A Visual Agent for Performance Graphics

Fred Lakin
The Performing Graphics Company
512 Military Way, Palo Alto, CA
lakin@pgc.com

Abstract

Visual Agents are software entities which assist people in performing graphical tasks. One useful and interesting graphical task is to make a text-and-graphic record of a group meeting. When the recording is done live in view of the group on a large display, such as on a whiteboard, it is a performing art which can be called "Group Graphics." During group graphics, collaboration is enhanced because the group members can see the record as it is being made, offering suggestions and corrections. And if graphic recording of this kind takes place on a large computer display, then visual agents can be brought into play. Presented in this paper is a visual agent which acts as a whiteboard assistant for group graphics, helping a person to graphically record the conversation and concepts of a working group on a large display.

Context: Performance Text-Graphics on the Big Screen

The long term goal of this work has been to construct a visual medium which will support the improvisational creation of text-and-graphics live for an audience – in short, to allow a human to "perform" a text-graphic image.

In general, there are many possible styles of performance graphics: abstract (live version of abstract film), concrete, poetic, etc. In particular, the style of performance graphics for which this medium has been designed is called "Group Graphics." It has the following features:

- One person is writing&drawing for a group (multiple operators also possible).
- The graphics are improvised live from group discussion.
- The purpose is to facilitate the working group in its task. Therefore, the images are representational: text and graphics which directly represent the conversation and concepts of a working group.

The Group Graphics style of performance graphics is an applied art: it has a definite function, which is to serve the group in developing and displaying its concepts.

A group graphics performance takes place on a big screen which can be easily seen by all members of the working group. It must be emphasized that both the physical size of the display and the amount of information it can present are much greater than what is normally used for "visual presentations." It means that there is a *lot* of graphical real estate available for the performer (and her visual agent) to play around in. For example, the LCD projection pannel currently used for a group graphics performance can display about five times the amount of information which can be shown in a viewgraph (using smallest readable characters as the measure, 7000 versus only 1500 for viewgraphs).

A Visual Agent to Assist Group Graphics

Previous work has been done on constructing an interface for improvising graphics in the service of facilitating cognition for both groups and individuals. Initially much of the effort simply went into making the human-machine interface agile and rich enough to support this kind of activity [Lakin80a, 80b, 88, 89]. The resulting medium can be thought of as an extremely quick graphics editor with no pop-up menus to distract the audience; it is called the **vmacs** system for performance text-graphics.[1]

However, from the very beginning of the effort, it was clear that group graphics was an extremely demanding task. Manipulating text-and-graphics at the speed of group thought is *hard*. Although gifted human performers can and do manage it on recalcitrant media such as large sheets of paper, it was clear that most performers could use some assistance. So in parallel with the interface design, effort also went into development of computational engines for processing text-graphic activity, both the static images resulting from the activity as well as the actual moment-to-moment dynamics of the activity itself [Lakin80a, 80c, 83, 84, 87, 90, 92].

This processing of text-graphic activity is the underlying technology which permits the functioning of visual agents. The basic idea is that, *first*, the visual media system's interface permits the performer to generate spatial-temporal patterns which are in fact the performance. This is just a way of saying that text-and-graphics are moving around on the screen over time. And then, *second*, because this performance takes place in a medium which has an underlying computational model, those same spatial-temporal patterns can be processed immediately by the system as they occur. This processing allows software modules – visual agents – to recognize certain visual patterns and assist the human in the task of creating/manipulating those patterns.

The result is that the system is dynamically processing the performance for the purpose of helping the human do the performance. This permits a style of human-machine interaction called Graphic Dialog.

Graphic Dialog with a Visual Agent

In the graphic dialog situation, the human is generating text-and-graphics, and the visual agent is watching what the human is doing. The visual agents's role can be fairly passive, just measuring aspects of the performance, or more agressive, actually *intervening* in the human's manipulation of text-and-graphics, in a sense becoming a *co-operator* of the display.

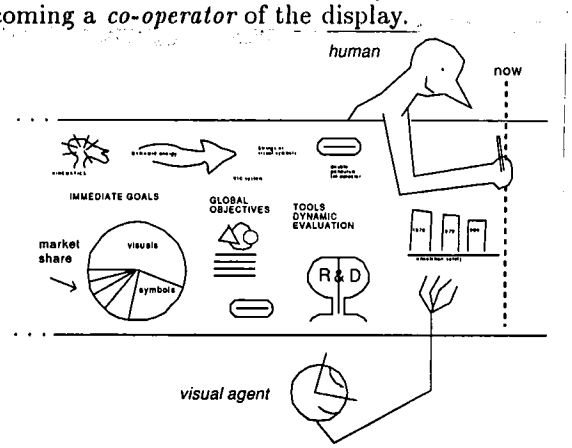

Figure 1. Visual Agents in **vmacs** have complete access to user actions and the state of the visual world in the midst of text-graphic performance.

