How Things Appear to Work: Predicting Behaviors from Device Diagrams

N. Hari Narayanan* and **Masaki Suwa** and **Hiroshi Motoda**
Advanced Research Laboratory, Hitachi Ltd.
Hatoyama, Hiki-gun, Saitama Pref. 350-03, Japan.
{narayan,suwa,motoda}@harl.hitachi.co.jp

Abstract

This paper introduces a problem solving task involving common sense reasoning that humans are adept at, but one which has not received much attention within the area of cognitive modeling until recently. This is the task of predicting the operation of simple mechanical devices, in terms of behaviors of their components, from labeled schematic diagrams showing the spatial configuration of components and a given initial condition. We describe this task, present a cognitive process model developed from task and protocol analyses, and illustrate it using the example of a pressure gauge. Then the architecture of a corresponding computer model and a control algorithm embodying the cognitive strategy are proposed.

Introduction

We often make use of diagrams while solving problems or explaining things to ourselves or others. Reading a book like *How Things Work* and understanding the multi-modal descriptions of machines it contains provides a persuasive illustration of how diagrams aid common sense reasoning. The problem solving task of hypothesizing qualitative behaviors of a device from its diagram is another example. The task is to predict the operation of the device by hypothesizing behaviors of its components, given a labeled schematic diagram of the device showing the spatial configuration of its components and an initial condition or behavior. In this task the diagram situates the problem solving process and guides it along the direction of causality as well as cues relevant prior knowledge.

Consider someone with a basic knowledge of mechanical components examining the cross-sectional diagram of a device, such as the pressure gauge shown in fig. 1, and reasoning about its operation. This requires that the person reason about spatial processes occurring inside the device. Information used in this type of reasoning is of two kinds: *visual* and *conceptual*. Visual information is obtained from the diagram, and includes spatial configurations and shapes of the device and its components. Conceptual information comes from the prior domain knowledge of the reasoner, and includes predictive knowledge used for making inferences about the device's operation.

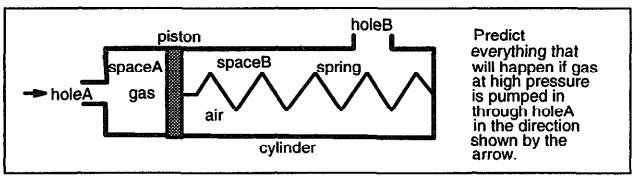

Figure 1: A Behavior Hypothesis Problem

In such reasoning situations diagrams clearly serve as compact representations of spatial information. However, this is only part of the story of the role diagrams play in this task. Diagrams also facilitate the indexing of relevant problem solving knowledge. Furthermore, diagrams support mental visualizations of spatial behaviors of device components during the course of reasoning. It has been shown that such mental visualizations guide human reasoning along the direction of causality (Hegarty 1992).

This paper describes an approach to automating reasoning about devices from diagrams. First, a cognitive process model for this task is proposed and behavior hypothesis steps for the pressure gauge according to this model are enumerated. Then the architecture and control algorithm of a corresponding computer model are described.

Predicting Behaviors from Diagrams

We conducted a set of protocol analysis (Ericsson & Simon 1983) experiments with five subjects solving six behavior hypothesis problems each. Verbal data (concurrent verbal reports) and gestural data were collected during the course of problem solving and studied (Narayanan, Suwa, & Motoda 1994). The main goal of these experiments was to characterize how visual infor-

*Current address: Knowledge Systems Laboratory, Stanford University, 701 Welch Rd., Palo Alto, CA 94304. narayan@ksl.stanford.edu.

mation from the diagram and conceptual information (prior knowledge) interact and influence the direction of reasoning during problem solving. Though the solutions that subjects provided were not always complete and contained inaccuracies, a preliminary analysis of experimental data indicated that the diagram played two crucial roles during problem solving.

