# BOMBYX: An Improved Interface to Chatroom Interactions

**Zed A. Shaw**
Faculty of Commerce
University of British Columbia
3714 W12th Ave
Vancouver, BC
V6R 2N6 Canada
+1 604 221 1117
zed.shaw@ubc.ca

**Karyn A. Moffatt**
Department of Computer Science
University of British Columbia
201-2366 Main Mall
Vancouver, BC
V6T 1Z4 Canada
+1 604 822 5108
kmoffatt@cs.ubc.ca

**ABSTRACT**
Internet chat networks are a popular medium for online group communication. However, they have been oft criticized for having cumbersome interfaces, unwieldy conversations and difficult navigation. Thus this paper offers a new chat client enhanced to combat these shortcomings. The Bombyx Chat Client offers a clean and intuitive graphical user interface that is simple to use. Its foremost contributions are however, Focused Chat and Finder Agent. Focused Chat allows users to filter a conversation channel, focusing on only the users in which they are interested. Finder Agent is a software agent capable of exploring channels looking for specific key words the user has specified as interesting. The Finder Agent was originally designed as an anthropomorphized agent acting through chat interface. However, this was modified, as it became evident that although it may be easy to distinguish an agent from a real person, it is impossible to distinguish a person (pretending to be an agent) from a real agent. Given that the user naturally trust what he regards as “his” agent, a system with anthropomorphized agents burdens the unsuspecting user with a security risk – they could be fooled into providing an untrusted source with private information.

**Keywords**
Software agents, anthropomorphization, user interface design, virtual meeting places, chat rooms, conversation visualization, auditory metaphors

## 1 INTRODUCTION

IRC (internet Chat Relay) is a very popular chat/communication system. It has been extended from its original design of "chat on channels" (similar to HAM radio communications) to allow file transfer, P2P style chat, and even agents called "bots". It is one of the most popular group communications mediums with over 300 networks with approximately 900K users, on over 400K channels on over 3K servers [5].

Yet, after all its development and use, IRC still has numerous problems. First, it has always been plagued with security problems. The clients that people use on IRC are also very difficult to use. They typically force the user to directly communicate with the IRC server's command language. They tend to flood users with information [2], forcing them to sort out conversations on their own. This is typically because the clients are designed by extremely expert computer programmers, and intended for their use. The few "good" clients still have the same "chat on channels" metaphor and only provide minor improvements.

A new kind of "IRC-like" chat system called SILC (Secure Internet Live Conferencing) has managed to create an entirely new communications system that is secure and very flexible. Yet, the clients it uses are in the same poorly designed IRC style. SILC does provide a very nice client API for designing chat clients, and the community is extremely interested in a graphical client.

There is thus motivation for designing an improved chat client interface to facilitate locating and following conversations on SILC or other IRC type chat systems. We have grouped the deficiencies of current chat clients into the following three resolvable issues:

- Unintuitive command based interaction
- Vast number of channels with little indication of topic
- Unstructured conversation within channels with large numbers of participants in a channel

To attack the first problem we have developed a simple and intuitive graphical user interface that guides the novice user through the necessary procedures while still providing a

clean interface for the more experienced user. An agent capable of joining and listening on channels to determine if the channel is of interest to the user has been developed to tackle the second problem. Finally, to reduce the flood of meaningless information the user must process, a focusing ability has been added that allows users to pick out which participants of a specific channel they wish to listen to, much as humans naturally filter large group conversations in a face-to-face setting.

## 2 RELATED WORK

Previous work in this area includes the Butterfly project [7] and Chat Circles [2]. Both of these systems aim to improve chat room interactions addressing some of the problems outlined in the previous section. Each of these systems, however, takes a very different approach and tackles a different subset of the problems.

Butterfly uses a Conversation-Finding agent to help users find conversations of interest on IRC networks. It leaves the actual interface to the chat network untouched, and so it is still ridden with many of the previous criticisms of IRC. The Butterfly agent interacts with the user through a standard chat client. Hence the user interacts with Butterfly as though Butterfly were any other user on a chat channel. This has the advantage of metaphor reuse: the user doesn't need to learn a new interaction paradigm in order to learn how to use the agent. However, it is plagued with a security problem in that a malicious "real" user could pose as an "agent" and dishonestly elicit information from the user. This problem will be discussed in greater detail in the Finder Agent subsection of the Bombyx discussion.

