Expert Systems Project Management

- Bruce G. Buchanan
- Reid G. Smith

Sixth National Conference on Artificial Intelligence

Conference Tutorial Program

July, 1987

EXPERT SYSTEMS PROJECT MANAGEMENT

Tutorial No. MA2

Bruce G. Buchanan and Reid G. Smith

PREFACE

It is increasingly important to understand how to turn research in expert systems into actual products and services in a business environment. This involves both a basic grasp of vocabulary and techniques and an understanding of the pragmatics of expert system development. The goal of this introductory tutorial is to enable designers and managers to understand the criteria for making decisions about expert system projects, including differentiating interesting prototypes from finished products. Several examples of successful (and other) projects will be used to illustrate the tutorial.

CONTENT

The tutorial is divided into eight sections.

1. Introduction: What Is An Expert System? We present the basic capabilities and architectural characteristics that distinguish expert systems from traditional programs.

2. Detailed Example: We use the Dipmeter Advisor system to demonstrate the basic technology. Both the system itself and the process of building a commercially viable version will be discussed.

3. Technical Details I: Representation--Three practical methods for encoding knowledge are logic, rules, and objects. Some of their relative strengths and weaknesses will be compared.

4. Technical Details II: Inference--Reasoning methods use the contents of a knowledge base to make inferences that solve a problem. Some of the methods to be discussed are: forward and backward chaining, event-driven inference, and inexact inference.

5. Technical Details III: Shell Systems--We discuss the utility of tools and shells, give examples, and develop the idea of an integrated development environment for expert systems. Criteria for selecting shells for development are contrasted with criteria for selecting run-time environments.

6. Pragmatics: Issues such as "How to choose a problem?", "How to select a shell?", "How to staff a project?", "What performance to expect?", and "What cost to expect?" are addressed. These issues go beyond the technical capabilities of AI methods to include economic, sociological, and political considerations.

7. Validation: We discuss methods for expert system testing and quality assurance. A clear statement of the problem to be solved is a major step in understanding what to validate but there are several methods for demonstrating that a system "solves the problem".

8. Future Potential and Current Assessment: We discuss current research areas, and likely progress over the next five years. An assessment of the current state of the art and its successes completes the tutorial.

INTENDED AUDIENCE

This tutorial is addressed to people who intend to manage or participate in the development of expert systems. It is also appropriate for those who need a basic understanding of the technology--the state of the art, suitable applications, considerations in tool purchase, current and potential impact. There are no prerequisites for the tutorial.

After this tutorial, attendees should be able to understand and participate in the decisions that must be made during expert system development. They will be familiar with the vocabulary and issues. They will understand the criteria involved in determining the suitability of problems, how to choose appropriate tools, realize performance and cost

expectations, basic issues in technology transfer for expert systems--in summary, the pragmatics of expert system development.

SPEAKERS

Dr. Bruce G. Buchanan, Professor of Computer Science Research and Professor of Medicine (by courtesy) at Stanford University, received his B.A. in Mathematics from Ohio Wesleyan University (1961), and his M.A. and Ph.D. from the Department of Philosophy at Michigan State University (1966). He was Instructor of Philosophy at Michigan State University and then, in 1966, joined Stanford as a Research Associate in Computer Science. In 1976 he was appointed to his present position. Professor Buchanan was a major contributor to the DENDRAL, Meta-DENDRAL, and MYCIN programs. He is currently working on several projects, including the interpretation of data about the 3-dimensional structure of proteins, constraint satisfaction in project management, and knowledge acquisition by various methods. Professor Buchanan is on the editorial boards of Artificial Intelligence, Expert Systems, Machine Learning, and The Journal of Automated Reasoning, and is Secretary-Treasurer of the American Association for Artificial Intelligence.

Dr. Reid G. Smith is a research manager for knowledge-based computer-aided engineering at Schlumberger Palo Alto Research. He received his Ph.D. from Stanford University (1979) and the M.S. from Carleton University (1969). He is the author of several papers on knowledge-based systems, object-oriented programming, man-machine interfaces, distributed problem solving, machine learning, and signal processing. He is also the author of A Framework For Distributed Problem Solving (UMI Research Press, 1981). Dr. Smith serves on the editorial board of Expert Systems: Research and Applications. His current interests lie in knowledge-intensive development environments and machine learning. He has lectured extensively on the pragmatics of knowledge-based system design and application.

Expert Systems Project Management

Bruce G. Buchanan
Stanford University

Reid G. Smith
Schlumberger

1. **Introduction**

2. **Detailed Example**
   Dipmeter Advisor

3. **Technical Details for Managers**

   - **Representation**
   - **Inference**

<Questions and Break>

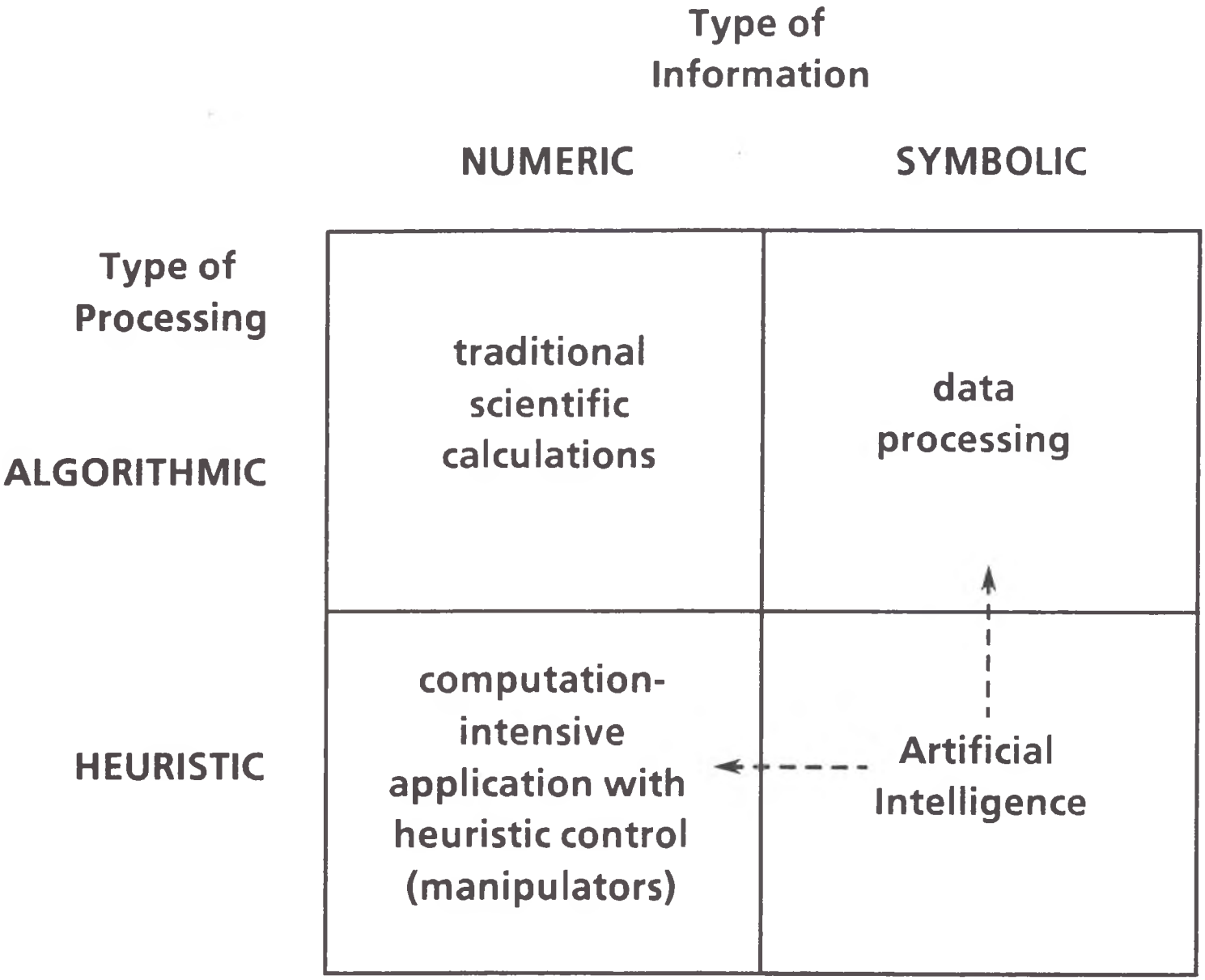
   - **Hardware/Software/Shells**

4. **Pragmatics**

5. **Summary: State of the Art**

<Questions>

FOUR AREAS OF COMPUTING



WHAT ARE EXPERT SYSTEMS?

1. AI Programs <methodology>
   symbolic information
   heuristic processing

2. Expert-level performance <quality>

3. Flexible <design>

4. Understandable <design>

5. Key Element = <implementation>
   separation of knowledge base
   from inference procedures

Why Build An Expert System?

- Replicate Expertise
- Combine Expertise

Motivations

avoid delays

distribute expertise to remote sites

make expertise available to less experienced personnel

preserve corporate knowledge

increase consistency of decisions

handle routine reasoning and bookkeeping

leave an "audit trail"

Some Applications of Expert Systems

Scheduling

Westinghouse — Plan manufacturing steps in a plant to avoid bottlenecks and delays

Configuration

Digital — Translate customers' orders for computer systems into shipping orders

Route Planning

U.S. Air Force — Plan an aircraft's route from base to target and back to avoid detection and threats

Loading

U.S. Army — Design loading plan of cargo and equipment into aircraft of different types

Equipment Design

Delco — Design special-purpose, low-voltage electric motors

Therapy Management

Stanford Medical Center — Assist in managing multi-step chemotherapy for cancer patients

Some Applications of Expert Systems

Portfolio Management

First Financial Planning Systems (Travelers Insurance) — Analyze an individual's financial situation and recommend investments

Equipment Tuning

Lawrence Livermore National Laboratory — Specify parameter settings to align a mass spectrometer

Intelligent Front Ends

Shell Oil — Advise persons on selecting and using subroutines in large Fortran library

Training

Elf-Aquitaine Oil Company — Train drillers to identify causes and repair drill bit sticking in oil wells

Some Applications of Expert Systems

Equipment Diagnosis

General Motors — Determine causes of noises and recommend repairs

Data Interpretation

Schlumberger — Interpret down-hole data from oil well boreholes to assist in prospecting

Risk Assessment

St. Paul Insurance Co. — Assess risk of insuring large commercial clients

Monitoring

IBM — Monitor operations of MVS operating system

Screening

U.S. Environmental Protection Agency — Screen requests for information with respect to confidentiality

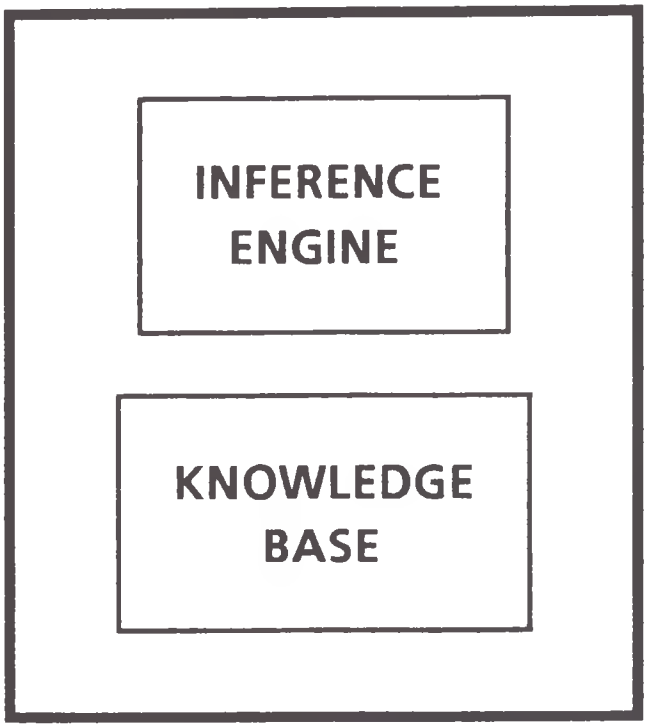
Troubleshooting In Manufacturing

Hewlett Packard — Diagnose causes of problems in photolithography steps of wafer fabrication

Crop Management

Virginia Polytechnical Institute — Assist in managing apple orchards

Basic Knowledge-Based System Architecture

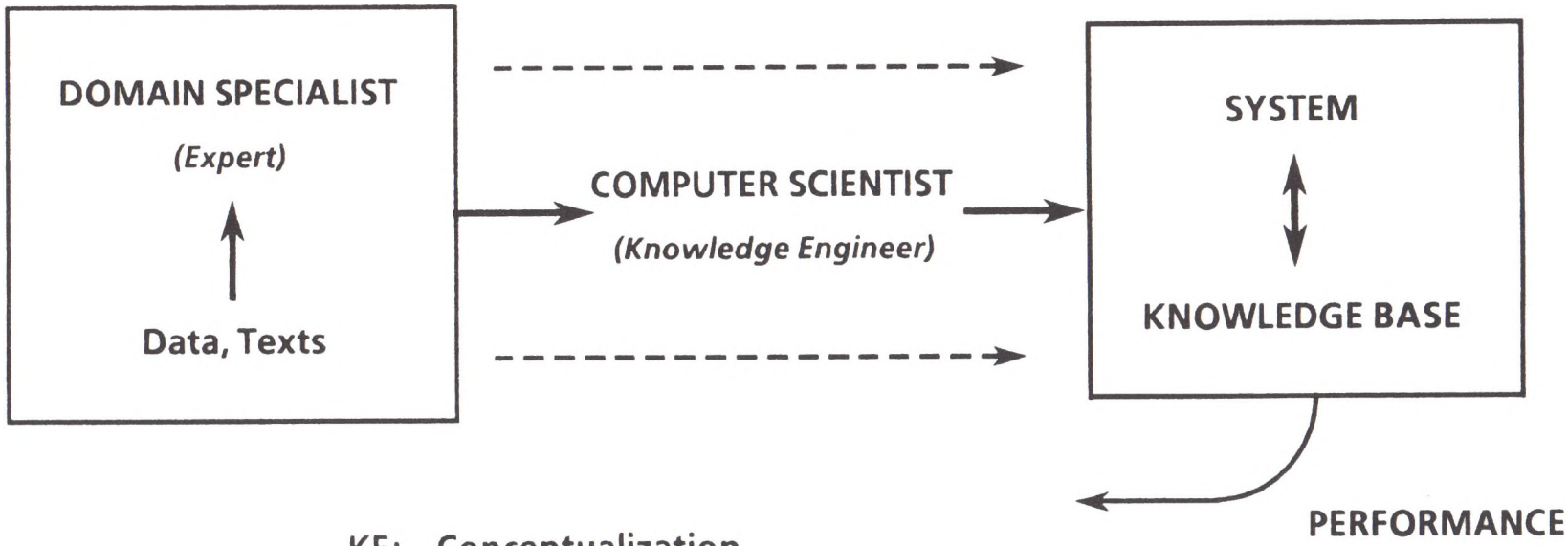


Control System Interpreter

Global Database

Major Architectural Lesson

Knowledge Acquisition



KE: Conceptualization
Construction
Refinement

KE: Using the framework
vs
Designing the framework

Why Automate Any Task?

- **Money**
- **Time**
- **Information**

"The initial overenthusiasm, which inevitably accompanies a project of this scope, can and does make the job harder..."

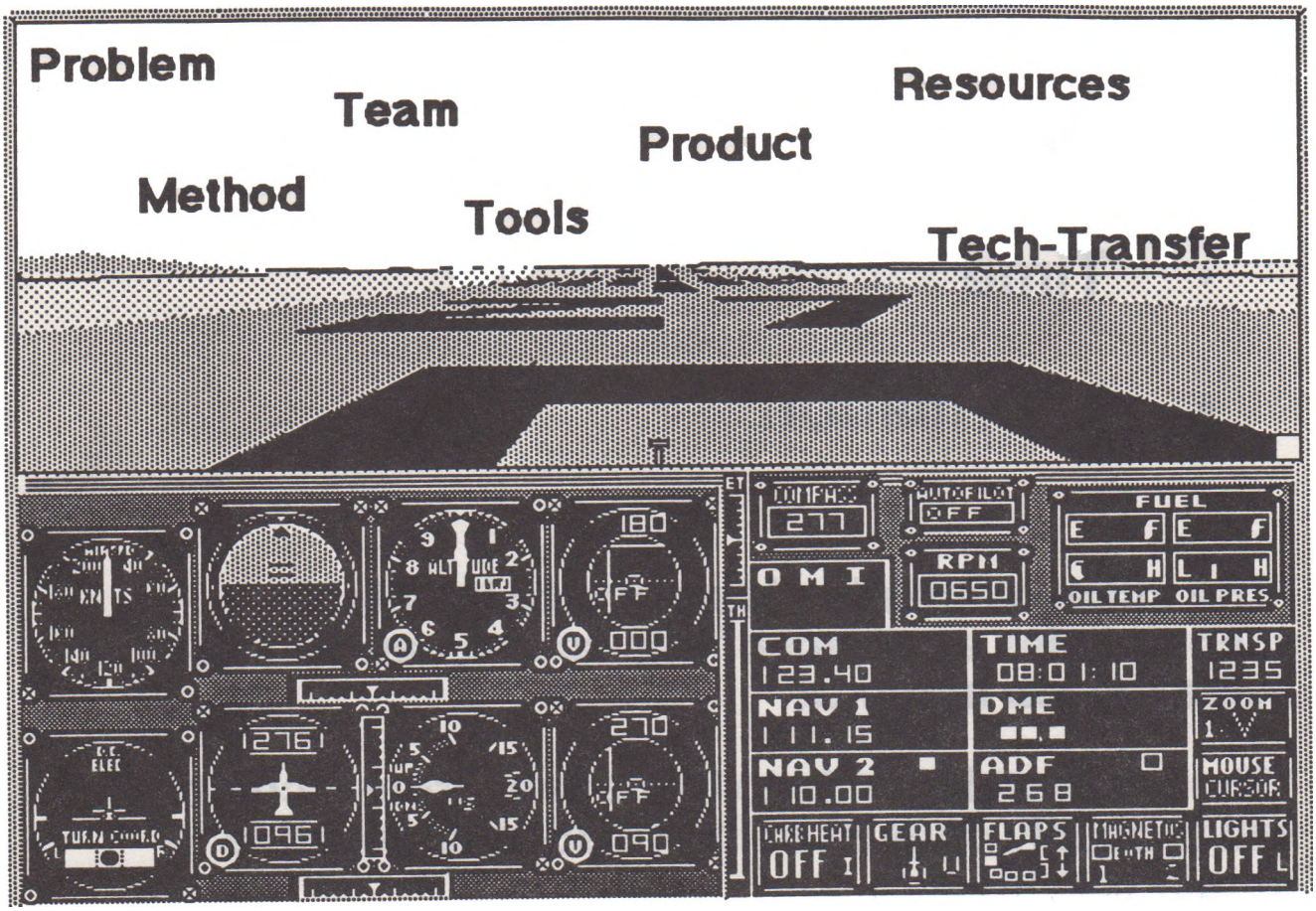
"The greatest benefits to be derived from a computer will probably consist of information impossible to obtain previously..."