Simply put, a graphic dialog occurs when two entities are trading text-graphics back and forth. entityOne creates some text-and-graphics, and, by putting them on the screen, submits them to entityTwo, who looks at the text-and-graphics, and then returns some text-and-graphics which entityOne looks at, and so on and so forth. In this case, one entity is the human and the other is a visual agent.

When a visual agent is assisting group graphics, the human generates text-and-graphics representing the group's cognition, and the agent intervenes on the display from time to time. For ease of reference, the visual agent for group graphics has been named "Dave."

When Dave goes into action, he appears on the screen as an eye (for seeing) and an arm with hand (for manipulating). Dave's intervention is triggered when the human creates certain visual patterns in the course of group graphics.

Patterns Dave currently recognizes include arrangements involving lists of textual items with hand-drawn bullets; pieces of text connected to any text-graphic object by a hand-drawn straight arrow; two text-graphic objects connected by straight lines (figure 2); and hand-drawn hollow arrows, with text inside and at the tail end, pointing to any text-graphic object.

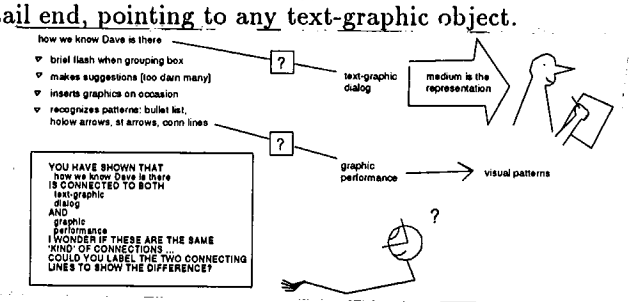

Figure 2. Dave's response when the user draws two connecting lines.

These patterns are used in a visual language developed for group facilitation by a talented graphical performer [Sibbet80]. When the human creates a pattern which is recognized, Dave then generates a text-graphic response and this object is displayed on the screen. The response is appropriate both to the general type of visual pattern which triggered it, and to specific elements in each individual triggering pattern (if part of the triggering pattern is an element that Dave cannot deal with, such as an arbitrary cartoon, then that element is simply copied directly into the response using a template based generation technique).

Visual Agent Architecture

At this point in the development of visual agents, it is felt that a relatively simple mechanism reacting appropriately to complex human activity in a rich graphic enviroment is a useful first step. When that behavior has been acheived, then it will be relatively easy to add sophistication to agent responses using standard technologies (see Future Work). **vmacs** provides a basic framework for visual agent implementation:

- Each user action is represented symbolically.
- Agents can examine each user action before and after execution.
- Agents can examine state of visual world before and after each user action.
- Agents can *change* state of visual world before and after each user action.
- Agents can use mechanisms based on visual parsing and logic programming.
- Agents can communicate with other agents through a visual blackboard which accepts multi-media postings (visual as well as symbolic objects, figure 4).

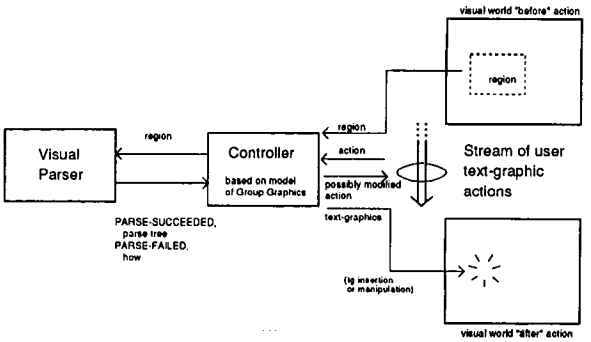

Figure 3. Visual Agent architecture showing relation between Controller and Visual Parser.

At this time 8 different visual agents have been implemented:

- Dave the group graphics assistant.
- A Fitt's Law daemon to group drawn lines during sketching.[2]
- A logic-based activity characterizer for conceptual design drawing.
- A logic-based pointing trainer (cursor control).
- A marking trainer (drawing skills).
- A visual language trainer for aphasics.
- A "graphic speedometer" which measures text-graphic manipulations per second.
- An agent which maintains spatial connectivity between straight arrows and their objects.