While Butterfly aimed to enhance chat communication with minimal change to the user interface, Chat Circles took a radically different approach redesigning the interaction metaphor entirely. In Chat Circles rather than have the conversation arranged in a logical pattern based on the time sequence of messages, users are spatially spread across the screen. Each user is denoted by a colored circle that grows when the user says something, and fades or brightens with the user's activity level. Users can move their circle around the screen to explore different conversations on the channel.

Chat Circles takes the auditory metaphor to the extreme: users can only hear parts of the channel they are near, although they are aware of the existence and activity of other parts of the channel through the peripheral visual cues given by the circles. Thus conversations begin to emerge in the channel and users must move themselves physically about the chat space to hear (see) the different conversations. Although chat circles does provide an interesting approach to overcoming many of the problems associated with current chat systems it fails to provide an overly exciting solution as it is completely incompatible with existing systems – users with chat circles cannot interact with systems with non-Chat Circles participants as the benefits of Chat Circles depends heavily on the emergence of conversations. This will only happen if all users have a perceived awareness of physical location within the conversation allowing conversation clusters to develop.

It is for this reason that we have proposed the following system, which combines the ideas presented here, overcoming the deficiencies of each.

## 3 THE BOMBYX CHAT CLIENT

Bombyx provides a clean intuitive chat client interface that addresses the problems outlined in the introduction. The simple graphical user interface guides the novice user, but is flexible enough that it is equally valuable to the more skilled user - specifically to the veteran user who is proficient with the current command line based interfaces. The Finder Agent is capable of joining and listening on channels to determine if the channel is of interest to the user. It gathers statistics on channels and uses heuristics to specify how it should explore the server. Finally, the Focus Chat functionality of the system allows users to focus in on the aspects of the channel that interest them, reducing the mass of information that users must filter through while following a conversation. Each of these pieces of functionality is described in the subsections that follow.

### Intuitive Design

The chat client was designed to be extremely undemanding with a minimalist, yet attractive interface. Feature creep was avoided by keeping the client as simple as possible: it simply connects to chat and lets people communicate. Extensive software design standards were used to ensure the software would be easy to use and intuitively designed [3].

In order to ensure that the client would be compatible with existing clients many of the interaction principles were kept the same. We designed with a layering theme, where at the base we kept the classic design and then built our enhancements as a layer above it, keeping all augmentations in the interface so it is compatible with existing clients. This ensures that users may chat with anyone, rather than only Bombyx users.

As much as we wanted to ensure the system would be easy to use for the inexperienced user, we didn't want to make a "dumbed down" piece of software that was easy to use merely because there was a loss of functionality or efficiency. Therefore, it was necessary to design the system with the experienced user in mind. Furthermore, as the largest demographic of potential users is the existing chat population, we wanted to create a system that they would find useful and agreeable, one that provided them with access to the other tools we were providing (the focused chat and the finder agent) without forcing them to retrain their existing skills to fit the new system. Therefore, the system was designed to understand the standard

commands. The user interface was designed so that at each step SILC commands could be entered instead of completing the graphical widgets, allowing the user to choose which type of system they prefer.

Figure 1 shows a screen shot of the *Connect* window. The system maintains a history of the servers the user has previously connected to and presents this list to them at this time. The first time the user connects their list is empty, so the system prompts the user asking if he wants the history populated with a list of default servers. At this time users also have the option of setting their nickname, the default is set to the username of their home directory.

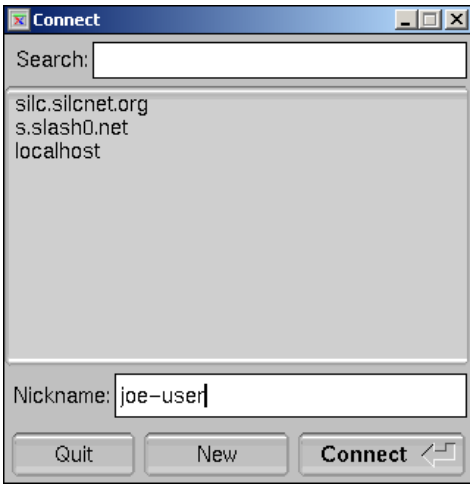


**Figure 1: Screen Shot of the Connect Window**

After connecting to a server, a *Join* window appears as shown in Figure 2. At this point, users can select a channel from the active channel list or start a new channel using the *New* button. The *Search* text box at the top of the screen allows users to constrain the list of active channels to help them find a specific channel.

