

Performance Modeling Decision Trees

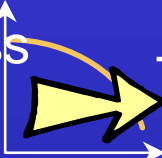




2002 Southeastern Conference
Pavement Management & Design
Nashville, TN
June 23 - 26, 2002

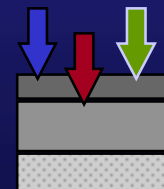
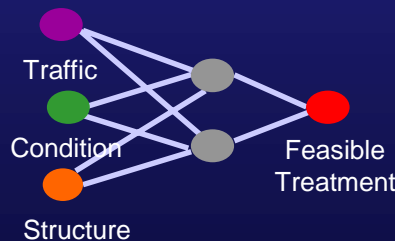
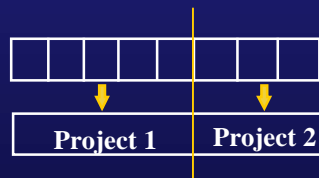
Presented by:

Adel Hedfi
Axiom Decision Systems, Inc.



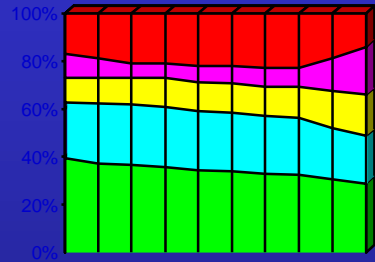
Pavement Management

Synthesize & Harness Data with State Expertise  Transform Data into Knowledge   Budget  Goals  Make Better Decisions about Pavement Maintenance

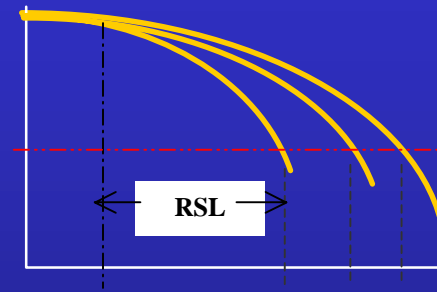


Performance Modeling and Decision Trees

Utility of Models



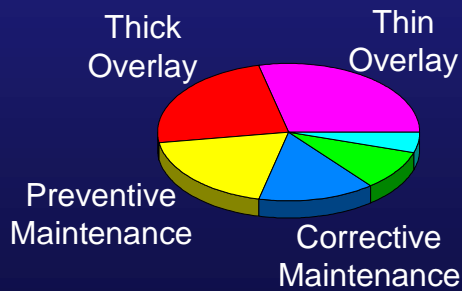
Forecasting Condition



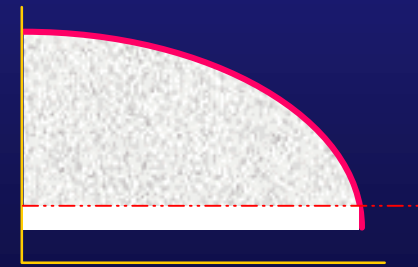
Remaining Service Life



Performance Models



M & R Strategy



Benefit Calculation

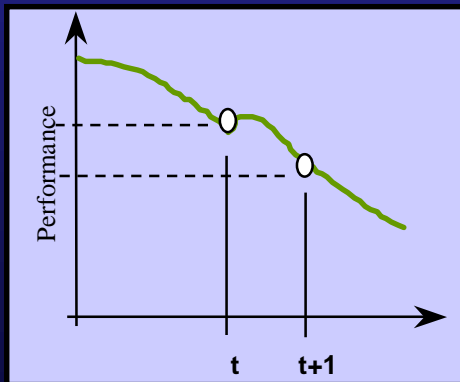
Examining the Modeling Alternatives

Pavement Performance

- Structure
- Traffic
- Environment
- Soil Type
- Const. Quality
- Moisture
- others

- Array of Parameters
- Complex
- Uncertainty

Deterministic



initial

Initial Condition

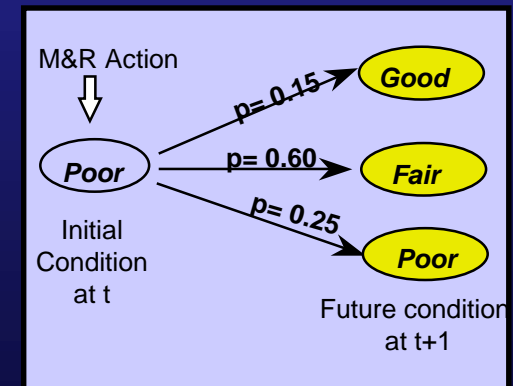
initial

future

Predicted Condition

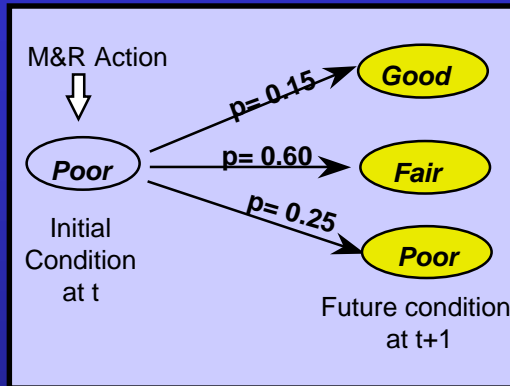
future

Probabilistic



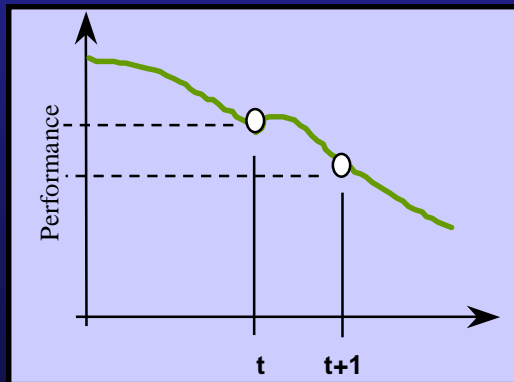
Modeling Choice: Probabilistic or Deterministic?