- It facilitated the indexing and recall of both factual knowledge about components and inferential knowledge using which the reasoner generated new hypotheses.
- It supported visualizations of hypothesized spatial behaviors of components, which in turn enabled the reasoner to detect effects of these behaviors.

Based on a task analysis (Narayanan, Suwa, & Motoda 1993) – which involved developing detailed and stepwise descriptions of problem solving in this task for some examples and analyzing these descriptions – and an examination of verbal reports and gestures that subjects generated in the aforementioned experiments, we developed a cognitive process model of problem solving in this task (fig. 2). This model explicates the visual reasoning strategy employed in solving behavior hypothesis problems from diagrams. Notice that reasoning proceeds in cycles. At first, short term memory contains only the given initial condition. So reasoning starts with the component and its behavior mentioned in the initial condition. In each cycle new hypotheses are generated in one of three ways: by reasoning about effects of the current non-spatial behavior, by observing the diagram to locate connected/contacting components and reasoning about how these will be affected by the current spatial behavior, or by mentally visualizing spatial behaviors, detecting component interactions that result, and reasoning about effects of these interactions. The new hypotheses are added to the short term memory, and the component and behavior to focus on in the next cycle are selected from the short term memory.

Now let us reconsider the problem in fig. 1 and enumerate steps of reasoning according to this process model. Following each step the alphabetic labels of corresponding parts of the process model from fig. 2 are given in parentheses. Due to limited space, this enumeration does not contain steps corresponding to every part of the process model in each reasoning cycle.

1. Consider the given initial condition (A).
2. Observe from the diagram that holeA opens to spaceA (D).
3. Infer that the pressurized gas will enter spaceA (G,K).
4. Observe from the diagram that spaceA is a closed cavity (A,D).
5. Recall the inferential knowledge that pressurized gas contained in a closed cavity will exert a force in the normal direction on walls of the cavity (G,K).
6. Observe from the diagram that the cylinder and piston form walls surrounding spaceA (A,D).
7. Infer that a force in the normal direction will be exerted on the piston and cylinder (G,K).
8. Recall the inferential knowledge that force can induce motion in a movable component (K).
9. Recall the factual knowledge that the piston is movable in a piston-cylinder assembly (K).
10. Infer that the piston will move (K).
11. Observe the piston in the diagram and conclude that it is free to move left or right (A,D).
12. Infer that the piston will move right (K).
13. Observe from the diagram that the piston is connected to a spring, and is in contact with air in spaceB; consider each in turn (A).
14. Recall the inferential knowledge that if a component is connected to another, and the former starts moving in one direction, it will exert a force on the latter in the same direction (G,K).
15. Infer that when the piston starts moving right, it will exert a rightward force on the spring (K).
16. Recall the inferential knowledge that a force applied on a spring will either compress it or expand it depending on the direction of the force (K).
17. Infer that the spring will compress (K).
18. Consider the air inside spaceB (A).
19. Observe from the diagram that spaceB is an open cavity with holeB (D).
20. Recall the inferential knowledge that if gas inside an open cavity is pushed, it will escape through the cavity's openings (G,K).
21. Infer that air in spaceB will exit through holeB (K).
22. All immediate effects of the hypothesized piston motion have now been considered (A,H).
23. Visualize the piston's rightward motion and the spring's compression (I).
24. Notice that the spring gets compressed more and more as the piston moves (J).
25. Recall the inferential knowledge that as a spring gets compressed or expanded, it will exert an increasing force in the opposite direction (K).
26. Infer that the spring will exert a force on the piston which, at some point, will equal the force exerted by the pressurized gas on the piston (K).
27. Observe from the diagram that this may happen before or after the piston reaches holeB; consider each case (A,D).
28. In the former case, infer that the piston will stop somewhere before holeB (K).
29. Infer that the spring compression will cease (K).

A similar enumeration can be done for the other case, in which inferences about the piston moving back and forth around the region of holeB are generated.

