Qualitative Structural Analysis Using Diagrammatic Reasoning

Shirley Tessler*, Yumi Iwasaki* and Kincho Law**

***Knowledge Systems Laboratory**
Stanford University
701 Welch Road, Bldg C
Stanford, California 94305
iwasaki@ksl.stanford.edu

****Civil Engineering Department**
Stanford University
Terman Engineering Center
Stanford, California 94305
law@cive.stanford.edu

Abstract

Diagrammatic reasoning is a type of reasoning in which the primary means of inference is the direct manipulation and inspection of a diagram. Diagrammatic reasoning is prevalent in human problem solving behavior, especially for problems involving spatial relationships among physical objects. Our research examines the relationship between diagrammatic reasoning and symbolic reasoning in a computational framework. We have built a system, called REDRAW, that emulates the human capability for reasoning with pictures in civil engineering. The class of structural analysis problems chosen provides a realistic domain whose solution process requires domain-specific knowledge as well as pictorial reasoning skills. We hypothesize that diagrammatic representations, such as those used by structural engineers, provide an environment where inferences about the physical results of proposed structural configurations can take place in a more intuitive manner than that possible through purely symbolic representations.

1. Introduction

Humans often use diagrams to facilitate problem solving. In many types of problems, including but not limited to problems involving behaviors of physical objects, drawing a diagram is a crucial step in the solution process. Drawing can reveal important information that may not be explicit in a written description, and can help one gain insights into the nature of the problem. Though such use of diagrams is an integral part of human problem solving behavior, it has not received nearly as much attention in AI as symbolic reasoning has.

One important advantage of diagrammatic representation in some types of problems is that it makes explicit the spatial relationships that might require extensive search and numerous inference steps to determine using a symbolic representation. Larkin and Simon have shown that, even when the information contents of symbolic and diagrammatic representations are equivalent, a diagrammatic representation can offer computational advantage in problems where spatial relationships play a prominent role [Larkin & Simon 1987]. Since humans reason with so much apparent ease in some problems, a program that could reason directly with a diagrammatic representation would be more understandable to the user than a program that reasons exclusively with a purely symbolic representation of the same information. In addition, a diagrammatic reasoning program should offer insight into the relationship between diagrammatic reasoning and symbolic reasoning. Such a program may also be useful in imparting visualization skills to students of disciplines where such a facility is crucial, such as in civil or mechanical engineering and design.

In this paper, we present our work aimed towards understanding the role of diagrammatic reasoning in problem solving. The problem we chose for studying diagrammatic reasoning is that of determining the deflection shape of a building frame structure under load. We have constructed a computer program called REDRAW (Reasoning with Drawings) that solves this problem qualitatively using a diagram in a way similar to human engineers.

1.1 Roles of diagrams in Problem Solving

Some research has been done on the roles that diagrammatic reasoning play in human problem solving. Novak and Bulko, [Novak & Bulko 1992], for example, have asserted that a diagram and its annotations serve as a short-term memory device in the problem solving process. Such a device allows temporarily-needed information to be retrieved later in the same manner that writing down intermediate results in multiplication problems frees the person to perform further calculations. They also postulate that a diagram may act as a substrate or concept anchor that allows the new part of a problem to be described relative to well-understood problem base. Larkin and Simon discuss extensively the advantages of diagrams for facilitating inference about topological or geometric relationships [Larkin & Simon 1987]. Chandrasekaran and Narayanan [Chandrasekaran & Narayanan 1992], Novak and Bulko [Novak & Bulko 1992], Borning [Borning 1979] and others have also pointed out the usefulness of diagrams to human problem solvers as a device to aid in visualization, "gedanken experiments" or prediction. Finally, Novak and Bulko [Novak & Bulko 1992], Koedinger [Koedinger 1992] and others have explored the idea that diagrams may sometimes be used not primarily for making base-level inference, but rather to help in the selection of an appropriate method to solve a problem; that is, as an "aid in the organization of cognitive activity" [Chandrasekaran et al. 1993].

