

The Design and Implementation of Vedo-Vedi, A Visual Language for Human Communication in the Internet

Steven L. Tanimoto

Carlo E. Bernardelli

Dept. Computer Sci & Engin.
Univ. of Washington
Seattle, Washington 98195, USA

Visual Epistemologist
I^o Art School of Via Ripetta
Via Ripetta 218, Rome, Italy

Abstract

Vedo-Vedi consists of a visual language and software that supports editing, viewing, and sending visual messages written in the language. Its purpose is to enable 8-15 year old children to communicate visually through the Internet without their having to read or write a common textual language. Novel aspects of this system include its frame-based syntax and semantics, an underlying machine representation of the meaning of messages, facilities for translation into several textual languages, movie-like animation of messages, and support for user extensions. Vedo-Vedi works with Java 1.1 enabled browsers or as a stand-alone application on PCs connected to the Internet. This paper describes its key design features and some of the implementation issues involved in realizing them.

1 Introduction

1.1 Rationale

One of the most exciting aspects of the Internet is that it permits communication among pairs or groups of people who were not able to easily communicate or even be introduced in the past. Most of this communication takes place using email, newsgroups, electronic chat, electronic talk or web pages. One barrier still remains for many potential users: language. In order for two people to communicate with email, for example, they must read and write a common language such as English.

There are three basic approaches to overcoming the language barrier: (1) education of more people in foreign languages, (2) automatic translation of text from one language to another, and (3) new visual languages that are easy to learn and use regardless of what one's native language happens to be. This paper describes a system that follows the third approach.

The visual approach is attractive for several reasons. First, many people today are visually oriented — they have grown up in a world of film, television, photography, and art. Second, most personal computers connected to the Internet have excellent graphic display capabilities, and software for managing and transmitting images is well-developed. Third, many meanings can be made particularly clear by the use of images, diagrams, and movies, so that it is often easy to understand a visually-presented message without any explanation.

1.2 Brief History

Visual communication by people to other people has been going on for many thousands of years. The cave paintings discovered at Ardeche exemplify one form of this. Elaborate languages based upon pictographs were developed in Egypt (Hieroglyphics) and China, to mention only two.

Modern-day systems of icons for general human communication include Bliss Symbols and Elephant Memory. While Bliss intended his symbol system for use in scientific communication as well as several other domains, the primary users of it today consists of disabled people. Other systems, such as Minspeak, have been created specifically for handicapped users. For a survey of these systems, see [4].

1.3 The Contribution of Computation

An important aspect of the potential for new visual languages is that they can exploit not only the electronic graphical display but also computation. It is possible, at least in many situations, for a computer to build up internal data structures that represent not only the appearance but also the meaning of an expression. This meaning can be linked to a simulation of the situation being described. The simulation then can be used to help the reader fill in gaps or resolve ambiguities in a message. It is also possible for translations to be generated as needed, for animations to be performed, and for other types of explanations to be provided.

The use of computer graphics in visual language systems for human-to-human communication includes the PLAY system of Tanimoto and Runyan for animation [3] and the work by Beardon [1] in which iconic sentences are organized around the case frame structure used in some computer systems for natural language understanding.

Our system, called Vedo-Vedi (which in Italian means “I see-You see,” brings computation into the language at several levels, including user interaction with the script, animation of the script, simulation of some of the underlying meaning, and automatic translation. As far as we know, it is the first visual language with any syntactic structure which operates using scanned imagery in the Internet. As such, it introduces a number of technical, linguistic, and interface challenges. Its notion of simulation, involving time, space, and story modes, is also novel.

2 Language Structure

2.1 Syntax of frames and slots

The grammar of a language specifies the structure of statements called sentences. Each sentence expresses some chunk of meaning, and it does this with an assembly of linguistic units usually called words.

A visual language also requires that its statements have form and structure. Vedo-Vedi uses a traditional scheme for its message or script structure; each sentence is described with imagery that is placed within a rectangle called a frame. This format is used by comic books. The frames form an ordered sequence that typically describes a series of events in temporal order of their occurrence.

Each Vedo-Vedi frame contains a “frame image” that is displayed in a region of the frame called the frame canvas. In addition, the frame may contain any number of subrectangles called slots, and each slot has a filler – an image that represents an object that plays some role in the sentence described by the frame. These objects serve as words (actually nouns) in the sentences expressed by frames.

Vedo-Vedi 1.0 uses a limited number of layout schemes for frames. For example, a transportation frame contains a frame image of a vehicle shown in the middle of the frame, a source slot (on the left side of the frame canvas) containing an image for the place that the transportation departs from, and a destination slot (to the right of the frame canvas) that holds an image representing the destination for the transportation. On the other hand, a calendar frame employs eight slots each of which holds the image of a numeral.

The position of each slot in the frame corresponds to the role that the slot’s filler plays in the sentence. Examples of these roles are shown later.

