QUALITATIVE BEHAVIOR HYPOTHESIS 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 "qualitative" problem solving task that humans are adept at, but one which has not received much attention within the qualitative physics community. This is the task of predicting the operation of a simple mechanical device, in terms of spatial behaviors of its components, from a labeled schematic diagram of the device showing the spatial configuration of its components and a given initial condition. Using the example of a pressure gauge we define this task, present a cognitive strategy for solving such problems, and describe the architecture of a corresponding computer model.

Introduction

We often use spatial information implicit in diagrams to make inferences. The task of *qualitative behavior hypothesis from device diagrams* is a case in point: given the labeled schematic diagram of a device that shows the spatial configuration of its components and an initial condition or behavior, predict the operation of the device by hypothesizing the behaviors of its components.

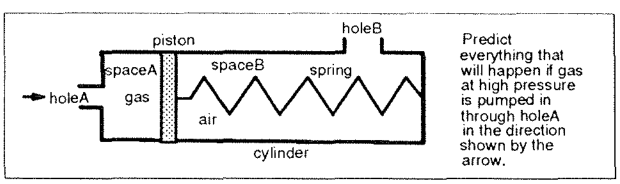

Figure 1: A Behavior Hypothesis Problem: Pressure Gauge

Consider someone 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 he or she 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 domain knowledge of the reasoner, and includes predictive knowledge used for making hypotheses about the device's operation.

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 as perceived from the diagram (Hegarty 1992).

This cognitive capability for "qualitative" visual reasoning from diagrams has not hitherto received much attention in the qualitative physics literature[1]. The automation of visual reasoning can be of benefit in a variety of domains and applications: in automating expert reasoning using phase diagrams (Yip 1991), in developing instructional or demonstration systems whose operation is easily explainable and understandable (Tessler, Iwasaki & Law 1993), and in developing systems whose reasoning spans multiple ontologies or models (Fishwick et. al. 1994, Kiriyama & Tomiyama 1993), to cite a few examples.

This paper describes an approach to automating visual reasoning about devices from diagrams. The reasoning task is defined first. Then hypotheses about the pressure gauge in fig. 1 that human subjects generated in an experimental study are presented. Following this, we develop a cognitive process model for this task and

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

[1]With the exception of Funt's early work (Funt 1980) and the more recent REDRAW system (Tessler, Iwasaki & Law 1993). The work of Forbus and colleagues (Forbus, Nielsen & Faltings 1987) on using a metric diagram for spatial reasoning addressed a different capability than the one being considered here.

