Azureus Plugin for Facebook Integration

Mike House-Vording, 100300955
mhvordin@connect.carleton.ca

1. Introduction

This project seeks to integrate two major trends currently taking place on the internet: *file sharing* and *social networking*. To accomplish this I have developed a plugin for the open source Java program Azureus, which implements the BitTorrent protocol. The plugin allows for integration between Azureus and Facebook, the later being a popular social networking site.

1.1 BitTorrent Protocol

The BitTorrent protocol is a peer-to-peer communications protocol. It was originally designed by Bram Cohen, with a first implementation being released on July 2, 2001. It was designed to allow large amounts of data to be distributed without the original distributor having to incur all of the costs associated with hosting data on a network. It accomplishes this by having downloaders share parts of the file(s) that they already have with other downloaders who do not yet have those parts. This allows the bulk of the file uploading work to be decentralized and spread out amongst the clients. In fact, the original distributor is not even needed after every part of the file has been obtained by at least one of the clients.

In BitTorrent terminology the original distributor is called the seeder. After a client has downloaded the entire file they are said to be seeding or sharing that file. A tracker is a computer that coordinates the file distribution. Like its name implies it keeps track of who is downloading and sharing each file that it is responsible for. When a client wishes to first begin downloading a file it must first request a list of other clients and seeders from the tracker. The address of the tracker is located within the torrent file, which is the file required for a client to start a download. The torrent file also contains information about the names and sizes of the files and the directory structure if applicable.

1.2 Azureus

Azureus is an open source BitTorrent client written in Java. It was originally released in June 2003. Since version 2.5.0.4 it has been released under the GNU General Public License

