

# The Visual Messenger Project

by

Alex Chen

A senior thesis submitted in partial fulfillment of  
the requirements for the degree of

Bachelor of Science  
With Departmental Honors

Computer Science & Engineering

University of Washington

June 2004

Presentation of work given on 14 April 2004

Thesis and presentation approved by Steven L. Tanimoto \_\_\_\_\_

Date: 4 June 2004

## 1. Project Introduction

Visual Messenger is a graphically based chat program developed in Java. But unlike most chat programs that use textual languages for communication, Visual Messenger implements a visual language. The purpose in implementing a visual language was to explore this alternative form of communication which, with further development, has the potential of alleviating some of the difficulties inherent with cross-cultural intercourse. Since an ideal visual language is universal—that is, one that all humans can intuitively understand—they are much easier to learn and appeal to a greater number of people than textual languages.

Though an important part of the project was the development of the actual chat program, the study of visual languages and their capabilities was also an interesting and informative facet. First, a brief discussion of visual languages will be presented, followed by a more in-depth introduction to Blissymbolics. Blissymbolics (also known as Bliss) is the visual language which was used as a base for the language implemented by the Visual Messenger. Additionally, the Visual Messenger chat program was designed specifically for the iPAQ Pocket PC platform—the iPAQ being a small, lightweight personal computer—though it can also run on any PC. After the capabilities and features of Visual Messenger are introduced, some of the problems and advantages of developing programs for the Pocket PC platform will also be discussed. The most notable challenge in developing the chat program revolved around the issue of language extensibility—that is, using a reasonably small number of symbols, while retaining the largest possible set of meanings. As will be seen, this consideration was taken to heart when choosing the language implemented by Visual Messenger.

## 2. Introduction to Visual Languages

Before diving into how Visual Messenger works and what its capabilities are, it is necessary to first provide an introduction to Visual languages—what they are, and what can be done with them. Furthermore, some additional information is provided about their history, as well as their possible future in today's world.

## **2.1 History & Resurgence**

Interestingly, the first known languages were pictographic. Take the cave paintings at Lascaux (dated to 15,000 – 13,000 BC) or other various locations. These drawings were meant to convey stories—a historic example in which meanings were transmitted pictorially. Such examples of "natural" visual languages are not difficult to find throughout history, as the custom permeated through time and culture for many centuries. A couple other examples include Egyptian hieroglyphics and Chinese story paintings—both of which, again, used pictures to convey meanings. But as time progressed, languages eventually morphed into more abstract alphabets due to the fact that this made reproduction easier. It is easy to see that the letters of many modern day languages are easier to reproduce than pictures—even today, it is much harder to have a keyboard (or similar device) for more pictorial languages such as Chinese.

Nevertheless, in recent years there has been a resurgence in the usefulness of, and interest surrounding visual languages. The primary reason for this is the fact that their application is becoming increasingly more manageable due to rapid advancements in technology—most notably, the speed and capacity with which computers can transmit information. The modern popularity of email (especially graphical email) and expansion of the world wide web have made the study of visual languages particularly relevant, as these continuing trends mean that visual languages have a bright future in the world of online communication.

## **2.2 Definitions & Discussion**

What makes a visual language? Defining what constitutes a visual language is a pioneer requirement to their implementation and research. In fact, people in today's world are surrounded by a plethora of everyday articles that could be considered "visual languages". These include everything from road signs to advertisements to graphs, charts, artwork, and even children's drawings. The reason why these may be considered visual languages is in the way they convey their intended meanings to the observer—that is, through visual symbols. Advertisements, for example, use images and pictures in a variety of ways in an attempt to make people desire their product. Similarly, road signs are often recognized by their color and shape, as opposed to any

text that might be present on them. So what exactly is the definition of a visual language? For the purpose of our research, a visual language can be defined as "a language in which communication occurs through visual symbols".

Closely related to this need for a definition is the notion of what makes a language visual. As the above definition implies, visual languages are composed of pictorial representations of objects and ideas. So, not only are objects like chairs and artichokes represented by pictures, all elements of the language are presented pictorially. Additionally, visual languages also rely on graphical likenesses of their images to the things they represent. Simply stated, this means that pictures should either physically or intuitively represent their intended meanings. The satisfaction of these conditions often gets interesting, as things such as feelings and activities are not always easily represented pictorially, especially if the images are to remain simple. Furthermore, it is important to note the distinction between pictorial and iconic language representations. Icons are universal representations of ideas—independent of context or user interpretation. However, this idealism is difficult to achieve, and therefore, we are often forced to settle with slightly less universal pictorial representations.

Finally, it is important to note that visual languages are in fact languages, and are therefore governed by their own set of syntactic rules. Their rules may be very strict or very lax, but they are unique to the language. The language is not defined by the grammar, and most certainly does not follow the syntactic rules of textual languages.