A salient feature of diagrammatic reasoning in many situations is its qualitativeness. People reason with diagrams to get rough, qualitative answers. If a more precise, quantitative answer is needed, they must resort to more formal, mathematical techniques. However, qualitative techniques are extremely useful in gaining valuable insight into the range of possible solutions. An initial qualitative understanding thus obtained can guide the later analysis for more detailed answers. In the context of structural analysis, knowing the qualitative deflected shape allows one to identify critical features of the shape. One can then set up relevant equations in order to obtain more precise information such as actual magnitudes of forces and displacements at specific points of interest.

How do diagrams actually help civil engineers to make qualitative inferences? From studying textbooks on elementary structural analysis, such as [Brohn 1984], that aim to develop a intuitive understanding of the response of the structure under a load, we find that diagrams fulfill many of the same roles as those articulated by researchers in other fields. First, diagrams are used as "a visual language of structural behavior that can be understood with the minimum of textual comments" [Brohn 1984]. The language allows the engineer to express explicitly the constraint or physical law that is relevant at each part of the proposed structure, in such a way that the constraints and some of the consequences are immediately apparent to the reader without further reasoning. Secondly, the diagram serves as a place holder or short-term memory device by allowing the designer to sketch out the result of one deformation and then go back to see if there is a further effect or interaction that needs to be addressed. Finally, visual inspection of diagrams seems to guide the engineer in choosing the next step, resulting in a more efficient problem solving process than it would be otherwise.

Having studied the use of diagrams in all these capacities in the context of determining deformation shape of frame structures, we have constructed REDRAW to use diagrams in all those capacities in ways similar to humans. We will first explain the deflection shape problem in Section 2. The architecture of REDRAW will be described in detail in Section 3.

2. Deflection Shape Problem

Determining the qualitative deflected shape of a frame structure under a load is a crucial step in analyzing the behavior of a structure. Structural engineers first make a simple, 2-D drawing of the shape of the given frame structure. Given a load on the structure, they modify the shape of the structural member under the load. They inspect the modified shape to identify the places where constraints for equilibrium of the structure are violated. Those constraint violations are corrected by modifying the shape of connected structural members, propagating deflection to other parts of the structure. This process is repeated until all the constraints are satisfied. The drawing thus produced shows the final deflected shape of the frame under the given load.

Given a diagram of a frame structure and a load, the program produces an underlying symbolic representation in order to facilitate reasoning about engineering concepts. Then the program will use its structural engineering knowledge to propagate constraints on the diagram of the structure and will inspect and modify this picture until a final shape is produced that represents a stable deflected structure under the given load.

As with the qualitative nature of human visual reasoning, the reasoning carried out by REDRAW is also qualitative. The answer it produces is a picture of a deflected shape. Although the resulting picture is qualitatively consonant with the problem solution, it is not (nor does it need to be) mathematically accurate or to scale.

