NEW GRAPHICAL REASONING MODELS
FOR
UNDERSTANDING GRAPHICAL INTERFACES

George W. Furnas

Cognitive Science Research
Bellcore
445 South Street
Morristown, NJ 07960

ABSTRACT
This paper aspires to make three points: (1) that certain graphical interfaces are especially easy to learn and use, (2) that special graphical deduction / computation systems are possible, and (3) that perhaps points (1) and (2) are intimately related, i.e., that graphical interfaces may be especially useful because they engage special human graphical reasoning processes.

KEYWORDS: Graphical interfaces, mental models, user models, visual reasoning

INTRODUCTION
This paper will argue for three points. The first is reasonably familiar: (1) Certain graphical interfaces are especially easy to learn and use for novices, and in some tasks, even for experts. The second is likely *not* to be familiar (since it is a new research point): (2) Inherently graphical deduction/computation systems are possible, having rules that map, not from one-dimensional sentences of symbols to one-dimensional sentences of symbols (like symbolic logic), but from picture-like entities to picture-like entities. The third point is: (3) Perhaps points (1) & (2) are intimately related, i.e., graphical interfaces may especially useful because they engage special, inherently graphical reasoning processes that humans have.

This third point cannot yet really be supported. In fact, the major achievement here is that the possibility can even be raised - basically by defining and exploring the space of inherently graphical deduction systems (systems which were explicitly presumed out of existence by Lindsay[1]). Opening up new possibilities for thinking about graphics is important because better design of graphical interfaces for humans will require better understanding of how humans interface to graphics.

In what follows, each of these points will be addressed in more detail. A brief survey will discuss evidence that graphical interfaces can be especially useful, and the difficulty there has been in understanding when and why. This will be followed by a general introduction to graphical reasoning, its recent history and some detail on our current new efforts. Finally, one of these new graphical reasoning tools will be used to model some familiar graphical interactions, providing a new metaphor for thinking about them.

GRAPHICAL INTERFACES ARE SPECIAL
That certain graphical interfaces are special can be substantiated in a number of ways. The most obvious is the phenomenal market success of the Macintosh®, or of video games. Both of these have made tremendous inroads with the computer naive, and the direct manipulation style made famous by the Macintosh has certainly established itself even with advanced computer users.

The explicit empirical evidence is more complex (e.g., good interface design makes the biggest difference, regardless of style [2] [3]) but there are several studies which bear out the superiority of graphical over command-language based interactions for certain broad classes of tasks (e.g., [4] [5]). Insofar as this is true, the question is, "Why?"

Typically when system A is easier to learn or use than system B, one looks for an explanation in the form of "A is simpler than B" where "simpler" means the knowledge and rules needed for performance are fewer or simpler. The knowledge and rules are typically modeled by the GOMS model of Card, Moran, and Newell [6] , or the Cognitive Complexity Theory, production-system models of Kieras and Polson[7].

Karat [4] attempted just such an analysis to compare command-based and graphics-based interactions. He used a Cognitive Complexity Theory (CCT) analysis, and studied a file deletion task. Karat indeed found that the graphical system was easier, and though the CCT analysis explained several effects found, it could *not* account for

---

® *Macintosh* is a registered trademark of Apple Computer Company

the marked superiority of the graphical system over the command-based one. Some of the performance difference they could attribute to error times - whose analysis is largely outside the scope of both GOMS and CCT - but not all. That is, the production system models could not, somehow, capture the specialness of the graphical interaction system.

So what is it that makes these graphical interfaces so good for what they do? Is it just that the sentential representation of knowledge and rules that characterize these systems gain magical advantages whenever they use spatial words like "move", "down", "right"? Perhaps,... but the conjecture here is that models like those used by Karat may be inappropriate for graphical interfaces. In particular there may be something wrong with modelling human use of spatial/graphical interfaces, with linear, sentential representations. Perhaps instead special, spatial or graphical reasoning is being used, based on special spatial/graphical representations and rules. That conjecture depends on the assertion that special spatial reasoning is possible. New research in reasoning indicates that it is.

EXPLICITLY GRAPHICAL REASONING IS POSSIBLE
The role of graphics and images in reasoning is not well understood. Historically, formalists have scorned pictures as at most suggestive, with no place in formal deduction and proof (e.g., [8]). Computation and reasoning have been carried on almost exclusively using "sentential" representations consisting of one-dimensional strings of symbols. Pictures, diagrams and other spatial representations have been at most treated as data (e.g., in image understanding contexts) whose content is to be parsed into sentential descriptions which can then be manipulated in the familiar mode.