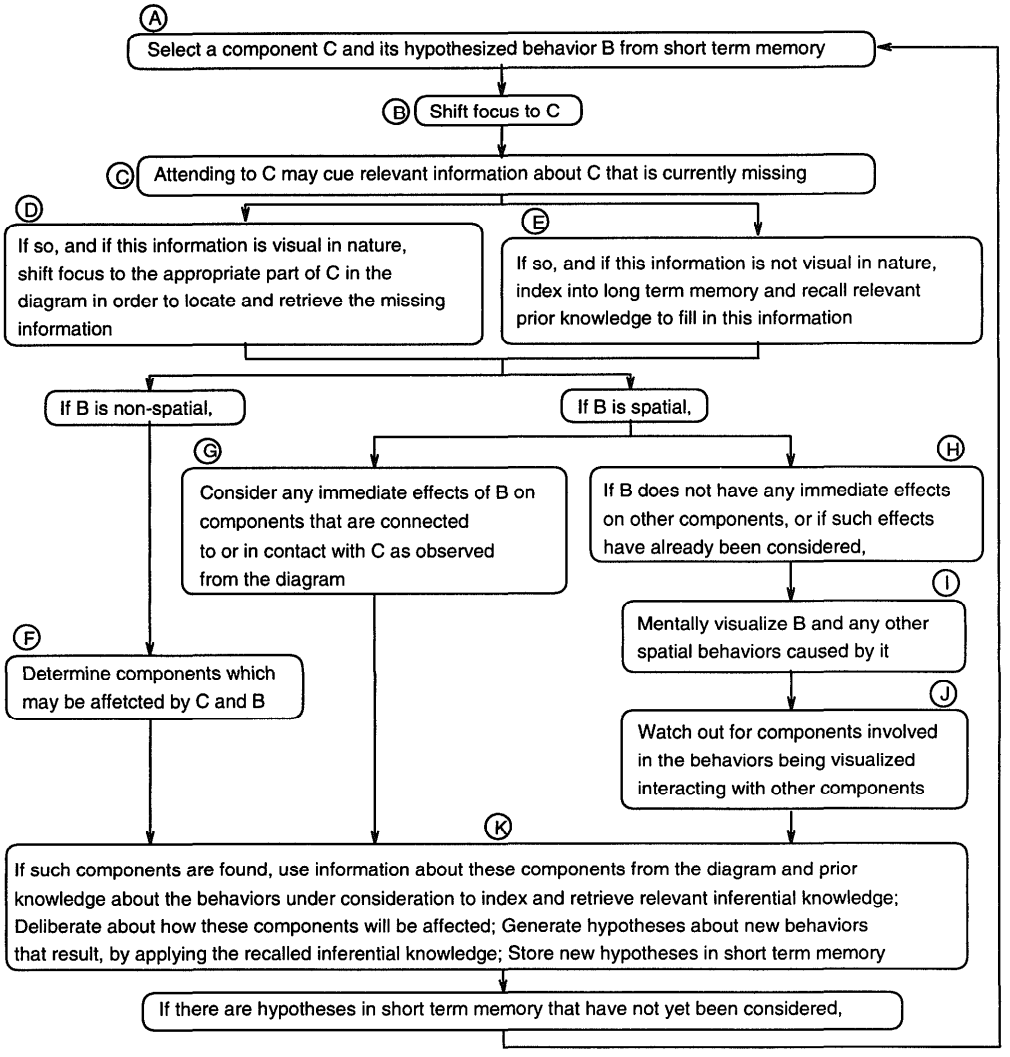

Figure 2: A Cognitive Process Model

An Architecture for Visual Reasoning

In this section we describe an architecture for visual reasoning from diagrams. It has five main elements: a user interface, a knowledge base, a rule base, a working memory, and an inference engine (fig. 3). The user interface allows the user to specify both descriptive and diagrammatic aspects of a problem, and to watch manipulations of the input diagram that the system carries out during the course of reasoning. The knowledge base contains two kinds of representations: descriptive and visual. The rule base contains domain-specific inference rules. The inference engine generates new inferences by accessing and manipulating information from both kinds of representations in the knowledge base in accordance with rules selected from the rule base. The generated inferences are stored in the working memory.

