the student to the *try question* page, Figure 4, where the question is displayed and the student can submit an answer. If the question type chosen is supported by the QTI renderer it will examine the students answer and give them feedback. The ratings panel is provided for students to rate the question for relevance. They are required to submit a rating before the system will allow them to navigate away from this page. At present the question answer process requires users to navigate back and forth between the *question list* and *try question* pages, it is recognised that this impacts on the usability of the system. Consideration is being given to allowing each of the questions in the *question list* to be displayed inline without forcing users to navigate between panes.

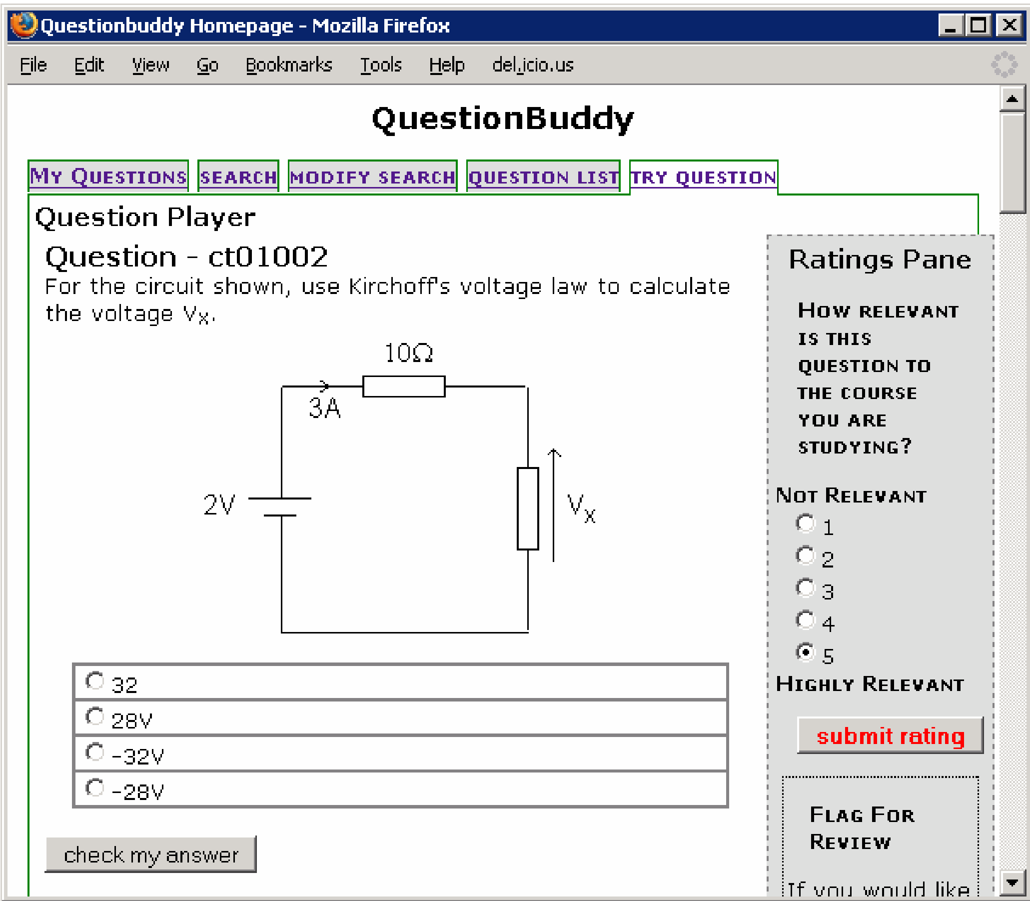

Figure 4

System Architecture

QuestionBuddy is implemented by aggregating several webservices. These services are: the APIS QTIv2 renderer and home grown services for maintaining user profiles and itembank searching.

Itembank Search Service

The search service is implemented on top of the E3AN itembank of electronic and electrical engineering questions. The interface to the search service provides four methods:

- getSearchTerms()
- getSearchTermValues(String searchTerm)
- search(String query)
- completeTentativeSearch(String searchIdentifier)

In addition four objects SearchResult, SearchTerm, SearchTermValue and Item are required by the interface. The service is designed to return lists of searchable terms rather than expecting users to guess how the content has been categorised. Whilst this adds extra complexity to the user interface, it should simplify the construction of sophisticated queries. The search service protocol places no restriction on the way questions are categorised so it would be possible to aggregate results from several itembanks if this is desired. The protocol has been kept deliberately simple to allow compliant services to be created for existing itembank systems. A version of this webservice search interface has also been implemented for the TOIA itembank. No changes were necessary to the client to allow this interface to work successfully.

QTI Rendering and Response

The APIS rendering and response service as downloaded from sourceforge required a small number of changes to the code to generate correct XML and to handle the QTI expression *match*. We look forward to integrating the R2Q2 QTI webservice renderer that is being funded under JISC toolkit development. This should extend the range of questions types that QuestionBuddy is able to play.

User Profiling and Collaborative Filter

The user profile service was developed independently from the itembank service. This was done to ensure that any developments made to the service were independent of the itembank used by the system. This service will work successfully with multiple itembanks providing the item identifiers are unique throughout the system.

