

Comic Chat: From Research To Product

David Kurlander
 Microsoft Corporation
 One Microsoft Way
 Redmond, WA 98052 USA
 djk@microsoft.com

ABSTRACT

Comic Chat began as a project at Microsoft Corporation's new research labs. Our goal was to build a new kind of chat program, that automatically enhanced a textual chat conversation by adding graphics, using conventions adopted from the comics domain. As people type text into Comic Chat, comics panels appear. The program uses traditional comics illustration rules to determine which characters should appear in each panel, how to position them, and a default gesture and expression for these characters. It also computes word balloon outlines that look hand-drawn, and lays out these balloons appropriately in each panel. Comic Chat also decides when to break from one panel to the next, and the zoom factor for each panel.

People find Comic Chat appealing, and it has become a successful Microsoft product. It has shipped millions of copies and been localized into 24 different languages. Here we describe many of the research issues surrounding Comic Chat, and some of the lessons that we learned turning the research prototype into a product.

Keywords

Comic Chat, chat programs, automated illustration, comics, virtual worlds, user interfaces, productization.

INTRODUCTION

Textual chat programs originated in the days before graphical displays, fast computers, and high-bandwidth networks. It is worth considering how such early applications can be enhanced using today's technology.

At Microsoft we have developed a new kind of chat program based on a comic strip representation. This program, called Comic Chat, automatically creates a visual representation of conversation, using techniques borrowed from professional comic artists. Graphical chat programs are becoming increasingly popular, in part because people can choose a visual representation for themselves, and communicate using gestures and expressions in addition to typed text. Habitat pioneered the concept of graphical chat programs [3].

Comic Chat has a number of advantages over other graphical chats. One is the availability of a *graphical history*. In other graphical chat programs there is no graphical representation of the events that have occurred. In some graphical chat programs, such as Worlds Chat [5] or The Palace [4], the text appears on the screen briefly. In others, like V-Chat [2], there is a separate

textual transcript of the conversation that can be perused later. However, if the graphical interactions are worth performing, they should also be included in the history. Comic chat has a natural panel-based history, and the complete interactions can be viewed and saved at any time.

Another advantage of the comics representation is that it can be dynamically varied. The characters chosen for each panel, the zoom factor, and the incorporation of other panel elements can be frequently changed, creating a dynamic visual representation. In contrast, other graphical chat programs always show a static representation of the entire room, or have a virtual camera fixed to the user.

A third advantage of Comic Chat is that much of the communications is automated. People need not wander through a three dimensional world, looking for others to talk to. They need not position their visual representation so that all other conversational partners are within their field of view, and make sure that nobody accidentally wanders in between. It is not necessary to explicitly choose a gesture or expression - Comic Chat chooses good defaults, by examining the text of the conversation.

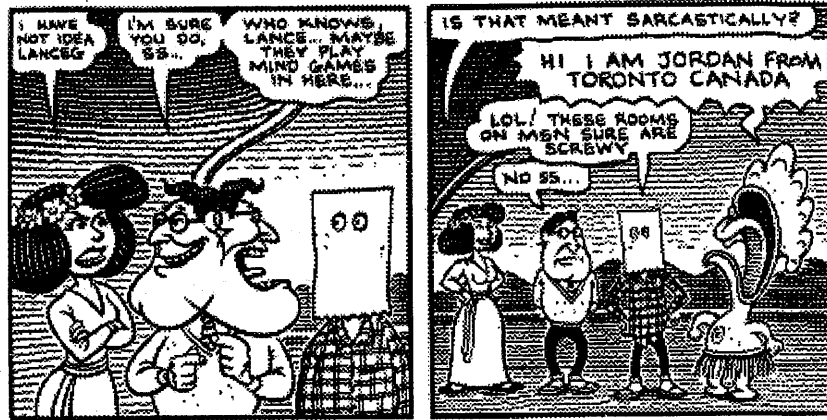
Comics also work well as a representation for textual chat conversations, because comics were created in part to represent interactions among multiple characters. The comics form has an incredibly rich vocabulary, which we are even now just beginning to exploit. People also find comics appealing, and they enjoy reading the panels produced by Comic Chat.

METHODOLOGY

It was apparent very early in the project that we needed to work closely with a professional comic artist, to learn rules and heuristics for representing conversations in comics form. We contacted Jim Woodring, one of the very best American comic artists, and fortunately he was willing to work with us. We collected textual chat transcripts, annotated them with information that we believed a computer would be able to extract, and asked Jim to produce comics of the conversations.

The hand-drawn prototypes that Jim drew were wonderful. They were visually interesting, and they had a richness that was lacking in the original text transcripts. Two panels from the original hand-drawn prototypes appear in Figure 1. Our goal became

Figure 1. Two panels from the original artist-drawn prototypes.



automatically representing conversation in a manner similar to these prototypes – but we still did not know what processes went on in the comic artist’s mind in order to produce these drawings. We needed to “reverse engineer” the artist, encapsulating some of his skills in computer code.