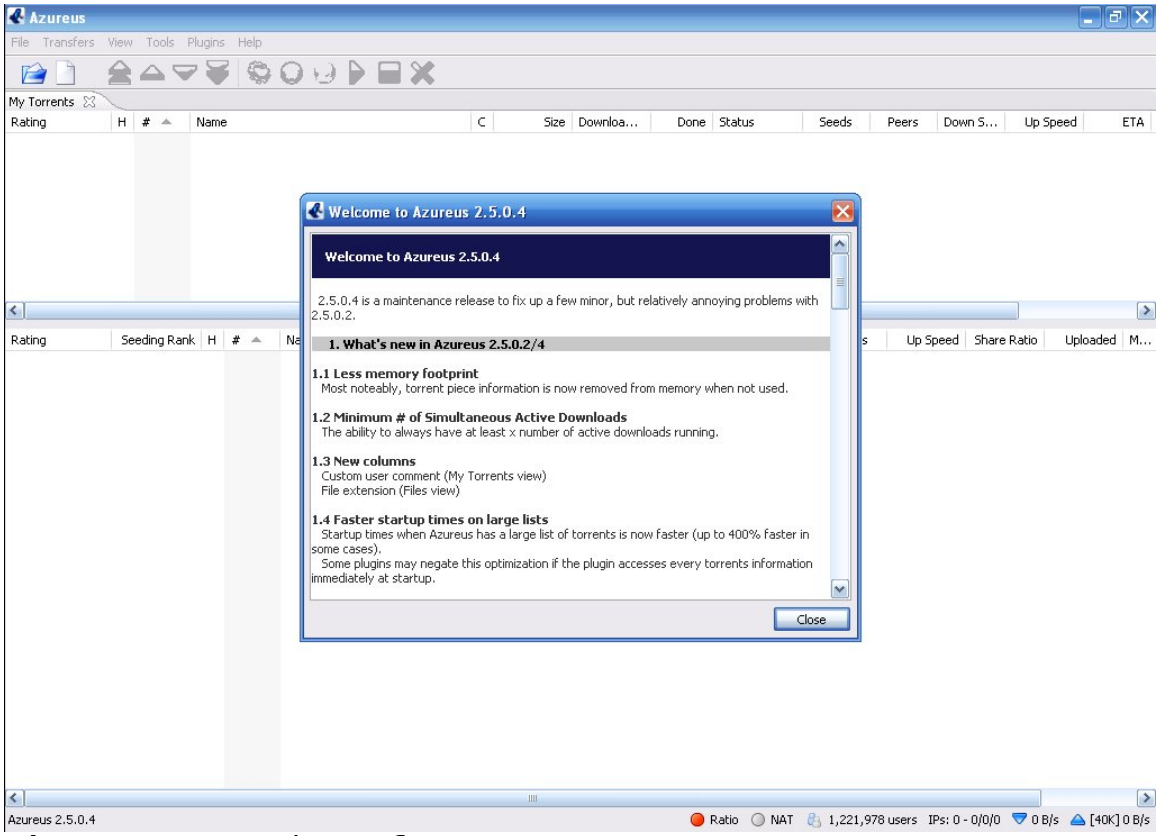

Fig. 1. Screenshot of Azureus 2.5.0.4

The current version of Azureus is 3.0.4.0. As of version 3 Azureus comes bundled with added functionality called Vuze. Vuze is a content service which allows users to upload and download content as well as to purchase commercial content. This added functionality is not relevant to this paper, but I mention it to avoid confusion. When Azureus of version 3 or higher is first started it defaults into the Vuze mode.

Fig. 2. Screenshot of new Vuze functionality wrapping the core Azureus program

Azureus includes in its architecture a plugin system that allows developers to add new functionality without having to change the core program. The new plugin classes are stored in a jar file and placed under the 'plugins' directory along with a proper property file. Upon startup Azureus loads all plugins and initializes them.

1.3 Social Networking

Online social networking software revolves around the concept of a user creating an account and associating themselves with other users. Most sites allow a user to upload a photo of themselves and then to add other users as friends. Popular versions of these include MySpace, Bebo and Facebook.

1.4 Facebook

Facebook was originally launched in 2004 by Harvard student Mark Zuckerberg. It began as a social networking service for Harvard students before expanding to all Ivy League schools, and then to all universities and colleges. From there it then allowed high school students and members of certain large companies to join before finally opening its

doors to anyone over the age of 13 in September 2006. Currently there are over 58 million registered users on the site.

The site features standard functionality such as a wall for leaving messages, a marketplace for posting classifieds, and a status feature for updating friends on your current status. These features are delivered by way of the facebook applications platform, with the standard features being applications made by facebook and installed by default. The facebook platform is also open to third party developers who wish to create their own facebook applications.

There are many different components in a facebook application, none of which are mandatory for the developer to implement, however almost all applications will implement the profile box. The profile box is the GUI that will appear in the user's profile. The user's profile is the main page that everyone sees when they want to view the user and it contains basic information about the user as well as the profile boxes of all of the facebook applications the user has chosen to install. This includes the default facebook applications which give facebook its basic functionality.

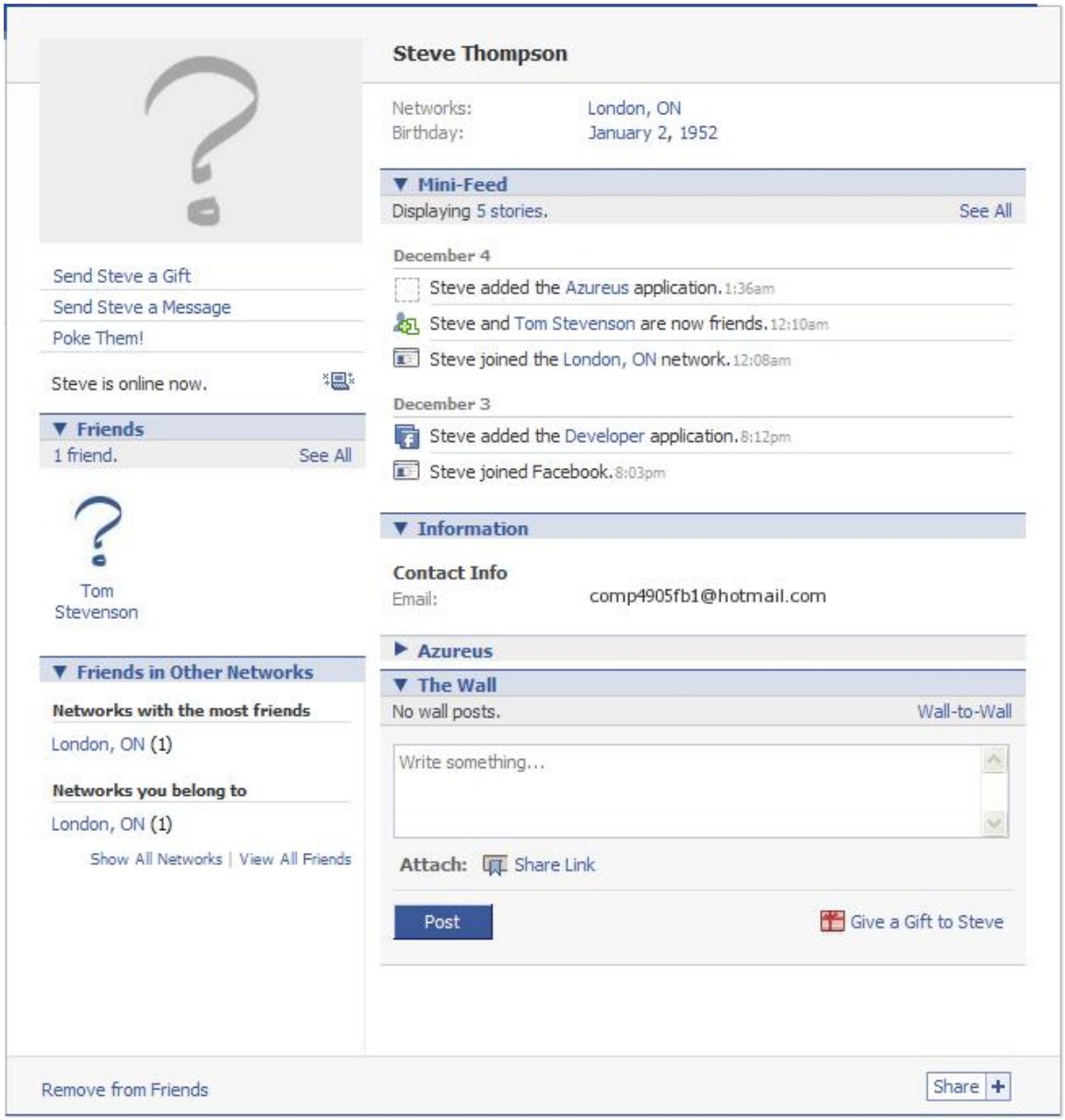

Fig. 3. Screenshot of the profile page of the facebook account used to demo project

2. Detailed Background & Setup

2.1 Setup

To run this plugin the user must have Azureus installed on their computer and a facebook account. There are two separate installations that must take place before the plugin can function, one for the plugin itself and one for the facebook application. In

order to install the plugin the user must take the ‘FacebookPlugin’ folder provided and place it under the ‘plugins’ directory of the Azureus installation. The ‘FacebookPlugin’ folder should contain the following files: *BrowserLauncher2-10rc4.jar, facebook.jar, azfacebook.jar,* and *plugin.properties*. After this is complete Azureus will automatically initialize the plugin upon its next startup or restart. Before starting or restarting Azureus however, the user should make sure the Azureus application is installed in their facebook account. This can be done by visiting the application’s page at http://www.facebook.com/apps/application.php?id=7873326263 and clicking the “add application” link.

2.2 How it Works

Now that the user has installed the plugin and the Facebook application they may start or restart Azureus. One of the first things Azureus does when it starts loading is to initialize each of the plugins that have been installed. On initialization the Facebook plugin will open an internet browser window which will request that the user login to their Facebook account. This is required to allow the plugin to establish a session with the Facebook application. It is recommended that the user login to Facebook before starting Azureus for convenience. The reason for this is that the plugin has no way to know when the user has completed logging in so in order to speed up the startup of Azureus only a fixed amount of time is given for the user to login to their Facebook account. If the user has already logged in to facebook then the session will be coordinated automatically.

Now all of the files that the user is downloading and seeding will be displayed on their Facebook profile page. Another user with this program installed who visits this Facebook page would be able to simply click on a file of interest and it would seamlessly begin downloading on their computer. My goal is to combine the power of social networking seamlessly with file sharing to speed up the spread of information. An example would be that of a band who wishes to spread a copy of one of their songs to generate interest in their band. They could seed this song on Azureus, and with my plugin installed everyone who visits their profile on Facebook would have the opportunity to start downloading at the click of a button. In turn, friends of the friends of the band may visit the page of one of these downloaders and decide to download the song out of interest since their friend was downloading it. This would quickly have a ripple effect that could easily spread throughout the world. And obviously the band example is just one of countless possibilities.

2.3 Technical Details

Plugins must have a class which implements the org.gudy.azureus2.plugins.Plugin class of the Azureus library. This class defines a method called ‘initialize’ which takes as an argument an org.gudy.azureus2.plugins.PluginInterface implementing object. This is the method that will be called by Azureus when it initializes the plugin. The

PluginInterface object provides the plugin with a means of controlling and viewing the information of Azureus.

There are three major components of the Facebook platform: Interface (API), Query (FQL), and Markup (FBML). The Facebook API uses a REST (Representational State Transfer) based server. Calls are made to the Facebook API server using HTTP GET and POST calls. The first thing a program that wishes to communicate with the Facebook server must do is create a session. This is done by requesting an authentication token from the server. Then using that token as a parameter you must have the user login to their facebook account. After the login is complete the program can then request the session key from the Facebook server. The program will than use that session key when it makes any further method calls.

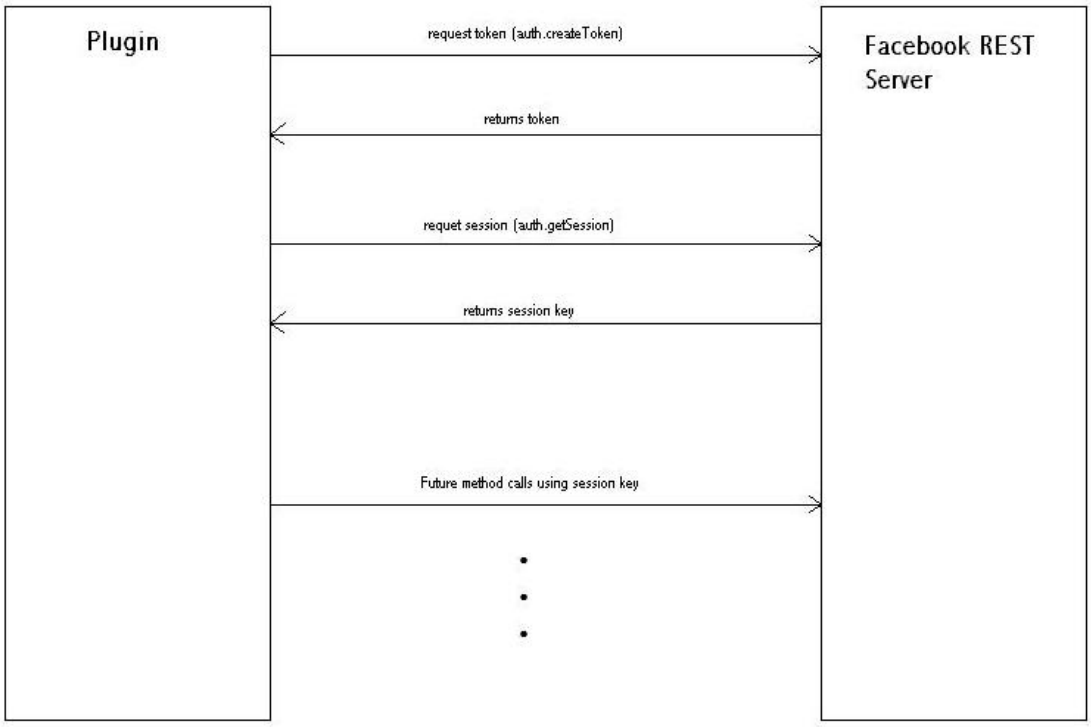

Fig. 4. Facebook REST server session authentication sequence.

FQL is Facebook's own query language which is heavily based on SQL. The developer can create their own queries to retrieve various pieces of information about users within Facebook.

FBML is Facebook's markup language. It is used to display information within the various components of the Facebook application, such as the profile box. It uses a subset of HTML with added Facebook tags.

The plugin starts a session with the Facebook REST server upon initialization. It keeps track of the status of files being downloaded or seeded within Azureus. When there is a change it calls the Facebook server and has it update the FBML of the profile box of the user to reflect which files are currently being downloaded or seeded.

When User A clicks on a file that is displayed in the Azureus profile box of User B's Facebook profile the browser of User A sends an HTTP GET request to User A's Azureus plugin. The request informs the Azureus plugin that User A would like to start downloading File C from User B. User A's plugin then connects to the plugin of User B and requests the torrent information of File C which is then sent by the plugin of User B. User A is now downloading File C with all of the setup done behind the scenes after User A clicks on the file.

3. Results

The plugin is currently functioning correctly. The following images will show Azureus downloading one file and seeding one file while the Facebook application correctly has those files listed.

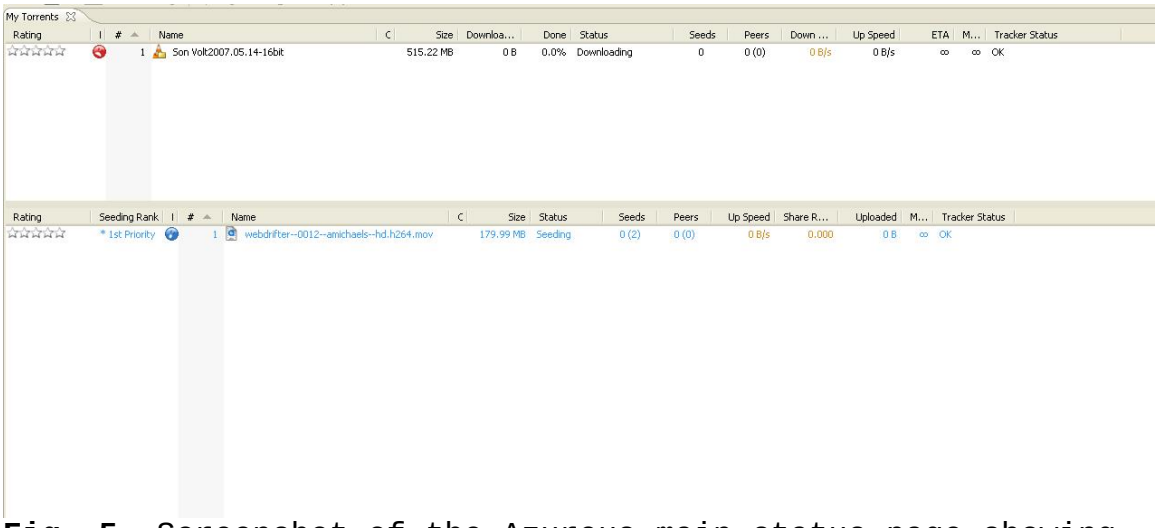

Fig. 5. Screenshot of the Azureus main status page showing one file being downloaded and one file being seeded

Fig. 6. Screenshot of the Facebook application displaying files being downloaded or seeded

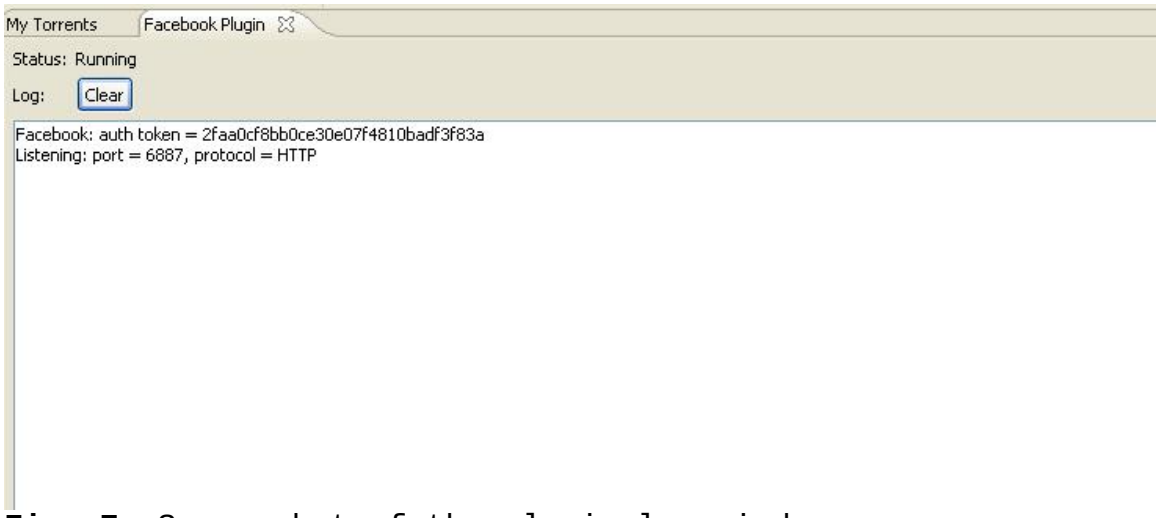

Fig. 7. Screenshot of the plugin log window

4. Future Work & Conclusion

The various other Facebook application components provide ample opportunity to expand on the functionality of this program. The first such opportunity comes from the "User Dashboard" component of a Facebook application. From facebook: "*The user dashboard is usually accessed from the top right drop-down menu in the user's profile. This is where users manage their own content within an application or see information about another user's content within that application.*" This would provide an excellent spot to add remote user interface controls from which the user could manage their own downloads from another computer by logging into their Facebook account. Such a feature would add greatly to this plugins usefulness.

Another idea would be to add functionality for the user to control which downloads are visible publicly and which are not. At present all files that the user downloads and seeds are visible on their Facebook profile. Adding this option may add convenience to users who may not wish to spread every file they download through their social network.

After these features are implemented the next step would be to create an appealing page in the Facebook product directory to attempt to gain as many users as possible. This page would offer installation instructions and guide them to a server where they could download the plugin for Azureus. Conversely, submitting the plugin to Azureus and getting it listed among their plugins would also help generate usage.

I believe there is great potential in combining social networking with file sharing seamlessly like I have. A site like YouTube is the closest example of something like this; however it has a few disadvantages. While it does have some aspects of social networking in the sense that you can subscribe to view user's content, it lacks the

personal details as well as the links that can be made between friends of friends of friends, etc.

My next steps will be to distribute this plugin to a test group of users and collect feedback while at the same time implementing the other possible features.

References

1. Facebook Developers Guide, [Online]. Available: http://developers.facebook.com/
2. Azureus Plugin Javadoc, [Online]. Available: http://azureus.sourceforge.net/plugins/doc/
3. Wikipedia, “Azureus” [Online]. Available: http://en.wikipedia.org/wiki/Azureus
4. Wikipedia, “Facebook” [Online]. Available: http://en.wikipedia.org/wiki/Facebook
5. BitTorrent Protocol Specification, [Online]. Available: http://www.bittorrent.org/protocol.html
6. Azureus Plugin Development Wiki, [Online]. Available: http://www.azureuswiki.com/index.php/Plugin_development