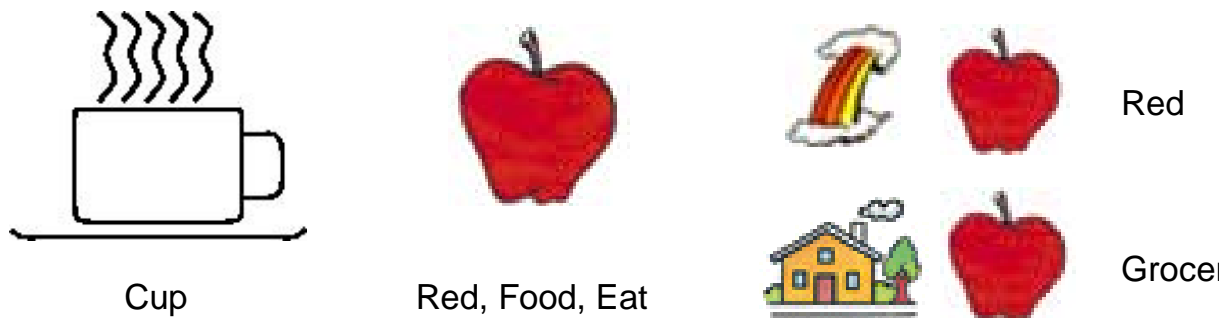
### **2.3 Examples of Visual Languages**

Though for the purposes of our research, things like advertisements and road signs are considered visual languages, some may argue that these are not true visual languages. Nevertheless, there are a host of formal languages that can be given as examples, which were constructed with specifically with the idea of being visual in mind. One such example is sign language, which uses all possible facets of visual communication—sight, movement, expression, etc. Minspeak is another example of a visual language, with some examples of images provided in *figure 1*. This language was developed explicitly with handicapped users in mind. One final example is Blissymbolics, which will be discussed in detail further on.

## 2.4 Motivation

As previously mentioned, visual languages are everywhere in today's world, permeating almost every aspect of our everyday lives. As a result, it can be argued that understanding their meanings and implications are necessary for everyday interactions. However, more relevant to the concern of this project is the idea that visual languages can help make communication easier—and not only cross-cultural communication, but all communication in general. The reason for this lies in their power to encapsulate ideas and emotions difficult to express in words. Visual languages also make improvements in the resolution of ambiguity. Most notably, the ever expanding popularity of the internet makes their usage more feasible, as our ability to develop a chat program that uses a visual language demonstrates. Examples of the power (and restrictions) of visual languages will be given in the next section where one specific visual language—Blissymbolics—is discussed at length.

*Figure 1: some examples of Minspeak symbols*

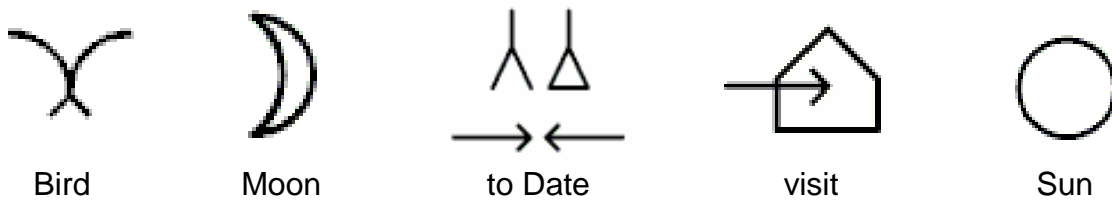


## 3. Blissymbolics

Blissymbolics is a visual language containing approximately 2000 symbols that was developed by Charles K. Bliss in 1949. Bliss's original vision was for his language to be used as a universal one, which could transcend traditional language barriers in international communication. Unfortunately, this goal has not yet been reached, though the language has been adopted for use by the paraplegically disabled. The pictorial symbols of Blissymbolics draw upon many influences, ranging from Chinese characters

to Egyptian hieroglyphics. Additionally, what makes the language so popular in the visual language community is its simplicity, as the symbols use only two shapes—straight and curved lines—and often resemble children's drawings. Furthermore, the language as it was developed is quite intuitive, using many constructs including symbol reuse, repetition, and a half-dozen others making it quite easy to learn quickly. These are a few of the reasons why Blissymbolics was chosen as the base language for Visual Messenger. A few examples of Blissymbols are provided in *figure 2*.

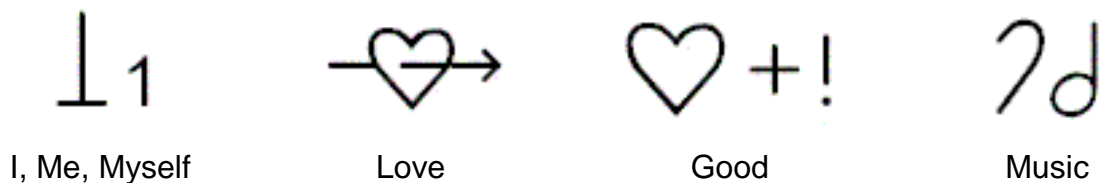
*Figure 2: some examples of Blissymbols*



### 3.1 Writing Sentences in Bliss

Before going further into how Blissymbols are constructed it is important to give an example of how sentences are composed in Bliss. Let's use the example sentence, "I love good music". First, the symbols are found which correspond to the elements of the statement, then placed in any sensible order. The traditional way of writing this sentence is shown in *figure 3*.