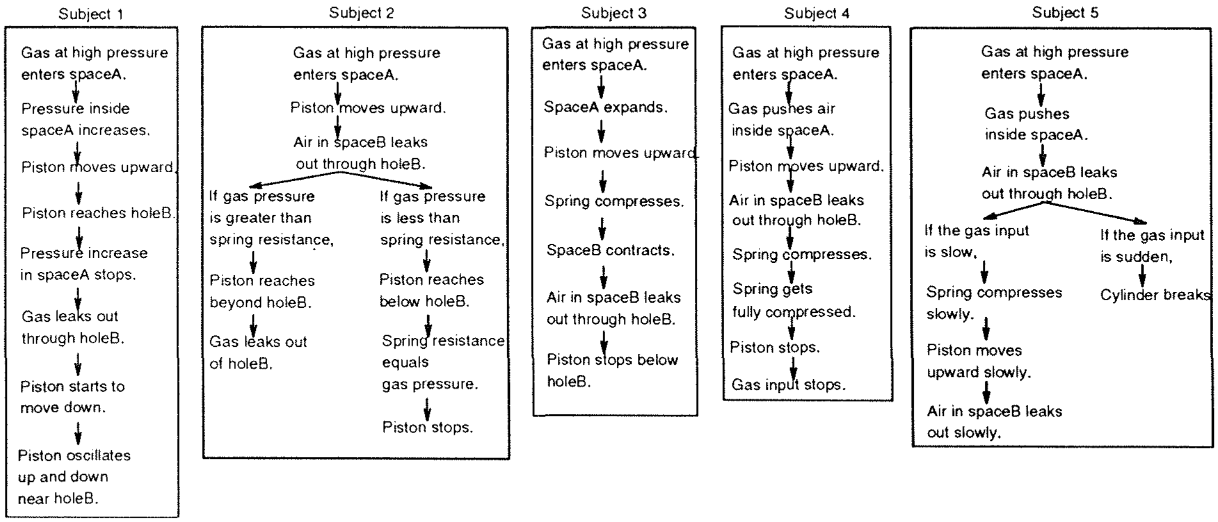

Figure 2: Subjects' Hypotheses about the Pressure Gauge

enumerate behavior hypothesis steps for the pressure gauge according to this model. Finally, the architecture and control algorithm of a computer system designed for this task are described.

Hypothesizing Behaviors from Device Diagrams

We conducted a set of protocol analysis (Ericsson & Simon 1993) experiments with five subjects solving six qualitative behavior hypothesis problems each. Verbal data (concurrent verbal reports) and gestural data were collected during the course of problem solving and analyzed. Details of these experiments can be found in (Narayanan, Suwa & Motoda 1994). What is of relevance here are the behavior hypotheses generated by the five subjects for the problem shown in fig. 1.

Fig. 2 shows these hypotheses, with the arcs indicating the order in which the hypotheses were generated.

The main goal of these experiments was to characterize how visual information from the diagram and conceptual information (prior knowledge) interact and influence the direction of reasoning during problem solving. Though the solutions that the subjects provided were not always complete and contained inaccuracies, the analyses we carried out - both of the task and the data collected - indicated that the diagram played two important roles during problem solving.

- It facilitated the indexing and recall of both factual knowledge regarding 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 task and data analyses (Narayanan, Suwa & Motoda 1993), we developed a cognitive process model of problem solving in this task. It is shown in fig. 3. It explicates the visual reasoning strategy employed in solving qualitative behavior hypothesis problems. Notice that reasoning proceeds in cycles. At first, short term memory contains only the given initial condition. So reasoning starts with a component and its behavior mentioned in the initial condition. In later cycles, a component and its behavior to focus on are selected from among the hypotheses in short term memory. The diagram facilitates the indexing and recall of relevant knowledge in two different ways: (i) attending to a component may cue some relevant factual information about it which is either recalled from long term memory or retrieved from the diagram, and (ii) configurational and shape information about components from the diagram together with prior knowledge about components and behaviors allow the indexing and recall of inferential knowledge. New hypotheses are generated in three ways: (i) by deliberating about effects of non-spatial behaviors, (ii) by observing the diagram to locate connected/contacting components and deliberating about how these will be affected by spatial behaviors, or (iii) by mentally visualizing spatial behaviors, detecting interactions among components that result, and deliberating about effects of these interactions. In each of these cases, the application of the recalled inferential knowledge creates new hypotheses in short term memory.

Now let us reconsider the problem in fig. 1 and enumerate steps of reasoning according to this process model to generate one solution to the problem.

