Group Report

Company India

Web 2.0 Research Gateway

Cheryl Hung, Alex Bradbury, Tom Coxon, Edwin Li, Enzhe Zhang

Timeline

talks.cam 2.0 started out with an ambitious brief, with a large number of varied functions to implement.

After the first project meeting, the first features to be implemented were the extraction of talks.cam data and user profiles. By the second meeting, we had the basics of a usable site, including user authentication (signing up and logging in), viewable user profiles, email reminders for talks and other web 2.0 essentials such as commenting, tagging and rating.

The remainder of the project time was dedicated to privacy (such as friending and restricting access to private data), administrative functions (feedback reports and comment/tag deletion), and messaging within the system. After hooking in the website with the back end, testing was the focus and ensuring that the project was well documented.

Successes and Failures

Almost all of the essential features of the site were well implemented. In particular:

- The XML importer successfully pulled data from over 5000 talks and nearly 500 lists, with a total of 12 errors.
- User authentication is available for both Raven log in, and external users. Validation in the form of a link sent in an email are used to verify non-Raven users, and to reset forgotten passwords.
- Cookies are used to create a session based login system, and "remember me" is also supported.
- The email reminder system uses Googlemail's SMTP server to send reminders if requested.
- Editing user profiles and some administrative functions are implemented using AJAX, so the whole site feels faster and more web 2.0. Non-Javascript methods are also provided for users who may be running older browsers, or have it turned off.
- Other web 2.0 features such as tagging, rating and commenting are integrated into the site.

There were a few implementation details and features that were eventually omitted. These are:

- Mysql migration. This was suggested in order to allow more scalability and extensibility. However using Apache Derby (a pure Java database) in development has been perfectly satisfactory and if mysql migration is required in the future, the syntax is sufficiently similar to make this relatively simple.
- Microformats. This way of attaching semantic meaning to HTML has not yet become widespread, so it is unlikely to be useful.
- Google calendar. Although a calendar would increase aesthetics, providing a listing of talks instead gives the same information.
- RSS. This was to aid people in returning to the site; however, it was decided that this would be covered by email reminders.
- Image upload and resize. Instead of dealing with the security implications of uploading, it was decided that allowing external urls would be easier and providing a uploader would be delayed until talks.cam 3.0.
- User recommendations. There are tens of metrics which we could have defined this by, and in the end, we decided that providing a simple list of talks that "People who went to this talk also went to" would be sufficient in encouraging users to explore new talks.

Division of labour

As originally intended, Tom and Alex implemented almost all of the database access and Java backend code. In some cases, they also integrated their own code into the front end for testing purposes.

Edwin has done some integration work, and also implemented some front end features - for example, the date picker. Enzhe designed the visual themes, and wrote the static html/css code very early on.

Cheryl has written most of the documentation, including the specification, user documentation, progress report etc. She has also done system integration testing and fixed minor bugs in code.

Lessons learnt

In terms of project management, the most important lessons learnt from the project were:

- It is of the utmost importance to agree on a good specification. Our documentation was constantly refined and used as a reference point, particularly for testing.
- On the other hand, the specification is likely to change as the project becomes more refined, and this is a vital part of the project. For example, the original Java interfaces were modified to provide better separation of a single User and a UserService.
- Having a centralised management place makes collaboration much easier (in our case, a group wiki and bug tracker).
- Researching frameworks carefully is much more productive than reinventing the wheel - we used Struts and Maven among others to simplify development and deployment.
- A detailed roadmap is the best way to ensure that the project does not fall excessively behind, and there is no forgotten features which need to be implemented at the last moment.
- It is good to be ambitious, but having slack in the schedule relieves unnecessary pressure.
- It is important to assess what the essential features are, and to consider the value to customers. For example, mysql was considered an added bonus, and as an implementation detail, would not even impact on the customers.

Future extensions

Ideally integration into the original talks.cam database would be highly beneficial, as this would also allow us to provide adding and editing of talks and lists. We would also be able to interact with the user data already stored, and offer integrated signups between both websites. talks.cam 2.0 would eventually supercede the original talks.cam website, adding greater usability and social networking aspects to the website.

However in terms of the data we already store, simple extensions would include:

- finer grained privacy controls
- image uploading
- providing RSS feeds, as talks.cam does
- providing detailed reports to the speakers or providers of the talks - for example, the number of users who were interested in a talk, who commented on a talk, or the highest rated lists.