*Figure 3: an example sentence in Bliss*



However, Bliss is very lax when it comes to its syntactic constraints. One could rearrange the four symbols in many different ways, and the same (or similar) meaning would still be obtained. Only few possible arrangements might be slightly confusing.

### 3.2 Constructing Blissymbols

Now, we can tackle the issue of how new symbols are created within the Bliss system. Given some elementary symbols (which are often nouns or adjectives), there are a multitude of constructs through which new meanings are derived. Seven of these constructs are described below:

#### 1. Reuse

When creating symbols that revolve around the same subject matter, Bliss often reuses a base symbol. In *figure 4*, we can see that the heart represents "feeling". Subsequently, added structures around a heart signify different emotions.

#### 2. Repetition

Similar to the reuse of symbols is the repetition of the same structure. Most often, this is used to denote an increase in magnitude. An example is given in *figure 5*.

#### 3. Rotation

Another form of symbol manipulation is rotation. In *figure 6*, we can see a couple examples of symbols being rotated to obtain different meanings.

#### 4. Combination

Here we get into the more robust ways of creating new symbols. One obvious way is through a combination of previously created symbols to derive linked meanings. Several examples are given in *figure 7*.

#### 5. Indicators

Another way of denoting different meanings is through indicators. There are three primary types of indicators in Bliss: physical inflectors (*figure 8*), emotional/feeling arrows (*figure 9*), and directional pointers (*figure 10*). It is interesting to note that the

figure of a hand in *figure 8* is derived from looking at ones right hand held with the four fingers together, and the thumb pointed out. When viewed in this way, it is apparent that the inflectors of the two subsequent images point to the thumb and finger, respectively.

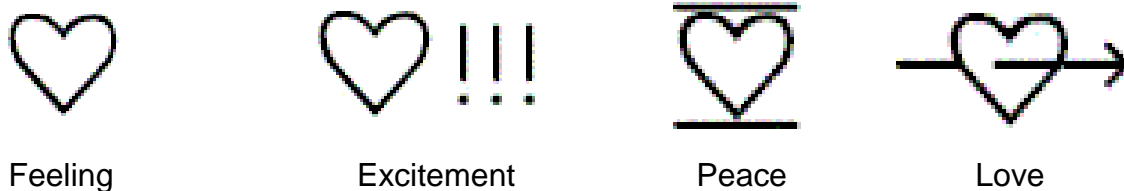
## 6. Function Changes

One of the most interesting ways of manipulating symbol meanings is through function changes. These changes are derived through the use of small indicators, which appear either as superscript or subscript add-ons to the base image. A few examples of how they are used are given (*figure 11*), followed by a comprehensive list of all such indicators (*figure 12*).

## 7. Intuitive Construction

The final way in which new symbols are created is through intuitive construction. This intuitiveness, combined with the strong graphical likeness to actual objects of many symbols, is what makes Bliss such an easy language to understand. These productions draw upon the previous six forms of symbol assembly described in this section. A couple examples of how symbols are intuitively built are given below (*figure 13*).