"Our experience has shown that the computer is more adaptable to some projects than others..."

"It is impossible to overemphasize the desirability of providing for convenient corrections or deletion of errors in data..."

"The maximum justifiable amount of flexibility for extending or integrating applications must be included in the initial programming..."

— G. M. Sheehan, *Proc. Automatic Data Processing Conf.,* September 1955.



Manager's Choices

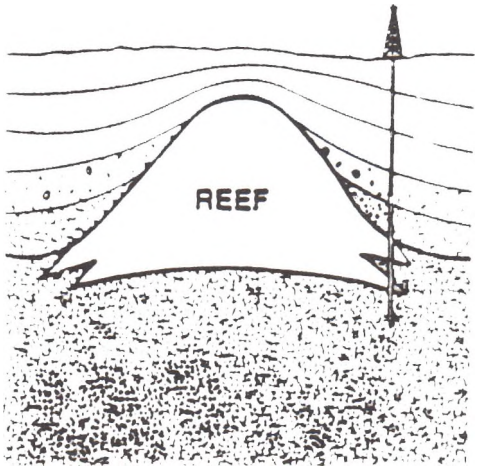
DIPMETER ADVISOR SYSTEM: OVERVIEW

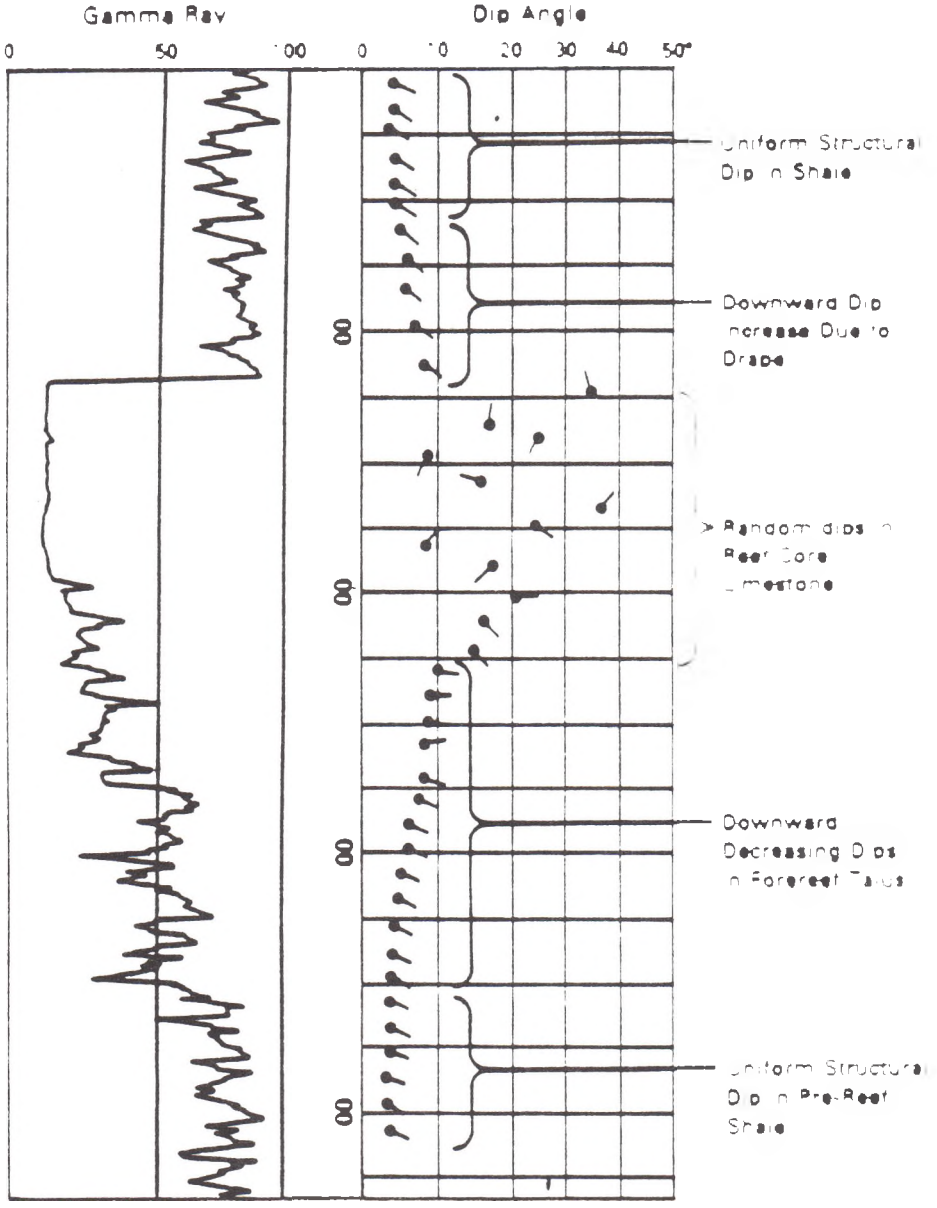
INPUT:

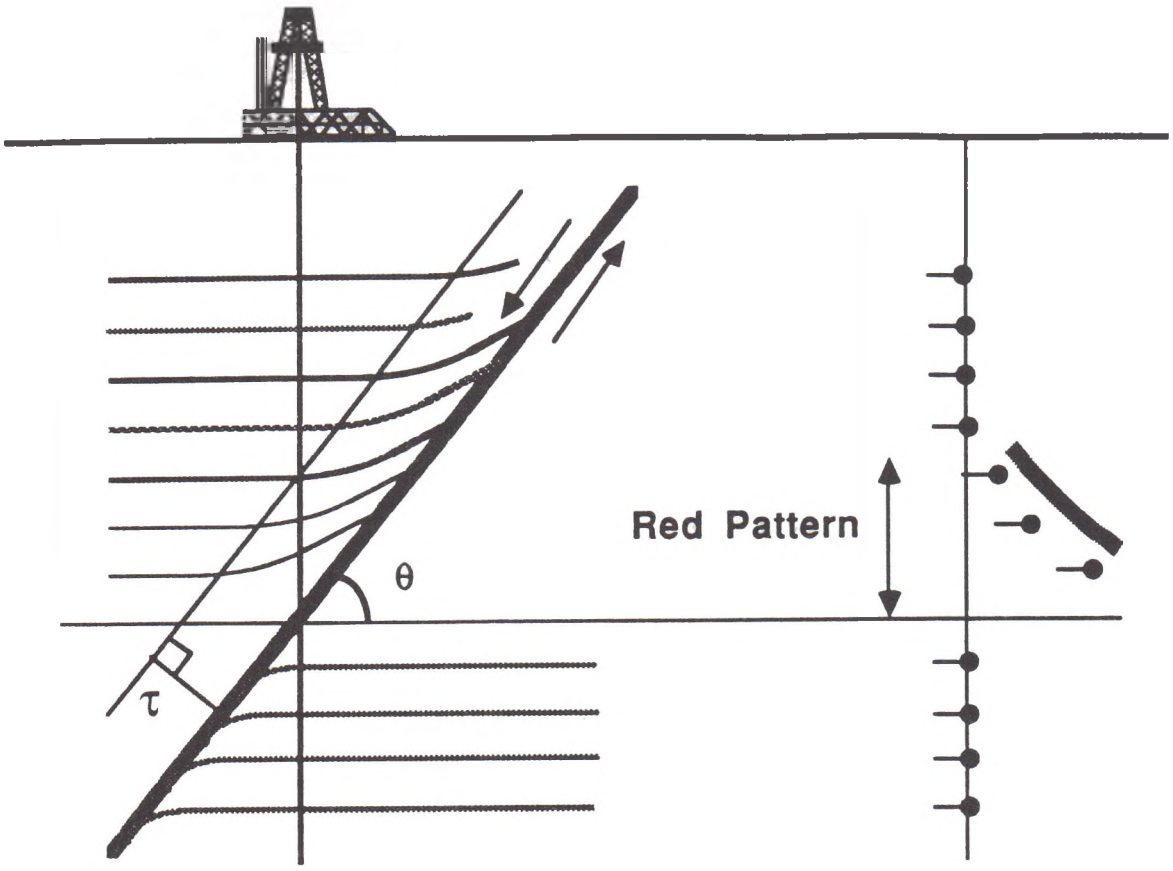
- **Well Logs**
  - **Correlated to indicate subsurface *dip***
  - **Conventional logs**
- **Geological Assertions**
  - **Local area knowledge**
  - **Specific feature knowledge**

OUTPUT:

- **Structural Dip Analysis**
- **Tectonic Feature Analysis**
- **Stratigraphic Feature Analysis**


REEF


Gamma Ray
0
50
100
Dip Angle
0
10
20
30
40
50°
Uniform Structural Dip in Shale
Downward Dip Increase Due to Drape
Random dips in Reef Core Limestone
Downward Decreasing Dips in Forereef Talus
Uniform Structural Dip in Pre-Reef Shale



IF

there exists a normal fault pattern (p), and

there exists a red pattern (p1),

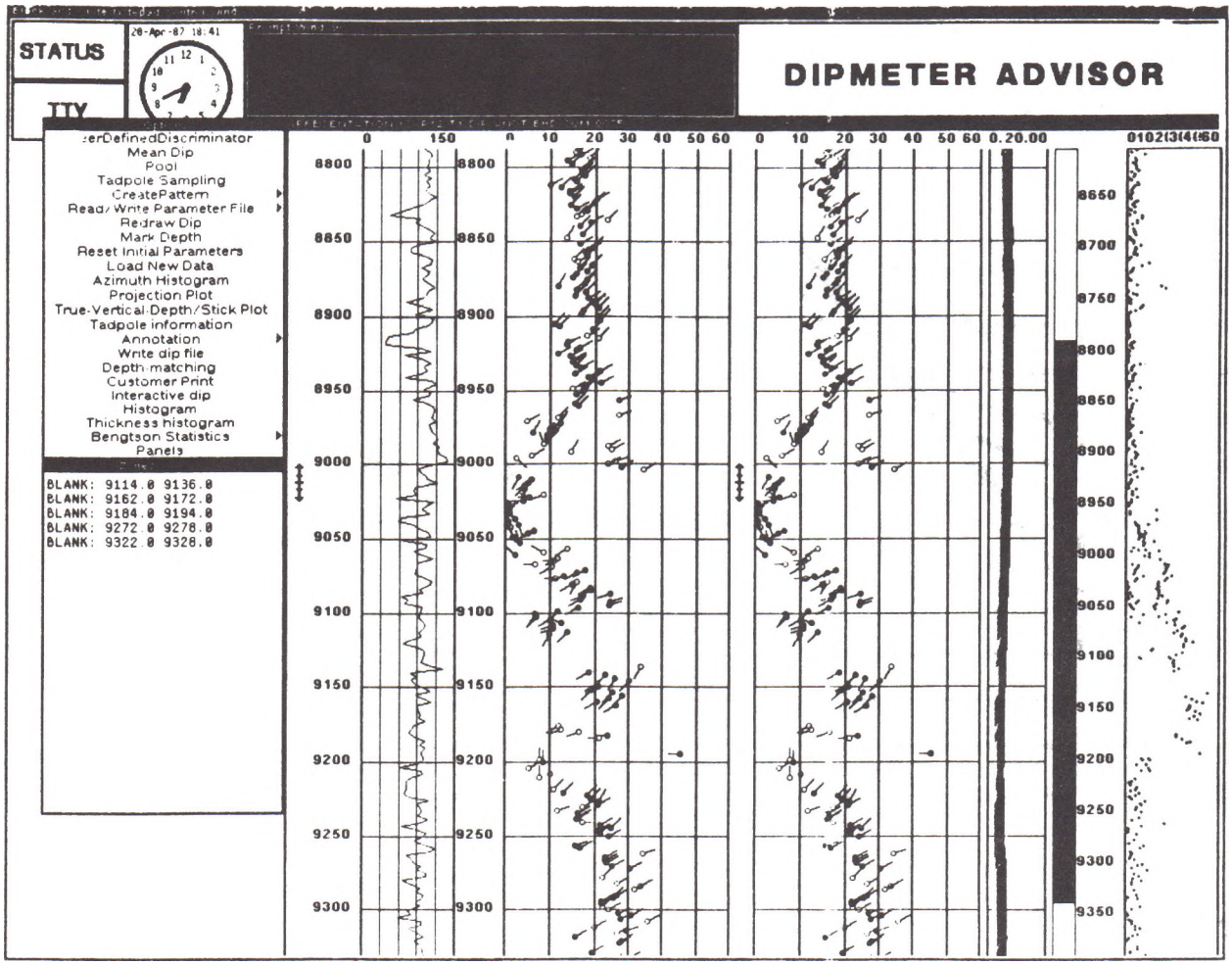
such that the length of p1 < 50 ft., and

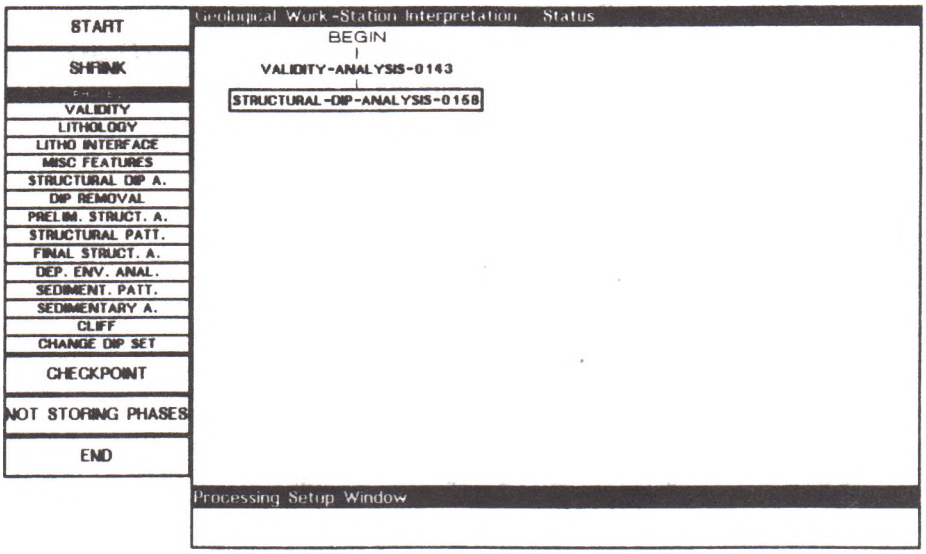
such that p1 is above the fault plane pattern of p,