Longer extensions would be

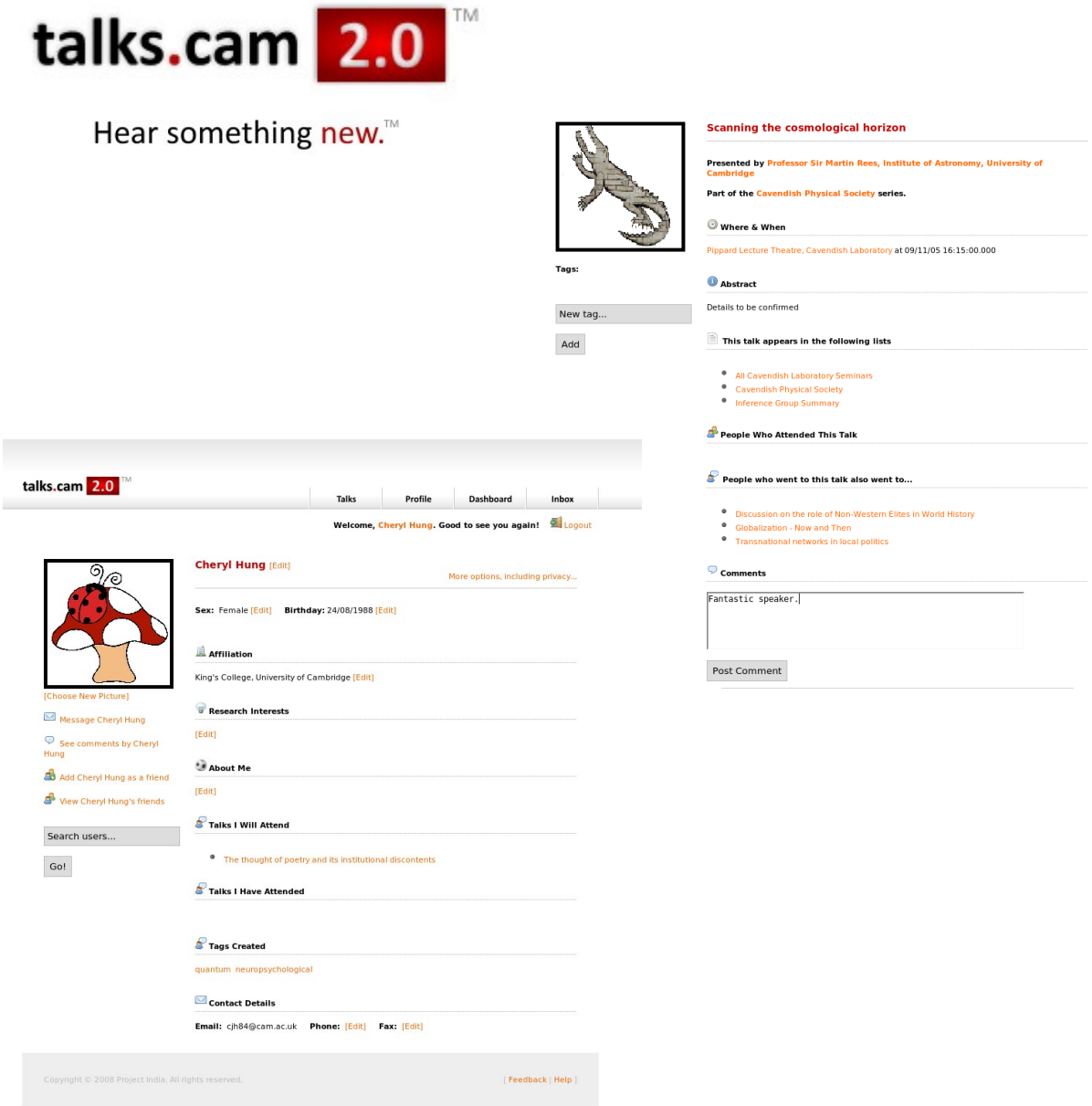
- allowing Facebook-style groups, which could be associated with a set of talks or lists and providing discussion and privacy controls for those
- providing multimedia uploading eg.

- live video feeds from those where cameras are set up, or by users who have a mobile phone camera to upload to the Internet (via an associated mobile application).
- uploading video and audio after the talks are done, in order to produce an archive for people who may want to see past talks.

Overall conclusions

The essential functions of the website (such as viewing a talk and viewing a profile) are effective, simple to use, and thoroughly integrated into the site. The original brief was for a Web 2.0 extension to talks.cam, which we achieved by extracting the talks data in an independent method from XML, then building our web 2.0 features (such as users, commenting, messaging) on top. We have more or less fulfilled our acceptance criteria, and almost every feature originally specified in the feature list is provided.

talks.cam 2.0 stands as a usable, productive website, and has been a successful project.



Company India

Web 2.0 Research Gateway

Cheryl Hung, Alex Bradbury, Tom Coxon, Edwin Li, Enzhe Zhang