

Probabilistic Assessment of Dependable Computer Systems

Joanne Bechta Dugan

Professor of Electrical & Computer Engineering

University of Virginia

jbd@virginia.edu

Galileo

Dynamic Fault Tree Analysis Tool



DFT: Dynamic Fault Tree Analysis

- Fault tree analysis (FTA) is a widely accepted methodology for reliability analysis and provides core functionality to PRA (Probabilistic Risk Assessment).
- Dynamic fault trees (DFT) extend FTA to allow accurate analysis of computer-based systems characterized by:
 - complex redundancy management
 - spares (cold, warm, pooled)
 - functional and sequence dependencies
 - hardware and software components
 - imperfect coverage and other common cause failures
 - phased missions

Static Fault Trees

- Combinatorial model (models combinations of events)
 - AND gates
 - OR gates
 - K-of-M gates
- New approach for solution: BDD (Binary Decision Diagrams)
- Advantages:
 - Exact analysis without cutsets
 - Can include repeated events
 - can include coverage modeling
 - Fast solution for very large models
- Disadvantage:
 - Static model: cannot include sequence dependencies

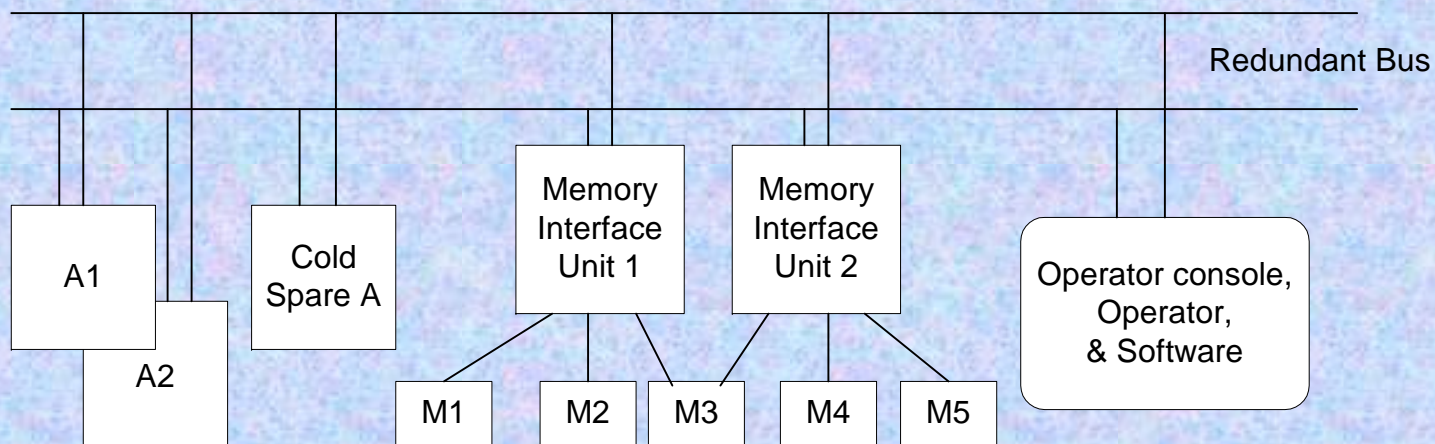
Dynamic Fault Trees

- Include special constructs for modeling sequence dependencies
 - functional dependencies
 - hot, warm and cold spares
 - priority-AND
 - sequence enforcing
- Solution: convert to Markov chain
- Advantages:
 - easier to use fault tree than Markov model directly
 - can model dynamic redundancy, shared pools of spares, etc
- Disadvantage:
 - state space explosion -- worst case exponential in number of basic events