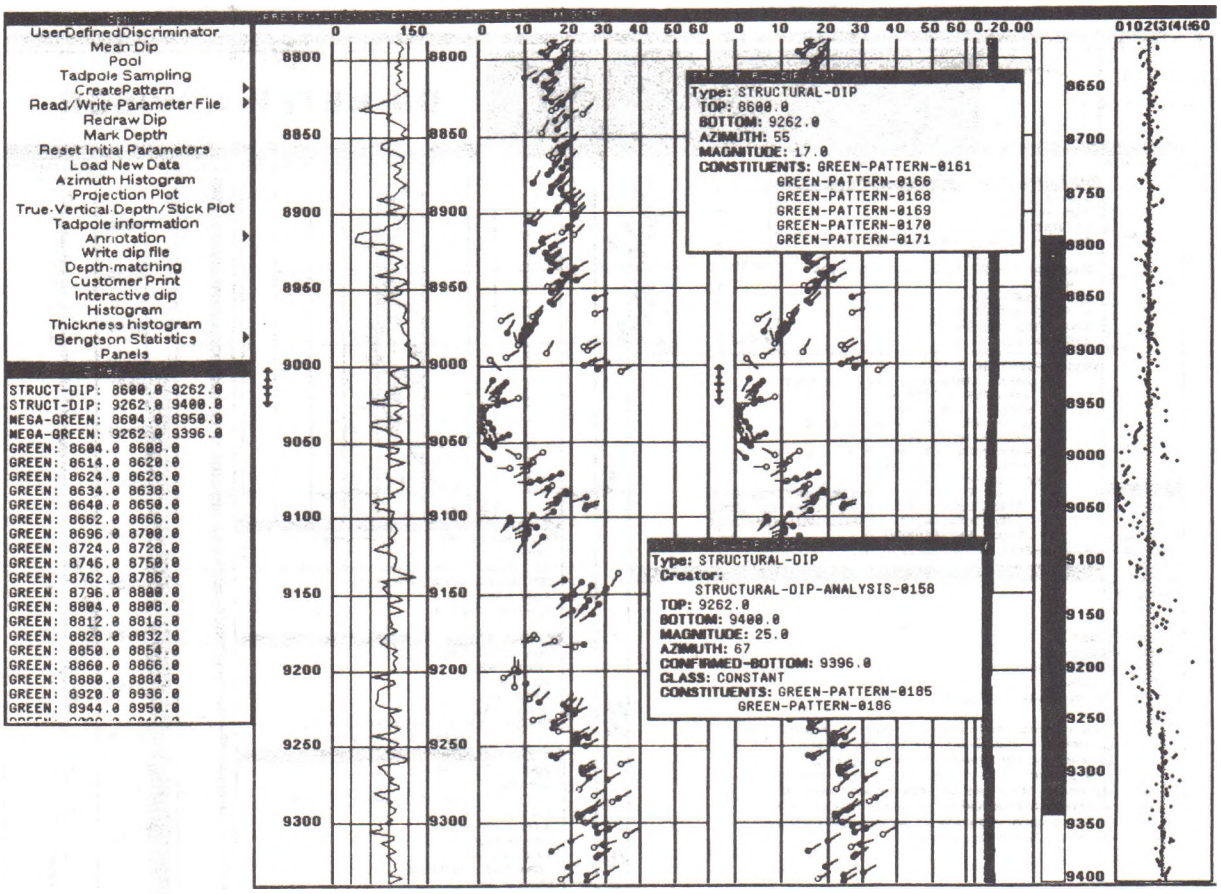
THEN

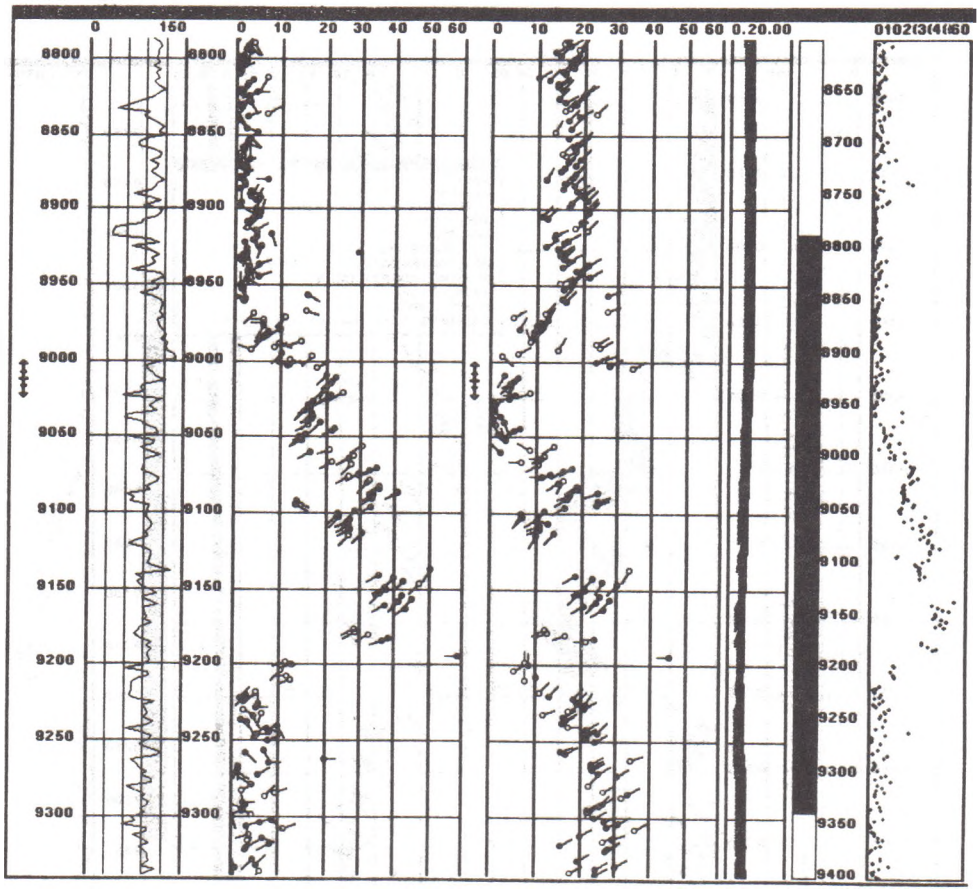
specialize p to be a late fault pattern

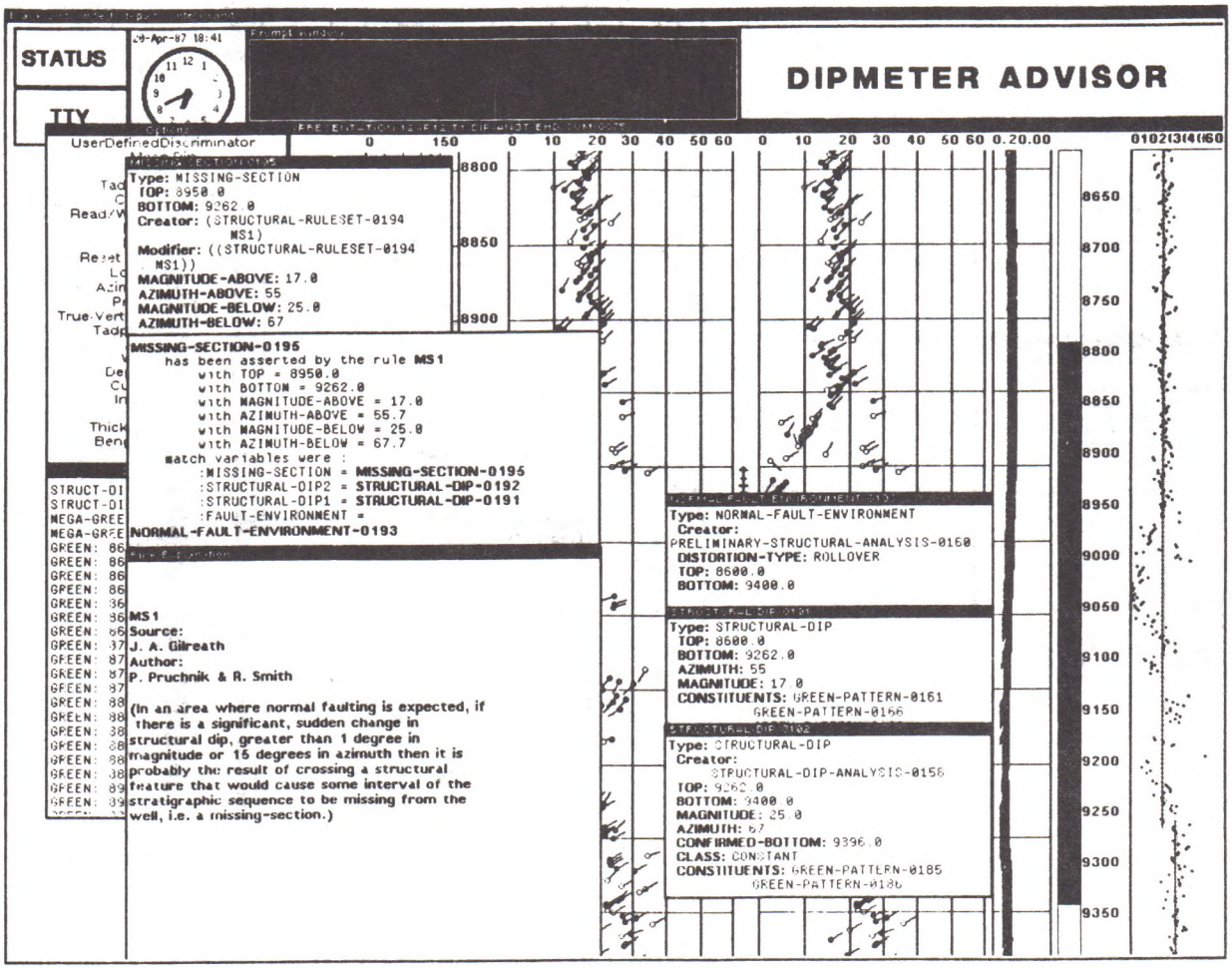
Dipmeter Advisor System
Late Fault Rule

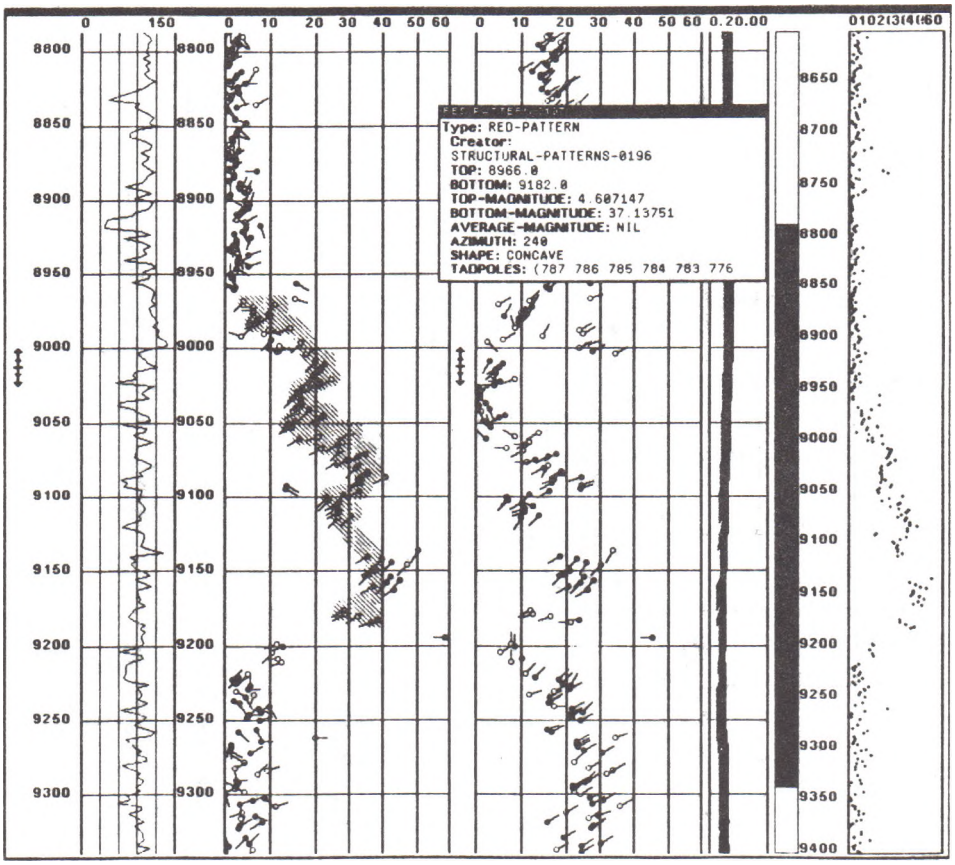

STATUS
TTY
20-Apr-87 18:41
DIPMETER ADVISOR
erDefinedDiscriminator
Mean Dip
Pool
Tadpole Sampling
CreatePattern
Read/Write Parameter File
Redraw Dip
Mark Depth
Reset Initial Parameters
Load New Data
Azimuth Histogram
Projection Plot
True-Vertical-Depth/Stick Plot
Tadpole information
Annotation
Write dip file
Depth-matching
Customer Print
Interactive dip
Histogram
Thickness histogram
Bengtson Statistics
Panels
BLANK: 9114.0 9136.0
BLANK: 9162.0 9172.0
BLANK: 9184.0 9194.0
BLANK: 9272.0 9278.0
BLANK: 9322.0 9328.0
0 150
0 10 20 30 40 50 60
0 10 20 30 40 50 60
0.20.00
8800
8850
8900
8950
9000
9050
9100
9150
9200
9250
9300
8650
8700
8750
8800
8850
8900
8950
9000
9050
9100
9150
9200
9250
9300
9350

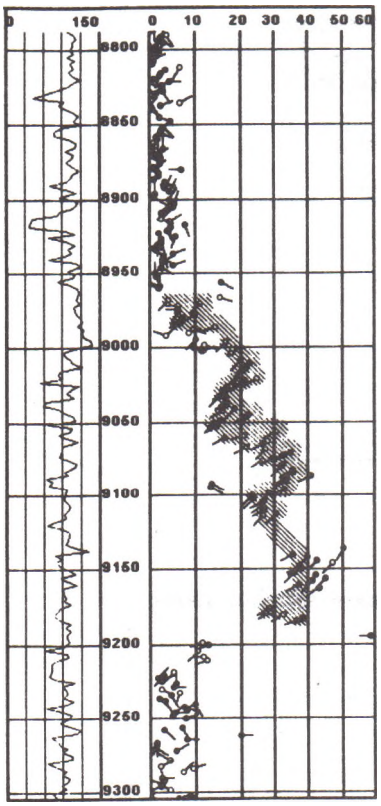

START
SHRINK
VALIDITY
LITHOLOGY
LITHO INTERFACE
MISC FEATURES
STRUCTURAL DIP A.
DIP REMOVAL
PRELIM. STRUCT. A.
STRUCTURAL PATT.
FINAL STRUCT. A.
DEP. ENV. ANAL.
SEDIMENT. PATT.
SEDIMENTARY A.
CLIFF
CHANGE DIP SET
CHECKPOINT
NOT STORING PHASES
END
Geological Work-Station Interpretation Status
BEGIN
VALIDITY-ANALYSIS-0143
STRUCTURAL-DIP-ANALYSIS-0168
Processing Setup Window


UserDefinedDiscriminator
Mean Dip
Pool
Tadpole Sampling
CreatePattern
Read/Write Parameter File
Redraw Dip
Mark Depth
Reset Initial Parameters
Load New Data
Azimuth Histogram
Projection Plot
True-Vertical-Depth/Stick Plot
Tadpole information
Annotation
Write dip file
Depth-matching
Customer Print
Interactive dip
Histogram
Thickness histogram
Bengtson Statistics
Panels
STRUCT-DIP: 8600.0 9262.0
STRUCT-DIP: 9262.0 9400.0
MEGA-GREEN: 8604.0 8950.0
MEGA-GREEN: 9262.0 9396.0
GREEN: 8604.0 8608.0
GREEN: 8614.0 8620.0
GREEN: 8624.0 8628.0
GREEN: 8634.0 8638.0
GREEN: 8640.0 8650.0
GREEN: 8662.0 8666.0
GREEN: 8696.0 8700.0
GREEN: 8724.0 8728.0
GREEN: 8746.0 8750.0
GREEN: 8762.0 8786.0
GREEN: 8796.0 8800.0
GREEN: 8804.0 8808.0
GREEN: 8812.0 8816.0
GREEN: 8828.0 8832.0
GREEN: 8850.0 8854.0
GREEN: 8860.0 8866.0
GREEN: 8880.0 8884.0
GREEN: 8920.0 8936.0
GREEN: 8944.0 8950.0
Type: STRUCTURAL-DIP
TOP: 8600.0
BOTTOM: 9262.0
AZIMUTH: 55
MAGNITUDE: 17.0
CONSTITUENTS: GREEN-PATTERN-0161
GREEN-PATTERN-0166
GREEN-PATTERN-0168
GREEN-PATTERN-0169
GREEN-PATTERN-0170
GREEN-PATTERN-0171
Type: STRUCTURAL-DIP
Creator:
STRUCTURAL-DIP-ANALYSIS-0158
TOP: 9262.0
BOTTOM: 9400.0
MAGNITUDE: 25.0
AZIMUTH: 67
CONFIRMED-BOTTOM: 9396.0
CLASS: CONSTANT
CONSTITUENTS: GREEN-PATTERN-0185
GREEN-PATTERN-0186




STATUS
DIPMETER ADVISOR
Type: MISSING-SECTION
TOP: 8950.0
BOTTOM: 9262.0
Creator: (STRUCTURAL-RULESET-0194 MS1)
Modifier: ((STRUCTURAL-RULESET-0194 MS1))
MAGNITUDE-ABOVE: 17.0
AZIMUTH-ABOVE: 55
MAGNITUDE-BELOW: 25.0
AZIMUTH-BELOW: 67
MISSING-SECTION-0195
has been asserted by the rule MS1
with TOP = 8950.0
with BOTTOM = 9262.0
with MAGNITUDE-ABOVE = 17.0
with AZIMUTH-ABOVE = 55.7
with MAGNITUDE-BELOW = 25.0
with AZIMUTH-BELOW = 67.7
match variables were :
:MISSING-SECTION = MISSING-SECTION-0195
:STRUCTURAL-DIP2 = STRUCTURAL-DIP-0192
:STRUCTURAL-DIP1 = STRUCTURAL-DIP-0191
:FAULT-ENVIRONMENT =
NORMAL-FAULT-ENVIRONMENT-0193
MS1
Source:
J. A. Gilreath
Author:
P. Pruchnik & R. Smith
(In an area where normal faulting is expected, if there is a significant, sudden change in structural dip, greater than 1 degree in magnitude or 15 degrees in azimuth then it is probably the result of crossing a structural feature that would cause some interval of the stratigraphic sequence to be missing from the well, i.e. a missing-section.)
Type: NORMAL-FAULT-ENVIRONMENT
Creator:
PRELIMINARY-STRUCTURAL-ANALYSIS-0160
DISTORTION-TYPE: ROLLOVER
TOP: 8600.0
BOTTOM: 9400.0
Type: STRUCTURAL-DIP
TOP: 8600.0
BOTTOM: 9262.0
AZIMUTH: 55
MAGNITUDE: 17.0
CONSTITUENTS: GREEN-PATTERN-0161 GREEN-PATTERN-0166
Type: STRUCTURAL-DIP
Creator:
STRUCTURAL-DIP-ANALYSIS-0156
TOP: 9262.0
BOTTOM: 9400.0
MAGNITUDE: 25.0
AZIMUTH: 67
CONFIRMED-BOTTOM: 9396.0
CLASS: CONSTANT
CONSTITUENTS: GREEN-PATTERN-0185 GREEN-PATTERN-0186


Type: RED-PATTERN
Creator:
STRUCTURAL-PATTERNS-0196
TOP: 8966.0
BOTTOM: 9182.0
TOP-MAGNITUDE: 4.607147
BOTTOM-MAGNITUDE: 37.13751
AVERAGE-MAGNITUDE: NIL
AZIMUTH: 240
SHAPE: CONCAVE
TADPOLES: (787 786 785 784 783 776



Type: GROWTH-FAULT
TOP: 9182.0
BOTTOM: 9262.0
Creator: (NORMAL-FAULT-RULESET-0239 . NFR3A)
Modifier: ((NORMAL-FAULT-RULESET-0239 . NFR9A)
(NORMAL-FAULT-RULESET-0239 . NFR3A))
STRIKE: "SSE-NNW (150 deg)"
DIRECTION-TO-DOWNTHROWN-BLOCK: "ENE (60 deg)"
MIN-FAULT-CUT: 216.0
ILLUSTRATION:

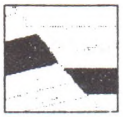
Normal Fault

rollover

NORMAL-FAULT-0240
has been asserted by the rule NFR3A
with TOP = 8950.0
with BOTTOM = 9262.0
match variables were :
:UNCONFORMITY = UNCONFORMITY-0241
:NORMAL-FAULT = NORMAL-FAULT-0240
:RED = RED-PATTERN-0197
:MISSING-SECTION = MISSING-SECTION-0196
has been modified by the rule NFR9A

NFR3A
Source:
J. A. Gilreath
Author:
P. Pruchnik & R. Smith, altered by D. Hammock 1-17-85

(If there is a missing section such that the magnitude of structural dip below the missing section is greater than the magnitude above, by at least 1 degree, but the azimuth of dip does not reverse- see rule nfr2a- and there is a red pattern associated with the missing section then the missing section is probably the result of the well crossing a normal fault or an angular unconformity.)



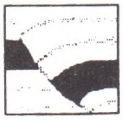
Normal Fault

Angular Unconformity

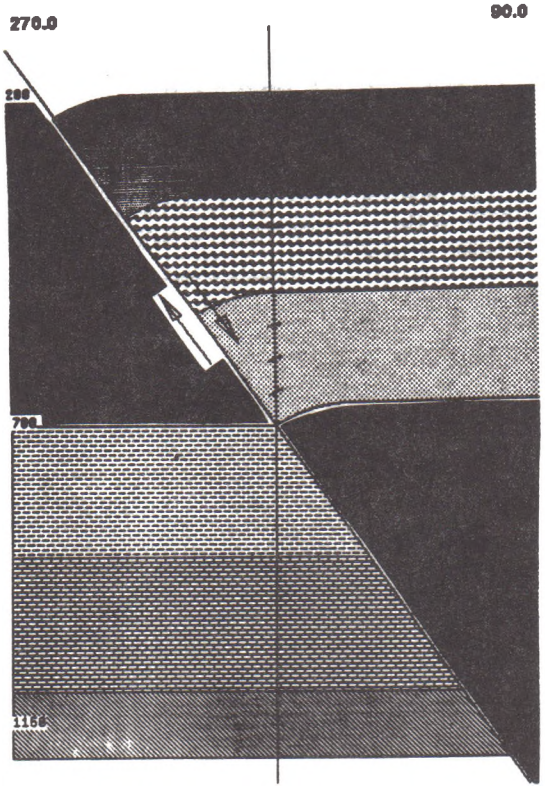
NFR9A
Source:
J. A. Gilreath
Author:
P. Pruchnik & R. Smith, altered by D. Hammock 10-25-84

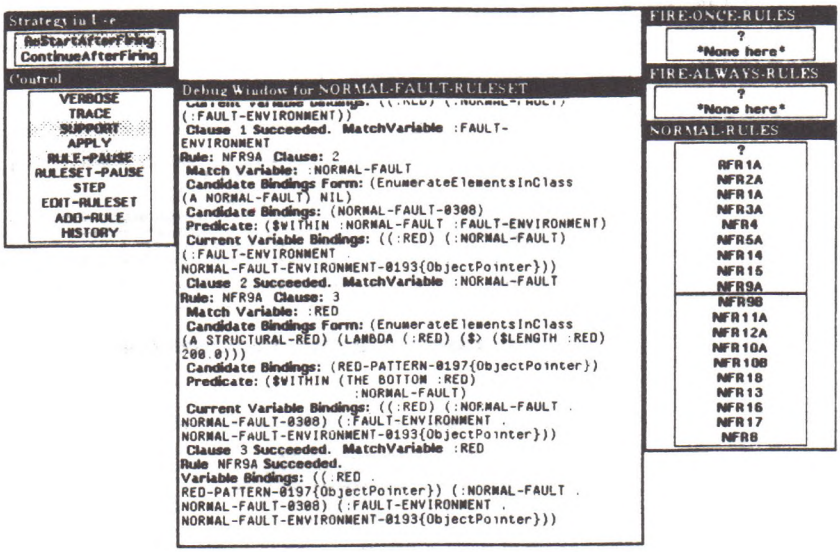
(In a region where the primary type of distortion is rollover, if there is a normal fault with a red pattern greater than 200 ft. in length associated with it then the fault is probably a growth fault; the fault cuts the well somewhere below the bottom of the red pattern, the strike of the fault is perpindicular to the azimuth of the pattern, the direction to the downthrown block is opposite the azimuth of the pattern & the length of the pattern gives a rough number for the minimum cut of the fault.)