The possibility that pictures might warrant a more first-class status in computation and inference has been given serious reconsideration in AI circles only in the last few years. Barwise and Etchemendy[9] , two highly respected formal semanticists from the traditional camp, now argue in an important paper, that deductions based on pictures have no more pitfalls than those based on sentential methods. The pitfalls are just different; but so are the strengths. They argue for, and develop, extensions to standard reasoning mechanisms, creating "heterogeneous" reasoning which uses both modes and is built upon serious formal theory[10]. Taking a different approach, Gardin and Meltzer[11] report a system that uses explicit 2-D graphical representations populated with object-oriented graphical objects obeying local graphical constraints to solve problems in naive physics (e.g., simulating pouring of liquids, or flexing of strings or rods). In discussing the theoretical role of images in inference Lindsay[1] considers solving the following problem using a 2-D grid representation: Take one step north, one step east and then one step south. From the grid it is easy to see that the final position is one step east of the start, yet trying to deduce the same conclusion from formal axioms of the geometry of space would be very difficult. Thus Lindsay argues that images are important because of their ability to implement

geometric constraints automatically and that this constraint-based inference is what distinguishes imaginal reasoning from "calculus-plus-proof-procedure," deductive systems. Furnas[12] argues that spatial constraint satisfaction is indeed a critical asset of imaginal representations, but it empowers (*not precludes!*) a more direct deductive model. This last, graphical/imaginal deductive model, is probably the most completely image-based model yet proposed, and is the one pursued here.

Deduction in Pictures
It is not obvious how to make deductive machinery for a purely picture system. If axioms somehow map from pictures to pictures, instead of strings to strings, important constructs of familiar logics, like variables and quantifiers, have no easy and direct analog. Furnas [12] [13] therefore identifies fundamental properties desired in any deductive system, and seeks other mechanisms, more suitable for images, to obtain them. For example, in an algebra manipulation system, rules like the commutative law "$\forall x \forall y: x + y \rightarrow y + x$" use variables and quantifiers to specify a whole set of low level facts of arithmetic at once: "$1+2 \rightarrow 2+1$", "$7+5 \rightarrow 5+7$", "$3+9 \rightarrow 9+3$", etc. Thus one desired property is the ability to write "high level" rules in a deductive system where pictures, not strings of symbols, are on the left- and right-hand sides of the rules. The important insight is that any such high level rule can be viewed as just a mapping between two sets: the set of things to which the rule can be applied and the set of things which result from the application. The deductive chaining of high level rules, another desirable property, is then equivalent to the partial composition of these mappings. Under this interpretation, natural generalizations of variable binding and unification emerge.

While one way to specify sets and mappings is with the standard devices of variables and quantifiers -- the sets are generated by all the possible substitutions of values for the quantified variables -- other strategies are possible. For example, collaborations with Karen Lochbaum have explored the specification of a set as the formal language of a formal grammar, but using 2-D array grammars [14] [15] instead of familiar string grammars. High level rules (mappings) are then defined by transformations of the spatial parse trees.

The BITPICT deduction system
Another strategy for specifying picture sets and mappings is used in a particularly simple formal system, the BITPICT system. Though it is not very powerful, the BITPICT system has several interesting formal properties, and has proven useful in giving a better sense of what picture deduction might be like.