Descriptive representations store domain-specific conceptual information. For the behavior hypothesis task, this includes both general knowledge about component types in the domain and particular knowledge about components of the device in the input problem. We use a frame-based representation that organizes knowledge around component types, but other types of descriptive representations may also be used. Visual representations contain diagrammatic information and have two parts: one contains information about diagram elements that stand for components or parts of components of the device (diagram frames) and the other is an array representation (Glasgow & Papadias 1992) in which the diagram is literally depicted by filling appropriate array elements with symbolic labels of components and substances that "occupy" the corresponding locations. This captures shape, geometry and configuration information. Diagram frames

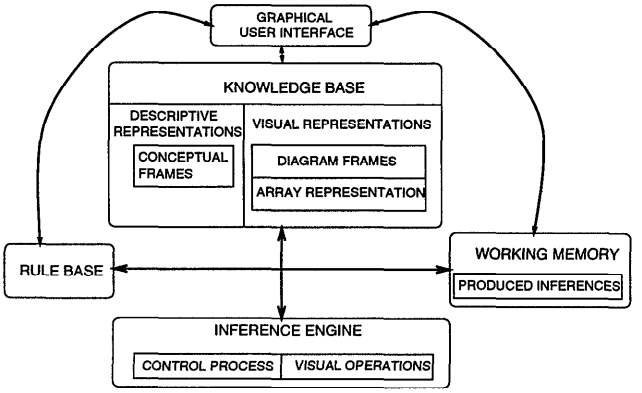

Figure 3: An Architecture for Visual Reasoning

contain information about diagram elements like segments, boundaries and areas.

A behavior hypothesis problem can be provided to the system by specifying, via the user interface, conceptual information about components of a device (component types, labels, etc.), the device diagram, and an initial condition. The user interface program takes this specification and stores information about the device's components in the descriptive part of the knowledge base, uses given information about component types to link this knowledge with general knowledge about various types of components that already exists in the knowledge base, represents the device diagram using both diagram frames and array representation, and stores the initial condition in the working memory in a last-in-first-out queue (LIFO-Q). Thus, the user

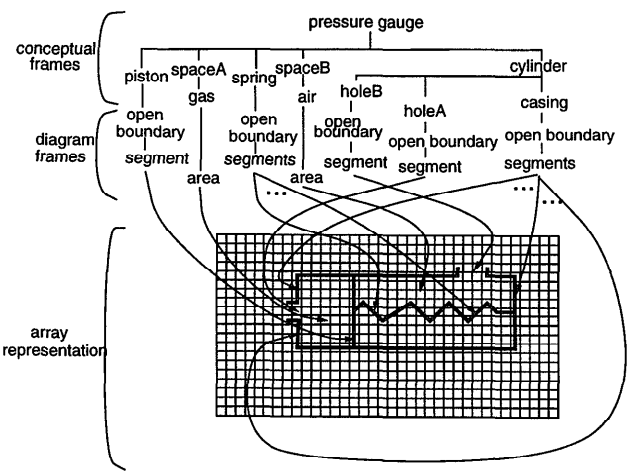

Figure 4: Problem Representation

interface generates an internal problem representation of the form shown in fig. 4. This representation is hierarchical, with conceptual frames linked to diagram frames and diagram frames at the lowest level containing pointers to the array. Symbolic labels stored in array elements provide connections in the reverse direction. This organization allows reasoning procedures to move between conceptual and diagrammatic information easily – an important characteristic of human reasoning with diagrams.

