

ChatVis: A Visualization Tool for Instant Messaging

Yaser Ghanam and Sheelagh Carpendale
Department of Computer Science
University of Calgary, Canada T2N 1N4
{yghanam, sheelagh}@ucalgary.ca

Abstract. This report presents ChatVis as a 2,800 LOC tool for visualizing activities in instant messaging conversations. The tool visualizes at four different levels, starting from the conversation level down to the message level, the word level and finally the character level. The tool provides means to visualize the typing speed and rhythms of an arbitrary number of conversers. Additionally, it attempts to visualize intangible characteristic of a typed message such as hesitance (uncertainty) and emphasis. ChatVis allows for mouse as well as keyboard interactions to give the user a variety of filtering and zooming options. The system state is always saved allowing for animation through temporal aspects of the conversation. Moreover, the conversation can be persisted to XML and loaded back on demand for comparison purposes amongst different people.

1 Purpose

The main purpose of this visualization, hereafter called ChatVis, is to augment activities in instant messaging conversations. This augmentation is attempted by including evidences of the theme, the tone and the length of the conversation as well as some manifestations of the personalities of conversers. The goal behind ChatVis is to make it possible for an observer to have a quick insight of what the conversation was all about, how long it lasted and how each of the conversers behaved in the conversation in terms of emphasis, hesitance, patience as well as typing speed and rhythm.

Figure 1 shows the output generated by ChatVis to visualize a sample conversation between User1 and User2. Later in this paper, every component of this visualization will be explained in detail. This visualization may look like CrystalChat [1], but the purpose of ChatVis, data mapping and visualization variables are completely different. While CrystalChat focuses on the personal chat history, ChatVis is interested in the visualizing instant messaging at the conversation level.

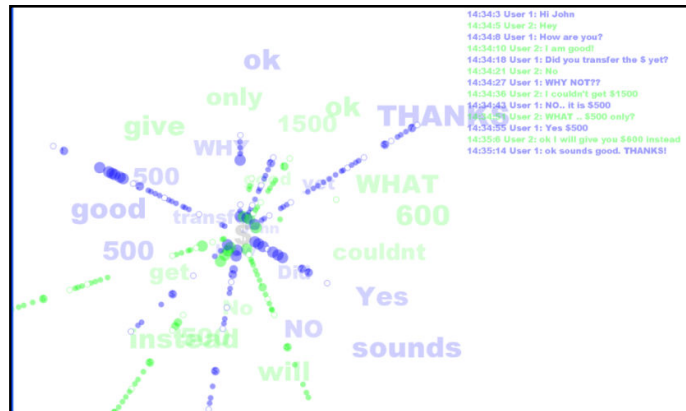


Figure 1 - A sample visualization produced by ChatVis

2 Overview

On a high level, as illustrated in Figure 2, the visualization is composed of four areas; each serves a specific purpose as follows:

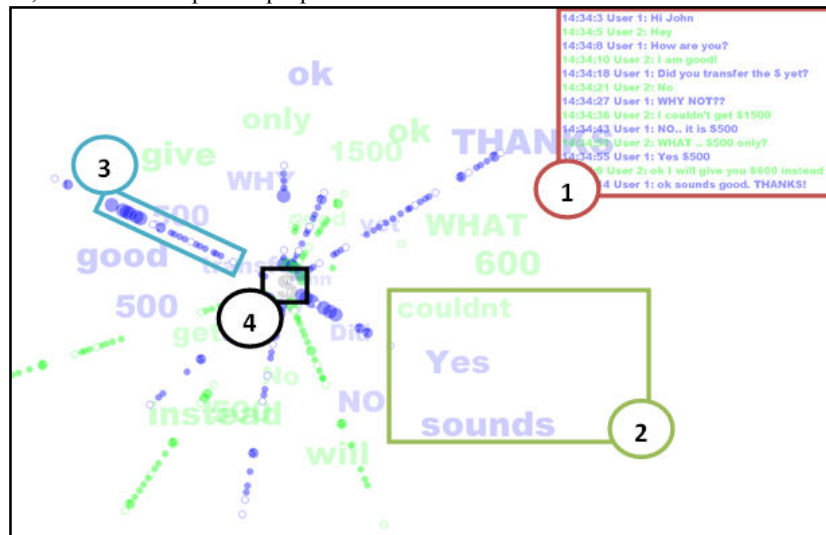


Figure 2 - The four high level areas of ChatVis

1. *The chat backlog area:* The box contains the original lines of conversation of all participants in a chronological order. This is an essential component that gives the viewer the option to look at the original textual content of the conversation.
2. *The words area:* This area consists of the “uncommon” words in the conversation distributed in a formal fashion as will be explained later. A word is “uncommon” if it does not belong to the set of common words (such as I, you, are, is ... etc). The set of common words is defined and can be changed by the user.