The architecture of the BITPICT system is similar to that of standard production systems[16] [17] [18]. It has a set of rules that interact via a shared blackboard. Each rule has a left-hand side that constantly looks at the blackboard for a match. When a rule finds a match, its right-hand side describes how to change the contents of the blackboard. The change is made, and the process iterates. The rules are engineered so the evolution of the blackboard solves
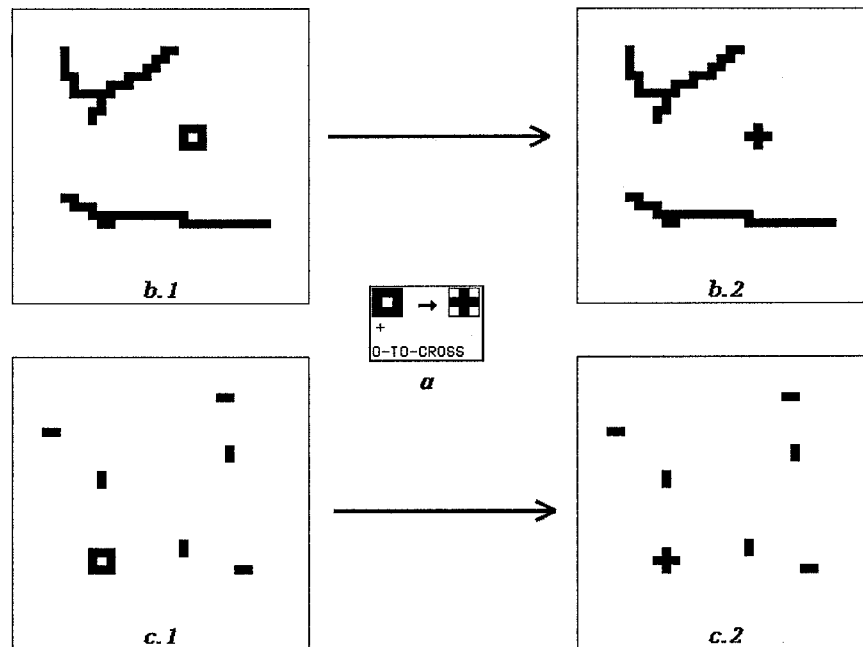
Figure 1. (a) A BITPICT rule mapping the 3x3 grid of pixels containing a circle to the 3x3 grid containing a cross. (b.1) and (c.1) show pictures in the left-hand set of the rule. (b.2) and (c.2) show the corresponding pictures in the right-hand set.

the intended problem.

Unlike standard production systems, where the structures on the board and in the rules are typically strings of symbols representing predicates, in the BITPICT system they are simple picture fragments. Picture fragments are matched and replaced and problems are solved by the fragment-by-fragment evolution of the picture on the blackboard.

More specifically, the universe of pictures considered by the BITPICT system are bitmaps, i.e., regular grids of picture elements (pixels) that are either black or white. A *bitpict* is a small such bitmap, and is taken to specify the set of all bitmaps which contain that bitmap fragment somewhere. A *bitpict rule* is pair of conformal bitpicts, and associates pictures in their corresponding sets that differ only in that the left-hand bitpict has been replaced by the right-hand one. Thus in Figure 1, the BITPICT rule in (a) associates a 3x3 grid of pixels containing a circle with another 3x3 grid of pixels containing a cross. This is taken to associate corresponding pictures in their two respective sets (e.g., b.1 with b.2, c.1 with c.2.) Description of bitpict rules in terms of these sets and mappings is important for deriving its formal properties, but it suffices for discussion here simply to think of them as graphical search and replace rules, where matching can allow, variously, translation, rotation, and reflection invariance.

A BITPICT Spatial Problem Solving Example: Counting the Tangled Forest

Before showing how the BITPICT system can be used to represent graphical interface interactions, it is useful first to get a sense of how it can do general spatial problem solving using purely graphical deductions. In Figure 2, the

problem is to count the disconnected components in a tangled forest of bifurcating trees. The task is not easy to represent in a sentential form, yet has a very natural solution with bitpicts.

Part (a) of the figure shows a sample tangled forest. The first set of rules, (b), simplify the problem, basically by nibbling back the tips of the branches: cutting back straight, corner, and "T" sections respectively. The rules neither create nor destroy connected components, and so each component is gradually reduced to a dot. A sample intermediate state in this process appears in b.1, the final state in b.2.

The next pair of rules, (c), move the dots down and, when they hit the bottom of the window, to the right. An intermediate state for this phase appears in c.1, where seven dots have been moved to the lower right corner, one is in transit at the middle bottom of the window, and four remain uncollected. Ultimately they are all canonically aligned in the lower right corner as seen in c.2. Again the invariant of number of components is preserved, and the problem of counting trees has been reduced to counting these neatly arrayed dots.

The final rules, (d), count the aligned dots by converting each to a vertical bar (an "I") and from there to Roman numerals, with rules for simplifying (e.g., five I's to a V), maintaining alignment (e.g., shifting V's and X's right as needed), and sorting (e.g., VX to XV). The final result is the Roman numeral, XII, indicating the number of trees in the original tangled forest.

