MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE

A.I. Memo No. 702 **April 1983**

Representations for Reasoning About Change

Reid G. Simmons & Randall Davis

ABSTRACT: This paper explores representations used to reason about objects which change over time and the processes which cause changes. Specifically, we are interested in solving a problem known as geologic interpretation. To help solve this problem, we have developed a simulation technique, which we call *imagining*. Imagining takes a sequence of events and simulates them by drawing diagrams.

In order to do this imagining, we have developed two representations of objects, one involving *histories* and the other involving *diagrams*, and two corresponding representations of physical processes, each suited to reasoning about one of the object representations. These representations facilitate both spatial and temporal reasoning.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. It is a revised version of a paper which appeared in the proceedings of the ACM Workshop on Motion, April 1983, Toronto, Canada. Support for the laboratory's artificial intelligence research has been provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-80-C-0505.

CONTENTS

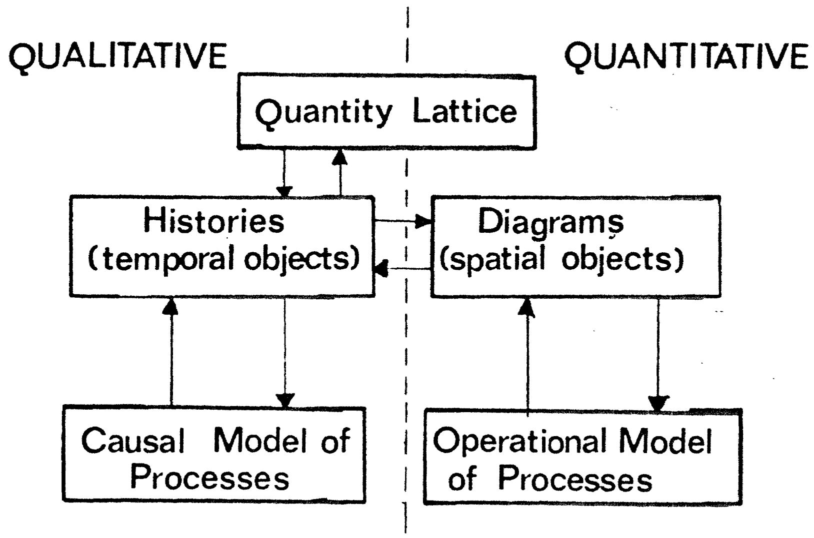
1. INTRODUCTION

A recent trend in artificial intelligence research is the construction of expert systems capable of reasoning from a detailed model of the objects in their domain and the processes that affect those objects [Davis]. We are currently developing a system built in this fashion which is designed to solve a class of problems known as geologic interpretation (see, for example, [Shelton]): given a cross-section of the Earth's crust (showing formations, faults, intrusions, etc.), hypothesize a sequence of geologic events whose occurrence could have formed that region. Solving this problem requires reasoning about change, in particular, spatial change. Doing this reasoning, in turn, requires representing objects, which show the effects of change, and processes, which are the causes of those changes.

A major focus of this research is to explore the machinery needed to represent and reason about both mutable objects and the processes that induce changes in them. To do this, we have developed two representations of objects, a qualitative representation called *histories* and a quantitative one called *diagrams*. We have also developed two corresponding representations of physical processes, each suited to reasoning about one of the object representations (see Figure 1). In addition, we have developed the *quantity lattice*, used for numeric reasoning, which contains both qualitative and quantitative elements. We have been careful to keep these representations well separated, limiting their interaction to a relatively small and clearly defined interface.

These representations were developed to enable us to perform a type of simulation which we call *imagining*. Imagining takes a sequence of geologic events and a goal diagram depicting a cross-section of the Earth, and simulates the effects of the sequence of events by constructing a series of diagrams.

Fig. 1. Information Flow Between Representations



Section 2 provides an outline of the major foci of interest in the paper. In Section 3, we describe the basic task of geologic interpretation, present a simple problem and demonstrate its solution. Section 4 describes the two representations of objects, while Section 5 presents the corresponding representations of physical processes. In Section 6, we show how our representations facilitate the *imagining* of a sequence of events, and in Section 7 we explore the utility of using multiple, specialized representations. Section 8 presents a comparison with related work.

2. OVERVIEW

Our concerns in this paper focus around four main issues, reviewed briefly here and explored in more detail in the remainder of the paper:

2.1 The Representation of Mutable Objects and Processes

In order to *imagine* the occurrence of geologic events, we need to reason about two basic types of changes to objects. First, objects have a life-span, that is, they exist for a certain period of time and can be created or destroyed. In our current domain of geology, for example, a rock can be created by deposition or destroyed by erosion. Second, an object has various attributes whose values can change over time. Again in geology, the attributes of a rock include its composition, thickness and location in space, all of which are subject to change over time.

Since changes to objects are caused by the occurrence of physical processes, we are also concerned with representing processes. The process representation must facilitate reasoning about which objects were created or destroyed and how the attributes of various objects changed.

2.2 The Organization of Representations to Facilitate Reasoning

One aspect of solving the geologic interpretation problem involves reasoning about the specific change to an object between two instances in time. Since most geologic changes are spatial in nature (e.g. a change in shape due to erosion), we have developed a special representation for reasoning about the spatial characteristics of objects at specific instances of time. Another aspect of solving the problem involves reasoning about the cumulative effects of changes over time (e.g. the overall effect on the location of a rock due to a sequence of uplifts, tilts and faultings). We have developed a second specialized representation specifically suited to reasoning about such changes. In addition, we have developed corresponding representations for processes, one suited for reasoning about spatial changes, the other suited for reasoning about temporal changes.

Spatial reasoning is done using *diagrams*, represented as collections of *vertices, edges*, and *faces*. The character and organization of diagrams facilitates inferences about changes in shape, location, orientation, etc. Temporal reasoning is done using *histories*. The history representation is frame-like, but the value of an attribute is a *time-line*, which is a sequence of values over time, rather than a single value. This time-line of values facilitates reasoning about the sequence of changes to an object.

2.3 The Use of Multiple, Specialized Representations

With five different representations in the system, we have of necessity been careful in organizing their design and interaction. The modularity suggested in Figure 1 has been one important principle for organizing the interaction and has aided us significantly. We have also defined selection criteria that allow us to design enough specialized representations to meet our needs, without permitting representations to proliferate unnecessarily. We find that two simple questions provide significant guidance: what do we want to describe about the world, and what questions do we want to answer using those descriptions. Section 7 shows how these criteria have guided the selection of our current set of representations.

2.4 The Use of *Imagining* and Simulation in Problem Solving

Our overall approach to the problem of geologic interpretation has much in common with generate and test. One part of the system generates a candidate solution while another tests it against the given cross-section (see Section 3).

Since a candidate solution is a sequence of geologic events, it is tested, in a process we call *imagining*, by simulating the effects of each event in turn and comparing the final result against the given cross-section. Unlike traditional generate and test, however, the test is not simply a binary predicate and failing the test does not necessarily disqualify the candidate. A discrepancy between the result of the simulation and the cross-section can provide important information for augmenting the solution, information that may be impossible to infer otherwise.

This interaction between candidate generation and simulation illustrates a useful approach to the integration of local and global information in problem solving. By "local", we mean the kind of information that can be found by examining a single rock or single boundary in the diagram. By "global" we mean the overall consistency of the proposed solution. Although each individual process may be plausible, we need to determine the plausibility of the entire sequence, that is, does it produce the desired result? As we will see in the next section, the candidate solutions are pieced together from local information; *imagining* then provides an important check on the global consistency of the solution.

3. GEOLOGIC INTERPRETATION

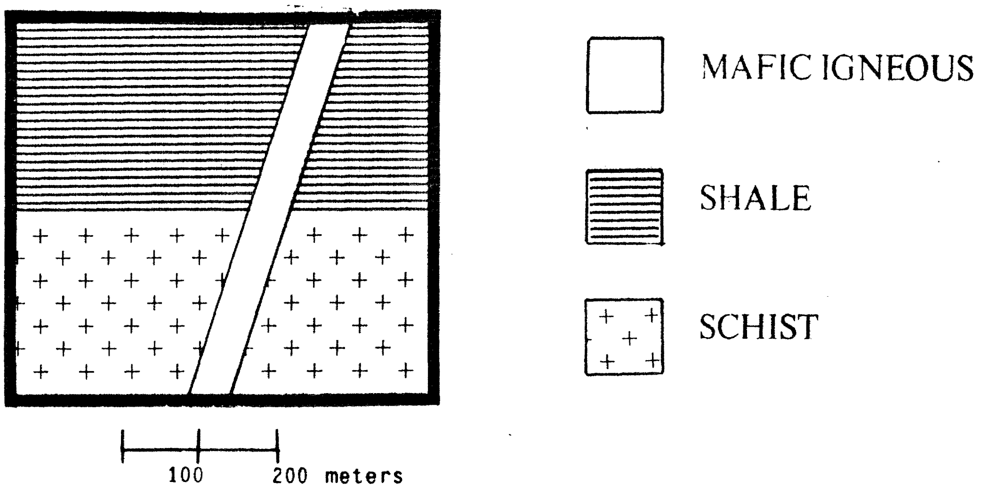
3.1 An Example

In the geologic interpretation problem, we are given a diagram that represents a vertical cross-section of a region along with a legend identifying each kind of rock formation (Figure 2a). The task is to infer a sequence of geologic events that plausibly could have formed the region.

A geologist typically approaches this problem by looking at boundaries between rocks and making a collection of simple inferences in an attempt to build up a sequence of events. In this case, for example, he might note that, since the **mafic-igneous** crosses the **schist**, it intruded through (i.e., forced its way through) the **schist** and hence is younger (Figure 2b, step 1; the sequence of partial orders shows the geologist's solution at each stage of development). The same reasoning would indicate that the **mafic-igneous** also intruded through the **shale** (Figure 2b, step 2). Thus the **shale** and the **schist** were both in place before the **mafic-igneous** intruded through them. To determine the order in which the **schist** and the **shale** appeared, the geologist would infer that, since sedimentary rocks are deposited from above onto the surface of the Earth, the **shale** (a sedimentary rock) must have been deposited on top of the **schist**, and hence is younger than the **schist** (Figure 2b, step 3). The geologist knows that the **schist** was created from existing rock by the process of metamorphism. However, metamorphism occurs to rocks buried deep in the Earth and deposition occurs on the surface, so somehow the **schist** must have gotten from the depths to the surface, in order for the shale to have been deposited upon it. The geologist might infer that a combination of the processes of *uplift* and *erosion*, neither of whose effects are reflected in the diagram of Figure 2a, would suffice to bring the **schist** to the surface (Figure 2b, step 4). The final inferred sequence of events is shown in Figure 2c.

Fig. 2. Simple Geologic Interpretation Problem

A. Geologic Cross-Section and Legend



B. Sequences of Partial Orders

1. **mafic-igneous ⇒ schist**
2. **mafic-igneous ⇒ schist**
   ⇒ shale
3. **mafic-igneous ⇒ schist ⇒ shale**
4. **mafic-igneous ⇒ uplift ⇒ erosion ⇒ schist ⇒ shale**

C. Solution of Geologic Interpretation Problem

1. Metamorphose **schist**
2. Uplift and erode to uncover the **schist**
3. Deposit **shale** on **schist**
4. Intrude **mafic igneous** through **schist** and **shale**

3.2 Problem Solving Technique

The problem solving technique used in the example above consists of two basic phases. In the first phase, we use a technique we call *scenario matching* to generate a sequence of geologic events that might explain how the cross-section came into existence. In the second phase, we use a technique we call *imagining* to test if the hypothesized sequence is correct. In addition, if the hypothesis is not correct we debug the hypothesis using a technique we call *gap filling*.

3.2.1 Scenario Matching

Scenario matching is a way of generating a sequence of events by reasoning backwards from the effects of processes to their causes using simple, one-step inferences. A scenario is a pair consisting of a diagrammatic *pattern* and a sequence, called an *local interpretation*, that could have caused that pattern. For example, in solving the example in Figure 2 we used the following scenario twice:

pattern	*local interpretation*
<rock> | **<igneous>** | **<rock>**	**<igneous>** intruded through the **<rock>**

A *pattern* represents the local effects of a geologic process and typically involves the boundaries between two or three formations. A *local interpretation* is a sequence of events that is a possible causal explanation for the pattern's occurrence. Each pattern may have several plausible interpretations. Although we have not further developed the scenario matcher, we have identified about a dozen scenarios, each consisting of from one to three interpretations, which we believe are sufficient for solving most geologic interpretation problems.

By matching scenario patterns throughout the diagram and combining the local interpretations obtained from the matches, we generate sequences that purport to explain how the region was formed. However, these sequences might not be completely valid for two reasons. First, local consistency does not imply global consistency. For example, if a local interpretation infers that a global process like tilting occurred, the whole sequence must be consistent with this occurrence of tilting. Second, the evidence for the occurrence of some physical processes might no longer exist in the geologic record (as reflected by the diagram). For instance, there is no evidence in Figure 2a for the occurrence of the processes of uplift and erosion of the schist, because the erosion has removed whatever once covered the schist. To detect both types of inconsistencies, some form of global reasoning is needed.

3.2.2 Imagining

We are developing a new simulation technique called *imagining* to detect inconsistent hypotheses. Based on the intuition of "viewing events in the mind's eye," imagining takes as input an initial state, a goal state (in our case, the diagram cross-section) and a sequence of events. The imaginer simulates each of the events in turn, producing a final state that is matched against the goal state. If the match is successful, then we can conclude that the sequence is a valid explanation for the formation of the goal state.

Aside from the final match, the imaginer has three tasks to perform for each event in the sequence.

1. It determines whether an event is applicable in the current state.
2. It determines quantitative values for the parameters of the events.
3. It simulates the event, in our case by modifying the diagram to reflect the geologic changes induced by the event.

In the rest of this section we discuss these tasks.

For each event, the imaginer must determine if it can be applied to the current state produced by the simulation. For example, an event might indicate "erode shale to sea-level", but clearly this would be inapplicable if the top of the shale was currently below sea-level. If the imaginer cannot continue, it should return an explanation of the problem encountered. This explanation would consist of the event that the imaginer could not simulate and the difference between the current state and the state that would be needed in order to simulate that event. In the above example, the difference reported would be that the shale is below sea-level, but should be above sea-level in order for the erosion to occur.

The sequence inferred by the scenario matcher (see Figure 2b) does not indicate values for the parameters of the events (such as the thickness of a deposition or the angle of an intrusion). In order to make tractable the problem of matching the goal state and the final state produced by the simulation, the parameters used in the simulation of an event must closely match those parameters used in the actual geologic process. For example, in order to simulate "deposit shale on schist" the imaginer must have some indication of the *thickness* of the shale formation. Thus, to do imagining requires that the system be able to infer values for the parameters of the geologic events being simulated.

The system uses measurements taken from the diagram, along with knowledge of geologic processes, to determine these parameters. Since each parameter represents some real-world quantity, we begin by measuring the quantity in the goal diagram. Then, we need to compensate for any changes that occurred to the quantity between the time when the event occurred and the time represented by the goal diagram.

A simple example will illustrate this parameter determination process. Suppose we wish to find the thickness of the **schist** when it was originally deposited. We can measure the current thickness of the schist formation in Figure 2a (which turns out to be 300 meters). However, since we also know that part of the original **schist** deposit had been eroded away earlier (in step 2, Figure 2c), we infer that the original thickness of the **schist** must have been greater than the measured thickness in the diagram. Since we cannot infer the exact amount of the erosion, the best we can do is to say that the original thickness was "greater than 300 meters". Reasoning in this fashion, we can establish *ranges of values* for the parameters of all the events. We can then use these to approximate in our simulation the effects of the actual geologic events.

The actual simulation phase of the imaginer is accomplished by constructing a sequence of diagrams, one for each event in the hypothesized sequence, to reflect the effects of our model of geologic processes. The use of diagrams is not crucial to the concept of imagining, but is useful in this case for two reasons. First, most geologic effects are spatial in nature, hence their changes are easier to represent in a diagram, which is a spatially organized representation. Second, an important check on the validity of the hypothesized sequence of events is to match the goal diagram against the final diagram produced by the simulation. Diagrams are thus useful for describing the effects of the changes and for validating the hypothesized sequence of events.

3.2.3 Gap Filling

If the imaginer detects a "gap" between the state needed for some event to occur and the actual state produced by the simulation (as would have occurred if we had not inferred the presence of the uplift and erosion in Figure 2), we need to hypothesize some sequence of events to fill the gap. As described in Section 3.2.2, the imaginer indicates why it could not continue in terms of the difference between two states, and from that, one can reason about which process or sequence of processes would have the effect of eliminating that difference. This is essentially means-end analysis [Newell, 1963] used in a restricted context.

3.3 Geologic Vocabulary

There are three basic geologic features which we need to reason about -- rock-units, boundaries, and geologic points. A *rock-unit* is simply a mass of rock. It can be of homogeneous composition, such as "the shale formation", or can include different kinds of rocks, such as "the down-thrown block of the fault". A *formation* is a rock-unit which is of homogeneous composition and was formed by a single event. For example, a shale formation is created by deposition, and a mafic-igneous formation is created by intrusion.

A *boundary* is the intersection between two rock-units, or between a rock-unit and the outside world. For example, a fault is the boundary between the rock-units forming the up-thrown and down-thrown blocks (the rock-units which move in relation to one another due to the faulting). The surface of the Earth is the boundary between the air or the sea and the existing rock-units of the region. A *geologic point* is a "piece of rock" which we want to reason about. For example, "the top of the shale", "the bottom of the surface of the Earth", and "the center of the sandstone" are all geologic points.

The geologic model we employ is a simple model known as "layer cake" geology (see, for example, [Friedman]), because it assumes horizontal depositions that stack up on top of each other like the layers of a cake. Erosion also occurs horizontally, like a knife slicing horizontally through the region. The "layer cake" model also deals with the spatial relationships between rock-units, rather than their internal characteristics. It is a good first approximation of geology and is adequate for solving most geologic interpretation problems.

4. REPRESENTING CHANGE IN PHYSICAL OBJECTS

The remainder of this paper concentrates on the representations and reasoning necessary to do imagining on a sequence of geologic events. In the previous section, we saw that in order to *imagine* a sequence of events, we need to (a) reason about how objects have changed over time due to the effects of the events, (b) determine values for the parameters of the events in order to approximate the effects of the actual geologic events and (c) simulate the effects of the events by modifying a diagram. This need for temporal, numeric and spatial reasoning has led to the representation of objects based on *histories, diagrams*, and a *quantity lattice*.

4.1 Histories

We have developed a representation for physical objects, which we call *histories* (the term is adopted from [Hayes]), that facilitates reasoning about the sequence of changes to objects. Objects are represented as frame-like structures (as in [Minsky]), organized into a type hierarchy. Each type of object has certain attributes associated with it and possibly some associated constraints. For example, a rock-unit has a "thickness" that is constrained to be positive.

To facilitate temporal reasoning, we have modified the basic frame representation in two ways. First, we associate a life-span with objects, enabling us to reason about when they were created or destroyed. Second, since we want to represent the situation in which the attributes of objects can change over time, the value of an attribute is represented as a *time-line*, rather than as a single value. A time-line is simply a totally ordered sequence of values over time. A time-line is divided into intervals, each of which represents the value of the attribute during a particular temporal interval. For instance, the "thickness" of a rock-unit is the sequence of all thickness values of that rock-unit over time.

Each distinct point in the time-line represents, by definition, an interval during which some change occurred to the attribute. Since we assume a "causal model" of the universe, that is, only physical processes can cause changes, each distinct point in the time-line is also associated with a process that caused the change. For example, one of the effects of erosion is represented by an interval in the "thickness" time-line of an affected rock-unit, indicating that the thickness of the rock-unit decreased as a result of the erosion.

4.1.1 The @ Operator

Since the attribute of an object is a time-line of values rather than a single value, we need a way to select the value of an attribute at a particular point in time. We have defined the @ operator for this purpose.

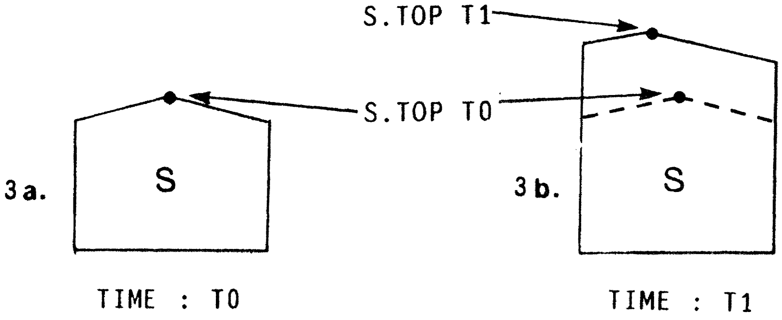
To illustrate the use of the @ operator, suppose that **S** represents a rock-unit. We use the dot notation to indicate attributes, so **S.thickness** refers to thickness, in fact, to all the thickness values over time. The referent of the expression **S.thickness@t0** is the thickness of **S** at time **t0**. If later **S** were partially eroded, then the thickness of **S** would change, and **S.thickness@t1** would not equal **S.thickness@t0** (assuming **t1** postdates the erosion process).

We have developed a formal notation that enables us to refer to the attributes of objects at a point in time. The BNF grammar for this notation is:

<temporal expression> :: = = <historical expression>@<time>
<historical expression> :: = = <object> | <historical expression>.<attribute>
<object> :: = = <temporal object> | <abstract object> | (<temporal expression>)

This notation is especially useful in dealing with more complex temporal expressions. For example, **S.top** is the time-line of the highest points of the rock-unit **S** (Figure 3). **S.top@t0** refers to the highest point of the rock-unit at time **t0** (Figure 3a) and **S.top.height@t0** refers to the height of that point at time **t0**. If more deposition occurred between **t0** and **t1**,

Fig. 3. Top of S Before and After Deposition



---

then the point referred to by the expression **S.top@t0** would not be the same point as the one referred to by the expression **S.top@t1** (Figure 3b). Note, however, that **S.top@t0** refers to a point that is still part of **S** at time **t1**, although it is no longer the top. Thus it makes sense to talk about **(S.top@t0).height@t1**, that is, the height at time **t1** of the point that was the top of **S** at time **t0**. This could be different from **S.top.height@t0** if, for instance, uplift occurred between **t0** and **t1**.

Since objects can be created and destroyed, it is useful to define the @ operator over objects as well as over attributes. If **A** is a history object, we define the value of **A@t** to be **A** if **A** exists at time **t**, otherwise the value is ⊥. ⊥ (bottom) is a special value which indicates "the query does not make sense." It is different from the value *unknown*, which indicates that the system has incomplete knowledge of the situation. In addition, ⊥ is a *strict* value, that is, any function applied to ⊥ returns ⊥.

In light of this, let us re-examine the interpretation of the expression **S.thickness@t0**. Since the referent of **S** might be ⊥ at **t0**, we need to "distribute" the @ operator through the expression to determine the value of the expression. The expression **S.thickness@t0** is in fact shorthand for **(S@t0).thickness@t0**. This is interpreted as follows: if **S** exists at **t0** then the value of the expression is the same as before; if **S** does not exist (e.g. it was "destroyed" by erosion or not yet deposited), then the referent of **S@t0** is ⊥ and the value of the whole expression is ⊥.

The general rule for expanding temporal expressions is to recursively replace occurrences of the form

<historical expression>.<attribute>@<time>

by the form

(<historical expression>@<time>).<attribute>@<time>

Thus the expression **S.top.height@t0** is shorthand for **((S@t0).top@t0).height@t0**, and **(S.top@t0).height@t1** is shorthand for **(((S@t0).top@t0)@t1).height@t1**.

4.1.2 Implementation of Histories

The temporal aspects of history objects are implemented in a straightforward manner. Each object has slots indicating the "start" and the "end" of the object. The "end" slot may remain unfilled, indicating that we do not know when the object was destroyed. The system will assume that an object continues to exist unless explicitly told otherwise.

The attribute time-lines are implemented as lists of intervals. There are two types of intervals - quiescent and dynamic. A quiescent interval indicates that nothing happened to the attribute during the interval, hence the value within the interval is constant. A dynamic interval indicates that some process induced a change during that interval to the attribute represented by the history. For reasons discussed in Section 5.1, the value within a dynamic interval is defined to be *unknown*.

To determine the value of an attribute at a particular time, the @ operator searches the time-line of the attribute to find the interval which contains that time point and returns the value found there. If the time point falls outside of the extent of the history time-line, then the value ⊥ is returned.

4.2 Diagrams

Histories are useful for dealing with certain types of changes, essentially characterized as one-dimensional. For example, the fact that the height of a point in a formation will increase if the formation undergoes uplift is well described using histories. However, many of the effects of geologic processes are two- or three-dimensional in nature, such as the change in shape of a formation caused by erosion, or the change in which point is the "top of the surface of the Earth" caused by deposition. To facilitate reasoning about these types of changes, we have developed methods for representing, reasoning about and manipulating diagrams.

In our system, a diagram represents a geologic cross-section, or more precisely, a 2-dimensional spatial abstraction of a geologic region at a particular point in time. By "spatial abstraction" we mean that diagrams represent only the geometric aspects, such as the size, shape and location of objects, and spatial relationships, such as *above* and *below*. In particular, there is no reference in the diagram to geology. In general, we have been careful to distinguish and separate the geologic representation from the geometric representation. They interact only through a small, simple and clearly defined interface (Section 4.3). This separation allows us to develop and reason about the two representations independently.

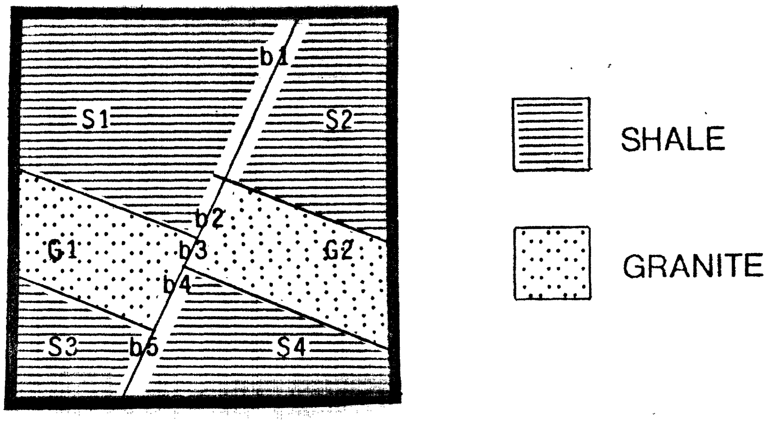
4.2.1 Diagrammatic Representation

A diagram consists of a collection of *vertices, edges,* and *faces*. Part relations, such as all the edges surrounding a face, or the end-points of an edge, are explicitly represented. Spatial relations, such as adjacency, "above" or "below", can be determined easily using the diagram. In addition, we can easily measure many metric properties, including the length of an edge, the location of a vertex and the maximum width of a face.

To illustrate the use of diagrams, we present a typical cross-section in Figure 4. The shale rock-unit is represented by the diagram faces S1, S2, S3 and S4, and the granite rock-unit is represented by the faces G1 and G2. In addition, the fault boundary is represented by the edges b1, b2, b3, b4 and b5. This correspondence enables us to determine many spatial and metric properties of the objects. For example, we can easily determine which rock-units are adjacent to the fault boundary by finding the faces adjacent to the edges b1 - b5 (the faces S1, S2, S3, S4, G1 and G2) and determining which rock-units those faces represent (the shale and granite). We can determine the orientation of the fault by averaging the angles of all the edges that represent the fault boundary.

Another use of diagrams, needed for the simulation phase of the imaginer, is in representing the effects of processes on objects. Since diagrams are a spatial abstraction of geologic objects, we can represent how objects change spatially by manipulating the diagram in accordance with our model of geologic processes. For example, as illustrated in Section 5.3, deposition can be simulated by drawing the new formation in the diagram.

Fig. 4. Simple Diagram Cross-Section



4.2.2 Implementation of Diagrams

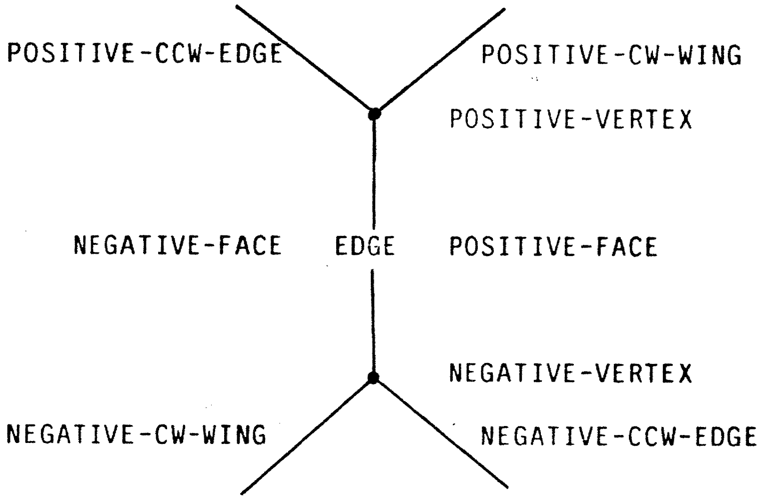
Our implementation of diagrams is based on the wing-edge structure of [Baumgart], adapted to 2-dimensional diagrams.

The primitive objects in this representation are *vertices, edges,* and *faces*. A vertex is represented by its (X,Y) coordinate position and has a pointer to one of the edges surrounding it. A face has a pointer to one of the edges of its perimeter. An edge is represented as shown in Figure 5. Each edge has pointers to exactly two faces, two vertices, and four "wings" (that is, the edges which share a common face and vertex). From these connections, we can easily compute such things as the perimeter of a face, the length of an edge, or the spatial relationship between two faces.

The wing-edge structure is well suited to our needs for three reasons. First, the primitive objects used in the representation -- faces, edges and vertices -- have a natural correspondence with the primitive objects used in the geologic representation -- rock-units, boundaries and geologic points. Second, the representation enables us to determine easily

---

Fig. 5. The Wing-Edge Representation of an Edge



the spatial relationships (such as "above") and metric properties (such as "angle of slope") that we need to do imagining. Third, the wing-edge representation was designed to facilitate manipulation of the geometric structures, which makes it easy to do the diagrammatic simulation of geologic processes. In particular, local changes to a diagram (such as adding or deleting edges or faces) can be accomplished with only local changes to the wing-edge structures.

There are only a few types of manipulation that we need to perform on diagrams to simulate all of the geologic processes we currently handle.[1] These manipulations are adding and deleting edges, faces and points; rotating and translating the entire diagram; splitting one diagram into two diagrams; and joining two diagrams into one. The relatively small number of primitive operations needed to simulate a large class of geologic processes suggest that diagrams are an appropriate form of representation, and that our vocabulary of primitive operations is reasonably well chosen.

4.3 Diagram-History Interface

As mentioned earlier, the interface between the history and diagram representations is relatively simple. Basically, it consists of a one-to-one mapping between primitive elements in each domain. A diagram corresponds to the world at a particular instant of geologic time. Each edge in the diagram corresponds to a single geologic boundary; each face corresponds to a single rock-unit; each vertex corresponds to a geologic point, such as the top of a rock-unit. Similarly, collections of rock-units or boundaries map into collections of faces or edges. So, for example, the collection of faces S2, G2, and S4 in Figure 4 corresponds to the rock-unit which is known as the *up-thrown block* of the fault.

---

[1] They are deposition, erosion, uplift, subsidence, intrusion, faulting and metamorphism.

In addition, there are several functions which map the spatial and metric relations in the diagram to the corresponding relations in the geologic world. For instance, we can determine if one rock-unit is above another by seeing if the corresponding faces in the diagram are above one another. Similarly, we can determine the orientation of a boundary by measuring the angle of slope of the corresponding edge.[2]

4.4 The Quantity Lattice

As discussed in Section 3.2.2, a major task of the imaginer is to determine parameter values for each event which approximate those values actually used in creating the geologic region. We have developed the quantity lattice to represent numeric values and to enable us to do arithmetic on and to determine ordering relationships between numeric values.

Due to the incomplete nature of the geologic record (the diagram), we cannot always determine numeric parameter values precisely. Thus we include both qualitative and quantitative elements in the quantity lattice. For example, we must be able to represent both that "the thickness of the shale = 500" and "the amount of uplift is greater than zero". The quantity lattice encodes ordering relationships using a partial ordering of quantities and encodes the numeric values of a quantity using a real-valued interval.

A quantity is simply an object which assumed to have a real number value, but typically we do not know that value precisely (see also [Forbus, 1982]). As a result, often the best we can do is to establish its relationships with other quantities. Thus, asserting that "$T1 < T2$" and "$T2 < T3$" indicates that all we know about the value of quantity T2 is that it lies between T1 and T3. Since our task domain also requires the concept of magnitude, we have extended this basic idea to include ordering relationships with real numbers. Thus, we can assert that "$T1 > 1$" and "$T2 \leq 100$".

---

[2] The definitions easily generalize for objects corresponding to collections of faces or edges.

To represent the relationships among quantities, we maintain a network of partial orderings. When we assert an ordering relationship between two quantities, a link is added to the network describing the relationship. For example, if we assert "$A \geq B$", the quantity **A** will have a "$\geq$" pointer to **B**, and **B** will have a "$\leq$" pointer to **A**. To determine if the relationship "X" holds between two quantities, the lattice is searched for a path of "X" links connecting the quantities.

Sometimes, however, the value of a quantity can be determined more directly, avoiding a search of the lattice. For example, suppose we assert that "$B \leq 1.1$" and "$A > 3.25$". From this we can conclude that $B < A$. We would like the quantity lattice to indicate this fact without explicitly recording that $1.1 < 3.25$. We accomplish this reasoning by associating with each quantity a real-valued interval. The value of the quantity is constrained to lie somewhere within the interval. This provides an efficient way to determine ordering relationships. If two intervals do not overlap, then the ordering relationship can be determined by comparing the limits of the interval, avoiding a search of the lattice. For example, since we know that "$B \leq 1.1$", we associate it with the interval $(-\infty,1.1]$[3] and similarly **A** is associated with the interval $(3.25,\infty)$. From this we can easily determine that $B < A$. To maintain these intervals, whenever an ordering between two quantities is asserted in the quantity lattice, the system checks to see if the range of one of the quantities can be constrained by the ordering and the range of the other quantity. For example, suppose **C** and **D** are quantities and assume that the interval range of **C** is $[0,\infty)$ and the interval range of **D** is $[1,\infty)$. If we assert that $C > D$, then the system will narrow the range of **C** to $(1,\infty)$. This narrowed range propagates to all quantities for which **C** has a "<", "$\leq$", or " = " link.

---

[3] A parenthesis indicates an open interval, a bracket indicates a closed interval.

This real-valued range is useful for another reason. As the values of quantities are known more precisely, more precise arithmetic operations may be performed. For example, if we know that **A** > **B** we know nothing about the relationship between **A** and **B** + **B**. However if we know that **A** lies within the interval [3,6] and **B** lies within the interval [0,1], then we can compute that **B** + **B** lies within the interval [0,2], and we can infer that **A** > **B** + **B**.

We have also found the quantity lattice to be very useful in doing temporal reasoning. A major component of temporal reasoning is reasoning about temporal relationships between points of time -- recall that to select an attribute value at a point in time we need to search the time-line to find the interval which contains the time point. By implementing time points as quantities in the lattice, we can use the mechanism described above (i.e., searching the lattice for a path between the quantities) to determine temporal relationships.

5. PROCESSES

Our chief interest in this paper is reasoning about how physical objects change. Since processes are the cause of change, our representation of processes focuses on describing them in terms of the changes they produce.

The previous section discussed histories and diagrams, two representations of objects which were developed to facilitate reasoning about different types of changes. We have also developed two corresponding representations for processes, one suited to dealing with histories, the other suited to diagrams.

5.1 Level of Representation

Both types of process representation make use of an "end-point" model of geologic processes. This model assumes that we can know the values of the affected attributes only at the beginning and end of a process and that nothing can be assumed about the intermediate values. For example, the composition of a rock-unit is known before and after metamorphism, but the exact composition during the process is unknown. Using an end-point model means that, in general, we cannot deal with simultaneous interacting processes, that is, processes that simultaneously affect the same attribute of the same object.[4]

Since most occurrences of geologic processes are non-interacting (although they may be simultaneous), the end-point model has proven sufficient in solving most geologic interpretation problems. The end-point model is also appropriate for two reasons. First, there are many cases where we do not know what occurs during a complex geologic process (as in metamorphism, where the composition of a rock-unit during the process is

---

[4] However, we can deal with simultaneous, non-interacting processes.

not well understood). Hence, in many cases the end-point model is the best that we can do. Second, even in cases where we have a fairly accurate model of a process (as in uplift), representing it in more detail (see, for example, the representation of processes in [Forbus, 1982]) would lead to a situation that was computationally infeasible for our problems.

5.2 Process Representation for Modifying Histories

Figure 6 presents a description of the deposition process, represented in a form useful for reasoning about changes to histories. This style of representation explicitly represents which objects and attributes are affected by the process. We call this a *causal* description of the process.

1. The `INTERVAL` field describes the temporal interval during which the process is active. A temporal interval I is simply an interval of time represented by its end points I_{start} and I_{end}.

2. `PRECONDITIONS` is a set of statements which must be true in order for the process to occur.

3. `PARAMETERS` is a list of parameters which indicate the magnitude of the effects of the process. The imaginer must determine values for these quantities in order to simulate the process.

4. `AFFECTED` is a list of the objects which exist at the time the process began and are changed in some way by the process.

5. `CREATED` is a list of the objects which are created by the process.

6. The `EFFECTS` field is a set of statements that describe how the process changes the various attributes of the affected and created objects.

Fig. 6. Description of the Deposition Process

```
DEPOSITION
  INTERVAL       I : temporal-interval
  PRECONDITIONS  {(<  SURFACE·bottom·height@I_start  SEA-LEVEL)}
  PARAMETERS     DLEVEL : positive-real, DCOMPOSITION : sedimentary-rock
  AFFECTED       SURFACE
  CREATED        A : sedimentary, BA : boundary
  EFFECTS        {(change = A·thickness DLEVEL I DEPOSITION)
                  (change = A·orientation 0.0 I DEPOSITION)
                  (change = BA·side-1 {A} I DEPOSITION)
                  (change = BA·side-2 C I DEPOSITION)
                  (change = A·composition DCOMPOSITION I DEPOSITION)
                  (change = A·top (dfn DLEVEL SURFACE@I_start) I DEPOSITION)
                  (change = A·bottom SURFACE·bottom@I_start I DEPOSITION)
                  (change = SURFACE·bottom (dfn DLEVEL SURFACE@I_start)
                          I DEPOSITION)}
RELATIONS        {(= SURFACE·bottom·height@I_end (+ DLEVEL SURFACE·bottom·height@I_start))
                  (equiv A·orientation A·bedding-plane·y-angle I_end)
                  (< A·top·height@I_end SEA-LEVEL)
                  (= C {r : rock-unit
                          (exists r I_start) ->
                             (and (< r·bottom·height@I_start
                                     (+ DLEVEL SURFACE·bottom·height@I_start))
                                  (on-surface r I_start))))})
                  (equiv A·orientation BA·orientation I_end)}
```

7. RELATIONS is a set of assertions that are constrained to hold as a result of the occurrence of the process.[5]

For purposes of reasoning about change, the field of primary interest here is the list of EFFECTS. The general form is

`(CHANGE <TYPE> <ATTRIBUTE> <CHANGE> <INTERVAL> <CAUSE>).`

ATTRIBUTE is an expression describing the attribute changed by the process. INTERVAL is

---

[5] In Figure 6, `(equiv A1 A2 T)` means that after time T, attributes A1 and A2 are equivalent, that is, their values at all points in time are identical.

when the change occurred and CAUSE is the process that causes the change. TYPE and CHANGE jointly describe how the old and new values of the attribute are related. If TYPE is " = ", then the value after the process occurs equals CHANGE. For example, the form

`(CHANGE = A.thickness DLEVEL I DEPOSITION)`

describes the fact that after the deposition process, the thickness of the created sedimentary deposit equals the value of the parameter DLEVEL (i.e., A.thickness@I_{end} = DLEVEL). TYPE can also be an arithmetic operator (+, -, *, /), in which case the new value is found by applying the operator to the value of the attribute at the start of the process and the CHANGE. For example, an effect of the uplift process can be described by

`(CHANGE + A.height UPLIFT-AMOUNT I UPLIFT),`

which indicates that the height of rock-unit A after the uplift equals its height before the uplift plus the amount of the uplift (i.e., A.height@I_{end} = A.height@I_{start} + UPLIFT-AMOUNT). Finally TYPE can be "function" in which case the CHANGE is a function to be applied to the old value.[6]

We have implemented a program that instantiates a process at a particular point in time by making changes to history objects. The input to the program is a "causal" description of a process, of the sort shown in Figure 6, along with some additional information which specifies values for some of the expressions in the process description. For example,O we might specify that "DLEVEL = 10 meters", and "BA.side-2@I_{end} = {BEDROCK}" (i.e., "bedrock" lies on one side of the newly created depositional boundary).

---

[6] The type "function" is the most basic type; all other types can be defined in terms of it. For example, the " + " type with change Q is equivalent to the "function" type with change (LAMBDA (X) (+ X Q)).

To instantiate a process, the system carries out four steps. First, it checks that the preconditions hold. Second, it creates a representation for each member of the list of "created" objects. Third, it modifies the attributes of the affected and created objects, according to the `CHANGE` statements in the `EFFECTS` field, by inserting a dynamic interval into the appropriate place in the attribute's time-line. This is accomplished by splitting a quiescent interval into two pieces and inserting the dynamic interval in between. Fourth, the program asserts that all of the statements in the `RELATIONS` field hold.

For example, to instantiate the deposition process in Figure 6, the system carries out the following:

1. It determines that the bottom of the surface of the Earth is currently below sea-level.

2. It creates the new rock-unit **A** (the sedimentary deposit), and the new boundary **BA** (the boundary between **A** and whatever it was deposited upon).

3. It updates the appropriate time-lines for all the `CHANGE` statements. For example, it updates the (newly-created) time-line corresponding to the thickness of **A** by inserting a dynamic interval from I_{start} to I_{end}. Prior to time I_{start} the thickness is 0, between I_{start} and I_{end} the thickness is defined to be *unknown* and after I_{end} the thickness is "`DLEVEL`".

4. It asserts that all of the `RELATIONS` shown in Figure 6 now hold.

5.3 Process Representation for Modifying Diagrams

The process descriptions used with the diagram representation are simply end-point style algorithms for manipulating the diagrams. That is, processes are described in terms of the steps that need to be done in order to simulate the effects of the process in the diagram. We call these *operational* descriptions of processes. For example, the representation of deposition is shown in Figure 7, where the process is described in terms of drawing a line in a particular way. Figure 8 shows the effects of running that algorithm.

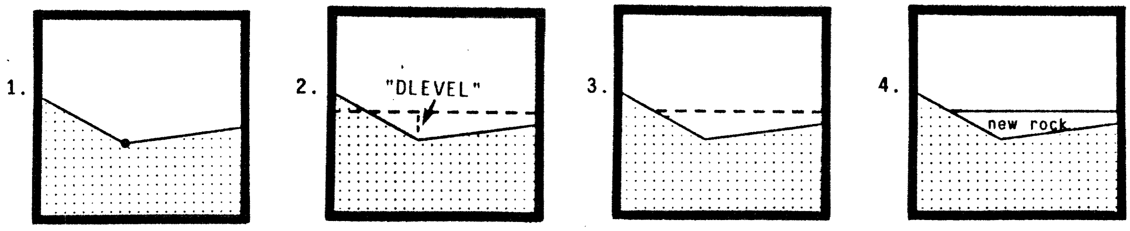
Note that although the diagrams themselves make no reference to geology, the diagram manipulation algorithms not only need to reference geometric properties of objects in the diagram, but also need to determine correspondences between diagram (geometric) and history (geologic) objects. For example, in Figure 7 a geometric property is "the lowest

---

Fig. 7. An Algorithm for Simulating Deposition in a Diagram

1. Find the lowest end-point of all the edges that represent the surface of the Earth.
2. Draw a horizontal line "DLEVEL" above that.
3. Erase all parts of the line that cut across a face corresponding to a rock-unit.
4. All other newly created faces below the line are part of the newly created sedimentary rock unit.

Fig. 8. (Diagram numbers correspond to the steps in Figure 7)



point of all the edges" and a correspondence is "all the edges that represent the surface of the Earth".

While using both process representations involves simulation, note that modifying histories involves a *qualitative* simulation and modifying diagrams involves a *quantitative* simulation. That is, in order to modify diagrams, the process parameters must be assigned exact values. This is due to the metric nature of diagrams. For example, a point in a diagram must be placed in a specific coordinate location -- it cannot have a "fuzzy" position in the diagram. Thus, the system can do the qualitative simulation when given the sequence of events, but it needs to determine the process parameters before it can do the quantitative simulation.

Since we describe diagram modifications in algorithmic terms, this operational representation of processes is implemented simply as LISP functions. These functions access the diagrams directly through the wing-edge structure primitives, and indirectly access the history objects through the diagram-history interface (see Section 4.3). The program to perform the quantitative simulation, which produces a sequence of diagrams, simulates an event as follows:

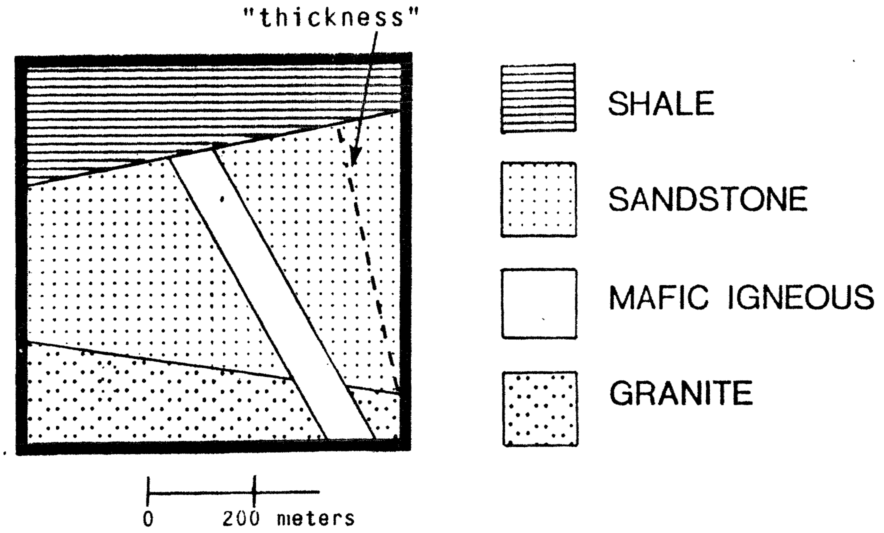
1. It copies the current diagram.
2. It determines the necessary numeric process parameters (see the next section).
3. It runs the LISP function representing the geologic process, modifying the copied diagram to reflect the effects of the process.

6. IMAGINING -- AN EXAMPLE

In this section, we present an example of the imagining process, showing how the representations we have developed enable us to do imagining. The input to the imaginer is shown in Figure 9 -- a cross-section representing the current geologic region and a sequence of events (produced by the scenario matcher) hypothesized to have produced that region.

Fig. 9. A Geologic Interpretation Problem and Hypothesized Solution

9a.



9b.

1. Deposit Sandstone on Bedrock
2. Intrude Granite into Sandstone
3. Intrude Mafic-Igneous through Granite and Sandstone
4. Erode Sandstone and Mafic-Igneous
5. Tilt by 13°
6. Deposit Shale on Sandstone and Mafic-Igneous

The first step in doing the imagining is a qualitative simulation of each event in the sequence. Each event is simulated, as described in Section 5.2.2, by using the appropriate process representation to modify the histories, that is, by creating objects and inserting dynamic intervals into their attribute time-lines to represent the changes. This step produces sequences of changes to the attributes that enable us to reason about the cumulative effects of the changes. For example, after the qualitative simulation the time-line for the thickness of the sandstone would contain dynamic intervals due to the initial deposition (step 1, Figure 9b), the intrusion of granite (step 2) and the erosion (step 4). However, at this stage the actual value of the thickness at any point in time is not known, beyond the fact that it is positive.

In order to do the next step, the quantitative simulation, we need to determine numeric values for the parameters used in each event. As an example, we consider how to determine the parameter `DLEVEL`, the amount of deposition (see Figure 6), for the deposition of the sandstone in step 1.

Parameter determination requires two steps. First, we measure the value in the goal diagram; second, we correct for the changes that have occurred to the parameter over time. For example, the system knows that the thickness of the sandstone (a sedimentary formation) corresponds to the maximum width of the corresponding diagram faces, measured perpendicular to its orientation. From the instantiation of the deposition process, the system knows that at the time of deposition the orientation was 0^{o} (see Figure 6). However, by examining the time-line of **sandstone.orientation** the system knows that there was a change in the orientation of 13^{o}, due to the tilt in step 5. Thus, the system measures the maximum width, perpendicular to 13^{o}, of the sandstone faces (Figure 9a), and determines that the thickness of the sandstone in the goal state is 500 meters.

Next, the system examines the thickness history and determines that the changes due to the granite intrusion (step 2) and erosion (step 4) must be accounted for. From the "layer-cake" model of geology that we use we know that the thickness of a formation being intruded into is decreased by the amount of the thickness of the intruding formation. Thus, to correct for the change in thickness due to step 2, the system needs to determine the thickness of the granite at the time of intrusion. It does this by measuring the width of the faces corresponding to the granite formation. Using the same reasoning as above, the system determines that it also must measure this width perpendicular to 13^{o}. The measured thickness is "greater than 200 meters" ("greater than" because some part of the granite formation continues outside the boundary of the diagram), so the current estimate for the thickness of the sandstone is "greater than 700 meters". Finally, the system knows that the thickness was decreased by the amount of erosion in step 4. The system tries to determine an exact value for the amount of erosion, but this information is not determinable from the goal diagram. The best that the system can do is to determine that the amount of erosion was greater than zero. Thus, the estimate of the initial amount of deposition is "greater than 700 meters". All of this numeric reasoning is done using the quantity lattice (see Section 4.4).

The imaginer can now quantitatively simulate the deposition process. The imaginer starts with a blank diagram to represent that, initially, just "bedrock" exists . Next, it chooses an exact value for `DLEVEL` within the allowable range of "greater than 700 meters" (we have chosen 800 meters) and uses the algorithm from Figure 7 to create the new diagram (Figure 10, diagram 1).

An exact value is needed because the simulation is done using diagrams which, as noted above, are metric in nature. For example, it is impossible to draw a horizontal line in the range "somewhat greater than 700 meters" because a line drawn in the diagram defines an *exact* equation for that line. So, to draw a line, exact parameters values must be chosen.

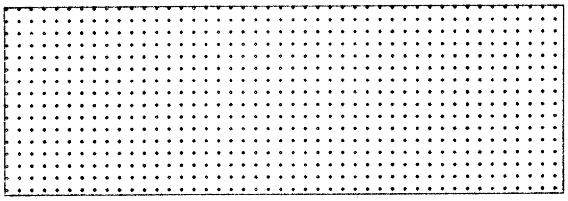
The question remains -- which value do we choose from within the range? Recall that the purpose of parameter determination is to choose values which approximate the actual geologic parameters used, in order to make tractable the task of matching the goal diagram and the final result of the simulation. Is the matching process affected by our choice of a specific value within the allowable range?

The answer is, no, it does not matter; choosing any arbitrary value within the range will eventually lead to the same final diagram. This can be seen by recalling why some values, such as the thickness of a rock-unit, are not known exactly: although the measurement from the goal diagram is exact, the magnitude of some subsequent change to that attribute is known only within a range. By choosing an exact value for the parameter when doing the simulation of one step, we also determine an exact value for the magnitude of the subsequent change. When the process which caused that change is later simulated, the magnitude of the change will already be determined exactly.

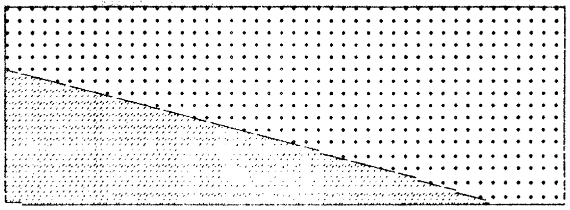
For example, by choosing `DLEVEL` to be 800 meters, we constrain the total change due to the granite intrusion and the erosion to be exactly 300 meters (since the measured thickness in the goal diagram was 500 meters). When the intrusion of granite is simulated (step 2, Figure 9b), the amount of granite is constrained to be between 200 and 300 meters. If we (arbitrarily) choose the amount of granite to be 250 meters, we automatically constrain the amount of erosion (step 4) to be 50 meters. After the erosion is simulated, the thickness of the sandstone in the final diagram will be 500 meters, the same as in the goal diagram. Thus, the *cumulative* effect on a measurable attribute will be the same as the value measured in the goal diagram, as long as we keep choosing values from within the allowable range and as long as those ranges are updated after each choice.

Fig. 10. Diagrammatic Simulation of Hypothesized Sequence

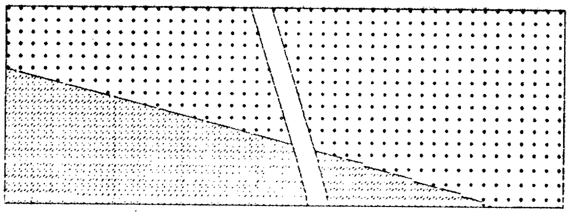
1. Deposit Sandstone



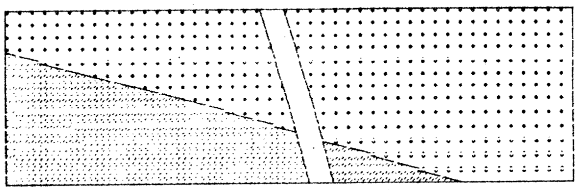
2. Intrude Granite



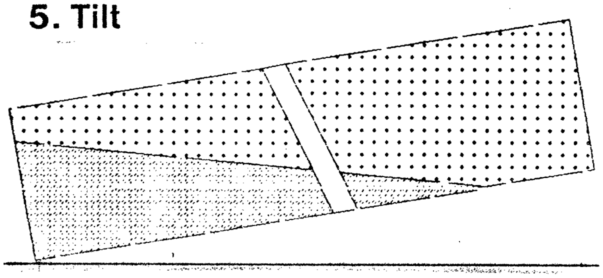
3. Intrude Mafic-Igneous



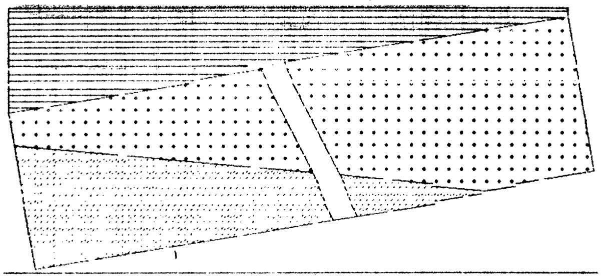
4. Erode



5. Tilt



6. Deposit Shale



This technique of determining numeric values for the process parameters and then simulating the process to produce a new diagram continues for each event in the sequence. The result of the simulation is shown in Figure 10, diagrams 1-6. Finally, the end result of the simulation (Figure 10, diagram 6) is compared with the goal diagram (Figure 9a) to check that they do in fact match. The system would then conclude that the sequence in Figure 9b is a valid hypothesis for describing how the geologic region was formed.

The problem of matching the diagrams has not yet been adequately explored in our current implementation. However, the basic algorithm has two steps. First, we check existence: each rock-unit or boundary in the goal diagram should have a corresponding entity in the simulated diagram. Second, we check adjacency: the rock-units adjacent to each rock-unit or boundary in the goal diagram should correspond to the rock-units adjacent to the corresponding entity in the simulated diagram.

7. LESSONS ABOUT REPRESENTATIONS AND PROBLEM SOLVING

While we have focused on one particular problem domain in this work, we have encountered several interesting issues in representation and problem solving whose relevance is clearly broader than this single domain. In particular, we have come to appreciate both the utility and difficulty of using multiple, specialized representations and have come to understand better the nature and role of simulation as a problem solving mechanism.

7.1 The Utility of Multiple, Specialized Representations

It became clear early on in this work that it would be difficult to enforce widespread uniformity of representation. Given the need to represent both objects and processes and the need to reason about them both spatially and temporally, it was difficult to propose a single representation well suited to all of those tasks. It is, for example, quite difficult to represent shape using a qualitative representation, but easy to do so in a quantitative representation like a diagram. Rather than trying to find a single representation that would meet all our needs, we adopted instead the approach presented in Section 4, using several carefully chosen representations, each specialized for solving a particular part of the task.

In this approach, we share the perspective developed from experience in systems like MACSYMA [Macsyma] and HEARSAY-II [Erman], where it became clear that specialized representations are often a worthwhile investment. The benefit accrues from the efficiency and ease of working with a representation tailored to the task at hand. MACSYMA, for example, uses several different representations of polynomials, each specialized to support efficient algorithms for doing particular arithmetic operations (multiplication, addition, exponentiation, etc.). In HEARSAY-II, different knowledge sources used different representations: the word recognizer used a network representation to deal with word pronunciation, while the word sequence recognizer used a bit matrix to support reasoning about word adjacency.

There is a cost, however, in using multiple representations. The cost stems either from translating between isomorphic representations (as MACSYMA does in shifting between its different polynomial representations), or from maintaining several distinct representations, each capturing some part of the problem (as in our separate representations for geology and geometry). In either case, experience has suggested that the tradeoff is worth making: it is often so expensive to work with the wrong representation that we are better off developing and using multiple, specialized representations.

One of the difficult issues in this approach is formulating principles for choosing and designing representations. It is easy to say that we will allow ourselves the luxury of multiple representations; it is somewhat more difficult to make sure that the ones we develop are both appropriate and necessary to our task. In reviewing our work, we have found four emerging principles useful:

1. Keep the different representations clearly separated, with a sharp interface at the intersections.

2. Representations should be defined and chosen operationally, that is, in the context of a particular task and usage.

3. Representations should be chosen to provide compactness and ease of reference.

4. Implementation of a representation should take account of the architecture of the underlying machine.

7.1.1 Keep the different representations clearly separated

This is simply the traditional call for modularity, directed here at representations. In our work, it is illustrated by the sharp boundary between the quantitative and qualitative representations (see Figure 1 and Section 4.3). Its utility lies, as usual, in simplifying the work needed on either side of the boundary. The quantitative representations deal solely with vertices, lines, and faces, while the qualitative representations keep track of things like rock composition and provides an interpretation for elements of the diagram.

7.1.2 Choose representations in the context of a particular task.

We cannot evaluate the appropriateness of a representation "in general", we can only evaluate it with respect to a particular problem. For example, we cannot "design a representation for physical space" without considering its use. For some problems, a simple listing of relations like *left-of* or *adjacent-to* is sufficient, but if we need to determine distances (as in our case), then this is clearly inadequate.

To define representations in this manner, we find it useful to consider two issues: What do we want to describe about the world, and What questions do we want to answer. For map interpretation, one of the things we need to describe about the world is the effects of processes like deposition and metamorphosis; one of the questions we might want to answer is, "Is rock-unit S1 above S2?"

Consider representing the effects of deposition and the effects of metamorphosis. As illustrated in Section 5, it is quite straightforward to conceive of deposition in terms of its effects as a modification to a diagram (Figure 7), but it is quite awkward to represent those same spatial changes in a qualitative representation (Figure 6). Metamorphosis, on the other hand, is easily described using histories (by indicating the relationship between rock composition before and after the process), but it is difficult to imagine how to represent it spatially.[7] Asking what we wanted to describe thus made it clear that we needed to represent and reason about space (leading to the use of diagrams) and time (leading to the use of histories).

Examining what questions we want our system to answer helps to further specify the representation. For example, consider answering the question noted earlier: "Is rock-unit S1 above S2?"

If we had only the assertional-style representation used in histories, then we might be forced to answer it by finding a sequence of relations of the form "Si is above Sj" and by using a transitivity rule to infer the answer. The metric character of the diagram permits us instead to measure the location of S1 and S2 and compute the answer directly.

7.1.3 Representations should provide compactness and ease of reference

Having established the need to represent things like spatial relations and spatial changes, we were naturally led to choosing a diagram (i.e., Euclidean geometry) as a representation. It is useful to ask what makes a diagram a "good" representation for this task.[8]

We believe that two important characteristics are the *compactness* and the *ease of reference* of a representation. A diagram is a compact representation for spatial relations because it encodes all of them with relatively few symbols. For example, from a single fact about each object -- its location -- along with the definition of each relation, we can easily

---

[7] Note that although rock composition is indicated in the diagram by means of textures (see Figure 2), the textures are not *used* as spatial representations, rather they are used as *symbols*, simply indicating labels for the regions. We perform no metric operations on the textures and the names of the rock compositions could easily have been used instead. This is another instance of our claim that representations are best defined and understood by their use.

[8] Despite how obvious a choice it seems to be, it is not the only possibility. It is, for example, quite possible (though not necessarily desirable) to represent spatial relations using a set of assertions of the form "Si is above Sj".

determine all possible spatial relations between two objects. This is considerably more compact than explicitly listing all independent relations between all pairs of objects.

Compactness also suggests that the size of the description be proportional to the complexity of the situation being represented. For example, a bit array is not a compact representation of a diagram because the description of a blank diagram is as large as the description of one that is arbitrarily complex.

Where compactness deals with the density of encoding, ease of reference refers to how easy it is to retrieve desired information from the encoding. As noted above, a question like "Is S1 above S2?" is easily answered using the metric properties of the diagram, while it would be considerably more difficult to get the answer via a string of transitivity relations.

Note that compactness and ease of reference are often at odds with one another. That is, in order to represent information compactly, we often need to encode it in ways that make it more difficult to reference. One of the major utilities in using a diagram is that spatial relationships can be both compactly represented and easily retrieved.

7.1.4 Implementation should take account of the machine architecture[9]

Consider a problem from the blocks world: given two blocks moving on specified trajectories, determine if they will come in contact (for simplicity, consider only two dimensions). If we are using a machine that happens to be good at arithmetic (as most computers are), it makes sense to represent blocks by their end-points and determine collisions by doing the relevant geometry. But imagine a machine composed of millions of very small processors connected in a grid, processors with little or no arithmetic capability, but very fast at marker propagation and very fast at exchanging information with their

---

[9] As we explore further in [Simmons], our distinction between representation design (Section 7.1.2) and implementation is similar in spirit to the guidelines suggested in [Marr].

neighbors. In that case it would be perfectly reasonable to represent blocks by appropriately shaped bit arrays. Movement would be simulated by shifting and rotation operations on the arrays and questions about collisions would be answered by asking whether any processor receives bits from two different arrays.[10] Thus having first established what we want to represent, the implementation can take strong advantage of the properties of the machine in use.

7.2 The Use of Simulation in Problem Solving

Simulation plays an important part in our approach to geologic interpretation, leading us to ask why and when in general it is useful as a problem solving tool. The relevant distinction appears to be between simulation, which involves *invoking* operators, and a different problem solving style which involves *reasoning about* the operators.

To illustrate, recall a standard problem: given a checkerboard with two opposite corners removed and dominoes the size of two squares of the board, the task is to cover the board exactly (i.e., with a single layer of dominoes and none extending over the edge of the board). Either find such a covering or show that none can exist.

Simulation would answer the question by invoking operators (placing dominoes on the board), attempting to find a covering. In order to infer that no covering is possible, all coverings must be tried. This clearly requires a lot of computation.

A different approach to the problem involves reasoning about the character of the operators. The crucial observation is that no matter how we place a domino, it covers one black and one red square; i.e., it always reduces the number of squares of each color by one. Since the mutilated board has 32 black squares and 30 red squares, no matter how we

---

[10] [Funt] describes work oriented along these lines.

lay down the dominoes we will arrive at a situation with 2 red squares left over.

This style of reasoning seems to have several names. Newell calls it reasoning about *hereditary* properties [Newell, 1965]; Simon extends this to *transmissible* properties [Simon]; the style has long been used in physics in the form of conservation principles. Whatever we call it, the important characteristic here is that we solve the problem by reasoning about operators rather than invoking them. That is, we find some property of the operator (and perhaps the problem space) and construct an argument based on the persistence of that property over all problem states.

When we have the knowledge required to support it, this approach is clearly more powerful than simulation, relying as it does on reasoning about the operators and the space rather than on searching the space. Simulation can be useful where knowledge about the operators is lacking, or where the analysis is too complex to afford a computationally feasible answer. In our geologic interpretation problem, for example, the net effect of a hypothesized sequence is determined by imagining ("running") the sequence, since we have no more sophisticated theory of geology akin to the observation about domino placement used above.

In slightly more abstract terms, simulation is a particular kind of movement through a state-space: the order of states encountered corresponds to the temporal order of events in the world being simulated. Reasoning about the operators and space, by contrast, provides a way of jumping from the initial to the final state (or shows that the transition is impossible), without going through all the intermediates. Considered in these terms, we can say that simulation is useful where our knowledge of the world is limited to information about how to get from one state to the next.

8. RELATED WORK

This paper has explored the use of imagining, a simulation technique, as a problem solving tool and discussed the representations used to support the task. In this section, we examine how other work has used simulation in problem solving and discuss representations designed to support tasks similar to our own.

8.1 The Use of Simulation in Problem Solving

Much of the work on "Naive Physics" has influenced our ideas on using simulation as a problem solving tool (particularly [deKleer, 1975] and [Forbus, 1981]) and has influenced our approach to representing change (particularly [Forbus, 1982] and [Hayes]). Simulation has often been used in problem solving (e.g. [deKleer, 1975], [Fikes], [Forbus, 1981], [Funt], [Hendrix], [Rieger]). One important characteristic of simulation is that it constructs all the intermediate states along the solution path. Many simulation techniques, however, do not keep a record of all the changes, but instead erase old values as the simulation progresses (see, for example, [Fikes], [Funt], [Rieger]). The results of these simulations cannot be used to reason about the temporal extent of the changes. Thus, they have been used for tasks where it is sufficient for the simulation to tell us *what* happened, rather than *how* it happened. However, for parameter determination and for generating plans the need to reason about the character of the changes necessitates maintaining the intermediate states (as done in [deKleer, 1975], [Forbus, 1981], and our work).

8.2 Representations to Support Imagining

8.2.1 Histories

As noted above, our task requires that we maintain the intermediate states produced by the simulation. The history representation is well suited to this task, since it enables us to keep track of and reason about sequences of changes to attributes. This type of reasoning is useful in many tasks and has resulted in the use of representations similar to our histories (e.g., [Forbus, 1982], [Hayes], [Shapiro], [Tsotsos]). These representations all maintain the sequences of values resulting from changes and have operators, similar to our @-operator, for selecting values at points in time. They differ from histories, however, in that the *cause* of a change is not represented, nor is any knowledge about the relationships between the values before and after the change occurs.

Many of our ideas on temporal selection and on representing the creation and destruction of objects were developed from work in tense logic (see, for example, [McArthur]). In particular, temporal logics have formalized the notion of change to the attributes of an object. However, these logics are all focused on relations between objects. This creates two difficulties. First, although it is easy, for example, to reason about all red objects at time **t0**, it is harder to reason about the change in color to a particular object over time. Second, it is difficult for these logics to use the assumption that values remain constant unless indicated otherwise. Since both of these are necessary for our task, our history representation employs temporal logic concepts but places them in an object-oriented setting.

8.2.2 Diagrams

The use of diagrams to do simulation is an important aspect of our approach to imagining. Using diagrams as an aid in problem solving has a long history in AI (e.g. [Gelernter]), and several efforts have investigated doing simulation using diagrams (e.g. [Forbus, 1981], [Funt]). In many cases, the rationale for using diagrams is similar to ours -- the task domain is largely spatial in nature and diagrams facilitate reasoning about and manipulating spatial properties of objects.

Although there is agreement as to the utility of diagrams, the complexities of spatial representation have led to the development of many different representations (e.g. [Baumgart], [Forbus, 1981], [Funt], [Gelernter], [Hunter]). Most of the differences involve tradeoffs between shape description and ease of use. For example, in domains where arbitrary shapes must be represented, a representation like [Hunter] might be preferred over [Baumgart], which uses only straight lines. However, the more complex representations often make simple manipulations, such as drawing a line, difficult to perform, while the primitive operations in our wing-edge structure can do them quite efficiently.

Another consideration in choosing a diagram representation is the vocabulary of primitives. For example, in [Hunter] the primitive is a face, in [Funt] it is a pixel of an array and in [Gelernter] it is lines and points. Our approach to representing diagrams is similar to [Forbus, 1981] and [Gelernter] in that the primitive objects in the diagram vocabulary closely correspond to the primitive objects in the task domain.

8.2.3 Processes

All systems which perform simulation must represent processes or actions. Our imaginer uses two types of process representations -- one to modify diagrams, which describes how to simulate a process, and one to modify histories, which explicitly describes the effects of a process.

Many AI systems specify actions operationally, in terms of the steps to perform (e.g., [deKleer, 1975], [Funt], [Winograd]). The major advantage of this style of representation is its ease of use -- it is often easy to describe a process in terms of the actions to be taken. In particular, we do not have to worry about describing *why* the steps accomplish the task. The major disadvantage is that the knowledge can only be used in one way -- typically for performing simulation.

For many tasks, the need to reason about changes to objects leads to a representation that explicitly encodes the effects caused by processes. Such "causal models" have been a focus of considerable attention (e.g. [deKleer, 1982], [Fikes], [Forbus, 1982], [Hayes], [Hendrix], [McDermott], [Patil], [Rieger]). All of these efforts explicitly represent the changes that result from actions. However, they differ in whether processes are represented explicitly. In some representations (e.g., [Hayes], [Patil], [Rieger]), processes are represented implicitly by the causal links between objects. Our representation follows another course (see also [Fikes], [Forbus, 1982], [Hendrix]) by packaging together the preconditions, parameters, affected objects, etc., which define the process. An advantage of this approach is that it facilitates determining such things as how a particular event affects the world or what parts of the world are affected. Without an explicit representation of processes, the only way this can be determined is through simulation.

Our process representation for modifying histories are most similar to those described in [Fikes] and [Forbus, 1982], in that processes are explicitly represented, as are their effects (i.e., the representations make explicit which objects are affected and created and what changes occur to the objects). The major difference from [Fikes] is that we describe the effects of processes in terms of both the current values and the magnitudes of the changes. Thus, we can reason about the cumulative effects of change over time.

The major difference between our process representation and that described in [Forbus, 1982] or [Hendrix], is our use of "end-point" or kinematic models, rather than dynamic models. In fact, most of the research in modeling change uses kinematic models, often for reasons similar to ours: kinematic models are sufficient to solve the problem at hand. While dynamic models provide a more detailed description of the diagram, they are extremely difficult to specify and may add unnecessary complexity to the task.

9. CONCLUSIONS

The research presented in this paper has been motivated by a desire to use the technique of *imagining* as part of solving the geologic interpretation problem. Imagining simulates a sequence of events by modifying both history and diagram objects. We have found the explicit representation of changes to objects useful in doing imagining. We reason about these changes in determining numeric parameter values from a goal diagram in order to do the quantitative simulation on diagrams.

We have developed two representations of objects to facilitate reasoning about such changes. The first representation, called *histories*, is a frame-like representation but with time-lines as the values of attributes. We designed the @ operator, which ranges over history objects and attribute time-lines, to select the value of an object or attribute at a particular point in time. We also have implemented an efficient representation for histories to facilitate using the @ operator and adding changes to an object. The second representation, based on the notion of *diagram*, incorporates a 2-dimensional diagram system that facilitates spatial reasoning, both in accessing and modifying spatial properties of objects. In addition, we have developed an interval-based quantity lattice which allows us to do arithmetic on and to reason about the relationships between numeric quantities whose actual values may be known only within some real-valued range.

Finally, we have presented two representations of processes. Each representation is geared to one of the two object representations. These process representations facilitate changing history and diagram objects in order to simulate and reason about the effects of geologic processes on the real world.

ACKNOWLEDGMENTS

We would like to thank Ken Forbus and Chuck Rich for their valuable suggestions and comments. The suggestions of Patrick Winston, Peter Szolovits and Reid Smith aided in the presentation of this paper.

REFERENCES

Baumgart, Bruce [1974] - "Geometric Modelling for Computer Vision", Stanford AIM 249.

Davis, Randall [1982] - "Expert Systems: where are we and where do we go from here", *AAAI Magazine*, Summer 1982.

DeKleer, Johan [1975] - "Qualitative and Quantitative Knowledge in Classical Mechanics", MIT AI-TR-352.

DeKleer, Johan; Brown, John Seely [1982] - "Assumptions and Ambiguities in Mechanistic Mental Models", CIS-9, Xerox PARC.

Erman, Lee D; Hayes-Roth, Frederick; Lesser, Victor R; Reddy, Raj D [1980] - "The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty", *Computing Surveys*, vol. 12, no. 2.

Fikes, Richard E; Nilsson, Nils J [1971] - "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving", *Artificial Intelligence*, vol. 2, pp. 189-208.

Forbus, Kenneth D [1981] - "A Study of Qualitative and Geometric Knowledge in Reasoning about Motion", MIT AI-TR-615.

Forbus, Kenneth D [1982] - "Qualitative Process Theory", MIT-AIM-664.

Friedman, Gerald; Sanders, John [1978] - *Principles of Sedimentology*, John Wiley and Sons, p. 197.

Funt, Brian V [1980] - "Problem-Solving with Diagrammatic Representations", *Artificial Intelligence*, vol. 13.

Gelernter, H [1963] - "Realization of a Geometry Theorem Proving Machine", in *Computers and Thought*, eds. Feigenbaum and Feldman.

Hayes, Patrick J [1979] - "The Naive Physics Manifesto", in *Expert Systems in the Micro-Electronic Age*, ed. Michie, Edinburgh University Press.

Hendrix, Gary [1973] - "Modeling Simultaneous Actions and Continuous Processes", *Artificial Intelligence*, vol. 4, p. 145.

Hunter, Gregory M.; Steiglitz, Kenneth [1979] - "Operations on Images Using Quad Trees", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2.

MACSYMA Reference Manual [1977], The Mathlab Group, Laboratory for Computer Science, MIT.

Marr, David [1982] - *Vision*, Freeman and Co, chapter 1.

McArthur, Robert P [1976] - *Tense Logic*, D.Reidel Publishing Co.

McDermott, Drew [1982] - "A Temporal Logic for Reasoning About Processes and Plans", *Cognitive Science*, vol. 6, pp. 101-155.

Minsky, Marvin [1975] - "A Framework for Representing Knowledge", in *The Psychology of Human Vision*, ed. Winston, McGraw-Hill, p. 211

Newell, Allen and Simon, H.A. [1963] - "GPS, A Program that Simulates Human Thought", in *Computers and Thought*, eds. Feigenbaum and Feldman.

Newell, Allen [1965] - "Limitations of the Current Stock of Ideas about Problem Solving", in *Electronic Information Handling*, eds. Kent and Taube, Spartan, Washington DC.

Patil, Ramesh; Szolovits, Peter; Schwartz, William [1981] - "Causal Understanding of Patient Illness in Medical Diagnosis", *IJCAI 7*, p. 893.

Rieger, Chuck; Grinberg, Milt [1979] - "The Declarative Representation and Procedural Simulation of Causality in Physical Mechanisms", *IJCAI 6*, p. 250.

Shapiro, Daniel G [1980] - "A Proposal for Sniffer: a System that Understands Bugs", MIT-WP-202.

Shelton, John [1966] - *Geology Illustrated*, Freeman and Co, chapter 21.

Simmons, Reid - "Representing and Reasoning About Change in Geologic Interpretation", Master's Thesis, MIT, May 1983.

Simon, H.A. [1968] - "On Reasoning about Actions" in *Representation and Meaning*, eds. Simon and Siklossy, Prentice-Hall, Englewood Cliffs, NJ.

Tsotsos, John [1980] - "A Framework for Visual Motion Understanding", CSRG-114, Department of Computer Science, University of Toronto.

Winograd, Terry [1972] - *Understanding Natural Language*, Academic Press.