Probabilistic (TPMs)



- Simulate the uncertainty in pavement behavior
- Statewide Planning and allocation of funds
- Forecasting at the Network Level
- Optimization using Linear Programming

Deterministic (Curves)



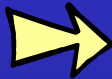
- Quantify benefits
- Calculate remaining service life
- Forecasting at project level
- Visual representation



Both Probabilistic and Deterministic Models are Needed

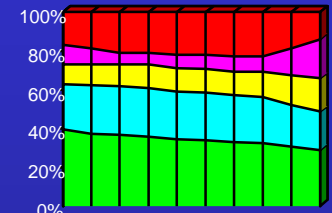
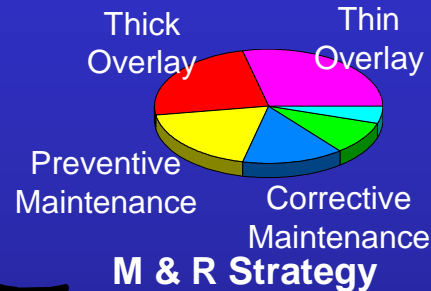
Using Both Types of Models

Probabilistic Models



\$ Distribution

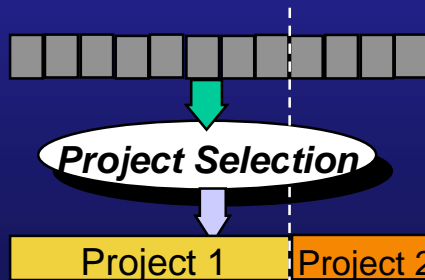
Network Level



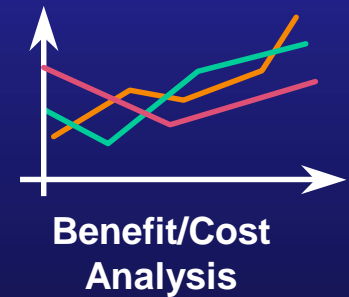
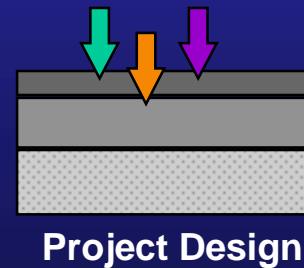
Network Health



Deterministic Models



Project Level

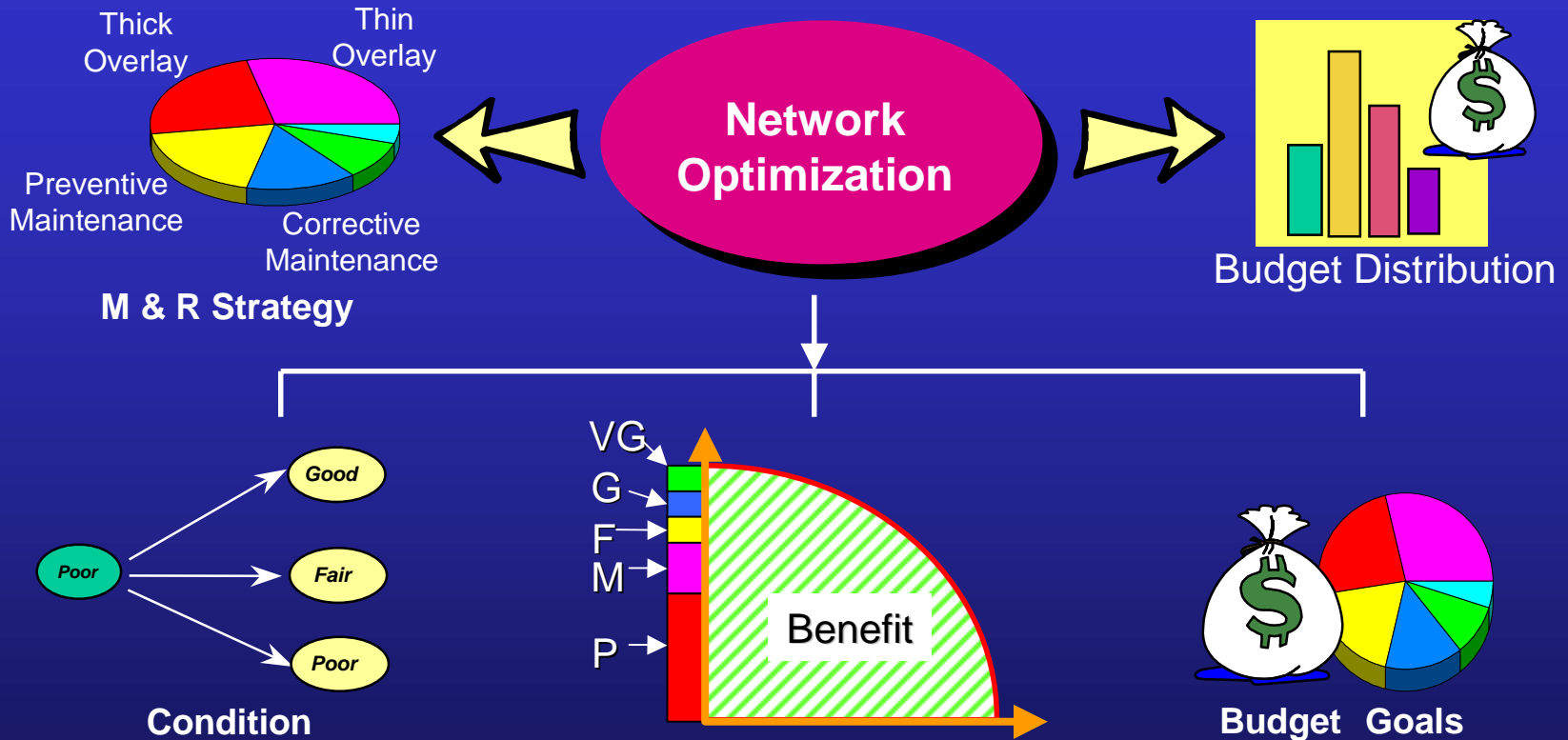


Benefit/Cost Analysis



Probabilistic and Deterministic Models Must Provide Equivalent Prediction Capabilities