The critical computational point is that the evolving state of the blackboard is governed by the picture-to-picture mapping rules, accomplishing a kind of imaginal deductive problem solving with no need for underlying
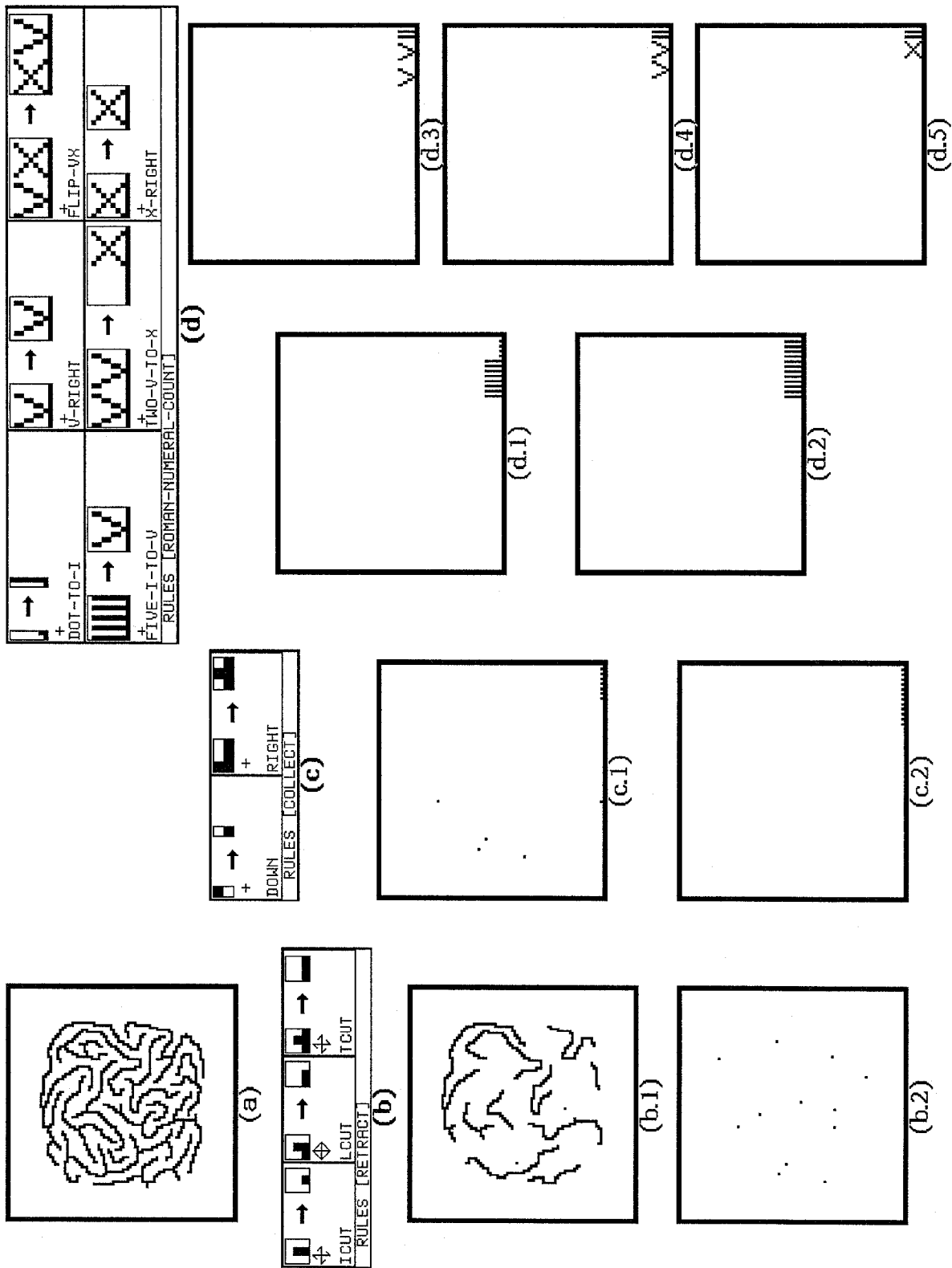
Figure 2. Counting the tangled forest. (a) A tangled forest of bifurcating trees. The rules in (b) nibble at the tips of straight, corner "L", and "T" sections, reducing each component (see b.1) to a dot (see b.2). The rules in (c) move the dots down, and to the right (c.1) into the lower right corner (c.2). The rules in (d) count the aligned dots in Roman numerals, yielding the answer, XII.

sentential representation.

The critical psychological point is that it is now at least conceivable that some mechanism of this general flavor (certainly one more sophisticated than BITPICTs) may be in peoples' heads, and brought to bear on visual/spatial problems.

Finally, the critical point for human-computer interface research is that such imaginal deduction mechanisms may be an important tool in thinking about graphical parts of humans' interactions with computers.