In reviewing the artist’s prototypes, we determined that there were many elements that needed to be automated, and these could be classified into three categories: *characters*, *balloons*, and *panels*. The next three sections briefly describe the automation that Comic Chat performs in these three categories. A more complete description appears elsewhere [1].

CHARACTERS

Chat rooms can contain many more members than can appear in any one comics panel. So one of the first decisions that Comic Chat needs to make is which characters to include in the panel that it is composing. Comic Chat makes sure to include in each panel those people that are speaking. It also attempts to include their conversational partners. Users can indicate whom they are talking to by selecting from a list of all people in the room, or by clicking on the appropriate conversational target in an earlier panel. Conversation can be directed towards multiple people at once. Everyone in the room sees these panels, unless the message is sent as a *whisper*, in which case only the recipient sees the message.

Next Comic Chat needs to determine how to place these characters in the panel. It uses the following rules:

1. Characters should face their conversational partners.
2. Characters should be positioned next to their conversational partners.
3. Characters should not move needlessly.
4. When it is unknown which characters are speaking to each other, position them so that it appears that they are having a conversation anyway.

Gestures and expression in Comic Chat are inferred from the text, but users are free to override this inferred pose by explicitly specifying another one. To infer gestures and expressions, we use simple word-spotting

rules created especially for the chat domain. To determine these rules, we reviewed a large number of textual chat transcripts. There are several categories of rules:

1. **Emoticons:** for example :-) and :-(indicate happy and sad.
2. **Chat acronyms:** such as LOL (“Laughing out Loud”), IMHO (“In My Humble Opinion”).
3. **Typesetting:** all caps indicates shouting.
4. **Punctuation:** !!! indicates shouting.
5. **Greetings:** The character waves if the user types “Hi”, “Hello”, “Goodbye”, “Welcome”, “Howdy”, etc..
6. **Self-references:** When the user refers to himself (“I” at beginning of sentence, “I’ll”, “I will”, “I’m”, “I would”, “I’d”, etc...) his character points to itself.
7. **Other-references:** When the user refers to his conversational partner (“You” at the beginning of a sentence, “are you”, “will you”, “did you”, etc..), his character points at the partner.

If the user would like to override the system’s choice of gesture and expression, this can be done via the *emotion wheel*. The emotion wheel is similar to a color wheel, except different emotions appear spaced along the perimeter of the circle rather than different hues. At the center of the wheel is the neutral expression, and as the cursor moves from the center of the wheel to the periphery, the emotion/gesture combination become more and more extreme. With a single mouse movement, the user can specify not only an emotion, but an intensity. The emotion that users choose accompanies the next text message that they send, or they can send an expression without accompanying text.

To render the characters, Comic Chat typically combines an appropriate facial expression with an appropriate body pose. By composing heads and bodies, we achieve a much larger combination of renderings, than we could with single monolithic head/body poses. We also render each character with a halo around it. This traditional comics technique helps to separate the character from the background, making the panels easier to read.

BALLOONS

The second class of automated tasks in Comic Chat relate to balloon construction and placement. Comic Chat can generate four different types of balloons: speech balloons, thought balloons, whisper balloons, and narration boxes. The user can specify a type of utterance, and the appropriate type of balloon is added to the panel.

Different comic artists use different balloon styles. We emulated the balloon style used by Jim Woodring, since the balloons wrap around text, making them look more hand-drawn. Comic Chat wraps beta-splines around the outside of the text according to a complex set of rules for following the text outline. Balloon tails can be either arcs or lines. The result is very natural, and does not appear to most people as computer generated at all.

Balloons, like characters, must be positioned in panels. Balloons tend to be read in an up-down, left-right fashion (at least in the Western comics tradition), and we developed an algorithm to place balloons in this way, such that the reading order is correct. This is described elsewhere [1]. The method is a greedy algorithm for placing the balloon bodies, while simultaneously keeping track of channels in which the balloon tails can be routed. After the balloon bodies have all been placed in a panel, the balloon tails are routed to the appropriate characters.

PANELS

Finally, several decisions need to be made relating to the

panels themselves. These include incorporating additional semantic elements, choosing how much content to fit in each panel, and choosing an appropriate zoom factor for the virtual camera.

Every time a new utterance is made, Comic Chat tries to add it to the last panel. Comic Chat starts a new panel under any of the following conditions:

1. The balloon layout algorithm is unable to fit the additional utterance in the last panel.
2. The resulting panel would have too many characters, and hence the characters would be rendered too small for their expressions and gestures to be clearly visible. Comic Chat limits the number of characters per panel to five.
3. A speaker in the last panel speaks again. Although there are comics conventions for representing multiple utterances for a single character in a panel, Comic Chat cannot draw such panels yet.