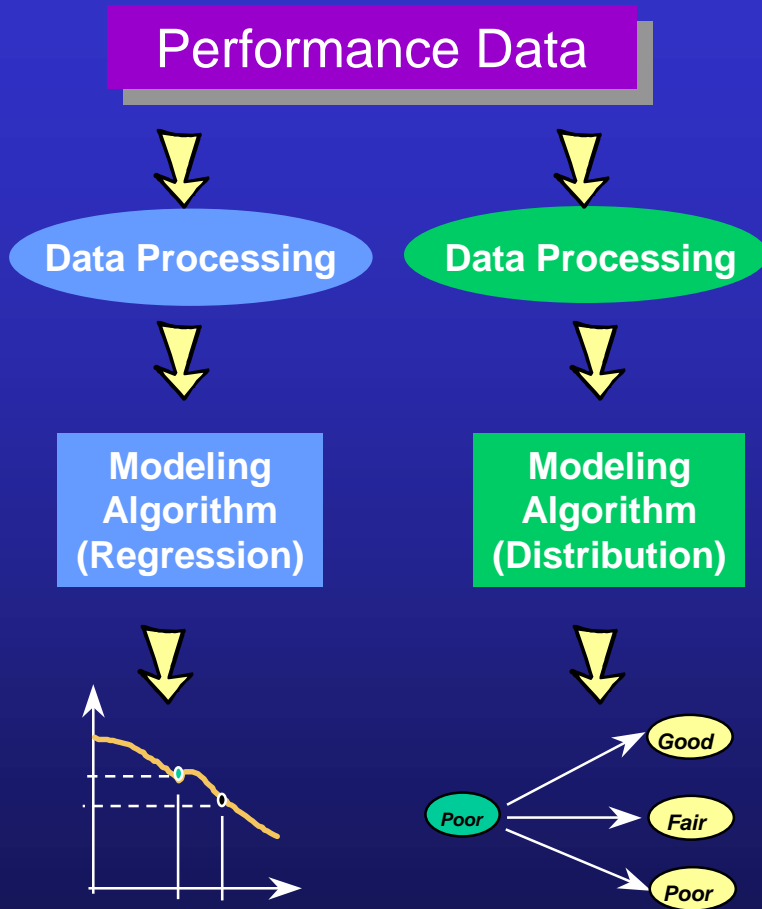
Using Both Types of Models



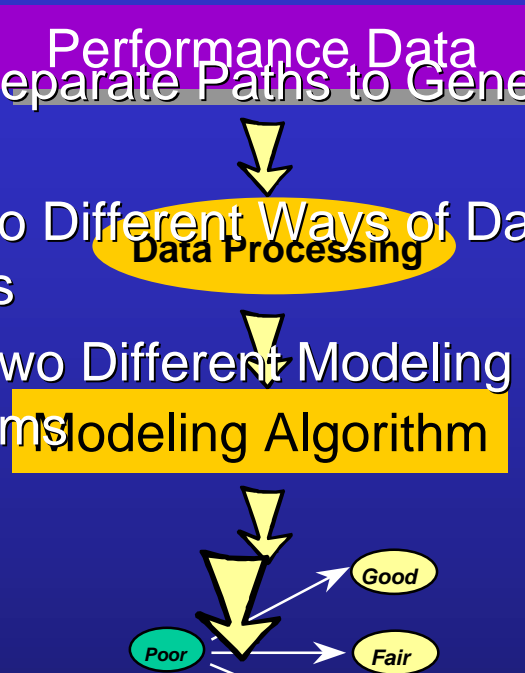
Probabilistic and Deterministic Models Must be in Synchrony to Provide Equivalent Prediction Capabilities



Approach to Synchronize Models



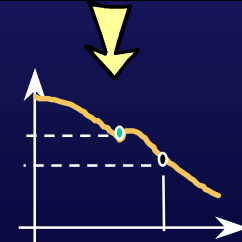
- Take Separate Paths to Generate Models
- Use Two Different Ways of Data Analysis
- Apply Two Different Modeling Algorithms



- Two Different Interpretations of Performance
- Prediction Models not in Synchrony



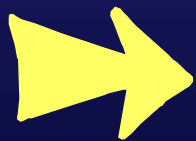
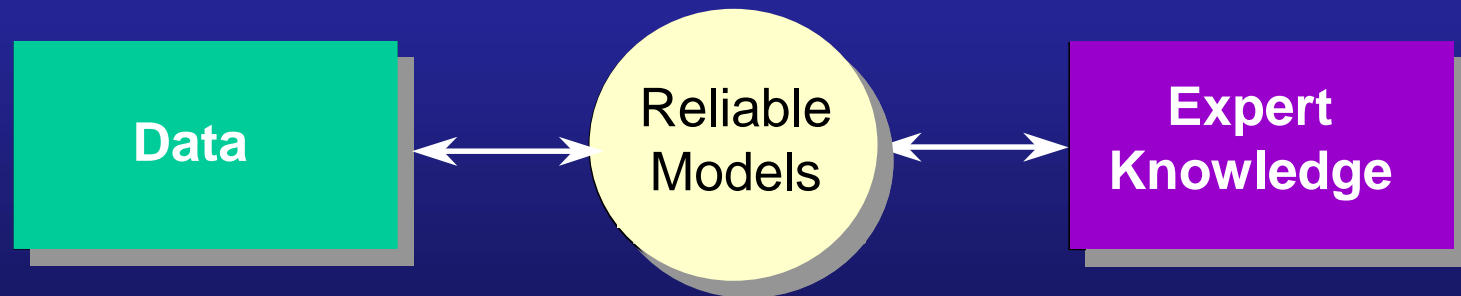
Develop a Single Process to Create Both types of Models



Dealing with Reliability

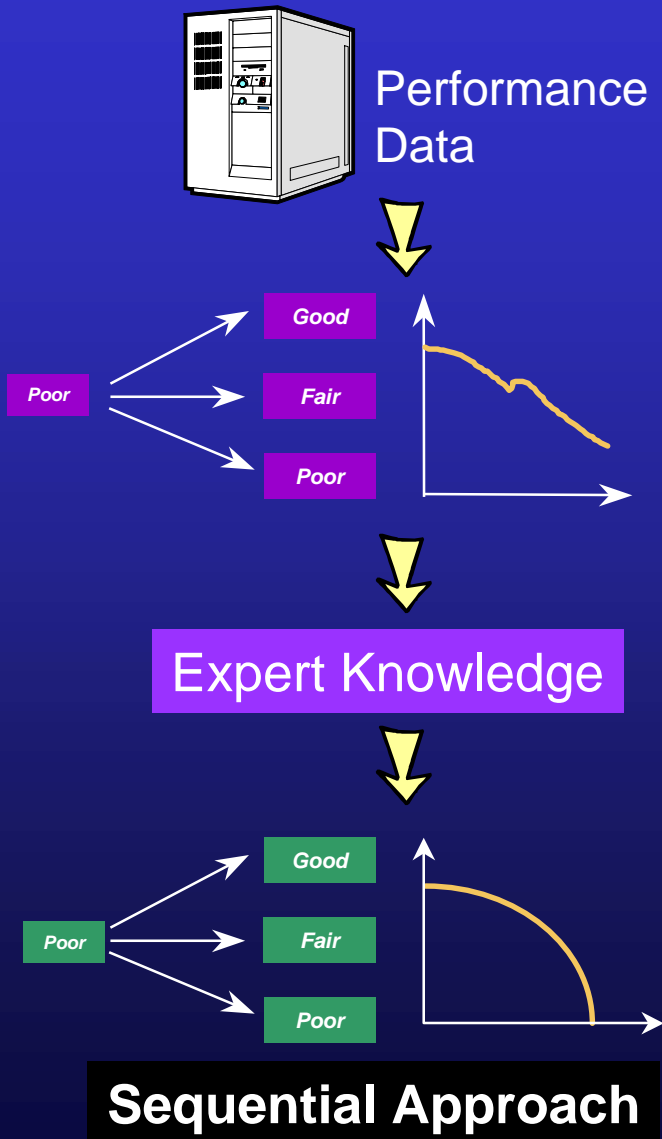
Our Definition:

“A reliable model is one that provides a true representation of the performance trends embedded in the data and is consistent with expert knowledge.”