2.2 Overview of Semantics

In Vedo-Vedi, meanings are represented in multiple ways including visual images, textual words, phrases, and sentences, internal data structures, and animations.

2.3 Ontology for VV

In order to permit internal representations of meaning using data structures that are not overly complicated, the set of possible meanings in Vedo-Vedi is explicitly restricted. Users may communicate about the following topics in particular ways: vacation travel, dates and time, places, family members, dreaming and thinking, and human feelings. The particular kinds of things that can be represented will become clearer as we present the various kinds of frames later. This set of meanings embodied in Vedo-Vedi we call a “postcard ontology” because it includes many of the meanings that a 10-year old child might express on the back of a postcard when writing to a friend or relative while travelling on vacation.

2.4 Internal representations

A Vedo-Vedi message or script consists of a sequence of frames, each representing one sentence. A frame has both an appearance on-screen and an internal representation in the memory of the computer. The internal representation of a frame consists of (1) a reference to the type of frame being instantiated (e.g., a “visiting” frame) and (2) a list of slot-filler pairs. The frame type effectively indicates the verb of the sentence, while the slot-filler pairs identify the various nouns in the sentence together with the roles that they play. It is not necessarily the case that all slots and their fillers for a frame are displayed in the script view; some slot values may either be completely invisible or visible only when the frame is viewed in an expanded view.

In addition to the internal representations of frames, there are other internal representations of the meaning of a Vedo-Vedi message or of its components. After it is created, a frame is usually interpreted by the system. The interpretation may happen immediately, in the case of frames that introduce new objects, or it may happen when an animation is requested. When a frame is interpreted, state variables may be created or assigned values within the program. For example, when a frame describing a transportation event is interpreted, a state variable that stores the current location in the world of the travellers is updated to contain a reference to the place which is the destination in the transportation event. Another example is that when a “next day” frame is interpreted, the internal calendar for the story is advanced one day. These dynamic internal representations constitute a kind of simulation of the story that is being communicated. By displaying aspects of this simulation, animated presentations are possible that show more than the mere surface representations entered directly by the user.

2.5 Animation

A Vedo-Vedi script by itself is a somewhat static object. However, users can interact with it in various ways. One way to interact with it is to “play” it by requesting an animation. An animation is generated from the script by a procedure which examines each frame in order, interprets the frame, updates the internal data structures corresponding to the simulation, and displays a timed, graphical presentation of the meaning of the frame.

A script can contain embedded scripts that represent thoughts, dreams, or side trips that take place within the main sequence of events. These embedded scripts are animated within “subscreens” – rectangular regions of the main animation screen. Embedded scripts typically involve not only nested animation but also nested simulations possibly involving separate clock and calendar objects and separate representations of the travelling group and current location in the world. The animations of embedded scripts involve not only showing the sub-stories corresponding to the embedded scripts, but also showing the transitions into and out of the sub-stories. For example, at the end of a dream, there is a display of a transition from the dream back to the place where the dreaming began.

2.6 Microworld semantics vs user interpretation

Although the Vedo-Vedi system includes specific semantics for its objects in the form of animations and natural language translations, users are not forced to use these meanings literally. A user may communicate a story explicitly with Vedo-Vedi but intend that it be interpreted analogically or allegorically. The objects

in a message presumably correspond to real objects in the sender's world, but both the sender and the receiver probably know that they don't look exactly like they do on-screen. The extent to which the Vedo-Vedi message is only a suggestion or an analog for the intended meaning in real users' communications is something we as researchers/designers do not yet understand, even though it has been our goal that Vedo-Vedi messages be used as literally as possible in order to minimize ambiguity and misinterpretation.

3 Use of the System

This section describes various parts of the Vedo-Vedi program from the point of view of users.

3.1 System Overview

The program is started in any one of three ways: (1) as an independent Java application, (2) as an applet without any particular message to present, and (3) as an applet with a message from somebody. In the latter two cases, the launch is accomplished by visiting particular URLs in a Java 1.1-enabled browser such as Netscape Communicator 4.0 with an extra Java patch.

After the program has been started, the screen contains three major panels. The largest of these is the script panel, where the current script is displayed as a sequence of rectangular frames containing imagery. On the left is a panel that contains various square icon buttons. In this panel is displayed a view of the hierarchical menu of frames and objects, which is used for authoring scripts. Along the bottom of the screen are the rectangular command icons; these are used in editing, starting animations and bringing up dialogs to get and send messages, etc.

Invisible are the software components that handle the following functions: maintaining the simulation, posting and getting messages through the Internet, and translating meanings into natural languages.

3.2 Animation

Let us assume that the program has been started, and that a script is present in the script panel. This script can be played by clicking with the mouse on the Animate command icon.

Animation is performed in a separate window that has been sized to full-screen. Each frame of the script is presented in the animation using a combination of image-display and transition effects. The rate at which the animation takes place is currently fixed by internal program parameters. However, the user may stop the animation at any point by clicking on the animation window. Then a dialog box appears with choices Cancel Animation and Continue Animation.

