Technical Report 749

# Representing and Reasoning About Change in Geologic Interpretation

Reid Gordon Simmons

MIT Artificial Intelligence Laboratory

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER<br>AI-TR-749 | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>Representing and Reasoning About Change in Geologic Interpretation | | 5. TYPE OF REPORT & PERIOD COVERED<br>Technical report<br>6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>Reid Gordon Simmons | | 8. CONTRACT OR GRANT NUMBER(s)<br>N00014-80-C-0505 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Artificial Intelligence Laboratory<br>545 Technology Square<br>Cambridge, Massachusetts 02139 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Advanced Research Projects Agency<br>1400 Wilson Blvd<br>Arlington, Virginia 22209 | | 12. REPORT DATE<br>December 1983<br>13. NUMBER OF PAGES<br>131 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)<br>Office of Naval Research<br>Information Systems<br>Arlington, Virginia 22217 | | 15. SECURITY CLASS. (of this report)<br>UNCLASSIFIED<br>15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Qualitative Simulation
Quantitative Simulation
Multiple Representation
Representing Processes
Spatial Reasoning
Temporal Reasoning
Numeric Reasoning
Geologic Interpretation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Geologic interpretation is the task of inferring a sequence of events to explain how a given geologic region could have been formed. This report describes the design and implementation of one part of a geologic interpretation problem-solver -- a system which uses a simulation technique called imagining to check the validity of a candidate sequence of events.

Imagining uses a combination of qualitative and quantitative simulations to reason about the changes which occurred to the geologic region. The spatial

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED (continued over)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20 cont.

changes which occur are simulated by constructing a sequence of diagrams. This quantitative simulation needs numeric parameters which are determined by using the qualitative simulation to establish the cumulative changes to an object and by using a description of the current geologic region to make quantitative measurements.

The diversity of reasoning skills used in imagining has necessitated the development of multiple representations, each specialized for a different task. Representations to facilitate doing temporal, spatial and numeric reasoning are described in detail. We have also found it useful to explicitly represent processes. Both the qualitative and quantitative simulations use a discrete "layer cake" model of geologic processes, but each uses a separate representation, specialized to support the type of simulation. These multiple representations have enabled us to develop a powerful, yet modular, system for reasoning about change.

# Representing and Reasoning About Change in Geologic Interpretation

by

Reid Gordon Simmons

Massachusetts Institute of Technology
December 1983

This report is a revised version of a thesis submitted to the Department of Electrical Engineering and Computer Science on May 13, 1983 in partial fulfillment of the requirement for the degree of Master of Science.

© Massachusetts Institute of Technology 1983

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-80-C-0505.

# Abstract

Geologic interpretation is the task of inferring a sequence of events to explain how a given geologic region could have been formed. This report describes the design and implementation of one part of a geologic interpretation problem solver -- a system which uses a simulation technique called *imagining* to check the validity of a candidate sequence of events.

Imagining uses a combination of qualitative and quantitative simulations to reason about the changes which occurred to the geologic region. The spatial changes which occur are simulated by constructing a sequence of diagrams. This quantitative simulation needs numeric parameters which are determined by using the qualitative simulation to establish the cumulative changes to an object and by using a description of the current geologic region to make quantitative measurements.

The diversity of reasoning skills used in imagining has necessitated the development of multiple representations, each specialized for a different task. Representations to facilitate doing temporal, spatial and numeric reasoning are described in detail. We have also found it useful to explicitly represent *processes*. Both the qualitative and quantitative simulations use a discrete "layer cake" model of geologic processes, but each uses a separate representation, specialized to support the type of simulation. These multiple representations have enabled us to develop a powerful, yet modular, system for reasoning about change.

## Dedicated

**In loving memory to my Father,
who taught me, through example,
to be a scientist
and a human being.**

## Acknowledgments

This thesis is a direct result of the support and guidance of many people. In particular, I would like to thank the following:
-- Randy Davis, my thesis supervisor, who always was available to listen to my ideas, complaints and musings, and was able to see them in perspective when I could not. He taught me how to be critical of my ideas and how to communicate them effectively, both in writing and orally.
-- Ken Forbus, Chuck Rich and Reid Smith, who were always eager to enter into intellectual debate, and from whom I learned a great deal about representing knowledge.
-- Roy Nurmi, who showed me how to do geologic interpretation, and was my geology mentor.
-- David McAllester, for developing and implementing RUP, the reasoning utility package used by the system, and Rod Brooks, for supplying the initial implementation of Baumgart's wing-edge structure.
-- The AI Lab, especially my officemates Boris Katz, Beth Levin, and Ken Yip, for providing such a fruitful research environment.
-- Jane Feinberg, whose love and friendship enabled me to keep my head above sea-level, and my feet on the ground.

To these and many others, my heartfelt thanks.

# CONTENTS


**1. OVERVIEW ..... 6**

1.1 Geologic Interpretation ..... 6
1.2 Imagining ..... 8
1.3 Multiple Representations ..... 8
1.4 Outline of the Report ..... 11

**2. GEOLOGIC INTERPRETATION ..... 12**

2.1 An Example of Geologic Interpretation ..... 12
2.2 Problem Solving Technique ..... 16
2.3 Geologic Vocabulary ..... 19

**3. IMAGINING ..... 22**

3.1 Simulation ..... 22
3.2 Why Use Two Simulations? ..... 25
3.3 The Imagining Technique ..... 29

**4. OBJECT REPRESENTATIONS ..... 45**

4.1 Qualitative Objects - Histories ..... 45
4.2 Quantitative Objects - Diagrams ..... 56
4.3 Quantity Lattice ..... 68

**5. PROCESS REPRESENTATIONS ..... 74**

5.1 Level of Representation ..... 74
5.2 Qualitative Processes ..... 76
5.3 Quantitative Processes ..... 84
5.4 Developing Qualitative Process Descriptions ..... 90

6. RELATED WORK ..... 92

6.1 The Use of Simulation in Problem Solving ..... 92
6.2 Representations to Support Simulation ..... 93

7. FUTURE WORK ..... 98

7.1 Improving the Imagining Technique ..... 98
7.2 Solving the Geologic Interpretation Problem ..... 99

8. CONCLUSIONS ..... 102

Appendix A. GEOLOGIC OBJECTS ..... 108

Appendix B. QUALITATIVE PROCESSES ..... 112

Appendix C. GEOLOGIC AXIOMS ..... 119

Appendix D. QUANTITATIVE PROCESSES ..... 122

Appendix E. EXAMPLES OF IMAGINING ..... 126

# 1. OVERVIEW

This report presents a technique which we call *imagining*, which is used to test the validity of candidate solutions for the geologic interpretation problem. Imagining uses a combination of qualitative and quantitative simulations to determine whether the candidate sequence of geologic events could have caused a given geologic region. The quantitative simulation is performed by constructing a sequence of diagrams to represent the spatial effects of the geologic events. We explore why simulation is a useful problem solving technique and, in particular, why the concept of imagining is useful in some domains.

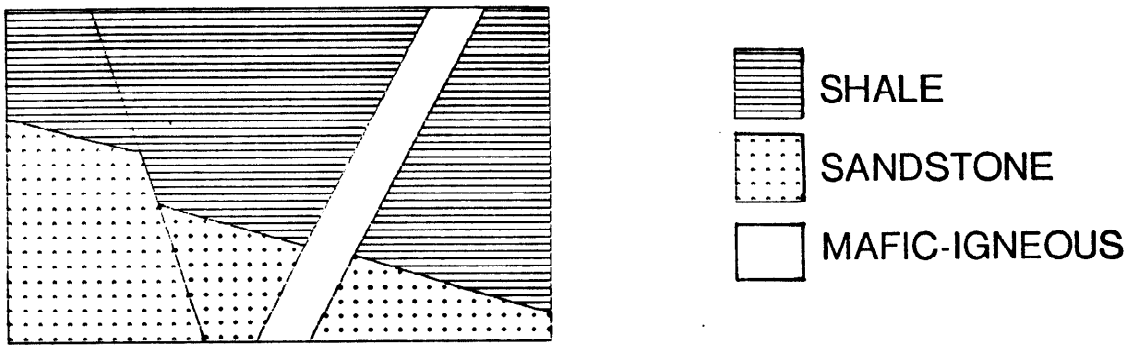
Since the imagining technique involves spatial, temporal and numeric reasoning, we have found it necessary to incorporate multiple, specialized representations into the system. The bulk of this report describes the representations chosen and discusses why they are appropriate. We have tried to use a "top-down" approach to choosing representations, since the criteria inherent in the task domain strongly constrains the types of representations which will be adequate for the task.

## 1.1 Geologic Interpretation

Geologic interpretation is the task of inferring, from a description of a region, the sequence of events which formed that region. In our case, the description of the region is a diagram representing a cross-section of the Earth, together with a legend identifying the rock types (see, for example, Figure Figure 1). Geologic interpretation differs from other interpretation tasks (e.g. [Buchanan], [Davis, 1981], [Nii]) in that it attempts to reconstruct the temporal sequence of events which occurred, rather than the static, physical structure of the domain.

Geologic interpretation is an important task for much of geology. It is used as the first step in many tasks to convert the "signal data" (the cross-section) from a spatial domain to the temporal domain of geologic processes. Geologic interpretation is typically taught to first year geology students in order to develop their understanding of temporal aspects of

Fig. 1. Input to Typical Geologic Interpretation Problem




---

geology -- how things form and how they interact. We use a simplified model, known as "layer cake" geology, in which the spatial effects of processes which occur laterally are largely ignored. For example, it is assumed that depositions occur horizontally, stacking up like the layers of a cake, and that erosions occur horizontally, slicing through the Earth like a knife.

We propose to solve the geologic interpretation problem using a three phase method based on "generate and test". *Scenario matching* would be used to generate candidate solution sequences by matching local patterns in the diagram to infer sequences of events. To test if a sequence could have formed the geologic region, we propose using *imagining*, a technique which performs both qualitative and quantitative simulations. Finally, if the imagining detects an invalid sequence, *gap filling* would be used to try to infer events which eliminate the inconsistencies found. This report concentrates on the use of imagining in problem solving; of the three phases, only the imaginer has been designed and implemented.

## 1.2 Imagining

The major aim of this research has been to explore the use of *imagining* to test the validity of a sequence of geologic events. Basically, imagining is a technique that simulates a sequence of events and matches the final result of the simulation against the goal state. If the match succeeds, we can conclude that the sequence is a valid explanation for the occurrence of the goal state. Imagining uses a combination of qualitative and quantitative simulations -- the qualitative simulation is used in augmenting the sequence of events with numeric parameter values so that a quantitative simulation can be performed. The result of the quantitative simulation is matched against the quantitative goal state. In our case, the quantitative simulation constructs diagrams to reflect the spatial effects of the sequence of geologic events.

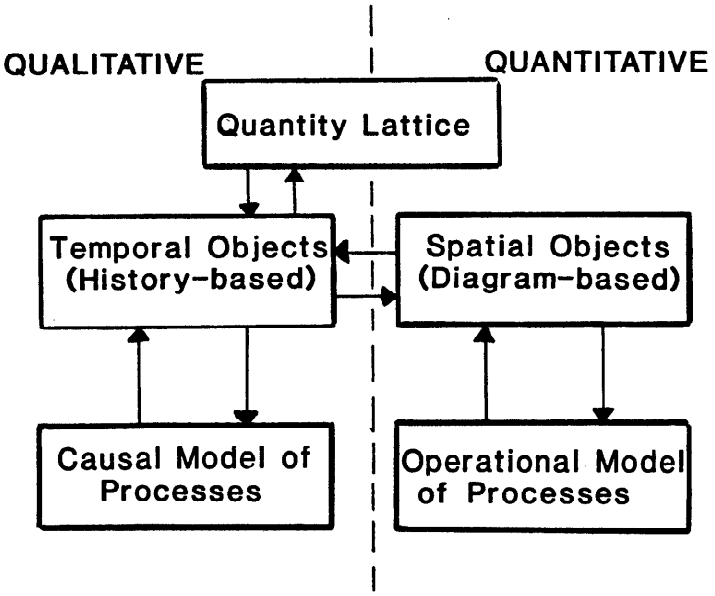
This report also examines the nature of simulation, contrasting it with other search techniques and discussing the circumstances under which it is a useful problem solving tool. We also examine why imagining must use both qualitative and quantitative simulations and try to characterize the domains in which imagining may prove useful.

## 1.3 Multiple Representations

A major focus of this report is the multiple representations used in doing imagining. Early on in the research it became clear that the diversity of reasoning skills needed to do the imagining was too great to insist on a uniform representation throughout the system. We have developed five representations to support the temporal, spatial and numeric reasoning skills needed by the imaginer (see also Figure 2) --

1. A qualitative temporal object representation based on *histories*;
2. A quantitative spatial representation based on *diagrams*;

Fig. 2. Representations for Doing Imagining




3. A qualitative causal model of processes;

4. A quantitative operational model of processes;

5. A *quantity lattice*, used for numeric reasoning, which contains both qualitative and quantitative elements.

Each of these representations consists of a set of primitive objects and inference mechanisms, specialized to support a specific imagining task or set of tasks. The qualitative representations were designed to support the qualitative simulation of the imaginer, which is performed in order to reason about the cumulative effects of changes. Since we need to determine when objects are created and destroyed and how their attributes change over time, we were led to describing objects using a representation based on *histories* [Hayes].[1] In particular, an attribute of an object is represented as a *time-line* of values. These criteria

1. Hayes' original formulation of *histories* was confined to describing the three-dimensional position of objects over time. We have expanded this notion (similar to [Forbus, 1982]) to describe the temporal changes to any attribute of an object.

have also led to describing qualitative processes in terms of their effects, in what we call a *causal model* of processes.

The quantitative representations were designed to support the quantitative simulation. As noted earlier, this simulation is done to construct a description of the geologic region to match against the goal state, which in our case is a diagram cross-section. Thus, it makes sense for the quantitative object representation to be based on diagrams. We have chosen to represent the quantitative processes in operational terms, that is in terms of how to simulate the geologic effects by modifying diagrams, to facilitate constructing process descriptions which are applicable over a wide range of inputs.

The quantity lattice was designed to support numeric reasoning for both the qualitative and quantitative simulations. Such reasoning includes inferring ordering relationships between terms and doing arithmetic on terms. Since the values of these terms may be only partially specified, we have chosen to include both qualitative and quantitative elements in the quantity lattice -- partial orderings and real-valued intervals.

Since each representation is designed for a specialized task, the internal structure and operations on a representation differ widely. As a result, we have had to be very careful to maintain modularity throughout the system, limiting the interaction between representations to a simple, clearly defined interface. For instance, the qualitative/quantitative object interface consists of a one-to-one mapping between the primitive objects of each representation plus a simple mapping between spatial properties (for example, "above" in the diagram corresponds to "above" in the geologic world). This interface is particularly important as it enables us to make geologic sense out of a diagram, which is merely a collection of lines.

We believe that AI research will benefit from a thorough understanding of the nature of representation. This report attempts to provide a step in that direction by documenting the rationale behind our choice of representations. For each of the five representations used by the imaginer, we present the descriptive and inferential requirements of the task domain which led to our choice of representations.

### 1.4 Outline of the Report

Chapter 2 presents an example of geologic interpretation and discusses the modified generate and test method suggested for solving geologic interpretation problems. In Chapter 3, we examine the imagining technique in detail, discussing the nature of simulation and describing the four stages of imagining -- qualitative simulation, parameter determination, quantitative simulation and matching. Chapters 4 and 5 describe the object and process representations that are used by the imaginer. In Chapter 6, we examine our research in light of other related work, and in Chapter 7, we discuss some of the limits of the system and the possibilities for future work.

# 2. GEOLOGIC INTERPRETATION

The task of geologic interpretation is to reconstruct the geologic history of a region, in order to determine how the region was formed (for example, see [Shelton]). The input to the problem is a diagram that represents a vertical cross-section of a region along with a legend identifying each kind of rock formation (Figure 3a). A solution consists of a sequence of geologic events that plausibly explains how the region was formed. Figure 3c presents a solution to the problem of Figure 3a.

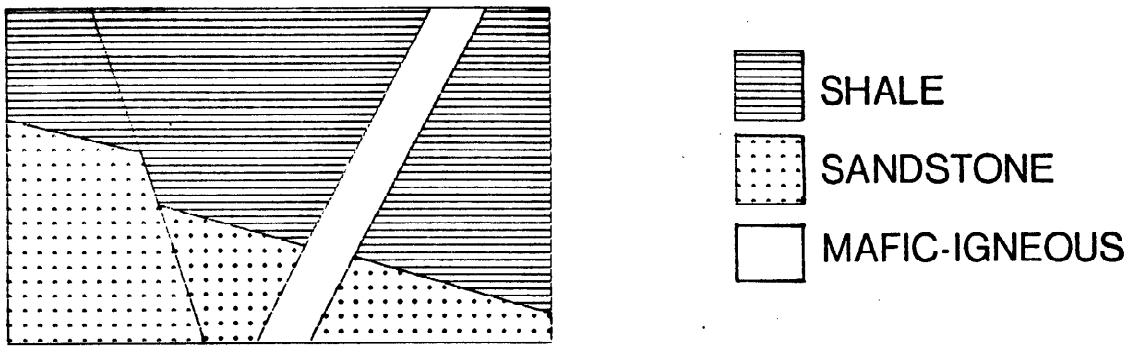
Geologic interpretation is similar to other signal interpretation problems, such as mass spectrograph analysis [Buchanan] or well-log interpretation [Davis, 1981]. These interpretation problems involve a transformation from a signal to a symbolic representation of the phenomena that gave rise to the signal. For example, a sequence of geologic events gives rise to a particular diagram cross-section. Many signal interpretation problems may be solved by "generate and test" [Newell, 1973] and we find that the basic idea applies to geologic interpretation as well. We believe that in solving the geologic interpretation problem, a geologist typically looks for local patterns of boundaries between rocks and *generates* interpretations from those patterns. He then *tests* the consistency of a candidate solution by "imagining", that is mentally simulating, what would happen if the events occurred and seeing if the result matches the diagram cross-section. If there are differences between the result of the simulation and the diagram cross-section, the sequence is *debugged* by using these differences to infer the existence of geologic events.

## 2.1 An Example of Geologic Interpretation

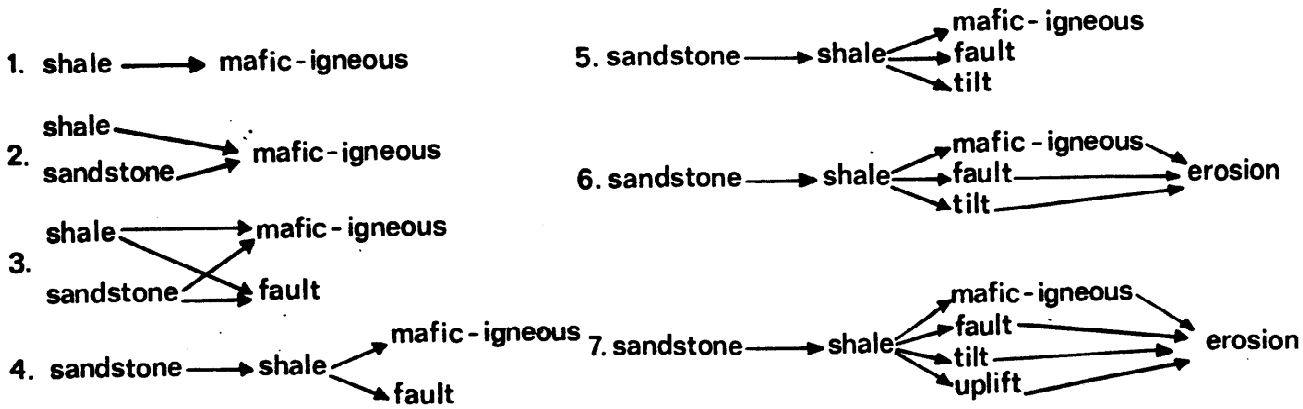
In trying to interpret Figure 3a, we might notice that the **mafic-igneous** crosses the **shale**. From this we would infer that the **mafic-igneous** intruded through (i.e., forced its way through) the **shale** and hence is younger (Figure 3b, step 1; the collection of partial orders shows our candidate sequence at each stage of development). The same reasoning would

Fig. 3. Geologic Interpretation Example

**a. Geologic Cross-Section and Legend**




**b. Partial Orders at Each Stage of the Problem**



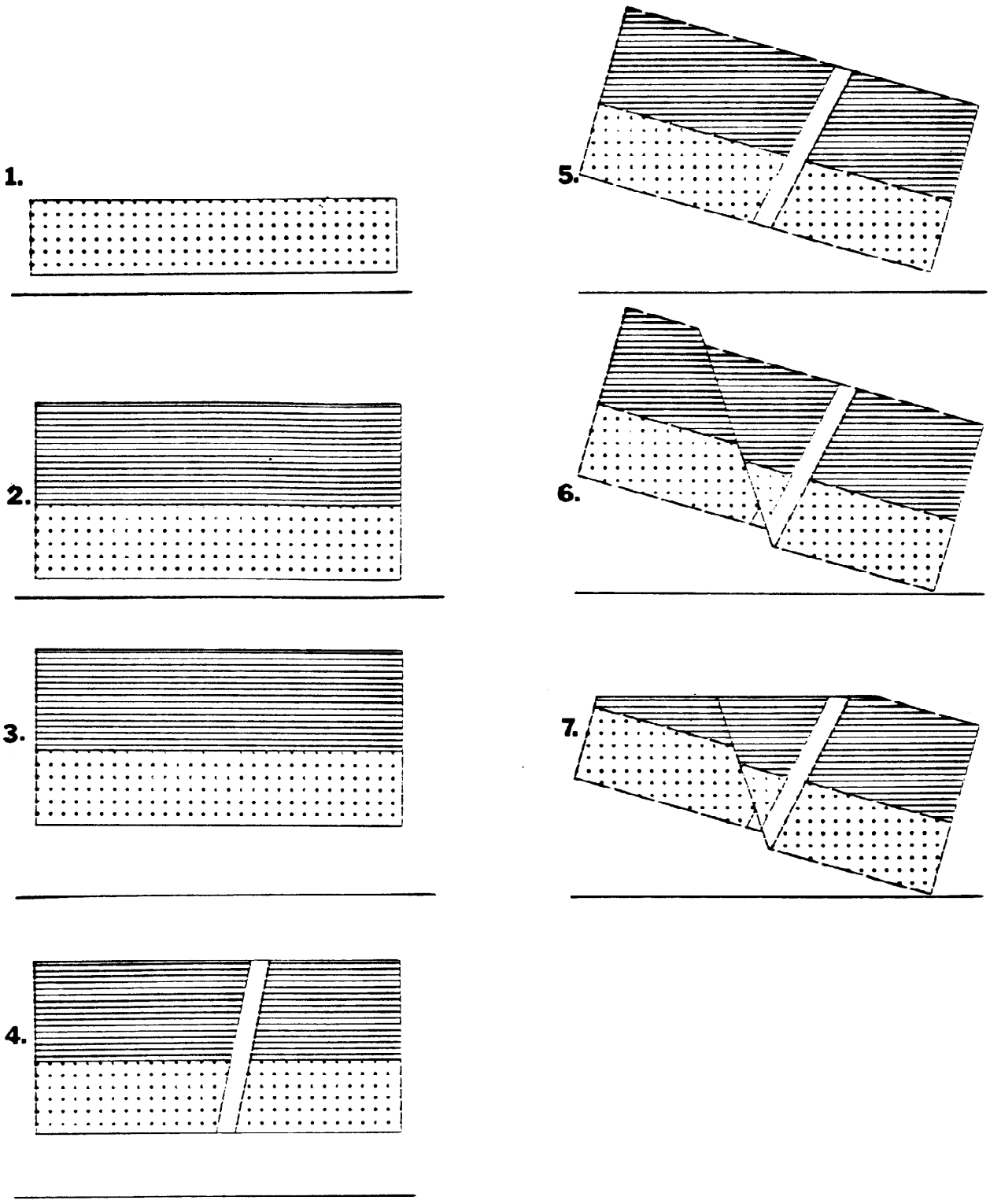

**c. Solution of Geologic Interpretation Problem**

1. Deposit **sandstone**
2. Deposit **shale**
3. Uplift
4. Intrude **mafic igneous** through **sandstone** and **shale**
5. Tilt
6. Fault
7. Erode **shale** and **mafic-igneous**

indicate that the **mafic-igneous** also intruded through the **sandstone** (Figure 3b, step 2). Similarly, by noting that the **fault** crosses the **shale** and the **sandstone**, we can infer that both formations are older than the **fault** (Figure 3b, step 3). Thus, the **shale** and the **sandstone** were both in place before the **mafic-igneous** intrusion or the **fault**. To determine the order in which the **sandstone** and the **shale** appeared, we could 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 **sandstone**; hence the **sandstone** is older (Figure 3b, step 4). Also, since we know that sedimentary formations (that is, the **shale** and the **sandstone**) are deposited horizontally, we can infer that each was tilted sometime after deposition (Figure 3b, step 5). Finally, the smooth, horizontal top surface indicates that erosion occurred (Figure 3b, step 6) after the tilt. However, we know that erosion occurs above sea-level, but the deposition of **shale** occurred under water. Therefore, we hypothesize that some uplift occurred after the deposition of shale but before the erosion. These inferences lead to the partially ordered sequence of Figure 3b, step 7 (note that we cannot infer the relative ages of the **fault** and the **mafic-igneous** just from the diagram).

Since it is easier to think about sequential events, and since the events which are unordered in Figure 3b-7 do not interact, we can linearize the candidate sequence without changing its validity (Figure 3c). We can now test this hypothesized solution by "imagining" what would happen if the events occurred. We accomplish this by drawing a sequence of diagrams to simulate the effects of each geologic event in the sequence (Figure 4, 1-7). Since the final result of the simulation (Figure 4-7) matches the diagram cross-section (Figure 3a) we can conclude that the hypothesized solution (Figure 3c) is one valid explanation for the origin of the region.

Fig. 4. Imagining Example -- Quantitative Simulation



## 2.2 Problem Solving Technique

The problem solving technique used in the example above consists of three phases -- generate, test, and debug. In the first phase, a technique we call *scenario matching* is used to generate candidate sequences of events to explain how the cross-section came into existence. In the second phase, a technique we call *imagining* is used to test if the candidates are correct. In the third phase, if a candidate sequence is not correct, it is debugged using a technique we call *gap filling*.