Merge Data with Expert Knowledge
Into the Modeling Process

Combining Data and Expert Knowledge



Parallel Approach

- Models are first created with no Performance Data
- Expert Opinion is Used After the Fact
- Effect of Performance Data is often Ignored/Overridden

- Models not driven by Performance Data

Generate models that preserve the trends in the data and simultaneously comply with expert knowledge

Interpreting Expert Knowledge

Expert Knowledge



	VG	G	F	M	P
VGood	0.85	0.09	0.03	0.02	0.01
Good	0.00	0.73	0.22	0.04	0.01
Fair	0.00	0.00	0.71	0.24	0.05
Mediocre	0.00	0.00	0.00	0.75	0.25
Poor	0.00	0.00	0.00	0.00	1.00

Limitations of Traditional Approach

- Capturing expert input in probabilities is not practical
- Tedious/very demanding process
- Input Varies from one expert to another



We are soliciting information from experts in the wrong format.



Gather information in a format that experts can relate to.



Acquiring Data for Model Development

Soliciting Information from Experts

Questions to ask Experts:

1. What life would you expect from an action?
2. What is the life duration in each condition state?

Example:

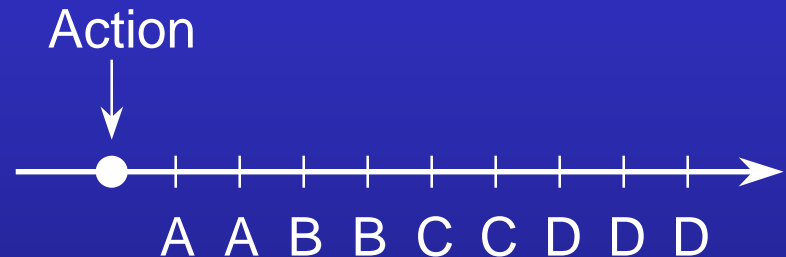
After a Thin Overlay:

1. 8 years of life

2. <u>State</u>	<u>Life</u>
Excellent	2 years
Very Good	3 years
Good	1 year
Fair	2 years



Extracting Information from Data



<u>State</u>	<u>Life</u>
Excellent	2 years
Very Good	2 years
Good	2 years
Fair	3 years

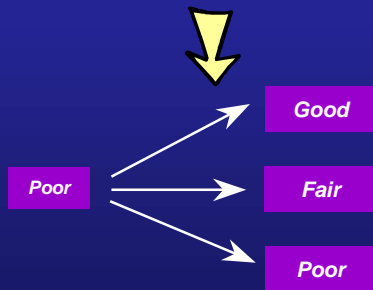
Extract information from experts and data records using life frequencies



Dealing with Minor Maintenance

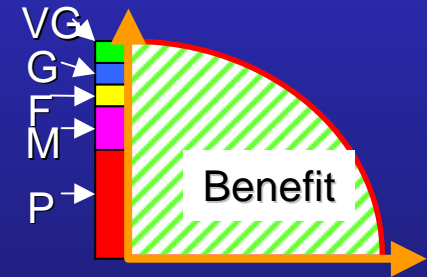
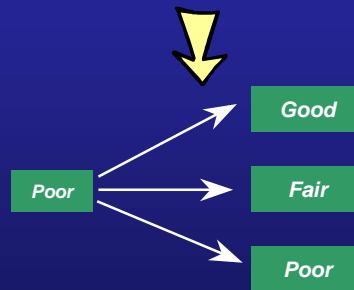
- No Maintenance

<u>Condition</u>	<u>Life</u>
V Good	1 years
Good	3 years
Fair	2 years
Mediocre	2 years
Total:	8 Years



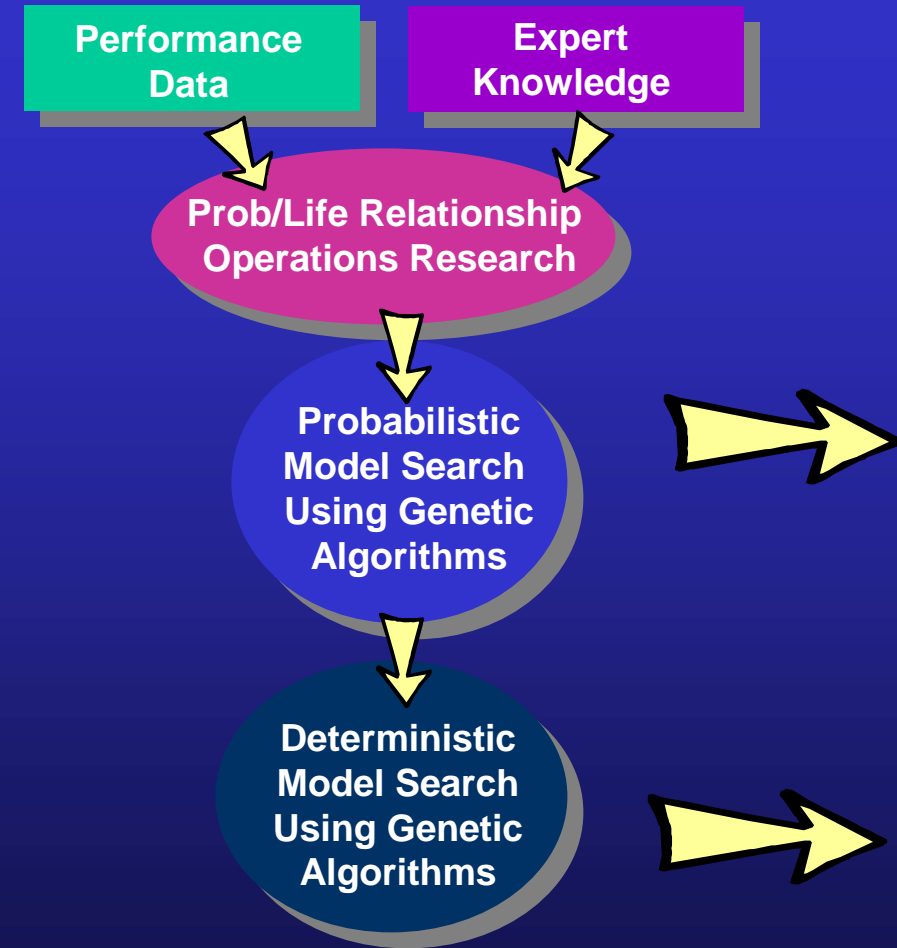
- Maintenance

<u>Condition</u>	<u>Life</u>
V Good	1 years
Good	4 years
Fair	3 years
Mediocre	2 years
Total:	10 Years