*Figure 4: some examples of how Blissymbols are reused*



*Figure 5: some examples Blissymbol repetition*

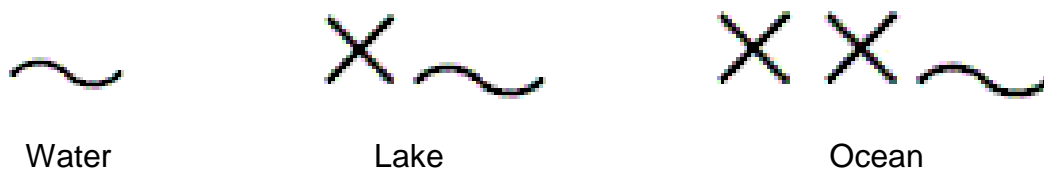




Figure 6: some examples of rotation

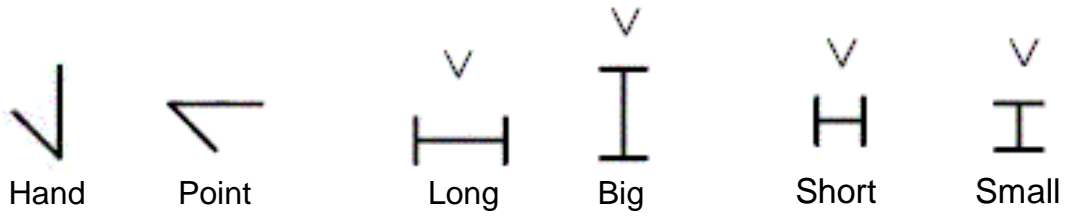


Figure 7: combining Blissymbols

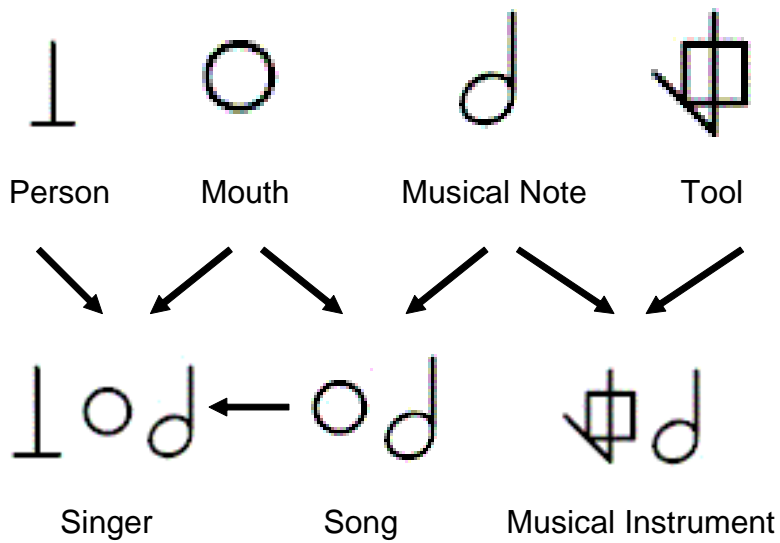


Figure 8: physical inflectors

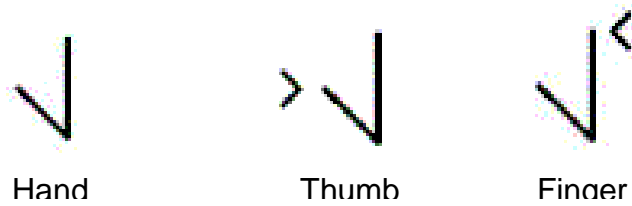


Figure 9: emotional arrows

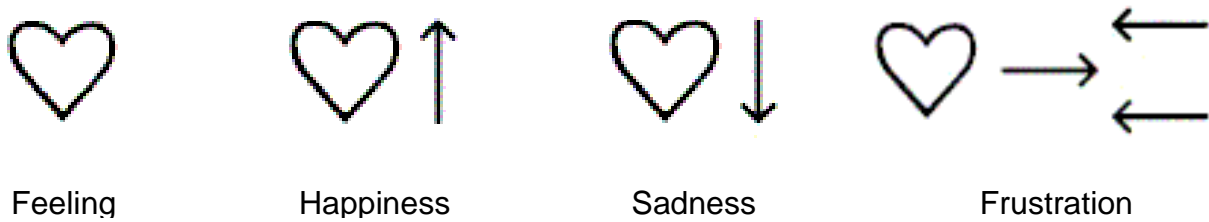


Figure 10: directional pointers

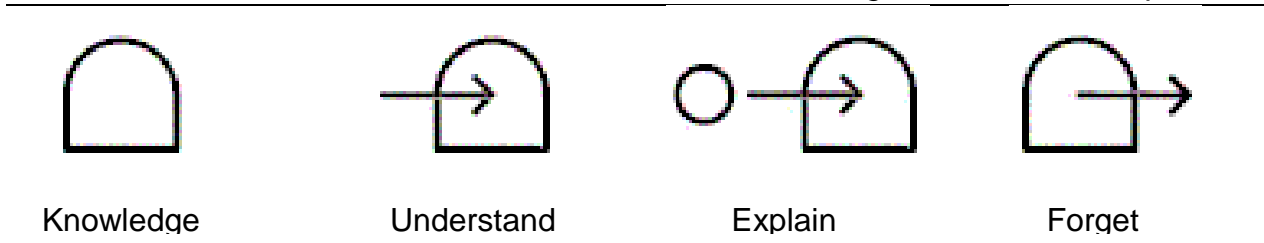


Figure 11: how function indicators change symbol meanings

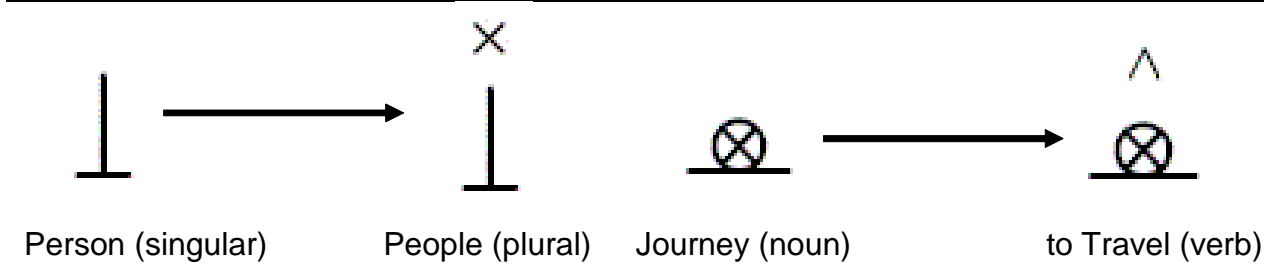


Figure 12: list of all Bliss function indicators

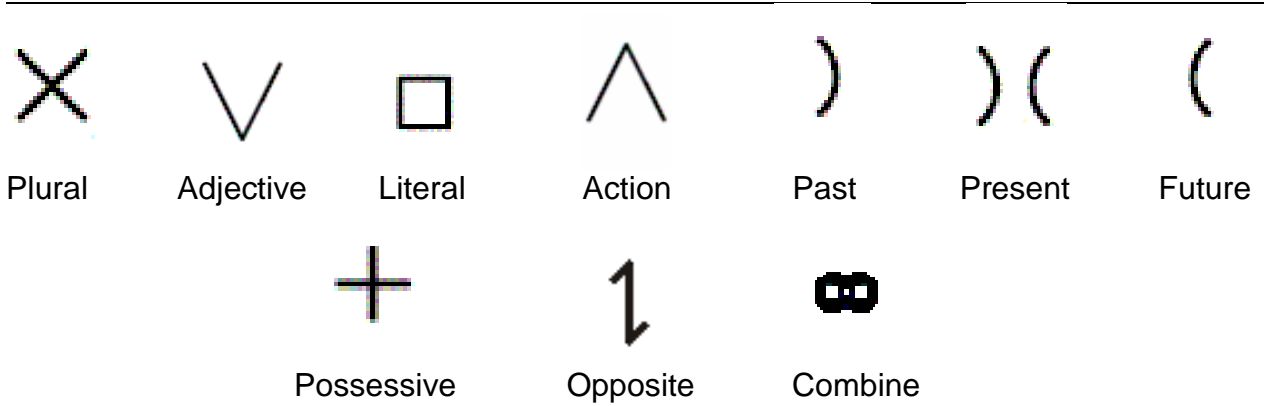


Figure 13: intuitive construction



### 3.3 What Symbols Mean

Before moving on to describe how and what symbols were implemented in the Visual Messenger, it is important to briefly note that the symbols of Bliss refer to objects or ideas, and not words. To illustrate this point, four symbols are presented (*figure 14*).

*Figure 14: four symbols that all translate to the same English word*



Each of the four symbols translates to the same English word, "bow", but in actuality have very distinct meanings. When communicating in Bliss, it is clear that the leftmost symbol means "person bending over", while the three subsequent symbols to the right mean "the front of the ship", "an archery tool", and "a ribbon tied in knot", respectively. In this way, visual languages such as Bliss alleviate some of the ambiguity problems inherent in textual languages.

### 3.4 Bliss in the Real World

As we have seen, programs such as Visual Messenger serve as one possible niche where Bliss could be used to perform a valuable function. However, in this section, we examine some of the ways in which Bliss is currently being used out in the real world. Thus far, Bliss has found its largest user base in the handicapped community, as it is actively used with and by persons with speech and/or physical impediments in over 33 countries. Additionally, according to