### 2.2.1 Scenario Matching

Scenario matching is a technique of inferring a sequence of events by reasoning backwards in simple, one-step inferences from the effects of processes to their causes. A *scenario* is a pair consisting of a diagrammatic *pattern* and a *local interpretation*, a sequence that could have caused the pattern. For example, in solving the example in Figure 3, we twice used the following scenario:

| *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 explanation for the pattern's occurrence. Each scenario may consist of several interpretations. For example, given the pattern

**<sedimentary-1>**
**<sedimentary-2>**

the most likely interpretation is that **<sedimentary-1>** was deposited on **<sedimentary-2>**. However, another plausible interpretation is that **<sedimentary-2>** was deposited on **<sedimentary-1>** and then the whole region was tilted upside-down. Simplicity would suggest the first interpretation, but if this leads to an inconsistency we would try the second interpretation. We have found that about a dozen scenarios, each consisting of from one to three interpretations, suffices for the geologic interpretation problems we have looked at so

far. These results, of course, are still tentative, and no program has yet been designed to generate interpretations (see Chapter 7).

By matching scenario patterns throughout the diagram and combining the local interpretations obtained from the matches, we can generate sequences that purport to explain how the region was formed. However, these sequences might not be completely valid explanations. One reason is that a collection of local inferences may not produce a globally consistent answer. This might occur in cases where the local interpretations include events which have global effects. For example, interpreting the pattern $\frac{\text{<sedimentary-1>}}{\text{<sedimentary-2>}}$ to mean that **<sedimentary-2>** was deposited on **<sedimentary-1>** and then the whole region tilted, implies that all the other local interpretations must be consistent with this occurrence of tilt. Another source of inconsistency is the incompleteness of the geologic record. For instance, there is no evidence in Figure 3a for the occurrence of the process of uplift (we inferred that uplift occurred because erosion takes place above sea-level, while deposition takes place under the sea). To detect both types of inconsistencies some form of global reasoning is needed. We have developed a new technique, called *imagining*, which suffices to test for an invalid sequence of events.

### 2.2.2 Imagining

Imagining is a simulation technique developed out of the intuition that one can test an hypothesized sequence of events by "viewing it in the mind's eye". Imagining takes as input an initial state, a goal state (in our case, a diagram cross-section) and a candidate sequence of events. The imaginer simulates each of the events in turn; for the geologic interpretation problem, this is accomplished by constructing diagrams which represent the spatial effects of the geologic processes. The final state produced by the simulation is matched against the goal state. If they match, we can conclude that the candidate sequence is valid.

As discussed in Chapter 1, the major focus of this research has been to investigate the use of the imagining technique in problem solving. Therefore, we present here only a brief overview of imagining and postpone a detailed account until Chapter 3.

First, for each event in the sequence, the imaginer must determine if the event can be applied to the current state of the simulation. For example, if the event is "erode shale" but the shale is currently below sea-level, then we cannot perform the erosion. If the imaginer detects an inapplicable event, it returns an explanation of the problem encountered, consisting of the difference between the current state and the state that would be needed in order to simulate the event. In the above example, the difference reported would be that the shale is below sea-level but that it should be above sea-level in order for the erosion to occur.

Second, the imaginer must infer numeric parameter values for the events being simulated. The sequences generated by the scenario matcher do 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 diagram and the final diagram produced by the simulation, the parameters used in the simulation of an event must closely match those parameters used in the actual geologic event. For example, in order to simulate "deposit shale" the imaginer must have some indication of the thickness of the shale to deposit.

The imaginer uses measurements taken from the goal diagram, along with knowledge of geologic processes, to determine these parameters. It first measures a parameter in the goal diagram. It then correct for any changes that occurred to the parameter between the time when the event occurred and the time represented by the goal diagram. A qualitative simulation of the sequence of events is performed in order to determine which changes occurred. For instance, the imaginer can measure in the goal diagram (Figure 3a) the thickness of the **shale** deposit (which turns out to be 500 meters). Since it also knows that part of the original **shale** deposit had been eroded away (in step 7, Figure 3c), it infers that

the original thickness of the **shale** must have been greater than the measured thickness in the diagram. Since it cannot infer the exact amount of the erosion, the best it can determine is that the original thickness was "greater than 500 meters". Reasoning in this fashion, the imaginer can establish ranges of values for the parameters of all the events involved in creating the region, and thus the quantitative simulation can approximate the effects of the actual geologic events.

### 2.2.3 Gap Filling

If the imaginer detects a "gap" between the state needed for some event to occur and the actual state of the environment (as would have occurred if we had not inferred the presence of the uplift in Figure 3), we need to hypothesize some sequence of events to fill the gap. As described in the previous section, the imaginer indicates why it could not continue in terms of the difference between two states. From that, one can reason about which process or sequence of processes would have the effect of eliminating the difference. This is means-end analysis [Newell, 1963] used in a restricted context. Thus, the failure of a step in the imagining can be used directly to "debug" the faulty sequence. As with the scenario matching technique, we have not yet further developed or implemented any of these ideas.

## 2.3 Geologic Vocabulary

There are three basic geologic features which we need to reason about -- rock-units, boundaries, and geologic points (see Figure 5). 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.

Fig. 5. Examples of Geologic Vocabulary

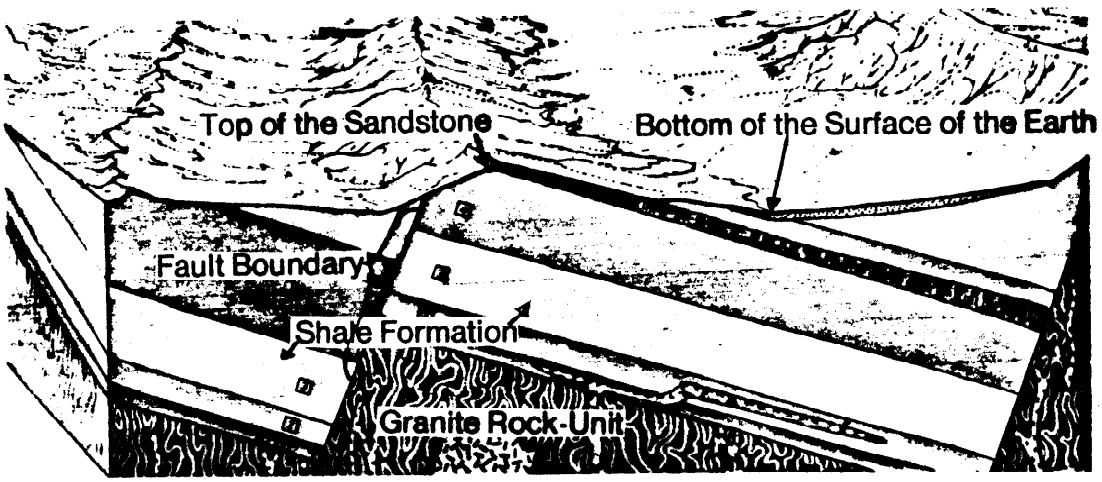



Illustration adapted from [Shelton], page 253.

---

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. When an intrusion occurs, a boundary is created between both sides of the intruded formation and the existing rock-units.

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. In essence, rock-units and boundaries can be thought of as sets of geologic points.

The geologic model we employ is a simple model known as "layer cake" geology (for example, see [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 through the region. Also, the "layer cake" model 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 many geologic interpretation problems.

# 3. IMAGINING

This chapter deals with three main issues. First, we argue that a simulation technique is useful for the task of testing sequences of geologic events. Second, we discuss why imagining, which performs both a qualitative and a quantitative simulation, is necessary in domains like geologic interpretation. Third, we examine in detail each of the four stages of imagining -- qualitative simulation, parameter determination, quantitative simulation and matching.

## 3.1 Simulation

As indicated in the previous chapter, the imagining technique is used to test the validity of candidate solutions, by determining whether the cumulative effects of a sequence of events will lead to a goal state. In this section, we discuss why a simulation technique was chosen for this task.

### 3.1.1 What is simulation?

First, what do we mean by simulation? We define simulation as a particular type of search through a state-space representing the states of the world at various points in time. The search task is to find a path through the space from a given initial state to a given goal state.

Starting with the initial state, the simulator chooses an *action* to be performed and generates the next state by representing the effects of the action on the current state. The transition is successful if the simulator can apply the action to the current state. A distinguishing feature of simulation is that states are traversed strictly in temporal order. Thus, a transition is always made to states representing the world at a later point in time, and the result of a simulation is a time-ordered sequence of world states.

Simulation can be contrasted with techniques like backward chaining, as in Mycin [Davis, 1977], where the search is not done temporally, or with cases where the initial and goal states are not temporally ordered, as in theorem-proving [Bledsoe]. Simulation, which is characterized by applying actions, can also be contrasted with techniques which reason about the character of the actions themselves. A classic example is the Konigsberg Bridge problem where the task is to cross exactly once each bridge connecting the islands of the city of Konigsberg. For years, people tried to solve this problem by simulation: choose a bridge to cross, subject to the constraint that it had not yet been traversed. Finally, Euler proved that it was impossible, using the fact that the number of times one enters and exits an island must be equal.

### 3.1.2 Why use simulation to check solutions?

Why use simulation to test the validity of a sequence of geologic events? The simple answer is: since we are given a sequence of events, it becomes trivial to choose which action to perform at each step of the simulation. This makes the search relatively easy: generate the successive states in accordance with the sequence of events and match the goal state with the final state resulting from the simulation.

What about using other types of search? We hold that for the task of verifying a sequence of geologic events simulation is the most straightforward search technique, both conceptually and computationally. For example, the technique of searching backwards from the goal state might seem attractive, since the initial state is usually very simple (such as "only bedrock") and so matching would be much easier. This type of search would involve trying to determine what the world looked like *before* an event happened. However, for the geologic domain we cannot invert these events unambiguously. For example, it would be quite difficult to look at an eroded surface and determine what it looked like before the erosion. Thus, we would still need a secondary search to eliminate the ambiguities.

Another technique might be to search for a path between the initial and goal states (using, for example, a best-first search) and then to match that path against the candidate sequence. The problem with this technique is that the search tree would be enormous, since almost any geologic process can occur at almost any time. In addition, since there may be multiple paths to the goal state, just finding any path is usually insufficient.

A powerful search technique is to reason about the characteristics of the events themselves. However, since the geology of the Earth is such a complex domain, and our knowledge of it is so limited, it is not even clear what such reasoning would involve. Perhaps one would create a mathematical model of the geologic processes and solve the equations. Surely geologists do not do this -- from our observations, they seem to solve these types of problems using some sort of simulation technique.

One feature of simulation is that it generates the sequence of world states along the search path. This is quite important in cases where we are more interested in how we get from the initial state to the goal state than in whether we can get there. One such case, needed for imagining, is reasoning about the changes that occur in order to infer the initial values of the process parameters (see Section 2.2.2). By generating all the states along the solution path, we also generate the necessary sequence of changes to objects. Another case where the intermediate states are needed is in *gap filling* (see Section 2.2.3). Recall that a step in the simulation fails if an action is inapplicable. We can describe this failure in terms of the difference between two states -- the current state and the one we are trying to achieve. The gap filling technique analyzes this difference in order to hypothesize missing events in the sequence.

## 3.2 Why Use Two Simulations?

Imagining uses a combination of qualitative and quantitative simulations to test the validity of a sequence of geologic events. Obviously, doing two simulations is more work than doing one. So why do both? We claim that neither alone is adequate in this domain because the candidate sequence of events is stated in qualitative terms, while the goal diagram is a quantitative description. This qualitative/quantitative distinction leaves us with two options. First, we can qualitatively simulate the sequence of events, transform the goal diagram into a qualitative description, and match the two to determine if the sequence is valid. Second, we can recast the sequence into quantitative terms, do a quantitative simulation, and match the final result of the simulation with the goal diagram. In this section, we argue that the first technique is inadequate due to the nature of qualitative representations and the second technique necessitates doing a qualitative simulation anyway, in order to establish the quantitative sequence of events.

### 3.2.1 Why not qualitative simulation alone?

Why is a qualitative simulation followed by a qualitative match insufficient? One problem is that certain geologic features that help determine a successful match, such as the shape of formations, are difficult to express in qualitative terms. It is even more difficult to simulate adequately the qualitative changes to these features. For example, how a formation will split due to faulting, or what the shape of a formation will be after erosion are both difficult to express qualitatively. However, these changes are important in determining whether two regions are identical.

A second problem is that qualitative models are inherently ambiguous, in that a single qualitative representation maps to many real-world situations. For example, if the tilt of a formation is specified qualitatively, such as "tilting clockwise", there is a wide range of actual tilts which match that description. This ambiguity makes the matching problem very difficult. As in other research dealing with qualitative representations (e.g. [deKleer, 1975],

[deKleer, 1979], [Forbus, 1981]), we have found it necessary to use quantitative knowledge to reduce or eliminate the ambiguities.

The source of both these difficulties is that qualitative representations abstract certain kinds of information, such as shape or degree of tilt, which are needed to determine a successful match. We need a quantitative representation of space, which in our case is obtained by constructing diagrams, to perform the matching adequately.

### 3.2.2 Why not quantitative simulation alone?

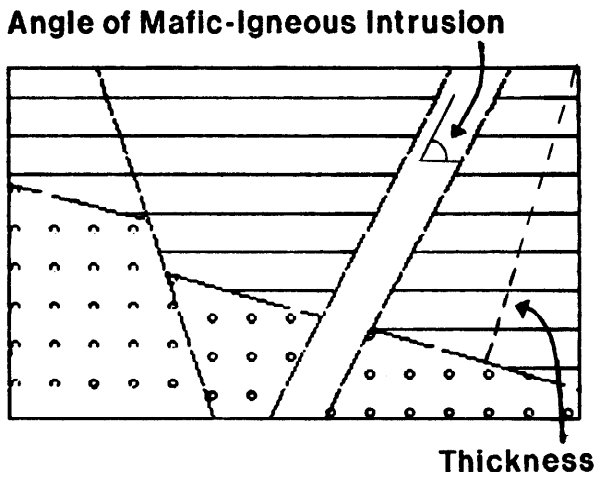
Having argued that a quantitative simulation is necessary for matching, we now show why it alone is not sufficient. In order to do the quantitative simulation, we need a sequence of events stated in quantitative terms. We are given the sequence of events, but without the *process parameters*. For instance, to do deposition we need to know the thickness of the deposit and for tilting we need to know the degree of tilt. Doing a quantitative simulation requires numeric values for these parameters.

One obvious way to obtain the numeric parameter values for a process is by making measurements in the goal diagram. This requires that the changes caused by the process leave some "trace" indicating the values of the parameters. Uplift, for example, has no discernible effect on the diagram, hence the amount of uplift is not measurable. However, the angle of an intrusion or the thickness of a deposit can be measured (see Figure 6a).

However, there is a problem. These measurements represent the final values for the parameters -- they might have changed over time due to earlier events. In order to determine the original value of a parameter, we must "correct" for changes to the parameter. This requires reasoning about the accumulated effects of events. For example, one of the parameters of intrusion (Figure 6b, Step 4) is the angle of intrusion. Measuring in the goal diagram (Figure 6a), we find that the angle of the mafic-igneous is $62^{o}$. However, since we know that the intrusion was rotated by the tilting (Figure 6b, Step 5), we can infer

Fig. 6. Measuring from the Diagram

a.




b.

1. Deposit **sandstone**
2. Deposit **shale**
3. Uplift
4. Intrude **mafic igneous** through **sandstone** and **shale**
5. Tilt
6. Fault
7. Erode **shale** and **mafic-igneous**

---

that the actual angle at the time of intrusion was ($62^{o}$ - THETA). Likewise, we can determine that the value of the tilt parameter THETA is $-16^{o}$, since we can measure that the sandstone formation now tilts by $-16^{o}$, although we know it was originally deposited horizontally. Thus the original angle of intrusion must have been $78^{o}$. Similiarly, the thickness of the shale measured in the diagram is 500 meters, but since we know that some was eroded (Figure 6b, step 7), we can infer that the initial amount of deposition must have been greater than 500 meters. We refer to these as "corrected" parameter values, since the values approximate those actually used in forming the region.

The above reasoning requires knowing the cumulative changes to each process parameter. To do this reasoning we need to *qualitatively* simulate the events in the candidate sequence. The qualitative simulation will yield the needed sequence of changes to the process parameters. Recall that one of the features of simulation is that it constructs the intermediate states along the solution path, each of which represents the changes to the world after the occurrence of an event in the sequence. Thus, the complete sequence of intermediate states produced by the qualitative simulation, together with the quantitative goal state, provides us with the knowledge necessary to determine corrected parameter values.

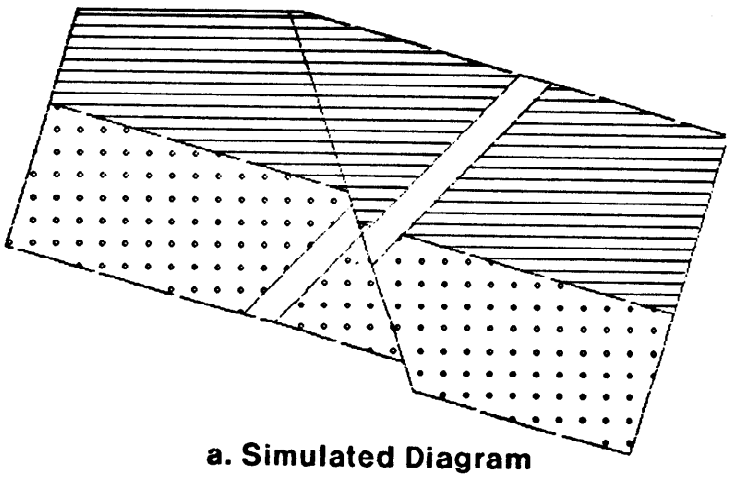
*Corrected* parameter values are necessary because matching is sensitive to the values used. In general, we cannot tell the difference between a mismatch resulting from an incorrect candidate sequence and one resulting from the simulation of a valid solution using uncorrected parameter values. For example, Figure 7a shows a simulation done with parameter values that were measured in the goal diagram (Figure 7b), but not corrected. In comparing Figure 7a with Figure 7b, it is not clear whether the differences arise because the candidate sequence is invalid or because the parameter values were badly chosen.

Imagining thus consists of four steps, each necessary to support the next step.

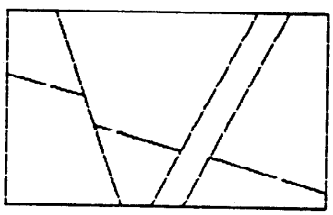
1. Perform a qualitative simulation using the qualitative candidate sequence in order to establish the sequence of changes to parameters.
2. Use the results of the qualitative simulation and the quantitative goal state to infer values for the process parameters.
3. Perform a quantitative simulation, which in our case involves constructing a sequence of diagrams.
4. Match the end result of the quantitative simulation with the goal state. If the match is successful, conclude that the sequence of events is a valid explanation of how the goal state was formed.

---

**Fig. 7. Simulation Using Uncorrected Parameter Values**



**a. Simulated Diagram**



**b. Goal Diagram**

We believe that imagining may be useful in other domains where a qualitative sequence of events must be tested to see if it yields a given quantitative result. Theories in sciences such as economics, chemistry or archaeology are often presented in the form of "the occurrence of this (qualitative) sequence of events explains this (quantitative) data". The scientist might test the theory by "imagining" the events occurring. These domains are adequate to support imagining because the parameters of many of the events can be inferred using the test data.

### 3.3 The Imagining Technique

In this section, we examine in more detail the four stages of imagining -- qualitative simulation, parameter determination, quantitative simulation and matching.

#### 3.3.1 Qualitative simulation

There are two reasons for the imaginer to do a qualitative simulation. First, the imaginer needs to generate sequences of changes for use in determining parameter values. Second, the imaginer must check that each event in the sequence can be applied to the current state and that the resultant state satisfies any constraints which might be imposed by the event. In this section, we examine how our formulation of qualitative simulation facilitates these tasks.

To perform the qualitative simulation we need a representation of objects, a representation of processes and a method of simulating the effects of the processes by applying them to the objects. In this section, we concentrate on the form of the initial world state and describe how a qualitative event is applied to a qualitative state representation, leaving a detailed examination of the representations of objects and processes until Chapters 4 and 5.

#### 3.3.1.1 The initial qualitative state

Before we can simulate the first event we must create a representation of the initial state of the world. This consists of objects representing bedrock and the surface of the Earth, which is asserted to be horizontal. We also assert that the bedrock lies along the surface of the Earth. All of these are assumptions, since the bedrock never actually appears in the goal diagram. However, the assumptions are the most likely in the absence of any other knowledge of the initial state of the world.

#### 3.3.1.2 Expressing change qualitatively

What happens when an event occurs? Things might be created (as in deposition), things might be destroyed (as in erosion), existing things might be altered (as in tilting or faulting) or constraints might be imposed (such as, in our model of geology, the top of the surface is below sea-level after deposition).

To simulate the creation of an object, such as a new sedimentary formation, we build a representation of the object and assert that it began its existence at the beginning of the event that created it. This enables us to determine which objects are affected by "global" processes, such as uplift, that affect all formations which exist when the processes occur. We also specify the initial values of the attributes of these created objects, such as the thickness and composition of a formation. To simulate the destruction of an object, such as the complete erosion of a formation, we merely assert that its existence ended at the end of the event which destroyed it.[2]

---

2. This is not totally accurate -- the object could have been destroyed at any time during the event. However, since we are using a discrete model of geology (see Section 5.1), this formulation of destruction is adequate.

Notice that we do not delete the representation of the destroyed object. Although we want to represent that the object no longer exists with respect to the simulated geologic time, we still want a record kept of its existence for reasoning about what the object used to be like. This is the essence of our approach to representing change -- the effects of change should be additive. The discovery that something has changed should only add knowledge, never delete any.

This idea is also used in dealing with changes to an attribute of an object, such as the change in orientation of a formation due to tilting. We want to maintain the complete history of values of an attribute over time. We can capture this idea of additive change by describing how attributes are altered in terms of "change equations".[3] These are symbolic expressions indicating how to update the value of an attribute to reflect the occurrence of an event. For example,

```
height of the shale after uplift =
          height of the shale before uplift + uplift-amount
```

represents the change that occurs to the shale as a result of an uplift. This description has the advantage of explicitly representing how the attribute changes from one point in time to another. By accumulating these equations at each step of the simulation, we can construct an expression representing the cumulative changes to an attribute. For example, the changes to the height of the top of the sandstone formation (see Figure 6a) are represented by the following change equations:

```
1. Deposit Sandstone -
    height of sandstone top after deposition =
              DLEVEL + height of surface bottom before deposition

2. Deposit Shale - <no change>

3. Uplift -
    height of sandstone top after uplift =
              height of sandstone top before uplift + uplift-amount

4. Intrude Mafic-Igneous - <no change>
```

---

3. The actual representation used is more complex, and will be described in detail in Chapters 4 and 5.

```
5. Tilt
    height of sandstone top after tilt =
                cos(theta) * height of sandstone top before tilt
                + sin(theta) * lateral of sandstone top before tilt[4]

6. Fault - <no change>

7. Erode - <no change>
```

The first equation expresses the initial value of the height of the top of the sandstone in terms of the deposition process parameter "DLEVEL". The rest of the equations indicate how the height changes over time due to the occurrence of the events. Now, to determine the cumulative effect on the height of the top of the sandstone, we combine the equations above, which yield:

```
height of sandstone top now (in goal diagram) =
                cos(theta) * (height of sandstone top after deposition
                              + uplift-amount)
                - sin(theta) * lateral of sandstone top before tilt
```

We can use the goal diagram and such equations to determine parameter values. We use the goal diagram to measure the final value of a parameter, such as the height of sandstone. The equations allow us to correct for the changes that occurred to the parameter.

Besides specifying which objects were created and how existing objects changed, the simulation needs to specify relations between the attributes of objects. For example, in our model of geology, the height of the top of the surface of the Earth after erosion is constrained to be greater than sea-level (that is, erosion occurs only above water). These constraints are also represented using equations and can be used to help determine parameter values more precisely.

---

4. The "lateral" of a point corresponds to the X-coordinate of the point in the diagram.

#### 3.3.1.3 Detecting inconsistent sequences

As described in Section 3.1, an important task in a simulation is determining whether a process is applicable and whether constraints on the new state are satisfied. To determine whether a geologic process can occur, we specify *preconditions* for the process which must be true in the current state in order to perform the simulation. In our model of the geologic world, there are few process preconditions -- most processes can happen anywhere and at any time. The two preconditions in our model of geology are that erosion can occur only above sea-level and that deposition can occur only below sea-level.

A second consistency check is whether constraints on the simulated state are satisfied. For example, the scenario matcher might infer the event "deposit shale on the sandstone". This embodies the constraint that after deposition the shale and sandstone share a common boundary. This can occur only if the sandstone lies along the surface of the Earth before the deposition begins. Thus, if some other formation is covering the sandstone, the imaginer will detect an inconsistency.[5]

#### 3.3.1.4 Doing the qualitative simulation

Using the requirements for a qualitative simulation presented above, we have developed a five step algorithm for qualitatively simulating processes.

1. Check whether the preconditions are satisfied.
2. Create a representation for each object created by the process.

---

5. In order to do the "gap filling" described in Section 2.2.3, the imaginer must both detect inconsistencies and determine the differences between two states. The implemented system detects inconsistencies but does not currently compute the differences. This is not due to any conceptual difficulties, but rather reflects a choice of implementation priorities.

3. Assert that the constraints on the simulated state must hold.
4. Construct change equations to represent the effects of the process on existing and newly created objects. Check if any of the equations are inconsistent with the current state of the world.
5. Assert that all of the relations induced by the process hold. Check if any of the relations are inconsistent with the current state of the world.

### 3.3.2 Parameter Determination

The task of parameter determination is to infer the values needed for doing the quantitative simulation. We use parameter determination to infer such things as the point in the goal diagram which corresponds to "the top of the shale after deposition", the numeric value of "the height of the top of the shale after deposition" or the numeric value of the expression "DLEVEL + UPLIFT-AMOUNT". The basic concept of parameter determination is quite simple. To determine the value of an entity, we try to measure it in the goal diagram. If it cannot be measured directly, we try to find an expression with the entity on the left hand side and try to determine the values of each component in the right hand side of the expression.