Unless specifically disabled by the user, the animation is accompanied by subtitles. These subtitles explain, in plain English (or in whatever language has been selected by the user), the events being depicted in the animation.

3.3 Script editor

In order to write a script in Vedo-Vedi, the user creates a sequence of frames in the script panel. A frame is created by first selecting a frames submenu (clicking on a blue icon button in the upper part of the menu panel) and then selecting a frame type by clicking on a blue icon button in the lower part of the menu panel.

Many frames have slots – subrectangles to hold objects representing nouns of the sentence. Most frames are created with default values for their slots. To change the value of a slot, the user first selects the slot by clicking on it. A green outline indicates that the slot has been selected. Next the user chooses a submenu for objects by clicking on one of the green icon buttons in the upper part of the menu panel. Finally, the user clicks on a green icon button in the lower part of the menu panel, and this causes the selected slot to be filled with the chosen object.

Frames can be deleted by selecting them (clicking on their borders) and then clicking on the Delete Selected command icon. A new frame can be inserted at a particular point in the sequence by first clicking

on the frame separator (a vertical bar) at the desired insertion point. The current insertion point is shown in red.

3.4 Hierarchical menu

The available types of frames and the available objects are presented in the hierarchical menu of frames and objects that we have already mentioned. The eight icon buttons at the top of the menu panel represent the eight major submenus. Four of these buttons have blue backgrounds and correspond to submenus for frames. The other four have green backgrounds and correspond to submenus for objects. Clicking on any one of these eight buttons causes the corresponding submenu to appear in the lower part of the menu panel. Only one of these eight submenus is displayed at any one time. Generally, these submenu buttons correspond to particular types of frames or particular objects to fill slots. However, it is possible for a submenu icon to represent a sub-submenu, in which case clicking on it brings to the menu panel additional icon buttons.

Icon buttons in the menu are either built-in or they are defined in scripts (either by the user or by other authors).

3.5 How meanings are carried

Each frame expresses a sentence of the message. A sentence may describe an event, may introduce a new person, place or portable object, may describe a feeling or attitude, may establish a time or date, may establish a relationship between a pair of objects, or may begin or end an embedded script representing a dream, a thought, or a side trip.

All events are described in the present tense. In order to express that particular events took place in the past or will take place in the future, particular dates and times may be established within the script at one or more points. Narrative is automatically expressed from a first-person perspective.

Although Vedo-Vedi only stores explicit representations of meaning for items within its ontology, it is possible to use juxtapositions of various frames and objects to express ideas outside of the ontology. How this is done is beyond the scope of this particular paper.

3.6 How to create new objects

The built-in set of people, places, and portable objects is currently rather small. There are fewer than 50 objects initially available. However, the user can easily define additional objects by using special frames called definition frames. A definition frame establishes a new object, associating it with one of the submenus for objects, and it contains a reference to an image (used both for animation and for the icon in the menu) and a textual label assumed meaningful in the currently selected natural language. Definition frames form part of the message, so that when sent to others, the definitions are available at the receiving ends, too. The definition frames can be removed from the script without losing them, if desired. They are maintained in a “hidden script” nonetheless.

In order to create the representation for, say, a new person in Vedo-Vedi, the user selects the “HERE IS” submenu and then clicks on the “new person” icon button. The new-person frame is instantiated, and it shows up in the script panel at the insertion point. However, an “expanded view” of the frame is immediately opened in a separate window, and in this expanded view the user can define an icon and a label for the new person. A large “person world” image is displayed in the expanded view, and the user can select any rectangular region of it for the new icon. If no part of this image is satisfactory to the user, the user may click on the “Gallery” icon button to bring up an iconic index to all of the images currently available within Vedo-Vedi, and any of these images can be selected for use as the base image of the new icon. A textual label can be entered in an appropriate text field in the expanded view to identify the new person. In order to make the definition take effect, the user clicks on the Do Define icon button. The expanded view is closed, and the definition frame shows the general new-person frame image together with the particular image just

created for the new person. Also, the user can find a newly-created icon button for the new person on the People submenu.

If the user wishes to edit this new-person definition to change either the icon or the label for the new person, it is enough to click on the frame canvas of the definition frame to open the expanded view once again. Any changes to the icon or label will take effect as soon as Do Define is clicked again. All references to the new person in other frames will have their images automatically updated if the image for the person has changed.

3.7 Posting a message

If the user's computer is connected to the Internet, then it is possible to send the message to another user. This is done by first clicking on the Mail-To command button. In the window that comes up, along the top are three text fields. The user types her or his own email address in the first. In the second belongs the email address of the person to whom the message is to be sent. The third field can be ignored when sending.

Then the user can click on the Send command button. This causes three things to happen. First, a key code is generated that will be used to identify this particular message when it is on the Vedo-Vedi server. Second, the message is transmitted to a server computer at the University of Washington in Seattle, where it is stored for subsequent pickup. Third, an email message containing the key code is sent to the recipient to notify her or him that a message has arrived and to enable her or him to obtain it.