3. *The keystrokes area:* This area contains the keystrokes of all typing activities in the conversation. Each straight line of keystrokes represents one message in the chat box (area 1). The orientation of these lines and the distribution of keystrokes over a given line are both visual variables that represent specific aspects of the conversation as will be detailed in the following sections.
4. *The theme area:* This is a relatively small area that is basically a circularly shaped hole in the middle of the keystrokes area. It contains a single character (like '\$', '?', '!'... etc) that indicates the main theme of the conversation. That is, it gives an insight whether the conversation was mainly question-answer oriented, exclamation-intensive or even money-related.

3 Scenario

ChatVis supports as many participants as required. When the application first runs, it gives the user the option to either start a new conversation, or look at a previously persisted one as illustrated in Figure 3.

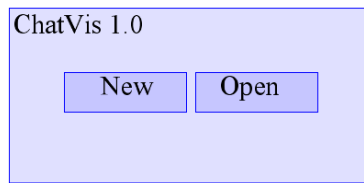


Figure 3 - Start a new conversation or open an old one

Let's assume the user chooses to start a new conversation. Then the user will be asked to input the number of participants in the conversation as shown in Figure 4.

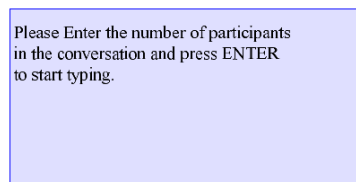


Figure 4 - User specifies the number of participants

Each participant is assigned a color. Then, when the white empty screen appears, the program is ready to receive contents. The users are expected to chat as they would in a typical instant messaging application. In this mock-up version of ChatVis, the chatters use one keyboard, and therefore pressing ENTER signals a switch of turns between participants. For instance, when User1 types "Hello Jenny" and presses ENTER, his input will be visualized momentarily as shown in Figure 5. First, while User1 is typing, keystrokes (representing the actual key hits on the keyboard) appear in the keystrokes area. When User1 presses ENTER, the backlog area will be updated with the message along with its timestamp. Also, any uncommon words will be displayed in the words area. The theme area will not be affected as the message "Hello Jenny" did not include any punctuation or special characters. All drawn objects on the screen will take the same color as User1's (Green in this example).

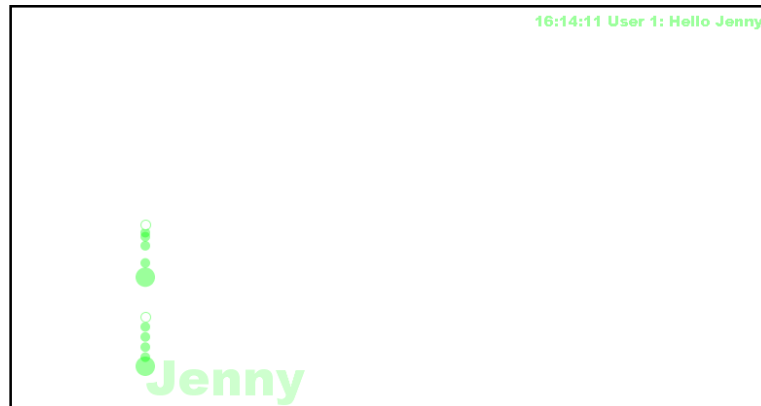


Figure 5 - "User 1: Hello Jenny"

Because User1 hit ENTER, it is now the turn of User2 to respond. Suppose User2 types "Hi How are you?" The effect of this message will be as shown in Figure 6. The same process will take place (keystrokes, message and uncommon words will be shown) in the unique color of User2. In addition, in this case, a question mark will appear in the middle as an initial indication of the theme of this conversation. Moreover, any previously drawn words (such as "Jenny") will fade out and get smaller in size to indicate older age.