The entities we can measure directly in the goal diagram are the current values of attributes. For example, we can measure "the height of the top of the surface at NOW". However, we cannot measure "the height of the top of the surface at the end of step 1", since it is not the current value, and we cannot measure "DLEVEL", since it is not an attribute. The actual mechanism for making measurements is described in detail in Section 4.2.2.3. We have implemented only a small subset of all possible measurements, in order to determine the minimum number necessary to do parameter determination. The measurements the system can make are:

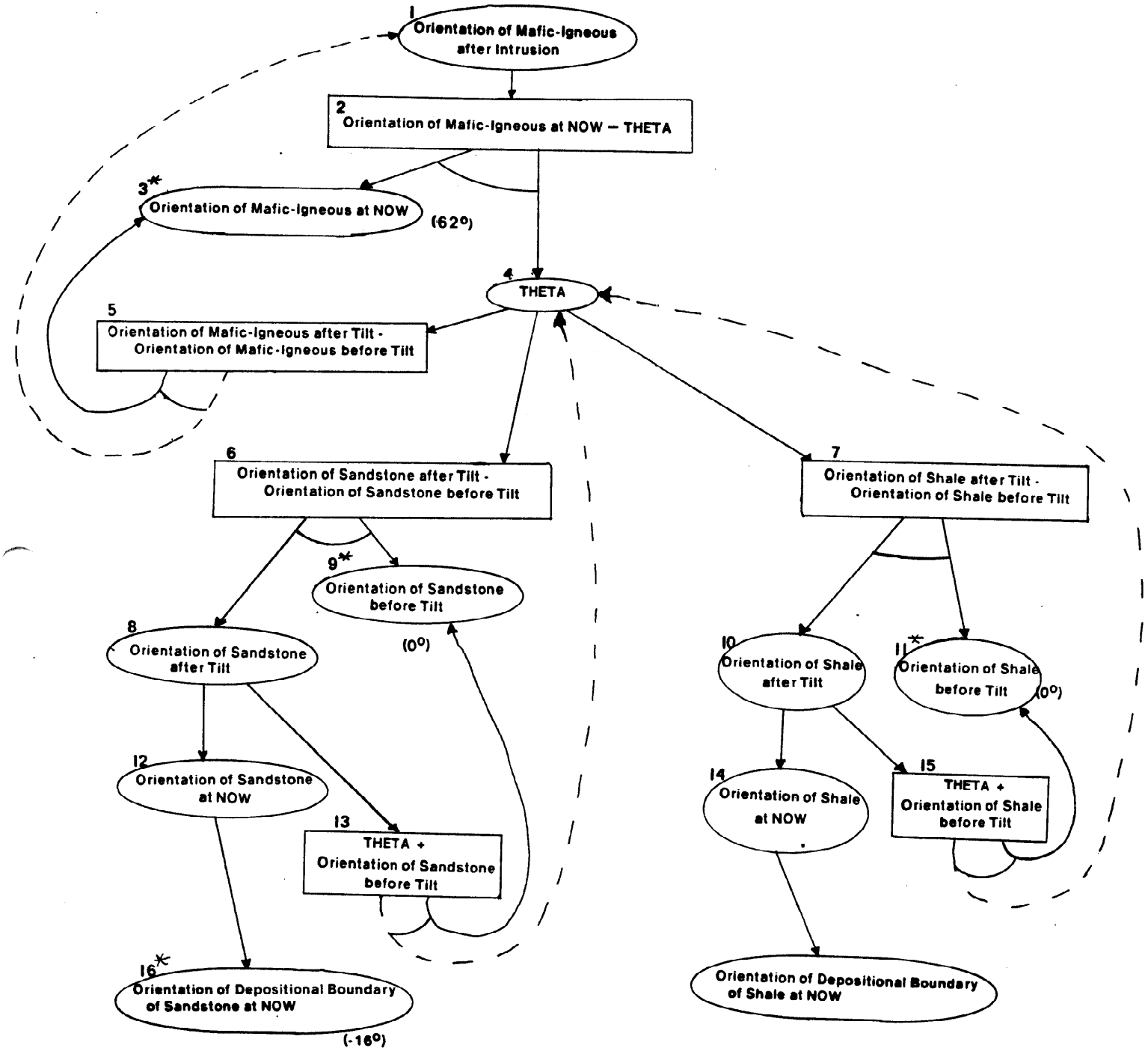
1. Finding the height (Y-coordinate) or lateral (X-coordinate) of a point (yielding a number).
2. Finding the top or bottom of a boundary (yielding a point)
3. Finding the "location" of a boundary (yielding a point).
4. Finding the thickness of a rock-unit (yielding a number).
5. Finding the slip of a fault (yielding a number).
6. Finding the orientation of a boundary (yielding a number).

If an entity is not measurable in the goal diagram, we try to find an expression that contains it. These expressions are constructed by the qualitative simulation and include both change equations and process constraints, as described in Section 3.3.1. The basic task is to find all expressions containing the entity, to solve symbolically for that entity and then to try to determine the value of one of the resulting expressions by finding values for the components of the expression. For example, suppose we want to determine the value of "C" and we find the equation "A = C - B". We solve for "C", determine the values of "A" and "B", and add them. This technique is naturally expressed as an AND-OR search: an AND node corresponds to determining the value of an expression, an OR node corresponds to determining the value of one of the equalities or inequalities.

Our parameter determination algorithm uses a simple breadth-first search, taking care to avoid infinite loops. It turns out that making measurements is computationally quite efficient. Thus, when faced with the choice of searching the AND-OR graph or making a measurement, the algorithm will always opt for making the measurement.

Figure 8 shows part of the AND-OR graph used to determine the value of "the orientation of the mafic-igneous at the time of intrusion". The nodes are numbered to show the order of the search. Dashed lines show those relationships which were not added because they would have led to cycles. Starred nodes indicate entities whose values can be determined

Fig. 8. Partial AND-OR Graph for Parameter Determination




directly. Working these values back through the graph, we can infer that the orientation of the mafic-igneous at the time of intrusion was $78^{0}$.

The graph indicates that the orientation of the mafic-igneous at the time of intrusion equals the orientation of the mafic-igneous measured in the goal diagram minus THETA, the change due to tilting. The parameter THETA can be determined as the difference between the orientation of the sandstone formation after the tilt (which is equal to the orientation of the depositional boundary, which can be measured in the goal diagram) and the orientation before the tilt (when the sandstone was horizontal).

There are cases where the parameter determination will come up with a range of values, rather than an exact value. Consider again the example in Section 3.3.1.2, where we wanted to determine the cumulative effect on the height of the top of the sandstone. We found that the resultant change equation was:

```
height of sandstone top now (in goal diagram) =
                cos(theta) * (height of sandstone top after deposition
                                + uplift-amount)
                + sin(theta) * lateral of sandstone top before tilt
```

Solving for "height of sandstone top after deposition" we get:

```
height of sandstone top after deposition =
                {[height of sandstone top now -
                   sin(theta) * lateral of sandstone top before tilt]
                 ÷ cos(theta)}
                - uplift-amount.
```

Running our parameter determination algorithm we come up with:

```
height of sandstone top after deposition
                = {[28.0 - (sin(-16°) * -15.0)] ÷ cos(-16°)} - "> 0"
                = 24.8 - "> 0"
                = "< 24.8"[6]
```

---

6. How we do arithmetic on quantities expressed as ranges of values will be described in Section 4.3.

This results from the fact that "height of sandstone top now" can be measured directly and that "theta" and "lateral of sandstone top before tilt" can be determined exactly using this parameter determination technique. However, "uplift-amount" cannot be measured or determined; all we know is that is some positive amount. Thus, the best we can do is to determine the value of "height of sandstone top after deposition" within some range.

This seems to present a problem because the quantitative simulation needs exact parameter values. Which value should we select from the range? Recall that we need values which closely approximate the actual geologic parameters to make tractable the matching of the goal diagram and the final simulated diagram. Is the matching process affected by our choice of value?

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 realizing that the value of an entity is known only within a range because it depends on an entity which leaves no trace in the goal diagram. The effects of uplift, for example, are not discernible by analyzing a single diagram, so the "uplift-amount" cannot be measured and so cannot be determined any more precisely than that it is positive. Since we cannot detect the effect of "uplift-amount" in the goal diagram, we can be assured that its effect on parameter values will eventually be canceled out. For example, we have determined that "height of sandstone top" is less than 24.8.[7] Suppose we arbitrarily pick a value for "height of sandstone top" within the allowable range, say 20.0. This, in turn, constrains "uplift-amount" to be 24.8-20.0 or 4.8. Then, after we simulate the uplift, the "height of sandstone top" will become 20.0 + 4.8, or 24.8. Now, suppose we had chosen "height of sandstone top" to be 18.2. "Uplift-amount" would be constrained to be 6.6 and, after uplift, "height of sandstone top" would become 24.8, the same height as before.

---

7. The unit of measurement is irrelevant.

In general, by restricting the range of values for a parameter to an exact value in order to do the simulation of one step, we constrain the magnitudes of subsequent changes. When the events which caused those changes are simulated, the effects caused by the arbitrary choice of value will be canceled out. Thus, the cumulative effect on an entity due to the quantitative simulation will be the same as the value measured in the goal diagram, no matter what value is actually chosen. If we cannot determine a parameter value exactly, we can be assured that choosing an arbitrary value within the allowable range will have no effect on the final result of the quantitative simulation.

### 3.3.3 Quantitative Simulation

The quantitative simulation is needed to produce a diagram that can be matched against the goal diagram. In order to have confidence that a successful match implies a valid solution sequence and an unsuccessful match means that the sequence is not a valid solution, the quantitative simulation must accurately model the changes produced by the sequence of events. In this section, we present our quantitative simulation technique and discuss how it facilitates the matching.

As noted earlier, the quantitative representation is based on the notion of diagrams. Although using diagrams is not crucial to the concept of imagining, we have chosen to use them for three reasons. First, the reason for doing the simulation is to produce a representation to match against the goal state, which in our case is a diagram. Thus, it makes sense for the result of the simulation also to be a diagram. Second, an important aspect of matching in this domain is to check whether the shape and adjacencies of rock-units are the same in both the goal state and the result of the simulation. Using a representation based on diagrams makes it is fairly easy to accurately represent knowledge about adjacency and shape.

The third reason for using diagrams is that most geologic effects are spatial in nature. For example, rock-units change shape due to erosion and the position of geologic points change due to tilting. It is relatively easy to model these spatial effects as changes to a diagram. For example, erosion may be modeled by erasing part of the diagram and tilting may be modeled by rotating the diagram.

#### 3.3.3.1 Quantitatively simulating change

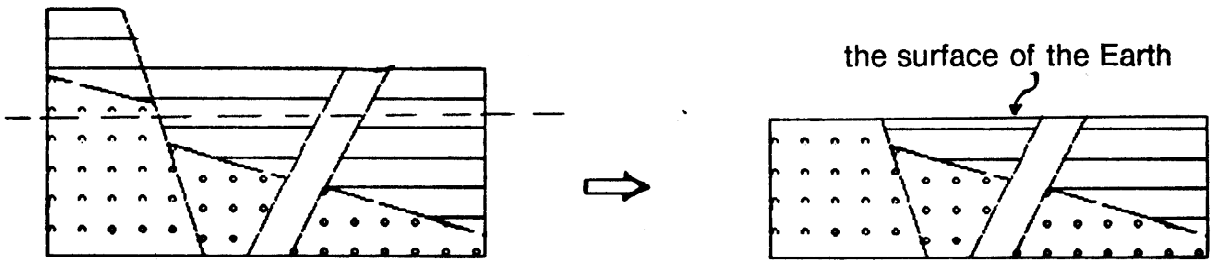
In keeping with our philosophy that the representation of change should be additive, the quantitative simulation creates a sequence of diagrams (one for each event in the candidate sequence), rather than simulating all the events in a single diagram. Thus, the quantitative simulation produces a "movie" of the formation of the region or, more accurately, a series of snapshots.

Quantitative simulation is done in three steps:

1. Copy the diagram representing the current state.
2. Determine all of the parameter values necessary to simulate the event, using the technique described in Section 3.3.2.
3. Apply the quantitative process representation to the copied diagram.

A quantitative process is represented as an algorithm which takes as input a diagram and process parameters, and modifies that diagram to reflect the spatial geologic changes. For example, since erosion occurs horizontally in our geologic model, its parameter is the height of the eroded surface. Erosion is simulated by drawing a line at that height and erasing all parts of the diagram which lie above the line (see Figure 9). In order to interpret the resulting diagram, we need to associate the new parts of the diagram with geologic objects. In Figure 9, for example, the line along the top of the diagram is associated with the surface of the Earth. This technique has been used for simulating a number of sequences of events and appears to work well. The sequences of diagrams in Figure 3 and in Appendix E were

**Fig. 9. Quantitative Simulation of Erosion**




---

produced by this system.

Recall that one of the tasks of simulation is to check for the applicability of events. Since we already check for this in performing the qualitative simulation (see Section 3.3.1.3), it might appear unnecessary to do so again here. However, this is not quite correct. There are cases where the more exact nature of the quantitative representation would enable us to detect an inconsistency that is not detectable by the qualitative simulation. For example, if the candidate sequence constrains an intrusion to cut across two faults, the quantitative simulation might discover that there is no angle of intrusion which satisfies both constraints. However, the qualitative simulation might not detect an inconsistency, due to the abstract nature of the qualitative spatial descriptions. Adding consistency checking to the quantitative simulation is quite straightforward -- it involves modifying the parameter determination algorithm to check whether all the expressions equated with a parameter have the same value. We felt, however, that since the number of such cases is small, adding this feature to the current system would be tedious without adding much conceptually to the task of doing imagining.

### 3.3.4 Matching

Because of the care taken to do the quantitative simulation using corrected parameter values, we can use a very simple algorithm to match the simulated and goal diagrams. Basically, the algorithm checks that the topology of the diagrams match, as do the orientations of all edges. In geologic terms, this means that all the rock-units must have the same adjacencies and shapes in both diagrams.

The first step is to cut the simulated diagram down to the same size as the goal diagram. The matching algorithm then proceeds as follows:

1. Pick the face in the upper left corner of each diagram. Assume that the faces correspond and add the pair to the correspondence queue.
2. Pop a pair of faces off the correspondence queue. If there are no more pairs, then the diagrams match.
3. Pick the upper left-most edge of both faces. Check for edge correspondence (see below). If the edges do not correspond, then the match fails. If they do correspond, then find the next pair of edges by traveling clockwise around the perimeter of the two faces. Check for edge correspondence between this pair, and continue until either the match fails or all edges on the perimeters have been checked. If all the edges correspond, go to step 2.

The algorithm for determining edge correspondence is as follows:

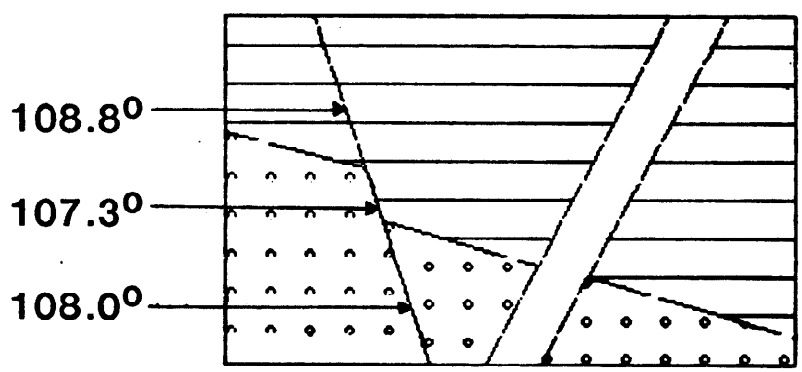
1. Check whether the orientation of the edges are approximately the same. If not, the edges do not correspond.
2. Check whether the faces on the other side of edges are in correspondence. The possibilities are:

1. They are asserted to be in correspondence with each other -- infer that the edges correspond.
2. Either face is asserted to be in correspondence with another face -- infer that the edges do not correspond.
3. Neither face has a face-correspondence assertion and the faces are associated with the same rock-unit -- assert that the faces correspond, add the face pair to the "correspondence queue" and assume that the edges correspond.

Why, in step 1, do we check if the edge orientation is nearly the same? If we are using corrected parameter values, shouldn't the orientation be exactly the same? The problem stems from inaccuracies in the goal diagram (the "signal"). For example, in Figure 10 we see that the fault is made up of three separate edges in the diagram. As drawn, these edges are not exactly colinear, so the orientation of the simulated fault will be different from any single edge making up that fault in the goal diagram.

---

**Fig. 10. Fault Drawn in the Goal Diagram is not Straight**



Due to a desire to work on other aspects of the imaginer and because the matching algorithm is so simple, the matcher has not yet been implemented. One of the open questions regarding the matching is whether the inaccuracies in the goal diagram will affect the match. For example, if two intrusions are very close to one another, slight inaccuracies in the parameter determination might cause the simulated intrusions to overlap slightly, causing the match to fail. Although in the vast majority of interpretations this type of situation does not occur, it might be useful to augment the matching algorithm to produce some indication of how good the match is, for example, how many edge or face correspondences fail.

# 4. OBJECT REPRESENTATIONS

The following two chapters present the five representations we have developed -- in this chapter, we discuss the qualitative and quantitative object representations and the quantity lattice. In the next chapter, we present the qualitative and quantitative process representations. Each representation is presented using the same format. First, we present the criteria for choosing the representation. We discuss the particular task in imagining for which the representation is designed and describe some of the features required of the representation. Second, we describe both the representation itself and our choice of implementation. Examples are presented to illustrate how the choice of representation facilitates the task for which it was designed.

## 4.1 Qualitative Objects - Histories

### 4.1.1 Criteria

There are three basic types of geologic objects we need to describe.

1. *Rock-units*, such as a shale formation or the down-thrown block of a fault.
2. *Boundaries*, which are the intersections between rock-units, such as a fault, or between rock-units and other features, such as the surface of the Earth.
3. *geologic points*, which are locations on rock-units or along boundaries, such as the top of a shale or the bottom of the surface of the Earth.

Each type of object has a set of *attributes*, such as the thickness of a rock-unit, the height of a point or the orientation of a boundary.

To reason about change to these objects, we need to represent object creation, object destruction and changes to attributes of objects. We need to reason about whether an object exists at a particular point in time, and what the value of an attribute is at a point in time. In order to do parameter determination, we also need to represent the cumulative effects of events, which involves describing "change equations" which relate the value of an attribute at the start of a process to its value at the end of the process.

### 4.1.2 Description

These criteria led us to choose a frame-like representation, modified to support temporal reasoning, which we call *histories* (our notion of histories is an extension of [Hayes] and is similar to the histories of [Forbus, 1982]). Geologic objects are represented as frames [Minsky] organized into a type hierarchy. Each type of object has certain attributes and constraints associated with it. For example, a **sedimentary deposit** has a **thickness** that is constrained to be positive.

However, histories differ from the common notion of frames in two important respects. First, there are actually two kinds of objects - *temporal* and *abstract*. Temporal objects correspond to physical, real-world objects. They have an associated life-span which enables us to reason about when they were created or destroyed. Trees, people, and sedimentary deposits are temporal objects. Abstract objects are non-physical objects that always exist. Geometric planes, vertices, and quantities are abstract objects.

Second, since we want to represent situations in which the attributes of objects can change over time, the value of a slot is a *time-line* of values rather than a single value. The time-line is a totally ordered sequence of values. For instance, the **thickness** of a sedimentary deposit consists of the sequence of all thickness values of that deposit. In order to retrieve a particular thickness value, we must specify the point in time of interest.

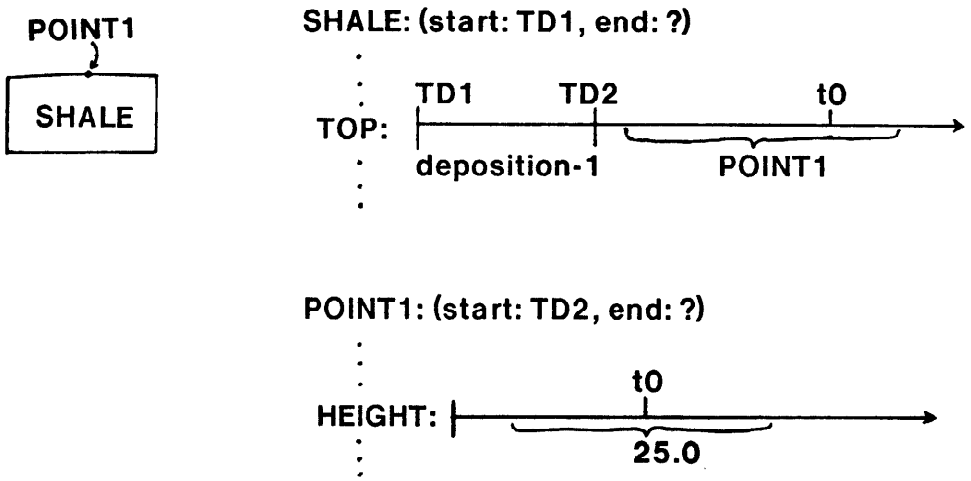
The representation of temporal objects is quite straightforward. Every temporal object has a "start" and an "end" which indicate when the object was created and destroyed. When a frame representing a temporal object is created, its "start" is set to the geologic time when the object began its existence. An object's "end" may be left unspecified -- the system assumes that an object continues to exist unless explicitly told otherwise.

The representation of the time-lines is a bit more complex. The extent of the time-line for an attribute of an object is the same as the lifetime of the object. The time-line is divided into *quiescent* and *dynamic* intervals. A quiescent interval indicates that nothing happened to the attribute during that interval, hence the value of the attribute during the interval is constant. A dynamic interval indicates that some process induced a change to the attribute during that interval. Since we use *discrete* process representations (see Section 5.1), the value within a dynamic interval is defined to be *unknown*, although the value is known at the beginning and end of the interval. In addition, associated with each dynamic interval is the event which caused the change and the "change equation", which relates the values of the attribute before and after the event occurred.

To determine the value of an attribute at a particular time, **t**, we search the time-line to find the interval that contains **t** and return the value found there. If the time point falls outside of the time-line, then the value $\perp$ is returned. $\perp$ is a special value which indicates that "the query did not make sense". For example, the value of the expression "George Washington's hair color in 1492" is $\perp$. $\perp$ is different from the value *unknown*, which indicates that the system has incomplete knowledge of the situation. In addition, $\perp$ is a *strict* value, that is, any operator applied to $\perp$ returns $\perp$.

Let us now examine how histories facilitate reasoning about change to objects. Figure 11a presents a shale rock-unit and a point which represents the top of that rock-unit. Our history representation of the situation is shown pictorially in Figure 11b. The frame representing the shale rock-unit indicates that the shale was created at time TD1, and that as far as we know

Fig. 11. History Representation after Deposition of Shale




---

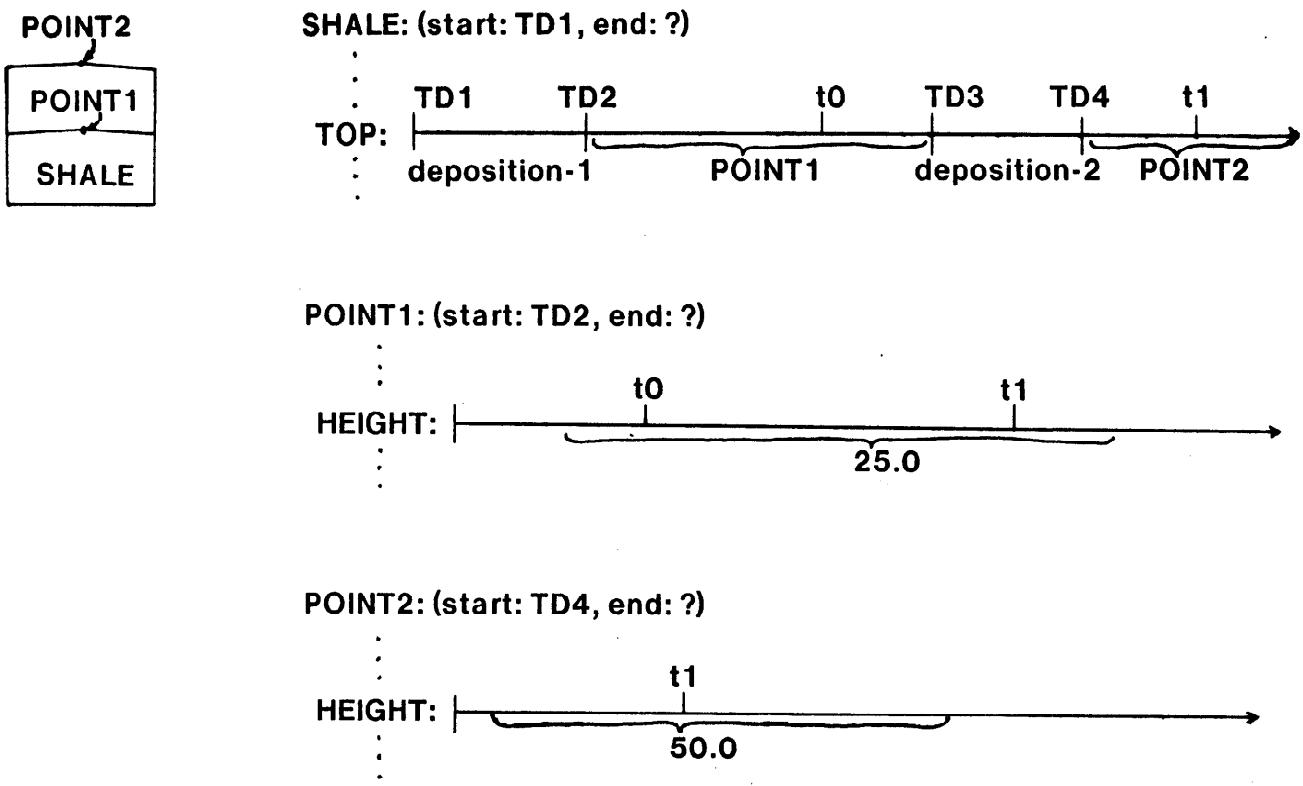
it still exists. One of the attributes of the shale is its top point. The time-line for "top" is divided into two intervals -- a dynamic interval which indicates that deposition occurred from time TD1 to TD2 (this is the deposition that created the shale deposit), and a quiescent interval which indicates that nothing changed the top of the shale after time TD2. To determine "the height of the top of the shale at time **t0**", we first check that the shale exists at time **t0**. We find that the shale exists, because its start time is less than **t0** and we assume that it still exists since we do not know its end time. We then search the "top" time-line to find the interval which contains **t0** and find that the value in the quiescent interval containing **t0** is the object "point1". We then search the "height" time-line of "point1" and find that **t0** falls within the quiescent interval. The value there is 25.0, which is interpreted as the referent of the expression "the height of the top of the shale at time **t0**".

Now, suppose we find out that more shale was deposited. Figure 12 shows the resulting region and its history representation. Note the following changes:

1. A dynamic interval, "deposition-2", is inserted in the quiescent interval of the "top" attribute to represent the new deposition. This also results in the creation of a new quiescent interval after the end of "deposition-2".

---

**Fig. 12. History Representation after Deposition of More Shale**



2. A new point object, "point2", is created which represents the top of the shale after time TD4, and initial values are given to its attributes.