An example of a conceptual frame is [SpaceB - *Type* : cavity; *Contains* : air; *Pressure* : (value); *Boundaries* : (cylinder holeB piston); *Part-of* : pressure gauge; *Parts* : none;]. A diagram frame has slots describing geometric attributes of the corresponding diagram element. For example, the diagram frame representing a line segment will have slots for starting and ending locations, length, and orientation. The array representation may be thought of as a "bitmap" except that each element (bit) contains symbolic labels of parts or substances instead of an intensity value. Conceptual and diagram frames are created at problem specification time by instantiating templates of the various types of frames that already exist in the knowledge base. Indices of array elements corresponding to each diagram frame are then computed and filled with labels of the corresponding part or substance. The user specifies the basic elements constituting the diagram (at present these are line segments and individual locations) and how they are to be grouped together to form open and closed boundaries or define areas.

Each inference rule in the rule base has three parts: an antecedent that refers to descriptive and/or diagrammatic information, a consequent containing new inferences that the system asserts in the working memory if conditions in the antecedent have been verified, and side-effects, which are procedures that manipulate both descriptive and visual representations in the knowledge base and which get activated when the corresponding rule is fired. Fig. 5 shows a sample in-

If a gas enters a cavity which is closed except for the opening through which it is entering, it will fill the cavity.

```
(and
("descriptive" (substance-move ?1<gas> ?2<part> ?3<opening> ?4<cavity>))
("diagrammatic" (closed ?4<cavity> ?3<opening>)))
↓
(assert (fill ?1<gas> ?4<cavity>))
```

Side effects:

Update "contains" and "pressure" slots of the conceptual frame of the cavity bound to ?4<cavity> with values of ?1<gas> and "pressure" slot of ?1<gas> respectively; Update the array by adding the symbolic label of gas bound to ?1<gas> to array elements representing ?4<cavity>.

Figure 5: An Inference Rule

ference rule. The working memory is the computer equivalent of short term memory, except that it is not subject to capacity limitations of human short term memory. The LIFO-Q of inferences is maintained in the working memory. It also contains all new information generated during the course of problem solving.

The reasoning steps carried out by the inference en-

gine fall into the following seven classes. *Diagram Observation*: Access the diagram frames and/or the array representation to find and retrieve spatial information. *Factual Retrieval*: General knowledge retrieval from descriptive representations. *Inference Rule Retrieval*: Indexing and retrieval of rules from the rule base. *Conceptual Inference*: An inference based only on conceptual information from descriptive representations. *Visual Inference*: An inference based only on spatial information from visual representations. *Hybrid Inference*: An inference based on both conceptual information and spatial information. *Visualization*: Simulating a spatial behavior by incrementally modifying the visual representation of the device diagram.

In order to facilitate accessing and manipulating the visual representations, a set of "visual operations" are made available to the inference engine. Visual operations are procedures for accessing and manipulating both types (diagram frames and the array representation) of visual representations. These are of four kinds: basic operations, indexing operations, scanning operations and visualization operations.

Basic Operations. These are operations on individual array elements. Read(x, y) returns labels l of the array element at location (x, y). Write(x, y, l) marks the array element at location (x, y) using labels l. Other basic operations are erase, test, add-label, and remove-label.

Indexing Operations. Indexing operations generate indices or addresses of array elements. At least four such operations are required. Directional indexing: Given an index (x, y) and a direction[1], generate the sequence of indices of cells which fall in that direction from (x, y). Boundary indexing: Given an index (x, y) and a symbol s, generate a sequence of indices of cells, each of which is adjacent to the previous one and contains s as a label. Neighborhood indexing: Given an index (x, y), generate the sequence of indices of its neighboring cells. Fill indexing: Given an index (x, y), generate a sequence of indices of cells such that these gradually cover the area surrounding (x, y) until a boundary is reached.

Scanning Operations. Scanning operations use indexing operations to generate indices of array elements and basic operations to test those elements for various conditions. At least three different kinds of scanning operations are required. Directional Scanning: Given a starting point in the array, a direction, and one or more conditions, test all array elements from the starting point that fall along the given direction for the given conditions. Boundary Following: Given a starting point on a boundary and one or more conditions, follow the boundary from the starting point and test the boundary elements or their neighborhoods for the given conditions. Sweeping: Like directional scanning, except that an area is scanned.