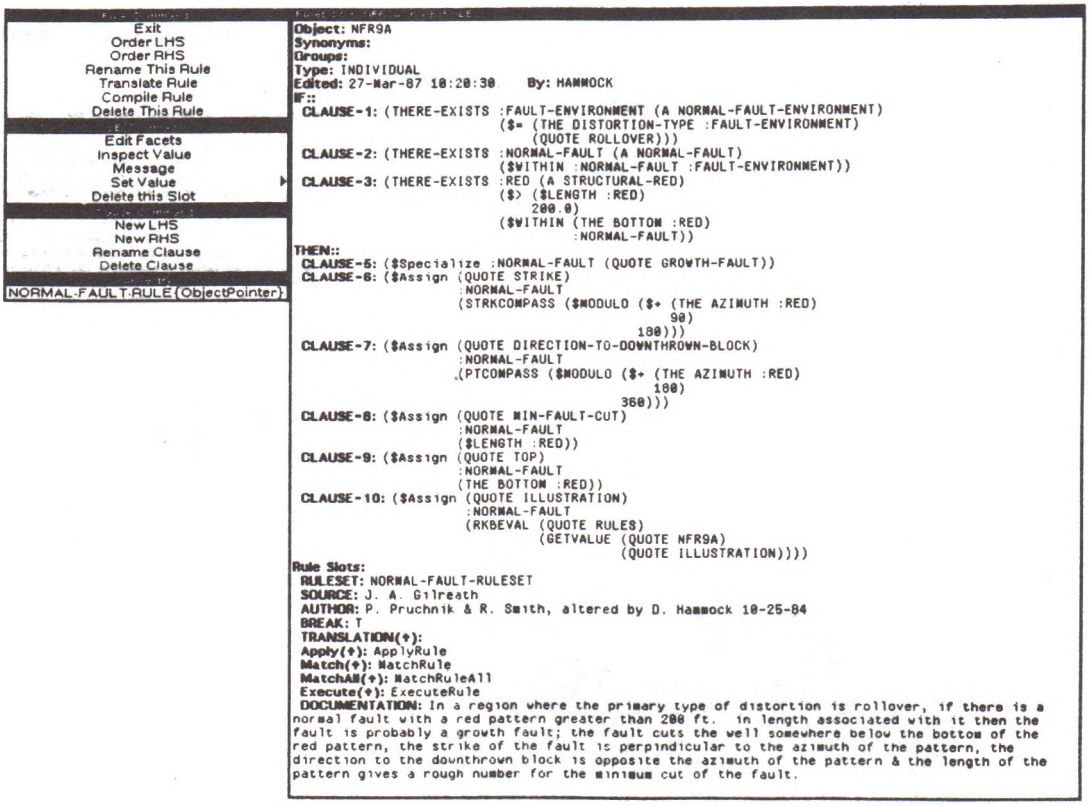
Normal Fault



rollover




Strategy Rule
ReStartAfterFiring
ContinueAfterFiring
Control
VERBOSE
TRACE
SUPPORT
APPLY
RULE-PAUSE
RULESET-PAUSE
STEP
EDIT-RULESET
ADD-RULE
HISTORY
Debug Window for NORMAL-FAULT-RULESET
(:FAULT-ENVIRONMENT))
Clause 1 Succeeded. MatchVariable :FAULT-ENVIRONMENT
Rule: NFR9A Clause: 2
Match Variable: :NORMAL-FAULT
Candidate Bindings Form: (EnumerateElementsInClass (A NORMAL-FAULT) NIL)
Candidate Bindings: (NORMAL-FAULT-0308)
Predicate: ($WITHIN :NORMAL-FAULT :FAULT-ENVIRONMENT)
Current Variable Bindings: ((:RED) (:NORMAL-FAULT . NORMAL-FAULT-0308) (:FAULT-ENVIRONMENT . NORMAL-FAULT-ENVIRONMENT-0193{ObjectPointer}))
Clause 2 Succeeded. MatchVariable :NORMAL-FAULT
Rule: NFR9A Clause: 3
Match Variable: :RED
Candidate Bindings Form: (EnumerateElementsInClass (A STRUCTURAL-RED) (LAMBDA (:RED) ($> ($LENGTH :RED) 200.0)))
Candidate Bindings: (RED-PATTERN-0197{ObjectPointer})
Predicate: ($WITHIN (THE BOTTOM :RED) :NORMAL-FAULT)
Current Variable Bindings: ((:RED) (:NORMAL-FAULT . NORMAL-FAULT-0308) (:FAULT-ENVIRONMENT . NORMAL-FAULT-ENVIRONMENT-0193{ObjectPointer}))
Clause 3 Succeeded. MatchVariable :RED
Rule NFR9A Succeeded.
Variable Bindings: ((:RED . RED-PATTERN-0197{ObjectPointer}) (:NORMAL-FAULT . NORMAL-FAULT-0308) (:FAULT-ENVIRONMENT . NORMAL-FAULT-ENVIRONMENT-0193{ObjectPointer}))
FIRE-ONCE-RULES
?
None here
FIRE-ALWAYS-RULES
?
None here
NORMAL-RULES
?
NFR1A
NFR2A
NFR1A
NFR3A
NFR4
NFR5A
NFR14
NFR15
NFR9A
NFR9B
NFR11A
NFR12A
NFR10A
NFR10B
NFR18
NFR13
NFR16
NFR17
NFR8


Exit
Order LHS
Order RHS
Rename This Rule
Translate Rule
Compile Rule
Delete This Rule
Edit Facets
Inspect Value
Message
Set Value
Delete this Slot
New LHS
New RHS
Rename Clause
Delete Clause
NORMAL-FAULT-RULE{ObjectPointer}
Object: NFR9A
Synonyms:
Groups:
Type: INDIVIDUAL
Edited: 27-Mar-87 10:20:30 By: HAMMOCK
IF::
CLAUSE-1: (THERE-EXISTS :FAULT-ENVIRONMENT (A NORMAL-FAULT-ENVIRONMENT) ($= (THE DISTORTION-TYPE :FAULT-ENVIRONMENT) (QUOTE ROLLOVER)))
CLAUSE-2: (THERE-EXISTS :NORMAL-FAULT (A NORMAL-FAULT) ($WITHIN :NORMAL-FAULT :FAULT-ENVIRONMENT))
CLAUSE-3: (THERE-EXISTS :RED (A STRUCTURAL-RED) ($> ($LENGTH :RED) 200.0) ($WITHIN (THE BOTTOM :RED) :NORMAL-FAULT))
THEN::
CLAUSE-5: ($Specialize :NORMAL-FAULT (QUOTE GROWTH-FAULT))
CLAUSE-6: ($Assign (QUOTE STRIKE) :NORMAL-FAULT (STRKCOMPASS ($MODULO ($+ (THE AZIMUTH :RED) 90) 180)))
CLAUSE-7: ($Assign (QUOTE DIRECTION-TO-DOWNTHROWN-BLOCK) :NORMAL-FAULT (PTCOMPASS ($MODULO ($+ (THE AZIMUTH :RED) 180) 360)))
CLAUSE-8: ($Assign (QUOTE MIN-FAULT-CUT) :NORMAL-FAULT ($LENGTH :RED))
CLAUSE-9: ($Assign (QUOTE TOP) :NORMAL-FAULT (THE BOTTOM :RED))
CLAUSE-10: ($Assign (QUOTE ILLUSTRATION) :NORMAL-FAULT (RKBEVAL (QUOTE RULES) (GETVALUE (QUOTE NFR9A) (QUOTE ILLUSTRATION))))
Rule Slots:
RULESET: NORMAL-FAULT-RULESET
SOURCE: J. A. Gilreath
AUTHOR: P. Pruchnik & R. Smith, altered by D. Hammock 10-25-84
BREAK: T
TRANSLATION(↑):
Apply(↑): ApplyRule
Match(↑): MatchRule
MatchAll(↑): MatchRuleAll
Execute(↑): ExecuteRule
DOCUMENTATION: In a region where the primary type of distortion is rollover, if there is a normal fault with a red pattern greater than 200 ft. in length associated with it then the fault is probably a growth fault; the fault cuts the well somewhere below the bottom of the red pattern, the strike of the fault is perpindicular to the azimuth of the pattern, the direction to the downthrown block is opposite the azimuth of the pattern & the length of the pattern gives a rough number for the minimum cut of the fault.

DIPMETER ADVISOR SYSTEM: ACCOMPLISHMENTS

- Consistent, high-quality interpretations *within areas of expertise*
- Vehicle for codification of interpretation expertise
- Provocation of serious discussion among experts
- Laboratory for interpretation experimentation and investigation
- Powerful interactive workbench supporting manual interpretation

Example: Dipmeter Advisor System

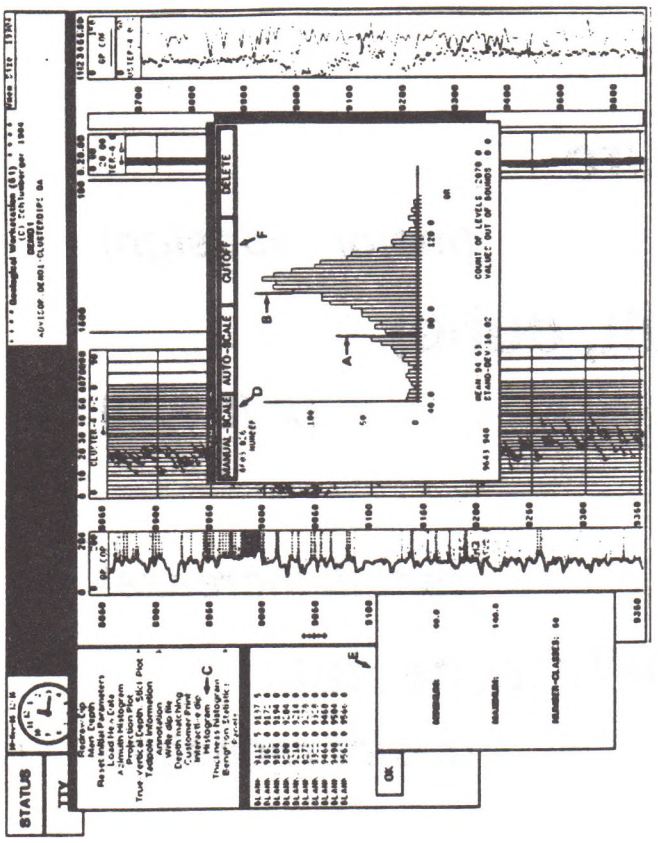
Evolution

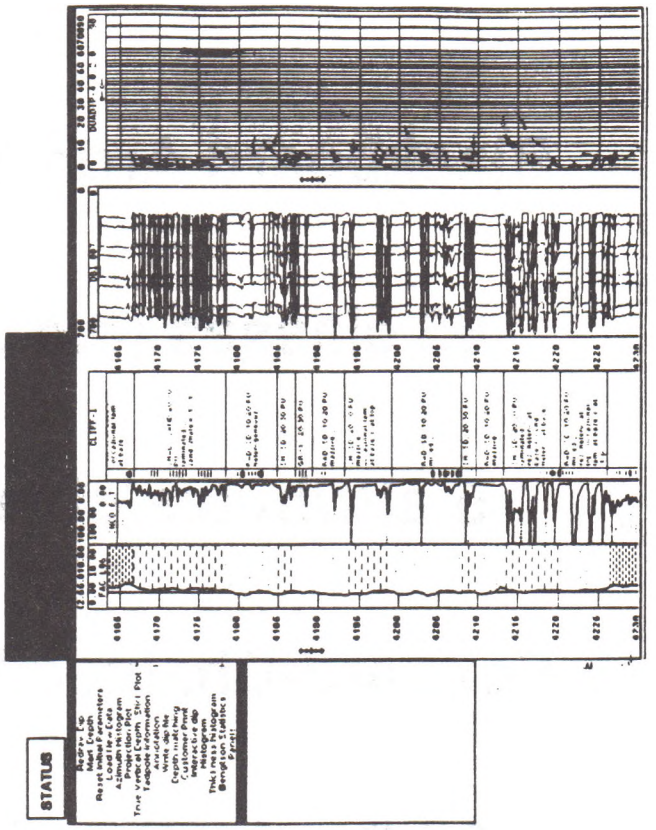
Dipmeter Advisor ➠ Geological Workstation

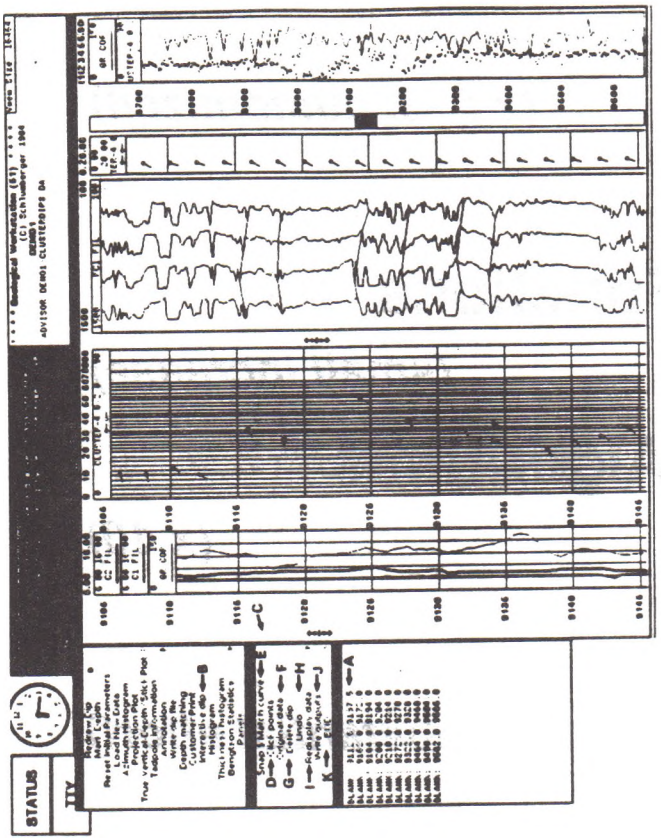
- **Scope**
- **Precision**
- **Effort**

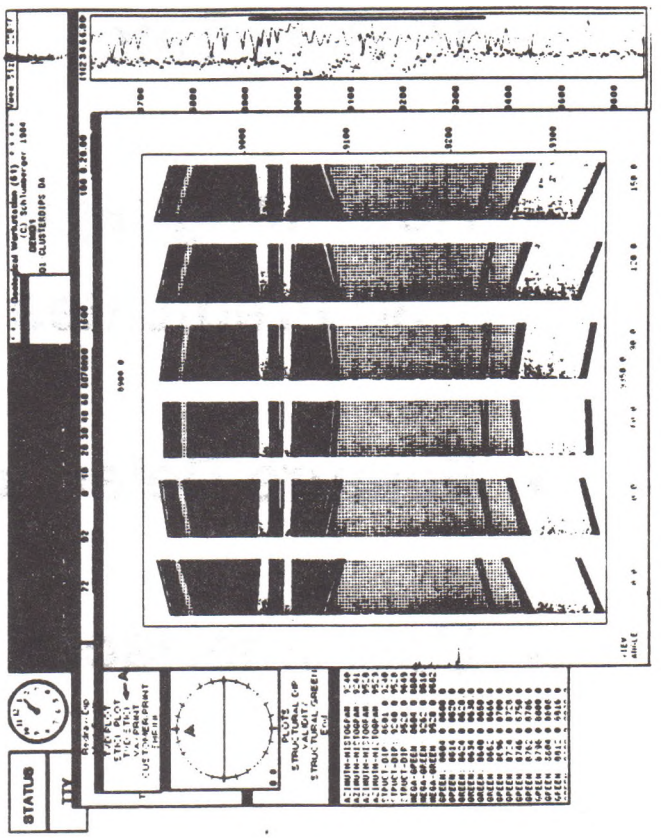
  prototype ⇨ fielded system ⇨ current system
  distribution of code

- **Style of Doing Business**









Dipmeter AdvisorPrototype Fact Sheet

- **Blackboard Architecture**

  (major redesign)

  65 classes (*e.g.*, fault, dune)

  (x 10 - 20)

  5 attributes/object

  (small increase)

- **Forward-Chained Rule Interpreter**

  (major redesign)

  90 Production Rules

  (x 2 -3, customized)

  15 Rule Sets

  (x 2 -3, customized)

  - **Rule Language**

    30 Predicates & Functions

    (small increase)

  - **Feature Detection Algorithms**

    (x 10)

  - **User Interface**

    (substantial effort)

DIPMETER ADVISOR SYSTEM CODE

Inference Engine:	*8%*
Knowledge Base:	*22%*
Feature Detection:	*13%*
User Interface:	*42%*
Support Environment:	*15%*

Example: Dipmeter Advisor System

Features

- **I/O**

  natural interaction style, vocabulary

  graphical output

  mouse input

  explanation (text & graphics)

  precomputed graphics vs generated graphics

- **Customized For Client**

  I/O

  methods

Example: Dipmeter Advisor System

Architecture

- **Interactive Assistant**

  user control of tasks, I/O modes
  hypothetical variations
  volunteered data
  modifications to conclusions

- **Simple Representation & Inference**

- **Integration**

  objects
  rules
  rulesets
  procedures
    signal processing + symbolic inference

- **Flexibility**

Knowledge Representation

Desiderata

- **Expressive Power**
  (*e.g.*, uncertainty)

- **Efficiency**
  human understandability
  computational tractability

- **Extensibility**

- **Flexibility**

... Knowledge Programming

Knowledge Representation

Ways to Model a Domain

- **Action-Centered**
  how to ...
  diagnose meningitis
  detect a late fault

  inferences
  procedures (*e.g.*, optimization)

- **Object-Centered**
  what is ...
  meningitis
  a late fault

  concept descriptions
  relations, (*e.g.*, taxonomies)

Action-Centered Paradigm: Rules & Logic

- **Primitive Unit**

  Fact

- **Primitive Action**

  Draw Conclusion From Facts ... Take Arbitrary Action

THINKING IN RULES

Situation / Action

if temp > 300C **then** turn off boiler.