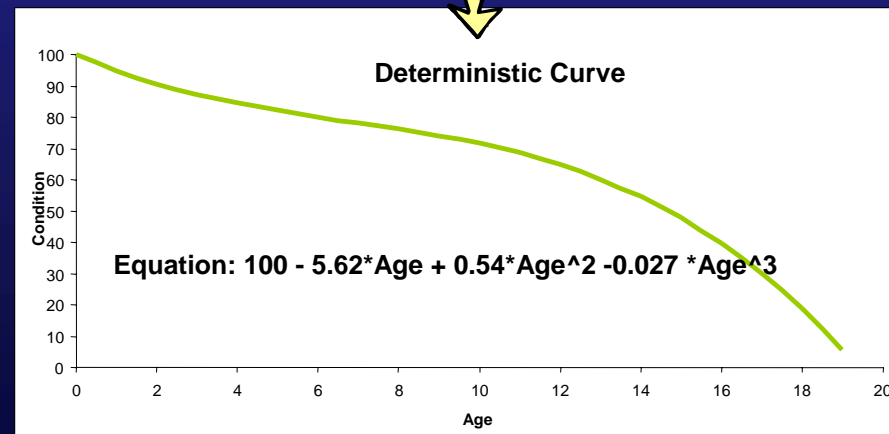
Apply Concept of Life Frequencies to Capture the impact of Minor Maintenance

Developing Models from Acquired Data

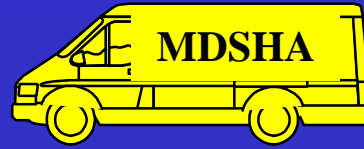


Condition	Life	Condition	Life
V Good	2 years	V Good	1 years
Good	1 years	Good	3 years
Fair	3 years	Fair	2 years
Mediocre	2 years	Mediocre	2 years

	VG	G	F	M	P
VGood	0.85	0.09	0.03	0.02	0.01
Good	0.00	0.73	0.22	0.04	0.01
Fair	0.00	0.00	0.71	0.24	0.05
Mediocre	0.00	0.00	0.00	0.75	0.25
Poor	0.00	0.00	0.00	0.00	1.00



OR & AI technology is applied to create models from the life frequencies format



Example Models

Expert Knowledge

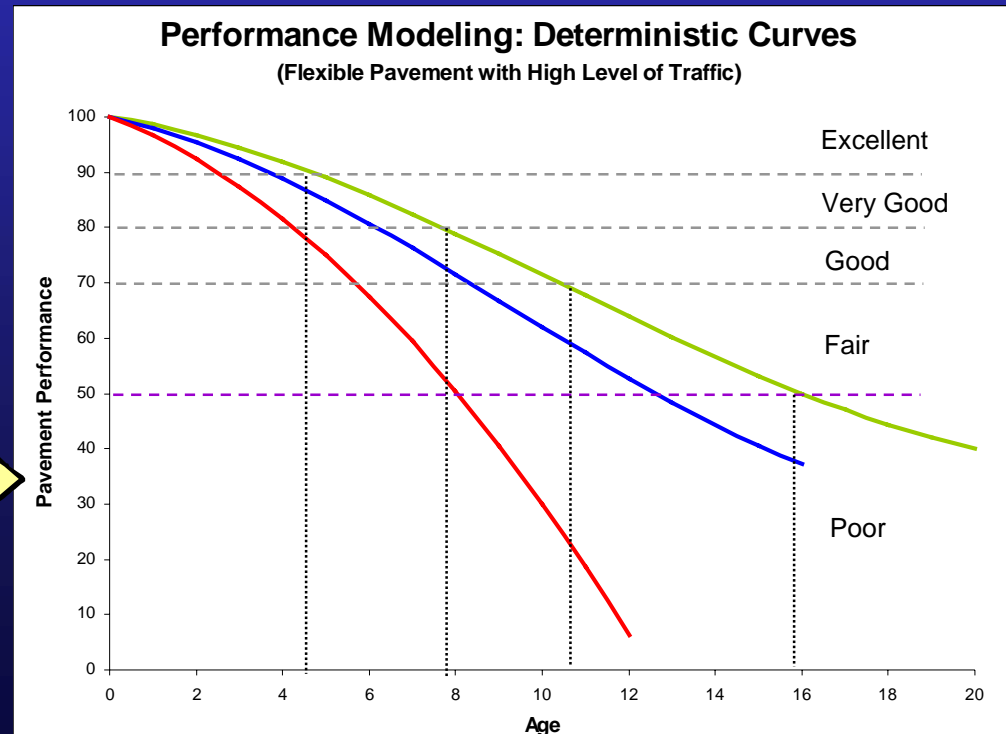
Performance Data

	Thick Overlay	Medium Overlay	Thin Overlay
A- Excellent	4.5	3.5	2
B - Very Good	3.5	2	1.5
C - Good	3	2.5	2
D - Fair	4.5	4	2.5
Total	15.5	12	8



Thick Overlay Matrix

	A	B	C	D	E
A	0.85	0.09	0.03	0.02	0.01
B	0.00	0.73	0.22	0.04	0.01
C	0.00	0.00	0.71	0.24	0.05
D	0.00	0.00	0.00	0.75	0.25
E	0.00	0.00	0.00	0.00	1.00



Performance Modeling: The Application

Adom - Asset Management System

File Edit Analysis Help

Performance Modeling Network Optimization Project Selection

Performance Modeling

Pavement Group
 Flexible - High Traffic - Mountain
 Flexible - High Traffic - Piedmont
 Flexible - High Traffic - Coastal

Modeling Scenario
 'Do Nothing' after a 'Thick Overlay' action

Add a Maintenance Action

Maintenance Action:
 Corrective Maint
 Preventive Maint

Condition State:
 Condition B
 Condition C
 Condition D

Weighting Factor
 Expert Data
 40 60

Probabilistic Models [Matrices]