the official Blissymbolics website, Bliss was at its height in 1982, at which time over 8,000 instructors attended workshops worldwide. It is safe to assume that these instructors subsequently interacted with a number of students, though the exact number is not known. Currently, it is estimated that around a quarter of a percent of students in school systems use Bliss as an alternative form of communication (a number also gathered from the official Bliss

website). However, as previously mentioned, recent technological advancements—most notably in the area of online communication—offer the opportunity to expand both the number of users, and the effectiveness of visual communication.

## 4. The Visual Messenger

As previously mentioned, Blissymbolics is the language that was used as a base when implementing Visual Messenger. Now that we have discussed what Visual languages are, and seen at length how Bliss symbols are constructed and operate, it is time to delve into the specifics of the Visual Messenger application.

### 4.1 Features

In this section, the primary features of the Visual Messenger chat program are discussed, being message composition, conversation capabilities, and translation. The following descriptions will refer to specific buttons and regions on the Visual Messenger's windows, shown in *figure 15*.

#### 1. Message Composition

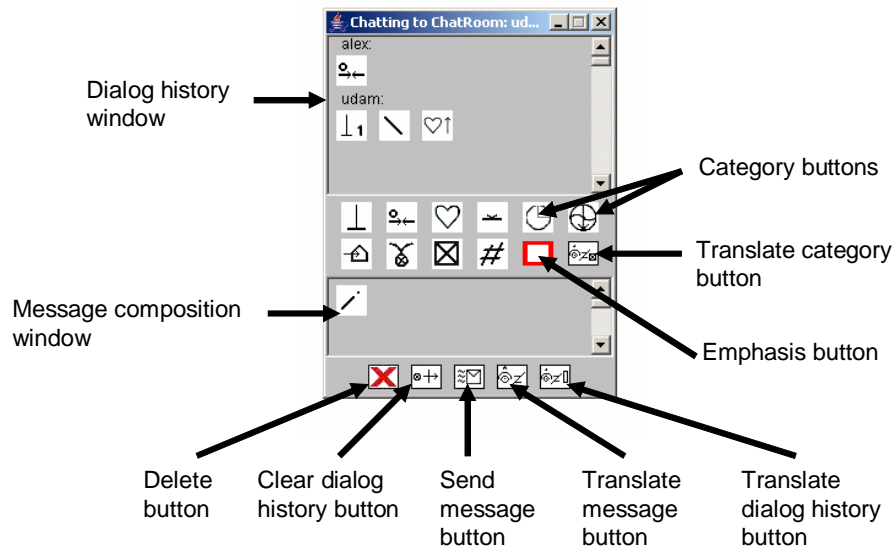
When composing messages, first an image category is selected, which causes the associated image window object to pop up. An image can then be selected, at which point the image window disappears. The image then shows up on the message composition portion of the main chat screen. Additionally, there is an emphasis button which can be turned on and off. When turned on, any image selected appears in the message composition window with a red box as an outline, indicating it is emphasized. Additionally, the delete button can be used to remove the last image selected from the message composition window, and the send button is used to send the message that has been composed.