Premise / Conclusion

if stain is grampos **then** organism is strep.

Antecedent / Consequent

if x is a dog **then** x is an animal

XCON Example Rule

A rule-based program that configures Vax and PDP-11 computer systems [~3000 rules in OPS5 shell; used for 20,000 orders (Jan 84)]

IF:

- the most current activity context is distributing massbus devices, and
- there is a single-port disk that has not been assigned to a massbus, and
- there are no unassigned dual-port disk drives and the number of devices that each massbus should support is known, and
- there is a massbus that has been assigned at least one disk drive and that should support additional disk drives, and
- the type of cable needed to connect the disk drive to the previous device on the massbus is known

THEN:

- assign the disk drive to the massbus

MYCIN Example Rule

An early rule-based program that diagnoses bacteremias

Rule 27

IF:

the gram stain of the organism is gram negative,
and
the morphology of the organism is rod, and
the aerobicity of the organism is anaerobic

THEN:

there is suggestive evidence (.7) that the identity of the organism is Bacteroides

Action-Centered Paradigm: Rules & Logic

Representation of Facts

Feature Vector
Attribute-Value Pairs
Attribute-Value-Object Triples
+ degree of certainty

Relations Among Facts

A and B implies C

... with certainty x

For All x, y.
f(x) and g(y) implies h(x,y)

... with certainty in f,g

Rules & Logic: Difficulties

- **Temporal Relations**
- **Sequencing ... Procedures**
- **Modularity**
- **Consistency**
- **Descriptive Models**
  - ***structural***
  - ***causal***

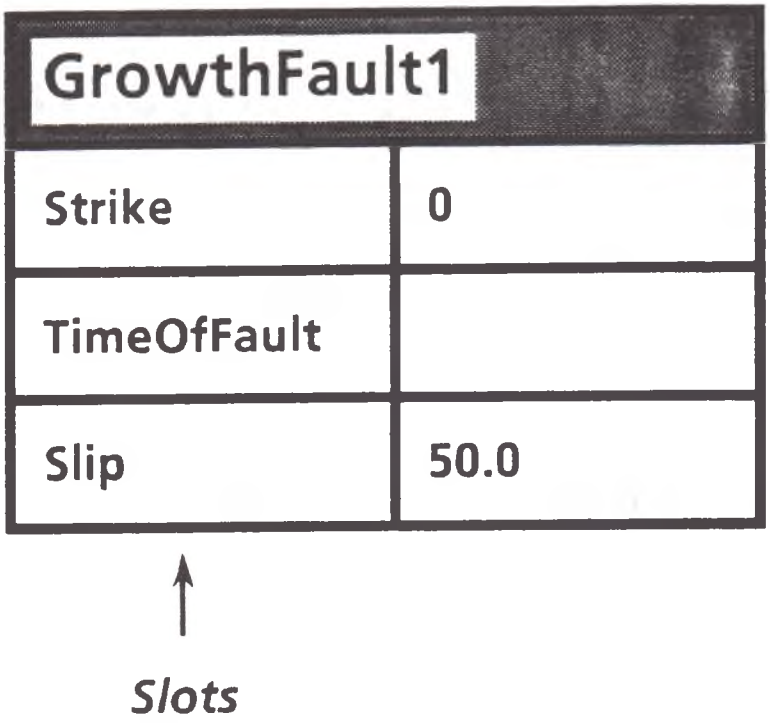
Object-Centered Paradigm: Frames & Objects

- **Primitive Unit**

  Object with Slots & Values

- **Primitive Action**

  Message to Object


GrowthFault1
Strike
0
TimeOfFault
Slip
50.0
Slots

Object-Centered Paradigm: Frames & Objects

Representation of Objects

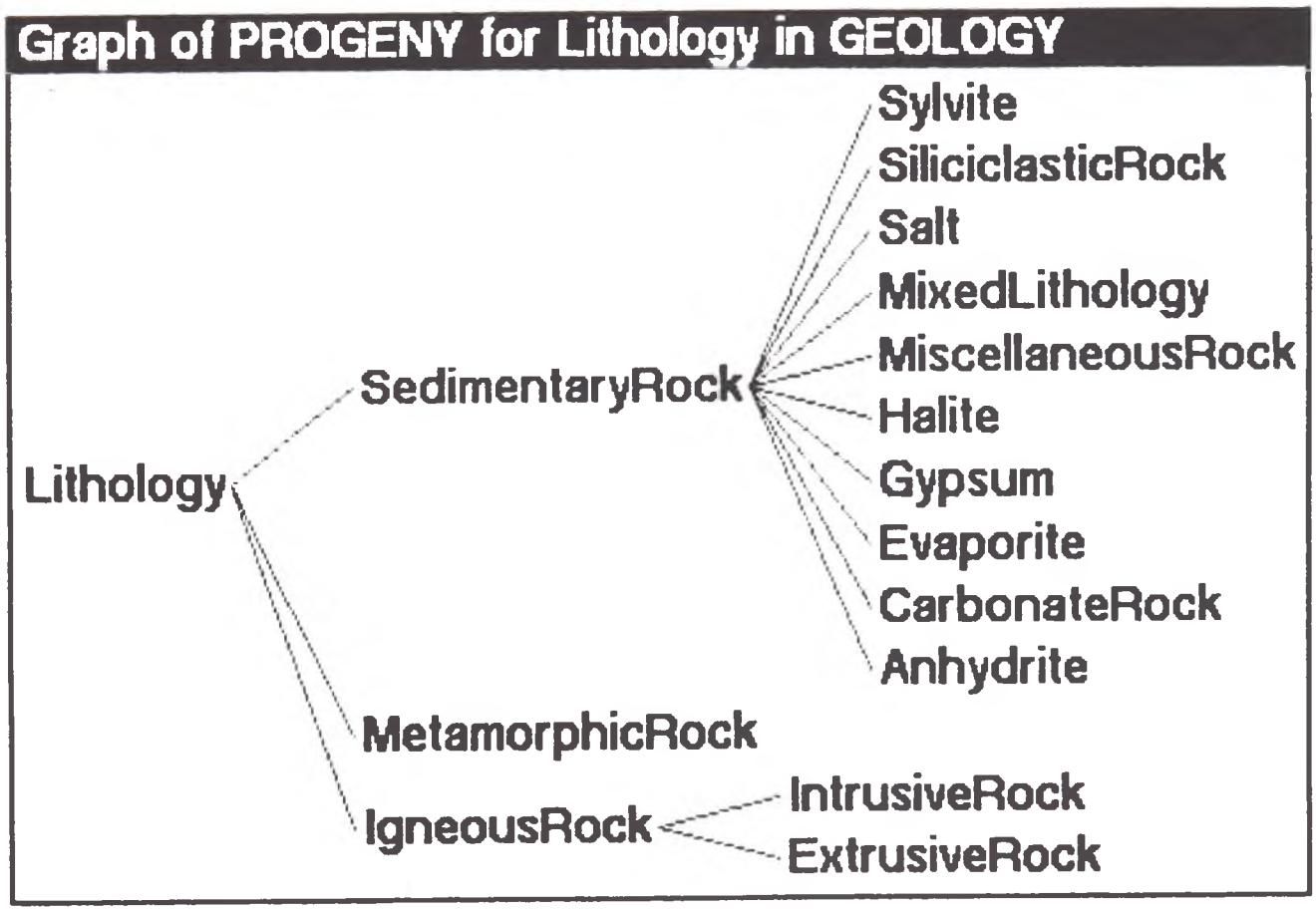
Attribute-Value-Object Triples
Methods
Facets
Attached Procedures

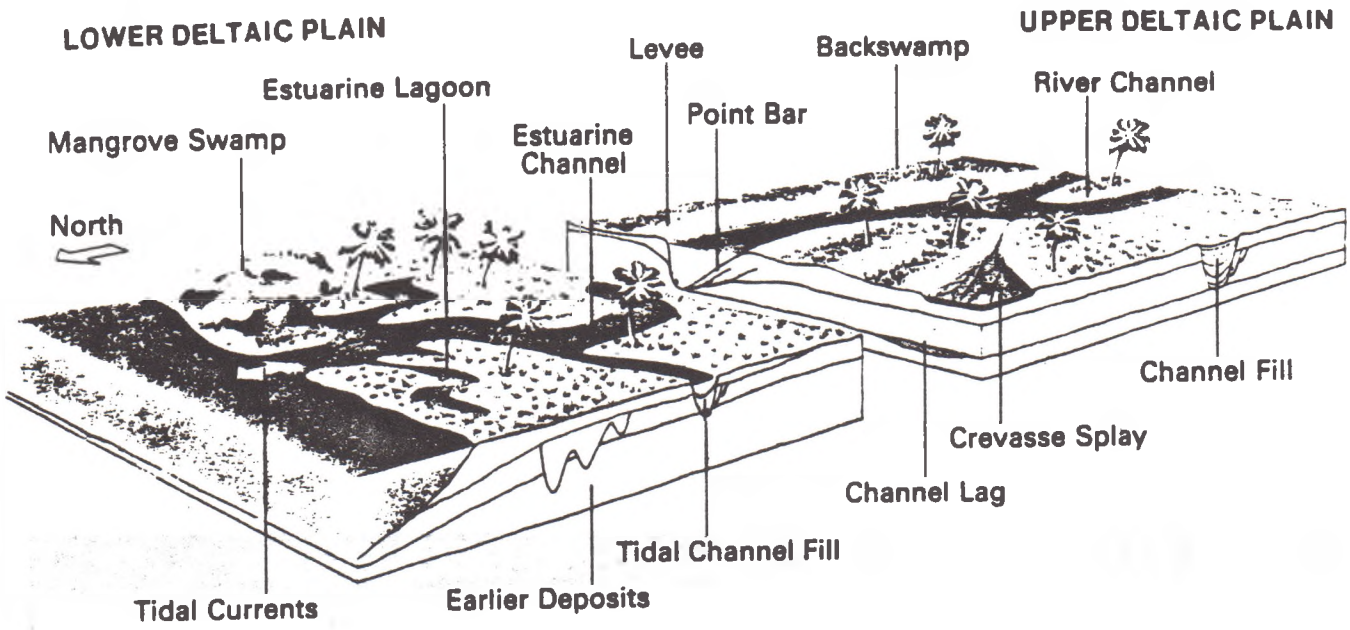
Relations Among Objects

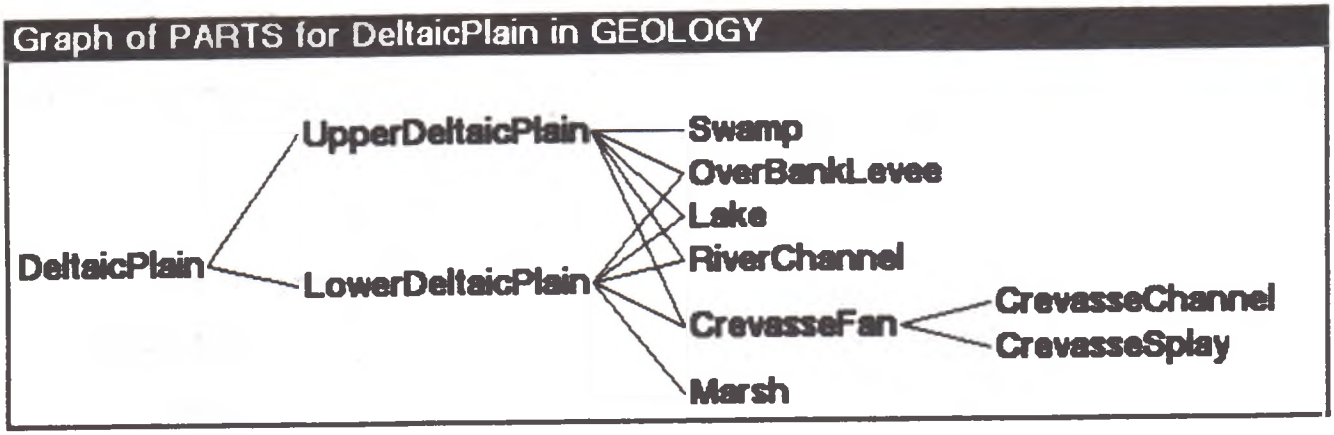
A is-a B

A <relation> B

... with certainty x


Graph of PROGENY for Lithology in GEOLOGY
Lithology
SedimentaryRock
Sylvite
SiliciclasticRock
Salt
MixedLithology
MiscellaneousRock
Halite
Gypsum
Evaporite
CarbonateRock
Anhydrite
MetamorphicRock
IgneousRock
IntrusiveRock
ExtrusiveRock


LOWER DELTAIC PLAIN
UPPER DELTAIC PLAIN
Levee
Backswamp
Estuarine Lagoon
River Channel
Point Bar
Mangrove Swamp
Estuarine Channel
North
Channel Fill
Crevasse Splay
Channel Lag
Tidal Channel Fill
Tidal Currents
Earlier Deposits


Graph of PARTS for DeltaicPlain in GEOLOGY
DeltaicPlain
UpperDeltaicPlain
LowerDeltaicPlain
Swamp
OverBankLevee
Lake
RiverChannel
CrevasseFan
Marsh
CrevasseChannel
CrevasseSplay

OE: PIRIE.NormalFault

Object: NormalFault
Synonyms:
Groups:
Type: CLASS
Edited: 13-Sep-84 13:08:06 **By:** REID
Picture:

HangingWallBlock {DownthrownBlock}:
UpperDistortionRegion:
BrecciaRegion {CrushedZone}:
FaultPlane:
LowerDistortionRegion:
FootWallBlock {UpthrownBlock}:
Strike:
FaultAngle {Hade}:
DirectionToDownthrownBlock:
Slip:
Throw:
TimeOfFaulting:
Draw: DrawFault
Instantiate: InstantiateFault
Detect: (RuleNFR1 RuleNFR3 RuleNFR4 RuleNFR5 RuleNFR7)
Specialize: (RuleNFR6 RuleNFR9 RuleNFR10 RuleNFR11
RuleNFR12)

Frames & Objects: Difficulties

- **No Inference Mechanism**
- **Soft Subclasses**
- **Consistency of Descriptions**
  different static/dynamic views

Inference: Desiderata

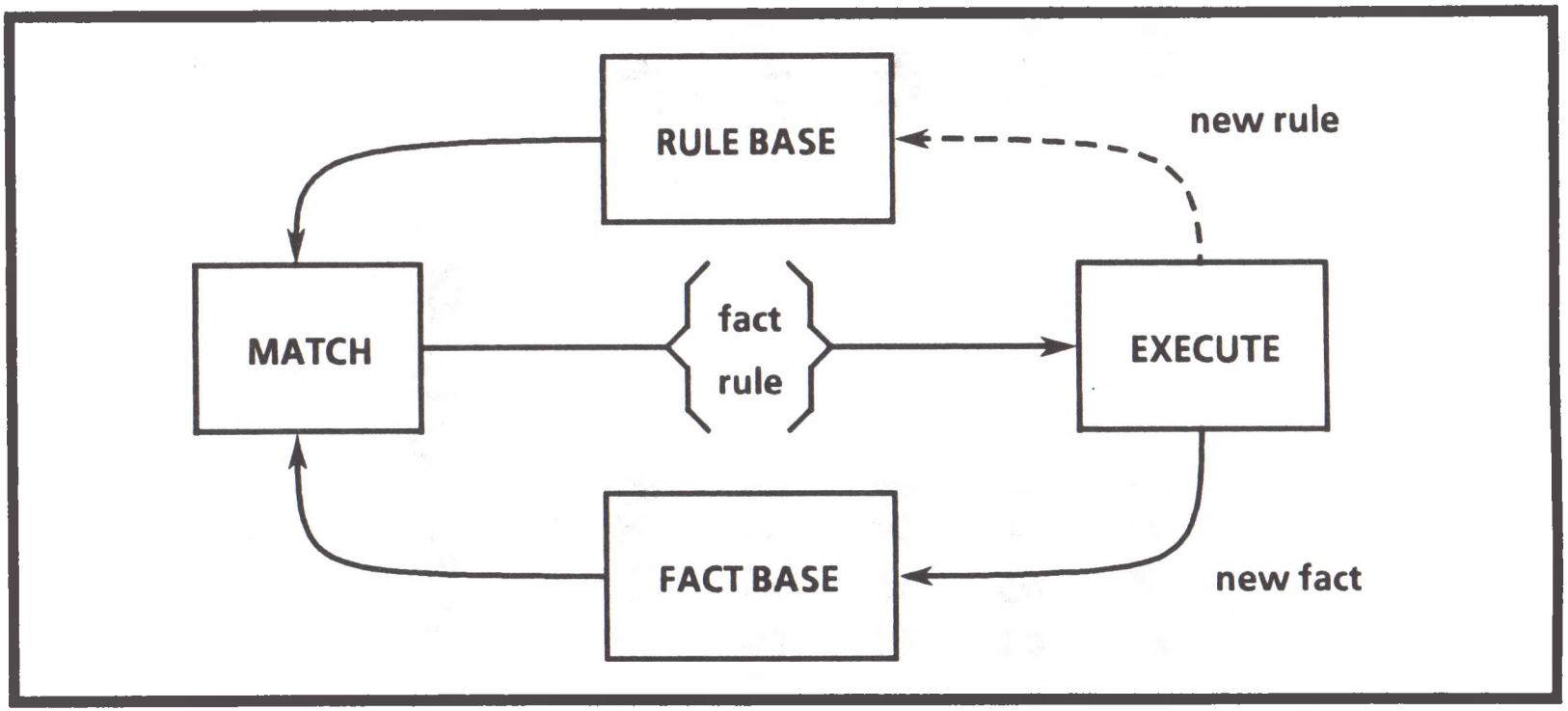
- Appropriate Use of Data
  - Models of Reasoning
  - Accuracy
- Efficiency
- Uncertain Reasoning
- Understandability
- Control of Interaction
  - I/O

Inference as Search: The Generator

Random or Systematic:

- Selection from a List
- Successor Function
- Plausible Move Generator

A Rule Interpreter



Data-Driven Reasoning: Schematic Example

Data... x1, x2, x5

Rules...

- **R1: IF x1 & x2 THEN y1**
- **R2: IF x3 & y1 THEN y2**
- **R3: IF x3 & x4 THEN y3**
- **R4: IF y2 & x5 THEN z1**
- **R5: IF y1 & x5 THEN z2**

Conclusions... (y1), z2

FORWARD CHAINING

if stain is grampos **then** organism is strep.

if stain is gramneg **then** organism is e.coli.

if organism is strep **or** bacteroides **then** penicillin is indicated.

if a drug is indicated **and** don't know whether allergic to the drug **then ask** whether allergic to the drug.

if a drug is indicated **and not** allergic to the drug **then** prescribe the drug.