TPM:

	A	B	C	D	E
A	0.86	0.00	0.00	0.00	0.00
B	0.13	0.79	0.00	0.00	0.00
C	0.01	0.19	0.73	0.00	0.00
D	0.00	0.03	0.22	0.75	0.00
E	0.00	0.00	0.05	0.25	1.00

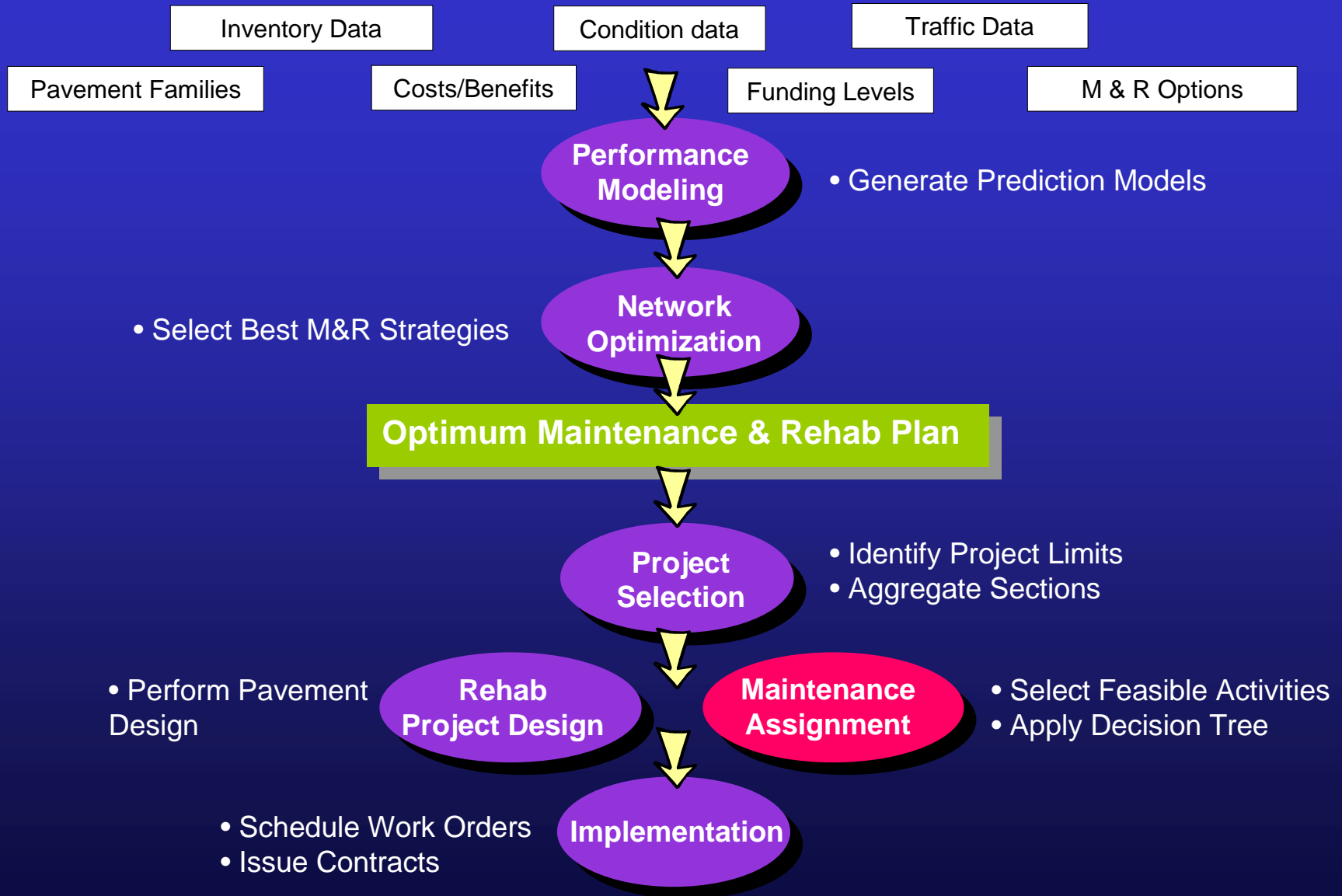
Deterministic Models [Curves]

Equation: $100 - 2.9802243709(\text{Age}) + 0.3146257698(\text{Age})^2 - 0.0166170746(\text{Age})^3$