1. Consider the given initial condition.
2. Observe from the diagram that holeA opens to

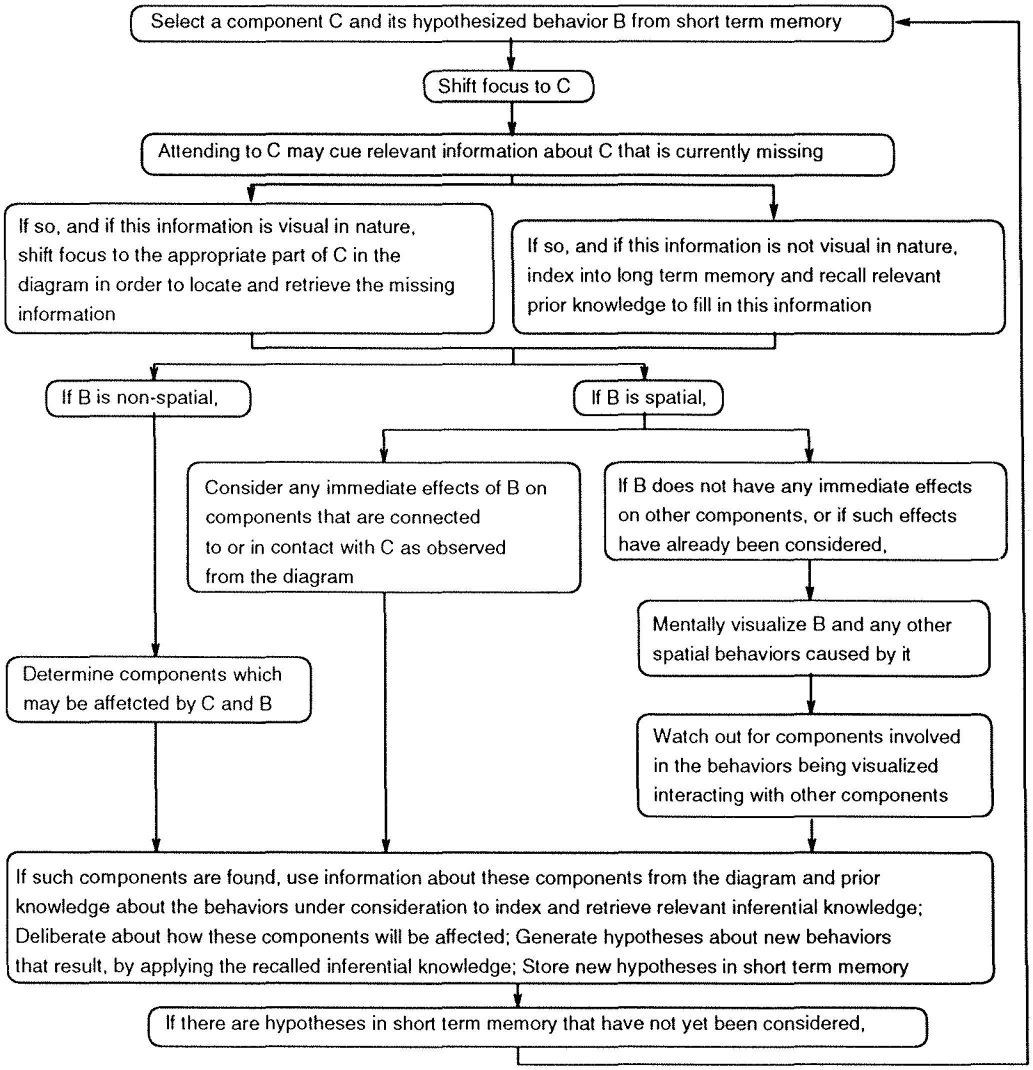

Figure 3: A Cognitive Process Model of Behavior Hypothesis from Device Diagrams

spaceA.

3. Infer that the pressurized gas will enter spaceA.
4. Observe from the diagram that spaceA is a closed cavity.
5. Recall the inferential knowledge that if a pressurized gas is contained in a cavity, it will exert a force in the normal direction on walls of the cavity.
6. Observe from the diagram that the cylinder and piston form walls surrounding spaceA.
7. Infer that a force in the normal direction will be exerted on the piston and cylinder by the pressurized gas.
8. Recall the inferential knowledge that force can induce motion in a movable component.
9. Recall the factual knowledge that the piston is movable in a piston-cylinder assembly.
10. Observe the piston in the diagram and infer that it is free to move up or down.
11. Infer that the piston will move upward.
12. Observe from the diagram that the piston is connected to a spring and is in contact with air in spaceB; consider each in turn.
13. 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.
14. Infer that when the piston starts moving upward, it will exert an upward force on the spring.
15. Observe from the diagram that the other end of the

spring is connected to the cylinder.

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.
17. Infer that the spring will compress.
18. Consider the air inside spaceB.
19. Observe from the diagram that spaceB is an open cavity with holeB.
20. Recall the inferential knowledge that if gas inside an open cavity is pushed, it will escape through the cavity's openings.
21. Infer that air in spaceB will exit through holeB.
22. Now that all immediate effects of the hypothesized piston motion have been considered, visualize its upward motion and the spring's compression.
23. Recall the inferential knowledge that as a spring gets compressed or expanded, it will exert an increasing force in the opposite direction.
24. 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.
25. Observe from the diagram that this may happen before or after the piston reaches holeB; consider each case.
26. In the former case, infer that the piston will stop somewhere below holeB.
27. Infer that the spring compression will cease.