GRAPHICAL REASONING FOR GRAPHICAL INTERFACES

If graphical reasoning systems are to be a useful tool for research in human-computer interaction, they must at least be adequate to capture the behavior of graphical interfaces. In general, good approximations seem quite easy, even with the simple BITPICT system. In video format Furnas [19] presented BITPICT renditions of graphical rules for Roman numeral addition, for tracing logic values through logic circuit diagrams, for moving a file-icon to a trash can, and for PacMan. The file-to-trash example is recreated here on paper, together with a new example, of rules for a typical "paint" program interface.

Note that the versions of the rules systems shown here are somewhat simplified. They do represent the basic knowledge needed for manipulating the interface, and are sufficient for the example interactions shown in the figures. In general, however, strategic aspects (e.g., which file to trash, via what path; what to paint, in what order) involve more complex AI processes which still seem to be representable by additional graphical rules within the framework. They are finessed here by simple conflict resolution strategies in the production system.

Moving a File into the Garbage using Graphical Rules

The first example is a mockup of a simple system which has the graphical knowledge necessary to do file deletions using the interaction style familiar in the Macintosh® interface.

Figure 3 shows a simple four rule system that "knows", in pictures, how to select a file icon and move it to the garbage can. The first rule selects the file, setting it to inverse video. The second rule moves the selected-file icon to the right. The third moves it down. These two rules will often conflict, for example in the starting configuration, where both rules match. For conflict resolution, one may assign explicit priorities, here indicated by the small numbers under the arrows in the rule windows. Thus the FILE-RIGHT rule, having a higher priority, will take precedence, and the file icon will be moved over to the right, until it runs into the upper-right, unselected-file icon. At that point the FILE-DOWN rule will get its chance, moving the selected-file icon down, until it is clear. Then the icon will be moved rightward again to the edge of the window, and finally down again towards the garbage can. When it reaches the top of the garbage can, the INTO-GARBAGE rule will trash the file.

Painting with Graphical Rules

This second example is a simple system containing the graphical knowledge needed for typical "paint" programs. Such systems have a paint brush which soaks up paint from an iconic paint source and will fill closed regions which it thereafter touches.

Figure 4 (a) shows the rules needed. First is a rule, shown in closeup in (b), that tells how the gray paint (implemented in detail by a checkerboard texture) can fill adjacent white space. Other rules tell how the brush can move, dip into the paint can, and touch white regions. The sequence in (c.1)-(c.14) shows the rules in action. Note that, given the paint-flow rule, paint can be made to flow in and out of the brush by temporarily breaking the boundary between the brush and an adjacent region. This device is needed since the BITPICT system currently inhabits a purely 2-d world, with no notion of a superposing layer in which the brush can move and touch down at will. Multiple layers are currently being added to the system using different bitplanes of a color buffer.

CONCLUSIONS

The fundamental lesson to be drawn here is that explicitly graphical reasoning and deduction systems do exist and can capture important aspects of using graphical interfaces.

Note that the existence of special human spatial reasoning mechanisms could not itself explain the relative ease of graphical versus command based interactions. It could at most explain why previous explanations, based purely on systems of sentential representations, have had trouble. The task now becomes to understand what is easy and hard in each system of computation, and to draw interface conclusions accordingly. Further understanding of graphical computation is thus of some priority.

From a practical standpoint, there may be ways in which artificial graphical reasoning systems might directly support better interfaces. Consider the case of logic circuit analysis, a domain where humans and computers have typically differed in their preferred representation. Typical computer programs use lists of gate interconnections; humans prefer diagrams. Furnas [19] showed a system for graphical logic circuit analysis that, like humans, works directly from the diagram -- tracing logic values along wires and through gate icons to deduce behavior. If one were to design a human/computer collaborative system, where the computer and human hope to interact during this circuit understanding process, it might make a lot of sense for them to share graphical representations and rules. Then their intermediate states would coincide, and be available for comparison and consultation. Any such advantages would presumably be similar for other sophisticated graphical tasks.