Once the user has joined a channel, they are presented with the chat interface shown in Figure 3. The conversation is still presented as an ephemeral scrolling list, however many of the system messages that are traditionally sent to the chat window will be rerouted to a log file[1], where users can view them if so desired, although we predict that most will not care. The system message alerting users of the arrival of a new participant on the channel has been preserved, however, in future versions of the system, this message will probably be replaced with a less intrusive indicator of arrival (perhaps by flashing the color of the participants name as it is added to the participant list).

[1] This Feature is planned.

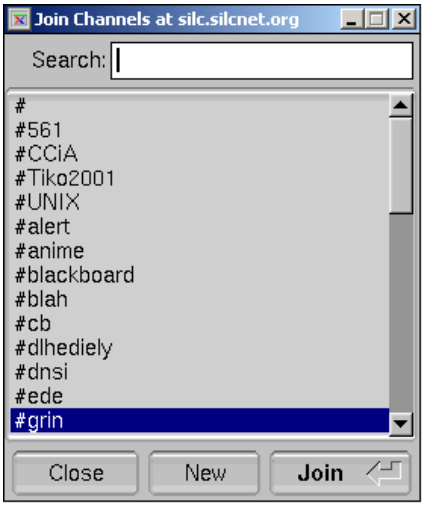


**Figure 2: Screen Shot of the Join Window**

Users send messages by typing them in the text box at the bottom of the screen. They can also enter SILC commands and have them executed as they would in a traditional chat client. The list on the right shows users who are currently on the channel. Traditional chat systems have been criticized on this point, as generally the only time a user is aware of another user on the system is when the user joins or leaves the channel, and when the user says something. It is therefore, common practice for individuals to add meaningless comments (to the already overloaded system) to maintain a perceivable presence [2].

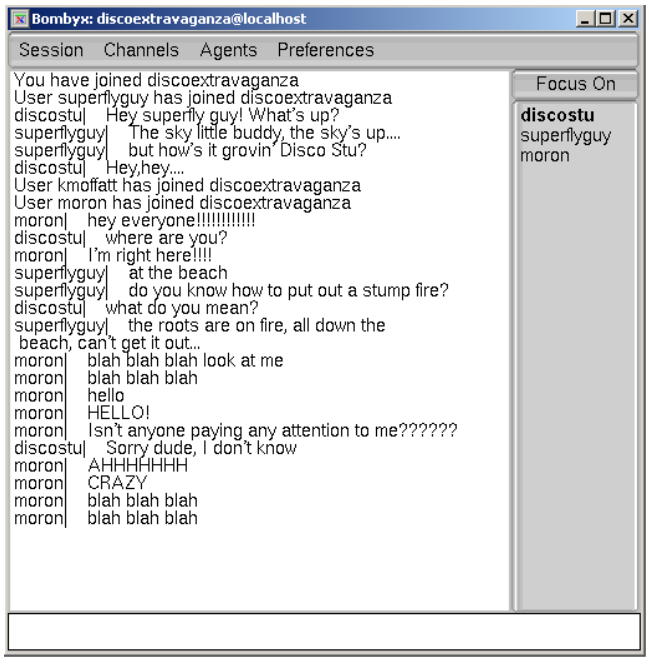


**Figure 3: Screen Shot of the Chat Window**

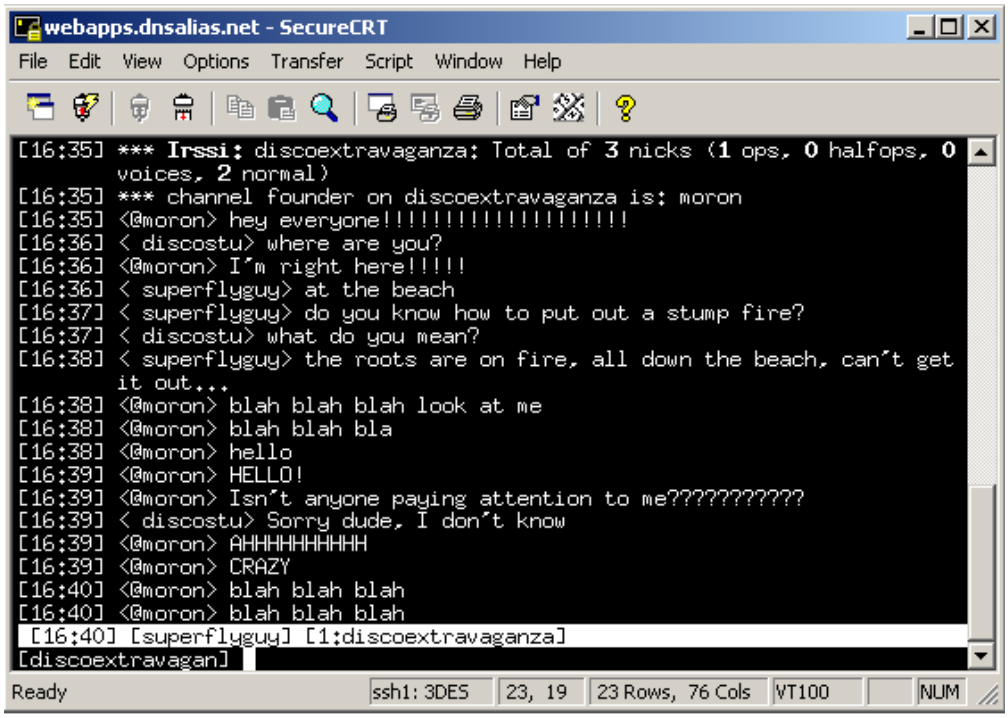


**Figure 4: Screen Shot of the tradition SILC Client interface showing how a SILC Client user and a Bombyx user can chat. Even though the Bombyx user has used the Focused Chat feature, the SILC Client is unaffected**

The list of users also serves as a gateway to the *Focused Chat* feature. From this list, the user can select individuals (or small groups) to chat with either privately[1] or focused. A private chat is a standard feature of SILC and other IRC type chat systems. It separates the users from the rest of the group and allows them to chat independently of others. A focused chat, however, is where the chat recipients are still chatting in the main channel, but the user has "broken" them out of the room so that she can focus on what they are saying.

Focused chat gives users the ability to narrow the conversation to what a few people are saying. Figure 5 shows an example of a focused chat of the chat presented in Figure 3 above. The figure shows how the user can select the participants in which he is interested to reduce the amount of uninteresting information he has to filter through visually.

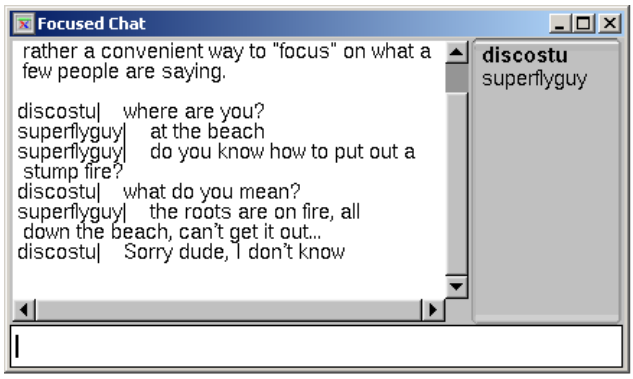


**Figure 5: Screen Shot of the Focused Chat Window showing how the focused chat filters to show just the participants in which the user is interested.**

The important thing to note is that at the system level nothing has changed. Everything is transmitted through the server in exactly the same manner as it is in the main *chat* window. Thus a user of the Bombyx system can chat and focus on users, regardless of what client(s) the other participants are using. The recipients of the chat would not know any differently, as is shown in Figure 4 above. In order to ensure users don't forget they are in a focused chat, the interface does not allow users to close the main chat window without leaving the channel. This way users can see the messages they enter in the focused chat window

[1] This feature is planned.

appear in the main window, reminding them that their conversation is not private.

**Finder Agent**

The conversation on channel metaphor for chat networks has been criticized for being impossible to find conversations on a particular topic. Channel names rarely correspond to the topics being discussed largely because of the dynamic nature of conversations. Channels begin to have a reputation and thus regular visitors don't want to change the name [4], however, this makes it incredibly difficult for new users to find these channels. Thus the only option for users is to search through the channels by joining them, listening long enough to determine what is being discussed, and then moving on until they find a conversation in which they are interested. We felt that this task could be delegated to an agent.

The original design for the agent was to have it use a regular chat interface, where the user would converse with the agent as it would in a regular conversation with any other user. This was the approach used in Butterfly. A screen shot of Butterfly is shown in Figure 6. Comparing figures Figure 4 and Figure 6, the problem with this approach becomes evident: although it may be difficult to create an agent that fully convinces the user, it would not be very difficult for a human to convince the user that he is an agent.

This demonstrates the downside of implementing an agent as an anthropomorphic entity: in certain domains (specifically where users are interacting with real humans they cannot trust) and agents (they should be able to trust), anthropomorphizing the agent creates the security risk that malicious humans may pose as trustworthy agents in order to gain the trust of the user and then use that trust to perform offensive acts.

For this reason we decided that in our domain, having an anthropomorphic agent would be inadvisable. Thus we opted to give the agent a distinct interface that would be impossible for a malevolent user to replicate. Figure 7 shows this interface. The interface is relatively straightforward. On the left the user enters key words and then starts the search. When the agent returns with results the results are displayed on the right. If a large number of results are returned the user can search within the list of returned channels for channel names containing a specific sequence of characters.