Our initial research implementation of Comic Chat allowed for the automatic creation of semantic elements in the panels. Semantic elements are additional scene elements that reflect the topic of the conversation. People in chat rooms often speak about common themes: they discuss where they are from, they talk about their families, pets, sports, and a few other favored topics. When such a theme is identified, the panel's composition is altered in one of several ways. The

THE GANG'S ALL HERE

STARRING





-  Rictor as Dan
-  MoonGirl as Margaret
-  Rabid_Dog as Mike
-  Chyrric as Tiki



Figure 2. Part of a fantasy role-playing chat, as rendered by Comic Chat.

background can change to reflect the topic, an additional scene element can be composited into the background, or an additional "Greek chorus" character can be added to the scene, making a quip about the topic. This feature has not yet been added to the product version, in part due to the difficulty of authoring semantic elements, and the additional art requirements. However, we expect this feature to be incorporated in a future release.

One last form of Comic Chat automation is the choice of a zoom factor for the virtual camera. Comic Chat always zooms out for an initial establishing shot. This technique is common not only in comics, but also film and video. After the initial establishing shot, Comic Chat zooms in as closely as possible on the conversational participants included in each panel, subject to some restrictions. Comic artists consider it bad practice to crop a character at the neck or ankles. Instead, it is better to zoom out a little more for better composition. Characters and backgrounds are both affected by the field of view of the virtual camera. However, balloons are special objects whose scale is constant, and independent of the size of the characters and the background.

Figure 2 shows part of a fantasy role-playing chat, as rendered by Comic Chat.

PRODUCTIZATION

Comic Chat is now available under the name *Microsoft Chat*. It has been localized to 24 languages (including Japanese), it runs under Windows '95 and Windows NT, and is available free of charge via the web from <http://www.microsoft.com/ie/chat>. Microsoft Chat is the official chat application for the Microsoft Network. It also ships with Internet Explorer 4, recent versions of Windows '95, and will be included with Windows '98 and Windows NT 5.0. Millions of people have already installed the application.

LESSONS LEARNED

One of the most important lessons that we learned is that it is vital for others to be able to try out the results of user interface research. Otherwise it is hard to determine what works well, what fails to work, and what needs additional refinement. We encourage all user interface researchers to make their results publicly available, either free on the Internet, or in product form.

People new to chat seem to enjoy the comics representation the most. We hear over and over again from people trying text-based chat after having used Microsoft Chat, that they really prefer having the comics visuals. However, people that have been using text-based chat for years have trouble moving to comics-based chat, and prefer the text mode of Microsoft Chat.

Comics work extremely well in small chat rooms with only a few simultaneous threads of conversations. When people try to view rooms with large numbers of people (twenty to several hundred), it becomes difficult to follow the conversation in comics mode. This problem is exacerbated by Microsoft Chat shipping with only a

small set of available characters – frequently multiple people in a room have the same choice of character.

When implementing the program, we did not foresee that the comics representation would affect the content of the conversations. However, it does. People tend to "play act" much more in comics, and they tend to take on the persona of the character that they choose. For example, people choosing the space-alien character often talk about flying saucers and extraterrestrial visitations. People choosing the cat character frequently pretend to be a cat.

One of the most frequently requested features for Microsoft Chat was the ability for people to draw their own characters, rather than being limited to one from a bundled set. So recently we released a Character Editor program enabling people to create their own art. Future versions of Microsoft Chat will allow people to distribute their own custom art to others.

When we initially released Microsoft Chat, we were naïve and thought that the unique visual representation would make all of our users completely satisfied with the program. In reality, our users wanted the visual representation, *plus* all of the features found in other chat programs. They liked the comics, but also wanted the ability to be in multiple chat rooms with a single instance of the application, to send formatted text to others, to define macros, and more. As a result, after releasing the first version of the product, we had to focus much of our efforts on matching competing products, feature for feature.

Perhaps one of our biggest surprises was how gratified we felt after many people started using our product, setting up unofficial web-sites dedicated to it, and monthly newsletters about it. Having others use our product and interacting with these people is the most rewarding aspect of my time at Microsoft.

REFERENCES

1. Kurlander, D., Skelly, T., and Salesin, D. Comic Chat. Proceedings of SIGGRAPH '96 (New Orleans, LA, August 4-9). In Computer Graphics Proceedings, Annual Conference Series, ACM, New York, 1996. 225-236.
2. Microsoft Corporation. Microsoft Introduces V-Chat Communications for MSN, The Microsoft Network. Nov. 30, 1995. Microsoft Press Release. Redmond, WA 98052.
3. Morningstar, C. and Farmer, F. R. The Lessons of Lucasfilm's Habitat. In *Cyberspace: First Steps*. Benedikt, M. ed. MIT Press, Cambridge, MA. 1991. 273-301.
4. The Palace, Inc. The Palace. <http://www.thepalace.com>.
5. Worlds, Inc. Worlds Chat. <http://www.worlds.net/product/wchat>