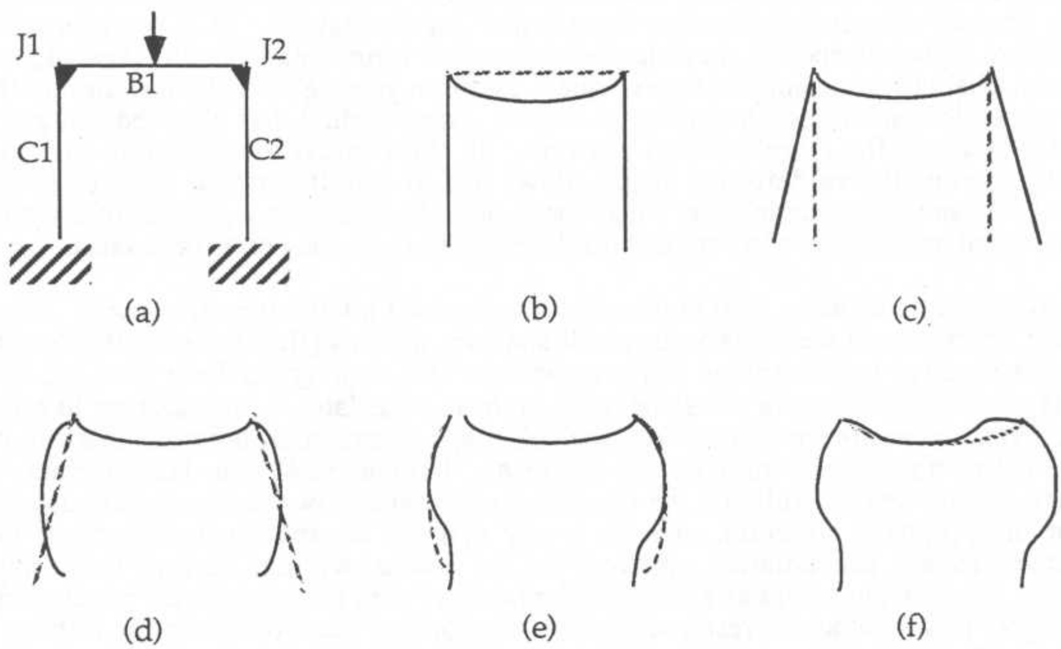

Figure 1: Steps in determining the deflected shape

REDRAW solves this type of deflected shape problems by directly manipulating a representation of the shape in the manner shown above. Although the problem could be solved by setting up equations, visualization is a indispensable first step that provides an engineer with an intuitive understanding of the behavior of the structure and enables her to recognize a good strategy for further analysis.

Before describing how REDRAW analyzes structures, we explain briefly the reasons for our choice of this deflected shape problem. An advantage of this civil engineering problem domain for studying the role of visual reasoning in problem solving is the fact that it is rich with domain-specific knowledge that has significant implications on how the diagram is manipulated and interpreted. One possible domain in which to study pictorial reasoning is geometry, where pictures are abstract diagrams without being a representation of anything in the world. In geometry, the only property one reasons about is the geometric property. There are no other types of information, apart from that represented in the diagram, that one must take into account when manipulating and inspecting the diagram.

In contrast, pictures used for reasoning in engineering design are not simply abstract geometric shapes but actually represent things in the real world. Furthermore, how a picture is interpreted and manipulated depends significantly on what it represents. For example, a line in our domain represents a beam or a column. Changing the length of the line would change the information represented by the diagram. In a circuit diagram, on the other hand, one could change the length or curvature of the line representing an electrical connection without changing the informational content of the diagram. For the goal of better understanding the role of visual reasoning in problem solving and its relation to symbolic reasoning, it is important for us to work with a problem requiring a wealth of domain knowledge that has significant influence on the way diagrams are used and interpreted.

3. Architecture of the system

From examining the way deflection shape problems are solved by humans, it is apparent that solving this type of problems requires not only an ability to manipulate and inspect diagrams but also substantial structural engineering knowledge. Structural engineering knowledge about the properties of various types of joints and supports is necessary to identify constraints on the shape for the structure to be in equilibrium. Such knowledge is best represented and manipulated symbolically. On the other hand, information about shapes is best represented as a picture. Many types of modification and inspection of the shape are also more easily carried out with a picture.

The requirement for both pictorial and non-pictorial representation and reasoning suggests a layered architecture. Thus, REDRAW includes both symbolic reasoning and diagrammatic reasoning components. The former contains the knowledge base of structural engineering knowledge about various types of structural members, joints, supports, and the constraints they impose on the structure. It also includes a constraint-based inference mechanism to make use of the knowledge. The latter, diagrammatic reasoning component includes an internal representation of the two-dimensional shape of the frame structure as well as a set of operators to manipulate and inspect the shape. These operators, some of which are shown in Figure 2, correspond to the manipulation and inspection operations people perform frequently and easily with diagrams while solving deflected-shape problems.

The Structure Layer contains a symbolic representation of domain-specific knowledge. It represents non-visual information (such as hinged joint rotation), various types of structural members, equilibrium conditions, as well as heuristic knowledge for controlling the structural analysis process.

The Diagram Layer represents the two-dimensional shape of a structure. There are several operators that directly act on this representation to allow inspection as well as transformation of the shape. These operators correspond to the operations people perform easily with diagrams. The internal representation of a shape is a combination of a bitmap whose elements correspond to each "point" in a picture, and a more symbolic representation where each line is represented by a set of coordinate points.

The Diagram Layer is independent of the structural engineering domain in the sense that it does not contain any structural engineering concepts. However, the types of both manipulation and inspection operators provided for the layer reflects the requirements of the domain. For example, the assumption that the frames consist of incompressible members made a particular set of operators necessary (e.g. the program requires a bend operator but not a stretch or compress operator), and also by the specific functioning of those required operators (for example, the bend operator creates a moderate curve rather than a complete bend that would cause the line endpoints to touch or cross; or, the inspect operator may look at components connected to the component in question, but will not compare that component to any other, as it might in some other domain.)