A similar enumeration can be done for the other case, generating the following behavior hypotheses: the piston will reach holeB and allow the pressurized gas to escape; this will decrease the force the gas is exerting on the piston, making it move downwards until a new equilibrium between the gas and spring forces is achieved; this will prevent the gas from escaping, increase its pressure, and the piston will start moving upward; this cycle will then repeat.

Architecture of a Visual Reasoning System

In this section we describe an architecture for a visual reasoning system designed to solve qualitative behavior hypothesis problems from diagrammatic representations by emulating the cognitive processes outlined previously. It has five main elements: a graphical user interface, a knowledge base, a rule base, a working memory, and an inference engine (see fig. 4). The knowledge base contains two kinds of representations: one which stores descriptive knowledge about the components of a physical device using knowledge structures such as frames that organize knowledge around each component type, and another that stores knowledge

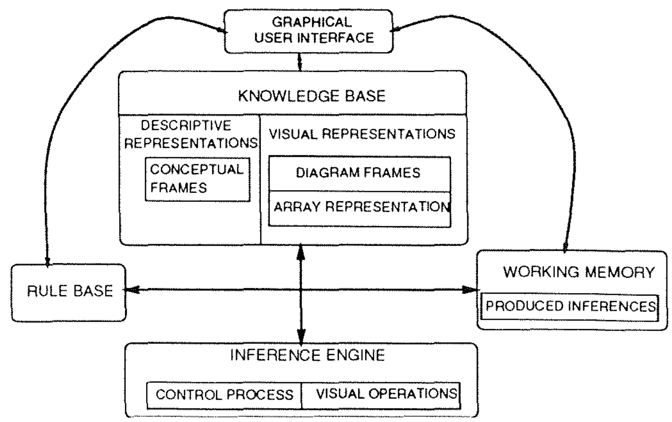

Figure 4: An Architecture for Visual Reasoning

about the shape and geometry of components and devices in a spatially distributed fashion. When a problem to be solved is given by the user, the user interface stores descriptive and spatial parts of the problem specification in the descriptive and visual representations in the knowledge base. Then inference is initiated by the inference engine. It 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 rule base contains inference rules with an if-part and a then-part. The if-part of a rule describes conditions regarding properties of components which may be verified by accessing the descriptive representations, and conditions regarding the shape, geometry and configuration of components which may be verified by accessing and manipulating information stored in the visual representations. The then-part of a rule contains new inferences that may be asserted in the working memory if conditions in the if-part are satisfied. The generated inferences/hypotheses are stored in the working memory.

The descriptive representations in the knowledge base contain conceptual information and the visual representations contain spatial information. Conceptual information includes both general knowledge about types of components in the domain and particular knowledge about the components of the device in the input problem. This information is stored in knowledge structures called "conceptual frames" that organize such information around component types. The spatial information consists solely of the shape, geometry, and spatial configuration of the components of the device in the input problem, information that a diagram of the device typically captures. This information is represented in two ways: one by "diagram frames" which are data structures similar to frames or records storing spatial information, and the second by means of filling in appropriate labels in appropriate elements of a two-dimensional array. This array may be seen as directly representing space, spatial configurations of components, their shape and geometry. Fig. 5

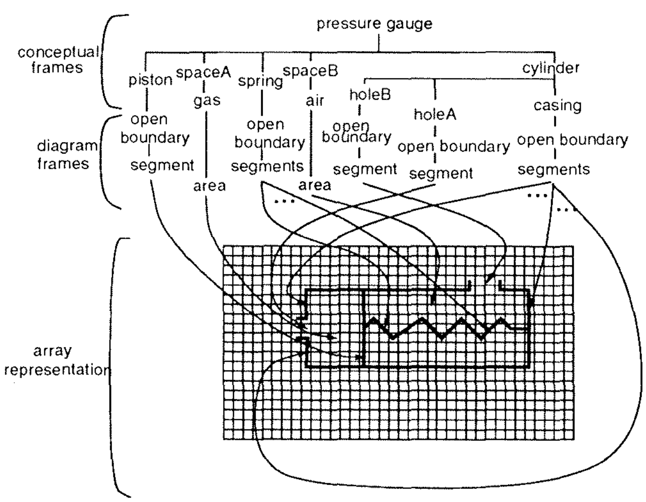

Figure 5: Problem Representation

shows how a device is represented using descriptive and visual representations. It also shows how conceptual frames, diagram frames, and the array representation are related.