More than one agent can run at the same time, jointly administrating the same performance. The order in which agents have access to user actions can be specified. 7 of the 8 agents have been run simultaneously.[3] Agents communicate with each other through the multi-media blackboard; image parsing allows agents to extract symbolic and visual information from the postings.

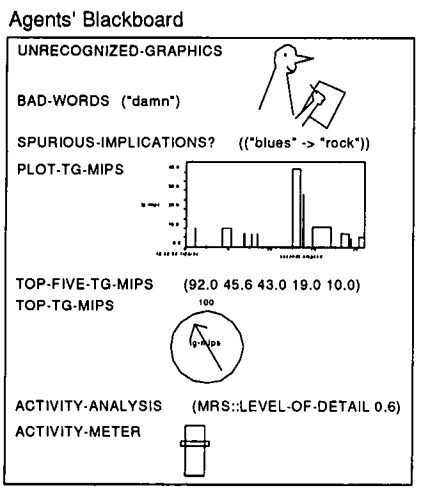

Figure 4. Visual Agents in **vmacs** communicate with each other through multi-media postings to a visual blackboard.

Implementation of Dave the Visual Agent

The underlying framework of the **vmacs** system supports humans in text-graphic performance, and also allows agents as performance administrators to observe each atomic user action committed by the human and its effect on the visual world (figure 3). When Dave is running, the Controller uses a simple model of the visual activity of group graphics in conjunction with on-the-fly Visual Parsing which watches for patterns signaling intervention situations.

The visual parsing is grammar-driven, utilizing a visually notated context-free grammar [Lakin87]. This allows Dave to recognize objects like hollow arrows, bullet lists, etc. In order to control Dave's pattern recognition behavior, one simply writes&draws the visual rules that he uses in recognizing when he should intervene.

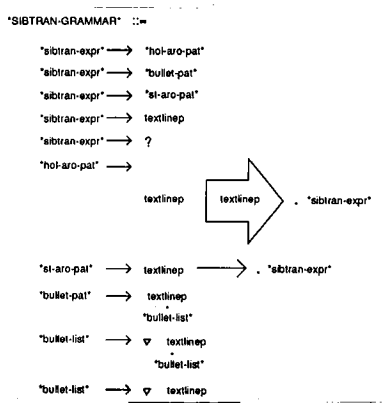

Figure 5. Machine-readable Visual Grammar Notation for the context-free grammar used by Dave in parsing user generated text-graphic patterns.

Features of Dave's implementation:

- Easy to Control: behavior is specified by visual rules (visual grammar driven).
- Permissive: user can write&draw whatever she wants; when Dave happens to see a pattern he recognizes, then he intervenes.
- Non-Exclusive: Dave can co-operate with other agents which may be administrating the same performance (for instance, the "graphic speedometer" agent); all agents communicate through the visual blackboard.
- Extensible: pixmap images and animations are available to the performer now; in the future the visual grammar will be extended to include those types of user actions.

Related Work

Maes and Kozierok have described an intelligent agent which employs machine learning to assist humans in the tasks of meeting scheduling and sorting electronic mail [Maes93]. Like the work presented here, that agent performs in real-time; but the time window is weeks or months, not just the current user session. Also, their agent operates only in a text-based domain.

Lieberman describes an agent which helps the user with graphical layout tasks [Lieberman93]; his "Mondrian" system does programming by example in a visual domain. Like the work presented here, his system does spatial parsing. However, the parsing in Mondrian is not accomplished using context-free visual grammars. Another difference is that **vmacs** agents do not use programming by example.

Current Status

A live Group Graphics performance was given at SGI in March of 1993 [Lakin93a] using a super VGA LCD projection pannel (1024x768) on an 8x6 foot screen. Dave the visual agent made his functional debut in a live demo at IBM's USER workshop in June of 1993 [Lakin93b].

Performing graphics live on a big screen (via LCD projection pannel) is the presentation format of choice for this research. If the work is selected for the symposium, I hope to have the opportunity to do a live performance.

The **vmacs** system with the visual agent architecture is implemented using Common Lisp and X windows; it currently runs on a Sun Sparcstation or any other platform running CL and CLX. A Windows NT port is underway.