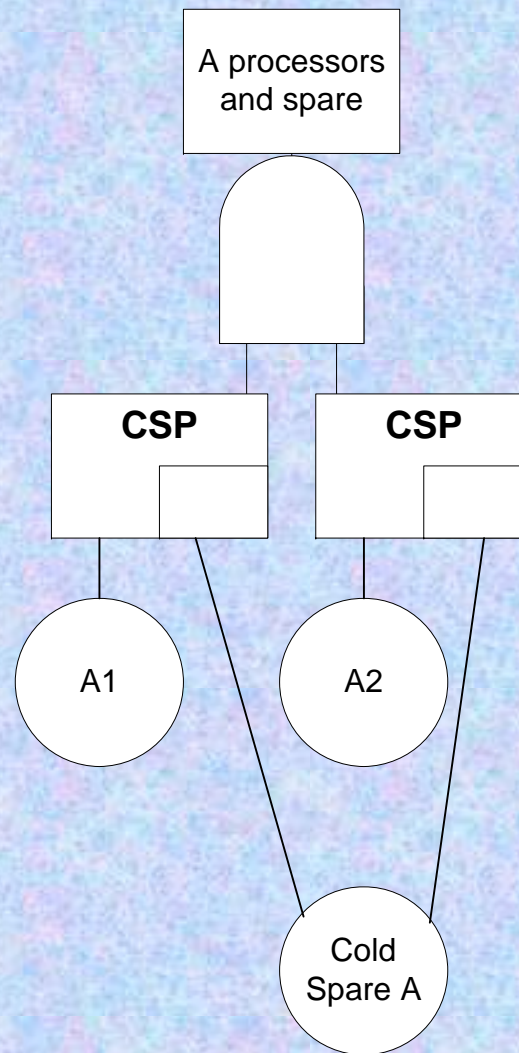
Coverage Modeling

- Adaptive (computer-based) systems can exhibit multiple failure modes
- *Covered* (benign) failure can be handled automatically
 - error is detected and located
 - switch in spare or bypass faulty component
 - system can continue operation without manual intervention
- *Uncovered* failure is globally malicious
 - undetected error escapes from embedded system
 - faulted component cannot be disabled
 - malicious behavior confuses recovery procedures
- System dependability measures are very sensitive to coverage
- Good techniques exist for incorporating coverage into static and dynamic fault trees.

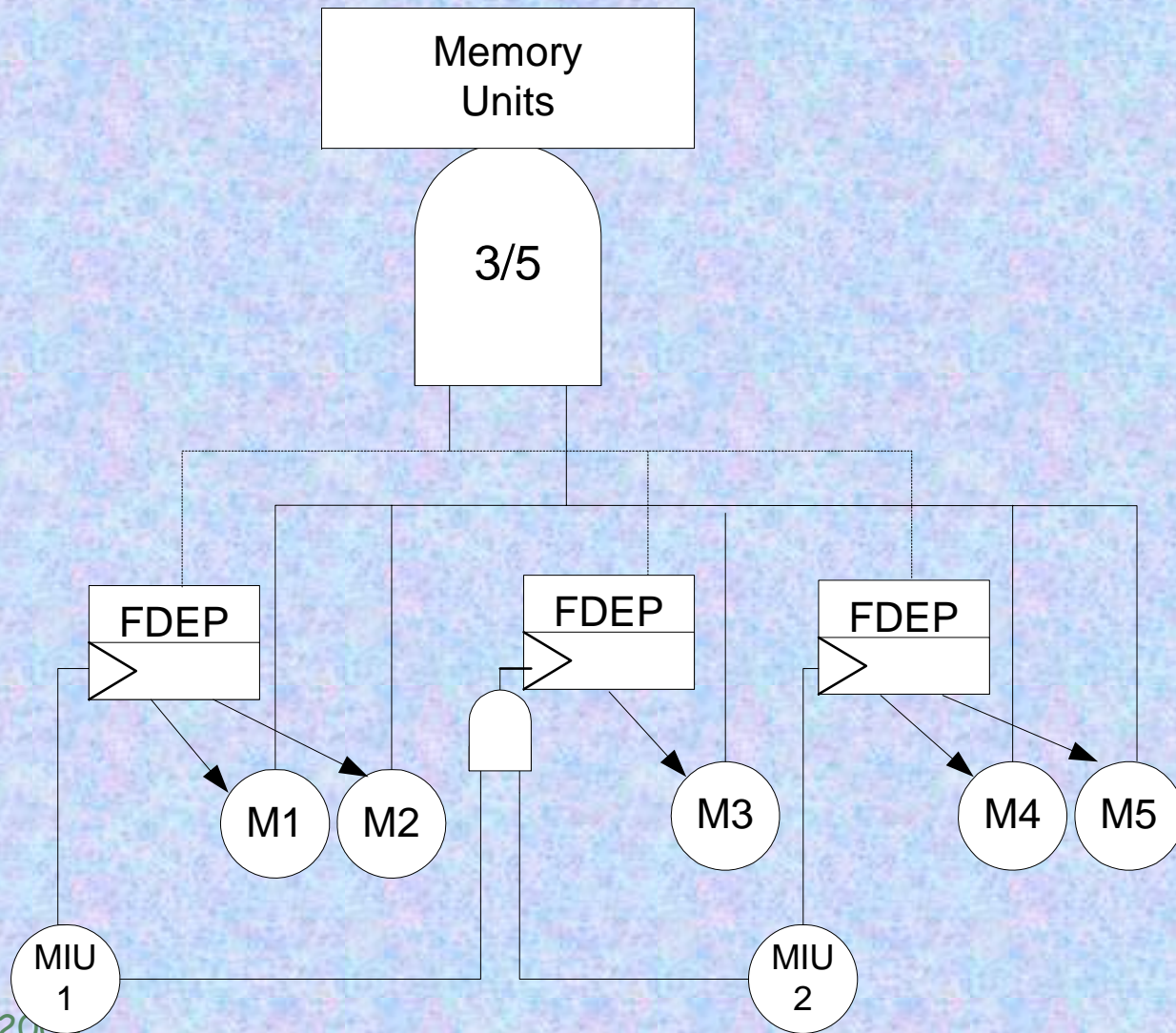
HECS: Hypothetical Example Computer System



