A Social Networking Approach to Integrated Communication in OLEs

Hussein Suleman
University of Cape Town
Rondebosch, Cape Town
hussein@cs.uct.ac.za

ABSTRACT

Online Learning Environments (OLEs) provide learners and facilitators with an ever-increasing collection of tools for group collaboration and communication. This unfortunately also means an ever-increasing number of information sources to track and manage. This papers describes an attempt to integrate both synchronous (chat) and asynchronous (forum) communication systems in an OLE. An experimental system was designed to look and function like a social networking system that gives users a constant stream of notifications of new activity. It was expected that advanced users, such as Computer Science students, would feel more comfortable when using such a system. Feedback from users indicates that some users do indeed accept the new approach readily and most users are able to identify with the social networking approach taken in the system.

Categories and Subject Descriptors

H.4.3 [**Information Systems Applications**]: Communications Applications; H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces; K.3.1 [**Computers and Education**]: Computer Uses in Education—*Collaborative learning*

General Terms

Design, Human Factors

1. INTRODUCTION

Online communication takes many forms depending on appropriateness to task and the facilities available in a particular environment. In Online Learning Environments (OLEs), this typically includes mailing lists, chat rooms and discussion fora. Mailing lists and discussion fora are meant for asynchronous communication, where users post messages at different times and not all users are online at the same time. In such systems, each post potentially leads to replies and posts can be archived for future reference. In contrast, chat rooms are meant for synchronous communication, where a group of participants is simultaneously online and each posts his/her comments in real-time. Both of these approaches to communication are group-based, thus supporting group-based learning in large classes.

In a learning environment this split modality of communication is not necessarily desirable as learners need to keep track of two streams of conversations that may have a bearing on classroom-related tasks. It is common for questions to be repeated in chat rooms and fora because individual students may place more emphasis on one or the other. FeedChat was therefore designed to address this problem by providing a combination of a chat room and forum as a single communications mechanism for students.

To avoid the difficulty of students needing to learn a new and foreign paradigm for communication, FeedChat was modelled on the notion of a newsfeed, as provided by many social networking systems and news websites. Rather than move student communication into a social networking system, communication in the learning environment was modeled on current practices in social networking so students would be familiar with it.

This paper presents the design of the FeedChat system and discusses the reactions of students to this unusual communication system.

2. BACKGROUND AND MOTIVATION

Social networking environments emphasize the integration of content from different internal and external sources. This is accomplished using techniques that are broadly named Web 2.0 [3]. This includes: thin clients based on AJAX [1], using Web Services for client-server communication; and feeds (lists of items) using RSS [4] for everything that is potentially useful to users.

Facebook[1] and Twitter[2] are two of the most popular social networking applications, and are exemplars of the types of interaction available. Facebook provides each user with a rich online profile that may contain text and multimedia such as photographs, and the key notion of a user-specifiable textual status. Users then are connected to other users directly (as *friends*) or indirectly (via groups and shared applications). In order for a user to keep track of the statuses and comments of others, all updates are gathered into a

[1] Website: http://www.facebook.com/

[2] Website: http://www.twitter.com/

central user-specific news feed that is the first page a user sees upon logging in. Twitter focuses on allowing users to specify a textual status and update it regularly. Users may subscribe to (*follow*) the status updates of other users via a feed similar to Facebook's. In both Facebook and Twitter, comments/statuses from users appear in reverse chronological order, with a text box at the top to add new comments/statuses.

Traditional pre-Web 2.0 applications behave somewhat differently. In discussion fora, users are able to post messages to a specific topic area or group. Posts and their replies within a single thread usually are sorted in chronological order so a single thread of conversation can be read top-down, with a text box to add to the thread at the bottom. PHPBB[3] is a popular open source tool implementing such a discussion forum system. In chat rooms, users typically see a list of comments that scroll upwards, and are able to add a message at the bottom of the screen. Most OLEs (e.g., Sakai[4]) have some form of chat room embedded within the application. Chat rooms are well established as a consequence of popular client-server protocols such as Internet Relay Chat [2] and XMPP [6]. Figure 1 illustrates the end-user interfaces of a typical PHPBB discussion forum and Sakai chat room.