A behavior hypothesis problem can be provided to the system by specifying, via the graphical user interface, the components of the device to be reasoned about, any relevant conceptual knowledge about these components, the device diagram, and an initial condition involving behaviors of the device's components. The user interface program takes this specification, and

1. stores conceptual information about the device's components in descriptive representations in the knowledge base;
2. 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;
3. takes the specification of the device diagram and converts it into a form suitable for representing as visual representations in the knowledge base;
4. represents this above information using both diagram frames and array representation; and
5. takes the initial conditions and stores these in the working memory in a last-in-first-out queue (henceforth referred to as LIFO-Q).

Thus, from the input specifications the user interface program generates an internal representation of the form shown in fig. 5.

Each inference rule in the rule base has three parts: an antecedent or if-part containing conditions that refer to descriptive information and/or visual information, a consequent or a then-part containing new inferences that the system hypothesizes to hold if the conditions in the if-part 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. 6 shows a sample inference rule. The rules contained in the rule base can be classified into the following four categories: rules whose if- and then-parts refer only to conceptual frames; rules whose if- and then- parts refer only to diagram frames and the array representation; rules whose if- and then- parts refer to conceptual frames, diagram frames, and the array representation; and rules for carrying out inferences about inequalities and numbers.

The working memory is the computational equivalent of short term memory, except that it is not subject to capacity limitations of human short term memory. The LIFO-Q of inferences/hypotheses 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 engine fall into the following seven classes:

1. *Diagram Observation (DO)*: Access the diagram frames and/or the array representation to find and retrieve spatial information;
2. *Factual Retrieval (R)*: Retrieval of general knowledge from descriptive representations.
3. *Inference Rule Retrieval (IR)*: Indexing and retrieval of relevant rules from the rule base.
4. *Conceptual Inference (C)*: Making an inference based only on conceptual information from descriptive representations in the knowledge base.
5. *Visual Inference (VI)*: Making an inference based only on spatial information from the visual representations in the knowledge base.
6. *Hybrid Inference (HI)*: Making an inference based on both conceptual information and spatial information.
7. *Visualization (V)*: The operation of simulating a spatial behavior by incrementally modifying the visual representation of the device diagram.

It should be evident that these operations use three different kinds of information: conceptual information from the descriptive representations, spatial information from the visual representations, and associative information in the form of inference rules from the rule base. Similarly, these operations produce inferences that may be classified as conceptual, visual or hybrid.

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:

If a gas enters a space that is enclosed except for the opening through which the gas is entering, then the gas will fill inside that space.

(and ("descriptive" (enter ?var-1[gas] ?var-2[opening] ?var-3[space]))
("visual" (equal (openings? ?var-3[space]) (?var-2[opening]))))

↓

(assert (fill ?var-1[gas] ?var-3[space]))

Side effects: (1) Add the value of the variable ?var-1[gas] to the "contains" slot of the conceptual frame of ?var-3[space] in the descriptive representation; (2) Add the label of ?var-1[gas] to the array cells representing ?var-3[space] in the visual representation;

Figure 6: An Inference Rule with Side Effects

basic operations, indexing operations, scanning operations and visualization operations.

Basic Operations:
Read(x,y) returns labels l of the array element with index (x, y); *Write(x,y,l)* marks the array element with index (x, y) using labels l; Two additional operations, *erase(x,y)* and *test(x,y)*, can be defined in terms of the previous two as: erase(x,y) = write(x,y,ϕ); test(x,y) = false if read(x,y) returns ϕ, true otherwise.