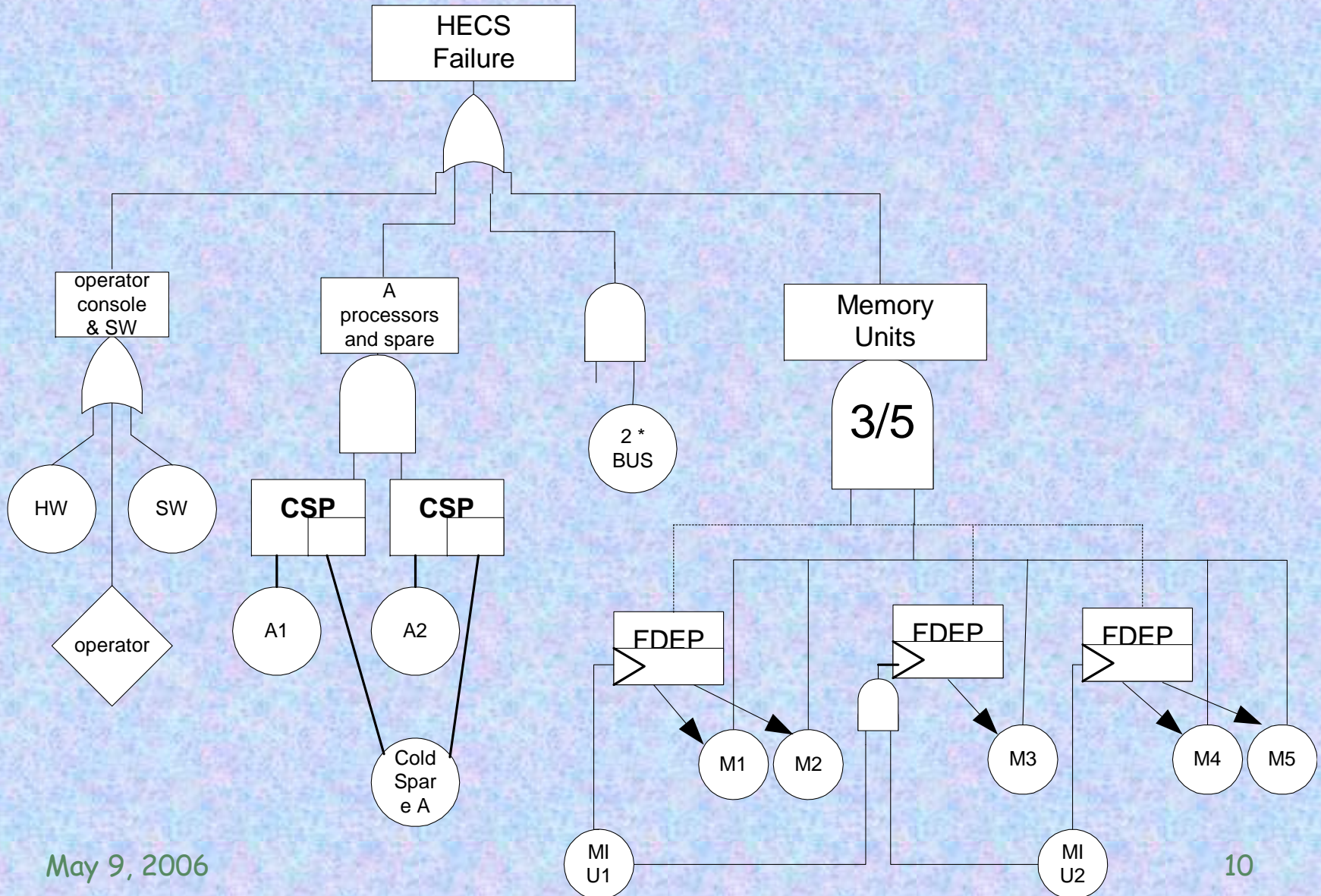
Modeling the processors



Modeling the memories



Full dynamic fault tree model



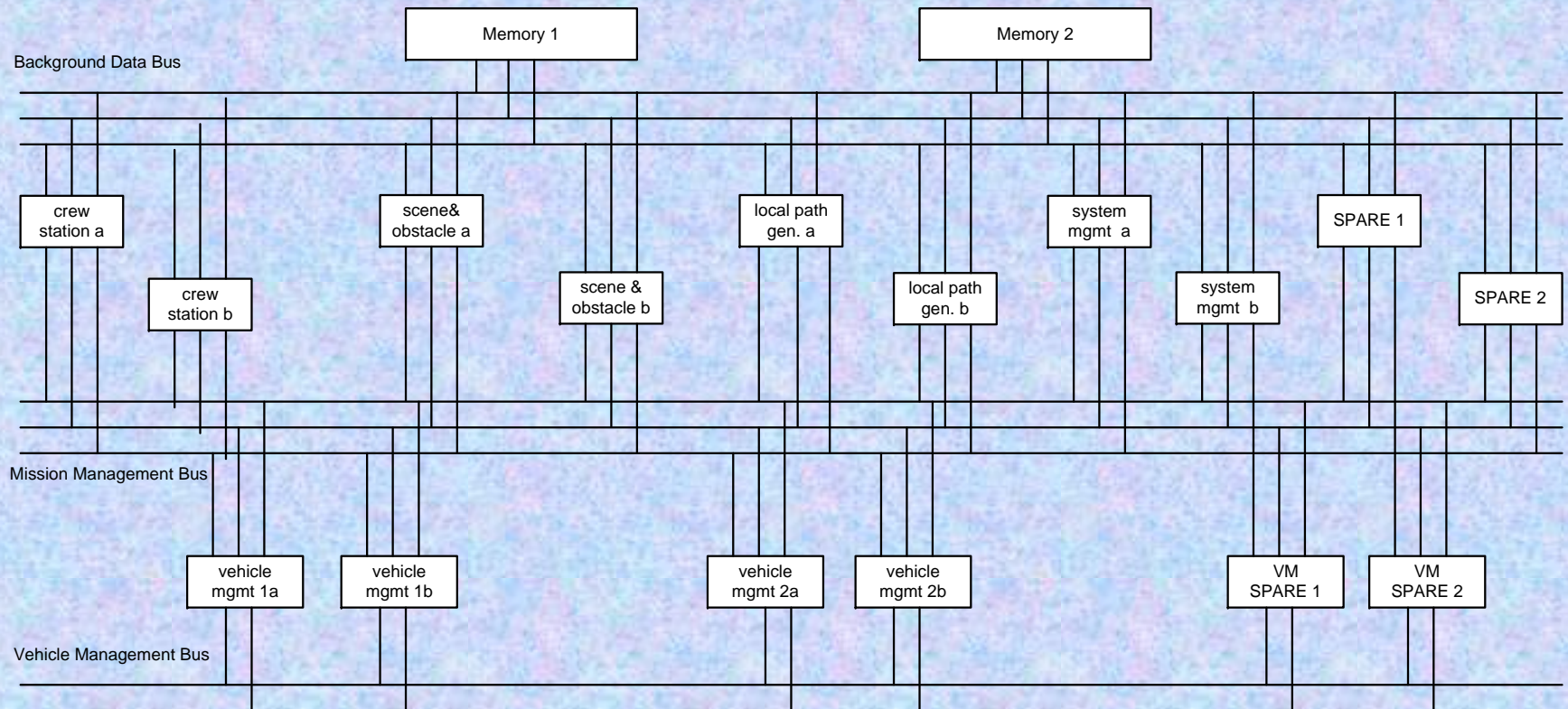
May 9, 2006

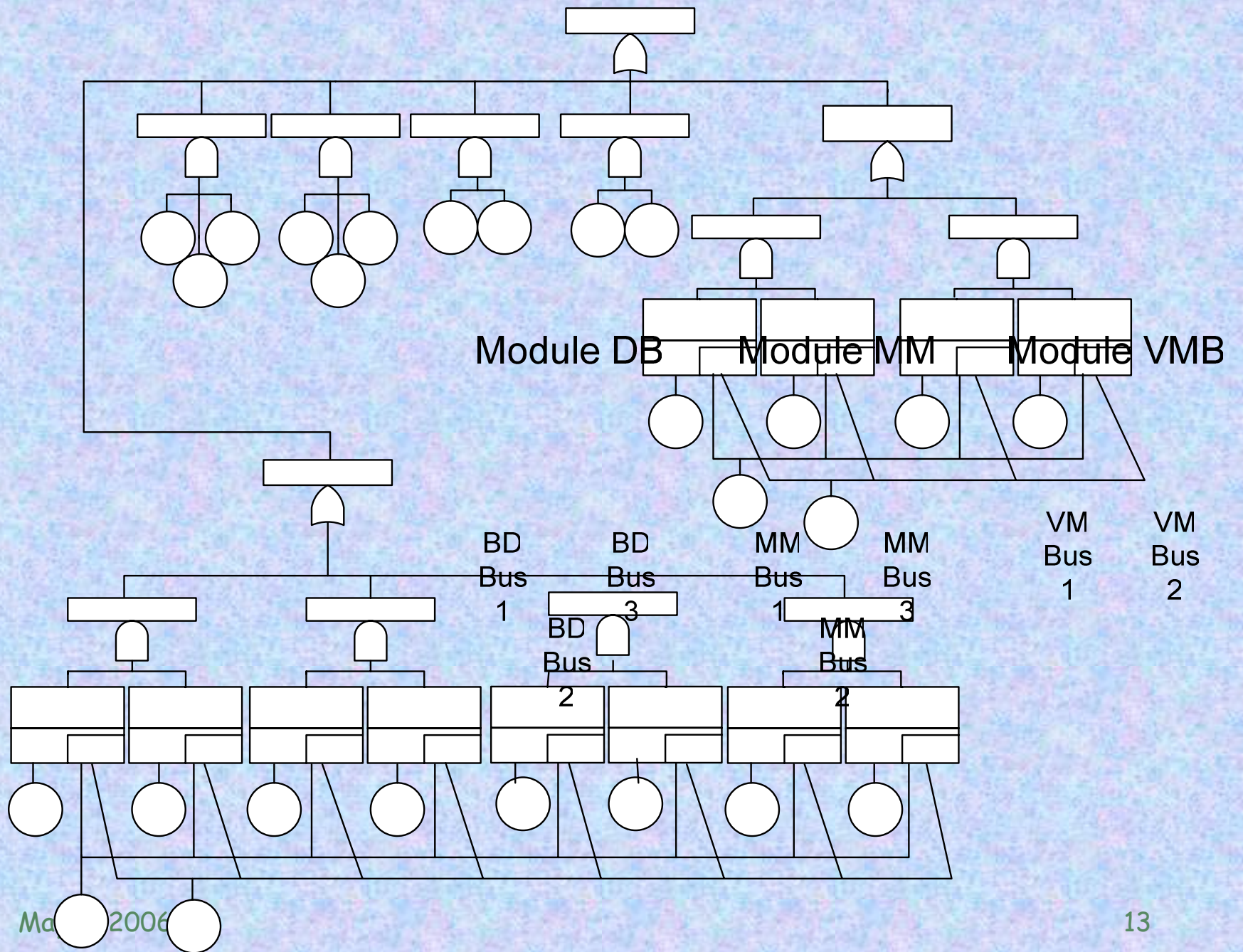
10

Mission Avionics System Example (MAS)

- The success/failure of the system is driven by the need to provide certain software functionality
 - crew station management
 - scene&obstacle processing
 - local path generation
 - system management functions
 - vehicle management
- Fault tolerance is achieved via redundant processors (hot spares), pools of cold spares and redundant buses.