Reasoning just as before, we can determine that the referent of the expression "the height of the top of the shale at **t1**" is 50.0. Note that we can also determine that "the height of point1 at **t1**" is 25.0, the same as its height at **t0**. This indicates that the additional deposition, while changing which point was the top of the shale, did not destroy "point1", which is the "bit of shale" which was the old top point, and that the location of "point1" was unaffected by the additional deposition.[8]

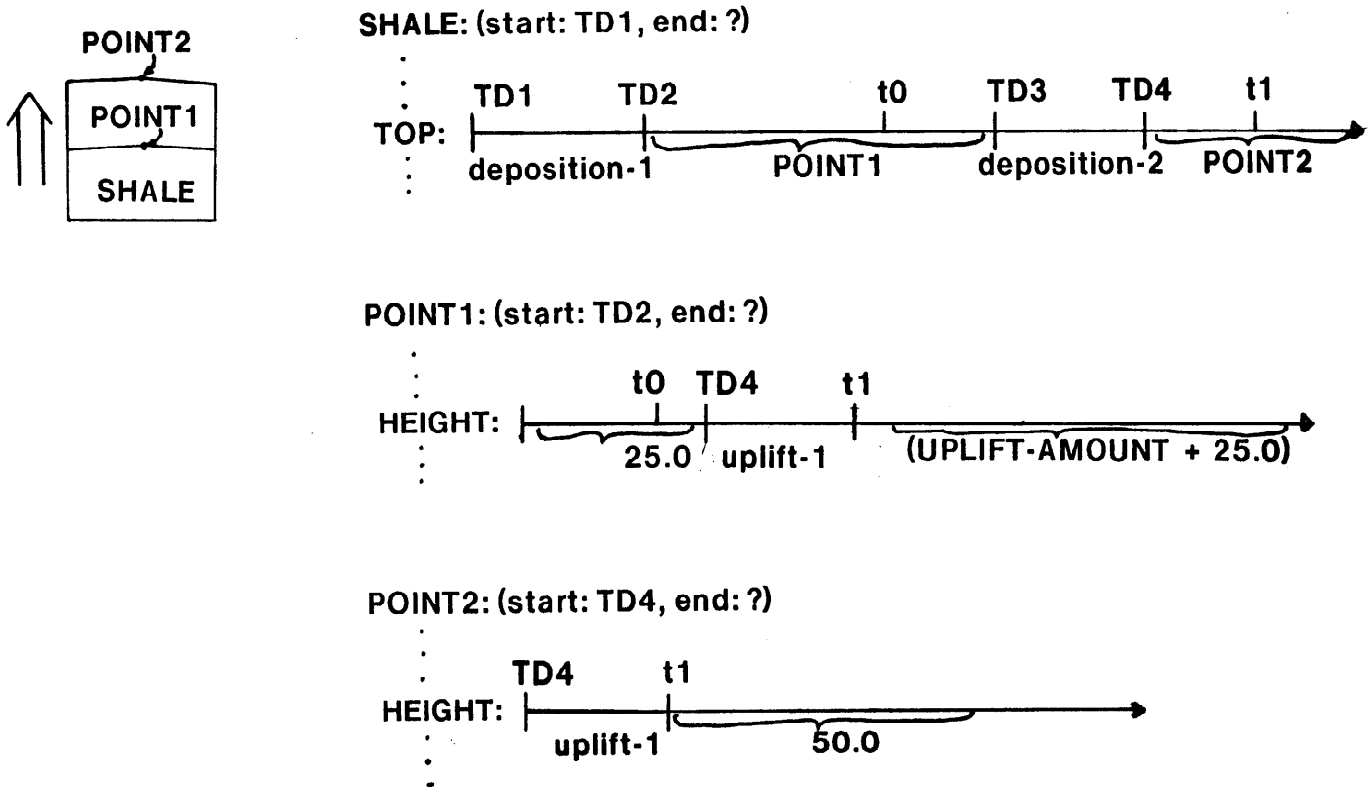
Note also what happens if we try to determine the referent of "the height of point2 at **t0**". Since "point2" did not yet exist at time **t0** (TD4, the "start" of "point2", postdates **t0**), the system returns $\perp$, indicating that it does not make sense to ask about the height of a non-existent object.

Finally, suppose we later learn that some uplift occurred between TD4 and **t1**. The situation would be represented as shown in Figure 13. Notice that only the "height" time-lines of both point objects have changed. These time-lines show how "change equations" relate the before and after values. Also, notice that our assumption of constancy within the quiescent intervals of the original "height" time-lines is no longer valid. In particular, due to the occurrence of the uplift we no longer believe that "the height of point1 at **t1**" is 25.0. This implies that we need some mechanism for informing those who queried about that height that the previous answer is no longer correct. This is discussed further in Section 4.1.2.2.

---

8. That is, given our model of geology. In reality, the added weight of the new shale would tend to compress the existing shale.

Fig. 13. History Representation after Uplift




---

#### 4.1.2.1 The @ operator

In this section, we present a notation developed to describe objects and their attributes at points in time. A notation for describing objects must have two characteristics. First, it must be able to refer to the attributes of objects, such as "the top of the shale". We use the dot notation for this purpose. Second, it must be able to describe the value of an attribute at a particular point in time. That is, since "the top of the shale" refers to a sequence of values over time, we need to select a single value out of the time-line. We have defined the @

operator for this purpose.

The dot notation is a commonly used notation for attributes. For example, "shale.top" means "the top of the shale" and "shale.top.height" means "the height of the top of the shale". The @ operator is a temporal selector and was adopted from work in temporal logic (see [McArthur]). "Shale.top@**t1**" means "the top of the shale at time **t1**". The referent of this description is the geologic point which is the top of the shale rock-unit at time **t1**. Similarly, "Shale.top.height@**t1**" means "the height of the top of the shale at time **t1**" and its referent is a number representing that height.

We have developed a formal notation for expressions involving the attributes of objects at points in time. The BNF grammar for this notation is:

<temporal expression> :: = = <historical expression>@<time>

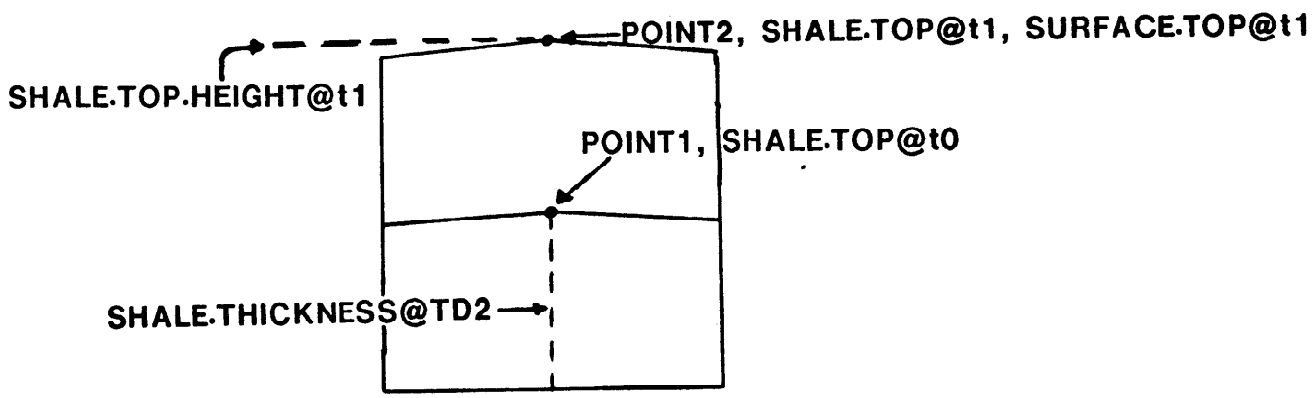
<historical expression> :: = = <object> | <historical expression>.<attribute>

<object> :: = = <temporal object> | <abstract object> | (<temporal expression>)

Figure 14 shows the example presented in the previous section, along with descriptions of the various features. One of the useful aspects of the @ notation is that multiple descriptions can refer to the same object. For example, "point1" can also be described as "Shale.top@**t0**", and so "point1.height@**t1**" can also be described as

---

**Fig. 14. @ Notation**



"(Shale.top@t0).height@t1". This could also be described as "(Surface.top@t0).height@t1" where "Surface" refers to the surface of the Earth. Thus, we can reason about an object as long as we know at least one way to describe it, and changes to an object stated in terms of one description will be reflected in all the descriptions which refer to that object.

Since temporal objects can be created and destroyed, it is useful to define the @ operator over temporal objects as well as over histories.[9] If **A** is a temporal object, we define the value of **A@t** to be **A**, if **A** exists at time **t**, otherwise the value is ⊥ (that is, "the query makes no sense"). For example, the value of "point2@t0" is ⊥, since "point2" did not yet exist at time **t0**.

In light of this, let us re-examine the interpretation of the description "Shale.top@t0". Since the referent of "Shale" might be ⊥ at **t0**, we need to "distribute" the @ operator through the description in order to evaluate the expression. The description "Shale.top@t0" is in fact shorthand for "(Shale@t0).top@t0". This is interpreted as follows: if the shale exists at **t0**, then the value of the expression is the same as before; if the shale does not exist (e.g. it was "destroyed" by erosion or not yet deposited), then the referent of "Shale@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 description "Shale.top.height@t0" is shorthand for "((Shale@t0).top@t0).height@t0", and "(Shale.top@t0).height@t1" is shorthand for

---

9. For any abstract object **B**, **B@t** equals **B**, since abstract objects always exist.

"(((Shale@**t0**).top@**t0**)@**t1**).height@**t1**".

This temporal notation extends naturally to many other temporal domains. For example, "USA.president.hair-color@1982", "USA.president.hair-color@1977", and "(USA.president@1982).hair-color@1950" all have natural interpretations.

#### 4.1.2.2 Descriptions and referents

The @ notation is a *description* of objects. The description must be evaluated in order to determine its *referent*, which is a history object. One problem in using temporal descriptions is that the referent of the description might change as we gain knowledge about the situation. For example, if we are told only that the referent of "Shale.top.height@**t0**" is 25.0 and that **t1** postdates **t0**, we would assume that "Shale.top.height@**t1**" is also 25.0. If we later found out that uplift occurred between **t0** and **t1**, we would infer that the referent of the description "Shale.top.height@**t1**" has changed. We would like our system to make such assumptions of constancy, but to update any consequences if the assumption is later contradicted when knowledge is added to the system.

In order for the system to automatically maintain a consistent view of the world while incrementally adding knowledge about changes, each temporal description in the knowledge base is associated both with its referent and with all statements which include the description. Also, each quiescent interval in a history time-line has a list of the descriptions which reference the value within the interval. If the interval changes due to the insertion of a dynamic interval, all descriptions which now refer to a different object are notified and any statements which include that description are re-evaluated. For example, suppose both "Shale.top.height@**t0**" and "Shale.top.height@**t1**" refer to the quantity to 25.0. Then the statement "Shale.top.height@**t0** < Shale.top.height@**t1**" would be false. However, if we later found out that there was uplift between **t0** and **t1**, "Shale.top.height@**t1**" would refer to a different value, the statement "Shale.top.height@**t0** < Shale.top.height@**t1**" would become true and logical formulae that

depended on that statement would have to be re-evaluated. Likewise, the value of the arithmetic expression "Shale.top.height@t1 + 6.0" would have to be recalculated. The use of a dependency directed backtracking package like RUP [McAllester] supplies the necessary machinery to perform this bookkeeping.

### 4.1.3 Appropriateness of histories for temporal reasoning

The history representation was designed to facilitate reasoning about the accumulated effects of changes caused by processes. There are four features of histories that facilitate reasoning about change.

1. Temporal Objects: Objects have an associated life-span. We can reason about when they were created or destroyed without having to delete the object from the knowledge base. Also, the system assumes that an object continues to exist (persists) unless explicitly told otherwise. This facilitates reasoning about which objects were affected by global processes, such as tilting, and enables us to ascertain the relative ages of formations.

2. Change is Additive: Change to an attribute of an object is represented by adding a dynamic interval to the time-line of the attribute. In addition, associated with each dynamic interval is the event which caused the change and the "change equation" relating the values before and after the event occurred. Thus, we always have a complete record of the changes and can reason about the cumulative effect of changes to the attribute.

   A major problem with our implementation of histories is that currently it cannot currently handle simultaneous interacting processes, that is, processes which change the same attribute at the same time. Although this is not a common occurrence in geologic interpretation, it is common in the physical world. In Chapter 7, we speculate on how to augment our representation to handle this problem.

3. Temporal Descriptions: In order to use the history representation, we have developed the @-notation, a flexible notation for describing temporal objects and their attributes. The @-notation provides both a syntax for describing objects and a semantics for interpreting such descriptions.

4. Frame Problem: The frame problem [McCarthy] involves inferring what things do *not* change when an event occurs. A typical solution in temporal logic is to define frame axioms for each process which state what does not change. We deal with the issue by implicitly assuming that the value within a quiescent interval remains constant and that the qualitative process representations completely describe all the relevant changes to the objects (see Section 5.2). This solution is "non-monotonic" in that when the assumption of constancy proves false, that is, we find out that something did in fact change during an interval, the system automatically updates the referents of the temporal descriptions which depended on that change.

Although this particular history-based representation was developed for use in geology (Appendix A lists the geologic object descriptions used by the system), the ideas can be used to represent change over time in many other domains. Other research (e.g. [Forbus, 1982], [Hayes], [Shapiro], [Tsotsos]) has incorporated similar ideas into temporal object representations for domains as diverse as medicine and programming.

## 4.2 Quantitative Objects - Diagrams

### 4.2.1 Criteria

In this section, we examine our quantitative representation of objects which is based on the notion of diagrams. The major criterion for this representation is to represent shape and spatial knowledge accurately and to enable it to be accessed and manipulated efficiently. Our history-based representation is useful for dealing with certain types of change, essentially characterized as one-dimensional. For example, it is easy to use our

history-based representation to describe that the height of a formation will increase if the formation undergoes uplift. However, many geologic effects 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" when a formation is tilted. To facilitate reasoning about these types of change, we have developed methods for representing, reasoning about and manipulating diagrams.

The use of diagrams is important in three of the four stages of imagining -- matching, quantitative simulation and parameter determination -- and criteria for the representation are somewhat different for each stage.

First, for purposes of matching, shape and adjacency information is essential. We need to determine the orientation of boundaries between rock-units and to find out which rock-units are adjacent. Second, the quantitative simulation technique needs to modify diagrams efficiently. This entails operations to add and erase lines (to represent erosion, deposition and intrusion), move one part of the diagram relative to another (to represent faulting) and to change the coordinate position of points (to represent uplift, subsidence and tilting). Third, to do parameter determination we need to make measurements such as the thickness of a rock-unit, the orientation of a boundary and the height of a point.

We also need a way of interpreting these diagrams. A diagram is just a collection of lines. We need to associate diagram features with geologic objects in order to make geologic sense out of a diagram.

### 4.2.2 Description

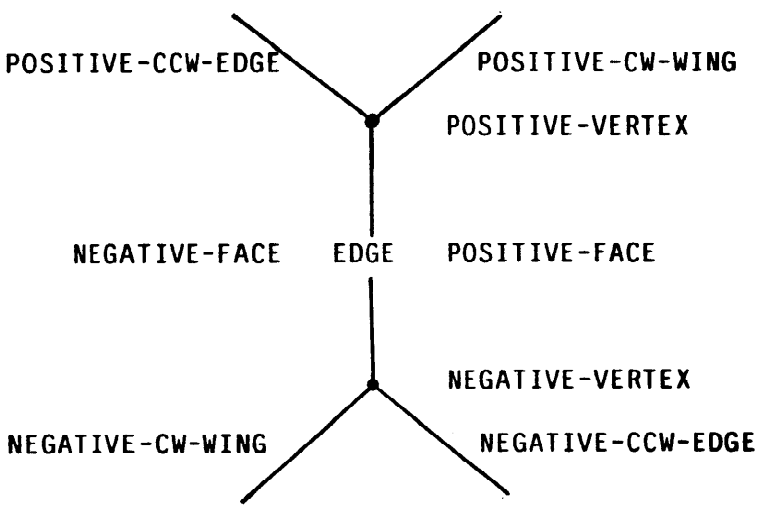
#### 4.2.2.1 The wing-edge structure

All of these requirements for a diagram representation are met by the wing-edge structure [Baumgart]. This representation, originally designed for three-dimensional modeling for computer vision, has been adapted here for representing two-dimensional diagrams.

The primitive objects in this representation are *vertices*, *edges*, and *faces*. A vertex includes 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 15. Each edge has pointers to exactly two faces, two vertices and four "wings", which are the edges that share a common vertex and a common face. 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, such as "above" or "adjacent-to".

---

**Fig. 15. The Wing-Edge Representation of an Edge**



Geometric properties are also easily represented and retrieved. The representation is quite simple -- vertices are associated with an (X,Y) coordinate position. Retrieval is efficient since the properties needed are either explicitly represented or can be easily calculated. For example, since an edge points directly to its two end-points, the length and orientation of an edge can be readily computed from the coordinates of its end-points.

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 allows easy computation of the spatial relationships (such as "above") and metric properties (such as "angle of slope") that are needed to do geologic interpretation.

Third, the wing-edge representation was designed to facilitate manipulation of geometric structures, which makes it easy to do the diagrammatic simulation of processes. For imagining this is its most important feature. The manipulations are accomplished by using a set of low-level operations called "Euler operations"[10] [Baumgart] which modify the topology of wing-edge structures. The set of Euler operations includes functions for adding and erasing edges, splitting a face or edge into two parts and merging two faces or two edges into one. These operations are extremely efficient because they involve only local changes to the wing-edge structure. For example, splitting a face involves only changes to the face itself and to the edges on its perimeter. Thus, the complexity of simulating an event does not depend on the size of the rest of the diagram. This is exactly what one wants from a spatial representation -- local spatial changes involve only local representational changes and local spatial queries involve only local search in the representation.

---

10. They are named "Euler operations" because they preserve the "Euler number" in the diagram -- that is, Faces + Vertices - Edges = 2

#### 4.2.2.2 Diagram/history interface

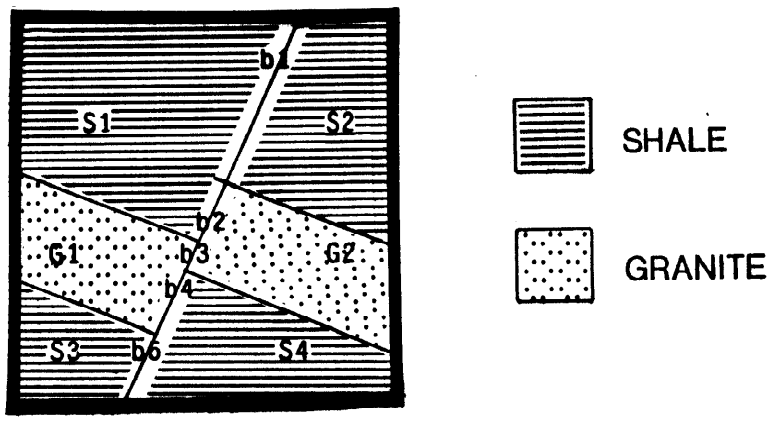
A diagram is merely a collection of geometric edges, faces and vertices. It contains no reference to geologic terms. In order to represent the spatial aspects of geology, we must interpret diagrams by setting up correspondences between features in the diagram and geologic objects. We can place a real-world interpretation on the diagram through the use of a simple and clearly defined diagram/history interface.

Basically, the interface consists of a one-to-one mapping between primitive elements in each domain. A diagram corresponds to the world at a particular instant of 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. In addition, several mappings are defined from spatial relations in the diagram to the corresponding relations in the geologic world. For instance, determining if a rock-unit is above another involves seeing if the face in the diagram corresponding to the rock-unit is above the other face.

Often, we need to associate collections of faces with a single rock-unit and collections of edges with a single boundary. For example, in Figure 16 the shale rock-unit is represented by faces S1, S2, S3 and S4 and the granite rock-unit is represented by faces G1 and G2. In

---

**Fig. 16. Relations Between Diagram and Geologic Features**

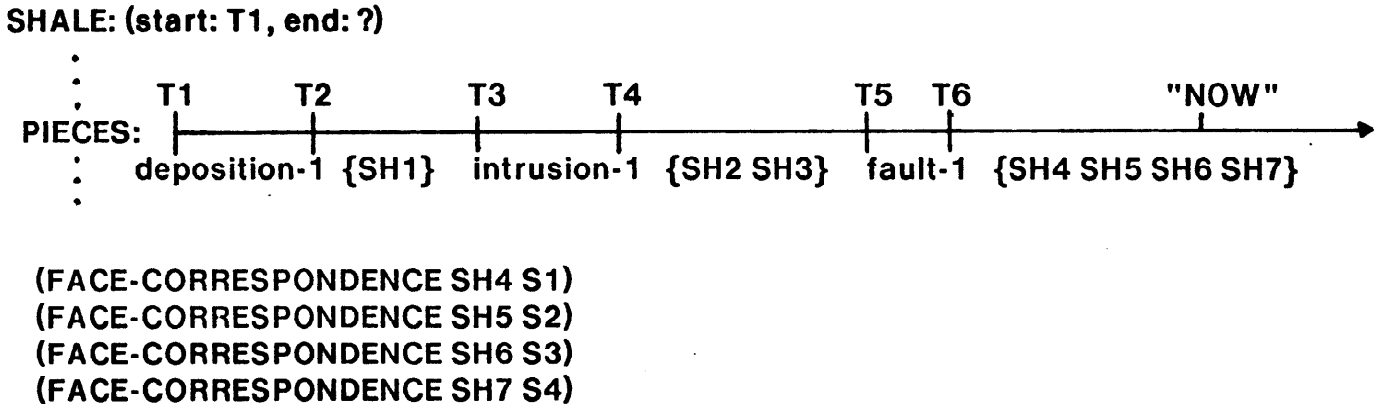


addition, the fault boundary is represented by the edges b1, b2, b3, b4 and b5. Sometimes we need to associate a single feature in the diagram with multiple geologic features. For example, if we assert that the collection of faces S1, G1 and S3 represents the *down-thrown block* of the fault, then face S1 corresponds to both the shale rock-unit and the down-thrown block.

The difficulty is that these needs are not consistent with a one-to-one mapping. However, we have devised a solution which, though not generally adequate, suffices for the imagining task. The simplicity of the diagram/history interface is maintained by moving the complexity to the qualitative object representation. Each object has an attribute named "pieces" which is a set of objects, each of the same type as the parent object. This attribute enables us to represent and reason about objects made up of smaller pieces of the same type.[11] For example, Figure 17 presents the representation of the shale rock-unit shown in Figure 16, along with the correspondences between the diagram and the histories. Notice that the "pieces" attribute, like all other object attributes, is a time-line. Thus we can reason about how objects fracture or consolidate over time.

---

**Fig. 17. Representation of the Pieces of Shale**




---

11. Since this definition is recursive, we can have a tree of pieces to any depth.

The use of "pieces" slightly complicates the operations on diagrams. When a face or edge is split into two pieces, the corresponding geologic features must be changed to reflect that split. For example, when faulting occurs a new dynamic interval would be added to the "pieces" attribute of each rock-unit, indicating that they split (see Figure 17). Thus, for each of the Euler operations which change a diagram feature, we have defined "maintenance operations" that change the corresponding "pieces" of the qualitative objects to reflect the changes in the diagram.

The major problem with this use of pieces, and one which is still unresolved, is how to inherit attributes from the parent object. For example, the composition of a piece of a rock-unit is the same as its parent, but its thickness is not. Currently we handle this on an *ad hoc* basis. Any function which retrieves the value of an attribute must know whether to get the information from the object itself or its parent. This is not a very clean solution in general, but it is adequate for our purposes.

#### 4.2.2.3 Establishing Associations

One of the main tasks in doing parameter determination is to "establish associations" - associating quantitative values in the goal diagram with qualitative attributes of geologic objects. "Establishing associations" refers to two related tasks. First, it refers to our common notion of making measurements using ruler or protractor, as in finding "the orientation of the fault" or "the thickness of the shale". Second, it refers to establishing the diagram/history correspondence between a given object and some diagram feature. For example, "establishing an association with the top of the fault" means finding the vertex in the diagram that corresponds to the geologic point which is the top of the fault.

Note that establishing an association is not the same as determining the referent of a description. The referent of "fault.top@NOW" is a geologic point, while the "association" of "fault.top@NOW" is a vertex in the diagram which corresponds to the geologic point. However, just as the description indicates which history time-lines must be searched in

order to determine a referent, it also indicates how the association is to be established. For example, the method of establishing an association for "fault.top@**NOW**" is :

> "Use the goal diagram, which corresponds to the time 'NOW'. Find all the edges in the goal diagram which correspond to the 'fault' boundary. Then find the end-point of all the edges which has the maximum Y-coordinate (the 'top')."

Given a description using the @-notation, the general method for establishing an association is:

1. The temporal part of the description ("NOW") indicates in which diagram the association is to be made.
2. The last attribute of the description ("top") indicates the operation to be employed for making the association.
3. The referent of the beginning part of the description ("fault") is the object to be operated upon.

We have defined a minimum number of association functions in order to test the robustness of the parameter determination technique (see Section 3.3.2.). The rest of this section presents the algorithms for establishing associations. [12]

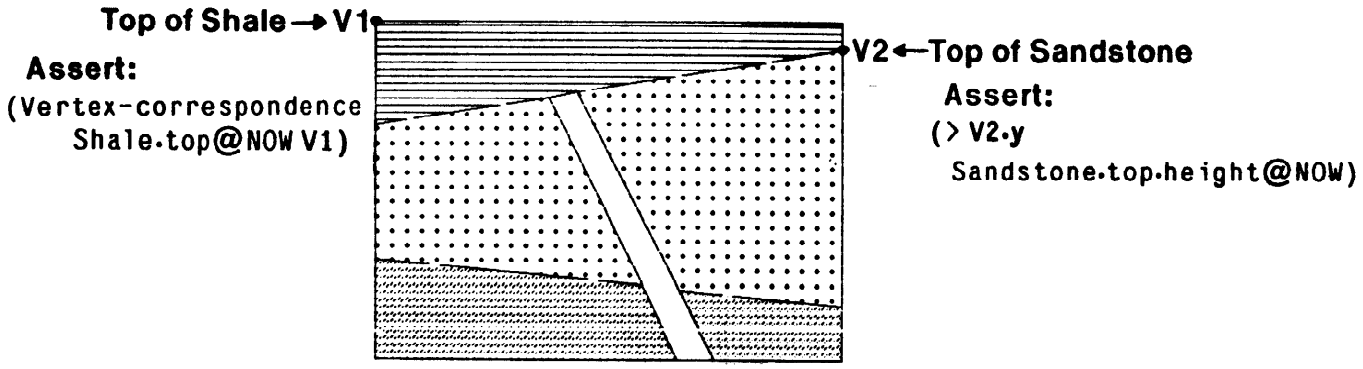
1. Top or Bottom Point of a Boundary --

   (This is the algorithm for finding the top point. The bottom can be determined similarly). Find the end-point with the maximum Y-coordinate of all the edges corresponding to the boundary. If that vertex abuts the outside face of the diagram and the boundary is sloping up at that point, assume that the actual top lies outside the diagram and just assert that the height of the top is greater than the maximum Y-coordinate found. Otherwise create a correspondence between the geologic point and the vertex (see Figure 18).