Goal-Driven Reasoning: Schematic Example

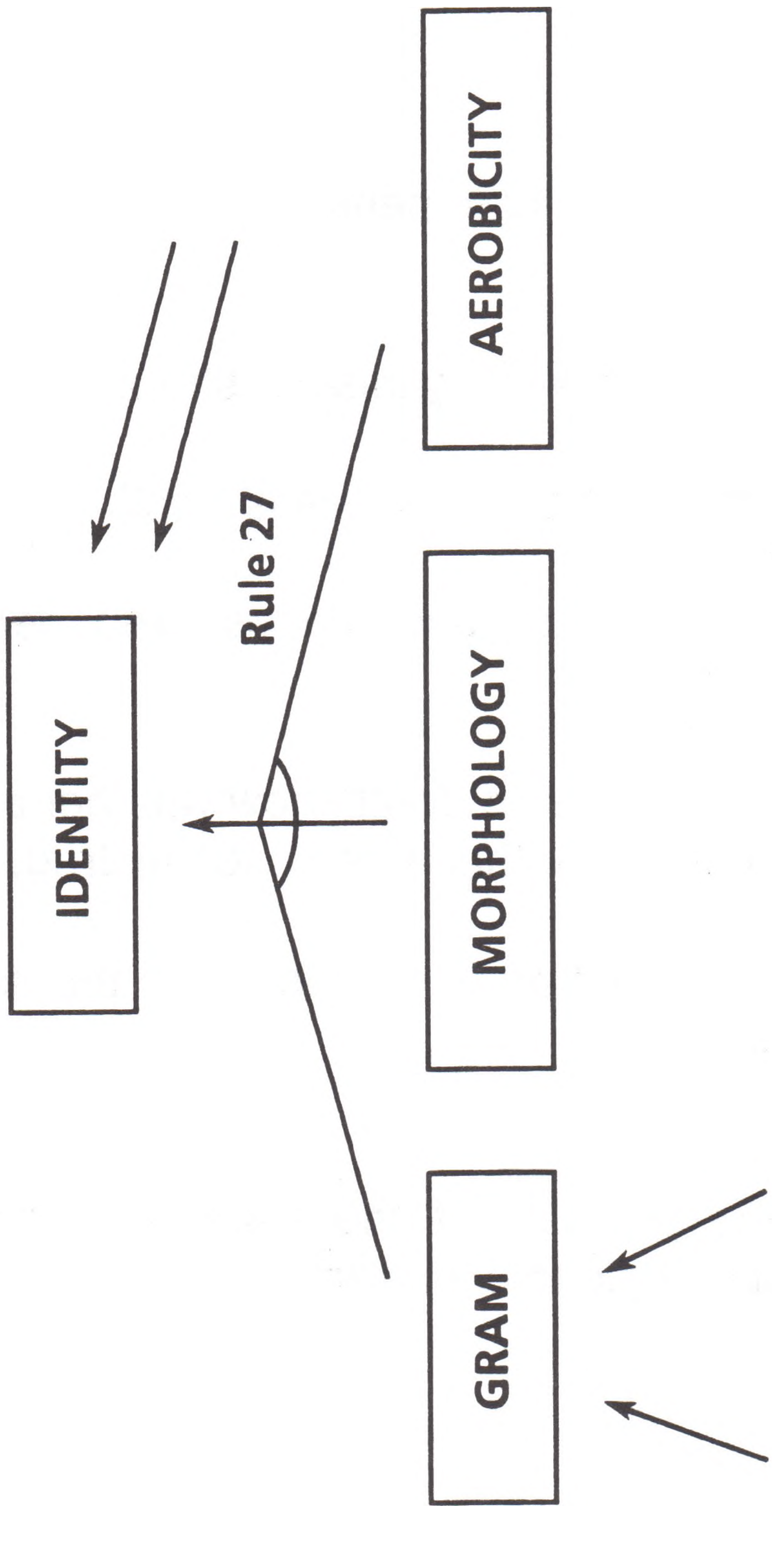
Goal... z2 (*i.e.*, is z2 true?)

Rules...

- **R1: IF x1 & x2 THEN y1**
- **R2: IF x3 & y1 THEN y2**
- **R3: IF x3 & x4 THEN y3**
- **R4: IF y2 & x5 THEN z1**
- **R5: IF y1 & x5 THEN z2**

Questions...

- **Q1: y1 (internal subgoal)**
- **Q2: x1 (?) KNOWN**
- **Q3: x2 (?) KNOWN**
- **Q4: x5 (?) KNOWN**


AEROBICITY
MORPHOLOGY
GRAM
Rule 27
IDENTITY

BACKWARD CHAINING
(SubGoaling)

Q: What about prescribing penicillin?

if stain is grampos **then** organism is strep.

if stain is gramneg **then** organism is e.coli.

if organism is strep **or** bacteroides **then** penicillin is indicated.

if a drug is indicated **and** don't know whether allergic to the drug **then ask** whether allergic to the drug.

if a drug is indicated **and not** allergic to the drug **then** prescribe the drug.

A: Prescribe penicillin if the stain is grampos and patient is not allergic to penicillin.

Event-Driven Reasoning
Schematic Example

Goal... interpret events

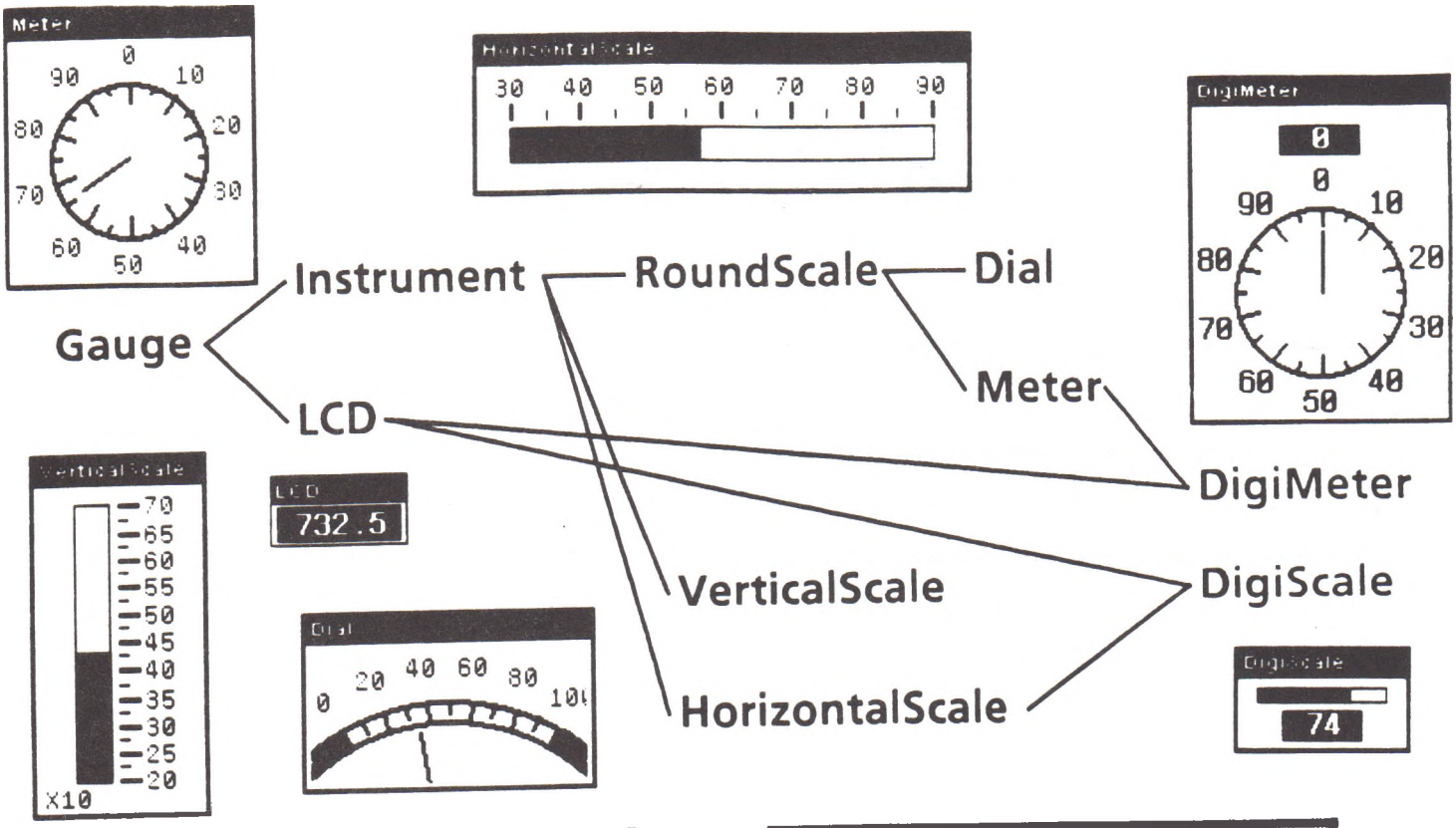
Rules...

- **R1: IF x1 & x2 THEN y1**
- **R2: IF x3 & y1 THEN y2**
- **R3: IF x3 & x4 THEN y3**
- **R4: IF y2 & x5 THEN z1**
- **R5: IF y1 & x5 THEN z2**

Event	Interpretation
x1	--
x2	y1
	➠ expect x5
x4	--
x5 (maybe)	z2

THE DIAGNOSTIC PROCESS

- NOTICE that a problem exists.
- ISOLATE the problem.
- GENERATE alternative hypotheses.
- EXPERIMENT to gather more information.
- RANK hypotheses.
- SELECT the best explanation.
- CONFIRM the choice.
- ACT on the diagnosis.



Meter	
Setting	
Draw	xyzzy

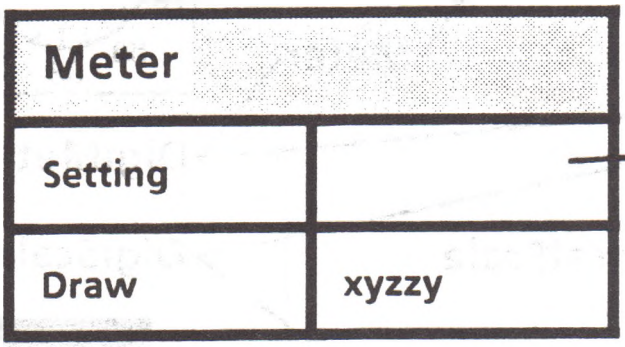
Dial	
Setting	
Draw	ProcF86

Meter11	
Setting	66
Draw	

Dial42	
Setting	34
Draw	

Loops Gauges

EVENT HANDLERS



WhenSet

```
procedure UpdateMeterDisplay (Setting)
   Send(ClearDisplay)
   Send(SetDisplay Setting)
end
```

Inference: Efficiency

- **Policies — usually implicit**

  ***e.g.*, satisficing, plausible set**

- **Heuristics — should be in knowledge base**

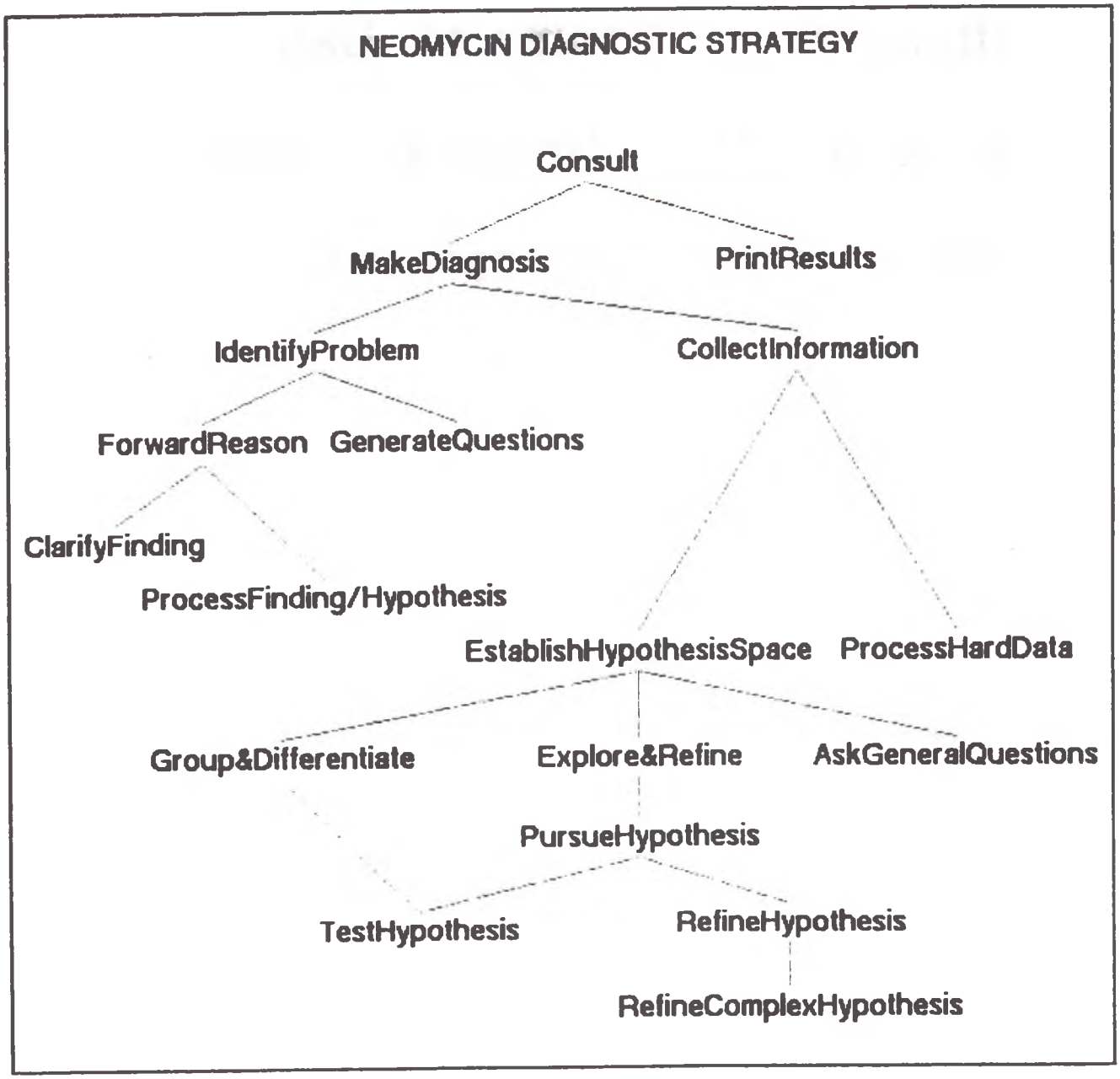
  ***e.g.*, rules, demons**

- **Strategies — trend to declarative form**

Strategic Reasoning: Implementations

General Advice about What to Do

- **Meta-Rules**
  - ***Prune***
  - ***Reorder***
- **Task Definitions**
  - ***Rule Sets***
  - ***Procedure Schemas***
- **Inherited Procedures**
- **Default Reasoning**


NEOMYCIN DIAGNOSTIC STRATEGY
Consult
MakeDiagnosis
PrintResults
IdentifyProblem
CollectInformation
ForwardReason
GenerateQuestions
ClarifyFinding
ProcessFinding/Hypothesis
EstablishHypothesisSpace
ProcessHardData
Group&Differentiate
Explore&Refine
AskGeneralQuestions
PursueHypothesis
TestHypothesis
RefineHypothesis
RefineComplexHypothesis

Uncertainty

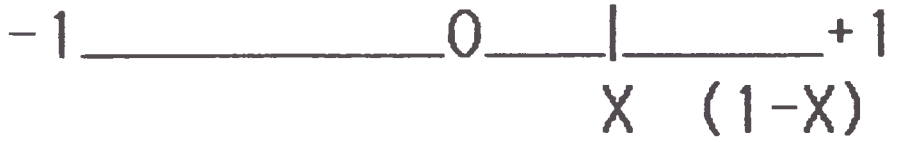
- Evidence Gathering Model
- Uncertain or Incomplete Data
- Probabilistic or Uncertain Rules
- Some Calculi of Uncertainty

Uncertain Reasoning: Some Calculi

1. CASNET/EXPERT
   - thresholds of major & minor findings

2. INTERNIST
   - Σ(pos.findgs) - Σ(neg.findgs)

3. MYCIN/EMYCIN
   - CF = increased importance
     $= X + Y(1-X)$



4. Bayes' Theorem [E= (e1 & e2)]
   [H={h,h2,...,hn}]

$$P(h|E) = \frac{P(e2 \mid h\&e1) * P(h|e1)}{\Sigma\, P(e2|hi\&e1) * P(hi|e1)}$$

Uncertain Reasoning: Some Calculi

5. Liklihood Ratios

6. Fuzzy Logic

7. Dempster-Shafer Belief Rules

Inference: Understandability

- Run time
  - confidence in correctness
  - responsibility
    - personal
    - audit trail

- Development time
  - confidence
  - debugging - localizing errors

- Types of Explanations
  - WHY questions
  - Factual questions / Definitions
  - HOW questions
  - WHY NOT questions

14) Is the patient's illness with ORGANISM-1
a hospital-acquired infection?
** WHY

[i.e., WHY is it important to determine whether or not the infection with ORGANISM-1 was acquired while the patient was hospitalized?]

[2.0] This will aid in determining the identity of ORGANISM-1. It has already been established that
[2.1] the morphology of ORGANISM-1 is rod
[2.2] the gram stain of ORGANISM-1 is gramneg
[2.3] the aerobicity of ORGANISM-1 is facul

Therefore, if
[2.4] the infection with ORGANISM-1 was acquired while the patient was hospitalized
then
there is weakly suggestive evidence (.2) that the identity of ORGANISM-1 is pseudomonas
[RULE050]

** WHAT WAS THE IDENTITY OF ORGAISM-1?
= ORGANISM-1
The identity of ORGANISM-1 is e.coli (.51), salmonella (.45), klebsiella (.4), pseudomonas (.22), enterobacter (.21), proteus (.19), pseudomonas-aeruginosa (.08), serratia (.02)

** HOW DID YOU DECIDE THAT ORGANISM-1 WAS AN E.COLI?

I used rule084 to conclude that the identity of ORGANISM-1 is e.coli. This gave a cumulative c.f. of (.51).
I used rule003 to conclude that the identity of ORGANISM-1 is e.coli. This gave a cumulative c.f. of (.43).

** WHY DIDN'T YOU THINK THAT ORGANISM-1 WAS NEISSERIA?

The following rules could have been used to determine that the identity of ORGANISM-1 is neisseria: 9
However, none of these succeeded in the context of ORGANISM-1. If you would like an explanation for why any of these rules failed, please enter their numbers:
** 9

Clause 2 of rule009 ["the morphology of the organism is coccus"] was already known to be false for ORGANISM-1, so the rule was never tried.