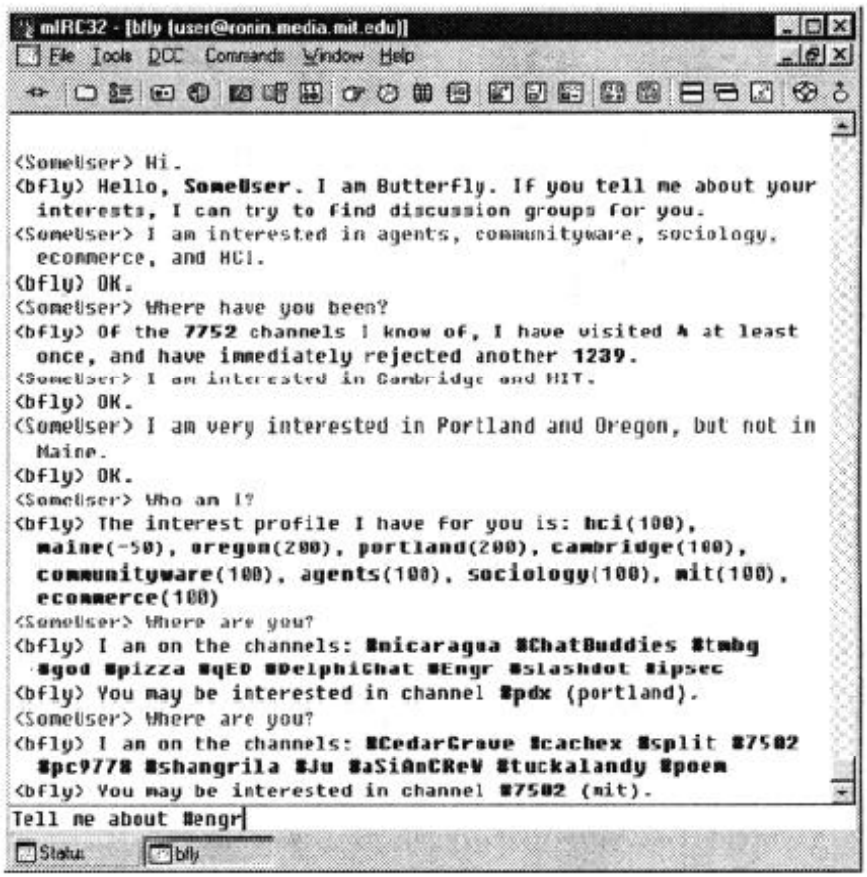


**Figure 6: Screen Shot of the Butterfly Agent for conversation finding**

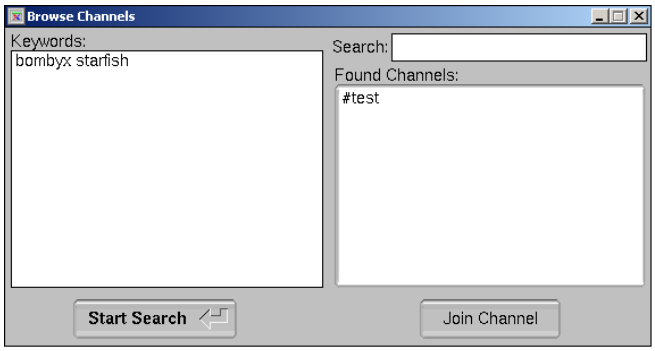


**Figure 7: Screen Shot of the Browse Agent Interface**

The agent works by joining a number of channels, listening to the conversation for the keywords entered by the user and then reporting back to the user any results found. Generally networks limit the number of channels to which a user can connect. The limit on the SILC network is approximately 10, therefore, the agent will connect to a maximum of 7 channels at a time, the actual number depending on how connections the user already has active. This is, because from the server's point of view, the user and the agent are the same person.

Every minute the agent examines how much activity has occurred on the channel. If the channel activity is below a certain level the agent leaves the channel and moves on to new channels. Every five minutes the agent examines the words that have been said on the channel and determines if the channel is of interest to the user, if it has not found anything by this time it moves on to a new channel. Once the agent has visited all of the channels it will limit itself to the most active channels and only move about within this subset. This is because in most networks most of the activity is limited to a small subset of the total number of channels. In this subset the activity is considerably greater than outside the high activity subset.

## 4 RESULTS

The Bombyx system was developed as an open source project on SourceForge [6]. Over the three months of project development, there were approximately 125 downloads of the Bombyx Client, validating both our approach and our inference that there was a need for this type of software. Figure 8 shows the usage statistics for the past 51 days. The upper line shows the number of website hits the Bombyx Development SourceForge site [1] received per day and the lower line denotes the daily number of Bombyx Client downloads requested. Although we did not ask users to provide feedback, some offered their comments. Those who offered feedback were generally pleased with the results and appreciative of our efforts. Their suggestions also provided us with valuable direction and suggestions on how to improve our interface and make it more useful.

## 5 FUTURE WORK

There are still many areas that could benefit from additional enhancements.

Our original concept for the user interface was to use a "chat canvas" that the user could use to control the conversation. The canvas would present the conversation room in a spatial way; with the user able to group people she would like to talk with near themselves, and others farther away. This would allow users to easily focus on conversations and people, seamlessly dispersing from areas of interest outward to areas of less interest.

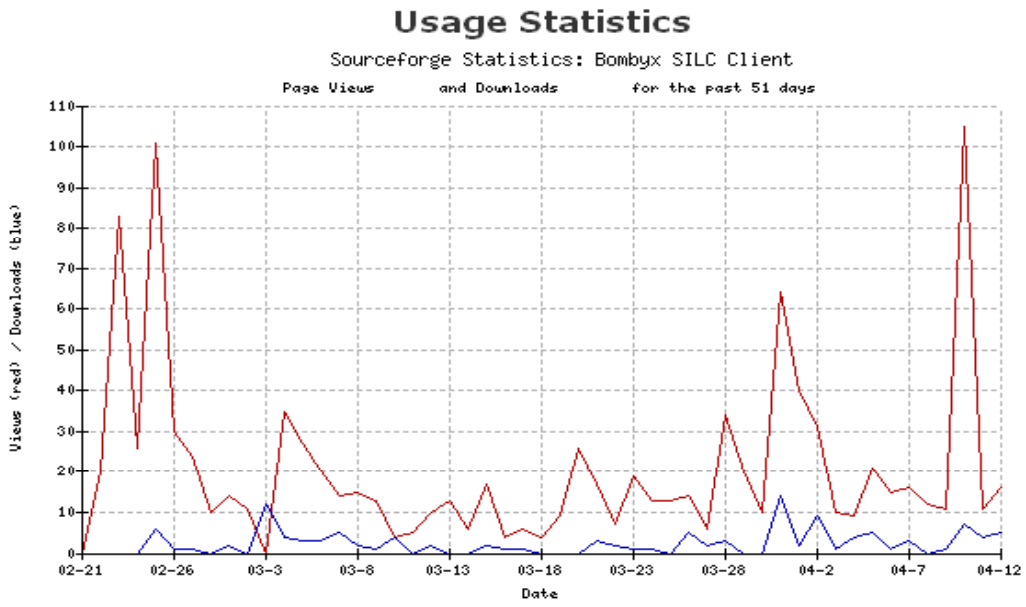


**Figure 8: Daily Page View and Download Frequencies for the past 51 days. The upper line denotes the number of page hits and the lower line denotes the number of downloads**