3.8 Picking up a message

When the recipient has obtained the email message with the notification and key code, there are two ways that the message can be retrieved. The first way requires that Vedo-Vedi be run as an application on recipient's machine. The recipient starts up Vedo-Vedi, clicks on the Mail-To command button, enters the key code into the third text field, and clicks on the Get command button. The message should soon appear in the script panel.

The other way to retrieve the message is to take the URL in the email message and access the World-Wide Web with a suitable browser. The browser must fully support Java 1.1, as mentioned earlier. Vedo-Vedi will then be run as an applet, and the message will be displayed in the script panel. Starting Vedo-Vedi as an applet can take several minutes, with the exact time depending primarily on available bandwidth through the network to the user's machine.

Once the message has been retrieved and displayed on the recipient's screen, the recipient can compose a reply by deleting all or part of the message and then creating new frames. All of the new-object definitions in the received message are available in creating the reply. The reply can include additional definitions, so that the available visual vocabulary gets gradually built up during a Vedo-Vedi dialog.

3.9 Reading a message

A user can "read" a message in three different ways. One is to look at the frames in the script panel, one after the other, scrolling or resizing if necessary in order to see them all. Another way is to go through the frames as before, but click on the frame image of each one in order to get a translation or expanded view of the frame. (Note, there is one kind of frame, the travelling group frame, that is an exception – clicking on it adds a new member slot for the travelling group). The third way to examine the message is to play it as an animation. An experienced user can most quickly read a message using the first method. The second method permits selective translation for a semi-experienced user. The animation is best for the new user, since all frames are shown with additional detail, with transitions, and with natural language translations in subtitles.

4 Example Messages

In this section, we present a variety of example Vedo-Vedi scripts, in order to explain how various features operate.

4.1 A Short Trip

Our first and simplest example is shown in Figure 1. Two transportation events are described, the first

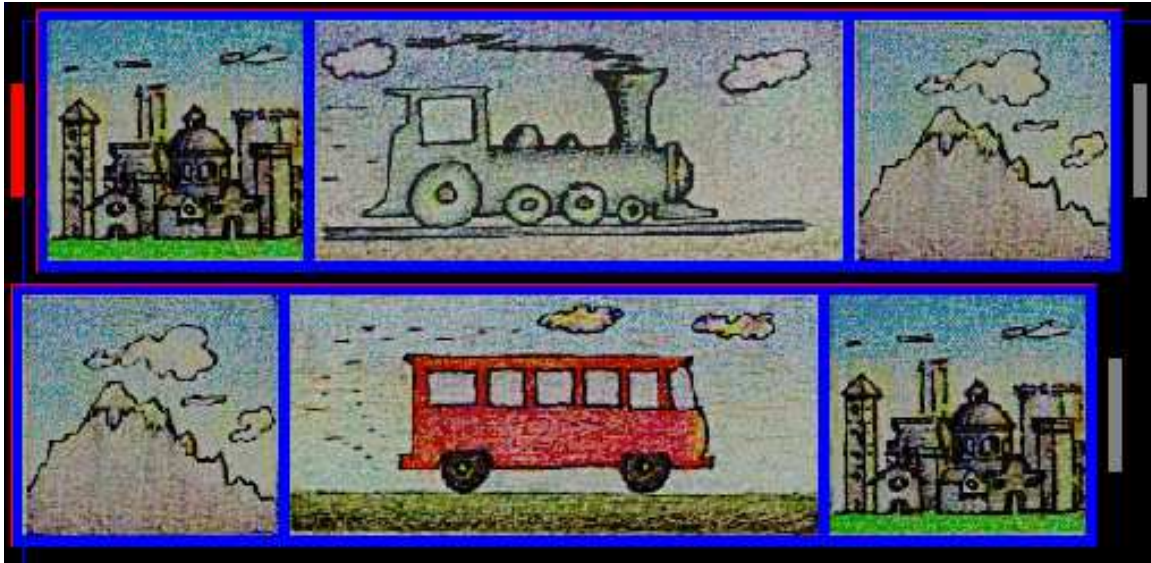


Figure 1: Script describing a short trip. Each frame represents one transportation event.

involving a train and the second involving a bus. Each of these travel frames has two slots: one on the left representing the source or place of departure, and that on the right representing the destination. The train and bus images are the frame images, which can be clicked on to obtain translation views.

4.2 Side Trips

Now we present a more elaborate example involving three additional features: introduction of new places, introduction of travelling groups, and side trips. This script is shown in Figure 2.

Three places are introduced: Seattle, Vancouver, and Victoria. (The names are not visible in this standard script view, but could be seen in the expanded views of frames, on the menu, and in the animation.) The main travelling group is introduced; it contains three people. The trip from Seattle to Vancouver is made by bus. All three members of the group visit Vancouver. Next a side trip to Victoria by boat begins with a travelling group consisting of only two of the members of the main group. At the conclusion of the side trip, the main trip continues, and all three travellers return to Seattle by bus.