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[2], 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 in its label.
Neighborhood indexing: given an index (x, y), generate the sequence of indices of its neighborhood cells. The exact definition of neighborhood may vary.
Fill indexing: given an index (x, y), generate a sequence of indices of cells such that these gradually cover the area surrounding (x, y).
Combining indexing routines with basic operations creates procedures that can be used to build the scanning and visualization operations described below.

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 for the given conditions.
Sweeping: Given a line in the array, a direction, and one or more conditions, test all array elements that would be covered if the line were to be moved in the given direction, for the given conditions.

Visualization Operations:
Visualization operations manipulate components in the array representation. At least the following four are required: *Move(component, direction)*, *Rotate(component, direction)*, *Delete(component)* and *Copy(component)*.

Figs. 7 and 8 together show the control process underlying the inference engine's operation. This process was developed from the model in fig. 3 by replacing mental representations and mental operations by corresponding knowledge representations and operations on those representations. For example, the storage and selection at the beginning of each cycle of a hypothesis (involving a component and its behavior) from short term memory is implemented using the LIFO-Q data structure in the working memory. The inferential knowledge that is indexed and recalled from long term memory during deliberation is represented as productions with antecedent conditions and consequents. Similarly, the computational process that corresponds to mental visualization is a simulation of spatial behaviors by applying visual operations on the array-based visual representation of the diagram and scanning for component interactions in the cells of the array. The immediate effects on connected/contacting