---

12. All algorithms assume that the appropriate diagram to use has already been determined.

Fig. 18. Measuring the Top and Bottom of a Boundary




---

2. Height or Lateral of a Geologic Point --

   Find the vertex corresponding to the point. If one exists, then the height is its Y-coordinate and the lateral is its X-coordinate.

3. Orientation of a Boundary --

   Find all the edges corresponding to the boundary. The boundary orientation is:

$$\frac{\sum_{e \in \text{edges}} \text{length}(e) * \text{orientation}(e)}{\sum_{e \in \text{edges}} \text{length}(e)}$$

4. Slip of a Fault --

   Find all the edges corresponding to the fault. For any other edge sharing a common vertex (i.e., crossing the fault), find an edge on the other side of the fault line whose faces above and below the edge correspond to the same rock-units as those faces of the initial edge (see Figure 19). The slip is the distance between the two vertices which are common to those edges and the fault edges.

5. Location of a Boundary -- The "location" is an abstraction of the spatial position of an object. Commonly, the location of an object is used rather imprecisely. For example, we talk about a rock-unit being 500 meters below sea-level, but we typically do not indicate if this refers to the height of the top, bottom, center of mass, etc. We

Fig. 19. Measuring the Slip of a Fault

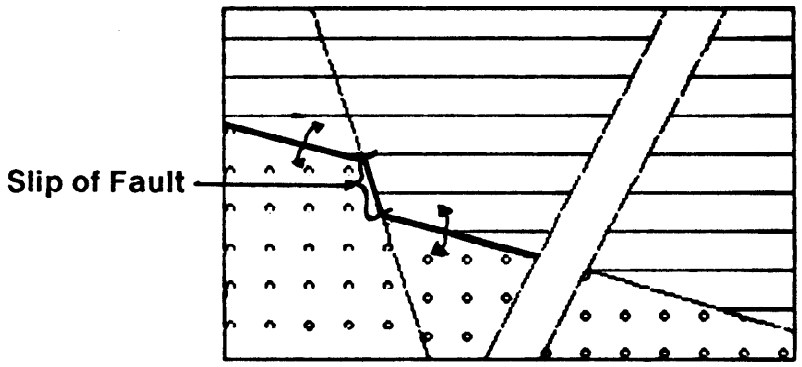
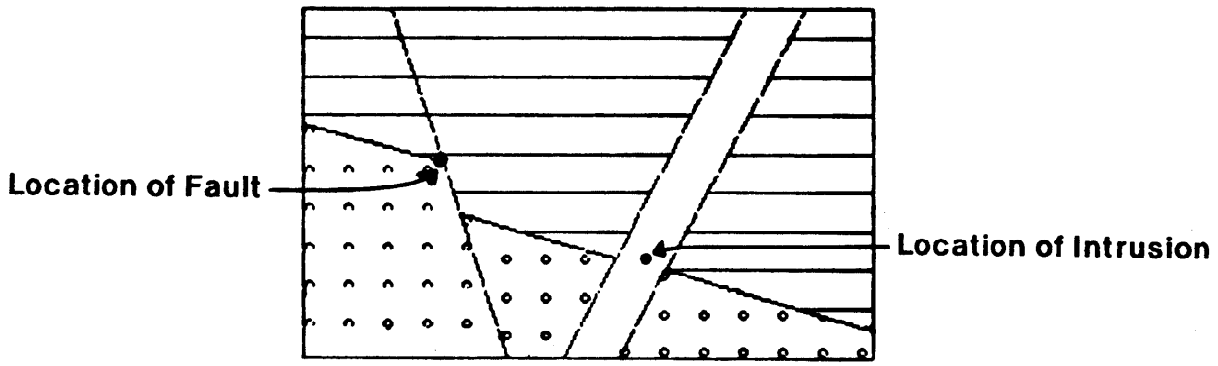



Fig. 20. Measuring the Location of a Boundary




have introduced the concept of "location" in an attempt to firm up this imprecise notion.

We define the location of a continuous boundary (like a fault) to be some arbitrary end-point of the edges corresponding to that boundary (see Figure 20). For a split boundary, such as the two sides of an intrusion, the location is defined to lie somewhere along the center-line of the edges (see Figure 20).

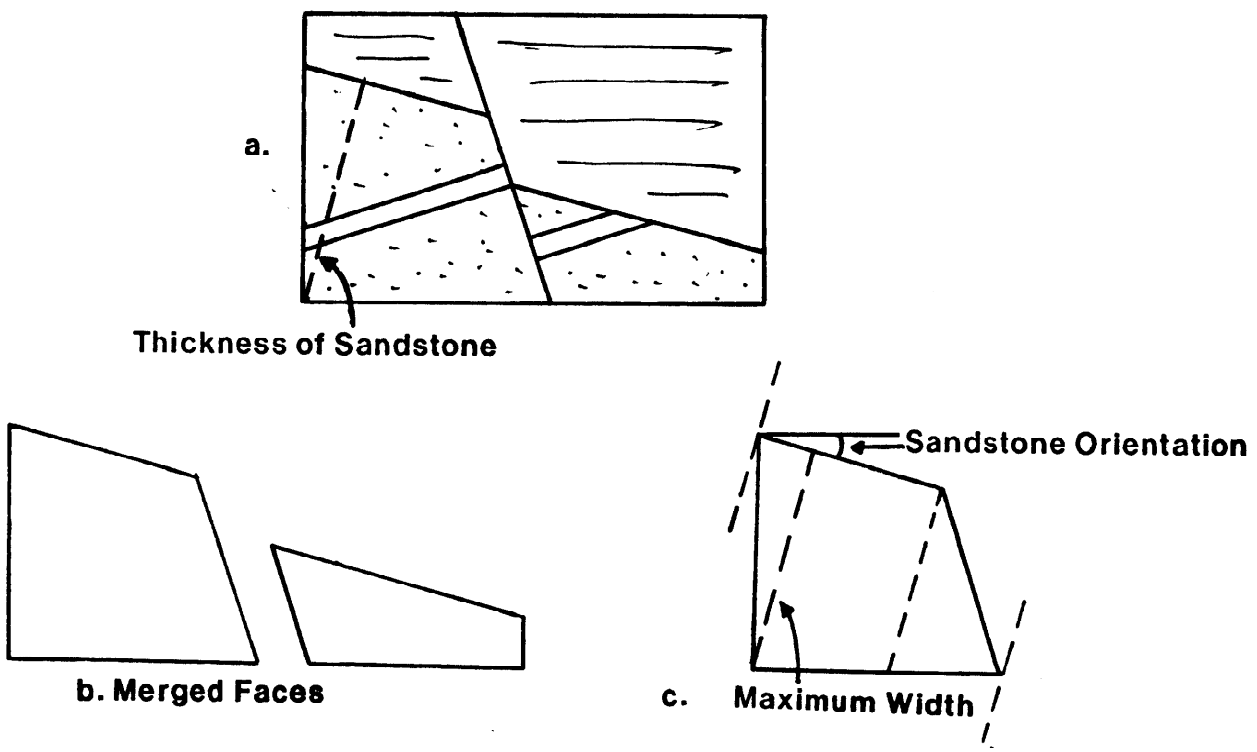
6. Thickness of a Rock-Unit --

This is the most complex measurement we make. The thickness of a rock-unit is defined as the maximum length along a line perpendicular to the orientation of the

rock-unit. First, we find all the faces corresponding to the rock-unit. Second, we use the parameter determination technique to find the orientation of the rock-unit. Third, we must "merge faces" across intrusions. This is because while the rock-unit may be cut by an intrusion, its maximum thickness may lie across both pieces of the rock-unit (see Figure 21a). We create dummy faces which include all the faces of the rock-unit which are separated by intrusions (see Figure 21b). For each of these faces, we find the maximum width along a line perpendicular to the orientation of the rock-unit. Basically, this is done by constructing a line from a vertex on the perimeter of the face perpendicular to the face's orientation. If the line intersects an edge of the perimeter, then the distance from the vertex to the intersection point defines the width from that vertex (see Figure 21c). The maximum width of all vertices on the face's perimeter is the width of that face. The maximum width of all

---

**Fig. 21. Measuring the Thickness of a Rock-Unit**



the faces is the thickness of the rock-unit.

One more correction must be made. If the measurement was made to the edge of the diagram (see Figure 21a), we assume that the actual thickness of the rock-unit is at least the maximum measured width and so the thickness is asserted to be greater than or equal to the maximum measured width.

### 4.2.3 Appropriateness of diagrams for spatial reasoning

There are several important reasons for using the wing-edge structure.

1. The wing-edge structure is efficient in terms of ease of reference and manipulation, that is, queries or constructions can be done with relatively little computational effort. As previously illustrated, the connections in the wing-edge structure make adjacency and geometric properties either explicit or easy to calculate. Topological knowledge is encoded in the wing-edge pointers. Geometric knowledge is encoded in the coordinates associated with vertices and in the pointers between edges and vertices or faces. In addition, the Euler operations provide a means of making local changes to the diagram by making local changes in the wing-edge structure.
2. The wing-edge structure is a compact representation because, unlike the other diagram representations, (such as the arrays used in [Funt]), a diagram can be encoded with relatively few symbols. In particular, the size of the description depends on the complexity of the diagram and not on its size.

   Note that compactness and ease of reference are often at odds with one another. That is, in order to represent some information compactly, referencing it is often computationally more difficult. An advantage of the wing-edge structure is that spatial relationships can be both compactly represented and easily retrieved.

3. The diagram primitives -- vertices, edges and faces -- correspond directly with the geologic objects that we need to describe -- geologic points, boundaries and rock-units. The idea that the diagram (spatial) vocabulary should match the real-world object vocabulary was also expressed in [Forbus, 1981], and stands in contrast to other diagram representations used to model the world, such as arrays [Funt] or quad-trees [Hunter]. In these representations, the real-world interpretation of diagrams is much more complicated than the simple one-to-one correspondence of our diagram/history interface.

Also, the insistence that diagrams contain no geologic information and that the diagram/history interface be kept simple has led to very modular representations. The diagram module deals with space, the history module deals with geology and time. This modularity simplifies the work needed to implement the representations and enables us to make modifications to one representation without worrying about the other.

### 4.3 Quantity Lattice

The quantity lattice bridges the qualitative/quantitative representational boundary in order to perform numeric reasoning. As with the history and diagram representations, the needs of the task constrain the choice of representation.

#### 4.3.1 Criteria

The quantity lattice is used for two related tasks, both of which involve determining ordering relations between entities.

One task is to do temporal reasoning. Recall that the value of an attribute at a particular point in time is selected by searching the time-line for the interval that contains that time point. This involves determining the ordering between points in time, which may be only partially ordered due to our incomplete knowledge of the situation.

The other task is to perform arithmetic reasoning. In doing parameter determination we often need to evaluate change equations such as "shale.top.height@t1 + uplift-amount". Although we can measure "shale.top.height@t1" exactly, the best we can say about "uplift-amount" is that it is positive. Still, we need to evaluate the expression in order to get a result that we can reason about. In particular, we must be able to infer that "shale.top.height@t1 + uplift-amount" is greater than "shale.top.height@t1". We would also like the representation to exhibit the characteristic that if the numeric value of a term is known more precisely, then the value of arithmetic expressions containing the term can be calculated more precisely and that ordering relations involving the term can be determined more efficiently.

### 4.3.2 Description

To handle these reasoning tasks, we have developed the *quantity lattice*, in which the value of a numeric quantity is represented in terms of their relationships with other quantities and real numbers. The value of a quantity is assumed to be a real number, but the actual value is typically not known to us. 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 the values of T1 and T3. This is the notion of quantity as described in [Forbus, 1982]. 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 > 1.0" and "T2 $\leq$ 100.0".

To represent the relationships among quantities, we maintain a lattice of partial orders. When we assert an ordering relationship between two quantities, a link is added between the two objects 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.[13]

Sometimes, however, the value of a quantity can be determined without searching the lattice. For example, suppose we assert that "**B** $\leq$ 1.1" and "**A** > 3.25". From this we can conclude directly that **B** < **A**. We would like the quantity lattice to conclude the same fact without explicitly recording that 1.1 < 3.25. This reasoning is accomplished by associating a real-valued interval with each quantity. The value of the quantity is constrained to lie somewhere within the range of the interval. This interval provides an efficient means of determining the relationship between two quantities. If the 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 $(-\infty,1.1]$[14] and similarly **A** is associated with the interval $(3.25,\infty)$. Then we can easily determine that **B** < **A**. To maintain these intervals when we assert an ordering relationship between quantities, the system checks to see if the range of one of the quantities can be constrained by the asserted ordering and the range of the other quantity. For example, suppose **C** and **D** are quantities, where 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.

The major advantage in using the real-valued intervals is in doing arithmetic. Using these intervals, the more precisely the values of the quantities are known, the more precisely the value of the arithmetic expression can be computed. For example, if all we know is that **A** > **B**, we can infer 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

---

13. Actually, the search is a bit more complicated, since a relationship like "$\leq$" can be found by following a path consisting of a combination of " = " and "$\leq$" links.

14. A parenthesis indicates an open interval, a bracket indicates a closed interval.

compute that **B** + **B** lies within the interval [0,2] and can infer that **A** > **B** + **B**.

The value of an arithmetic expression is computed using "interval arithmetic". The interval range of the expression is determined by applying the arithmetic operator to the end points of the terms of the expression. For example, "[3,6) + [-1,5]" yields [2,11). Figure 22 presents the rules for doing the interval arithmetic.

One difficulty with this approach is that the quantity lattice is a dynamic system. The interval ranges of quantities are constantly being narrowed by the incremental addition of ordering relationships. Thus, after an arithmetic expression is initially evaluated the values of its components might become known more precisely. The quantity lattice takes care of this problem by maintaining dependencies between the value of an arithmetic expression and its terms. When the interval range of a quantity is changed, expressions which depend on it are

---

**Fig. 22. Interval Arithmetic Operators**

```
(+ [X_l, X_u] [Y_l, Y_u]) ⇒ [(+ X_l Y_l), (+ X_u Y_u)]

(- [X_l, X_u] [Y_l, Y_u]) ⇒ [(- X_l Y_u), (- X_u Y_l)]

(* [X_l, X_u] [Y_l, Y_u]) ⇒ [(min (* X_l Y_l) (* X_l Y_u) (* X_u Y_l) (* X_u Y_u)),
                             (max (* X_l Y_l) (* X_l Y_u) (* X_u Y_l) (* X_u Y_u))]

(/ [X_l, X_u] [Y_l, Y_u]) ⇒ if (AND (< Y_l 0.0) (> Y_u 0.0))
                             then (-∞, ∞)
                            else [(min (/ X_l Y_l) (/ X_l Y_u) (/ X_u Y_l) (/ X_u Y_u)),
                                  (max (/ X_l Y_l) (/ X_l Y_u) (/ X_u Y_l) (/ X_u Y_u))]

(- [X_l, X_u]) ⇒ [(- X_u), (- X_l)]
```

(Notice that the ranges for the trigonometric functions are much wider than need be. These definitions are used only for ease of implementation; if it were necessary, more precise definitions could be specified. Also notice the singular situation in division when the interval range crosses zero.)

```
(sin [X_l, X_u]) ⇒ if (= X_l X_u) then [(sin X_l), (sin X_u)] else [-1.0, 1.0]

(cos [X_l, X_u]) ⇒ if (= X_l X_u) then [(cos X_l), (cos X_u)] else [-1.0, 1.0]

(tan [X_l, X_u]) ⇒ if (= X_l X_u) then [(tan X_l), (tan X_u)] else [-1.0, 1.0]
```

automatically recomputed.

A more serious difficulty with this approach stems from shortcomings inherent in interval arithmetic. Interval arithmetic will often result in intervals which are larger than common-sense would dictate. For example, suppose we know that "**A** > **B**", **B** $\in$ [0,1), and **A** $\in$ (0,1]. Using interval arithmetic we compute that "**A** - **B**" $\in$ (-1,1], but we should be able to infer that "**A** - **B**" $\in$ (0,1] since **A** is greater than **B**. The problem is even clearer when we realize that by using interval arithmetic we can never determine that "**A** - **A**" is zero unless the value of **A** has been determined exactly.[15]

We have compensated for this deficiency in interval arithmetic by combining it with an arithmetic technique based on ordering relationships. For each arithmetic operator axioms are defined which relate the value of the arithmetic expression to orderings between the terms of the expression. Figure 23 presents these axioms for the four basic arithmetic operators. Using the above example, we can deduce from these axioms that **A** > **B** $\Rightarrow$ **A**-**B** > **0**. Thus, the relational arithmetic infers that "**A** - **B**" $\in$ (0,$\infty$), while the interval arithmetic infers that "**A** - **B**" $\in$ (-1,1]. Due to the automatic narrowing of interval ranges, our system would compute the intersection of these ranges, (0,1], which is, in fact, the common-sense answer.

When we discover that objects have changed, we often must retract certain assumptions about the world (see, for example, Section 4.1.2.2). Since some of these assumptions involve ordering relationships between quantities, we need a retraction mechanism to remove orderings from the quantity lattice.

---

15. Thanks to Gerry Sussman for pointing this out to me.

Fig. 23. Relational Arithmetic Axioms

```
(FOR-ALL (X,Y,REL) (AND (IFF (REL X 0.0) (REL (+ X Y) Y))
                        (IFF (REL Y 0.0) (REL (+ X Y) X))))

(FOR-ALL (X,Y,REL) (IFF (REL X Y) (REL (- X Y) 0.0)))

(FOR-ALL (X,Y,REL) (AND (⇒ (AND (> X 0.0) (> Y 0.0))
                            (AND (IFF (REL X 1.0) (REL (* X Y) Y))
                                 (IFF (REL Y 1.0) (REL (* X Y) X))))
                        (⇒ (AND (> X 0.0) (< Y 0.0))
                            (AND (IFF (REL X 1.0) (REL Y (* X Y)))
                                 (IFF (REL Y -1.0) (REL (* X Y) (- X)))))
                        (⇒ (AND (< X 0.0) (> Y 0.0))
                            (AND (IFF (REL X -1.0) (REL (* X Y) (- Y)))
                                 (IFF (REL Y 1.0) (REL X (* X Y)))))
                        (⇒ (AND (< X 0.0) (< Y 0.0))
                            (AND (IFF (REL X -1.0) (REL (- Y) (* X Y)))
                                 (IFF (REL Y -1.0) (REL (- X) (* X Y)))))))

(FOR-ALL (X,Y,REL) (AND (⇒ (AND (> X 0.0) (> Y 0.0))
                            (IFF (REL X Y) (REL (/ X Y) 1.0)))
                        (⇒ (AND (> X 0.0) (< Y 0.0))
                            (IFF (REL X (- Y)) (REL -1.0 (/ X Y))))
                        (⇒ (AND (< X 0.0) (> Y 0.0))
                            (IFF (REL X (- Y)) (REL (/ X Y) -1.0)))
                        (⇒ (AND (< X 0.0) (< Y 0.0))
                            (IFF (REL X Y) (REL 1.0 (/ X Y))))))
```

---

The major difficulty in implementing such a retraction mechanism is that if the ordering had been used to narrow interval ranges, then the ranges must be recomputed based on the remaining orderings, and any dependent arithmetic expressions must be recalculated. This expansion of ranges may be propagated throughout the lattice. Unfortunately, the current retraction algorithm used by our system is not totally correct. If cycles exist in the dependency structure among arithmetic expressions (such as "A = B + C" and "C = A - B") then the interval ranges are not expanded properly. We are currently considering alternative algorithms to solve this problem.

# 5. PROCESS REPRESENTATIONS

Simulation involves applying processes to objects. In this chapter, we examine the two representations of processes which were developed to perform the qualitative and quantitative simulations.

## 5.1 Level of Representation

In choosing process representations, we need to consider the level of detail of the physical model and the level of detail of the processes. The required levels of detail place great constraints on the choice of the qualitative and quantitative process representations.

### 5.1.1 Geologic Model

To what level of detail do we need to represent objects in order to do geologic interpretation? Since the task of geologic interpretation is to provide a fairly high-level view of the events which formed the region, we do not need to represent such knowledge as the rate at which deposition occurs or what happens to the rock at the boundary of a molten intrusion. We want only the gross characteristics of the region -- which events occurred and in what order.

A geologic model consistent with this level of detail is the "layer cake" model (see, for example, [Friedman]). This model assumes that deposition occurs uniformly, forming deposits which are always flat on top and stack up like the layers of a cake. We treat formations as homogeneous units of rock, with the "rock-unit" as our basic geologic component. Erosion is also assumed to be uniform, removing existing rock as if a knife sliced horizontally through the region. A fault is modeled as a clean, straight break, with no distortion occurring to the neighboring rock-units. Tilting, uplift and subsidence are assumed to affect the region uniformly, causing no shape distortions. Finally, the model of intrusion used is that it replaces the material intruded through, as if the molten material

melts away the existing rocks.

The "layer cake" model is a good first approximation and was used by most geologists until the mid-1800's. In reality, deposition and erosion follow the contours of the Earth's surface, Earth movements are not uniform and intrusion mostly spreads rocks apart rather than melting them. However, none of these differences affect the topological relations between rock-units, and it is the topological relationships, rather than internal shape and composition, which are of primary importance here. Hence, the "layer cake" model is sufficient for our task.

### 5.1.2 Process Model

A basic question in choosing a model for processes is whether to use a discrete or continuous model. The discrete ("end-point") model determines the state of the world at the end of a process given the state at the beginning of the process. It assumes that how the process influences the world during its occurrence is unknown. The continuous model, on the other hand, can model the state of the world at every point in time during the occurrence of the process. For example, a discrete model of uplift is

$$\text{"A.height@}I_{end} = \text{A.height@}I_{start} + \text{uplift-amount"}.$$

A continuous model is

$$\text{"A.height@t} = \text{A.height@}I_{start} + (t - I_{start}) * \Delta U\text{"},$$

where $\Delta U$ is the rate of uplift (assumed constant for this model).

Using a discrete model means that, in general, we cannot deal with simultaneous, interacting processes.[16] To do so would require determining the composite effect of the interacting processes, which in general is not possible without knowledge of what happens during the processes (i.e., without a continuous model). However, since most occurrences

---

16. However, we can deal with simultaneous, non-interacting processes.

of geologic processes are non-interacting (although they may be simultaneous), the use of a discrete model has proven sufficient in solving most geologic interpretation problems.

A discrete model is also appropriate because in many cases we do not know what occurs during a complex geologic process. For example, we can determine the final composition of a rock which undergoes metamorphism, given its initial composition, but its composition during the process is not well understood. Hence, in many cases a discrete model is the best that we can do.

Both the qualitative and quantitative process representations use a discrete, "layer cake" model. However, they are represented differently, both because they operate on different object representations (histories and diagrams) and because they have different roles in imagining. The next two sections examine in greater detail the qualitative and quantitative process representations. The final section briefly describes how the two process representations support one another.

## 5.2 Qualitative Processes

### 5.2.1 Criteria

Since the qualitative simulation is done to determine the cumulative effects of a sequence of events, an important criterion for the qualitative process representation is to make explicit which objects are created, destroyed or affected by the process, and the magnitudes of the changes. In addition, in order to describe our "layer cake" model we need to represent constraints on objects, induced by the process, which cannot be represented as changes. For example, due to erosion an already existing point might become the new top of a formation. Although the height of this point is unaffected by the erosion, we know that it is constrained to be equal to the level of the erosion.

We must also be able to reason about physical connections in the world. For example, we know that a boundary is connected to two rock-units. If the rock-units move, then we should infer that the boundary moves.

Finally, we want to describe the effects of processes only incompletely. Since, for example, our qualitative objects have no attributes representing shape, we do not want the process representation to force us to describe changes to the shape of objects. For those processes which are more understood, such as uplift, we would like to represent them more accurately.

### 5.2.2 Description

The criteria presented above point clearly toward a model based on describing a process in terms of its effects, which we call a "causal model". Most previous work in qualitative reasoning about events have used such models, either explicitly or implicitly (e.g., [deKleer, 1979], [Fikes], [Forbus, 1982], [Rieger]), because they facilitate representing and reasoning about the effects of events, a basic task for solving many problems involving change.

Figure 24 represents erosion, using a causal model of processes and a "layer cake" model of geology. (Appendix B lists the qualitative geologic process descriptions used by the system). Roughly translated into English, this description of erosion states that:[17]

1. The surface of the Earth must be above sea-level in order for erosion to occur (Preconditions).
2. The erosion occurs to a level "ELEVEL", which is above sea-level (Parameters; Relations 1, 7).

---

17. The English sentence is followed by the associated collection of statements in Figure 24.

**Fig. 24. Description of the Erosion Process**

```
EROSION
   INTERVAL       I : temporal-interval
   PRECONDITIONS  (> SURFACE.top.height@I_start SEA-LEVEL)
   PARAMETERS     ELEVEL : real
   AFFECTED       C-PART : (set-of rock-unit), C-ALL : (set-of rock-unit),
                   SURFACE
   CREATED        BA : boundary
   EFFECTS        1. (change = BA.side-1 {THE-AIR} I EROSION)
                  2. (change = BA.side-2 C-PART I EROSION)
                  3. (change = BA.orientation 0.0 I EROSION)
                  4. (change = SURFACE.orientation
                             (efn3 ELEVEL SURFACE@I_start) I EROSION)
                  5. (change function SURFACE.top (efn2 ELEVEL) I EROSION)
                  6. (change function SURFACE.bottom (efn2 ELEVEL) I EROSION)
                  7. (change function SURFACE.location (efn2 ELEVEL) I EROSION)
                  8. (for-all c1 ∈ C-PART
                        (and (change function c1.top (efn2 ELEVEL) I EROSION)
                             (change function c1.location (efn2 ELEVEL) I EROSION)
                             (change - c1.thickness
                                (* (efn1 c1 ELEVEL) (- c1.top.height@I_start ELEVEL))
                                I EROSION))
                  9. (for-all c2 ∈ C-ALL (change = c2 ⊥ I EROSION))
   RELATIONS      1. (> ELEVEL SEA-LEVEL)
                  2. (= C-PART {r : rock-unit |
                                 (and (exists-at r I_start)
                                      (≥ r.top.height@I_start ELEVEL)
                                      (< r.bottom.height@I_start ELEVEL))})
                  3. (= C-ALL {r : rock-unit |
                                (and (exists-at r I_start)
                                     (≥ r.bottom.height@I_start ELEVEL))})
                  4. (⇒ (≥ SURFACE.bottom.height@I_start ELEVEL)
                        (= SURFACE.bottom.height@I_end SURFACE.top.height@I_end))
                  5. (= SURFACE.top.height@I_end ELEVEL)
                  6. (≤ SURFACE.bottom.height@I_end SURFACE.bottom.height@I_start)
                  7. (for-all c1 ∈ C-PART
                        (and (= c1.top.height@I_end ELEVEL)
                             (≥ (efn1 c1 ELEVEL) 0.0)
                             (≤ (efn1 c1 ELEVEL) 1.0)))
```

3. An erosional boundary "BA" is created by the process; this boundary occurs at the intersection of "the-air" and those rock-units which are partially eroded, and its orientation is horizontal (Affected; Effects 1, 2, 3).

4. The top, bottom and location points of the surface of the Earth are affected by the erosion, and the orientation of the surface changes (Effects 4, 5, 6, 7).

5. If the bottom of a rock-unit is below ELEVEL and its top is above ELEVEL, then it is partially eroded. In particular, the top point changes to a point whose height is ELEVEL and the thickness of the rock-unit decreases (Effects 8; Relations 2).

6. If the bottom of the eroded rock-unit is above ELEVEL, then it is totally eroded away (destroyed) (Effects 9; Relations 3).

7. The top point of the surface of the Earth changes to a point whose height is ELEVEL. If the bottom of the surface before the erosion is higher than ELEVEL, then the top and bottom points of the surface have the same height after the erosion (Effects 5; Relations 4, 5).

In general our qualitative process represents have the following form:

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

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

3. `PARAMETERS` is a set of parameters that indicate the magnitude of the effects of the process.

4. `AFFECTED` is a set of the objects that exist at the time the process began and which are changed in some way by the process. An element of this set is either an individual object or a set of objects.

5. CREATED is a set of objects that are created by the process.

6. The EFFECTS field is a set of statements that describe how the process changes attributes of the affected and created objects. A statement is either a "change form" (described below) or a universally quantified statement containing change forms.

7. RELATIONS is a set of statements that are constrained to hold as a result of the occurrence of the process.

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

(CHANGE <type> <attribute> <change> <interval> <cause>).

ATTRIBUTE describes the attribute changed by the process, INTERVAL is when the change occurred and CAUSE is the process that caused the change. TYPE and CHANGE are used to construct the change equations by describing how the values of the attribute at the start and end of the process are related. TYPE can be " = ", in which case after the process occurs the attribute's value equals CHANGE. For example, the form

(CHANGE = BA.orientation 0.0 I EROSION)

represents the fact that after the erosion, the orientation of the erosional boundary is horizontal. The change equation resulting from this change would be[18]

$$BA.orientation@I_{end} = 0.0.$$

TYPE can also be an arithmetic operator (+, -, *, /), in which case the end value is found by applying the operator to the starting value of the attribute and the CHANGE. For example, one of the effects of the uplift is represented by

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

which indicates that the height of a rock-unit after the uplift process equals its height before

---

18. Note that in this case, the value of the attribute at the start of the process does not appear in the change equation. This can occur either because the start and end values are in fact unrelated or because we cannot model their relationship.

the uplift plus the amount of the uplift. The resultant change equation would be

$$\text{A.height@}I_{end} = \text{A.height@}I_{start} + \text{UPLIFT-AMOUNT}.$$

Finally, TYPE can be "function" in which case CHANGE is a function of one argument which is applied to the starting value of the attribute.[19]

Notice the change form "(CHANGE = A ⊥ I EROSION)" in Figure 24 (Effects 9). This form is specially interpreted to mean that **A** does not exist after $I_{end}$. Thus, the process representation can describe both object creation (via the CREATED list) and object destruction (via a CHANGE form).

In addition to describing the effects of processes, we also describe physical connections or *dependencies* between objects, an important part of causal reasoning about change. A dependency encodes the knowledge that changing one object induces a change in the other object. For example, suppose we put a block on a table. Lifting the table causes the height of the table to increase. The block, being connected to the table, also rises (in fact it rises by the same amount as the table). We say that a *dependency* exists between the height of the table and the height of the block. Note that this dependency exists only in one direction, that is, if we lift the block, the height of the table is not affected. It is important to be able to represent such dependencies and to reason about how they are made and broken over time.

We have included a rough notion of dependency in our system.[20] The statement

(FDEPENDS <dep-attr> <attr> <function> <time>)

means that the attribute DEP-ATTR is dependent on the attribute ATTR starting after TIME. Whenever a change to ATTR occurs, we can infer that a change to DEP-ATTR occurs. The magnitude of the change is described by a function of two arguments: the magnitude of the

---

19. The type "function" is the most basic type in that it can be used to define all other types. For example, the "+" type with change Q is equivalent to the "function" type with change (λ (X) (+ X Q)).
20. the "Qprop" statement of [Forbus, 1982] is a more refined use of the same notion.

change to ATTR and the value of DEP-ATTR before the change occurred. This is expressed in the schema

```
(FDEPENDS DA A FN TIME) ⇒
    [(CHANGE TYPE A EFFECT I CAUSE) AND (> I_start TIME)] ⇒
        (CHANGE FUNCTION DA FN(EFFECT) I CAUSE)[21]
```

Since it is often used, we have defined a specialized form of FDEPENDS:

```
(=DEPENDS <dep-attr> <attr> <time>).
```

In =DEPENDS, FUNCTION is implicit - it is always the same as the "type" of the change to ATTR. Thus, the effect on DEP-ATTR (the dependent attribute) is the same as the effect on ATTR. For example, if a process causes an additive change, then the dependent attribute will experience an additive change of the same magnitude. This is the case in the block and table example described above.

Using dependencies we can make many useful inferences about change in the geologic world. For example, by establishing a dependency between the position of a rock-unit and all the geologic points within that rock-unit (such as its top or bottom), we can infer that uplifting or tilting the rock-unit changes the positions of the dependent points. In addition, we can establish an equivalence between two attributes by stating that each is dependent on the other. For example, the orientations of a rock-unit and its boundary are asserted to be equivalent. Thus, tilting the rock-unit implies that its boundary tilts, and tilting the boundary implies that the rock-unit tilts. The dependencies and domain-wide knowledge used by the system are listed in Appendix C.

---

21. FN(EFFECT) is a curried function of one argument which FN(EFFECT) represents the function (λ (X) (FN EFFECT X)).

We have implemented a program to qualitatively simulates an event using these process descriptions. The input to the simulator is a set of qualitative objects representing the world, a causal process description of the form shown in Figure 24 and some additional information used to parameterize the process. An example of this additional information is

"BA.side-2@$I_{end}$ = {Shale Mafic-Igneous}".

This represents the constraint that for this event the erosional boundary must lie along the shale and the mafic-igneous rock-units, that is, they are the rock-units affected by the erosion.

To instantiate an occurrence of erosion (see Figure 24), using the additional information that "BA.side-2@$I_{end}$ = {Shale Mafic-Igneous}", the simulator would:

1. Check that the precondition is true. Does the system believe that the top of the surface of the Earth is above sea-level at the start of the erosion?
2. Create a new object (BA-1, the erosional boundary) and assert that it was created at "I-$7_{end}$", the start of the erosion.
3. Assert that the additional information is true.
4. Update the appropriate time-lines using the CHANGE statements in the EFFECTS field. For example, update the time-line of the orientation of BA-1 by inserting a dynamic interval from I-$7_{start}$ to I-$7_{end}$. For universally quantified statements, the system instantiates the CHANGE forms when bindings for the quantified variables become known. This allows us to add knowledge incrementally to the system about which objects are affected.
5. Assert that all of the RELATIONS shown in Figure 24 now hold. Universally quantified statements are handled as above.

## 5.3 Quantitative Processes

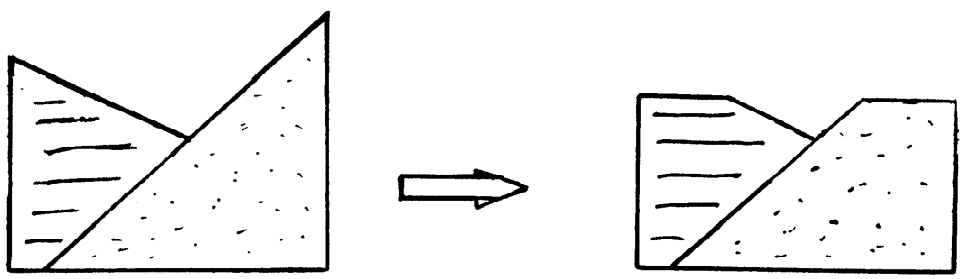
### 5.3.1 Criteria

It is imperative that the quantitative process representations accurately and completely model the spatial effects of the "layer cake" geology, because the results are used directly to determine whether the sequence of events could have formed the region represented by the goal diagram. If the quantitative simulation produced inaccurate results, the matching might fail in cases where the sequence is actually valid or succeed where the sequence is invalid. Getting such false negatives and false positives would be totally unacceptable.

In contrast, many of our *qualitative* process descriptions are not complete or particularly accurate. For example, the description in Figure 24 indicates that one effect of erosion is to change the bottom of the surface of the Earth. Although this is accurate for many geologic situations, Figure 25 illustrates an erosion in which the bottom of the surface remains the same. However, for the examples we have tried so far, these inaccurate inferences have not affected the correctness of the values determined by parameter determination. This is because the parameter determination technique can usually determine the parameter values using other, redundant geologic knowledge contained in the process description, without

---

**Fig. 25. Erosion Above Bottom of Earth's Surface**



relying on the errant change.[22]

### 5.3.2 Description

A quantitative process is represented using an "operational model", which is essentially an algorithm describing how to manipulate diagrams to represent the spatial effects of the process. Our model of erosion, for example, is presented in Figure 26a and Figure 26b shows the effects of executing that algorithm. (Appendix Diagram lists all the quantitative process descriptions used by the system).

The major advantage of an operational model is that the complete set of effects resulting from some occurrence of the process do not have to be known in advance or included in the model. For example, a cake recipe is an operational description of baking. If we follow the recipe, we will get some baked good as the result. However, it might not be a normal cake because the baking powder was too old and so the cake did not rise, the oven was too hot and so the cake burned or due to any other number of exceptions. The point is, since the effects result from the interaction of the initial state and the operational model, the actual set of effects (resulting from all possible initial states) do not have to be explicitly included in the model.

---

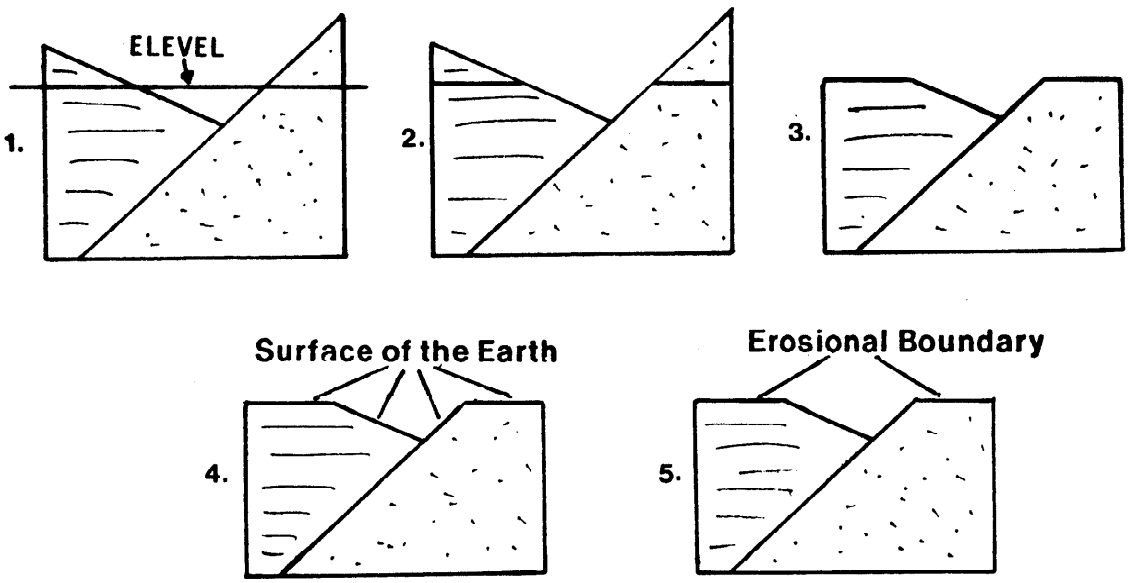
22. The qualitative processes could be made more accurate by conditionalizing the change statements. For example, for erosion we could state

(> SURFACE•bottom•height@$I_{start}$ ELEVEL) ⇒ (CHANGE FUNCTION SURFACE•bottom I EROSION),

so that we infer a change only if the bottom is above "ELEVEL". This formulation has problems stemming from the fact that when the qualitative simulation is done, the truth of the antecedent of the implication is often unknown. Since, in the majority of cases, we can assume that the bottom point does change, we would really like to state "unless we can infer that this exceptional situation holds, assume that . . . ". This extension to our qualitative process representations is discussed in Chapter 7.

Fig. 26. An Algorithm for Simulating Erosion in a Diagram

1. Draw a horizontal line at "ELEVEL".
2. Erase all parts of the line which do not cut across a face corresponding to a rock-unit.
3. Erase all faces above the horizontal line.
4. The edges corresponding to SURFACE, the surface of the Earth, are all the old SURFACE edges which were not erased, plus the remaining edges of the horizontal line.
5. The edges corresponding to the erosional boundary are the remaining edges of the horizontal line.




---

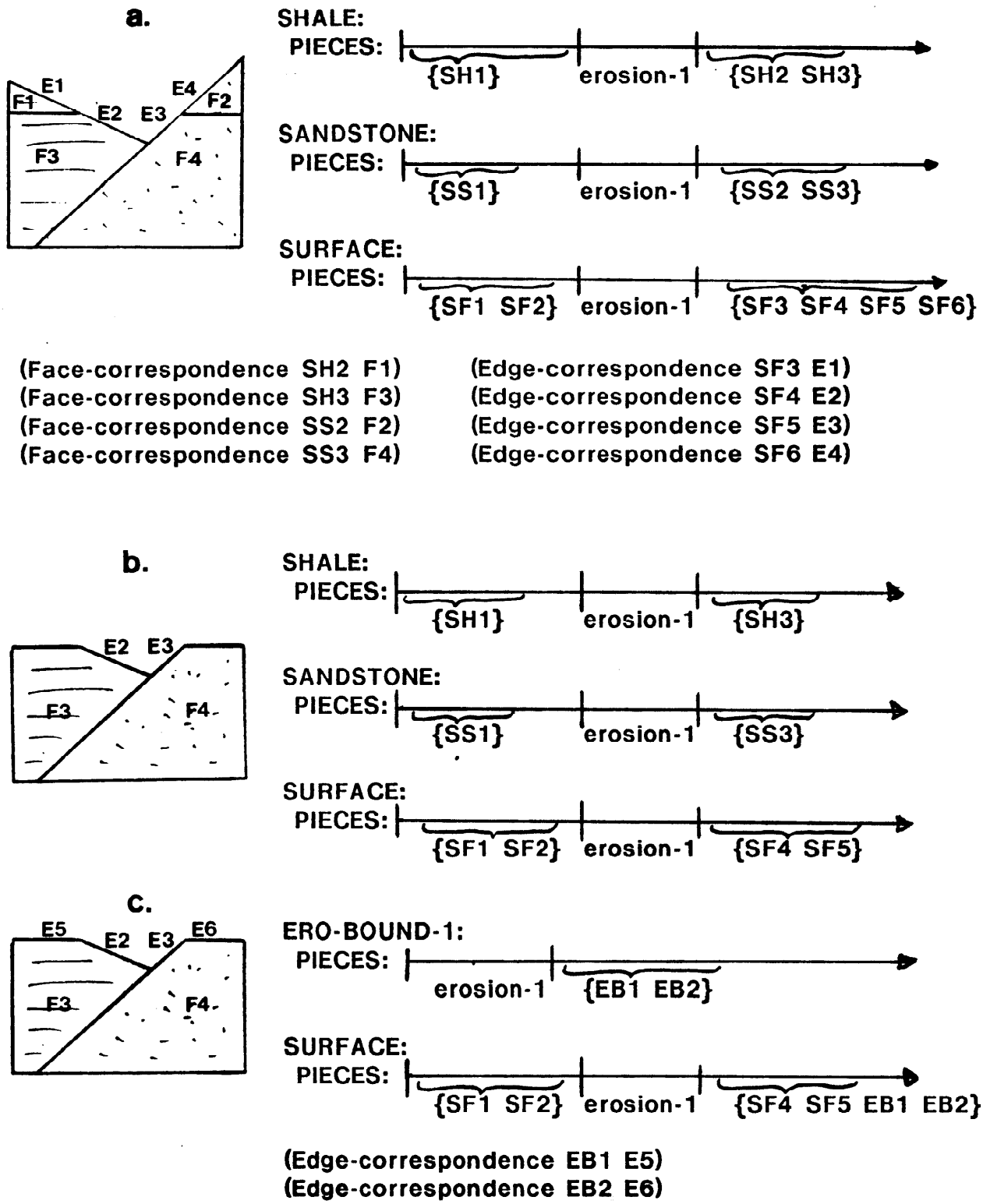
Another advantage of these models is that we can usually come up with relatively concise operational descriptions of processes. Compare, for example, Figure 26 with Figure 24 in Section 5.2.2. This conciseness makes it easier to construct a process description and to check if it is correct and makes it more efficient to simulate the description. Much of this conciseness and efficiency in simulation comes from the fact that control flow (sequentiality and conditionals) can be explicitly and concisely encoded in an operational description.

We have found that a relatively small number of primitive operations are necessary for describing geologic processes as algorithms for modifying diagrams. These include determining relations like: which faces are adjacent to an edge, which face is above an edge and, given a diagram feature, which geologic object does it correspond to. The primitive operations also include diagram manipulation functions to simulate the effects of geologic processes. These functions, which are built out of the Euler operators, are: drawing a line, erasing a face or an edge, adding an edge, splitting a diagram in two and joining two parts of a diagram (the last two are used to simulate faulting). In addition, we need functions for rotating and translating the coordinates of the vertices in the diagram. Some of these operations can be seen in the erosion model (Figure 26), such as "draw a horizontal line at ELEVEL", "erase all faces . . . " and "erase all parts of the line . . . ".

When these diagram manipulations are performed, faces and edges are often split or erased. This changes the correspondence between diagrammatic and geologic objects. In particular, a geologic object may have more or fewer "pieces". Thus, one of the tasks of these manipulation functions is to maintain the diagram/history interface correspondences and to maintain the *piece structure* of the qualitative objects by adding or deleting pieces. For example, in Figure 27a after the horizontal line is drawn both the shale and sandstone rock-units consist of two pieces and the surface of the Earth corresponds to four edges. This is reflected in the "piece" attributes of the objects. In Figure 27b, some faces and edges have been removed; this change is also reflected in the qualitative objects. These "piece maintenance" operations are built on top of the diagram manipulation functions and are invoked whenever the diagram is changed.

Some processes need additional piece maintenance. When a new face or edge is added to the diagram or an existing object changes drastically, the qualitative objects must be updated to reflect the new piece structure. For example, in erosion (see Figure 27c), we must create a piece structure for the newly created erosional boundary and update the piece structure for the surface of the Earth.

Fig. 27. Maintaining the Piece Structure When Manipulating Diagrams



Each process model takes as input a diagram and a list of *quantitative* "process parameters" which must be assigned precise values by the parameter determination technique in order to approximate the actual geologic events. However, to do the parameter determination each quantitative "process parameter" must be associated with terms from the qualitative process descriptions. Thus, for each process we have defined a "quantitative simulation template" which associates the list of quantitative parameters with corresponding qualitative terms. For example, quantitative parameters of erosion are the level of erosion and the erosional boundary. In the qualitative process description, these are referred to as "ELEVEL" AND "BA", and so the simulation template is "(ELEVEL, BA)". The complete list of quantitative simulation templates is presented in Appendix D.

Note that the parameters of the qualitative and quantitative process descriptions need not be the same. We can choose process parameters which are most appropriate for the particular process description. For example, the qualitative parameter for deposition is "DLEVEL", representing the thickness of the material deposited. However, for the quantitative simulation it is more convenient to parameterize deposition in terms of the height of the top of the deposit. This parameter is described as "DLEVEL + SURFACE.bottom.height@$I_{start}$".

Each quantitative process is implemented as a LISP function. To quantitatively simulate an event, the imaginer applies the LISP function to the current diagram and the values obtained by doing parameter determination on the quantitative simulation template. The initial quantitative state is created by a special function which constructs a diagram consisting of a single horizontal line, representing the surface of the Earth.[23]

---

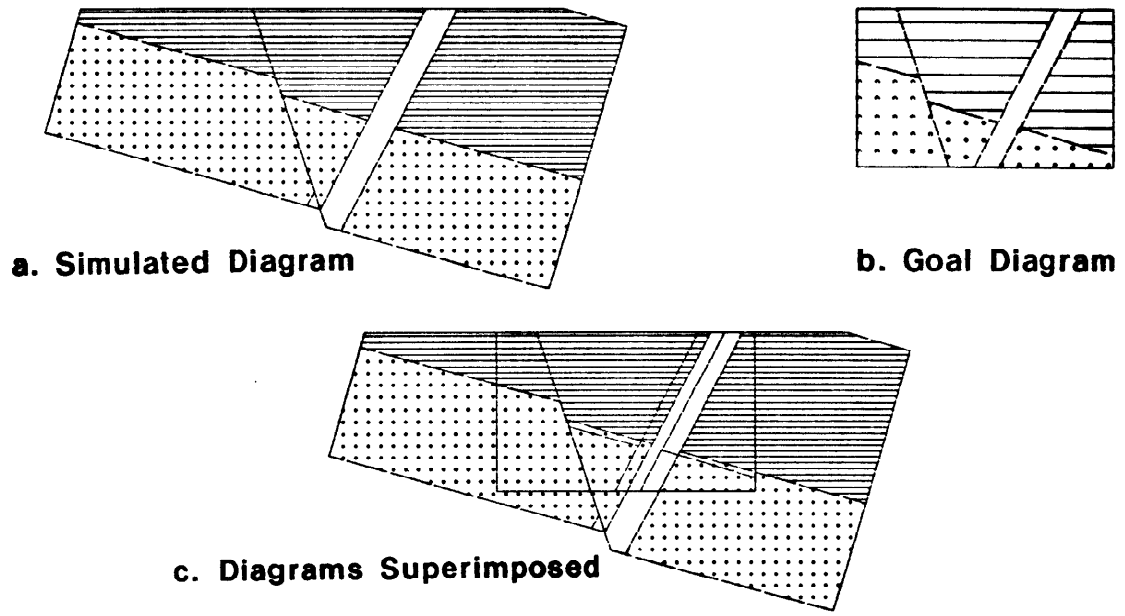
23. The height of this line is "SURFACE.bottom.height@<Step1>$_{start}$", where <Step1> is when the first event of the imagined sequence occurs.

## 5.4 Developing Qualitative Process Descriptions

Qualitative process descriptions are difficult to develop because we need a good understanding of what happens to the world when the process occurs. It turns out that doing the *quantitative* simulation is a helpful tool when developing qualitative descriptions. Since the accumulated changes and process constraints produced by the qualitative simulation are used to determine parameter values, an incomplete process description will give rise to an imprecise parameter value. Using this imprecise value will yield a final simulated diagram which is quantitatively different from the goal diagram. Thus, we can "see" the effects of incomplete qualitative process descriptions. For example, Figure 28a was produced using a process description of faulting which did not include the fact that the rock-units on one side of the fault slide downward. Although the simulated diagram resembles the goal diagram (Figure 28b) topologically, when we superimpose the two (Figure 28c) we see that the incomplete description of faulting caused a slight quantitative inaccuracy in the simulated diagram. In particular, the mafic-igneous intrusion is lower than in the goal diagram because its displacement due to the fault sliding was not corrected for

---

**Fig. 28. Quantitative Simulation Using Incomplete Process Description**



when doing parameter determination.

Although it is not trivial to determine the source of the error from the difference between the two diagrams, the comparison does provide an indication of which parameter is inaccurate. By checking the qualitative sequence of changes for that parameter against our own geologic knowledge of what was supposed to happen, we can usually pinpoint which process description is incomplete and in what ways. By applying this methodology over several geologic interpretation examples, we have greatly refined our models and our understanding of the geologic processes.

Notice that in *using* the system to test the validity of sequences, the qualitative simulation is needed in order to perform the quantitative simulation accurately. However, in *developing* the system the quantitative simulation supports the qualitative by enabling us to "see" the bugs in our qualitative process descriptions.

# 6. RELATED WORK

This report has concentrated on exploring the use of imagining, a simulation technique, as a problem solving tool and on discussing the representations used to support the task. In this chapter, we examine how other researchers have used simulation in problem solving and discuss representations designed to support tasks similar to our own.

## 6.1 The Use of Simulation in Problem Solving

Much of the work on "Naive Physics" has influenced our ideas on simulation as a problem solving tool (particularly [deKleer, 1975] and [Forbus, 1981]) and has influenced our approach to representation of change (particularly [Forbus, 1982] and [Hayes]). Simulation has often been used in problem solving (e.g. [deKleer, 1975], [deKleer, 1979], [Fikes], [Forbus, 1981], [Funt], [Hendrix], [Rieger]). Simulation is typically used to verify a plan (e.g. [Fikes] and our own quantitative simulation), to generate a set of candidate solutions (e.g. [deKleer, 1975], [deKleer, 1979], [Forbus, 1981]), or to predict changes in the world (e.g. [Funt], [Rieger], and our qualitative simulation). All of these use the basic technique presented in Chapter 3 -- they represent objects and events and perform the simulation by applying the event to the current state description (a collection of objects).

One important characteristic of simulation is that it constructs all the intermediate states along the solution path. However, many of the systems employing simulation do not maintain all the changes (as do [deKleer, 1975], [deKleer, 1979], [Forbus, 1981], and our work), but instead erase old values as the simulation progresses (e.g., [Fikes], [Funt], [Rieger]). The results of these simulations cannot be used to reason about the temporal extent of the changes. Thus, such simulations have been used for tasks where only the final character of the world is needed, that is, where it is sufficient for the simulation to indicate *what* happened, rather than *how* it happened. However, for parameter determination and for generating plans (e.g. [deKleer, 1979]) the need to reason about the character of the

changes necessitates maintaining the intermediate states.

## 6.2 Representations to Support Simulation

### 6.2.1 Histories

As noted above, imagining requires us to maintain the intermediate states produced by the simulation. Our history representation (adapted from [Hayes]) is well suited for this task in that it enables us to maintain sequences of changes to attributes and to reason about object creation and destruction. This type of reasoning is necessary for many tasks, and so it is not surprising that representations similar to our history representation have been developed (e.g., [Forbus, 1982], [Hayes], [Shapiro], [Tsotsos]). These representations all maintain the sequences of values resulting from changes, have operators for temporally selecting values at points in time, similar to our @-operator and assume that the value at a time between any two known values can be interpolated. They differ mostly in what types of attributes can be represented (3-space for [Hayes], numeric quantities for [Forbus, 1982], numbers and symbols for [Shapiro], [Tsotsos] and our own work) and in whether they treat the underlying time-line as continuous ([Hayes], [Forbus, 1982] and our own work) or discrete ([Shapiro] and [Tsotsos]).