MAS system architecture



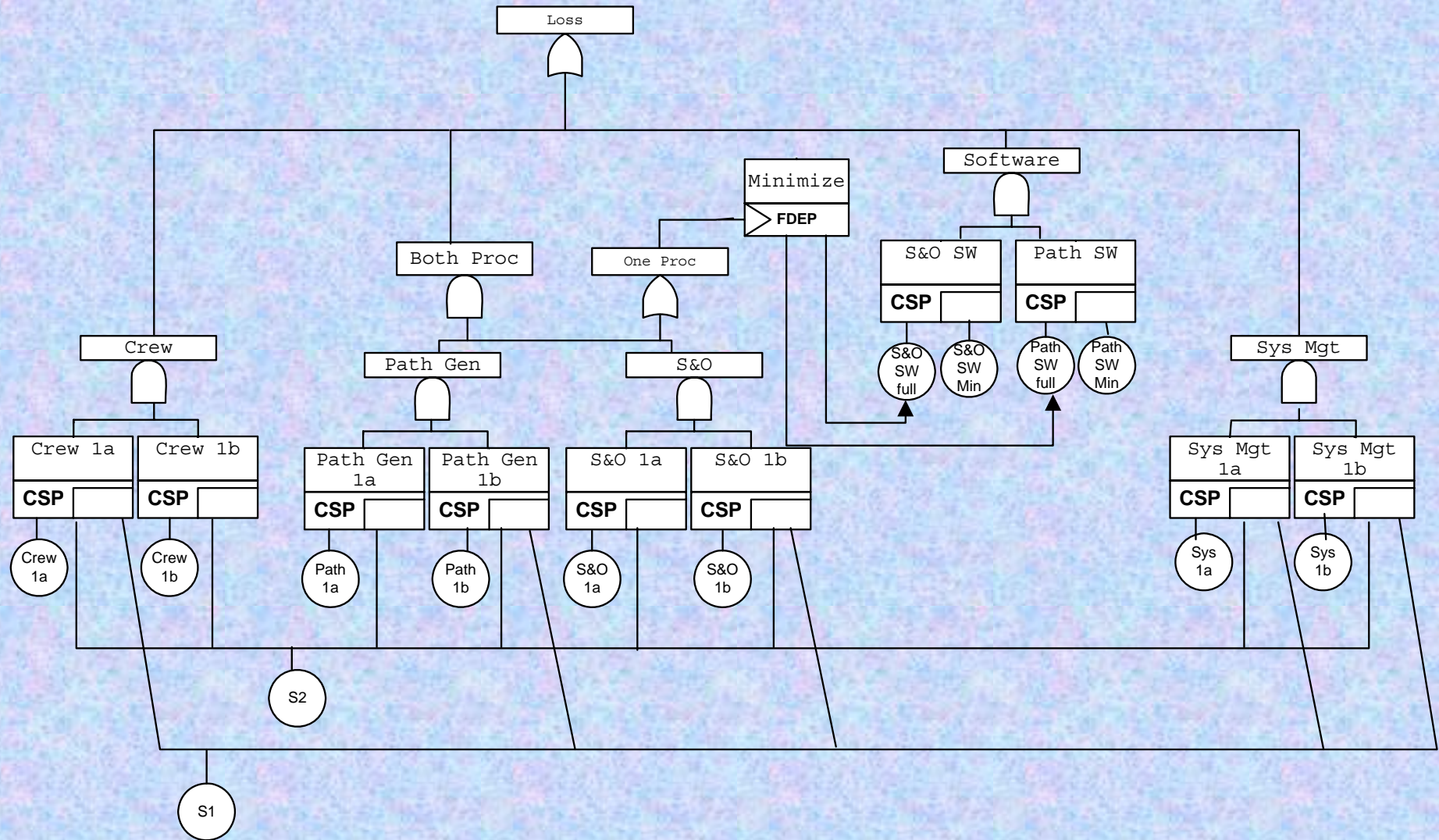


Ma 2006

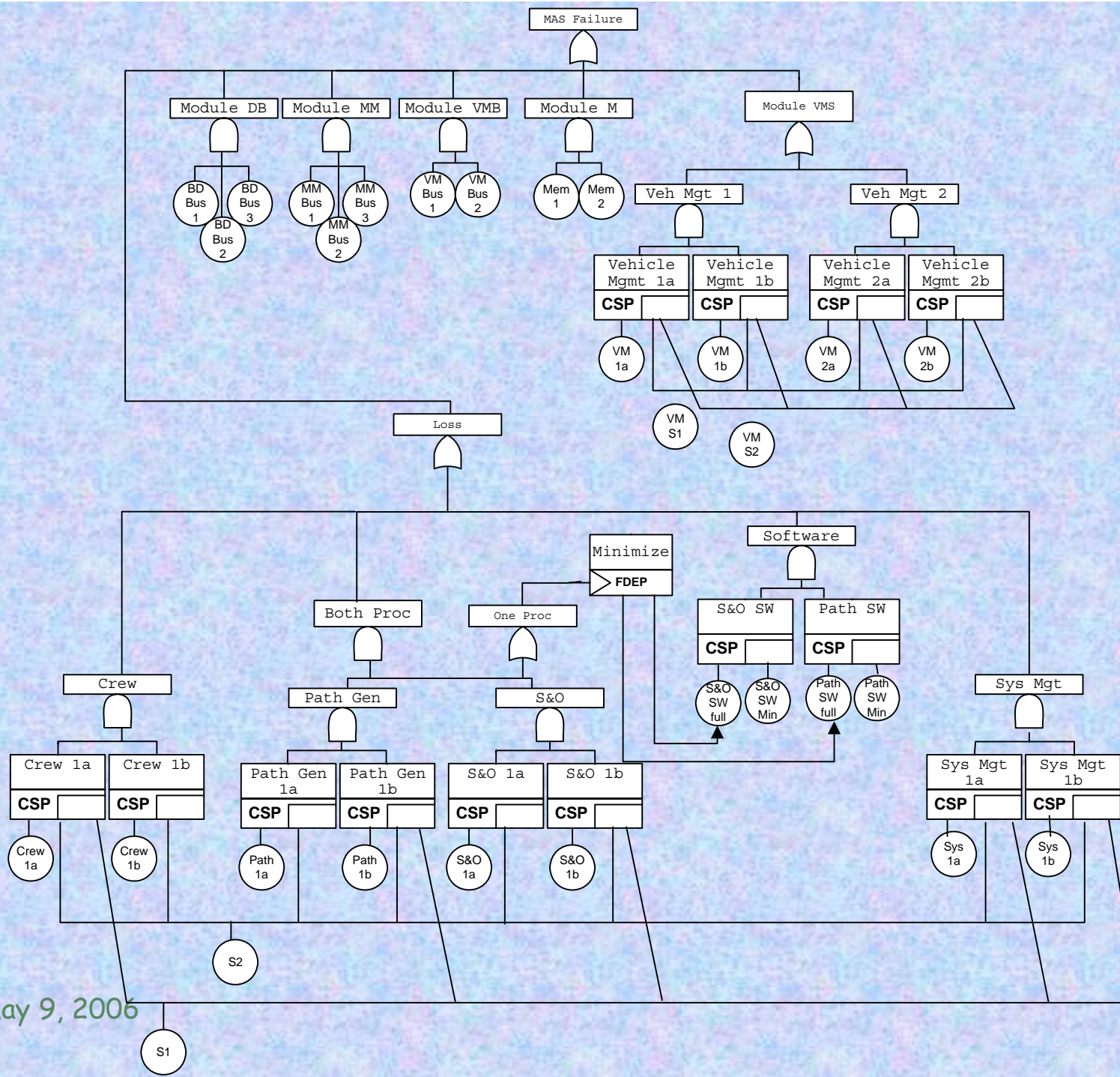
MAS Redundant Software Architecture

- Now consider the path generation (PathGen) and Scene&Obstacle (S&O) functions.
- Each function requires single processor to provide full functionality
- There is also a reduced version of each function that can provide minimum functionality (PathGenMin & S&Omin)
- In the event of a detected software fault in PathGen the system can switch to PathGenMin. (Same for S&O)
- Further, if there are no longer 2 full processors available, the system will switch to PathGenMin and S&Omin running on a single processor.