Thick Overlay Deterioration

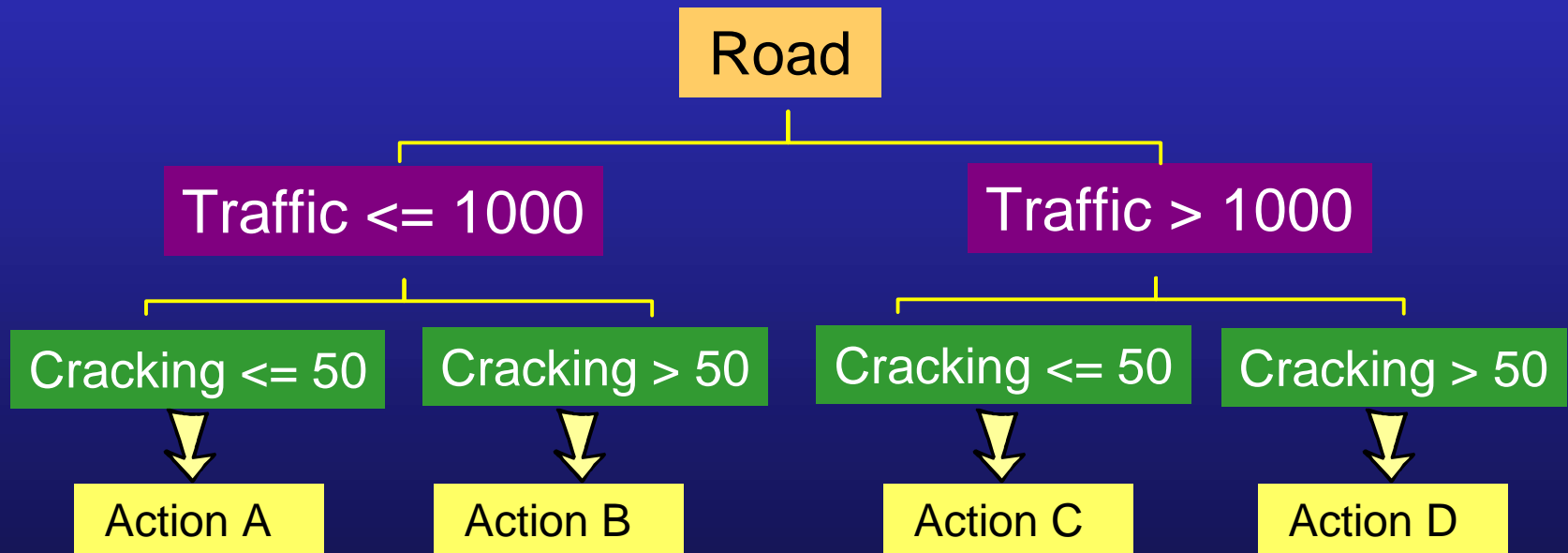
3/3/2002 6:10 PM

PMS: Analysis Process

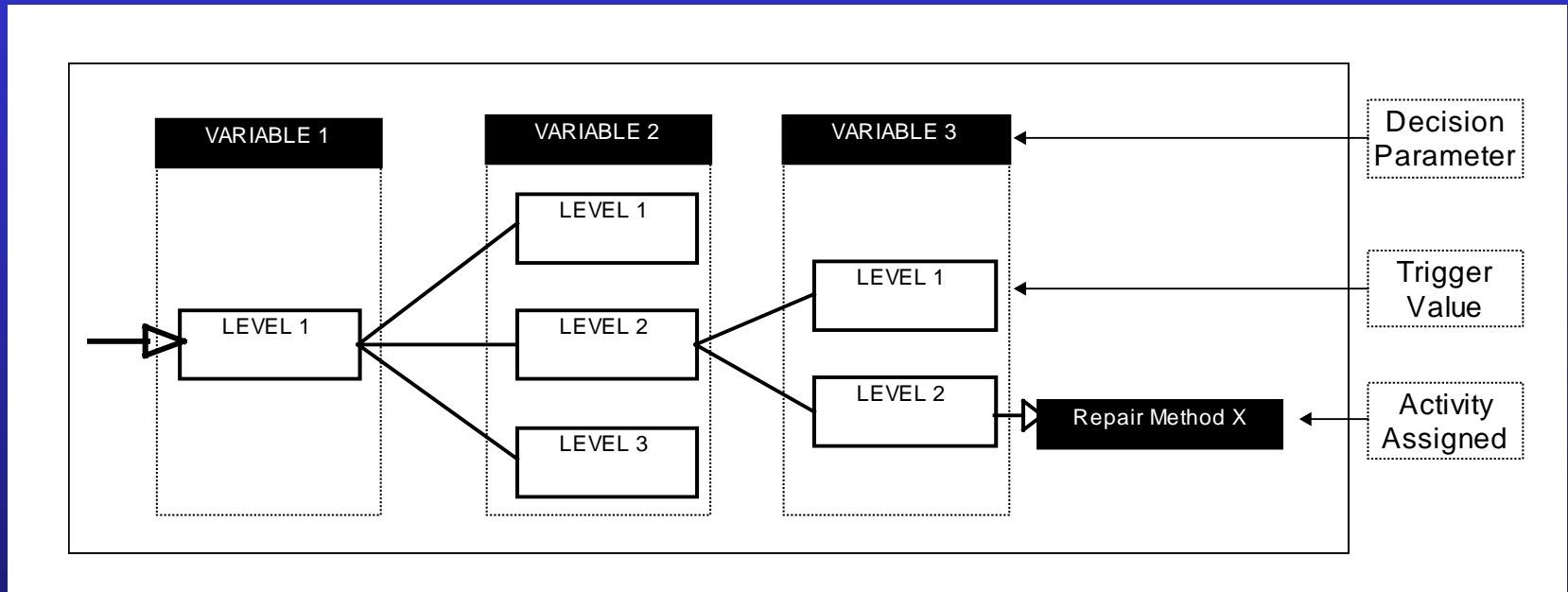


Utility of Decision Trees

- Identify Appropriate Maintenance Activities
- Mimic the Selection Rules Practiced by the State

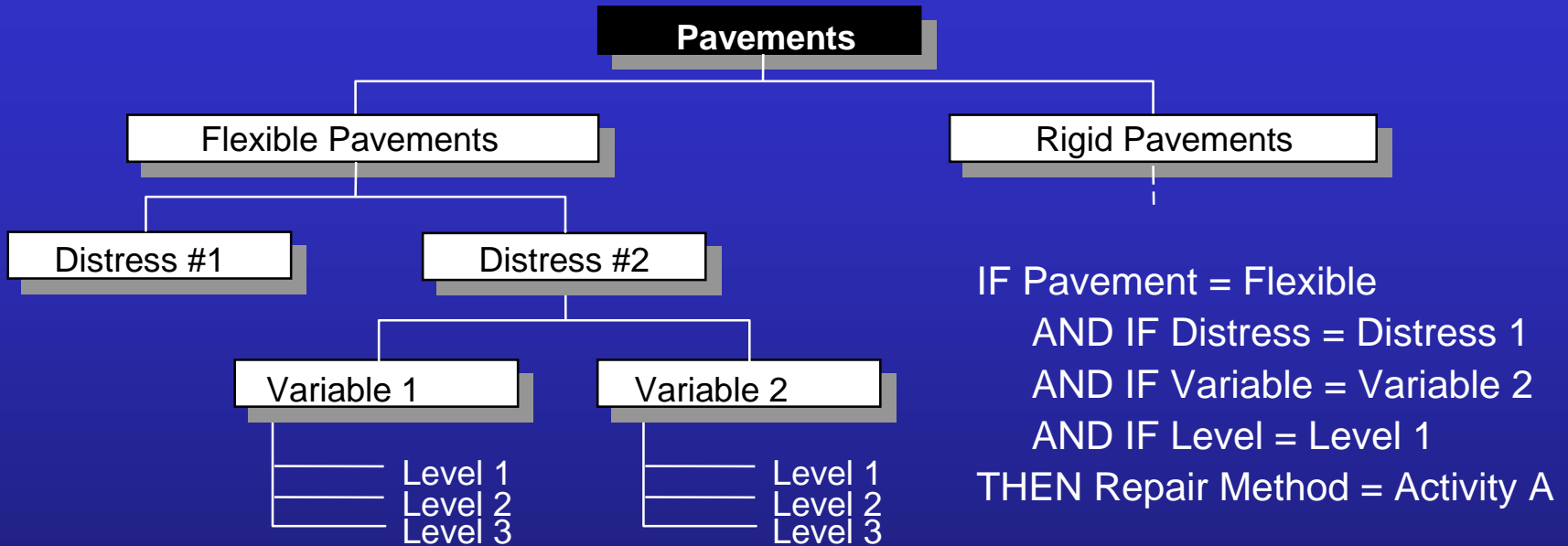


Building Blocks of Decision Trees



- Selection of the Decision Parameters
- Specification of the Threshold/Trigger Values
- Assignment of Repair Activities