Lessons Learnt

Trying to create meaningful descriptions of items to display in a list for students is not easy. A similar problem would be asking someone with no knowledge of science fiction to choose a science fiction book as a gift. With little knowledge of the sub-classification of the genre much of the information they could be shown about the book will be meaningless. The solution chosen for QuestionBuddy works best when users understand the specific educational language used in the metadata. This display is augmented with statistics of previous question usage. The collaborative filter should help to compensate for less than ideal question descriptions by ensuring that a greater proportion of the questions offered are relevant.

The system contains more information about each question than it is useful to present to the student when helping them to choose questions to attempt. In part this is caused by the specific/specialist nature of some of the metadata. For example, E3AN contains a description for cognitive level, indicating what level of skills the question assesses. This information is likely to be helpful to a teacher compiling an assessment but is probably not helpful for the target student audience. As a result of this more than 50% of the data about a question provided by the search service was discarded.

The decoupling of the search parameters from the user interface complicated the user interface design. This feature is important to enable the interface to work with different itembanks. Knowing how many categories existed and how many possible values they could take, would allow for a more intuitive interface design.

It is possible to calculate an average rating for each item, but as the rating depends on the context of the student this would ignore the fact that different students will be studying different, if subtly, courses. As a consequence a definite decision was taken not to display the average rating of an item.

Future Work

Once the system is in regular use it will be possible to look for trends in rating. It may be possible to use these to learn more about the content and to tune the recommendation system. One way of doing this is by analysing item ratings in conjunction with the search criteria used to find them. For example, if users searching for questions on ‘circuit theory’ always rate question X 1/5, either this question is not about circuit theory or, it is simply not a very useful question.

After calculating the discrimination of each question it may become desirable to filter out questions with low discrimination. This would ideally allow students to take fewer questions to get an accurate assessment of their ability. Because of the formative nature of the system, this is problematic as hopefully the students’ ability is improving from one session to the next.

In a much the same way as Amazon allows customers to remove certain purchases from their recommendation profile, it might be helpful to allow students to specify constraints on their recommendations. For example a student that has taken 90% of the questions on electromagnetism is likely to get recommendations for the other 10%. The student may feel they have studied this area sufficiently and wish to exclude these questions. It should be possible to make this decision automatically by examining the students performance in previous questions. This is related to a more general issue that the recommendation system should be transparent. The system should be capable of displaying to the user how their recommendations are generated and wherever feasible they should be able to adjust the parameters controlling what is offered to them.

Social bookmarking, the act of creating personal tags for collections of resources is currently a popular way of allowing users to describe things for their own and others use. For a good introduction to tagging see (Wikipedia). Allowing users to create a folksonomy of an itembank may create harvestable information about the question held. In conventional itembanking terminology, this is very similar to pool creation. The ability to create pools and share pool identifiers with other users would support other use cases for the system. This type of feature needs to be examined carefully as malicious users could create deliberately disparate collections that might poison the system for others.

Conclusion

QuestionBuddy is ready to offer students a novel way of self-testing. By analysing the use of the system in combination with the existing item metadata we anticipate being able to augment the user experience. It will be possible to utilise the usage data recorded about each item to increase the value of the item in the future.

References

Adomavicius, G. and A. Tuzhilin (2005). "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions." Knowledge and Data Engineering, IEEE Transactions on 17(6): 749.

Amazon.com, http://www.amazon.com, last accessed: 01/04/2006.

Anderson, M., M. Ball, et al. (2003). RACOFI: A Rule-Applying Collaborative Filtering System. COLA'03, Halifax, Canada, IEEE WIC.

APIS, http://apis.sourceforge.net/, last accessed: 01/04/2006.

Cross, R. (2004). Item Banks Infrastructure Study. N. Sclater, JISC: 87-100.

Downes, S., H. Fournier, et al. (2004). Projecting Quality. MADLaT, Winnipeg,Manitoba.

E3AN, http://www.e3an.ac.uk/, last accessed: 01/05/2006.

InDiscover, http://www.indiscover.net, last accessed: 01/04/2006.

Last.fm, http://www.last.fm, last accessed: 1/04/2006.

Lemire, D., H. Boley, et al. (2005). "Collaborative Filtering and Inference Rules for Context-Aware Learning Object Recommendation." International Journal of Interactive Technology and Smart Education, 2(3).

Linden, G., B. Smith, et al. (2003). "Amazon.com recommendations: item-to-item collaborative filtering." IEEE Internet Computing 7(1): 76-80.

McAlpine, M. and R. Cross (2004). Item Banks Infrastructure Study. N. Sclater, JISC: 77-85.

MovieLens, http://movielens.umn.edu/main, last accessed: 01/04/2006.

PyAssess, http://pyassess.ucles.cam.ac.uk/, last accessed: 01/05/2006.

Swearingen, K. and R. Sinha (2002). Interaction Design for Recommender Systems. Designing Interactive Systems (DIS2002), London, ACM.

Wikipedia, http://en.wikipedia.org/wiki/Folksonomy, last accessed: 01/04/2006.