Redundant Software MAS model



May 9, 2006



May 9, 2006



ADORA

Diagnostic Decision Trees

(from Qualitative & Quantitative reliability Analysis)

Diagnosis

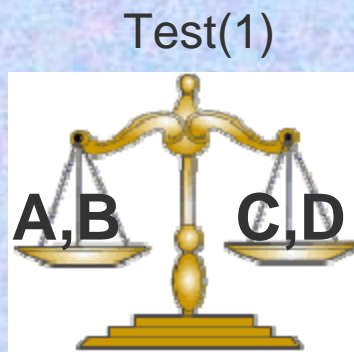
- Process of identifying the root cause of the failure of a complex system.
- Can be presented as a decision tree of tests.
- The decision tree is formed by identifying the best possible test ordering.



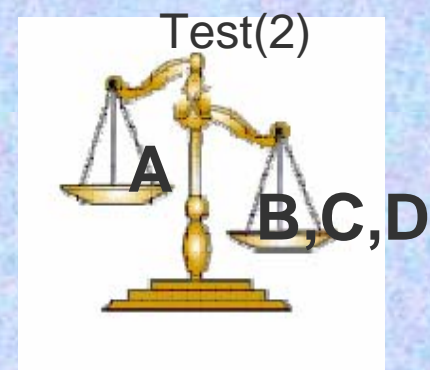
Traditional Diagnosis

Generate decision trees using a set of tests that implicate/exonerate sets of components.

1. Initially expert systems sequenced tests according to information gain (entropy reduction):



vs.



Traditional Diagnosis (cont.)

2. Lately, Expert Systems such as Sequential Diagnosis and Greedy Diagnosis utilized the probability of failure of components to sequence tests according to information gain (entropy reduction):

Test(1)



vs.

Test(2)



Problems with Traditional Diagnosis

- The assumption that one component has failed, thus try to identify one component.
- No redundancy in system (no hidden failures)
- However, failures are more complex than single cause:
 - Dependence
 - Redundant
 - Sequential
 - etc

How to model complex failure models?

- Using the DFT (dynamic fault tree)
 - Captures sequence, dependency & redundancy
- DFT analysis produces:
 - Reliability analysis
 - Minimal cutsets (structural analysis)
 - Sensitivity analysis
- DFT analysis software available: Galileo, Relex, etc.

Our Solution

ADORA (Automatic Diagnosis based On reliability Analysis)



- A methodology that **improves diagnosis**:
 - by **bridging the gap** between reliability analysis in the design phase and diagnosis performed in the usage phase.
 - by **reducing the dependence** on human expertise by systematically automating diagnosis.