Visualization Operations. Visualization operations transform the represented diagram by manipulating diagram elements that represent individual components. There are both general operations such as move, rotate, copy, and delete, and component-specific ones like compress-spring.

A simplified version of the control algorithm underlying the inference engine's operation is given below.

```
If the LIFO-Q is empty, Then halt;
Else
  Retrieve the latest inference from LIFO-Q;
  If this inference is about a spatial behavior,
  Then
    If all its consequences have been considered,
    Then visualize this behavior and
      other spatial behaviors it causes;
    Else visualize this behavior for
      a small number of steps;
    Detect component interactions by
      scanning the array representation;
    Generate inferences corresponding
      to detected interactions;
    Assert these in the working memory and
      add to LIFO-Q;
  Else (a non-spatial behavior)
    Match and retrieve rules from the rule base;
    Attempt to verify antecedents;
    For each rule thus found to be applicable,
      If its consequent has not already been asserted,
        Assert the consequent in working memory
          and add to LIFO-Q;
        Activate side-effects;
Return to the beginning of the cycle.
```

This algorithm was developed from the model in fig. 2 by replacing mental representations and mental operations by corresponding knowledge representations and operations on those representations. For example, the storage and selection of an inference at the beginning of each cycle from short term memory is implemented using the LIFO-Q data structure in working memory. The inferential knowledge recalled from long term memory during deliberation is represented by inference rules. The computational process corresponding to mental visualization is the simulation of spatial behaviors that visualization operations carry out on the array representation, combined with scanning for component interactions in the cells of the array. The immediate effects of a spatial behavior can also be detected by a similar computational process: simulating the behavior for a small number of steps and scanning for interactions.

This algorithm combines rule-based reasoning with diagram-based reasoning. Each reasoning cycle begins by extracting an element from the LIFO-Q. Reasoning

[1] At present sixteen discrete directions are defined on the array, with each differing from the next by 22.5 degrees; this is an arbitrary choice.

changes to a diagram-based mode when either this inference is a hypothesis about a spatial behavior of a component and its immediate consequences on other components need to be determined, or it is a hypothesis about a spatial behavior whose immediate consequences have already been determined. In the former case a simulation of the behavior for a few steps is carried out. This will detect any immediate consequences of that spatial behavior on any nearby components. These detected effects are stored in the LIFO-Q. In the latter case, we have a behavior whose immediate consequences (which may be spatial behaviors of affected components nearby) have already been determined. So the next step will be to simulate this behavior and its spatial consequences. This simulation will not be terminated after a few steps. Instead, it will proceed until some interaction between components is detected (or it becomes clear that no interactions will occur within the diagram boundaries). Once an interaction is detected, its effects are determined and stored in the LIFO-Q.

For illustrative purposes, fig. 6 shows the trace of a simple example requiring about twenty rules (the same as in fig. 1, but without the complication introduced by the placement of holeB). Reasoning be-