We moved to the Focused Chat interface due to its design simplicity and to facilitate development. Its simplicity does provide an argument for it as a viable opponent to our original design idea; however, it would be interesting to implement the Chat Canvas approach so that it could be compared to the Focused Chat implementation.

Currently, the finder agent uses a naïve word-matching algorithm. This provides useful results; however using a more sophisticated matching algorithm could enhance the agent's capabilities. One such method, described in [8] suggests using an algorithm for empirical learning of term-categorization associations. Using such an approach the agent would build up a database associating words to their categorization based on how users make these categorizations. This would allow the agent to find a broader range of results for a given query.

An area that also needs addressing is user testing. Although the informal response we have received so far has been good, it is still necessary to do a formal evaluation of the system.

We would like to study the implications (both security and moral) of anthropomorphizing agents. For example, can people tell the difference between an agent and a person acting as an agent? This, in a way, is the opposite of the Turing test, and we suspect that people cannot tell the difference. However, only further experimentation can confirm this suspicion.

## 6 CONCLUSION

This paper presented an improved interface for online chat room communication. Predominately, it combated three criticisms of traditional chat interfaces: their cumbersome command driven user interfaces, their vast lists of ill-named channels, and the unstructured and unmanageable conversation that occurs within conversations.

The first of these issues is resolved by providing a simple clean graphical interface that does not require users to have any knowledge of IRC commands. Support for the commands is implicitly provided, however, so as to not turn off experienced users.

The user may use Focused Chat to narrow the conversation space as needed. It differs from private chat in that the conversation still occurs in the public channel. However, instead of concentrating on the entire channel and having to visually disregard messages from participants in which the user is uninterested, the focused chat provides a mini-viewer that only contains the messages from participants the user has specified as interesting.

Finally, a Finder Agent is presented to help the user locate channels of interest. As channels evolve over time their names often become inappropriate making it difficult for new users to locate the channel. This forces users to join channels haphazardly, and then wait on them to determine whether they are interesting. The Finder Agent performs this task for the user. The user tells the agent what keywords to look for, and then the agent carries out the task of connecting to channels, listening for the keywords, and determining when he should move on to try a new channel. As the agent finds channels of interest it reports them immediately to the user, but continues to check for new channels until the user tells it to stop.

The original design of the Finder Agent presented an anthropomorphic agent that communicated with the user through the chat interface. This approach, however, demonstrated a security risk as it became apparent that having agents (which the user would generally perceive as personal and trustworthy) and humans (who most definitely cannot be trusted) communicate with the user through the same interface would result in confusion. Although, humans can generally distinguish an agent from a human, they cannot easily distinguish a human (posing as an agent) from an actual agent. Thus, we decided that this was one domain where anthropomorphic agents were definitely a bad idea, and accordingly designed the Finder Agent with a non-anthropomorphic search interface.

## REFERENCES


1. Bombyx Development Web Site. Online at http://bombyx.sourceforege.net.
2. Donath, J., K. Karahalios and F. Viégas. "Visualizing Conversation". Proceedings of HICSS-32, Maui, HI, January 5-8, 1999.
3. Mayhew, M. "The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design." Morgan Kaufmann Publishers, 1999.
4. mIRC Intro to IRC Web Site. Online at http://www.mirc.com/irc.html
5. Networks of the IRC. On line at http://netsplit.de/networks/
6. SOURCE FORGE Web Site. Online at http://sourceforge.net.
7. Van Dyke, N., H. Lieberman and P. Maes. "Butterfly: A Conversation-Finding Agent for Internet Relay Chat". Proceedings of the 1999 International Conference on Intelligent User Interfaces, January 1999.
8. Yu, E., P. Koo and E. Liddy. "Evolving Intelligent Text-Based Agents" Fourth International Conference on Autonomous Agents, Barcelona, Spain, June 3-7, 2000.