Software Quality Assessment with Bayesian Belief Networks

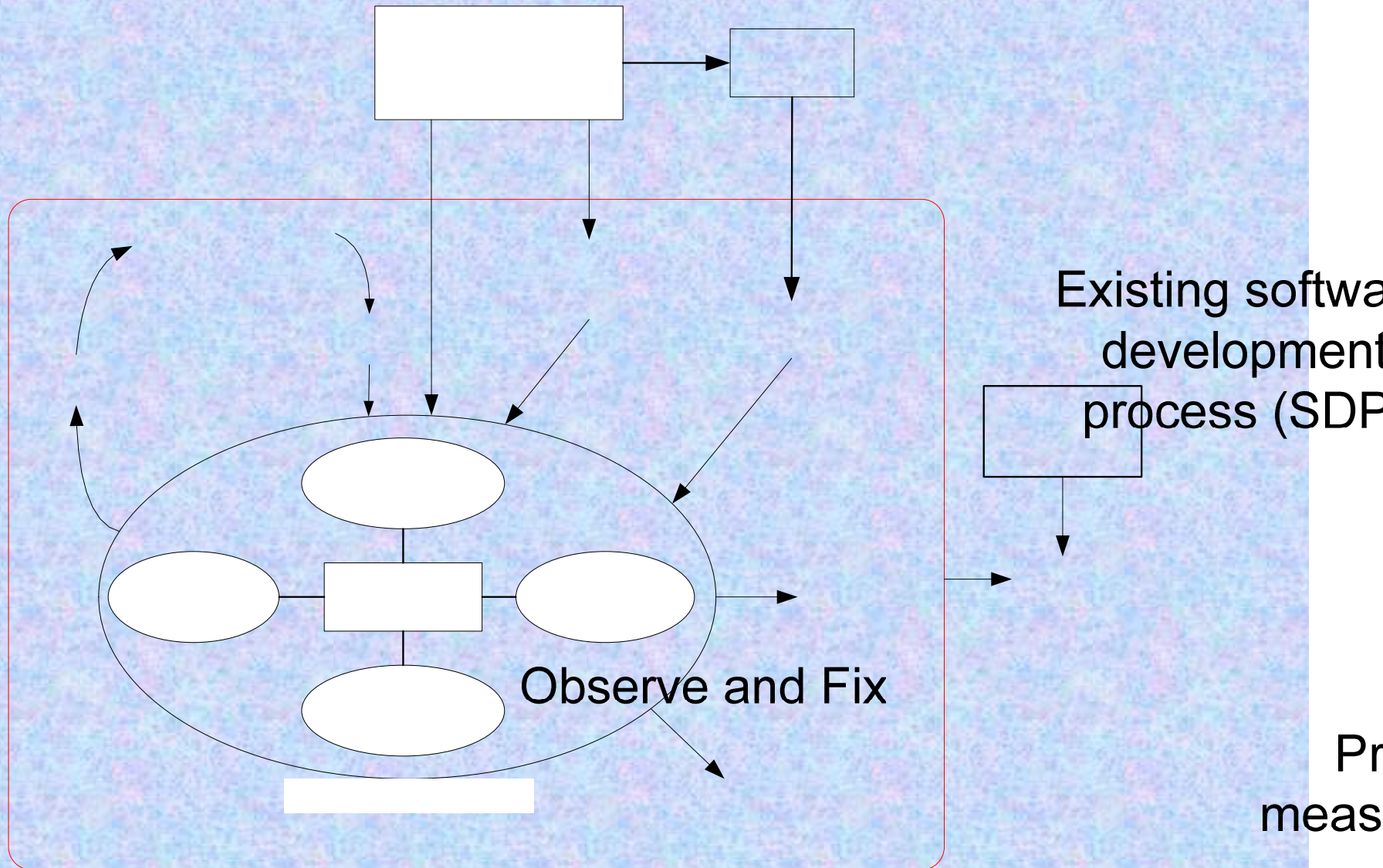
Ganesh J. Pai and Joanne Bechta Dugan
{gpai,jbd}@virginia.edu

University of Virginia, C.L. Brown Department of
ECE, Charlottesville, VA 22903

Problem

- **Pervasive use of software as the central control-component**
 - State of the art
 - Precisely stated requirements and formal development
 - Choose the process to assure that dependability requirements are met
 - State of the practice
 - A development process may already be in place
 - Requirements may be ambiguous
- **Software quality assessment (SQA)**
 - Does the process produce a software product of desired quality?
 - Existing methods
 - Regression models
 - Use product metrics and sometimes process metrics
 - Influence of diverse factors not considered
 - Inadequate to appreciably explain defect content
 - Checklist-based procedures (NASA procedures and guidelines 303-PG-7120.2.1.A)
 - Deterministic in nature, whereas factors being examined are probabilistic
- **Develop a probabilistic approach to SQA**
 - Consider diverse sources of information *i.e.* process and product metrics, subjective information
 - Evaluative and explanatory analysis capabilities

Approach



May 9, 2006

Evidence

26

Evaluate

Approach

- **Bayesian belief networks (BBN) for SQA**
 - Directed acyclic graphs representing probabilistic relationships between variables
 - Specifying the network structure
 - Build a model of the software process
 - Data flow, showing input, output, agents and activities
 - Model entities annotated with desired properties
 - BBN built algorithmically from the process model
 - Model is examined to see if the causal structure makes sense
 - Numerical specification
 - Empirically obtained
 - Distributions of metrics/ other variables obtained from version history
 - Expert opinion or subjective judgement
 - Prior distributions which reflect unknown knowledge
 - Model updating and feedback
 - Update model when data/evidence is available
 - Evidence @ input → refined output (defect content) estimation
 - Evidence @ output → potential cause of observation (feedback)

Importance/Benefits

- **Both the process and product are considered in SQA**
 - Items from the checklist based approach can be included
 - Both deterministic and probabilistic analysis is possible (since deterministic is a special case of probabilistic)
 - Formalising the reasoning behind quality analysis
 - Repeatable, robust and highly flexible framework
 - Reasoning and decision-making under uncertainty
 - Reasoning about possible factors influencing observations of poor quality *i.e.* resource allocation to problem areas
- **Evaluating process assertions about product quality**
 - Use model estimations to establish whether a process purported to produce high quality software, actually does so
 - A mechanism for comparing processes
 - A mechanism for sensitivity analysis when key process activities change
 - Model learning
 - If significant data is available, the BBN model structure and parameters can be learned