This paper has avoided one of the dominant "explanations" of the success of graphical interfaces, namely their typical "direct manipulation" character [20] [21]. The question is, what is direct manipulation? The simplest answer is that it contrasts with more verbal (e.g., command oriented) interfaces. That is, direct manipulation is visuo-motor
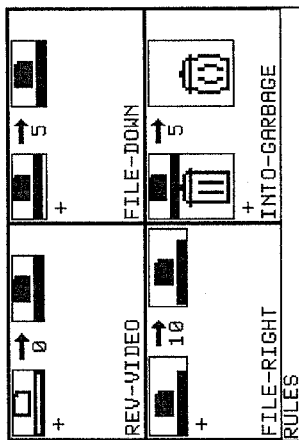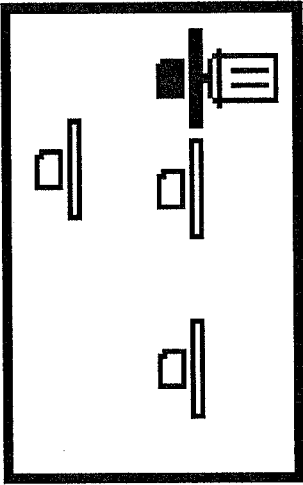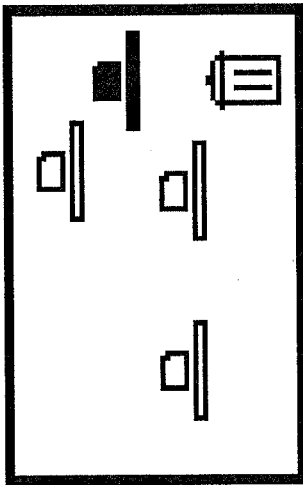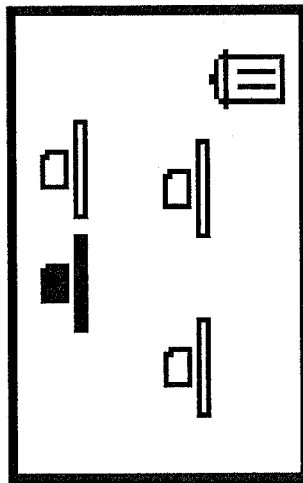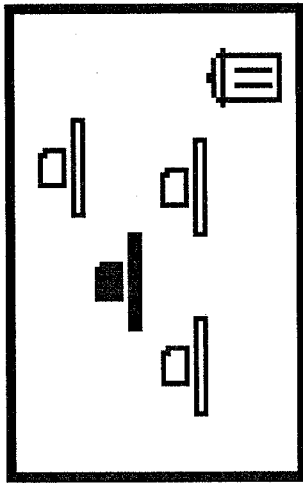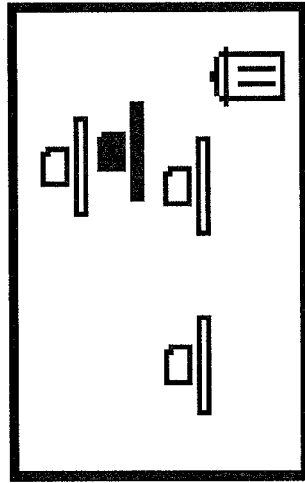