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

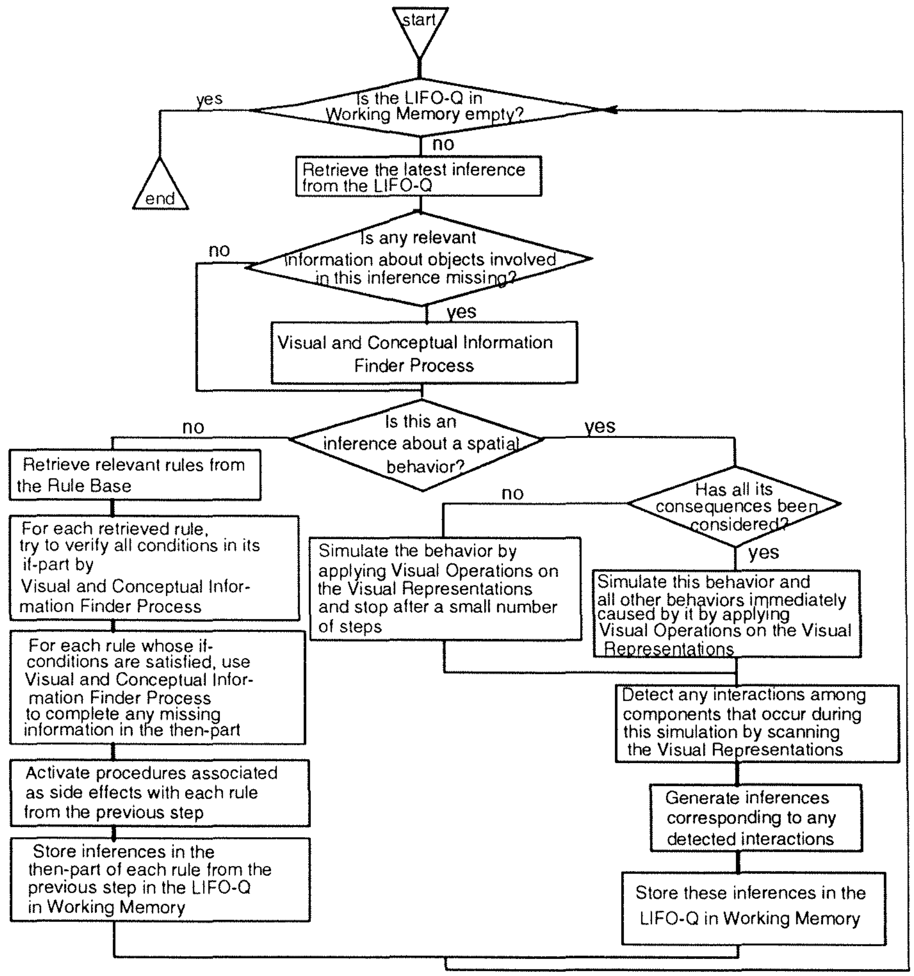

Figure 7: Control Algorithm

components 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 inference method is a mixture of rule-based reasoning and diagram-based reasoning. The left pathway in the flowchart in Fig. 7 describes rule-based reasoning and the right pathway describes diagram-based reasoning. At the start, reasoning begins in the forward reasoning mode, acting on the first element in the LIFO-Q. Reasoning changes to a diagram-based mode when either (i) the current inference is a hypothesis regarding the spatial behavior of a component and its immediate consequences on other components need to be determined, or (ii) it is a hypothesis regarding a spatial behavior whose immediate consequences have already been determined. In the case of (i) 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, and forward reasoning will resume at the beginning of the next cycle. In the case of (ii), we have a behavior whose immediate consequences (which might be other behaviors by affected components nearby) have already been determined. So the next step will be to simulate this behavior and its consequences using the visual representations. In this case, the simulation will not be terminated after a few steps. Instead, it will proceed until a spatial interaction between components is detected. Once an interaction is detected, its effects are determined, and stored in the LIFO-Q. Forward reasoning will resume in the next cycle. This is an overview of the control process.

Conclusion

This paper presented a study of visual reasoning from diagrams in qualitative behavior hypothesis tasks. We began with a definition of the task, following which results from experimental studies were discussed. Anal-

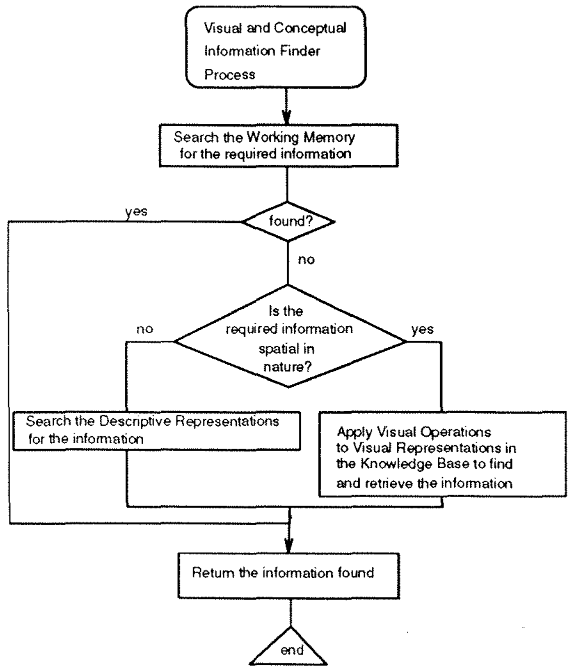

Figure 8: Control Algorithm - Contd.

yses of this task and a close examination of 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 we described a control process and the computational architecture of a visual reasoning system designed to solve behavior hypothesis problems using diagrammatic representations. Elements of this architecture were explained. Implementation of a prototype of this system is currently in progress.

Acknowledgments We thank Dr. Toyoaki Nishida for providing software for part of the user interface module of the architecture.

References

Forbus, K. D., Nielsen, P., and Faltings, B. (1987). Qualitative kinematics: a framework. *Proc. 10th International Joint Conference on Artificial Intelligence*, 430-436.

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

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

Ericsson, K. A. and Simon, H. A. (1993). *Protocol Analysis: Verbal Reports as Data (Revised Edition)*, MIT Press, Cambridge, MA.

Fishwick, P., Narayanan, N. H., Sticklen, J. and Bonarini A. (1994). A multimodel approach to reasoning and simulation. *IEEE Transactions on Systems, Man and Cybernetics*, in press.

Kiriyama, T. and Tomiyama, T. (1993). Reasoning about models across multiple ontologies. *Proc. Int'l Workshop on Qualitative Reasoning about Physical Systems*, 124-131.

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

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

Tessler, S., Iwasaki, Y., and Law, K. (1993). Qualitative structural analysis using diagrammatic reasoning. *Proc. Int'l Workshop on Qualitative Reasoning about Physical Systems.*

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