Inference: Control of Interaction

- Hired Gun Model
  e.g., DENDRAL, PUFF
- Consultant Model
  e.g., MYCIN, Prospector
- Assistant Model
  e.g., Dipmeter Advisor, Oncocin
- Tutor Model
  e.g., Guidon

Inference: Difficult Issues

- Common Sense Reasoning
- Accuracy vs Precision Tradeoff
- Knowing What You Know
- Brittleness
- Reasoning with Very Large, Unstructured Data
- Non-monotonic Reasoning

Hardware for Expert Systems

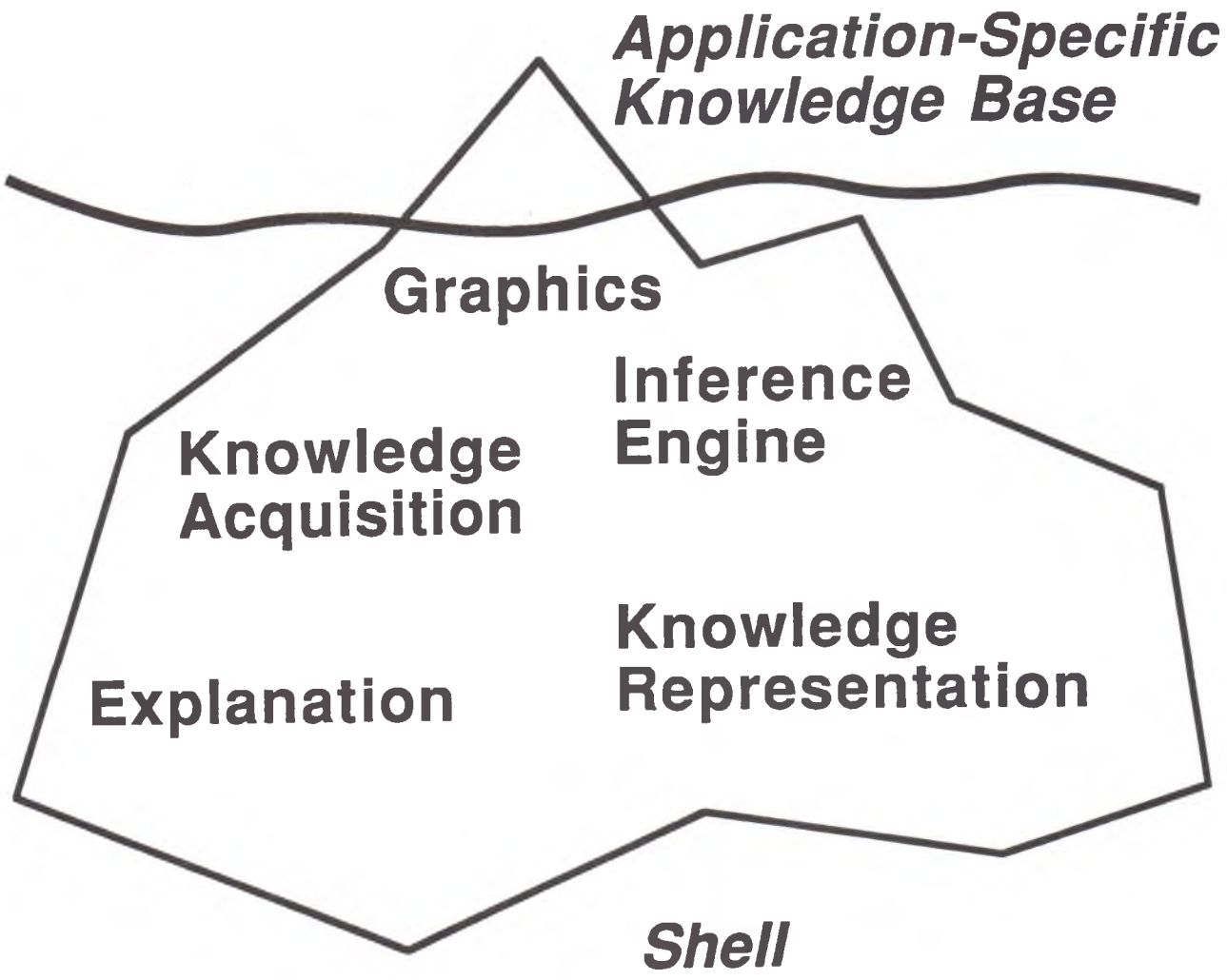
Possible Choices

- PC
- Lisp Machine
- Workstation
- Mainframe

Evaluation Criteria

- Cost, Availability, Support
- Performance
- Graphics & Interaction
- Development Environment
  including language support
- Standards
  including networking

Development Platform
vs
Delivery Platform



The Importance of Powerful Tools

Evaluating Shells

—What To Look For—

- **Representation Choices**
  objects, rules, tasks
- **Inference Mechanisms**
- **Built-In Problem-Solving Methods**
  heuristic classification
- **Specific Features**
  multiple hypothesis support
  dependencies
  uncertainty
- **Extensibility**
- **Ability to Scale Up**

Evaluating Shells

—What To Look For—

- **Editing/Debugging Facilities**
  browsing
  managing complexity
- **Graphics & User Interaction**
- **Use in Fielded Systems**
- **Intended Users**
  novice, expert, programmer

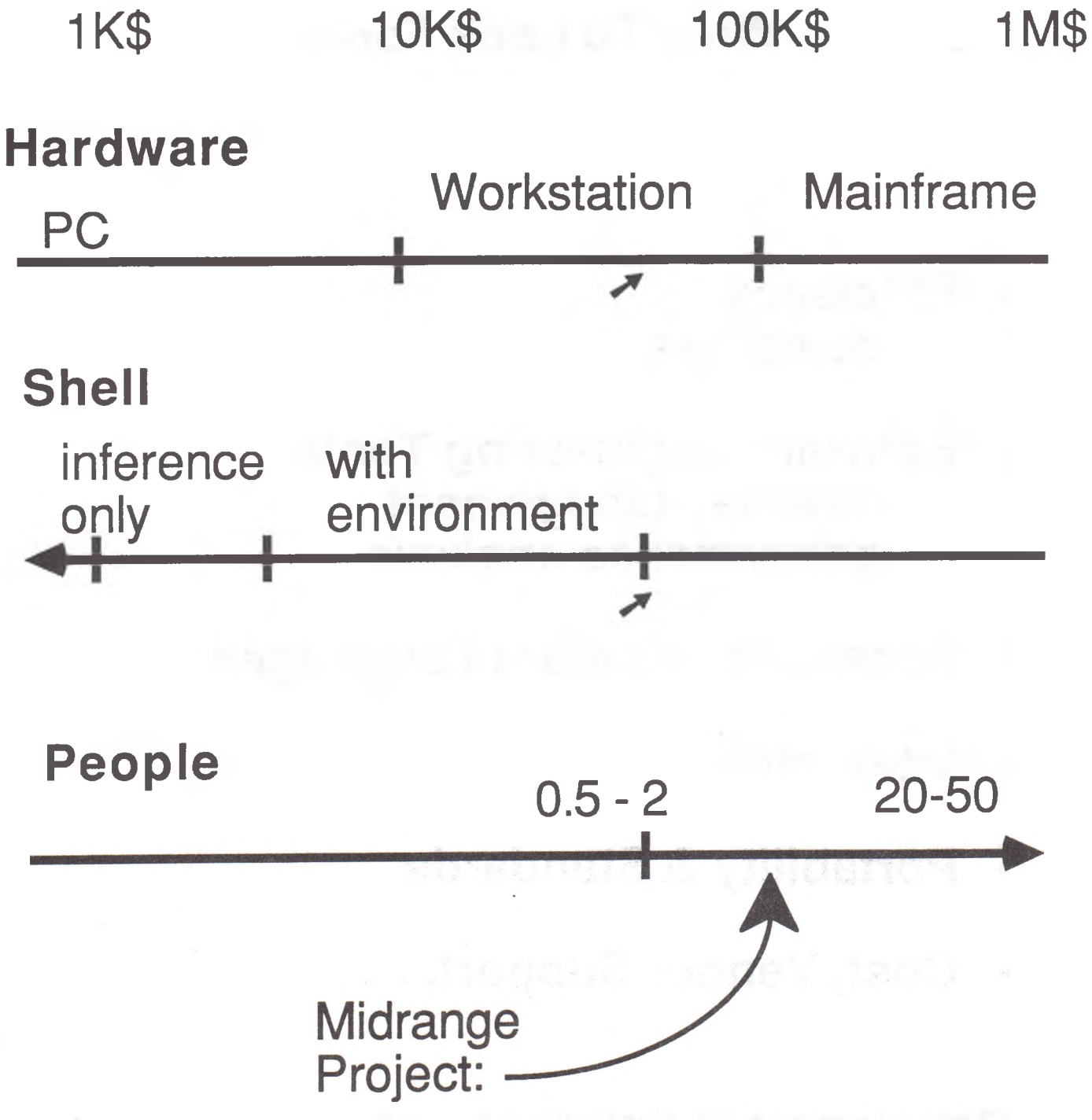
Evaluating Shells

—What To Look For—

- **Efficiency**
  compilers
- **Software Engineering Tools**
  release management
  performance analysis
- **Access to Standard Languages**
- **Integration**
- **Portability & Standards**
- **Cost, Vendor Support, ...**

Development Environment
vs
Delivery Environment

Direct Development Costs



3 workstations & shells, 5-10 person-years

Pragmatics

- Selecting an application
- Steps in constructing an expert system
- The development team
- Technology transfer: Steps in fielding an expert system
- Pitfalls
- Models of Successful Efforts

What Makes A Good Application?

Problem Definition

Importance:

the task has a high payoff

the benefits of using a system justify the costs of developing and using it

What Makes A Good Application?

Expertise

there are recognized experts

the experts are provably better than amateurs

there is general agreement about the knowledge

What Makes A Good Application?

Managerial Components

the commitment of an expert can be obtained

there is a supportive manager with clout

adequate computation resources, machines and staff, exist

a product development organization exists

What Makes A Good Application?

Target Community

the target users have been defined

the target users want a system and are ready to use it

the context of use has been defined

users can exercise common sense

the users and the experts share a conceptual framework

What Makes A Good Application?

Problem Definition

Scope:

the skill can be routinely taught to neophytes

the task takes an expert a few minutes to a few hours

the knowledge is bounded

the knowledge is primarily symbolic

algorithmic solutions are not practical

incremental progress is possible

data and test cases are available

Steps in Constructing an Expert System

- Identification

  Problem
  Target Community
  Resources

- Conceptualization & Formalization

  Concepts
  Methods
  Representation

- Implementation by Exploratory Programming

  Incremental Refinement
  Experimentation with Real Data and Real Users
  Revision, Extension ...Redesign

Steps in Constructing an Expert System

Time

- **Identification**
  days/weeks
- **Conceptualization & Formalization**
  weeks
- **Implementation**
  months

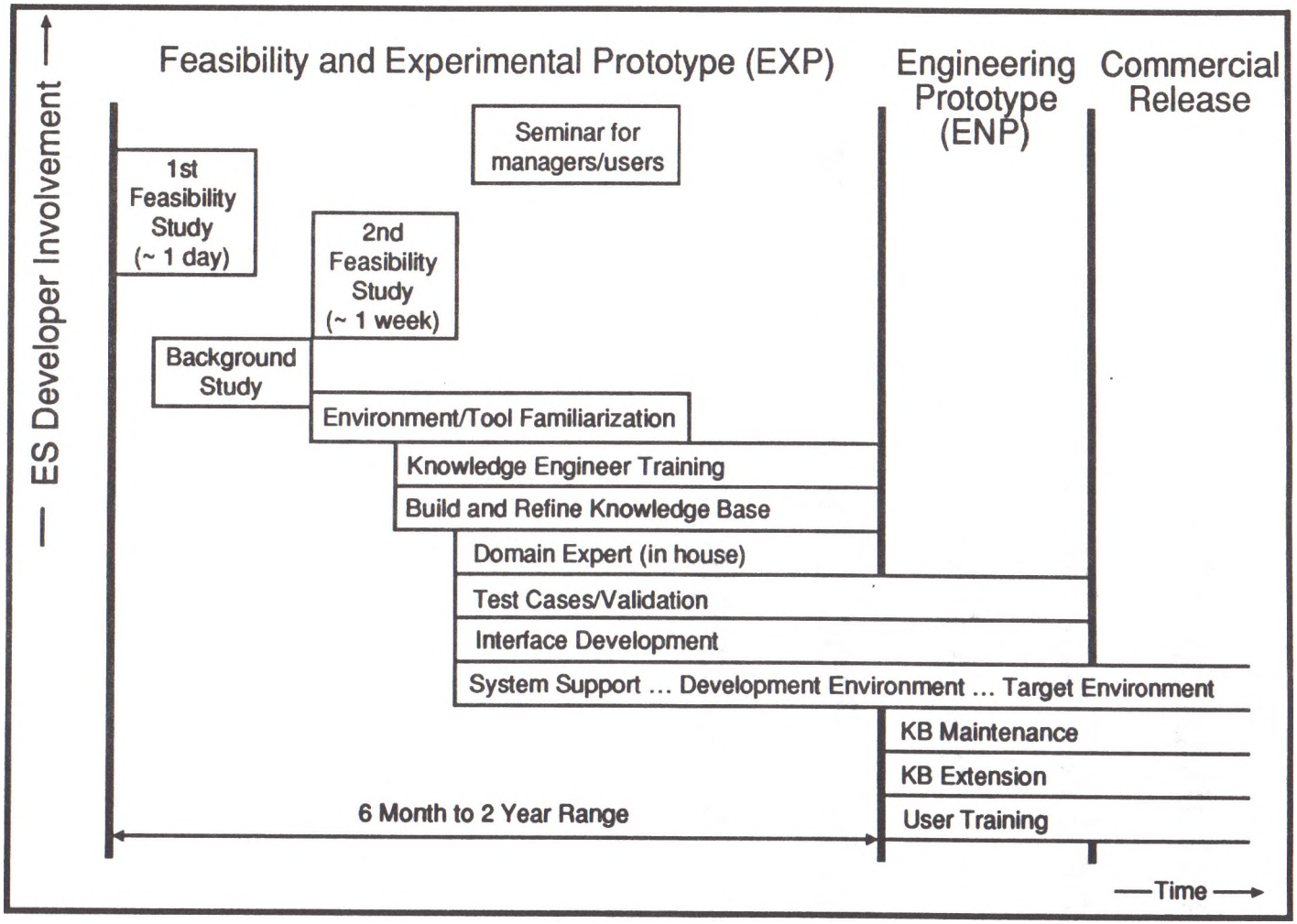
Total

6 months to 2 years for systems that "interest" the target community

Assumptions

Developers, machines, expertise exist in-house

Tools that fit the problem exist in-house along with knowledge of how to use them



Expert System Timeline

The Development Team

- Domain Expertise
- Prototypical User
- Shell & Tool Design
- Knowledge Engineering
- System/Programming Support
- Software Engineering

Development Team Training

- a few days
  criteria, intuition
- a few weeks
  how to..., hands-on experience
- a few years—degree program
  conceptual understanding

Technology Transfer: Steps in Fielding an Expert System

- Testing and Validation

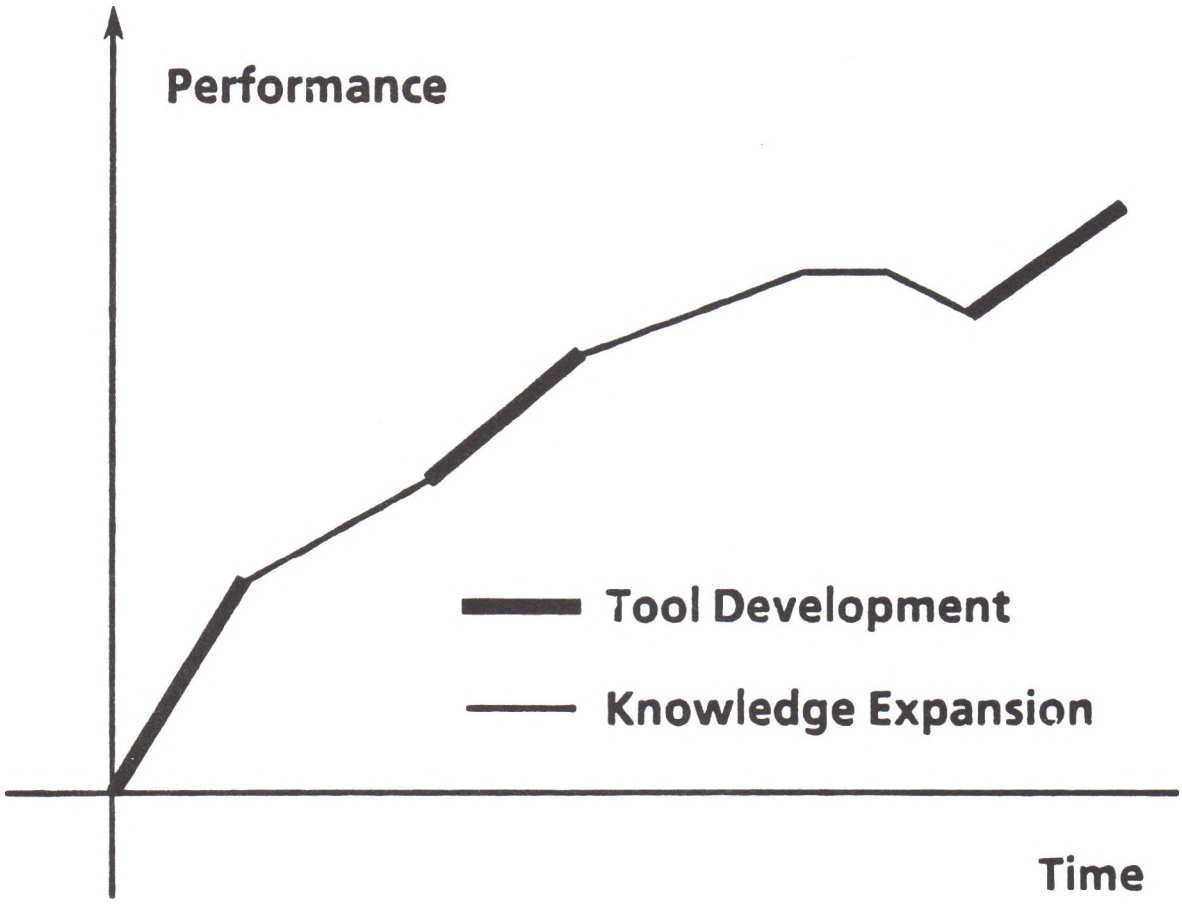
  Performance
  Scope, Accuracy, Efficiency
  Human Engineering
  Efficiency, Robustness