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 an object -- both in terms of change to its attributes and in terms of its creation and destruction. However, these logics are all oriented towards relations between objects instead of towards the objects themselves. This creates two difficulties. First, it is difficult for these logics to use the assumption that values remain constant unless indicated otherwise. Second, it is difficult to reason about the cumulative changes over time to an attribute of an object. Since both of these are necessary for our task, our history representation employs temporal logic concepts but places them in an object-centered setting. The value of such object-centered

temporal representations was first discussed by [Hayes].

### 6.2.2 Diagrams

The use of diagrams in quantitative simulation is an important aspect of our approach to geologic interpretation. 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]). For these tasks the rationale for using diagrams is the same as our own -- the task domain is largely spatial in nature and diagrams are a representation especially suited for 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 diagrammatic representations (e.g. [Baumgart], [Forbus, 1981], [Funt], [Gelernter], [Hunter]). Most of the differences in these representations involve tradeoffs between shape description and ease of use. For example, in many domains arbitrary shapes must be represented, and so a representation like [Hunter] might be preferred over [Baumgart], which uses only straight lines. However, such representations often make spatial manipulations (e.g., drawing a line) difficult to perform, while the Euler operations in our wing-edge structure do these manipulations 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.

### 6.2.3 Quantity lattice

Parameter determination involves reasoning about and doing arithmetic on quantities whose values are only partially specified. This type of numeric reasoning has also figured in several recent efforts (e.g. [Brooks], [Forbus, 1982], [McDermott, 1981]), where quantities are often specified in terms of equations or inequalities. The implementation of our quantity lattice is a cross between that in [Forbus, 1982], which uses partial orderings, and that of [McDermott, 1981], which uses real-valued intervals.

These representations are all adequate for reasoning about orderings between quantities. However, the arithmetic performed in both [McDermott, 1981] and in [Brooks] is considerably more sophisticated than our combination of interval and relational arithmetic. The basic difference is that our system deals with expressions only on a local basis. In particular, it has no notion of algebraic simplification or of symbolically combining expressions. Thus, for example, if **B** equals **A / X**, our system would not be able to infer that **B * X** is equal to **A**. Although our arithmetic technique is sufficient for the current system, we believe that the eventual use of continuous models (see Chapter 7) will necessitate the use of a more sophisticated (and more heavily computational) arithmetic technique.

### 6.2.4 Processes

All systems which perform simulation must represent processes or actions in some form. The imaginer uses two types of process representations -- an operational model which emphasizes describing how to simulate a process and a causal model which explicitly describes the effects of a process.

Many AI systems specify actions operationally (e.g. [Buchanan], [deKleer, 1975], [Funt], [Winograd, 1972]). The major advantage of this type of representation is that it is usually easy to describe an action in terms of which steps to perform. 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. This is the major distinction between "procedural" and "declarative" representations (see [Winograd, 1975] and [Sussman]). Procedural (operational) representations are used to encode sequences of steps. Control issues - specifically sequentiality and branching - are easily and explicitly encoded in procedural terms and global interactions among the steps can be dealt with easily. In addition, the step-by-step nature of the representation makes it easy to build an interpreter to simulate these processes.

In contrast, the major advantage of a declarative representation is that facts can be added independently. That is, when a fact is added one does not have to change existing facts and one does not have to worry about the order in which the statements are added. Also, facts encoded declaratively can be used for different types of reasoning. For example, a declarative process representation can be used both for simulation and for reasoning about the changes themselves. For example, in gap filling (see Section 2.2.3) we analyze the list of process effects to determine which process could give rise to a specified change.

One type of declarative model is the "causal model" which explicitly encodes the effects caused by processes. This type of model has been a focus of considerable attention (e.g. [deKleer, 1982], [Forbus, 1982], [Hayes], [Hendrix], [Kuipers], [McDermott, 1982], [Patil], [Rieger]). All of these efforts explicitly represent the changes that result from actions. Our representation packages together the preconditions, parameters, affected objects, etc., which define the process (see Section 5.2). This approach facilitates determining such things as how a particular event affects the world or what parts of the world are affected.

Our models have the most in common with those described in [Fikes] and [Forbus, 1982], where processes are represented explicitly and where process representations make explicit which objects are affected, created and changed. 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. Another difference is that we can model dependencies (see Section 5.2.2), which enable us to reason about how objects are affected by changes to other objects. Our particular notion of dependencies follows the lines of the "Qprop" statement of [Forbus, 1982].

The major difference between our causal models and those in [Forbus, 1982] or [Hendrix], is that ours are discrete models rather than continuous. In fact, most of the research in modeling change uses discrete models, often for reasons similar to ours: discrete models are sufficient to solve the problem at hand. In addition, while continuous models may provide a more detailed description of the process, some changes are extremely difficult to specify in continuous terms (e.g., the continuous change in rock composition due to metamorphosis).

It is clear, however, that some sort of continuous model will be needed to solve the full range of geologic interpretation problems (see Chapter 7). One way to combine the advantages of both discrete and continuous models is to use multiple levels of abstraction (as in [Patil]) where the appropriate model is chosen depending on the task and the necessary information is passed between the levels. In the next chapter we speculate on extending our system to include such multi-level models.

# 7. FUTURE WORK

The two main avenues of research we hope to pursue are improving the imagining technique and designing a system which will generate, test and debug sequences of geologic events.

## 7.1 Improving the Imagining Technique

Although the imaginer has been run on several examples (Appendix E presents two additional examples), it has many limitations which we hope to rectify in the future.

The major limitation of the current system is its inability to handle simultaneous events, both interacting and non-interacting. Non-interacting events are relatively easy to deal with. The qualitative simulation currently can handle such events and the quantitative simulation can be done simply by linearizing the partially ordered sequence of events. Since the events are non-interacting, any total ordering consistent with the partial ordering will work.

Interacting events, that is, ones which affect the same attribute at the same time, pose a more difficult problem. In particular, a discrete model of processes is inadequate to reason about the interactions. We need to devise some form of continuous model (such as in [Forbus, 1982]). However, due its computational cost, we want to use a continuous model only when necessary. Thus, we intend to develop a multi-level process representation, along with the necessary inference mechanisms to reason about continuous models.

A major difficulty with interacting events is representing their effects using our time-line representation. We would like to extend this representation without destroying its utility in reasoning about change. A possible approach is to extend the concept of dynamic interval. We can describe the fact that **A** and **B** are interacting events by inserting three consecutive dynamic intervals in the time-line of an attribute affected by both processes -- one for the interval when **A** occurs alone, one for their joint occurrence and one for when **B** occurs

alone. The "change equation" of the joint dynamic interval will represent the combined effects of both processes. Thus, we will need some way to combine the effects of continuous models of processes (possibly along the lines of [Forbus, 1982] or [Patil]).

Another important effort to extend the scope of the imaginer system involves making the qualitative process models more robust, primarily by conditionalizing the changes. Thus, we would like to predict different effects depending on various initial conditions. For example, in erosion we want to infer that the bottom of the surface changes only if it is currently above the erosional level (see also Section 5.3.1). We can do this by having "if <condition> then <change>" clauses for each exceptional case.

### 7.2 Solving the Geologic Interpretation Problem

The primary avenue of research will be to complete the system for doing geologic interpretation. This involves the design and implementation of the scenario matcher, to generate candidate solution sequences, and of the gap filler, to debug solutions. The resulting system will be a test of the applicability of a "generate, test and debug" paradigm for solving interpretation problems. We believe that the representations used in doing imagining will be applicable to the rest of the system. In fact, these representations, particularly the qualitative object and process representations, were developed with an eye towards the rest of the geologic interpretation tasks. Some of the research problems to be addressed are:

1. How to use differences between states to debug candidate solutions --

   The most important question is what does it mean to be a "difference" and how differences should be reported. For example, if the difference is the orientation is $20^{o}$ instead of $30^{o}$, should this be reported as "the orientations were different," "the orientation is off by $10^{o}$" or "actual orientation is $20^{o}$; desired orientation is $30^{o}$". Should all differences be reported or just the "important" ones, and if so, what is the measure of "importance".

Given that we can report differences between states, the gap filler must find processes whose effects will eliminate those differences without causing any others to occur. This is done by analyzing the effects and relations of the process models, instead of by simulating the processes. It is likely that this new task will necessitate modifying the qualitative process representation.

2. How to describe spatial patterns and recognize them in the goal diagram --

   It is probably not general enough to represent patterns as partial wing-edge structures. We need a grammar for describing patterns and a matching technique to scan the diagrams and recognize examples of the patterns. Control issues (which patterns to look for and in which parts of the diagram) will also be important here.

3. How to improve the "piece structure" of physical objects --

   This is especially important in scenario matching, where part of the task is to build up complete formations from the pieces in the diagram. For example, if we see an igneous rock-unit crossing two rock-units of the same composition, we can conclude that the igneous rock-unit intruded through *and* that both rock-units belong to the same formation. As it currently stands, our representation of piece structures is quite weak, especially in the area of inheriting knowledge from the aggregates or in abstracting knowledge from the pieces. For example, the system should infer that the pieces of a rock-unit have the same composition as the rock-unit and that if all the pieces of a rock-unit change orientation, then the rock-unit also changes orientation.

4. How to handle multiple candidate solutions --

   Each candidate solution must be verified using the imaginer. However, most sequences will be nearly the same, with only a few events in different order. We would like the system to avoid redundant work in doing the imagining and gap filling. This is a fairly difficult problem involving the use of shared contexts, and we currently have no good ideas for solving it.

There are two other important areas of research, which we do not plan to tackle in the near future -- generality and learning.

By "generality" we mean the applicability of the imagining technique to other domains. We believe that this combination of qualitative and quantitative simulation will prove useful in other fields, particularly where causal theories to explain data are not well understood, such as in economics or biology. We also believe that using histories for representing change to objects will prove useful in many temporal domains. In particular, the maintenance of complete change sequences and the assumptions of constancy during quiescence should facilitate many temporal reasoning problems.

There are three specific learning problems of interest to us. First, how can we derive scenarios, that is diagrammatic patterns and local interpretations, given geologic interpretation problems and their solutions. Second, how can we construct "multi-process" descriptions. Geologists will often talk about "uplift/erosion" as if it were one event. The problems for this task are in combining the effects of the processes and deciding which combinations are useful to make. Third, as discussed in Section 5.4, the qualitative models can be refined by comparing the results of the quantitative simulation with known results, that is, with the goal diagram. We believe this process could be automated, by presenting the system with "near misses", having it determine the differences and patching the appropriate process model to eliminate the cause of those differences.

# 8. CONCLUSIONS

This report has discussed *imagining*, a simulation technique used to test whether a sequence of geologic events could have led to the formation of a particular region. Imagining, which uses a combination of qualitative and quantitative simulation techniques, requires spatial, temporal and numeric reasoning. We have developed separate, specialized representations to facilitate each of these reasoning tasks.

Simulation is useful for determining the effects of a sequence of events. It is particularly applicable where knowledge of the domain is limited and cannot support more sophisticated techniques which involve reasoning about the character of the events themselves. Imagining uses two kinds of simulation. A *qualitative simulation* is used in conjunction with the goal diagram to transform the initial qualitative sequence of events into a quantitative sequence. Then a *quantitative simulation* is performed by constructing a sequence of diagrams that represent the spatial effects of the events. The final diagram in the sequence is then matched against the goal diagram. We believe that imagining may prove useful in other domains where a qualitative sequence of events is hypothesized in order to explain quantitative data.

The diversity of reasoning skills needed by the imaginer necessitates the use of multiple representations, where a representation is defined as a set of primitive objects and inference mechanisms, specialized for a particular task. Our system uses five representations -- qualitative history-based objects, quantitative diagrams, qualitative causal models of processes, quantitative operational models of processes and the quantity lattice.

There are two reasons for using multiple representations -- efficiency in reasoning and adequacy of representation. First, systems like [MACSYMA] demonstrated an important principle: the efficiency gained by performing inferences in representations designed to support those inferences is often worth the cost of translating between representations.

Second, often no single representation will be adequate for encoding every type of domain knowledge in a form which can support the necessary inferences and manipulations. For example, it is quite hard to represent shape qualitatively in a form suitable for reasoning about changes to the shape, but it is easy to do so in a quantitative representation based on diagrams. Conversely, it is difficult to describe rock composition quantitatively, but qualitative descriptions, such as "shale" or "sandstone", are easy to represent and are adequate for doing geologic interpretation.

Still, choosing representations is a difficult task. We have tried to follow a "top-down" methodology for choosing representations similar to the one described in [Marr]. First, we establish criteria for choosing the representation based on the task that the representation needs to support. We do this by determining what needs to be described about the world (objects and relations) and what inferences and manipulations are needed. The inferences and manipulations are especially important to determine as they delineate the representations which will be adequate for the task. Second, we choose "the best" of all the representations which satisfy the criteria. A representation is considered "good" if knowledge can be represented compactly and inferences can be made efficiently. It is important to choose a representation which makes explicit the natural constraints in the task domain (see [Marr]). This tends to make local operations in the representation correspond to local inferences and manipulations in the real-world. Third, we choose a computationally efficient implementation of the representation.

Throughout this work, a conscious effort has been made to define the criteria for each representation. The need to reason about the temporal extent of changes led us to adapt the notion of *histories* [Hayes] to represent physical objects which change over time. In particular, objects have associated life-spans, and the values of attributes are represented as time-lines. These time-lines incorporate an assumption of constancy that an attribute does not change until it is asserted otherwise.

The need to describe and modify spatial properties of objects led to the development of a quantitative representation based on diagrams. The diagram is used for measuring geologic properties, like thickness, for simulating the effects of processes, like erosion, and for matching against the goal diagram.

We have found that using a discrete process model is sufficient for solving most geologic interpretation problems. However, the two simulations performed by the imaginer are most easily carried out using different process representations. The quantitative simulation is easily accomplished using an operational process description which encodes the steps to be performed, while the qualitative simulation uses a causal description which explicitly represents changes to objects. We have also found it very useful to represent processes explicitly, both for describing the geologic process knowledge and for performing the simulation.

As our realization of the complexity of intelligence grows, we become increasingly aware of the need for multiple representations, specialized to perform specific tasks, and of the need to formalize our understanding of what constitutes a suitable representation. We hope that this report has helped to take a step in that direction.

# REFERENCES

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

Bledsoe, W; Boyer, R; Henneman, W [1972] - "Computer Proofs of Limit Theorems," *Artificial Intelligence*, vol. 3, pp. 27-60.

Brooks, Rodney [1981] - "Symbolic Reasoning Among 3D Models and 2D Images," *Artificial Intelligence*, vol. 17, pp. 285-348.

Buchanan, B; Sutherland, G; Feigenbaum, E [1969] - "Heuristic DENDRAL: A Program for Generating Explanatory Hypotheses in Organic Chemistry," in *Machine Intelligence 4*, American Elsevier.

Davis, Randall; Buchanan, Bruce; Shortliffe, Edward [1977] - "Production Rules as a Representation for a Knowledge-Based Consultation Program," *Artificial Intelligence*, vol. 8.

Davis, Randall; et. al. [1981] - "The DIPmeter Advisor: Interpretation of Geologic Signals," *IJCAI 7*.

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

DeKleer, Johan [1979] - "Causal and Teleological Reasoning in Circuit Recognition." MIT AI-TR-529.

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

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.

Kuipers, Benjamin [1982] - "Getting the Envisionment Right," *Proceedings of the National Conference on Artificial Intelligence*, Pittsburgh, PA, p. 209.

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

McAllester, David A [1982] - "**R**easoning **U**tility **P**ackage User's Manual, Version One," MIT AIM-667.

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

McCarthy, John; Hayes, Patrick [1969] - "Some Philosophical Problems from the Standpoint of Artificial Intelligence," in *Machine Intelligence 4*, eds. Meltzer and Michie.

McDermott, Drew; Davis, Ernest [1981] - "Planning and Executing Routes Through Uncertain Territory."

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

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

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 [1973] - "Artificial Intelligence and the Concept of Mind," in *Computer Models of Language and Thought*, eds. Schank and Colby.

Nii, Penney; et. al. [1982] - "Signal-to-Symbol Transformation: HASP/SIAP Case Study," *AI Magazine*, vol. 3, no. 2.

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 [1981] - "Sniffer: a System that Understands Bugs," MIT-AIM-638.

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

Sussman, Gerald J [1975] - *A Computer Model of Skill Acquisition*, American Elsevier.

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.

Winograd, Terry [1975] - "Frame Representations and the Declarative-Procedural Controversy," in *Representation and Understanding*, eds. Bobrow and Collins, Academic Press.

# Appendix A - GEOLOGIC OBJECTS

This appendix lists the geologic object descriptions used by the imaginer system. The objects are arranged in a type hierarchy, with type "OBJECT" as the root. An object type inherits all the attributes of its parent, except if the same attribute name already appears in the object type description. The general form of a description is:

<NAME> ⇒ <AKO> {CONSTRAINTS}
<ATTRIBUTE> : <TYPE>, <ATTRIBUTE> : <TYPE>, ...

All the attributes of the descendants of "temporal-objects" represent time-lines of values, rather than a single value, except for the "start" and "end" attributes. For "abstract-objects" and its descendants, the attributes represent single values. Also, the object type "rock-material" and its descendants are not used to any great extent in the system, except for filling the diagram faces with patterns, but are included anyway for completeness.

1. ABSTRACT-OBJECT ⇒ object

2. QUANTITY ⇒ abstract-object
   value : (member-of {-∞ ... ∞})

3. REAL ⇒ quantity

4. POSITIVE-REAL ⇒ real {(> positive-real.value 0.0)}

5. ANGLE ⇒ real

6. TIME ⇒ quantity

7. TEMPORAL-INTERVAL ⇒ abstract-object {(< temporal-interval.start temporal-interval.end)}
   start : time,
   end : time

8. TEMPORAL-OBJECT ⇒ object
   start : time,
   end : time

9. PHYSICAL-OBJECT ⇒ temporal-object
   pieces : (set-of physical-object),
   aggregates : (set-of physical-object)

10. POINT ⇒ temporal-object
   height : real,
   lateral : real

11. PHYSICAL-FEATURE ⇒ physical-object

12. SEA ⇒ physical-feature

13. AIR ⇒ physical-feature

14. GEOLOGIC-FEATURE ⇒ physical-feature
   top : point,
   bottom : point,
   location : point,
   orientation : angle

15. BOUNDARY ⇒ geologic-feature
   side-1 : (set-of physical-feature),
   side-2 : (set-of physical-feature)

16. FAULT ⇒ boundary
   fault-plane : gplane,
   fault-type : (one-of 'normal, 'reverse, 'lateral),
   slip-direction : angle,
   slip : real

17. ROCK-UNIT ⇒ geologic-feature
   thickness : positive-real,
   composition : rock-material,
   rock-type : (one-of 'sedimentary, 'igneous, 'metamorphic)