Decision Tree Design



Traditional Decision Tree Process Uses Hard Coded IF_THEN Rules



Tree Can't be Updated/Modified
without Coding



Need a Flexible Decision Tree Builder that Does Not Rely on Hard Coded Rules

Flexible Decision Tree Design

- Represent Decision Trees with a set of Mathematical Equations Instead of IF-Then Rules
- Equations Adapt Dynamically to accommodate building and modifying Decision Trees

- Decision Parameters
- Threshold & Levels
- Repair Methods

Dynamic
Mathematical
Algorithm

Flexible
Decision Tree



An Adaptive Rule-Based Decision Tool that Evolves with Users Needs

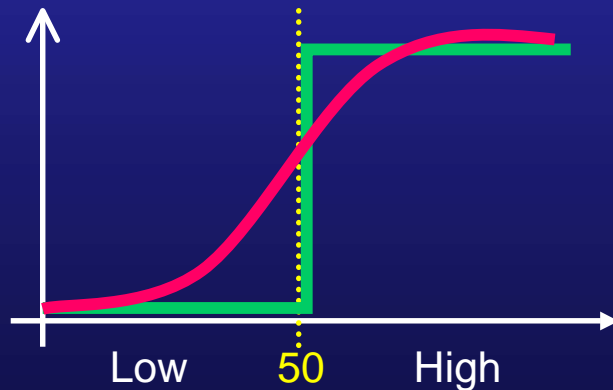
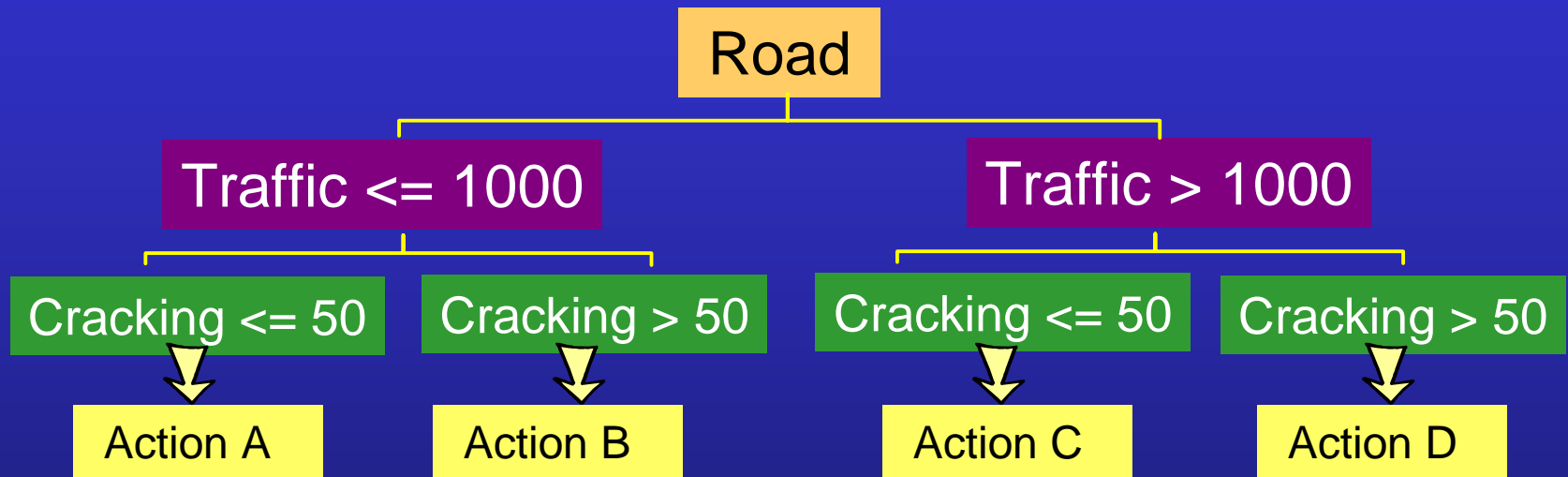
Decision Trees: The Application

The screenshot displays the Axiom - Asset Management System interface. On the left, a vertical toolbar lists various asset categories: All Assets, Pavements, Bridges, Roadside Features, Signs, Signals, Equipment, Landscape, Sound Walls, and Rest Areas. A tree view on the left side shows a hierarchy of Districts (1-7), Counties (Allegany to Worcester), and Road Types (Interstates, Non-Interstates). The main window features several tabs: Performance Modeling, Network Optimization, Project Selection, Performance Constraints, Unit Cost, Benefit, Aggregation, Budget Levels, Needs Parameters, Needs Activities, and Needs Assignments. The 'Needs Assignments' tab is active, showing a decision tree for 'Optimized/Thin Overlay'. The tree starts with a 'Start' node, which branches into 'OPI is Low' and 'OPI is High'. 'OPI is Low' further branches into 'Crack Index is Low' and 'Crack Index is High'. 'OPI is High' branches into 'Crack Index is Low' and 'Crack Index is High'. Each of these four branches leads to two 'RSL Index' outcomes: 'RSL Index is Low' and 'RSL Index is High'. The status bar at the bottom shows '3/3/2002' and '6:10 PM'.

```
graph LR; Start[Start] --> OPI_Low[OPI is Low]; Start --> OPI_High[OPI is High]; OPI_Low --> Crack_Low_L[Crack Index is Low]; OPI_Low --> Crack_High_L[Crack Index is High]; OPI_High --> Crack_Low_H[Crack Index is Low]; OPI_High --> Crack_High_H[Crack Index is High]; Crack_Low_L --> RSL_Low_L1[RSL Index is Low]; Crack_Low_L --> RSL_High_L1[RSL Index is High]; Crack_High_L --> RSL_Low_L2[RSL Index is Low]; Crack_High_L --> RSL_High_L2[RSL Index is High]; Crack_Low_H --> RSL_Low_H1[RSL Index is Low]; Crack_Low_H --> RSL_High_H1[RSL Index is High]; Crack_High_H --> RSL_Low_H2[RSL Index is Low]; Crack_High_H --> RSL_High_H2[RSL Index is High];
```

Work in Progress: Enhancing the Decision Process

- Build Smarter Decision Trees to Select Maintenance Activities



Empowering the Decision Trees with Fuzzy Logic

Thank you...