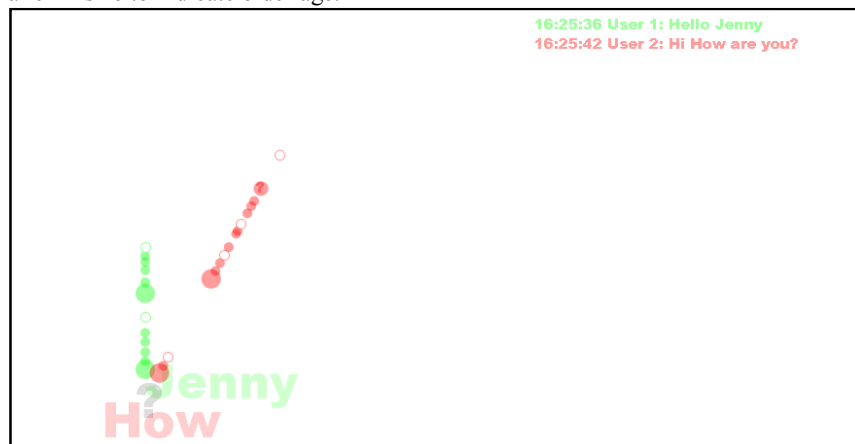


Figure 6 - "User 2: Hi How are you?"

4 Data Mapping

4.1 What is to be visualized?

In the design of ChatVis, the following points have been considered when thinking about what needs to be visualized in a conversation. The rule of seven was considered when choosing what to visualize so that the viewer is not overwhelmed with many visual variables.

1. A conversation consists of a list of chronologically ordered messages.

2. Every message in the conversation is to be visualized with its original textual content and timestamp (time when it was submitted).
3. A conversation may have a theme that can be extracted from the punctuations and special characters used in the messages.
4. Each message consists of a set of words. Some of these words are common words and need not be visualized; whereas other words should be visualized to indicate the topic of the conversation.
5. Each message consists of a series of chronologically ordered keystrokes.
6. The visualization of keystrokes needs to address the following:
 - a. Length: Number of characters and number of words (separated by spaces).
 - b. Typing rhythm: the time when the keystroke occurred.
 - c. Emphasis: capital letters in the message.
 - d. Hesitation: backspace and delete keystrokes.
 - e. Theme: the theme of the individual message represented using any punctuations or special characters in the message. Notice that this is different from the overall theme of the conversation.

4.2 Message Visualization

It is of high importance to visualize the message itself so that the viewer can always refer back to the original dataset when looking at the drawn words and keystrokes. Hence, the viewer will be able to compare different messages and know exactly how the different components of the visualization relate to the original submitted message.

This visualization takes place after the whole message is submitted. When the user presses ENTER, the message appears in the chat backlog area as shown in Figure 7.



Figure 7 - Message visualization

The visualization component of the message consists of three parts: the timestamp (16:25:42), the message owner (User 2), and the textual content (Hi How are you?). The color of the component is used to associate the message to its owner (color will be discussed later). Whereas the vertical position of the message indicates its order. For example, in Figure 6, “Hi How are you?” was submitted after “Hello Jenny” and therefore it should have a larger value for the position on the y-axis. This decision was deduced based on the convention used in all instant messaging applications.

4.3 Word Visualization

Visualizing words has a specific purpose of conveying what topics the conversation is mainly about. For instance, through a quick look at the visualized words in the words area, if words that pop out include “war”, “Iraq”, “oil”, “US”, “soldiers”, one can quickly conclude that the conversation was about war in Iraq. Nevertheless, words in messages are not always key. Common words such as “are”, “we”, “I”, “is” and many others may not directly convey any special meaning. For this reason, the implementation of ChatVis makes it possible for the users to exclude the set of words they perceive as common. Any word that belongs to this set will not be visualized in the words area. In the current version of ChatVis, we considered a list

of common words some of which are amongst the most 100 common words in English according to [2].

Visualized words are distributed in the words area in a phyllotactic pattern in order to: 1) maximize the utilization of space with a minimal overlap, and 2) provide an insight on how old the word is relative to when the conversation has started (i.e. the closer to the center, the older the word). However, given the diverse lengths and sizes of these words, this pattern might not be obvious at times. The second visual variable in visualized words is color which is the same as the color of the message owner but with a smaller value.