Structure Layer
- Objects: beams, columns, connections, supports, load, etc.
- Operators: generate-force-equilibrium-conditions, generate-moment-equilibrium-conditions, etc.

Diagram Layer
- Objects: lines, splines, circles
- Operators: Manipulation: rotate, bend, translate, smooth, etc.
  Inspection: get-angular-displacement, get-displacement, symmetrical-p, etc.

Figure 2. Types of objects and operators in REDRAW program

There is a close link between the information in the two layers. The system relates the representation of a particular beam in the Structure Layer to a spline in the Diagram Layer, and the concept of deflection of a beam to an operation on a spline to transform its shape. Likewise, the system is able to identify features of a shape (e.g. direction of bending, existence of an inflection point) and to communicate them to the Structure Layer.

Communication between the two layers takes place by sending commands and posting constraints by the Structure Layer, which is carried out or checked by the Diagram Layer. Figure 5 shows the two-layered architecture schematically. There is a translator between the two layers to mediate the communication between the two layers. When the Structure Layer posts a constraint or a command, the Translator translates it into a call to a Diagram Layer operator that can directly act on the representation of the shape to manipulate or inspect it. The result is again translated back to concepts that the Structure Layer understands.

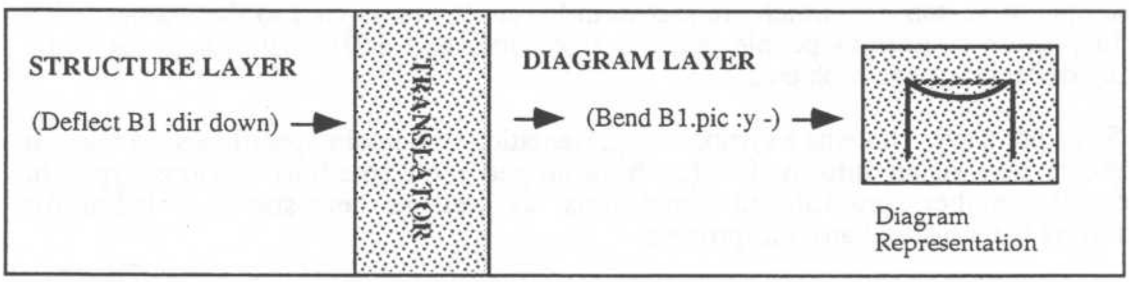

Figure 3: Two-layered architecture of the REDRAW program

The REDRAW program has been implemented and has successfully analyzed six of the 23 basic deflected shape problems described by Allen [Allen 1978]. An informal evaluation by a civil engineer shows that the program reflects the qualitative reasoning process used in analyzing frame structures, and that it would be useful in helping students and novice engineers learn to solve this type of problem.

3.1 Example

In this section, we illustrate the problem solving process by REDRAW with the example presented earlier in Figure 3.

We illustrate the type of communication that takes place between the layers. Given the frame structure of Figure 4(a), with a load, *Load3*, placed on it, the Structure Layer, S , sends a command, "Deflect *Beam3* in the same direction as the load," which the Translator, *T* , translates into an operation "Bend *Beam3.pic* in the negative direction of the y-coordinate." Carrying out this operation will result in the shape shown in Figure 4(b). *S* infers that since *Joint3* is a rigid joint, *Beam3* and *Column3* must remain perpendicular to each other at *Joint3*. *S* issues a query to test this constraint. The query is translated into "get the angle between *Beam3.pic* and *Column3.pic* at the ends connected by *Joint3.pic* for the Diagram layer, *D*. The answer, the actual angle between the two lines, is communicated to *S* as the answer that the constraint is not satisfied. *S* now issues a command to satisfy this constraint while keeping *Beam3* fixed, which is translated into "make the angle between *Beam3.pic* and *Column3.pic* at *Joint3.pic* be 90 degrees without modifying *Beam3.pic* for *D*. Carrying out the operation will result in the shape shown in Figure 4(c). Communication will continue in this manner until all the constraints are satisfied. Figure 5 shows REDRAW's symbolic reasoning activity for the same example.

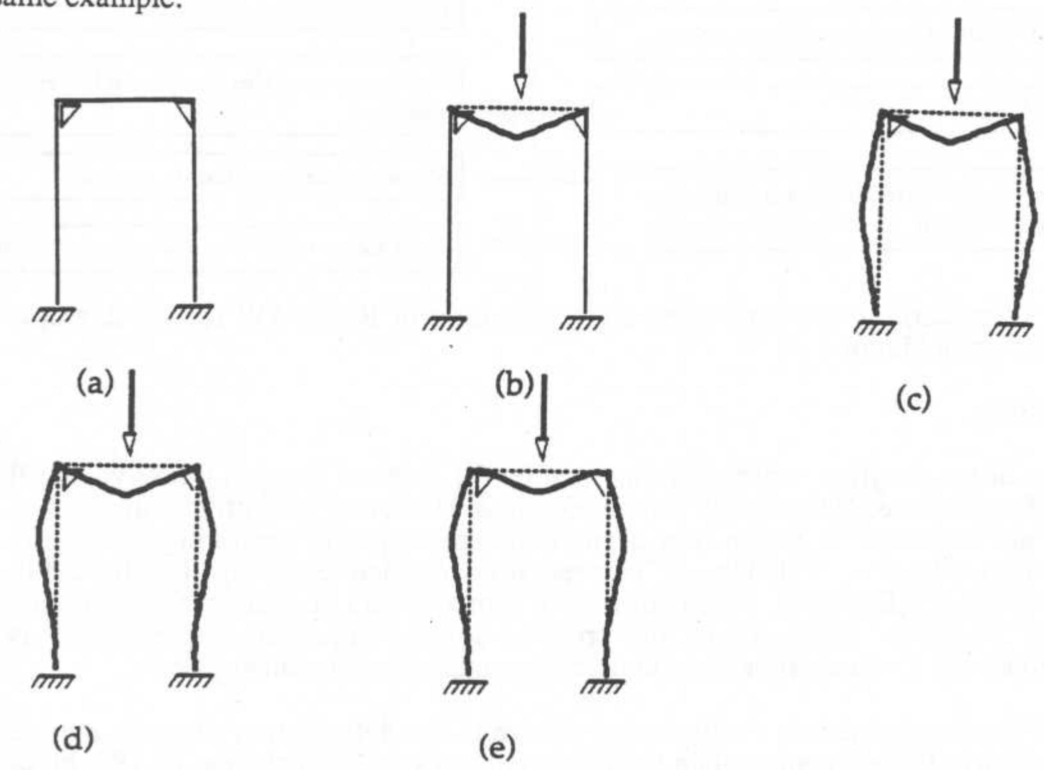

Figure 4. REDRAW solution to frame structure problem sketched in Figure 1

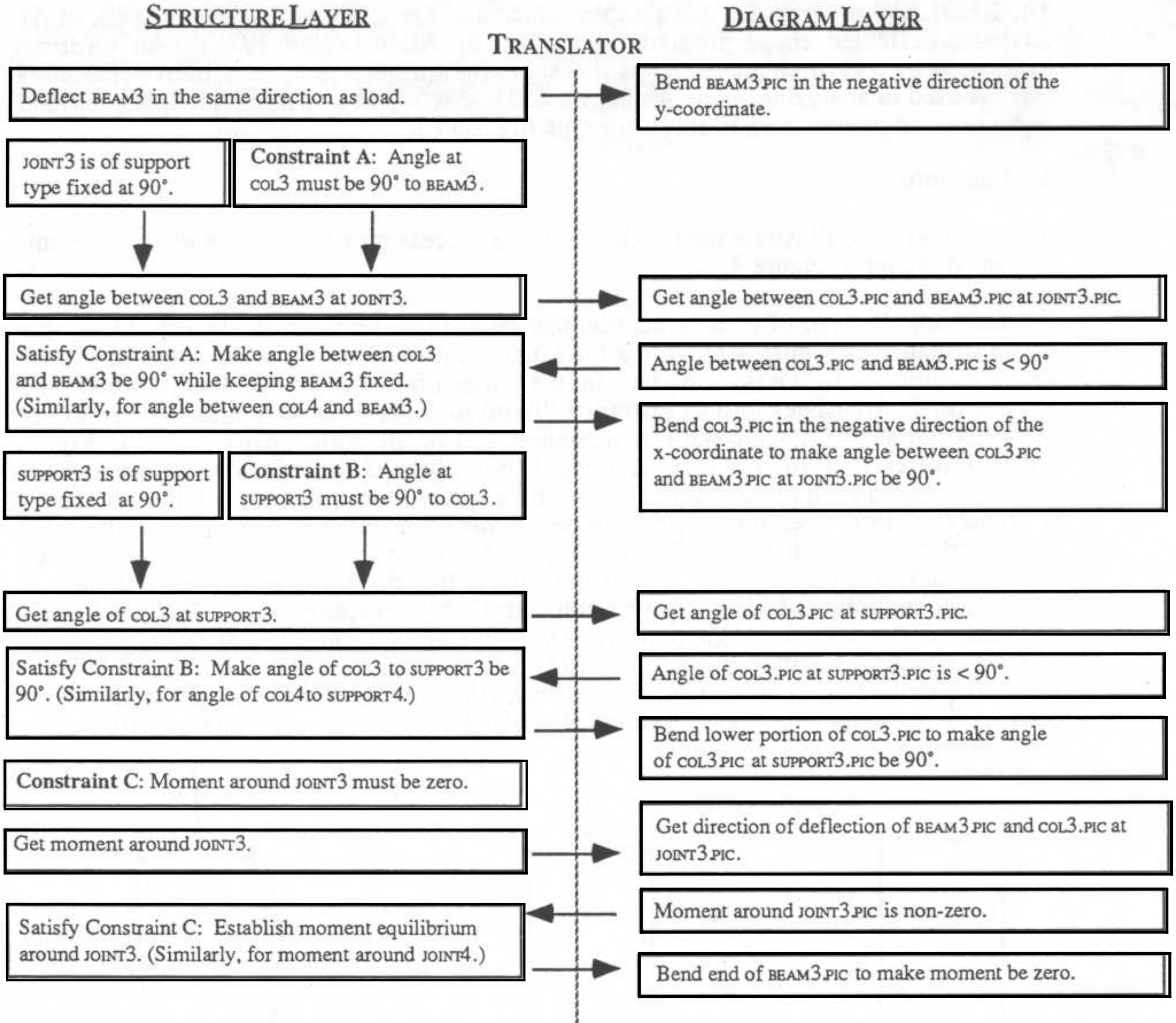

Figure 5: Illustration of the inter-layer communication of REDRAW for the example problem shown in Figure 4.

3.2 Discussion

The design of the program is greatly influenced by the ideas of Kosslyn [Kosslyn 1980] and Chandrasekaran & Naryanan [Chandrasekaran & Naryanan 1992] regarding human cognitive architectures, in which they argue that some types of reasoning are tightly coupled with perception. This idea of "perceptually grounded reasoning" is reflected in the architecture of REDRAW, which consists of symbolic and diagrammatic layers that are closely coupled. Furthermore, the problem solving approach of REDRAW is designed to mimic the qualitative structural analysis method of human engineers.

REDRAW produces a satisfactorily correct picture of the deformation shape in a more computationally efficient manner than a similar system, Qstruc [Fruchter et al. 1991b], in which a purely symbolic approach was taken to the same frame structure problem. The program architecture is unencumbered by the more complex features necessary to precisely calculate the true deformation of a frame structure under a load. Its purpose is rather to provide a good environment for studying diagrammatic reasoning, and how that type of reasoning is integrated with symbolic reasoning. This approach allows us to examine and model more readily the flow of pictorial and symbolic reasoning as well as to better identify the visual operators which are truly fundamental to that reasoning process.

The relationships among the pictorial objects are also quite straightforward. The objects relate to each other in qualitative spatial terms such as connected-to, near, left, right, above and below. Moreover, only those primitive geometric properties that are easily identified by visual inspection rather than by reasoning involving multiple steps are used in the process of determining the deformation shape of a structural component. Such properties include whether two lines are approximately parallel and whether the angle between them is acute, obtuse or right angle. The pictures are not drawn in accurate scale or proportion. Only such information as approximate relative size, shape and proximity are used to draw them.

The diagrammatic operators that have been implemented thus far meet only the minimum required to complete the target set of tasks. We are still in the process of determining what visual operators are essential to our diagrammatic reasoning task, how they should function, and how general these operators can be made. We initially intended all the diagrammatic operators, such as bend, rotate and smooth, to be domain- and task-independent. However, it has become clear that while some operators are domain-independent, others are quite domain- and task-specific. For example, our "bend" operator bends a straight line into a simple curve that resembles the curve even a novice would draw to indicate the shape of a stick under a load. However, the implementation of this "bend" operator reflects the assumptions implicit in the domain and the task -- for instance, the curvature of the bent line is large enough so that it can be clearly seen, but not so large that the structural member would appear to be broken. Also, the particular choice of inspection operators we have implemented reflect the nature of the problem we chose to work on. Inspection of a truss structure would encompass more components of the structure taken as one unit when evaluating its stability than is necessary when performing the same inspection task on the drawing of a frame structure.

REDRAW also shows us that the domain knowledge found in the Structure Layer influences how reasoning proceeds. With regard to the flow of the direction of attention through the diagram, the constraints applied in the symbolic layer contain implicitly the knowledge that deformations propagate from one component to those connected to it. Examination of the diagram thus also proceed from the component sustaining the original load to the components connected to it, and so forth. In addition, an issue arises concerning the necessity of a "local vs. extended" examination of a component in the propagation of the deformation. A hinge joint, for example, allows rotation of the components connected to it. The effect of the hinge on two connected components is localized at the connection point. A fixed joint, on the other hand, requires an examination of the type of attachment at the other end of the component so that an appropriate constraint can be applied and the correct deformation shape be imposed. Thus a more complex or extended examination of a component must take place to correctly implement the fixed joint constraint.

From the point of view of an engineer, the design of the program allows the user to concentrate on qualitative features of the structure, without requiring the specification of details. The diagrammatic components of the system facilitate the visualization of the particular deformation problem and its likely range of solutions. To aid in this visualization, the Diagram Layer operators include a "write-over" ability; that is, after a shape transformation, dotted lines show the original structure, just as a person draws a deformation right over the original line rather than create a separate new drawing.

Displaying the before and after shapes allows him to visually inspect and verify the inference process that was used in the shape transformation. The explanation facility of REDRAW, which explains every step of the reasoning process, provides the user with further insight into the constraints imposed and the inferences made to arrive at the final stable deflected shape.

4. Related Work

We have previously built a program called QStruc to solve the same deflected shape problem described in this paper, but using a traditional, symbolic AI approach. The program determines the qualitative values of forces, moments, and displacements in a frame structure under a load. The inputs to the system are a symbolic representation of the structure in terms of its members and connections, and a load on the structure. There is no explicit representation of the shape of a structure in the program. The shape is implicitly represented by the existence of such physical processes as bending, and the qualitative values (positive, negative, zero or unknown) of such parameters as displacements. QStruc has successfully analyzed several simple two-dimensional structures, thus demonstrating the feasibility of performing qualitative analysis of structures on a computer. However, our experience with QStruc shows us that a program of this type does not help an engineer to gain an intuitive understanding of the deflection process.

Within the artificial intelligence community, Lindsay's research in qualitative geometric reasoning [Lindsay 1992] is one notable work. He is developing a computational model of human visual reasoning in the domain of plane geometry. Lindsay uses constraint maintenance techniques to manipulate a diagrammatic representation to make inferences and test conjectures. His goal is to demonstrate that a combination of propositional and pictorial representations offers more psychologically plausible and computationally efficient ways of reasoning about mathematical problems.

Another work in progress is Chandrasekaran and Narayanan's on commonsense visual reasoning [Chandrasekaran & Narayanan 1990]. They propose a visual modality-specific architecture, using a visual representation scheme, consisting of symbolic representations of the purely visual aspects (shape, color, size, spatial relations) of a given situation at multiple levels of resolution. The visual representation is linked to an underlying analogical representation of a picture so that visual operations performed on the analogical representation are immediately reflected on the visual representation and vice versa. Chandrasekaran and Narayanan's objective is "to propose a cognitive architecture underlying visual perception and mental imagery that explains analog mental imagery as well as symbolic visual representations" [Chandrasekaran & Narayanan 1990].