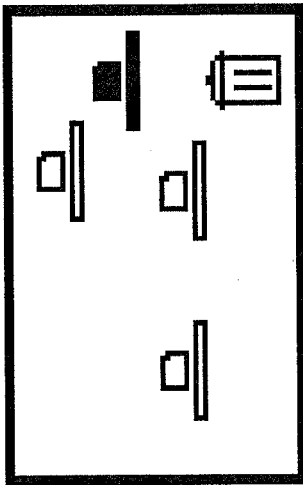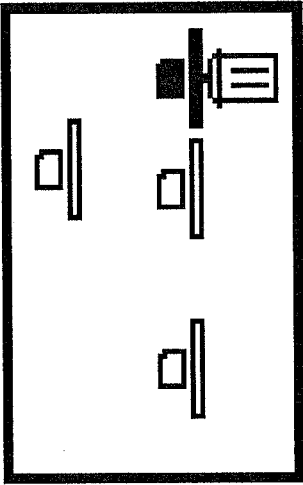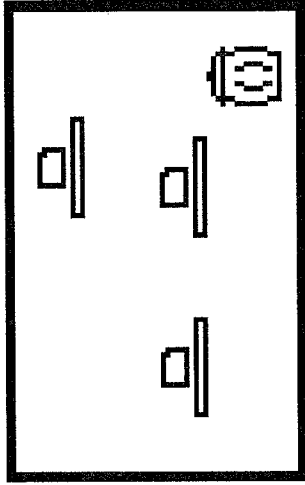**Figure 3.** A BITPICT rule system to move a selected file to the garbage can. (a) Rules for selecting, then moving a file right and down, and into the garbage can. (b) Sample of 8 views from the sequence of 84 steps that take one file from the original arrangement in b.1 to the final configuration in b.8 with the full garbage can.
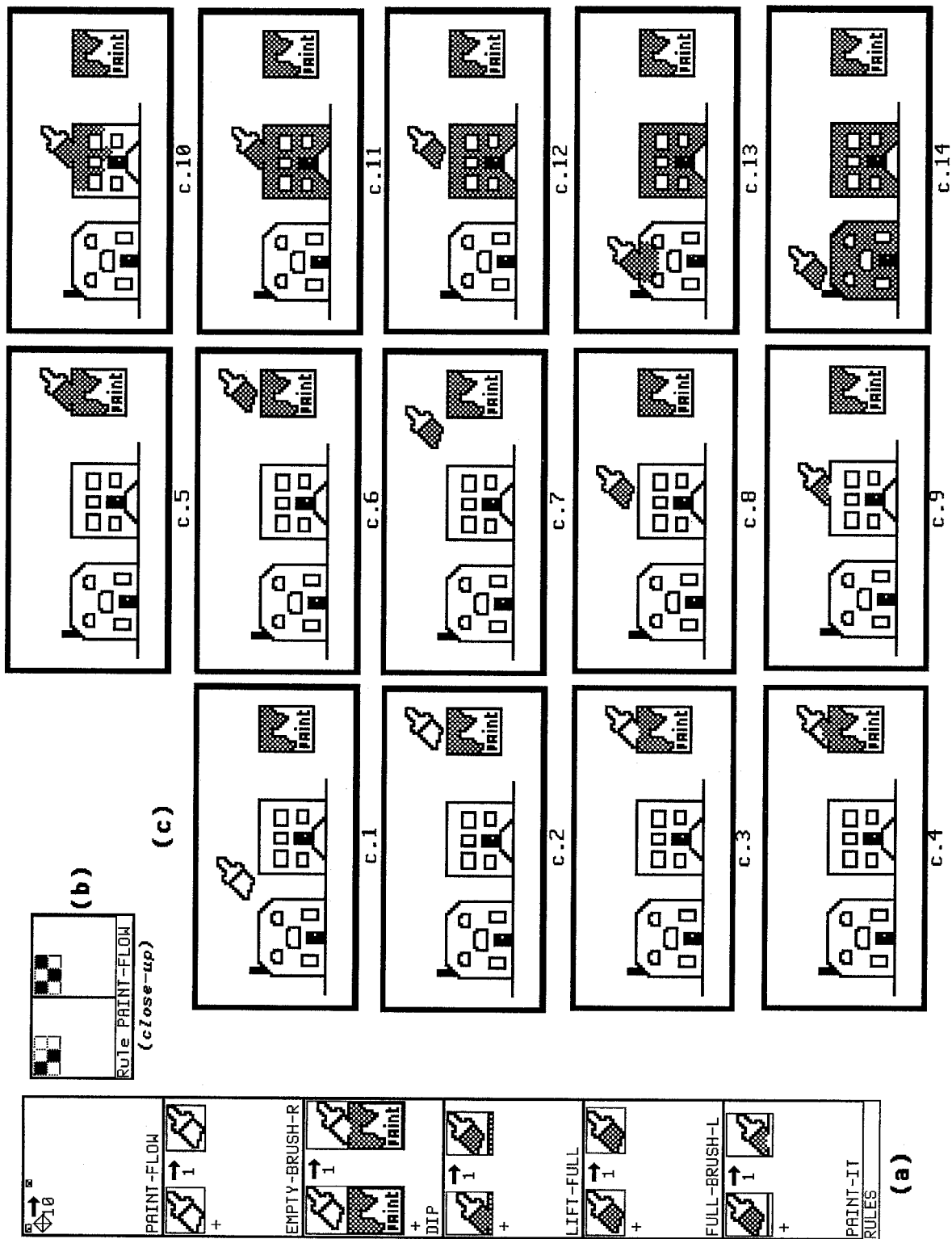
76