4.3 Dreams and Mind Trips

Simple narrations can be presented as representations of fact. However, they can also be presented as thoughts or dreams. Thoughts and dreams can themselves be parts of narrations including thoughts and dreams, and so scripts can essentially be embedded within other scripts.

A dream is represented by a sequence of frames that starts with a Begin-Dream frame and ends with an End-Dream frame. The dream is animated in a “subscreen” of the animation window. An example of

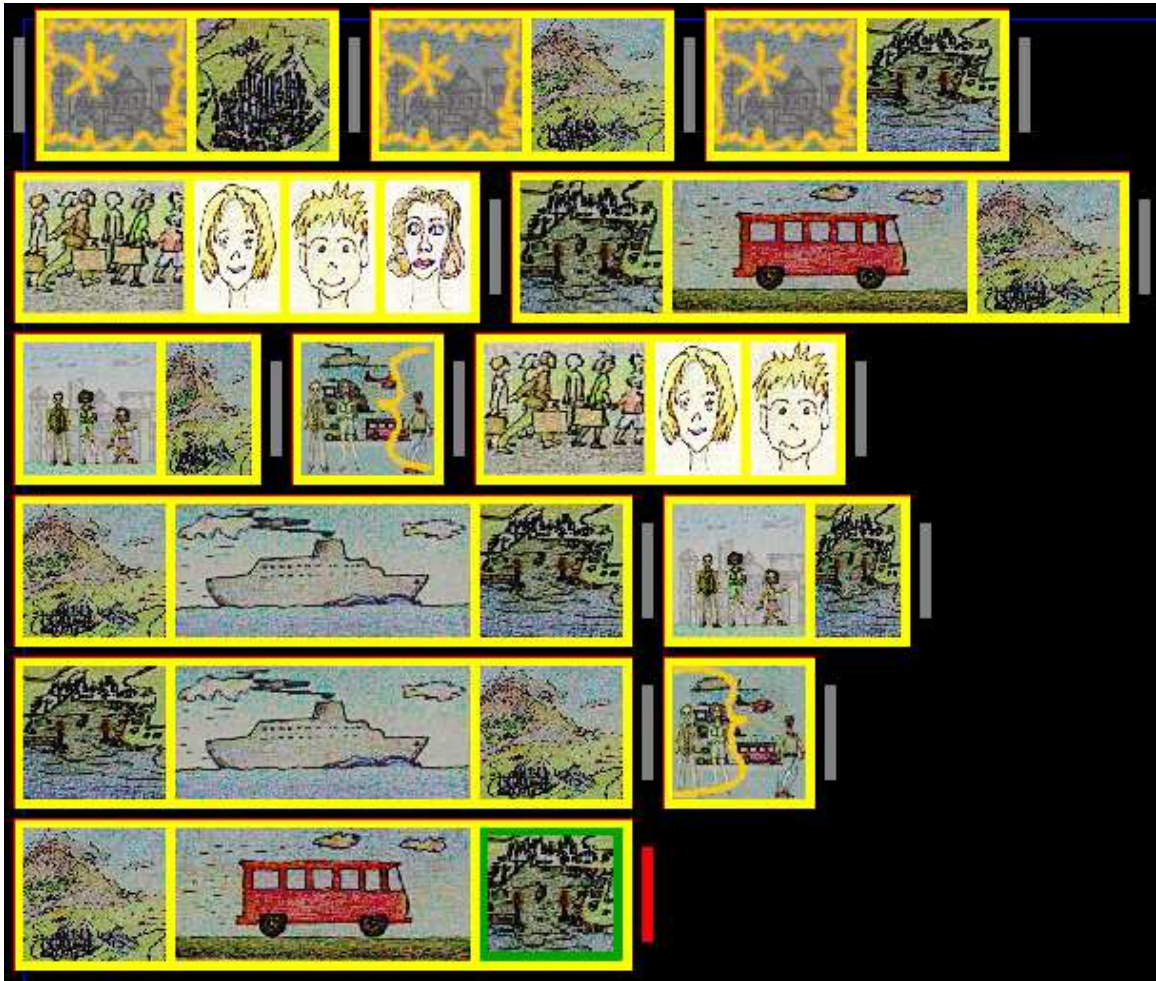


Figure 2: Script describing a main trip, a side trip, and new places. The first three frames introduce new places, the fourth introduces a traveling group of three people, and the seventh begins a side trip.

a script that contains a dream is given in Figure 3. This script begins by describing a trip from Seattle to Tacoma. After going to a hotel, the narrator has a dream which consists of a trip from Seattle to Hawaii by boat. At the conclusion of the dream, the narration transitions back to the current location, and the main trip continues. A still image taken from the animation of this script shows how Vedo-Vedi animates the dream in a subscreen of the main window (see Figure 4).