Since the position of the word has been used as an indicator of its age, the limited space of the words area was a challenging obstacle. That is, when the pattern starts to exceed the boundaries of the words area, the phyllotactic pattern will repeat again resulting in two problems: newer words will overlap with older ones, and the position as a visual variable will be distorted since new words are being drawn close to the center. In order to overcome the first problem, the whenever the phyllotactic pattern repeats, it does with different parameters to decrease the overlapping. To resolve the second issue, a fading out effect was added to older words using both the transparency value along with the size of the word. That is, the older the word is in the conversation, the more transparent and smaller it becomes. As illustrated in Table 1, we can see the word “smoke” at different times of the conversation. The fading out effect is a realistic metaphor that relates to what happens in real life to older things, as they get paler with time.




Table 1 - Fading out effect



Smoke	Smoke	Smoke
T = 0	T = 10s	T = 20s

4.4 Keystroke Visualization

Each hit on a keyboard results in a keystroke. A keystroke is represented using an ellipse that has different characteristics according to the key it represents. A keystroke has three visual variables, namely: size, color and position. The position of the ellipse represents the time it was generated by the user relative to the time when the user has started to type the message. The farther the ellipse is from the center of the keystrokes area, the more time separation is between the keystroke and the start time of the message. Table 2 illustrates the size and color mapping of each type of key in ChatVis.

Table 2 - Keystroke mapping

Char Type	Interpretation	Examples	Mapping Metaphor	Visual variables
Regular	n/a	Numbers and all small alphabetic such as 1,2,3,a,b,c		Size: small Color: message creator's color
Capital	Emphasis	All capital alphabetic such as A,B,C		Size: large Color: message creator's color
Special	Theme	Punctuation and special chars such as ?,!,\$,@,*		Size: Medium Color: message creator's color with the special char appearing in the center

Space	No of words	Space and Enter		Size: small Color: white filling with a stroke of the message owner's color
Backspace	Hesitance	Backspaces		Size: small Color: black filling with a stroke of the message owner's color

It is noticeable from the table above that in all cases an indication of the message owner has been maintained in the mapping metaphor of the keystroke. Whether it is the filling color or the stroke color, the color of the message owner appears in the keystroke to make cross-referencing and association amongst different visualization components more intuitive. In the case of special characters, the character itself is drawn in the center of the ellipse to give an indication of the theme of the message. For instance, if we see a question mark in a message, we can immediately spot that this message was basically a question or had a questioning flavor.

In instant messaging applications, the lack of the ability to express emotions through facial expressions and body gestures is substituted by some typing behavior. For example, if someone needs to emphasize a point or express an exaggerated exclamation, they would capitalize all the letters in the message such as “ARE YOU SERIOUS??” We visualize such behavior in ChatVis using a larger sized ellipse that conveys the emphasis factor.

The reason behind using the white filling for spaces is to make it easier for the eye to locate spaces in the message without a cognitive burden. For one, the white ellipse will outstand in the message. And secondly, it is a realistic metaphor of what is called a “white” space.

In our visualization, we interpret backspaces as hesitance in writing the message. This is not necessarily true all the time. But an excess presence of backspaces in one message may strongly indicate uncertainty in the converser's intention of what to write or how to write it. The black color has been chosen as a filling of the ellipses representing backspaces for that it gives the hesitance factor a pre-attentive characteristic [3].

Example:

To illustrate the visualization of keystrokes as described above, let's take this example (← means backspace): User1 writes: “I will surely←←←←← be there” The visualization will be as in Figure 8.



Figure 8 - Example

The first impression we get when looking at the keystrokes visualization is that the creator of the message is User1 because his color is grey. We can also immediately spot that the creator of the message hesitated about something. Pre-attentively, we can spot the six black dots that represent the backspaces. Also, by looking at the white ellipses, we can count the number of words in the message which is four. Notice that in the message visualization the word “surely” doesn't appear because it was deleted.

The distances between keystrokes represent time intervals between key hits on the keyboard. These distances eventually reflect the typing rhythm of the user. Every 1 sec interval is visualized as approximately 60 pixels on the screen. The number of keystrokes the screen can accommodate is dependent on how large the canvas of the application is set to be. But obviously, this can be a limitation when users tend to use

long sentences. A solution to this would be an automatic adjustment of the scaling factor when the sentence gets too long to be accommodated within the screen boundaries.

Typing rhythms can be a very expressive and probably unique technique to reflect the personal touch on messages. For example, the grey (a) and the blue (b) lines of keystrokes in Figure 9 are visualizations of the same message typed by the same user twice. Notice the similarity of the typing rhythm between the two.



Figure 9 - Typing rhythm of the first user

When another user was asked to type the same message twice, here are the two visualizations ChatVis produced:

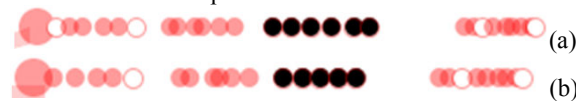


Figure 10 - Typing rhythm of the second user

We can notice that the second user types faster (closer ellipses) and has a different transition pattern over the keyboard compared to the first user.

4.5 Theme Visualization

In ChatVis, a simplistic approach is followed in order to deduce the theme of the conversation. The frequency of each special character or punctuation is recorded. The character with the highest frequency is chosen as the theme of the conversation. For example, if '?' appears 10 times in the conversation whereas '!' appears 4 times, then the '?' will be selected to reflect a question-answer type of conversation. Also, characters like '\$' may give a good indication of what the conversation is about (i.e. money).

The theme of the conversation appears in the center of the keystrokes area. It always has a grey color with a transparency effect to make it less interfering with the keystrokes area. Initially, the size of the theme reflected the frequency of its occurrence in the conversation. However, because the theme started to overlap in a disturbing manner with the surrounding keystrokes, a decision has been taken to give it a fixed size so that it stays in the empty area at the center. Figure 11 shows an example of a conversation full of questions. The user has the option to show/hide the theme by pressing on the F3 button.

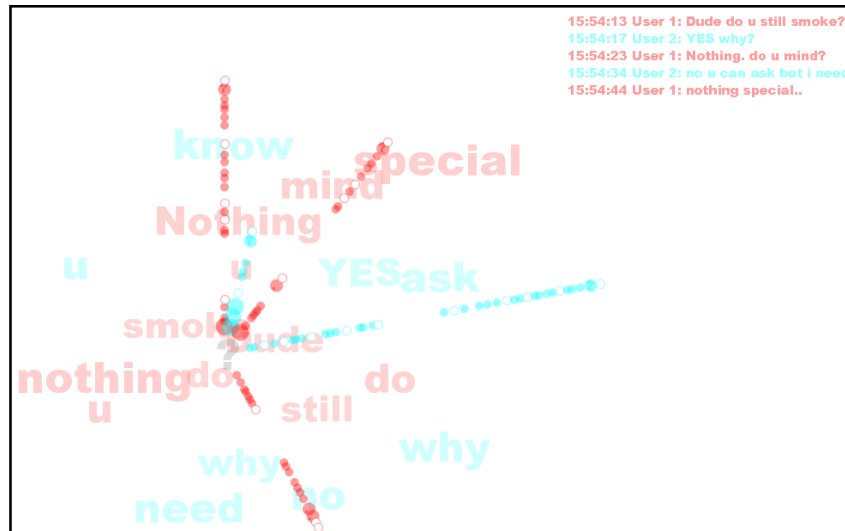


Figure 11 - The theme

4.6 Visualizing Time

As illustrated previously, a series of keystrokes on a straight line represents a full message by one of the conversers. This line of keystrokes is not always horizontal or vertical; but rather it can have an angle with the y-axis depending on the time the message was started relevant to the time when the conversation had started. That is, the lines of keystrokes start at an angle of 0 degree with the y-axis, and this angle starts to increase as time passes in the conversation. This design decision has been made to imitate the motion of the arms in the analog clock. The angle between two consecutive messages initiated by different authors indicates how long the first author has waited before he got a response back from the second – we call this waiting time “*patience*”. Every 1 sec is visualized as 2 degrees allowing for a 3-minute conversation before the circle starts to repeat. This scaling can be easily changed by manipulating a variable in the source code to accommodate longer conversations.

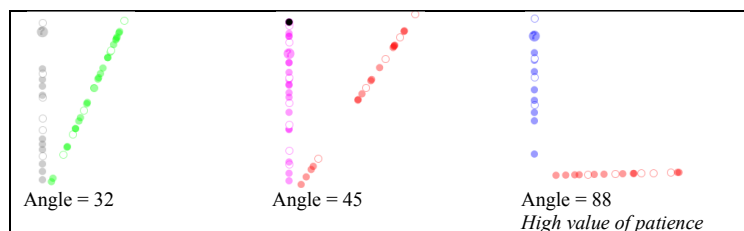


Figure 12 - Visualizing patience

The decision of using the clock analogy in visualizing time introduced a limitation of how many messages the circle can accommodate before it gets full (or repeats). One solution is to provide a new circle whenever needed while keeping the old ones visible in thumbnail versions. This solution is not evident in the current version of ChatVis due to time limitation.

4.7 Color Utilization

ChatVis uses color as an indication of the message owner (the participant who typed the message). Each participant possesses a unique color. This color will be used across all components that relate to that participant in the visualization. In an ideal design of ChatVis, each user will be able to pick their favorite color. Nevertheless, in this version, participants are assigned colors on a controlled random basis. The control of the randomness is introduced in order to choose colors with high contrast with the white background. These colors include blue, red, green, grey, cyan and magenta. The design is flexible enough to add more colors to this set.

The decision of choosing color, as a visual variable, to represent different participants was based on the following reasons (based on Bertin [4]):

1. Color allows for association amongst different components. That is, just by looking at the word “How” in Figure 6, one can tell that it is associated with User2 whose color is red.
2. Color has a theoretically infinite length. Even when considering the practical length of color as a visual variable, its length is sufficient for an application like ChatVis where the number of participants in a certain conversation is very limited.
3. Color works well with both points (keystrokes) and lines (associations).
4. Since the order of participants does not matter, there is no need to choose a visual variable with ordering capability. Thus, the use of color to distinguish different (but not ordered) participant is appropriate.

Also, color is used in ChatVis as an indication of separation of words with spaces or hesitation in typing as described in the previous section. The black color on a white background has a pre-attentive characteristic that helps the viewer quickly spot backspaces in the message.

5 Interaction

ChatVis provides different interaction techniques that can fall under the categories of selection, elaboration, filtering and connection [5]. The interaction decisions were made based on the three-level visualization model suggested by [6]:

- Overview
- Zoom/Filter
- Detail-on-demand

5.1 Overview:

Originally, the user has an overview of what is going on in the conversation. He can see the visualized components simultaneously in the four different visualization areas (chat backlog, words, keystrokes and theme). The user can interact with the visualization to see the connections between the different components in the graph. By moving the mouse over a word, the user will be able to identify to which message this word belong to and which line of keystrokes represents this message as shown in Figure 13.

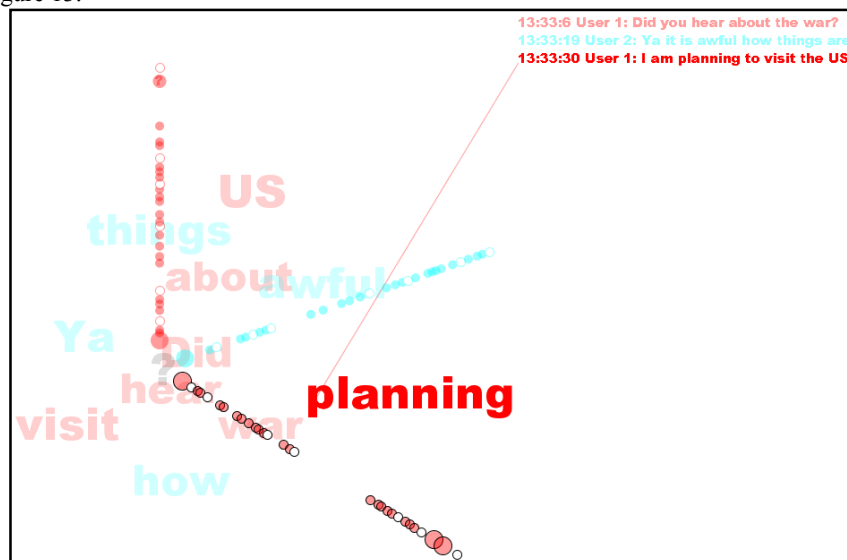


Figure 13 - Visualizing associations

5.2 Filtering:

Then, the user has the option to filter some components as shown in Figure 14. For example, if the user presses F2, the chat backlog area will disappear. The user can retrieve it back by pressing on F2 again. Similarly, if the user doesn't want the words to be visualized, pressing F1 hides all the words and pressing it again shows them back. In the same fashion, the F3 button can be used to control the appearance of the theme. These options are given to the user so that he is capable of filtering information he deems unnecessary or less important. Also, the user may choose to hide some components so that he can focus more on other details in the visualization. Figure 14 shows how the messages area has been hidden.