Figure 4. A BITPICT rule system to paint houses. (a) The rules. First is a rule for gray paint flowing to adjacent white space (see closeup in (b)). It is given highest priority, so all other rules must wait for this to finish flowing. Other rules are for moving an empty brush right, for dipping the empty brush into the can, for lifting the full brush, for moving the full brush left, and for touching a white. (c) A sampling of the black board states drawn from the sequence of 626 rule applications (most of them paint-flows) that take picture c.1 through the filling of the brush and the painting operations to final painted houses in c.14.

77

interaction. One could argue that visuo-motor interactions may be those whose descriptions are best given in explicitly graphical and motoric rules. The graphical part of that assertion is the focus of the current efforts, but theoretical work on non-sentential deduction systems raises the possibility of motoric and visuo-motoric systems, and perhaps an even better understanding of what "direct manipulation" can really mean.

At the very least, the explicit imaginal emphasis of the current work draws attention to the coherence of what is happening in the behavior of the graphics *per se*: What, in graphical terms, are the the rules which govern how the interface behaves and which users must therefore understand? Regardless of the adequacy of the BITPICT system or any other particular graphical computation model, this question may provide a useful focus.

REFERENCES

1. Lindsay, Robert K., "Images and inference" *Cognition*, **29**, 1988, 229-250.

2. Whiteside, J., Jones, S., Levy, P. S., and Wixon, D., User performance with command, menu, and iconic interfaces, *Human Factors in Computer Systems, Proceedings of CHI'85*, 1985, 185-191.

3. Nielsen, J., Traditional dialogue design applied to modern user interfaces, *Communications of the ACM*, **33**(10), 1990, 109-118.

4. Karat, John Evaluating user interface complexity. *Proceedings of the Human Factors Society - 31st Annual Meeting*, 1987, 566-570.

5. Guastello, S. J., Traut, M., and Korienek, G., Verbal versus pictorial representation of objects in a human-computer interface, *International Journal of Man-Machine Studies*, **31**, 1989, 99-120.

6. Card, S. K., Moran, T. P., and Newell, A., *The psychology of human computer interaction*. Hillsdale, NJ:Erlbaum, 1983.

7. Kieras, D. E. and Polson, P. G., An approach to the formal analysis of user complexity, *International Journal of Man-Machine Studies*, **22**, 1985, 365-394.

8. Tennant, N., The withering away of formal semantics, *Mind and Language*, **1**, 1986, 302-318.

9. Barwise, J. and Etchemendy, J., Visual information and valid reasoning, in *Visualization in Teaching and Learning Mathematics*, S. Cunningham and W. Zimmerman (Eds.), Washington: Mathematical Association of America, in press, 1990.

10. Barwise, J. and Etchemendy, J., Information, infons, and inference, in *Situation Theory and its Applications, I*, R. Cooper, K. Mukai, and J. Perry (Eds.) Chicago:University of Chicago Press, 1990.

11. Gardin, F. and Meltzer, B., Analogical representations of naive physics, *Artificial Intelligence*, **38**, 1989, 139-159.

12. Furnas, G. W., Formal models for imaginal deduction, *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, (July 25-28, 1990, Cambridge, Mass.) Hillsdale, NJ: Lawrence Erlbaum, 662-669.

13. Furnas, G. W., Working paper on theory, methods and examples of imaginal deduction, *Bellcore Technical Memorandum*, 1991.

14. Siromoney, G., Siromoney, R. and Krithivasan, K., Picture languages with array rewriting rules, *Information and Control*, **22**, 1973, 447-470.

15. Rosenfeld, A., Array and web grammars: an overview. In A. Lindenmayer and G. Rozenberg (Eds.), *Automata, Languages, Development*, North-Holland, 1976, 517-529.

16. Post, E. L., Formal reductions of the general combinatorial decision problem, *American Journal of Mathematics*, **65**, 1943,197-268.

17. Newell, A., and Simon, H., *Human Problem Solving*, Englewood Cliffs, NJ: Prentice-Hall, 1972.

18. Waterman, D. A. and Hayes-Roth, F., *Pattern-Directed Inference Systems*, NewYork: Academic Press, 1978.

19. Furnas, G. W., "Graphical Reasoning for Graphical Interfaces" SIGGRAPH VIDEO #56. Shown in the Formal Video track of CHI'90, Seattle Washington, April 1990. Also shown in the INTERACT'90 video program, Cambridge England, August 1990.

20. Schneiderman, B., The future of interactive systems and the emergence of direct manipulation. *Behavior and Information Technology*, **1**, 1982, 237-256.

21. Hutchins, E., Hollan, J., and Norman, D. A., Direct manipulation interfaces, in D.A. Norman and S. Draper (Eds.), *User Centered System Design: New Perspectives on Human-Computer Interaction*, Hillsdale, NJ: Lawrence Erlbaum Associates, 1986.