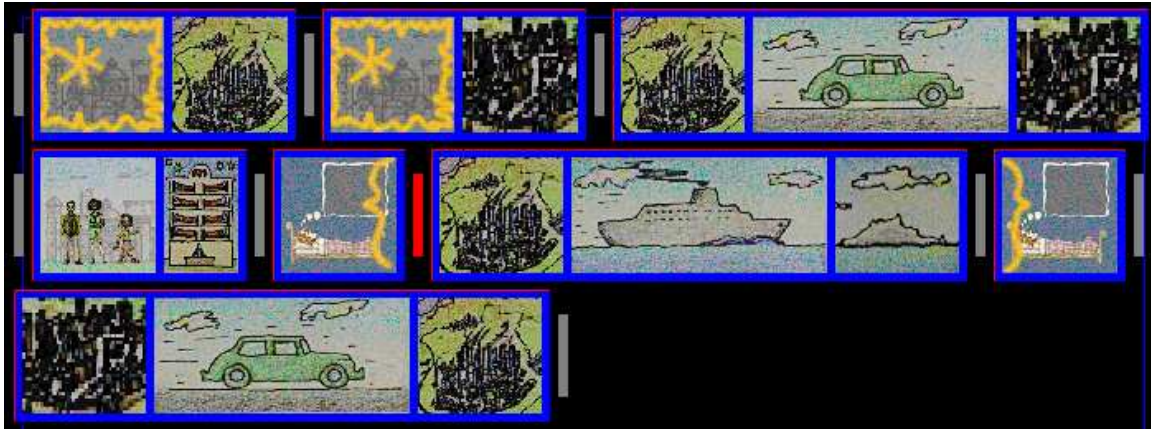


Figure 3: Script that includes a dream. The Begin-Dream and End-Dream frames are the 5th and 7th, respectively, with the boat cruise frame making up the body of the dream.

4.4 Souvenir Collection

In addition to new people and new places, Vedo-Vedi supports the definition of new portable objects. These objects can serve as souvenirs in stories; they can be acquired by the agent who tells the story. Figure 5 shows a script that includes frames that describe the acquisition of several objects. At the end of a story including acquisitions of objects, the animation displays the full set of collected objects with an appropriate subtitle. For this script such a presentation is that shown in Figure 6.

4.5 Dates and Time

Normally one of the simplest aspects of the events in a story is their chronology. This means when each event happened, or at least, what order they occurred in. Yet chronology can sometimes be confusing in a story, because details are omitted, or there are multiple temporal reference frames to keep track of. Vedo-Vedi supports four temporal notions: absolute dates (by the calendar), relative days (from the start of the trip), absolute time (by the clock), and relative time (in terms of amounts of time passing). The script shown in Figure 7 illustrates two of these temporal notions: absolute clock times and relative dates. As time passes (as indicated by the hourglass frame), the clock is updated. As days pass, a symbolic weary traveller gradually makes progress from left to right in front of the setting sun (see Figure 8). In contrast, if an absolute date is established using a calendar frame, the weary traveller animations are not shown, but an updated calendar is displayed as the date changes. In addition to these basic notions, one can have more than one time frame in a message. For example, a dream or a thought has a time frame independent of that of the surrounding script.

4.6 Where is That?

Using a new-place definition frame, the user can create a new place with its own icon and textual label. However, there is more to a place than a look and a name. At least there is location. In Vedo-Vedi, it is

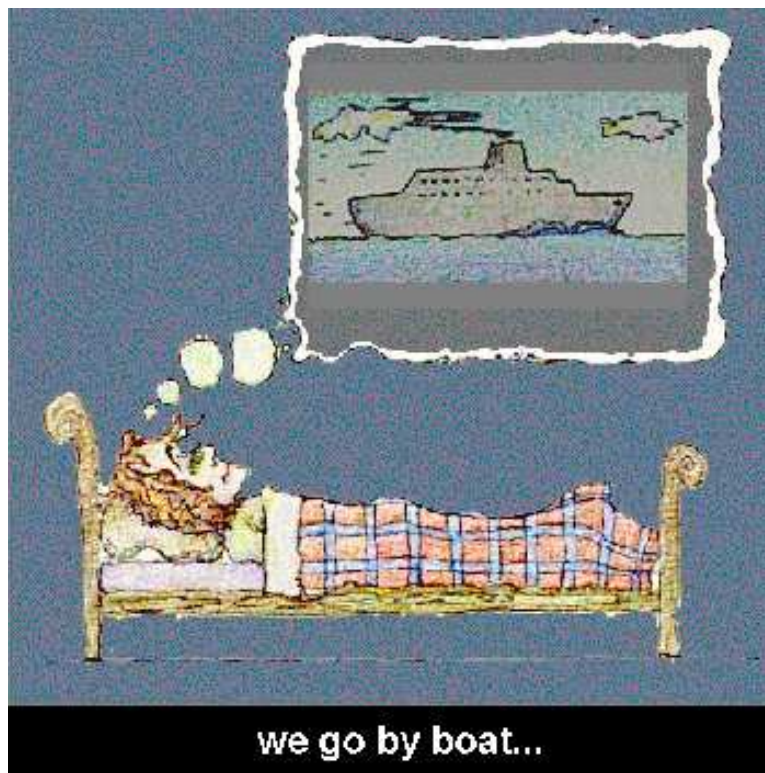


Figure 4: A snapshot of the animation of the dream within the script in the previous figure. The cruise event is presented within a subscreen.