There is a subtle but definite disconnect between the distinct and separate services of traditional communication systems and the integrated world view presented in a social network feed. FeedChat attempts to provide a merger of the traditional communications systems (chat and forum) but within a single environment, using the familiar user interface structure of a social networking application with its Web 2.0 leanings.

3. DESIGN OF FEEDCHAT

3.1 Requirements

FeedChat is meant to look like a news feed of comments from students, but have the behaviour of a real-time chat room. The following is a list of the features that were deemed necessary for the system to be useful to students who are accustomed to other online communication tools:

- The most recent messages appear at the top of the screen, with the comments in reverse chronological order. This is because the most recent discussion is probably of greatest interest to students.
- All messages are persistent so students may easily refer back to past conversations. This is unlike most chat room technology where it is not possible to access archives of prior conversations.
- The list of messages are updated in real-time as new messages are posted by students. Thus a student need not reload the page to see new activity and the system can be used like a chat room.
- The messages are partially threaded so discussions with a related theme or replies to a question still maintain their associations, unlike in a chat room. Replies may only be added to the main post, and not to other replies - this keeps the user interface simple.
- Only the most recent messages are displayed, with an option to display older messages on demand. This reduces the bandwidth requirements of the client application.
- Users are alerted when others respond to their questions or comment further on discussions they are involved in. This ensures that the poser of a question is notified when her/his question is answered, without her/him having to constantly scroll down to the original message. There is a link for the user to navigate directly to this discussion.
- Users may have a profile picture or avatar for easier identification of users/conversations.
- Announcements are assigned a special status within the system such that they are persistent and given a higher priority than other messages. They are displayed in a separate box in the user interface.

Figure 2 shows the user interface for the FeedChat application. The left pane displays messages in reverse chronological order. The link to add messages causes a form to appear at the top of the page, wherein users can type a new message. The Reply links function similarly, except the forms are at the bottom of each thread. On the right-hand-side, announcements are at the top, followed by a list of notifications and users who are online.

As a compromise between chat rooms and discussion fora, new FeedChat discussions are added to the top of the list but replies to existing posts are added to the bottom of the respective posts.