#### 2. Conversation

Upon sending a message, it then appears in the dialog history window. Messages which are received also appear in this window. The 30 most recent messages that have been sent and received are stored in this window. A clear button on the bottom panel can also be used to clear this screen of all dialog.

Figure 15: the Visual Messenger Windows

Chat Screen



Main Screen

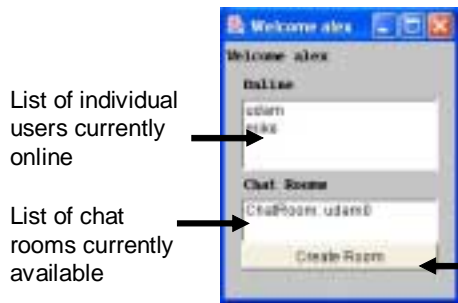
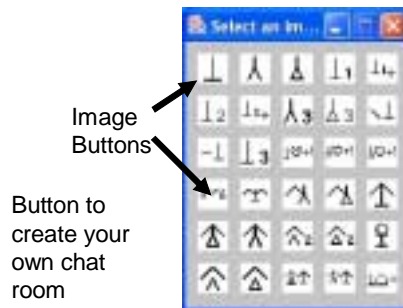


Image Window



Translation Window



Furthermore, two types of conversations can take place: person-to-person, and chat room. The appearance of the chat screen, and the way in which messages are constructed are exactly the same for both modes of conversation; the only difference is the number of people who receive each message. Initiating a person-to-person conversation is done by selecting any name in the "online" portion of the main screen window, and joining a chat room is done by selecting any chat room from the "chat room" portion of the main screen window.

### 3. Translation

One major issue that deals with the use of Visual Messenger is the matter of learning the language. The translation feature was therefore added to assist with comprehension. There are three types of translation that have been implemented. First, a translate category button appears next to the category selection buttons. When this button is turned on (by clicking it), clicking a category button results in a translation of the entire category, which appears in a new translation window. Images cannot be selected until this button is turned off (by clicking it again).

Additionally, two more translate buttons are provided on the bottom panel. There is a translate message button which translates the message that is currently being composed in the message composition window. Finally, there is a translate history button, which translates all messages that appear in the dialog history window.

#### **4.2 The Implemented Language**

The language of the Visual Messenger chat program is essentially a 300-symbol subset of Blissymbolics, with some added features to improve language extensibility and enhance clarity of meaning. As shown in section 3.2, Bliss offers a variety of function change inflections, which can be used to change the meanings of symbols, sometimes drastically. Therefore, an entire category of Visual Messenger has been dedicated to these indicators, allowing for a large amount of extensibility.

Another obvious issue was the matter of how to choose which symbols would be included in the 300-symbol subset of Bliss. In order to make this determination, a brief survey was given to 100 people. In one section, the takers were asked to compose short sentences in general areas of every day conversation—greetings, the weather, family, hobbies, one's day, and expressions. All of the words used were then recorded, each instance meticulously counted, and the results sorted by frequency. The top 20 or so words from each of the above mentioned categories were then chosen for use in the program. Added to these were about a dozen words in each category that may not have been used extensively, but the inclusion of which were deemed necessary.

### **4.3 The Survey**

As mentioned in the previous section, a short survey was given with one goal of determining which symbols should be chosen for use in the Visual Messenger application. However, this survey also contained within it other sections which provide interesting (if not directly useful) information about Visual Languages. For example, one section (section 4) presented the takers with 20 Blissymbols, and asked them to identify what the intended meanings were. In addition, there was another section (section 5) in which the takers were given a Blissymbol and its meaning, and then asked to determine the meaning of a similar symbol. Furthermore, there was a "personal information" section which asked the taker for information regarding their language capabilities, age, ethnicity, etc. After tabulating all of this data, several trends could be observed. First of all, people in every category did equally poorly in identifying symbols straight away (section 4), with the average identification rate around eleven percent (11%). When given the meaning of a similar symbol, however, the identification rates jumped dramatically, with percentages often reaching the forties or higher. Additionally, the results from section 5 (the "intuitive" section) indicates that people with different backgrounds were more or less capable than others at inferring what symbols meant. For example, people who were multi-lingual did better than those who only spoke English; and the subjects who were fluent in an Asian languages (Chinese, Japanese, Korean) did even better than the multi-lingual group as a whole. This makes sense, since the Bliss system is often compared in its appearance to Chinese characters, which means it may be more intuitive to those who speak Chinese or similar languages.