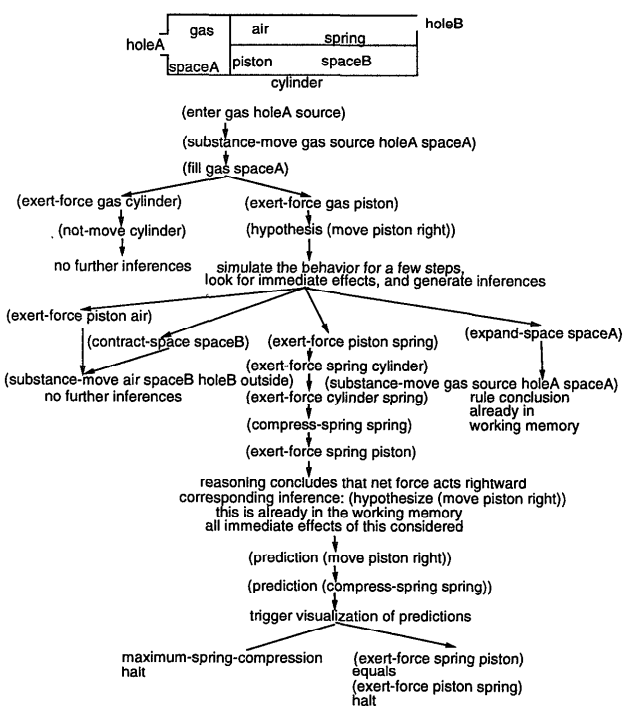

Figure 6: An Example and Corresponding Inferences

gins with the given initial condition, *(enter gas holeA source)*. The next inference is derived from the application of an inference rule that checks the diagram to ensure that holeA is not blocked. The inference *(fill gas spaceA)* requires both knowledge about gases at high pressure filling spaces and a diagram observation to ensure that spaceA is closed except for holeA. At this point rule-based reasoning generates two inferences, one of which leads to a dead end. The other leads to the hypothesis *(move piston right)*. Since this is a hypothesis concerning a spatial behavior, its immediate effects are investigated. This results in four new inferences. Two of them lead to the same inference *(substance-move air spaceB holeB outside)* for which no further rules are found to be applicable. The fourth inference generates another which already exists in the working memory. So the only path further followed by the inference engine is that leading from *(exert-force piston spring)*. This eventually leads to the inference *(exert-force spring piston)*. At this point two forces on the same object have been hypothesized. Knowledge about springs leads the system to conclude that the initial force exerted by the spring is small. The resulting hypothesis, that of a rightward move of piston, is already in the working memory. At this point all immediate effects of this hypothesized motion have been considered and therefore two spatial behaviors—the motion of piston and the compression of spring—can be predicted. This triggers a simulation of these. This simulation will detect the fact that as the piston moves, the spring will get progressively more compressed. Based on general knowledge about behaviors of springs and using rules for reasoning about inequalities, the system will then discover two possibilities: either the spring will get maximally compressed or the piston will reach a point at which the two forces are in equilibrium. At this point there are no more entries in the LIFO-Q and the inference process will come to a stop. These inferences amount to a series of hypotheses about the given device's behaviors, generated along the direction of causality.

Conclusion

This paper presented a study of visual reasoning from diagrams in device behavior hypothesis tasks. Analyzing this task and examining hypotheses generated by human subjects allowed us to formulate a cognitive process model of problem solving in this task. Using this model as a basis we showed how an example problem could be solved. Then the architecture of a corresponding visual reasoning system was described.

Limitations of the proposed model and aspects that have not been addressed yet provide avenues for current and future research. One such issue is the notion of relevance. In the present computer model, slots of conceptual frames associated with arguments of the current inference (in a cycle) constitute the "relevant" facts, and rules retrieved from the rule base by matching with the current inference constitute the "relevant" inferential knowledge. However, a more sophisticated and operational notion of relevance to filter facts and rules may be required to efficiently deal with large amounts of conceptual information and a large rule base. Support for visualization and support for in-

dexing and recall of relevant factual and inferential knowledge were mentioned earlier as the two major roles that diagrams seem to play in problem solving. The current control algorithm, with its forward chaining strategy, implements only the former. In order to implement the latter (for instance, noticing something unexpected in the diagram may cue factual knowledge or inferential knowledge that leads to a new prediction), additional strategies (such as an opportunistic one) need to be incorporated in the control algorithm. Furthermore, visual indexing schemes for conceptual knowledge and inference rules need to be developed. The computer model is designed to support 2-D visualizations of spatial behaviors such as translations, rotations, and deformations. This can in principle be extended to 3-D by using a three dimensional array representation. But the visualizations do not capture temporal aspects of spatial behaviors (e.g., velocity). Granularity of visual representations is another aspect requiring further research. It has not been critical for dealing with relatively abstract schematic diagrams of the sort discussed in this paper. Also, larger arrays can be used for finer resolution. Nevertheless, issues such as selecting an appropriate grain size to ensure that component interactions are not missed during visualizations and information loss that occurs when continuous shapes are rendered onto discrete arrays need to be addressed.