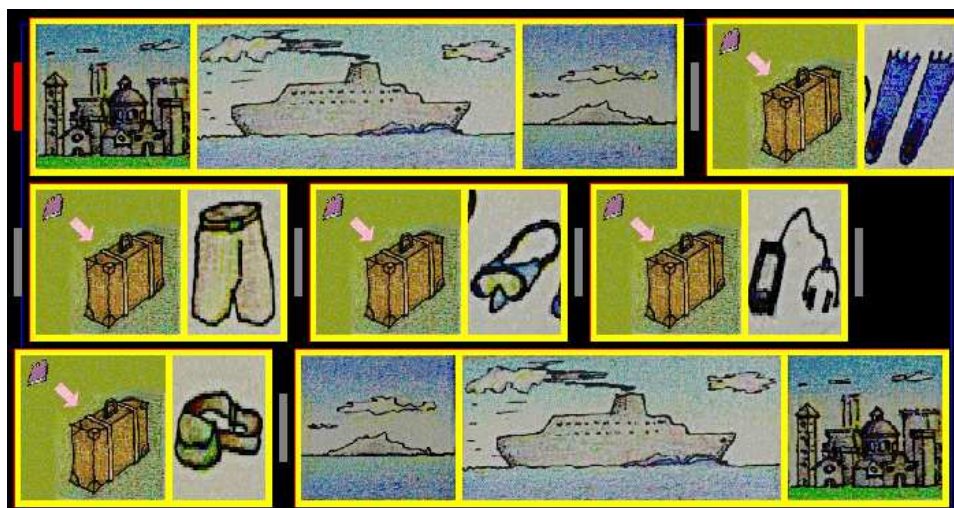


Figure 5: Souvenir acquisition events in a script that describes a trip to Capri.



Figure 6: Snapshot of the animation taken at the end of the story showing all the souvenirs collected.

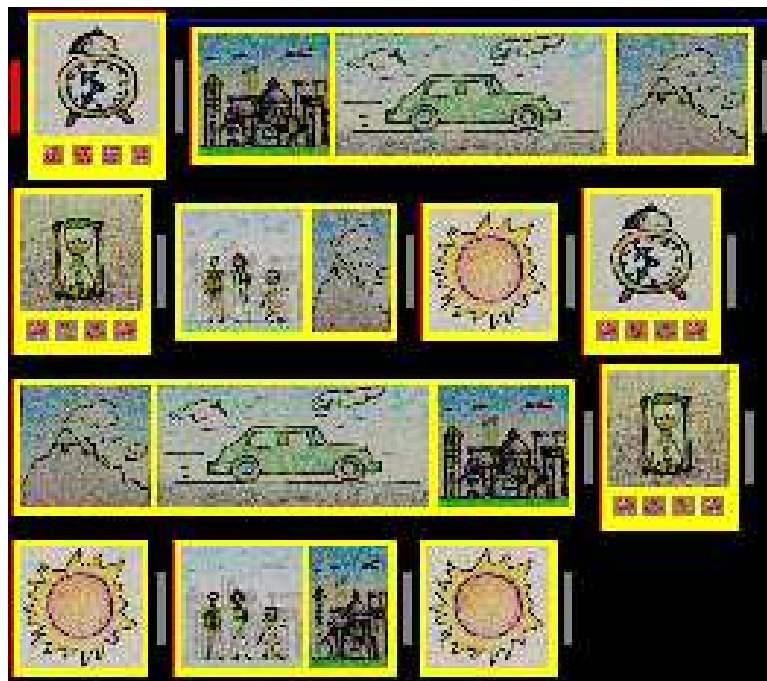


Figure 7: An example of a story involving manipulation of time and dates.



Figure 8: Animation of the transition from one day to the next, shown when absolute calendar dates are not specified.

possible for a script to include an explanation of what region (or other place) a new place is a part of. No mechanism is currently provided to express other geographical information such as the shape, size, latitude, or longitude of a place. However, something about these attributes can usually be inferred from context. In Figure 9 is a script that expresses inclusion relationships among those places. Each “inside” frame expressed the fact that one place is inside of another place. For example, Milan is in Italy. The “Show Space Tree” frame at the end of this script provides, in the animation, a visual summary of all the places that have been established as directly within Italy. A snapshot of this portion of the animation is shown in Figure 10.