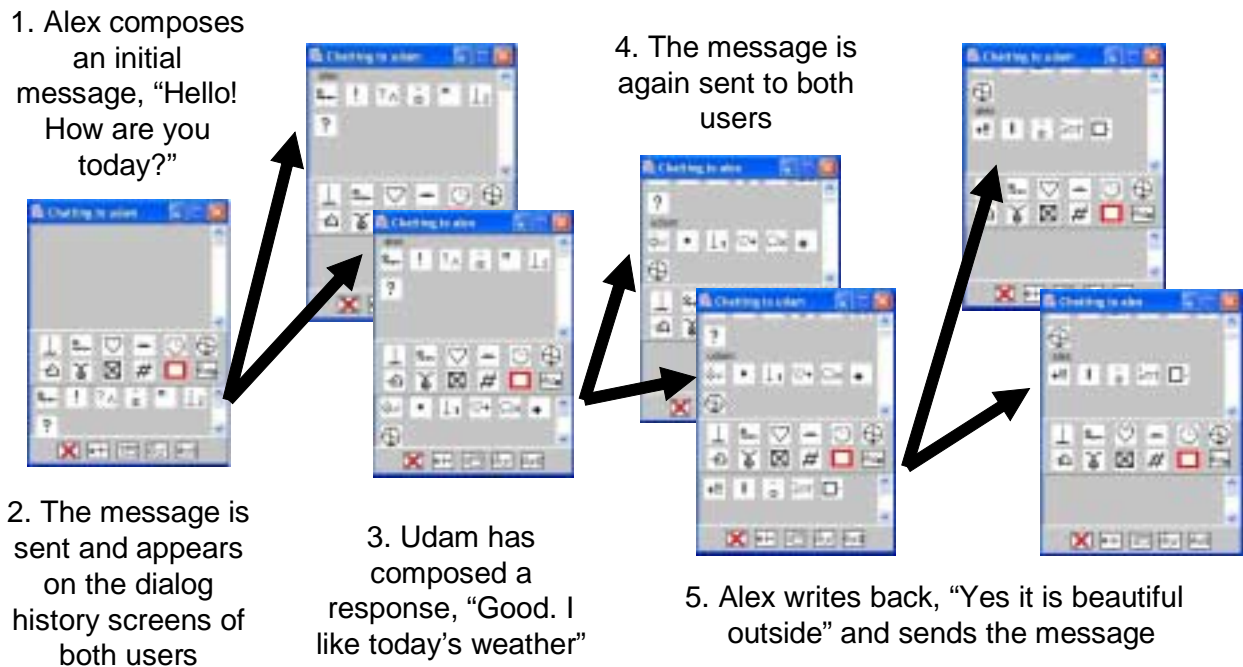
### **4.4 An Example Conversation**

*Figure 16* on the next page maps out an example conversation in Bliss using the Visual Messenger chat program.

## **5. Developing Java Applications for the iPAQ Pocket PC Platform**

As we have seen, the Visual Messenger application provides an alternative to textual languages when it comes to online communication. As previously discussed, visual communication comes with certain advantages—most notably, it reduces some of

Figure 16: an example conversation



the barriers inherent with traditional cross-cultural communication. Additionally, when it comes to Visual Messenger's development, there are both advantages and disadvantages to developing programs for the iPAQ's Pocket PC platform, as there are with most platforms. In this section, a few of these issues are discussed. The primary advantage of the iPAQ is portability. As briefly mentioned, the small size and light weight of the iPAQ makes it easily transportable. It is about the size of a daybook or planner, but has capabilities that are far greater, including the support of our Visual Messenger chat program. However, in developing Java applications for the iPAQ, there are four notable challenges—language extensibility, limited screen size, restricted implementation capabilities, and responsiveness to location. These four issues are discussed further, as the majority of Visual Messenger's development revolved around resolving these issues.

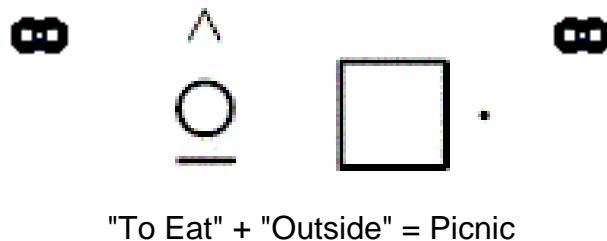
### 5.1 Language Extensibility

When implementing the language of the Visual Messenger, it was important to keep in mind that, while it was necessary to contain enough symbols for elementary



conversation, too many symbols could lead to confusion, as well as inefficiency of usage. This is why a subset of the entire Bliss dictionary was chosen, as 2000 symbols would be too many to keep track of. However, this led to the issue of how to contain the greatest possible ontology in this limited collection. As discussed in section 4.2, Bliss's function change indicators provide one way of vastly extending the set of meanings contained within the language. Also notable are the combination indicators, which can be used to derive conjoined meanings such as those in section 3.2.4. An example of how to do this manually is shown in *figure 17*. Additionally, the emphasis feature described in section 4.1.1 allows people, among other things, to stress different parts of a message—another way of subtly changing its meaning.