Future Work

The Performing Graphics Company was formed for the purpose of giving live computer graphic performances. At the moment performances are service oriented: facilitative graphics for working groups. We hope in the future to branch out into entertainment oriented performances as well. It is intended that Dave participate more fully in both situations.

In the system described here, we claim that responsiveness, usability and verisimilitude for visual agents can arise out of a relatively simple mechanism reacting appropriately to complex human activity in a rich graphic enviroment. Agent behavior is currently determined by a Controller which uses a very sparse procedural model of Group Graphics. Now that the system can function in real-time to administrate live performances, construction of more sophisticated control regimes is underway. Work on a logic based descriptive model of Group Graphics has been started. When finished, the logic model will plug into the Controller in place of the procedural one, providing more varied and finely grained responses to user graphic activity.[4] We are also considering a Controller design utilizing machine learning techniques like those described in [Maes93]; such a Controller would use data from visual parsing to build and continually update a spatial performance profile for the user.

Acknowledgements

To Ted Selker for encouraging me in resuming investigation of the visual agent theme, and for many useful suggestions in preparing the first public demo of Dave for Ted's USER (*New Paradigms In Computing*) WORKSHOP at IBM Almaden in June 1993. And to the hardy few who graciously suffered being rehearsed at and then offered invaluable feedback: Jeff Aldrich, Peter Broadwell, David Brown, Tony Chang, Ron Fischer, Becky Fuson, Wade Hennessey, Scott Kim, Henry Lieberman, Rob Myers, and Greg Nuyens.

Footnotes

[1] "vmacs" and "Performing Graphics" are trademarks of The Performing Graphics Company.

[2] Implemented to support design research of David M. Cannon at Stanford.

[3] The marking trainer preempts control of the interface in a way that cooperation with other agents is not possbile.

[4] The Controllers for several of the other visual agents already employ logic programming, including the pointing trainer and the design characterizer.

References

[Lakin80a] Lakin, Fred, "A Structure from Manipulation for Text-Graphic Objects", published in the proceedings of *SIGGRAPH 80*, Seattle, Washington, July, 1980.

[Lakin80b] Lakin, Fred, "Diagramming on the Electric Blackboard," video tape, *SIGGRAPH 80*, July 1980.

[Lakin83] Lakin, Fred, "Measuring Text-Graphic Activity", published in the proceedings of *GRAPHICS INTERFACE 83*, Edmonton, Alberta, May 1983.

[Lakin84] Lakin, Fred, "Visual Communication Expert", Public Broadcast television segment on *COMPUTER CHRONICLES KCSM*, as part of show on Artificial Intelligence, March 22, 1984.

[Lakin87] Lakin, Fred, "Visual Grammars for Visual Languages," published in the proceedings of conference of the *AMERICAN ASSOC for ARTIFICIAL INTELLIGENCE*, Seattle, Washington, July 12-17, 1987, pp 683-688.

[Lakin88] Lakin, Fred, "A Performing Medium for Working Group Graphics," chapter in *COMPUTER-SUPPORTED COOPERATIVE WORK*, edited by Greif, Irene, Morgan Kaufman Publishers, Palo Alto, 1988.

[Lakin92] Lakin, F, V Baya, D Cannon, M Brereton, L Leifer and G Toye, "Mapping Design Information," paper for the *AAAI 92 Workshop on Design Rationale Capture and Use*, San Jose, CA, July 1992.

[Lakin93a] Lakin, Fred, "WORKING GROUP DISPLAY ... revisited," 45 minute text-graphic piece performed at Silicon Graphics in Mountain View, CA, March 25, 1993.

[Lakin93b] Lakin, Fred, "A Visual Agent for Group Graphics," talk and demo at the *USER: NEW PARADIGMS FOR USING COMPUTERS Workshop*, IBM Almaden research lab, June 15, 1993.

[Lieberman93] Lieberman, Henry, "Mondrian: a Teachable Graphic Editor. In: A. Cypher ed. *WATCH WHAT I DO: PROGRAMMING BY DEMONSTRATION.* Cambridge, Mass.: MIT Press. Forthcoming.

[Maes93] Maes, Pattie and Kozierok, Robyn, "Learning Interface Agents," published in the proceedings of the conference of the *AMERICAN ASSOC for ARTIFICIAL INTELLIGENCE*, 1993.

[Sibbet80] Sibbet, David, *A WORKSHOP GUIDE TO THE GRAPHIC FACILITATION OF GROUPS*, The Grove Consultants, San Francisco, CA, 1980.