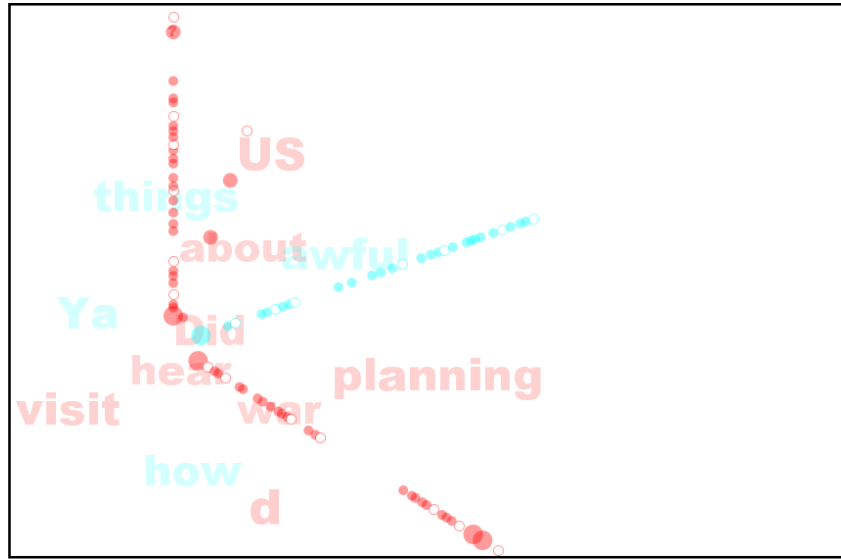


Figure 14 - Filtering some components

5.3 Detail-on-demand:

Besides being able to have a high level overview of the whole visualization as well as being able to filter out some of the visualized components, the user can also choose to go to a deeper level of detail. If the user wants to investigate a specific message to see how each ellipse map to the original character, he can do so by clicking on a word in the words area. The corresponding line of keystrokes will be magnified with annotations being added to give information about the message as shown in Figure 15. In addition to the magnified line of keystrokes that shows a direct mapping between ellipses and characters, information about the message will also be given including:

1. The final status of the message (deleted characters are not shown)
2. The creator of the message.
3. The length of the message in terms of the number of words and number of characters.
4. When the message was started.
5. When the message was submitted.
6. The typing speed in char per second.
7. The hesitation factor if any.
8. The patience factor if any.

The user can drag the line of keystrokes and drop it anywhere in the canvas in order to be able to have the most convenient positioning.