*Figure 17: the combine indicator at work*



## 5.2 Limited Screen Size

In developing programs for the iPAQ, one of the principle considerations was the computer's very limited screen size. The total size of the iPAQ screen is 240 by 320 pixels, though the actual size to work with is smaller. Permanent panels on the top and bottom of the iPAQ's window further restrict the amount of vertical space there is to work with. This issue was particularly constraining to the development of the Visual Messenger, which relies almost exclusively on graphics. When constructing the graphical windows of the chat program, there were several issues to take into account. First, there was the fact that the set of symbols which comprise the implemented language is far too great to be displayed on the screen all at once at any given instance.

It is for this reason why the symbols were divided up into categories, which each appear on their own pop-up menus (see section 4.1.1).

Additionally, we wanted to be able to store a history of messages that had been sent and received. This, too, would require much more than the available space. In fact, it is also a foreseeable possibility that even a single message, while being composed, could use more than the available space. For this reason, both the dialog history window and message composition window use virtual scroll panels. In this way, all the information is stored in virtual space, with only a small portion appearing on the screen at any given moment.

Finally, there was an inherent trade off between the size of each image, and the amount of detail that could be contained in each. Although larger images would allow for more detail—which could allow for greater clarity of meaning—this would mean reducing the number of images that could be seen at once. Seeing too few images is undesirable, since it becomes difficult to understand the meaning of a sentence. This is another reason why Bliss was chosen as our base language, as the simplicity of each symbol allows them to be shrunk to a reasonably small size and still retain their form.

### **5.3 Restricted Implementation Capabilities**

Unfortunately, when it comes to the iPAQ, there are several factors that greatly restrict what can be implemented. In the case of Java applications, the Pocket PC platform lags behind in what it can support. JEODE is the sole virtual machine created for the platform, and is equivalent in power to Java version 1.1.8. Most notably, this means that programming must be done without the Java SWING package, which contains a wealth of GUI and graphics libraries. Instead, all of the graphical requirements of Visual Messenger were implemented using Java's AWT package, which supports far fewer functions and whose components look more archaic than more modern day packages.

Additionally, the restrictions placed on the program's development by the iPAQ's speed is also noteworthy. When running Visual Messenger on the iPAQ, it is noticeably slower than running it on a desktop. As a result, it is necessary to carefully scrutinize the way in which features are implemented, so that unnecessary operations do not take

place. Furthermore, the touch-screen interface of the iPAQ is less responsive than the mouse clicks of regular PC's, making it slightly more difficult to select images.

#### **5.4 Responsiveness to Location**

The final issue to consider when developing programs for the iPAQ is responsiveness to location. Most conversations in everyday life make use of context—where a people are, where they have been, where they are going, etc. The Visual Messenger does contain a symbol category of common places—but they do not get any more specific than "house" or "library". Unfortunately, it is impossible to provide the means with which to describe all possible aspects of travel and context, as such a feat would require an extremely large set of symbols. As an example, suppose we wanted to include a symbol for every country in the world. This is plausible assumption, as people would often like information about where others reside. However, implementing this is not very realistic, as there are 175 countries recognized by the US in the world. This alone would require more than half the allotted number of symbols (300).

One possible solution to this would be the integration of a GPS tracking system, which would keep track of where the user currently is and update the symbol set available based on that information. Examples of these symbols could include the current country, as well as nearby geographical landmarks and features. Such a system, however, has yet to be developed.

### **6. Concluding Remarks & Acknowledgements**

As we have seen, formal visual languages now have a very bright future in the world, largely due to large strides in computer technology. The Visual Messenger chat program was developed with the aim of exploring some of these possibilities. Hopefully, further research and technological advancements will allow for similar improvements in the area of visual messaging.

The completion of this project would not have been possible without the help of several individuals. I'd like to thank Steve Tanimoto for sponsoring the project, as well as my project partners Udam Dewaraja and Michael Asavareungchai. Additionally, Nick Benson, Eric Bessette, and Aaron Timss were instrumental in getting the program up

and running on the iPAQ computers. Finally, I'd like to thank Christy Lindquist, Jack Black, and Debbie Benzinger for their time administering the above mentioned surveys.

## References

Blissymbolics Communication International [Online home of Blissymbolics]:  
<http://home.istar.ca/~bci/>

Blissymbolics Dictionary [Online]: <http://www.symbols.net/blissymbolics/dictionary/>

Kim, Eugene Eric. Visual Language: Global Communication for the 21<sup>st</sup> Century.  
*Web Techniques* 6: 68, 2001.

Tanimoto, S. Representation and learnability in visual languages for web-based interpersonal communication. *Proceedings of the International Symposium on Visual Languages* (Capri, Italy, 1997) pp. 2 – 10.

Tanimoto, S., et al. Four Challenges for Visual Messaging on Wireless PDA's [unpublished].