Increasing attention is currently being paid to the coupling of perception and reasoning. This coupling is bidirectional. On one hand, high level knowledge, goals and reasoning can significantly influence aspects of perception such as directing the focus of attention and disambiguating image interpretations (Brand, Birnbaum, & Cooper 1993). On the other hand, visually perceived spatial properties and the mental manipulation of visual representations such as diagrams can considerably aid reasoning and problem solving. Studying this latter phenomenon in various problem solving tasks is the focus of our research. Despite previous pioneering work (Sloman 1971; Funt 1980; Forbus, Nielsen, & Faltings 1987), only recently has the cognitive capability for common sense reasoning using diagrams and imagery begun to receive renewed attention in artificial intelligence (Narayanan 1992). The computer modeling of visual reasoning has been shown to be of benefit in a variety of domains and applications, e.g., automating expert reasoning using phase diagrams (Yip 1991), geometry theorem proving (McDougal & Hammond 1993), and analysis of load-bearing structures (Tessler, Iwasaki & Law 1993). Current research in this area is still exploratory in nature. However, its maturity promises many applications in intelligent multimodal interfaces, knowledge-based graphics, and instructional systems for imparting visualization skills in problem solving.

Acknowledgments. We thank T. Nishida for providing software for part of the user interface, Y. Iwasaki for many useful discussions, and Knowledge Systems Laboratory, Stanford University, for facilitating the preparation of this paper.

References

Brand, M., Birnbaum, L., and Cooper, P. (1993). Sensible scenes: visual understanding of complex structures through causal analysis. *Proc. 11th NCAI*, 588-593.

Ericsson, K. A. and Simon, H. A. (1983). *Protocol Analysis: Verbal Reports as Data*, MIT Press, Cambridge, MA.

Forbus, K. D., Nielsen, P., and Faltings, B. (1987). Qualitative kinematics: a framework. *Proc. 10th IJCAI*, 430-436.

Funt, B. V. (1980). Problem-solving with diagrammatic representations. *Artificial Intelligence*, 13: 201-230.

Glasgow, J. and Papadias, D. (1992). Computational Imagery. *Cognitive Science*, 16(3): 355-394.

Hegarty, M. (1992). Mental animation: inferring motion from static displays of mechanical systems. *JEP: Learning, Memory, and Cognition*, 18(5): 1084-1102.

McDougal, T. F. and Hammond, K. J. (1993). Representing and using procedural knowledge to build geometry proofs. *Proc. 11th NCAI.*

Narayanan, N. H., (Ed.). (1992). *Working Notes of the AAAI Spring Symposium on Reasoning with Diagrammatic Representations.*

Narayanan, N. H., Suwa, M., and Motoda, H. (1993). Behavior hypothesis from schematic diagrams: a hybrid approach. *Proc. IJCAI-93 Hybrid Representation and Reasoning Workshop*, 50-61.

Narayanan, N. H., Suwa, M., and Motoda, H. (1994). A study of diagrammatic reasoning from verbal and gestural data. *Proc. 16th Ann. Conf. of the Cognitive Science Society*, to appear.

Sloman, A. (1971). Interactions between philosophy and AI: the role of intuition and non-logical reasoning in intelligence. *Artificial Intelligence*, (2): 209-225.

Tessler, S., Iwasaki, Y., and Law, K. (1993). Qualitative structural analysis using diagrammatic reasoning. *Proc. Qualitative Reasoning Workshop.*

Yip, K. M. (1991). Understanding complex dynamics by visual and symbolic reasoning. *Artificial Intelligence*, 51: 179-221.