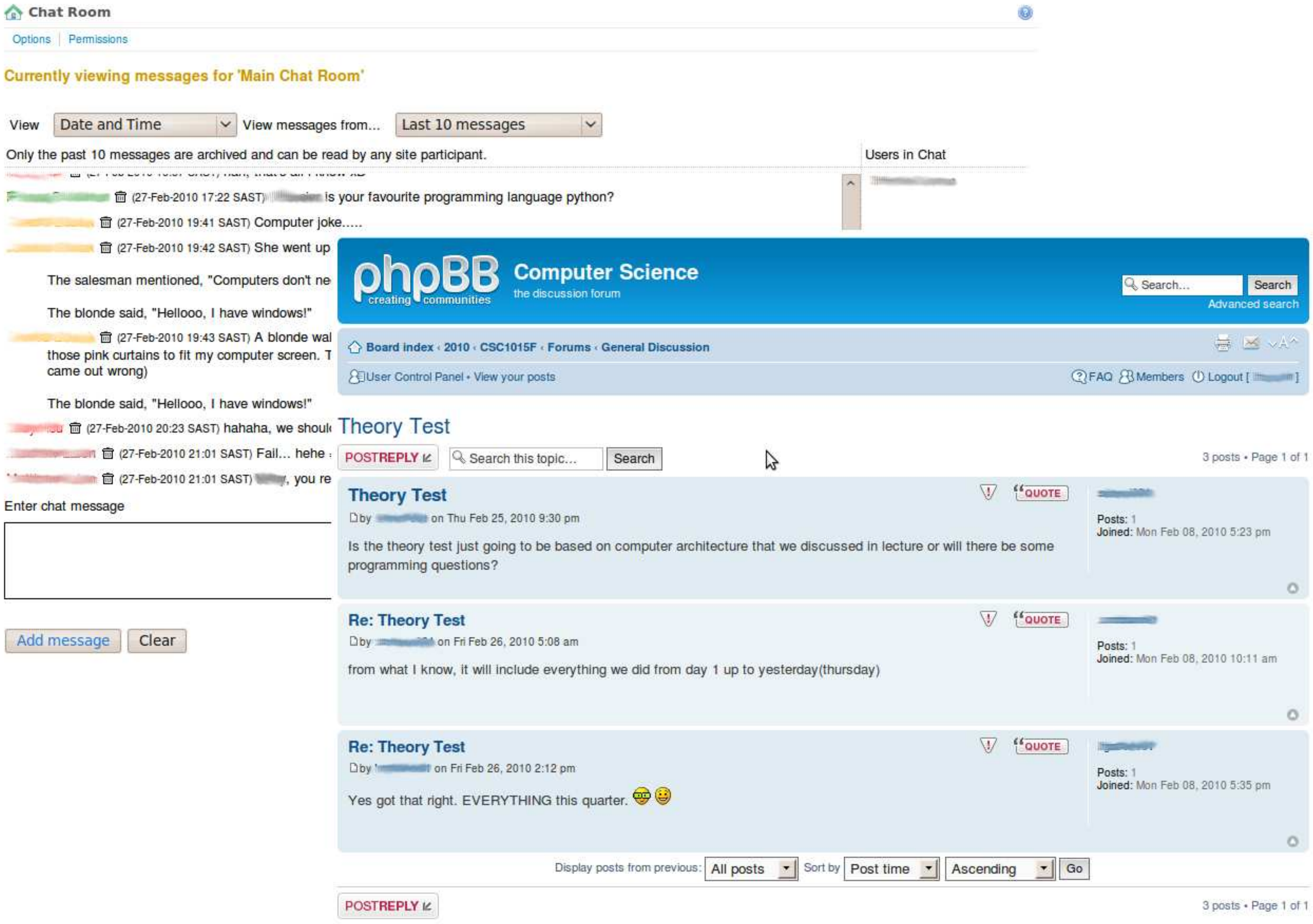
3.2 Technical Details

FeedChat was implemented as a client-server Web-based application using AJAX/Javascript [1] for a thin browser-based client and Perl+Mysql for the server. The client contains all the user interface logic while the server manages the data and provides a Web Service interface for communication.

The biggest difficulty with such Web applications is the lack of persistent connections to a server. This persistent connection is necessary for a real-time chat facility. FeedChat uses the Comet [5] programming technique to simulate this functionality using long-lived connections. A client makes a request and, if the server does not yet have data to formulate a response, the server waits until data is available before responding; the client then processes the data and immediately makes another request to the server. The Feedchat server polls its database once every second for new messages - thus any new messages will be sent to online users within 1 second. The server always times out after 60 seconds, so if a user closes his/her browser, the virtual connection terminates within 60 seconds.

To maintain a list of online users, every request from a user causes the updating of a timestamp. Every response then contains a list of all users with timestamps within the last

[3] Website: http://www.phpbb.com/
[4] Website: http://www.sakaiproject.org/



Currently viewing messages for 'Main Chat Room'

View: Date and Time | View messages from... Last 10 messages

Only the past 10 messages are archived and can be read by any site participant.

Users in Chat

[redacted]

[redacted] (27-Feb-2010 17:22 SAST) [redacted] is your favourite programming language python?

[redacted] (27-Feb-2010 19:41 SAST) Computer joke.....

[redacted] (27-Feb-2010 19:42 SAST) She went up

The salesman mentioned, "Computers don't ne

The blonde said, "Hellooo, I have windows!"

[redacted] (27-Feb-2010 19:43 SAST) A blonde wal those pink curtains to fit my computer screen. T came out wrong)

The blonde said, "Hellooo, I have windows!"