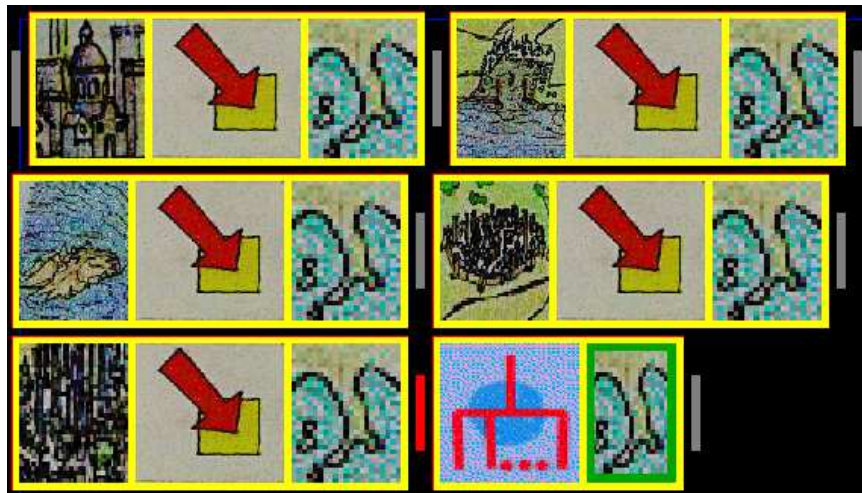


Figure 9: Script that introduces new places and describes their containment relationships.

The simulation code includes a method to determine whether one place is inside another by transitivity of the “Is In” relation. We haven’t yet put this to use in the system, but in principle it could support inferences about likely means of travel by inferring the city, country, or continent containing the current place.

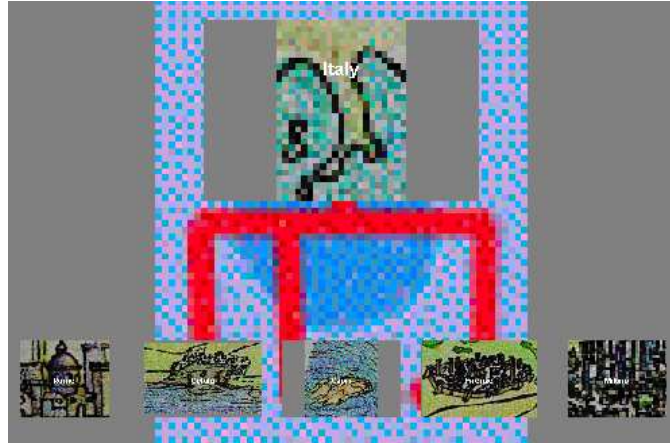


Figure 10: Still image during the animation of the relationship between Italy and five cities that it contains.

5 Conclusions, Limitations, Future Work

Our work on Vedo-Vedi has focused our attention on the challenges of providing a system for concrete semantics for use in multilingual and visual communication about everyday phenomena. The current implementation is limited to some of the most basic aspects of time, space, transportation, modes of narration, people and objects. The language translation facility does not yet use a dictionary of nouns, nor does it yet support any languages that use non-latin alphabets such as Cyrillic or Chinese. Our framework, however, is consistent with support for these languages and translation features. Thus, we feel that it successfully demonstrates the key elements of a computer-based visual language for international communication.

Acknowledgments

We would like to thank Stefano Levialdi, Luigi Cinque and Paolo Bottoni of the Pictorial Computing Laboratory, Rome, Adam Carlson of the University of Washington, Genny Tortora of the University of Salerno, Margaret Burnett of Oregon State University, Allen Ambler of Kansas State University, and S.K. Chang and Robert Korfage of the University of Pittsburgh for their help or encouragement on this project.

Software Availability

The applet version of Vedo-Vedi can be found at the following URL:

<http://trillium.cs.washington.edu:8080/tanimoto/vv/applet/run-vvap.html>

It can take several minutes to initialize version 1.0 of the applet, due to the volume of image data and class files that must be downloaded.

References

- [1] Beardon, C. 1993. Computer based iconic communication. In Ryan, K., and Sutcliffe, R. (eds.) *AI and Cognitive Science '92*, London: Springer-Verlag, pp.263-276.
- [2] Repenning, A. 1994. Bending icons: Syntactic and semantic transformations of icons. *Proc. VL'94*, held at St. Louis, MO, Oct. 4-7. pp.296-303.

- [3] Tanimoto, S. L. and Runyan, M. S. 1985. PLAY: An iconic programming system for children. In Chang, S. K., Ichikawa, T., and Ligomenides, P. A. (eds.) *Visual Languages*, New York: Plenum Press, 1986, pp.191-205.
- [4] Tanimoto, S. L. 1997. Representation and learnability in visual languages for web-based interpersonal communication. *Proc. VL'97*, held at Capri, Italy, Sept. 23-26, pp.2-10.
- [5] Van Reeth, F., and Flerackers, E., 1990. Visual programming in a computer animation environment. *Proc. VL'90*, held at Skokie, IL, Oct. 4-6, pp.194-199.