18. DOWN-THROWN-BLOCK ⇒ rock-unit

19. UP-THROWN-BLOCK ⇒ rock-unit

20. SEDIMENTARY ⇒ rock-unit
    bedding-plane : gplane,
    rock-type : 'sedimentary,
    composition : sedimentary-rock

21. IGNEOUS ⇒ rock-unit
    rock-type : 'igneous,
    composition : igneous-rock

22. INTRUSIVE ⇒ igneous
    itilt : angle

23. BATHOLITH ⇒ intrusive
    composition : (one-of granite, basalt),
    bounding-plane : gplane

24. DIKE-OR-SILL ⇒ intrusive
    composition : (one-of mafic-igneous, granite)
    center-plane : gplane

25. METAMORPHIC ⇒ rock-unit
    rock-type : 'metamorphic,
    composition : metamorphic-rock

26. ROCK-MATERIAL ⇒ physical-object
    minerals : (set-of mineral),
    petrogenesis : geologic-process

27. SEDIMENTARY-ROCK ⇒ rock-material
    petrogenesis : deposition-of-sediments

28. IGNEOUS-ROCK ⇒ rock-material
    petrogenesis : cooling-of-molten-rock-material

29. METAMORPHIC-ROCK ⇒ rock-material
    petrogenesis : extreme-temperature-and-pressure

30. CLASTIC ⇒ sedimentary-rock
    detrital-sediment : detritus

31. SHALE ⇒ clastic
    detrital-sediment : mud

32. SANDSTONE ⇒ clastic
    minerals : (set-of quartz),
    detrital-sediment : sand

33. CONGLOMERATE ⇒ clastic
    detrital-sediment : gravel

34. GRANITE ⇒ igneous-rock
    minerals : (set-of mineral, quartz, feldspar),
    texture : 'varied-grain-sizes

35. MAFIC-IGNEOUS ⇒ igneous-rock

36. SCHIST ⇒ metamorphic-rock

37. GPLANE ⇒ temporal-object
    xz-angle : angle,
    y-angle : angle,
    location : point

38. XZ-PLANE ⇒ gplane {(= xz-plane.xz-angle 0.0)}

# Appendix B - QUALITATIVE PROCESSES

This appendix lists the qualitative process descriptions used by the imaginer system. The representation used is described in Section 5.2. In addition, some processes are represented as specializations (AKO) of others (for example, "DIP-SLIP-FAULTING" is a special kind of "FAULTING"). Specialized processes inherit everything from their "AKO" process (relations, effects, parameters, etc.), except if the same variable name appears in a PARAMETER, AFFECTED, or CREATED list, then the occurrence in the specialized process replaces the more general one.

```
1. 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 function SURFACE.side-2 (dfn1 A) I DEPOSITION)
                   (change = SURFACE.orientation
                           (dfn3 DLEVEL SURFACE@I_start) I DEPOSITION)
                   (change function SURFACE.bottom (dfn DLEVEL) I DEPOSITION)
                   (change function SURFACE.location (dfn DLEVEL) I DEPOSITION)
                   (change function SURFACE.top (dfn DLEVEL) I DEPOSITION)
                   (change = A.thickness DLEVEL I DEPOSITION)
                   (change = A.bedding-plane.y-angle 0.0 I DEPOSITION)
                   (change = BA.side-1 {A} I DEPOSITION)
                   (change = BA.side-2 C I DEPOSITION)
                   (change = BA.bottom SURFACE.bottom@I_start I DEPOSITION)
                   (change = BA.top (dfn4 DLEVEL SURFACE@I_start) 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)
   RELATIONS       (= SURFACE.bottom.height@I_end
                      (+ DLEVEL SURFACE.bottom.height@I_start))
                   (equiv A.bedding-plane.y-angle A.orientation I_end)
                   (< A.top.height@I_end SEA-LEVEL)
                   (=depends A.orientation BA.orientation I_end)
                   (member A SURFACE.side-2@I_end)
                   (⇒ (> SURFACE.top.height@I_start A.top.height@I_end)
                       (for-all r ∈ SURFACE.side-2@I_start
                           (⇒ (> r.top.height@I_start A.top.height@I_end)
                               (member r SURFACE.side-2@I_end))))
                   (= A.top.height@I_end SURFACE.bottom.height@I_end)
                   (⇒ (≤ SURFACE.top.height@I_start SURFACE.bottom.height@I_end)
                       (= SURFACE.top.height@I_end A.top.height@I_end))
                   (⇒ (= SURFACE.top.height@I_start SURFACE.bottom.height@I_start)
                       (= BA.top.height@I_end BA.bottom.height@I_end))
                   (= C {r : rock-unit |
                           (and (exists-at r I_start)
                                (< r.bottom.height@I_start SURFACE.bottom.height@I_end)
                                (on-surface r I_start))})
```

```
2. EROSION
   INTERVAL       I : temporal-interval
   PRECONDITIONS (> SURFACE.top.height@I_start SEA-LEVEL)
   PARAMETERS     ELEVEL : real
   AFFECTED       C-PART : (set-of rock-unit), C-ALL : (set-of rock-unit),
                   SURFACE
   CREATED        BA : boundary
   EFFECTS        (change = BA.side-1 {THE-AIR} I EROSION)
                  (change = BA.side-2 C-PART I EROSION)
                  (change = BA.orientation 0.0 I EROSION)
                  (change = SURFACE.orientation
                          (efn3 ELEVEL SURFACE@I_start) I EROSION)
                  (change function SURFACE.top (efn2 ELEVEL) I EROSION)
                  (change function SURFACE.bottom (efn2 ELEVEL) I EROSION)
                  (change function SURFACE.location (efn2 ELEVEL) I EROSION)
                  (for-all c1 ∈ C-PART
                      (and (change function c1.top (efn2 ELEVEL) I EROSION)
                           (change function c1.location (efn2 ELEVEL) I EROSION)
                           (change - c1.thickness
                              (* (efn1 c1 ELEVEL) (- c1.top.height@I_start ELEVEL))
                              I EROSION))
                  (for-all c2 ∈ C-ALL (change = c2 ⊥ I EROSION))
   RELATIONS      (> ELEVEL SEA-LEVEL)
                  (= C-PART {r : rock-unit |
                               (and (exists-at r I_start)
                                    (≥ r.top.height@I_start ELEVEL)
                                    (< r.bottom.height@I_start ELEVEL))})
                  (= C-ALL {r : rock-unit |
                              (and (exists-at r I_start)
                                   (≥ r.bottom.height@I_start ELEVEL))})
                  (⇒ (≥ SURFACE.bottom.height@I_start ELEVEL)
                     (= SURFACE.bottom.height@I_end SURFACE.top.height@I_end))
                  (= SURFACE.top.height@I_end ELEVEL)
                  (≤ SURFACE.bottom.height@I_end SURFACE.bottom.height@I_start)
                  (for-all c1 ∈ C-PART
                      (and (= c1.top.height@I_end ELEVEL)
                           (≥ (efn1 c1 ELEVEL) 0.0)
                           (≤ (efn1 c1 ELEVEL) 1.0)))
```

```
3. FAULTING
   INTERVAL      I : temporal-interval
   PARAMETERS    FFAULT-PLANE : gplane, FSLIP : positive-real, FDIRECTION : angle,
                 FFAULT-TYPE : (one-of 'NORMAL, 'REVERSE, 'LATERAL)
   AFFECTED      C : (set-of rock-unit), SURFACE
   CREATED       F : fault, DTB : down-thrown-block, UTB : up-thrown-block
   EFFECTS       (change - DTB.location.height
                         (* FSLIP (ffn1 FDIRECTION FFAULT-PLANE)) I FAULTING)
                 (change + DTB.location.lateral
                         (* FSLIP (ffn2 FDIRECTION FFAULT-PLANE)) I FAULTING)
                 (change = F.slip FSLIP I FAULTING)
                 (change = F.fault-plane FFAULT-PLANE I FAULTING)
                 (change = F.slip-direction FDIRECTION I FAULTING)
                 (change = F.side-1 {DTB} I FAULTING)
                 (change = F.side-2 {UTB} I FAULTING)
   RELATIONS     (equiv F.orientation F.fault-plane.y-angle I_end)
                 (equiv F.location F.fault-plane.location I_end)
                 (= (ffn1 FDIRECTION FFAULT-PLANE)
                    (* (sin FDIRECTION) (abs (sin FFAULT-PLANE.y-angle@I_end))))
                 (= (ffn2 FDIRECTION FFAULT-PLANE)
                    (* (sin FDIRECTION) (cos FFAULT-PLANE.y-angle@I_end)))

4. DIP-SLIP-FAULTING
   AKO ⇒ FAULTING
   PARAMETERS    FFAULT-TYPE : (one-of 'NORMAL, 'REVERSE)
   EFFECTS       (for-all r1 ∈ R
                     (change function r1.bottom
                             (ffn3 FSLIP FFAULT-PLANE FDIRECTION)
                             I DIP-SLIP-FAULTING))
                 (for-all b1 ∈ B
                     (change function b1.bottom
                             (ffn3 FSLIP FFAULT-PLANE FDIRECTION)
                             I DIP-SLIP-FAULTING))
   RELATIONS     (= FDIRECTION 90.0)
                 (= R {ru : rock-unit | (exists-at ru I_start)})
                 (= B {ba : boundary | (exists-at ba I_start)})
                 (for-all b1 ∈ B
                         (⇒ (= b1.orientation@I_start 0.0)
                             (and (is-point-of b1.bottom@I_end DTB)
                                  (= (b1.bottom@I_end).height@I_start
                                     b1.bottom.height@I_start))))
```

```
5. INTRUSION
   INTERVAL      I : temporal-interval
   PARAMETERS    IWIDTH : positive-real, ICOMPOSITION : igneous-rock
   AFFECTED      C : (set-of rock-unit)
   CREATED       A : igneous, BA : boundary
   EFFECTS       (for-all c1 ∈ C
                     (change - c1.thickness (* (ifn c1 A) IWIDTH) I INTRUSION)
                 (change = A.thickness IWIDTH I INTRUSION)
                 (change = A.composition ICOMPOSITION I INTRUSION)
                 (change = BA.side-1 {A} I INTRUSION)
                 (change = BA.side-2 C I INTRUSION)
   RELATIONS     (for-all r1 ∈ C
                     (and (≤ (ifn r1 A) 1.0) (≥ (ifn r1 A) 0.0)))
                 (= C {r2 : rock-unit | (and (exists-at r2 I_start)
                                             (spatially-intersects r2 A)))

6. BATHOLITHIC-INTRUSION
   AKO ⇒ INTRUSION
   PARAMETERS    IBOUNDING-PLANE : gplane
   CREATED       A : batholith
   EFFECTS       (change = A.bounding-plane
                         IBOUNDING-PLANE I BATHOLITHIC-INTRUSION)
                 (for-all r ∈ C
                     (change function r.bottom
                             (ifn1 bounding-plane) I BATHOLITHIC-INTRUSION))
                 (change = A.orientation 0.0 I BATHOLITHIC-INTRUSION)
   RELATIONS     (for-all r : rock-unit
                     (⇒ (exists-at r I_start)
                         (iff (spatially-intersects r A)
                              (≤ r.bottom.height@I_start
                                  IBOUNDING-PLANE.location.height@I_end))))
                 (=depends A.orientation BA.orientation I_end)
                 (equiv BA.orientation A.bounding-plane.y-angle I_end)
                 (equiv BA.location A.bounding-plane.location I_end)
                 (= BA.top.height@I_end A.top.height@I_end)
                 (for-all r1 ∈ C (= r1.bottom.height@I_end BA.bottom.height@I_end))
```

```
7. DIKE-OR-SILL-INTRUSION
   AKO ⇒ INTRUSION
   PARAMETERS    ICENTER-PLANE : gplane
   CREATED       A : dike-or-sill
   EFFECTS       (change = A.center-plane
                           ICENTER-PLANE I DIKE-OR-SILL-INTRUSION)
   RELATIONS     (equiv A.orientation BA.orientation I_end)
                 (equiv BA.orientation A.center-plane.y-angle I_end)
                 (equiv BA.location A.center-plane.location I_end)
                 (for-all r : rock-unit
                     (⇒ (exists-at r I_start)
                         (and (iff (spatially-intersects r A)
                                   (plane-intersects ICENTER-PLANE r))
                              (= (ifn r A) 1.0))))

8. METAMORPHISM
   INTERVAL      I : temporal-interval
   AFFECTED      C : (set-of rock-unit)
   EFFECTS       (for-all c1 ∈ C
                     (and (change = c1.rock-type
                                  'METAMORPHIC-ROCK I METAMORPHISM)
                          (change = c1.type 'METAMORPHIC I METAMORPHISM)
                          (change function c1.composition
                                  metamorphic-counterpart I METAMORPHISM)))
   RELATIONS     (= C {r : rock-unit |
                           (and (exists-at r I_start) (is-deep r.top I_start))})

9. SUBSIDENCE
   INTERVAL      I : temporal-interval
   PARAMETERS    SUBSIDE-AMOUNT : positive-real
   AFFECTED      C : (set-of geologic-feature)
   EFFECTS       (for-all c1 ∈ C (change - c1.location.height
                                          SUBSIDE-AMOUNT I SUBSIDENCE))
   RELATIONS     (= C {gf : geologic-feature | (exists-at gf I_start)})

10. TILT
    INTERVAL     I : temporal-interval
    PARAMETERS   THETA : angle
    AFFECTED     C : (set-of geologic-feature)
    EFFECTS      (for-all c1 ∈ C (change + c1.orientation THETA I TILT))
    RELATIONS    (= C {gf : geologic-feature | (exists-at gf I_start)})
```

11. UPLIFT

```
    INTERVAL     I : temporal-interval
    PARAMETERS   UPLIFT-AMOUNT : positive-real
    AFFECTED     C : (set-of geologic-feature)
    EFFECTS      (for-all c1 ∈ C (change + c1.location.height
                                          UPLIFT-AMOUNT I UPLIFT))
    RELATIONS    (= C {gf : geologic-feature | (exists-at gf I_start)})
```

# Appendix C - GEOLOGIC AXIOMS

This appendix lists the geologic and domain independent axioms (implemented in RUP [McAllester]) which are necessary to complete the geologic reasoning system. Each logical axiom will be accompanied by a brief English description.[24]

The first four axioms are domain independent :

1. If a dependency exists between DA and A, then if A changes after TIME, DA changes. The magnitude of the change to DA is found by applying FN, a function of two arguments, to EFFECT and the value of DA at the start of the change.

```
(FOR-ALL (DA, A, FN, TYPE, TIME, I, CAUSE, EFFECT)
         (⇒ (FDEPENDS DA A FN TIME)
             (⇒ (AND (CHANGE TYPE A EFFECT I CAUSE) (> I_start TIME))
                 (CHANGE FUNCTION DA FN(EFFECT) I CAUSE))))
```

2. If a dependency exists between DA and A, then if A changes after TIME, DA changes in the same way (that is, the same magnitude of change).

```
(FOR-ALL (DA, A, TIME, TYPE, I, CAUSE, EFFECT)
         (⇒ (=DEPENDS DA A TIME)
             (⇒ (AND (CHANGE TYPE A EFFECT I CAUSE) (> I_start TIME))
                 (CHANGE TYPE DA EFFECT I CAUSE))))
```

3. If attributes ATT1 and ATT2 are equivalent starting from TIME, then they have the same value at that time and if one changes the other changes by the same amount.

```
(FOR-ALL (ATT1, ATT2, TIME)
         (⇒ (EQUIV ATT1 ATT2 TIME)
             (AND (= ATT1@TIME ATT2@TIME)
                  (=DEPENDS ATT1 ATT2 TIME) (=DEPENDS ATT2 ATT1 TIME))))
```

---

24. For efficiency reasons, some of the actual axioms implemented in RUP differ in form from those presented here, although they are logically equivalent.

4. An object exists at TIME if, and only if, its "start" (creation) time is not after TIME and its "end" (destruction) time is not before TIME.

```
(FOR-ALL (GF, TIME)
         (IFF (EXISTS-AT GF TIME) (AND (≤ GF.START TIME) (≥ GF.END TIME))))
```

The rest of these axioms are domain-specific to geology :

5. For all geologic-features, their tops and bottoms are points associated with the geologic-feature.

```
(FOR-ALL (GF : GEOLOGIC-FEATURE, TIME)
         (AND (IS-POINT-OF GF.TOP@TIME GF)
              (IS-POINT-OF GF.BOTTOM@TIME GF)))
```

6. In our model of geology, a point of a geologic-feature must have existed at least as long as the feature itself.

```
(FOR-ALL (PT : POINT, GF : GEOLOGIC-FEATURE)
         (⇒ (IS-POINT-OF PT GF) (≤ PT.START GF.START)))
```

7. The height and lateral position of a point depend on the height and lateral position of the "location" of its associated geologic-feature.

```
(FOR-ALL (PT : POINT, GF : GEOLOGIC-FEATURE)
         (⇒ (IS-POINT-OF PT GF)
             (AND (=DEPENDS PT.HEIGHT GF.LOCATION.HEIGHT GF.START)
                  (=DEPENDS PT.LATERAL GF.LOCATION.LATERAL GF.START))))
```

8. The height and lateral position of a point changes if the orientation of its associated geologic-feature changes.

```
(FOR-ALL (PT : POINT, GF : GEOLOGIC-FEATURE)
         (⇒ (IS-POINT-OF PT GF)
             (AND (FDEPENDS PT.HEIGHT GF.ORIENTATION
                            (OFN3 PT.HEIGHT PT.LATERAL) GF.START)
                  (FDEPENDS PT.LATERAL GF.ORIENTATION
                            (OFN4 PT.HEIGHT PT.LATERAL) GF.START))))
```

9. Describes the change in height of a point which starts at position (HEIGHT, LATERAL) and undergoes a rotation by TILT.

```
(FOR-ALL (HEIGHT, LATERAL, TILT)
         (= (OFN3 HEIGHT LATERAL TILT)
            (+ (* (SIN TILT) LATERAL) (* (COS TILT) HEIGHT))))
```

10. Describes the change in lateral position of a point which starts at position (HEIGHT, LATERAL) and undergoes a rotation by TILT.

```
(FOR-ALL (HEIGHT, LATERAL, TILT)
         (= (OFN4 HEIGHT LATERAL TILT)
            (- (* (COS TILT) LATERAL) (* (SIN TILT) HEIGHT))))
```

11. For all geologic-features, the height of its bottom point is less than the height of its top (or $\leq$, in the case of boundaries), and its "location" is in-between the top and the bottom points.

```
(FOR-ALL (GF : GEOLOGIC-FEATURE, TIME)
         (AND (≤ GF.BOTTOM.HEIGHT@TIME GF.TOP.HEIGHT@TIME)
              (≤ GF.BOTTOM.HEIGHT@TIME GF.LOCATION.HEIGHT@TIME)
              (≤ GF.LOCATION.HEIGHT@TIME GF.TOP.HEIGHT@TIME)))
```

12. If the top and bottom of a boundary have the same height, then the orientation of the boundary is zero.

```
(FOR-ALL (BND : BOUNDARY, TIME)
         (⇒ (= BND.BOTTOM.HEIGHT@TIME BND.TOP.HEIGHT@TIME)
            (= BND.ORIENTATION@TIME 0.0)))
```

13. Something is ON-SURFACE at TIME if, and only if, at TIME it is a member of SIDE-2 of SURFACE, the surface of the Earth.

```
(FOR-ALL (RU, TIME)
         (IFF (ON-SURFACE RU TIME) (MEMBER RU SURFACE.SIDE-2@TIME)))
```

14. In our model of geology, a point is "deep" at TIME if, and only if, its height below the bottom of the surface of the Earth is greater than the quantity *DEEP*.

```
(FOR-ALL (PT, TIME)
         (IFF (IS-DEEP PT TIME)
              (> (- SURFACE.BOTTOM.HEIGHT@TIME PT.HEIGHT@TIME) *DEEP*)))
```

# Appendix D - QUANTITATIVE PROCESSES

This appendix lists the operational descriptions, in English, of the quantitative processes which simulate the effects of geologic processes by manipulating diagrams. These process descriptions are actually implemented as pieces of LISP code. Following each process name is its "quantitative simulation template" (see Section 5.3.2), stated in terms used by the qualitative processes (see Appendix B), which is used to determine the arguments to the quantitative process functions.

1. DEPOSITION -- <(+ DLEVEL SURFACE.bottom.height@$I_{start}$), A, BA>
   (We will refer to (+ DLEVEL SURFACE.bottom.height@$I_{start}$) as "DEP-TOP-HEIGHT").

   1. If the boundary edges of the diagram are lower than DEP-TOP-HEIGHT, then extend the edges of the diagram to the height DEP-TOP-HEIGHT.

   2. Draw a horizontal line at DEP-TOP-HEIGHT.

   3. Erase all parts of the line which cut across a face corresponding to a rock-unit.

   4. All newly created faces (those below the remaining edges of the horizontal line) are pieces of A, the newly deposited rock-unit.

   5. The edges corresponding to SURFACE, the surface of the Earth, are all the old SURFACE edges which are above the horizontal line plus the remaining edges of the horizontal line.

   6. The edges corresponding to BA, the newly created depositional boundary, are all the old SURFACE edges which are below the horizontal line.

2. EROSION -- <ELEVEL, BA>

    1. Draw a horizontal line at ELEVEL.

    2. Erase all parts of the line which do not cut across a face corresponding to a rock-unit.

    3. Erase all faces above the horizontal line.

    4. The edges corresponding to SURFACE, the surface of the Earth, are all the old SURFACE edges which were not erased, plus the remaining edges of the horizontal line.

    5. The edges corresponding to BA, the erosional boundary, are the remaining edges of the horizontal line.

3. DIP-SLIP-FAULTING --
   <FFAULT-PLANE.y-angle@$I_{end}$, FFAULT-PLANE.location.lateral@$I_{end}$,
   FFAULT-PLANE.location.height@$I_{end}$, FFAULT-TYPE, FSLIP, F, DTB, UTB>

    1. Draw a line with slope FFAULT-PLANE.y-angle@$I_{end}$ passing through the point:
       (FFAULT-PLANE.location.lateral@$I_{end}$,
       FFAULT-PLANE.location.height@$I_{end}$).

    2. Split the diagram into two pieces along the line drawn.

    3. If FFAULT-TYPE is "NORMAL", then the down-thrown side of the fault is the piece of the diagram which is above the drawn line; if it is "REVERSE", then the down-thrown side is the piece below the drawn line. For all the vertices of the piece of the diagram being moved, the X-coordinates are translated by (* -FSLIP (COS FFAULT-PLANE.y-angle@$I_{end}$)) and the Y-coordinates are translated by (* -FSLIP (SIN FFAULT-PLANE.y-angle@$I_{end}$)).

    4. Join the two pieces of the diagram together.

    5. The edges of F, the fault boundary, are all the edges along the line drawn in step 1.

6. The edges corresponding to SURFACE, the surface of the Earth, are all the old SURFACE edges, plus the edges corresponding to the fault F which lie between the SURFACE edges.

7. The pieces of DTB, the down-thrown-block, are all the rock-unit pieces corresponding to faces on the down-thrown side of the fault. Also assert that all geologic points corresponding to vertices on the down-thrown side of the fault are points within DTB. Do the same for UTB, the up-thrown-block.

4. BATHOLITHIC-INTRUSION --
<IBOUNDING-PLANE.y-angle@$I_{end}$, IBOUNDING-PLANE.location.lateral@$I_{end}$, IBOUNDING-PLANE.location.height@$I_{end}$, A, BA>

1. Draw a line with slope IBOUNDING-PLANE.y-angle@$I_{end}$ passing through the point:
(IBOUNDING-PLANE.location.lateral@$I_{end}$, IBOUNDING-PLANE.location.height@$I_{end}$).

2. The edges corresponding to BA, the intrusional boundary, are the edges of the line.

3. Erase all the edges within the area bounded by the line and the boundary edges of the diagram.

4. The face bounded by the line and the boundary edges of the diagram corresponds to A, the newly intruded rock-unit.

5. DIKE-OR-SILL-INTRUSION --
<IWIDTH, ICENTER-PLANE.y-angle@$I_{end}$, ICENTER-PLANE.location.lateral@$I_{end}$, ICENTER-PLANE.location.height@$I_{end}$, A, BA>

1. Draw two lines, both with slope ICENTER-PLANE.y-angle@$I_{end}$. One line passes through a point IWIDTH/2 away from the point (ICENTER-PLANE.location.lateral@$I_{end}$, ICENTER-PLANE.location.height@$I_{end}$) along a line perpendicular to ICENTER-PLANE.y-angle@$I_{end}$. The other line passes through a point -IWIDTH/2 away.

2. The edges corresponding to BA, the intrusional boundary, are all the edges of both lines.

3. Erase all the edges within the area bounded by the lines and the boundary edges of the diagram.

4. The face bounded by the lines and the boundary edges of the diagram corresponds to A, the newly intruded rock-unit.

6. SUBSIDENCE -- <SUBSIDE-AMOUNT>

1. Subtract SUBSIDE-AMOUNT from the Y-coordinates of all the vertices in the diagram.

7. TILT -- <THETA>

1. Rotate all the vertices in the diagram by THETA degrees around the point (0,0).

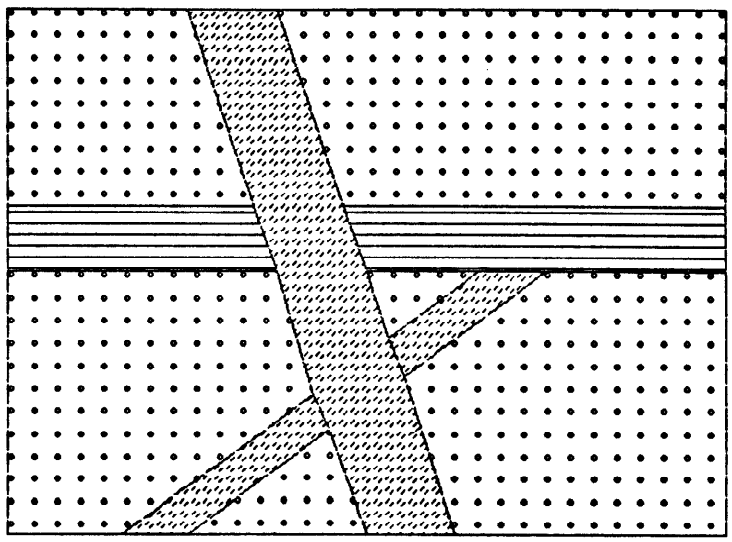
8. UPLIFT -- <UPLIFT-AMOUNT>

1. Add UPLIFT-AMOUNT to the Y-coordinates of all the vertices in the diagram.

# Appendix E - EXAMPLES OF IMAGINING

This appendix presents two additional geologic interpretation problems and their solutions. The validity of the solutions were tested using the imagining technique described in this report. The sequence of diagrams resulting from the quantitative simulation is also presented for each example.

**A.**

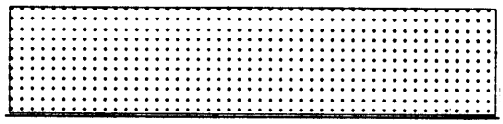


## SOLUTION SEQUENCE

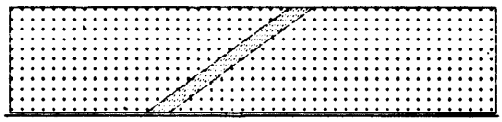
1. Deposit **Sandstone-1**
2. Intrude **Granite-1** through **Sandstone-1**
3. Deposit **Shale**
4. Deposit **Sandstone-2**
5. Intrude **Granite-2** through **Sandstone-1, Granite-1, Shale** and **Sandstone-2**
6. Uplift
7. Erode **Sandstone-2** and **Granite-2**

# SIMULATION OF SOLUTION SEQUENCE

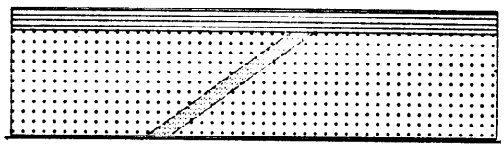
**1.** Deposit Sandstone-1



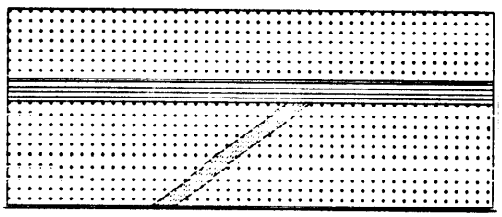
**2.** Intrude Granite-1



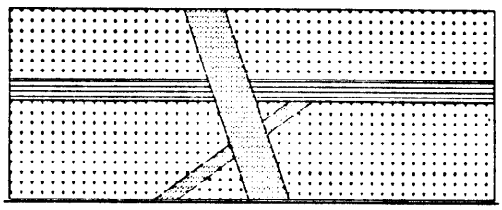
**3.** Deposit Shale



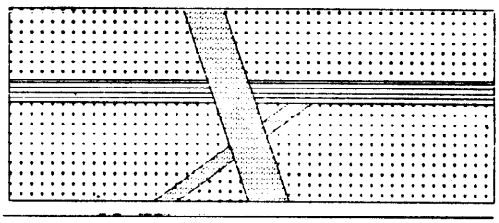
**4.** Deposit Sandstone-2



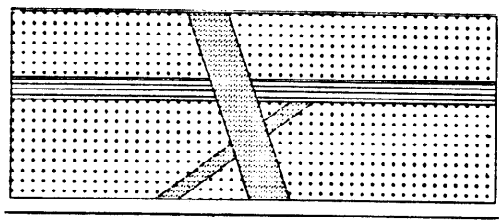
**5.** Intrude Granite-2



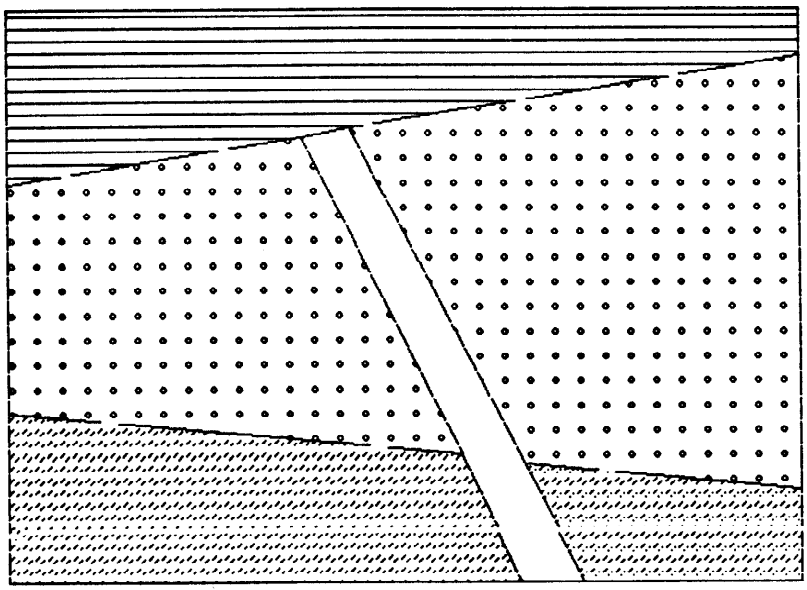
**6.** Uplift



**7.** Erode



**B.**



SHALE SANDSTONE GRANITE MAFIC-IGNEOUS

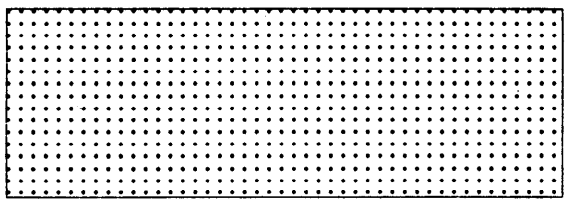
## SOLUTION SEQUENCE

1. Deposit **Sandstone**
2. Intrude **Granite** into **Sandstone**
3. Intrude **Mafic-Igneous** through **Sandstone** and **Granite**
4. Tilt
5. Deposit **Shale**

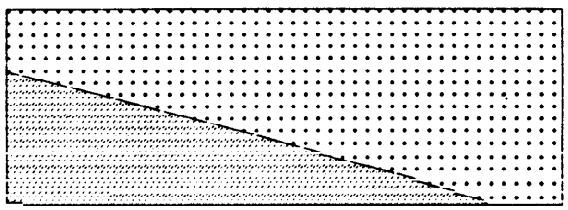
# SIMULATION OF SOLUTION SEQUENCE

This simulation has a slight inaccuracy, although the matching would still succeed. The level of the sandstone in the simulated diagrams is higher than in the goal diagram. This inaccuracy is due to an error in the specification of one of the qualitative process descriptions but we have not yet tracked down the source of the error.

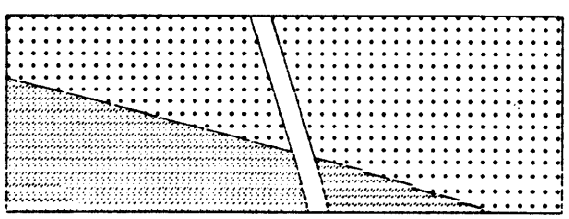
**1.** Deposit Sandstone



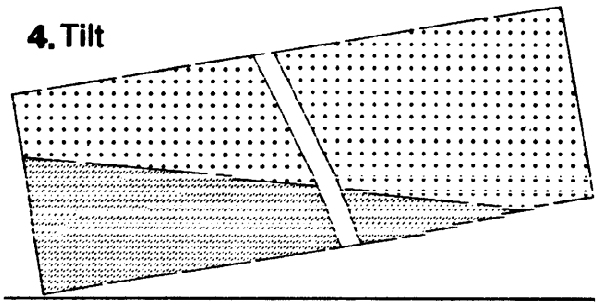
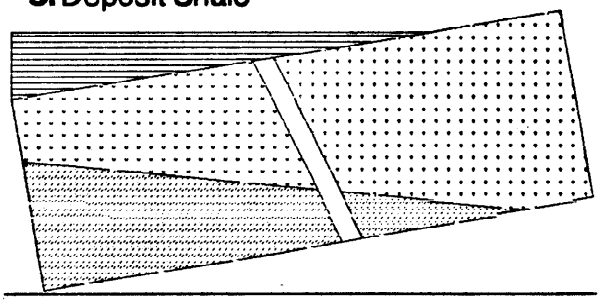
**2.** Intrude Granite



**3.** Intrude Mafic-Igneous



**4.** Tilt



**5.** Deposit Shale