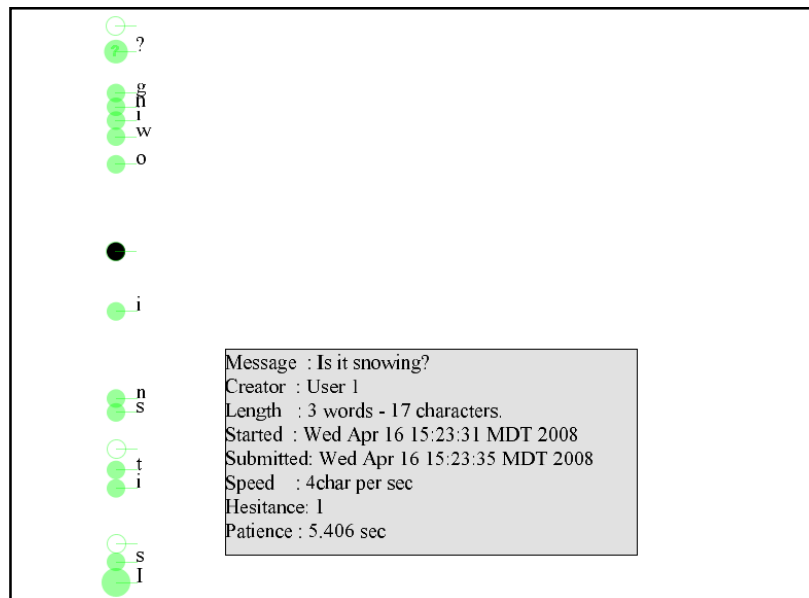


Figure 15 - Detail-on-demand

5.4 Mouse interactions:

- Mouse-Hover: when the mouse is moved over any word in the visualization, this word will be highlighted (given a larger color value) and all associations connecting this word to other components will be shown.
- Right-click: when the user clicks on any white space using the right button of the mouse, all highlighting will be removed.
- Left-click: when the user clicks on a word, more details will be shown about the corresponding message, and the line of keystrokes will be magnified.
- Middle-click: when the user wants to reset everything and start from the beginning, he can click on the middle button of the mouse.
- Drag-and-drop: when the user is in the investigation mode (where the line of keystrokes is magnified), he can drag the line of keystrokes and drop it anywhere within the canvas.

5.5 Keyboard interaction:

- Input: when the program first starts, the user can input the number of participants in the conversation.
- ENTER: pressing this button after typing a message terminates this message and updates the conversation and all related visualization components.
- F1: as long as all messages are terminated, this button shows/hides the words area.

- F2: as long as all messages are terminated, this button shows/hides the chat backlog area.
- F3: as long as all messages are terminated, this button shows/hides the theme.
- F4: as long as all messages are terminated, this button persists the conversation to an XML file named: “convN.xml” where N is an index given by the program. The destination folder is always c:\ and is easily changeable in the source code.

6 Saving System State

6.1 Run-time Save (animation & efficiency):

At any point of time, the system state is guaranteed to be fully saved. This saving process is automatic and is triggered by any change in the system state. Saved elements include all data object models in the conversation. That is, all messages, words and keystrokes are saved along with their attributes of time and associations in hashing-capable collections that allow for efficient random access. Moreover, the state of any drawn objects on the screen is also saved to allow for animation, zooming & filtering capabilities as well as efficient redraw.

6.2 XML Persistence (analysis & comparison):

The user has the option to save the state of the system at any point of time by pressing F4. This will persist the whole conversation in a human-readable XML file that looks like the one in Figure 16. The user can open this file on demand through ChatVis to analyze the conversation. Although ChatVis does not currently support loading more than one conversation at a time, comparison can be achieved by running multiple instances of ChatVis and loading different XML files.

```

- <Conversation>
  <startTime>1208444864328</startTime>
  <endTime>0</endTime>
  - <participants>
    - <Participant>
      <id>0</id>
      <nickname>User 1</nickname>
      - <color__p>
        <red>0</red>
        <green>0</green>
        <blue>255</blue>
        <alpha>100</alpha>
      </color__p>
      <active>false</active>
    </Participant>
    - <Participant>
      <id>1</id>
      <nickname>User 2</nickname>
      - <color__p>
        <red>255</red>
        <green>0</green>
        <blue>0</blue>
        <alpha>100</alpha>
      </color__p>
      <active>true</active>
    </Participant>
  </participants>
  - <messages>

```

Figure 16 - XML persistence

7 Design Modularity and Flexibility

ChatVis was designed in a modular fashion following a Model-View-Controller pattern. All data objects are stored in complete separation from the view objects and the controllers to allow for additional possibilities of different visualizations of the same dataset without having to infer changes in the object model. All parameters and scaling factors are coded so that they can easily be changed to examine the best combination of parameters for the visualization. Also, ChatVis provides support for as many participants as required and provides easy methods to add colors and common words into the consideration of the visualizing engine.



Figure 17 - A three-participant conversation

References

1. Tat, A., Carpendale, S. (2006) CrystalChat: Visualizing Personal Chat History, *International Conference on System Sciences, 2006. HICSS '06. Proceedings of the 39th Annual Hawaii*, vol.3, no., pp. 58c-58c.
2. The First 100 Most Commonly Used English Words, available at <http://www.duboislc.org/EducationWatch/First100Words.html>, last accessed April 17, 2008.
3. Ware, C. (2004). *Information Visualization: Perception for Design*. Morgan Kaufmann, 2nd edition.
4. Bertin, J. (1967) *Semiology of Graphics: Diagrams, Networks, Maps*. Translated by W. J. Berg. University of Wisconsin Press, 1983. Cited from the slide set #9 by Carpendale, S. (2008).
5. Ji Soo Yi, Youn ah Kang, Stasko, J., Jacko, J.A. (2007) Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*, vol.13, no.6, pp.1224-1231.
6. Shneiderman, B. (1992) *Designing the user interface: strategies for effective human-computer interaction*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.