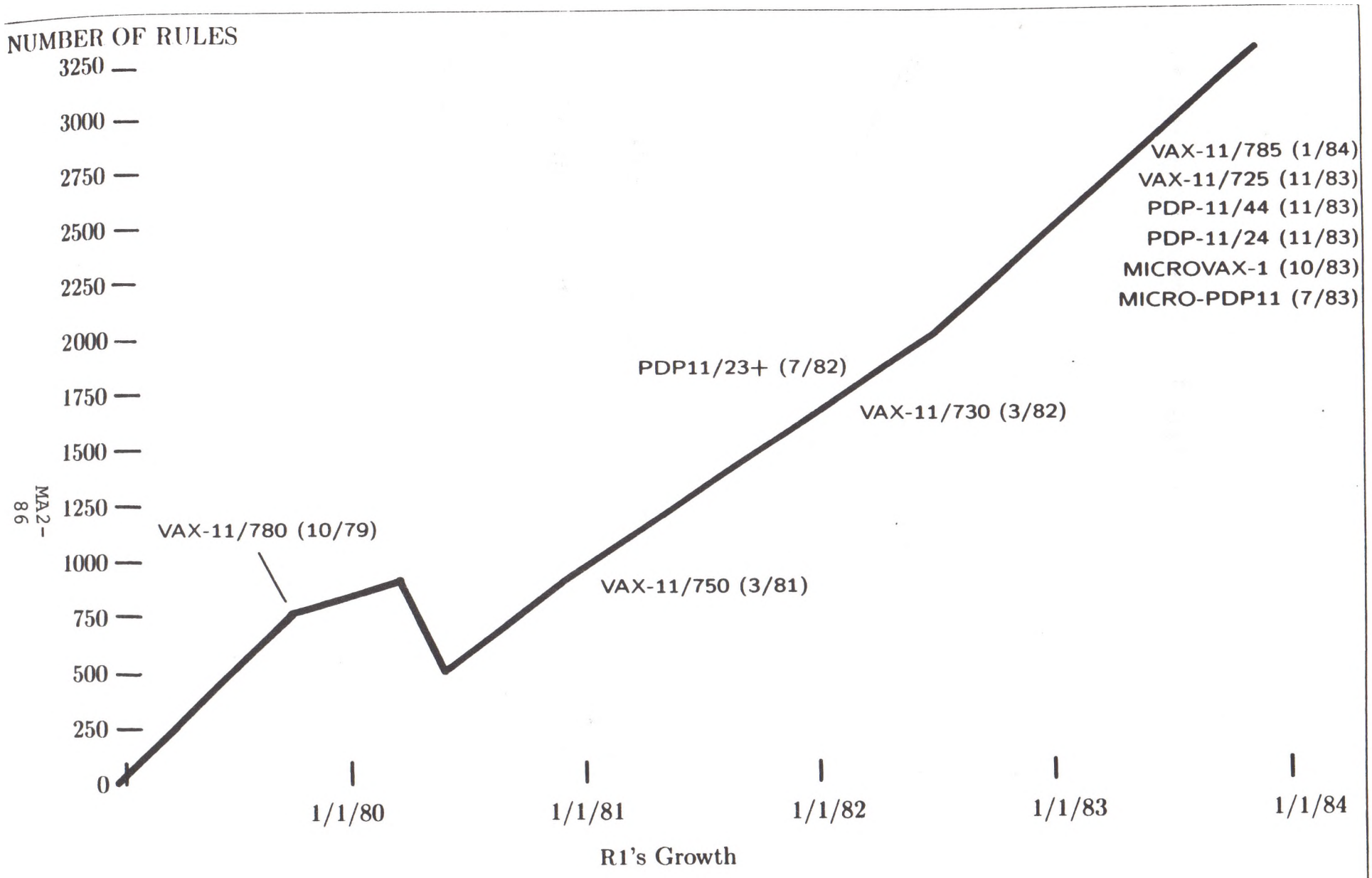
- Software Engineering for Target Environment

  Hardware, Network
  Software
  Interface
  Integration

- Documentation
- Training
- Marketing & Sales
- Maintenance

Incremental Development




NUMBER OF RULES
3250
3000
2750
2500
2250
2000
1750
1500
1250
1000
750
500
250
0
MA2-86
VAX-11/780 (10/79)
VAX-11/750 (3/81)
PDP11/23+ (7/82)
VAX-11/730 (3/82)
VAX-11/785 (1/84)
VAX-11/725 (11/83)
PDP-11/44 (11/83)
PDP-11/24 (11/83)
MICROVAX-1 (10/83)
MICRO-PDP11 (7/83)
1/1/80
1/1/81
1/1/82
1/1/83
1/1/84

R1's Growth

Pragmatics: Pitfalls

- Excessive Aspirations
- Inadequate Resources
- Inadequate Management Support
- Poor Problem Selection
- Forgetting the User
- Premature Optimization
- Technology Transfer & Sociology

Pragmatics: Ways To Be Successful

- **Digital**
  collaborate with universities
- **IBM**
  redirect computer science talent
- **Schlumberger**
  build a research group
- **General Motors**
  form strategic partnerships
- **Travelers Insurance**
  contract with AI company
- **Kawasaki Steel**
  buy shells & train programmers

Use AI to get a single job done
vs
Broad committment to computer science

Validation:

- What is the question?
- What counts as an answer?
- How do you get the answer?

Validation:
Questions to Ask

- How good is this program?
  - a. conceptual framework
  - b. particular knowledge base

- Is this progam at least "as good as" specialists [novices, users, experts] over problems in domain D, for users of class U?

- Bottom line = productivity
  i.e., cost/benefit tradeoff

Validation:
Dimensions of Answers

- **Computational**
  - *time & space*
  - *robustness*
  - *consistency*
  - *completeness*
  - *portability*
  - *extensibility*

- **Psychological**
  - *ease of use*
  - *ease of learning*
  - *understandability*
  - *elegance—look & feel*

- **Performance**
  - *accuracy*
  - *precision*
  - *reliability*
  - *scope: breadth & depth*

Validation: Methods Used for Some Expert Systems

- **Commercial Use**

  XCON
  Dipmeter Advisor

- **Comparison with Test Data**

  DENDRAL
  Paradise
  ABLE/SLAC
  AI/RHEUM

- **Comparison with Specialists**

  INTERNIST
  MYCIN

Validation:
Summary of Considerations

- Consider validation in initial problem definition
- Define the question
- Define the gold standard
- Measure the appropriate characteristics
- Use good statistical sense in design & execution of study

State of the Art

- Level of Effort
- Problem Size
- Problem Scope
- Shells
- Limitations/Current Research
- Some Scenarios

COSTS OF BUILDING KNOWLEDGE-BASED SYSTEMS

ASSESSMENT
a few days or weeks

PROTOTYPING
1-2 man-years knowledge engineer
0.5 man-years domain specialist

DEVELOPMENT
2-5 man-years knowledge engineer
half-time from domain specialist

FIELDING
software engineering

Sizes of Some Solution Spaces

MYCIN $\binom{120}{6} \sim 10^9$

$\longrightarrow 6 \times 10^6$

INTERNIST $\binom{571}{3} \sim 31 \times 10^6$

DIPMETER ADVISOR $\sim 500^{65}$

XCON $\binom{94}{20}$

RULE-BASED & OBJECT-CENTERED EXPERT SYSTEMS

(# RULES / # OBJECT NAMES)

MYCIN	62.3	=(1059 / 17)
XCON	61.0	=(5739 / 94)
XSEL	27.1	=(2148 / 79)
XFL	21.8	=(1618 / 74)
INTERNIST	5.2	=(2600 / 500)
DIPMETER	1.4	=(90 / 65)
TEKNOWL.	0.4	=(1242 / 3317)

VOCABULARY SIZE

		(#obj + #attrib + #vals)
MYCIN	715+	=(17 + 257 + 441+)
INTERNIST	4674	=(571 + 4100 + 3)
XCON	934+	=(94 + 840 + ??)
XSEL	408+	=(79 + 329 + ??)
XFL	326+	=(74 + 252 + ??)

NOTES:

1. Attributes may take continuous numerical values.
2. Objects may be instantiated many times.
3. Rules may apply to many different contexts.

RANGES OF KB SIZE

	straight-forward
Vocabulary	
# objects [concepts]	10 - 100
# attributes/object	10 - 1000
# legal values/attribute	3 - 100
Inferential Relations*	
Depth of Inference Chains	4 - 10
Degrees of Uncertainty	continuous
Data / Case Information	
Noisy Data	some
Missing Data	some
Inconsistent Data	some

*The number of inferential links (rules) is dependent on the number of things being linked and the complexity of the inferences in the domain.

Problem Scope

- Importance
  - small ≠ unimportant
  - cost/benefit analysis
  - number of experts
  - training time for new persons
  - lost time from not getting it right the first time
- Feasibility
  - telephone test
  - training manual
- Size
  - number of input descriptors
  - size of solution space
  - size of total vocabulary
  - average time for experts
  - size of manuals & handbooks

Problem Solving Shells

- Representation Choices
- Inference Mechanisms
- Run-Time Environment
  - Explanation
  - Presentation
  - Options - data entry
    - task
  - Integration
- Development Environment
  - Case management
  - Editor
  - Explanation
  - Debugging aids
  - Compiler
  - Software Engineering Tools

Problem Types: Questions

- **Diagnosis/ Troubleshooting**
  What is the cause of the problem?

- **Data Interpretation**
  What do these data mean?

- **Monitoring/ Real-Time Control**
  What's going on?

Problem Types: Questions

- **Scheduling/ Planning/ Therapy**
  What is the sequence of steps to get to the goal?

- **Configuration/ Layout**
  What is the 2-d plan that satisfies the constraints?

- **Design/ Spatial Arrangement**
  What 3-d configuration fits the specifications?

- **Constraint Satisfaction**
  What description satisfies all of the constraints?

Problem Types: Methods

- Search = General Model
- Classification / Evidence Gathering
- Skeletal Planning / Stepwise Refinement
- Stepwise Construction / Plan-Generate-Test
- Means-Ends Analysis / Subgoaling
- Constraint Propagation

WHAT IS THE STATE OF THE ART?

Expert-level performance on narrow problems

Sufficient knowledge to solve important problems

Understandable, but limited explanation of line of reasoning

Natural human interface, both graphical and text, but with stylized language and limited vocabulary

Flexible knowledge bases

Requirement for an experienced "knowledge engineer"

Limited to **one** expert as the "knowledge czar"

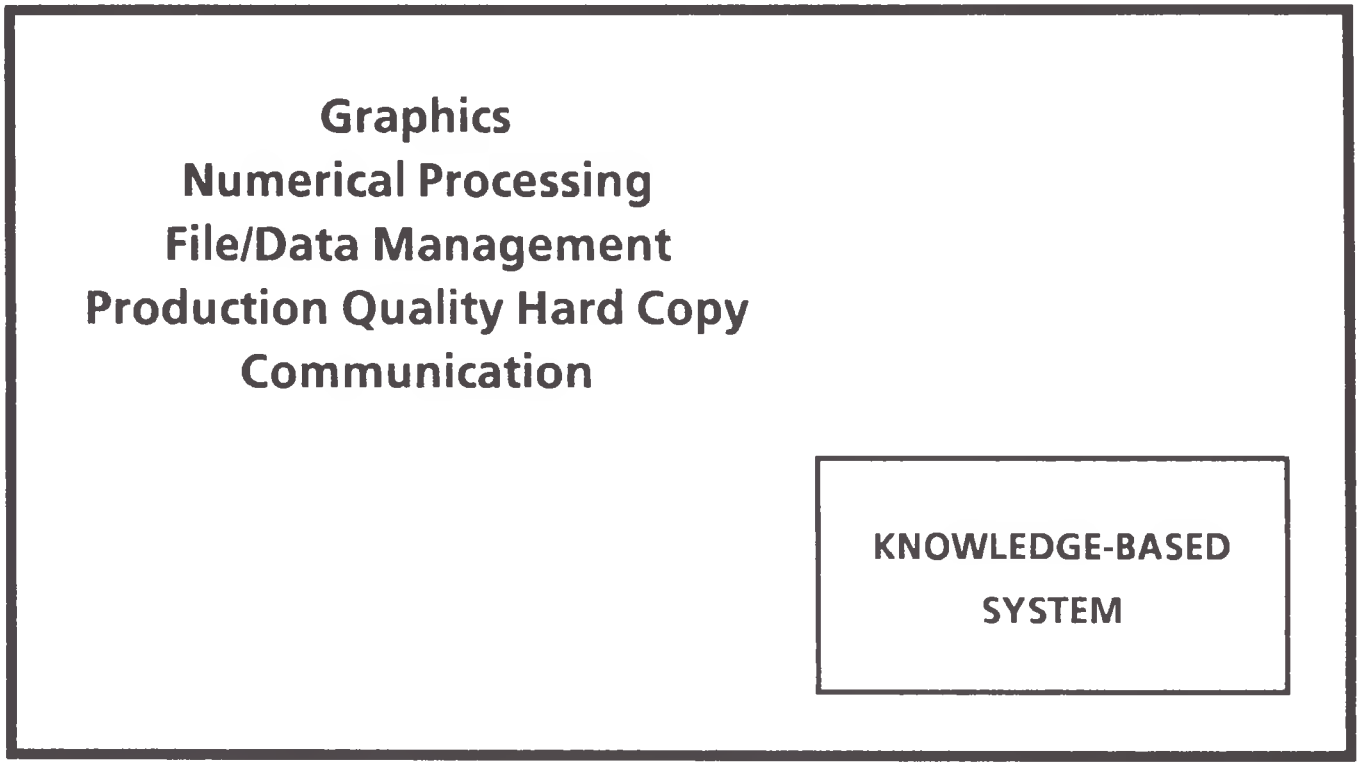
THE CURRENT STATE OF SOME HARD PROBLEMS

	PRACTICE	THEORY
Inexact Reasoning	CF Model	Almost OK
Knowledge Engineering	An Art	Unexplored
Learning By Induction	Hand-Crafted	Over-Developed
Default Reasoning	Inheritance	Emerging
Common-Sense Knowledge	Add Items To KB	Puzzling
Strategies	Meta-Level Knowledge	Not Well Explored

Some Scenarios

- Autonomous Agent
- Consultant
- Assistant
- Critic
- Tutor

Embedding a Knowledge-Based System: An Intelligent Assistant



A user gets a number of advantages from using the system–one of which is symbolic inference.

In watching the system operate, an observer might never realize that any intelligence is involved.

References

GENERAL

Buchanan, B.G.
Expert Systems: Working Systems and the Research Literature.
Expert Systems 3(1):32-51, January, 1986.
Also KSL Report No. 85-37.

Hayes-Roth, Frederick, Donald A. Waterman, and Douglas B. Lenat[eds.].
Building Expert Systems.
Addison-Wesley, Reading, MA, 1983.

Jackson, Philip C., Jr.
Introduction to Expert Systems.
Addison-Wesley, Reading, MA, 1985.

Mishkoff, H.C.
Understanding Artificial Intelligence.
Texas Instruments Information Publishing Center, P.O. Box 225474, MS-8218, Dallas, Texas 75265, 1985.

Waterman, D.
A Guide to Expert Systems.
Addison-Wesley, Reading, MA, 1985.

TECHNICAL

J. Bachant and J. McDermott.
R1 Revisited: Four Years in the Trenches.
AI Magazine 5(3):21-32, Fall, 1984.

B. G. Buchanan and E.H. Shortliffe.
Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.
Addison-Wesley, Reading, MA, 1984.

R. Fikes and T. Kehler.
The Role of Frame-Based Representation in Reasoning.
CACM 28(9):904-920, September, 1985.

Harmon, P. .
Inventory of Existing Expert Systems Applications.
Expert Systems Strategies 2(2), February, 1986.

Nii, H. P.
Blackboard Systems Part One: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures.
AI Magazine 7(2):38-53, Summer, 1986.

Nii, H. P.
Blackboard Systems Part Two: Blackboard Application Systems, Blackboard Systems from a Knowledge Engineering Perspective.
AI Magazine 7(3):82-106, Conference, 1986.

Richer, M.H.
An evaluation of expert system development tools.
Expert Systems 3(3):166-183, July, 1986.

R. G. Smith.
On the Development of Commercial Expert Systems.
AI Magazine 5(3):61-73, Fall, 1984.

M. J. Stefik and D. G. Bobrow.
Object-Oriented Programming: Themes and Variations.
AI Magazine 6(4):40-62, 1986.

JOURNALS & NEWSLETTERS

Applied Artificial Intelligence Reporter, published by University of Miami. Subscription inquiries to ICS Research Institute, P.O. Box 1308-EP, Fort Lee, New Jersey 07024.

AI Magazine, published by the American Association for Artificial Intelligence. Subscription inquiries to AAAI, 445 Burgess Drive, Menlo Park, California 94025.

Expert Systems, published by Learned Information. Subscription inquiries to Learned Information, 143 Old Marlton Pike, Medford, New Jersey 08055.

Expert Systems Strategies, published by Cutter Information Corp. Circulation Office: Expert Systems Strategies, 1100 Massachusetts Avenue, Arlington, Massachusetts 02174.

IEEE Expert, published by the IEEE Computer Society. Subscription inquiries to IEEE Headquarters, 345 East 47th Street, New York, New York 10017.

AAAI-87 TUTORIAL EVALUATION FORM

********PLEASE TEAR THIS PAGE OUT OF BOOKLET AND LEAVE AT THE DOOR********
OR FOLD, STAMP, AND MAIL TO ADDRESS ON THE OPPOSITE SIDE.

TUTORIAL #: NAME OF TUTORIAL:.....
Speaker 1:.....
Speaker 2:.....

****

CONTENT:
Were any topics covered that might have been omitted?_____

Were any topics omitted that you wanted to have covered?_____

Was the conference brochure description accurate?_____

Was the technical level of the tutorial appropriate?_____
Too general?____ Too detailed?____ Too difficult?____Too simple?_____

Were the speakers well prepared? (1)_____(2)_____
Were the speakers understandable? (1)_____(2)_____

OVERALL RATINGS:
Content: Excellent____ Good ____ Fair____ Poor____
Speaker (1) Excellent____ Good ____ Fair____ Poor____
(2) Excellent____ Good ____ Fair____ Poor____

Would you recommend this tutorial to your colleagues? _____
Why or why not?_____

Was the advance reading material you received (if any) useful?_____
Additional/other readings you would recommend?_____

COMMENTS REGARDING THE TUTORIAL PROGRAM:
Any other tutorials you would especially like to attend next year?_____

Any other speakers you would especially like to hear next year?_____

Specific changes in the format of the tutorials you would like, such as lengthening/shortening, time for questions, etc.?_____

Other advice to give the tutorial chair next year?_____

ABOUT YOU:
Affiliation: Hardware Manufacturer___ Software Manufacturer/Publisher___
Computer/DP services/timesharing___ Research/education/consulting___
Government:Federal/state/local___ Banking/finance/insurance___
Transportation/communications___ Other(specify):_____

Role: Student__ Graduate student__ Staff Scientist__ Research Scientist__
Consultant__ Engineer__ Programmer/Analyst__ Middle management__
Project leader__ Systems analyst__ University/College educator__
Administrator__ Other(specify):_____

PLEASE WRITE OTHER COMMENTS IN THE MARGINS: