Implementing a Usenet-Style Interface On Top of The Gnutella Network

Computer Science 490 under Carl Lagoze
Digital Library Research
Alex Faaborg
May 8th 2002
Part I - Review
• Review rational behind the new interface
• Features implemented, not implemented
• Code demo
• Process

Part II - Technical Discussion
• Top 5 Problems during implementation
• What I learned

Part III - Future
• Future research: metadata interoperability
Part I - Review
Encompassing Metadata vs. Refining Metadata

books about the universe

bigbang.txt

chapters: 11
How this manifests itself in the user experience

Refining searches with metadata

Usenet style interface Groups
Features Implemented
The application is really just 50% complete

If you looked at it with a client server mentality:
only the client is finished.
• **Favorites List, Address Bar**

  - Address bar detects if the group entered is already a favorite group
  - Add/Remove capability
  - Changes are automatically saved to the hard drive in an XML settings file.
Features Implemented

- Favorites List, Address Bar
- **Results Pane / Download**

![Table with file results]

- Shows Results!
- Download files from the network
Features Implemented

- Favorites List, Address Bar
- Results Pane / Download
- XML Rich Queries for metadata

//Calls the new overloaded version of RouterService.query(...)  
guidByte=GUIMediator.instance().getRouter().query(regString, xmlString, minSpeed, type, schemaURI);  
GUID guid = new GUID(guidByte);  
LimeXMLSchemaRepository rep = LimeXMLSchemaRepository.instance();  
LimeXMLSchema schema = rep.getSchema(schemaURI);

searchView.addResultTab(guid,regString,schema,xmlString,this);

Sends a XML query onto the network. The XML rich query packets are ignored and forwarded by uninformed applications.

Note: these methods were written by the LimeWire developers, I just had to understand them enough to use them.
Features Implemented

- Favorites List, Address Bar
- Results Pane / Download
- XML Rich Queries for metadata

- Find Groups Window just gui
• Favorites List, Address Bar
• Results Pane / Download
• XML Rich Queries for metadata

• Find Groups Window  just gui
• Build Group List Wizard  just gui

This wizard will find active groups on the Gnutella Network. You may wish to run this wizard from time to time to keep your group list current.

This may take several minutes.
Features Implemented

- Favorites List, Address Bar
- Results Pane / Download
- XML Rich Queries for metadata

- Find Groups Window ➔ just gui
- Build Group List Wizard ➔ just gui
- Create a Group Wizard ➔ just gui
Features Implemented

- Favorites List, Address Bar
- Results Pane / Download
- XML Rich Queries for metadata

Features Half-Implemented

- Find Groups Window \(\equiv\) just gui
- Build Group List Wizard \(\equiv\) just gui
- Create a Group Wizard \(\equiv\) just gui

Features Not Implemented

- Post Files
- Scoring System (annotate other peoples files)
<table>
<thead>
<tr>
<th>Week</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/06/02</td>
<td>28 hrs</td>
</tr>
<tr>
<td>4/13/02</td>
<td>12 hrs</td>
</tr>
<tr>
<td>4/20/02</td>
<td>37 hrs</td>
</tr>
<tr>
<td>4/27/02</td>
<td>19 hrs</td>
</tr>
<tr>
<td>5/3/02</td>
<td>40 hrs</td>
</tr>
</tbody>
</table>

Credit Hours

Time Spent (195%)
**Hardware**
- 2 Desktop Computers
- 4 Laptop Computers
- 2 Network Hubs
- 1 Router
- 7 Patch Cables

**Software (on each)**
- Fresh install of Win XP Pro with Remote Desktop Connection enabled
- JDK 1.3.1.02
- JBuilder 6.0 Personal
- Current build of Code
Part II – Technical Discussion
Some Facts about LimeWire 2.0.2

• 388 Java Files

• 75,414 lines of code
Some Facts about The Groups Feature

- 14 Java Files
- 2,634 lines of code
What I Learned

• How the innards of a massive commercial application are organized (lots of object oriented programming techniques, very modular).
• How the Gnutella protocol works
• How a Gnutella servant can be implemented in Java
• XML manipulation in Java
• A large number of small programming tricks
• Thread based programming
• Techniques for debugging a networked application
• Use the unique environment of a peer to peer network to improve metadata quality in forced scoring system

• Try to adapt this forced scoring system to solve metadata interoperability problems.