[redacted] (27-Feb-2010 20:23 SAST) hahaha, we shoul

[redacted] (27-Feb-2010 21:01 SAST) Fail... hehe :

[redacted] (27-Feb-2010 21:01 SAST) [redacted], you re

Enter chat message

Computer Science
the discussion forum

Search...
Search
Advanced search
Board index ‹ 2010 ‹ CSC1015F ‹ Forums ‹ General Discussion
User Control Panel • View your posts
FAQ
Members
Logout [[redacted]]

Theory Test

Theory Test
by [redacted] on Thu Feb 25, 2010 9:30 pm

Is the theory test just going to be based on computer architecture that we discussed in lecture or will there be some programming questions?

[redacted]
Posts: 1
Joined: Mon Feb 08, 2010 5:23 pm

Re: Theory Test
by [redacted] on Fri Feb 26, 2010 5:08 am

from what I know, it will include everything we did from day 1 up to yesterday(thursday)

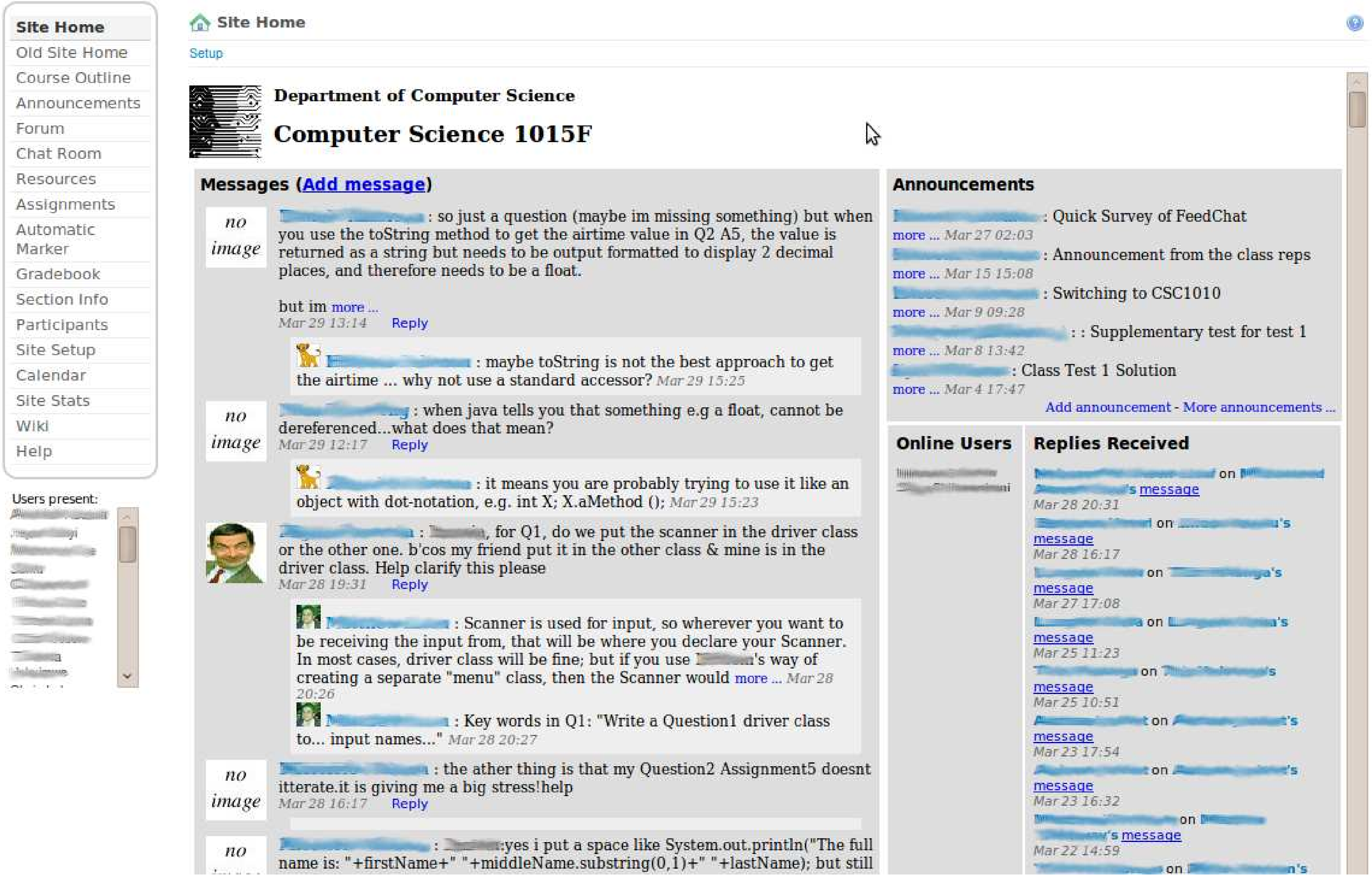
[redacted]
Posts: 1
Joined: Mon Feb 08, 2010 10:11 am

Re: Theory Test
by [redacted] on Fri Feb 26, 2010 2:12 pm

Yes got that right. EVERYTHING this quarter. 😎 😀

[redacted]
Posts: 1
Joined: Mon Feb 08, 2010 5:35 pm

Figure 1: Sample chat room (Sakai) and discussion forum (PHPBB) (with student names anonymised)



Department of Computer Science

Computer Science 1015F

Messages (Add message)

no image [redacted] : so just a question (maybe im missing something) but when you use the toString method to get the airtime value in Q2 A5, the value is returned as a string but needs to be output formatted to display 2 decimal places, and therefore needs to be a float.

but im more ...
Mar 29 13:14 Reply

[redacted] : maybe toString is not the best approach to get the airtime ... why not use a standard accessor? *Mar 29 15:25*

no image [redacted] : when java tells you that something e.g a float, cannot be dereferenced...what does that mean?
Mar 29 12:17 Reply

[redacted] : it means you are probably trying to use it like an object with dot-notation, e.g. int X; X.aMethod (); *Mar 29 15:23*

[redacted] : [redacted], for Q1, do we put the scanner in the driver class or the other one. b'cos my friend put it in the other class & mine is in the driver class. Help clarify this please
Mar 28 19:31 Reply

[redacted] : Scanner is used for input, so wherever you want to be receiving the input from, that will be where you declare your Scanner. In most cases, driver class will be fine; but if you use [redacted]'s way of creating a separate "menu" class, then the Scanner would more ... *Mar 28 20:26*

[redacted] : Key words in Q1: "Write a Question1 driver class to... input names..." *Mar 28 20:27*

no image [redacted] : the ather thing is that my Question2 Assignment5 doesnt itterate.it is giving me a big stress!help
Mar 28 16:17 Reply

no [redacted] : [redacted]:yes i put a space like System.out.println("The full name is: "+firstName+" "+middleName.substring(0,1)+" "+lastName); but still

Announcements

[redacted] : Quick Survey of FeedChat
more ... *Mar 27 02:03*

[redacted] : Announcement from the class reps
more ... *Mar 15 15:08*

[redacted] : Switching to CSC1010
more ... *Mar 9 09:28*

[redacted] : : Supplementary test for test 1
more ... *Mar 8 13:42*

[redacted] : Class Test 1 Solution
more ... *Mar 4 17:47*

Add announcement - More announcements ...

Online Users

[redacted]

Replies Received

[redacted] on [redacted]'s message
Mar 28 20:31

[redacted] on [redacted]'s message
Mar 28 16:17

[redacted] on [redacted]'s message
Mar 27 17:08

[redacted] on [redacted]'s message
Mar 25 11:23

[redacted] on [redacted]'s message
Mar 25 10:51

[redacted] on [redacted]'s message
Mar 23 17:54

[redacted] on [redacted]'s message
Mar 23 16:32

[redacted] on [redacted]'s message
Mar 22 14:59

[redacted] on [redacted]'s

Figure 2: Feedchat user interface (with student names anonymised)

Table 1: Web Services provided by FeedChat server

Function Name	Description
getFirstMessages	get initial view of most current messages, announcements and notifications
getMoreMessages	get next batch of messages, announcements or notifications
getOneMessage	get a single message in response to user clicking on a notification
postMessage	post a message, reply or announcement

minute. This results in the list of online users being at most inaccurate by a minute, which is arguably a reasonable approximation.

Messages, announcements and notifications are all interspersed in a transaction-oriented manner. The client application processes each differently and makes updates to the relevant pane when one of the appropriate data types is received.

Definitions of the functions provided by the Web Service interface are listed in Table 1.

4. ANALYSIS AND EVALUATION

The system has been in use in a first year Computer Science course for just over one month. There are approximately 300 users, almost all of whom claim to have had some experience with social networking sites such as Facebook. In analysing the usage of the system, there are multiple sources of evidence.

4.1 Online feedback

Students commented in the system itself about its features when it was first launched. These comments took on 3 flavours:

- Some students were very positive about the system and liked it a lot.
- Some students were very negative about the system because it was considered to be substantially different from the previous chat room and discussion forum. This was an unexpected reaction and could indicate that some students prefer stability over the possibility of a more integrated communications medium.
- Some students were reserved in their comments and suggested that the system needed mechanisms to search messages, edit messages, etc. It is important to note that no such features were available in the old chat room and students have never commented on that before. This change may be because students identify more with the style of the interface and therefore understand the possibilities in relation to social networking sites.

4.2 Survey of students

A short online survey was conducted among the students in the class to get a sample of their opinions on the system. All students were invited to participate and 25 responses were obtained in a 2-day period.

Most of the respondents (84%) had used computers for more than 6 years and only 1 user had used computers for less than 3 years. 17 users had used chat rooms for at least 3 years and 14 users had used discussion fora for at least 3 years. All respondents had used Facebook for more than a year, implying that they had all used it prior to coming to university.

Users were asked which FeedChat features they had used. More than 80% of respondents had viewed messages and navigated through the interface using the links provided. Approximately 70% had posted a message or replied to a message from another student.

Users were asked if they understood the features of the system. 72% understood how the notification system worked but only 40% realised that the system worked in real-time. This was confirmed in the following question where users were asked how well FeedChat met its design goals of a multi-purpose communication tool. 44% of users thought that the system was average or below-average as a chat room. In contrast, 60-70% of users felt that the system was at least ”good” as a threaded discussion forum, feed or general-purpose merged communication platform.

The students were finally asked for general comments. The responses mirrored those on the site (some positive, some negative, some asking for more features). Two students asked for better usability and two asked for a means of categorising/searching through the data. One student commented on the system's inability to work properly on particular computers.

4.3 Reflections on changes in communication

While FeedChat was designed to operate as a chat room, students are very wary of posting messages and most messages are class-related. It was expected that students would post more messages but this has not been the case. There are more questions than would have been asked in the discussion forum, but there are fewer discussion threads than occurred in the old chat room. This unexpected phenomenon needs to be investigated more thoroughly.

5. CONCLUSIONS AND FUTURE WORK

The development of an integrated chat room and discussion forum has potential to change the way students communicate in OLEs. Social networking has imposed its stamp on most prospective Computer Science students before they arrive at university - so it seems sensible that students will identify with the mode of communication used in social networking.

The initial results of observing student interaction point to a number of useful and unexpected outcomes. Firstly, while an unconventional interface is interesting and exciting to some students, others prefer what is well known and well-understood. Secondly, once students categorise the system as one that falls within the social networking paradigm, their expectations are different - they expect more advanced features of such a system than one that is obviously Web 1.0!

Current work focuses on making the system more cross-platform and prettier as well as incorporating more advanced social networking features (e.g., a tag cloud) that will make navigation easier for students. In addition, the layout of the system is being redesigned to provide the look and feel of a chat room to emphasize this aspect of the system.

6. REFERENCES

[1] J. J. Garrett. Ajax: A new approach to Web applications. 2005.

[2] J. Oikarinen and D. Reed. Internet relay chat protocol: RFC 1459. May 1993.

[3] T. O'Reilly. What is Web 2.0: Design patterns and business models for the next generation of software. September 2005.

[4] RSS Board. RSS 2.0 specification. March 2009.

[5] A. Russell. Comet: Low latency data for the browser. 2006.

[6] P. Saint-Andre. Extensible messaging and presence protocol (XMPP): RFC 3920. October 2004.