5. Conclusion

This paper has described our approach to developing a system to better understand the role that visual reasoning plays in a concrete problem-solving context. We have built a prototype program that reasons qualitatively about pictures in the same way that people do. Our decision to work with the deflection of shape problem in the domain of civil engineering gives us two advantages: first, since we have already built a system to solve the deflection problem using a traditional symbolic approach, we can directly compare the pictorial and symbolic reasoning approaches; and secondly, this is a knowledge-rich, real-world domain, which will allow us to study the role of pictorial reasoning in solving problems that require both types of reasoning.

In addition to examining the role of diagrammatic reasoning in problem solving, we are considering the generality of our work and its extendibility to other areas of technical design such as in architecture and mechanical engineering. Larkin and Simon [Larkin & Simon 1987] show that even with a symbolic representation, problem solving efficiency in some cases can be greatly improved by organizing the information in a way that reflects the physical structure of the object represented. With a mixed symbolic and diagrammatic approach, interesting problems concerning the organization of the information and the computational complexity of the problem solving algorithm may arise that could later effect both scalability and generality. By developing a strong understanding of the role that visual reasoning plays in the overall problem-solving process, we hope to be able to construct a general tool that can be used to build diagrammatic reasoning systems in other problem domains.

References

Allen, R. (1978). Elementary Deflected Structural Theory. Unpublished manuscript. Department of Civil Engineering, Stanford University.

Barwise, J. & Etchemendy, J. (1990). Visual information and valid reasoning. Visualization in Mathematics, ed. W. Zimmerman. Washington, D.C.: Mathematical Association of America.

Borning, A.H. (1979). Thinglab -- a constraint-oriented simulation laboratory. Ph.D. dissertation, Stanford University.

Brohn, David (1984). Understanding Structural Analysis. BSP Professional Books, Oxford.

Chandrasekaran, B., & Narayanan, N. H. (1992). AAAI Symposium on Reasoning with Diagrammatic Representations, Stanford, CA.

Chandrasekaran, B., Narayanan, H., & Iwasaki, Y. (1993) Reasoning with Diagrammatic Representations: A Report on the AAAI Spring Symposium, March 25 - 27, 1992. To appear in AI Magazine.

Chandrasekaran, B., & Narayanan, N. H. (1990). Towards a theory of commonsense visual reasoning. Foundations of Software Technology and Theoretical Computer Science, Springer-Verlag.

Fruchter, R., Iwasaki, Y. & Law, K. H. (1991 a). Generating qualitative models for structural engineering analysis and design. Proc. The Seventh Conference on Computing in Civil Engineering, Washington D.C.

Fruchter, R., Law, K. H., & Iwasaki, Y. (1991b). "QSTRUC: An approach for qualitative structural analysis" in Artificial Intelligence and Structural Engineering.. Proceedings of the Second International Conference on the Application of Artificial Intelligence to Civil and Structural Engineering. Civil-Comp Press.

Iwasaki, Y (1989). Qualitative Physics. In The Handbook of Artificial Intelligence, Vol. 4 edited by A. Barr, P. Cohen, and E. Feigenbaum, Addison-Wesley.

Koedinger, K. (1992). Emergent properties and structural constraints: advantages of diagrammatic representations for reasoning and learning. AAAI Symposium on Reasoning with Diagrammatic Representations, Stanford, CA.

Kosslyn, S. M. (1980). Image and Mind. Harvard University Press, Cambridge, MA.

Larkin, J.H. & Simon, H.A. (1987). Why a diagram is (sometimes) worth ten thousand words. Cognitive Science, Vol. 11.

Lindsay, R.K. (1992). Diagrammatic reasoning by simulation. AAAI Symposium on Reasoning with Diagrammatic Representations, Stanford, CA.

Norris, C. H. & Wilbur, J. B. (1948). Elementary Structural Analysis. McGraw-Hill.

Novak, G & Bulko, W. (1992). Uses of diagrams in solving physics problems. AAAI Symposium on Reasoning with Diagrammatic Representations, Stanford, CA.

Qin, Y. & Simon, H.A. (1992). Imagery and mental models in problem solving. Proc. AAAI Symposium on Reasoning with Diagrammatic Representations., Stanford, CA.