Improving the Reliability of Mobile Software Systems through Continuous Analysis and Proactive Reconfiguration

Sam Malek, David Kilgore
Ibrahim Elhag

George Mason University

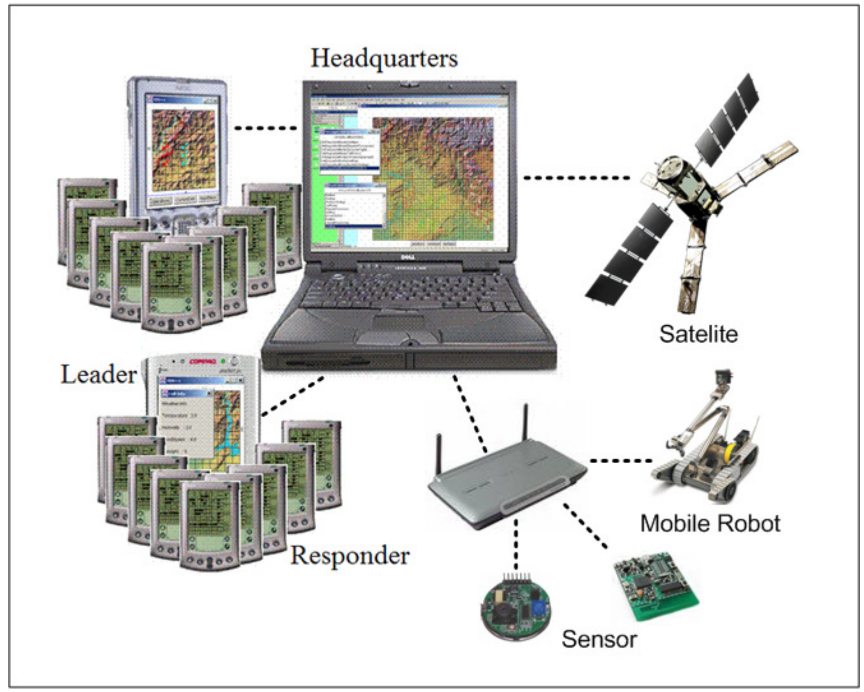
Roshanak Roshandel

Seattle University

ICSE NIER – Vancouver, BC, Canada, May 2009

Motivation

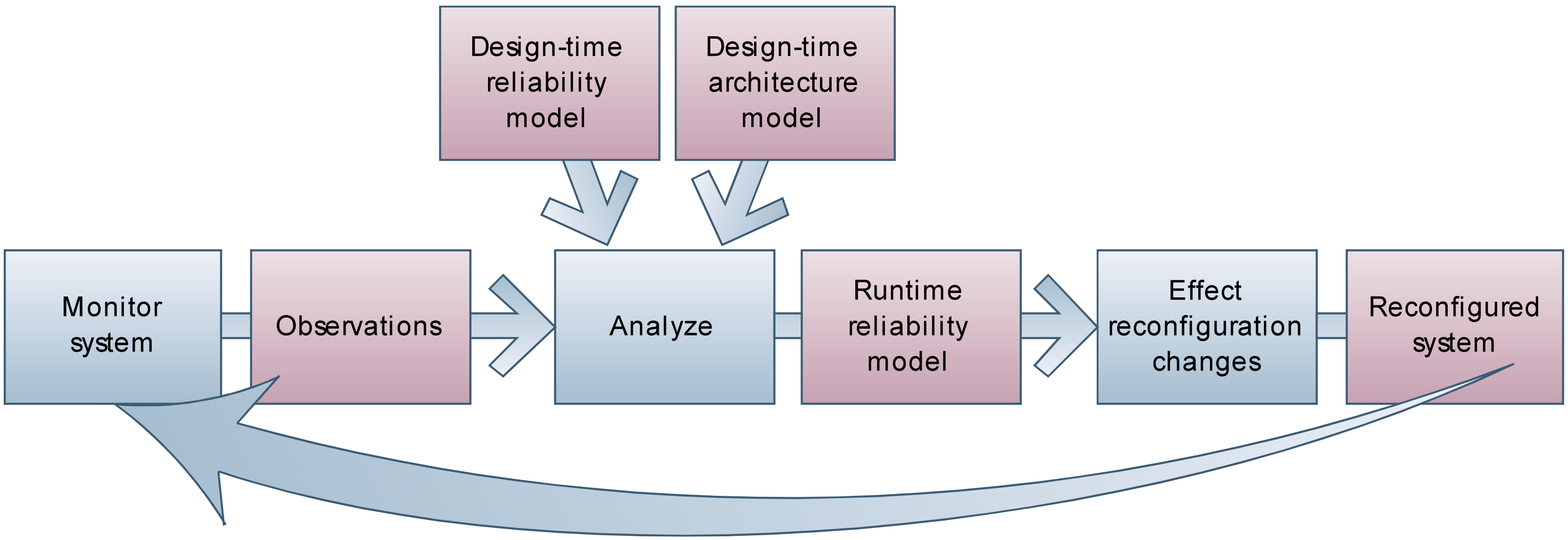
- Proliferation of mobile and pervasive software systems
- Increasingly deployed in safety or mission critical settings
- Existing reliability analysis approaches are not suitable
  - Dynamic configuration
  - Fluctuating execution context
  - Changing operational profile



Challenges

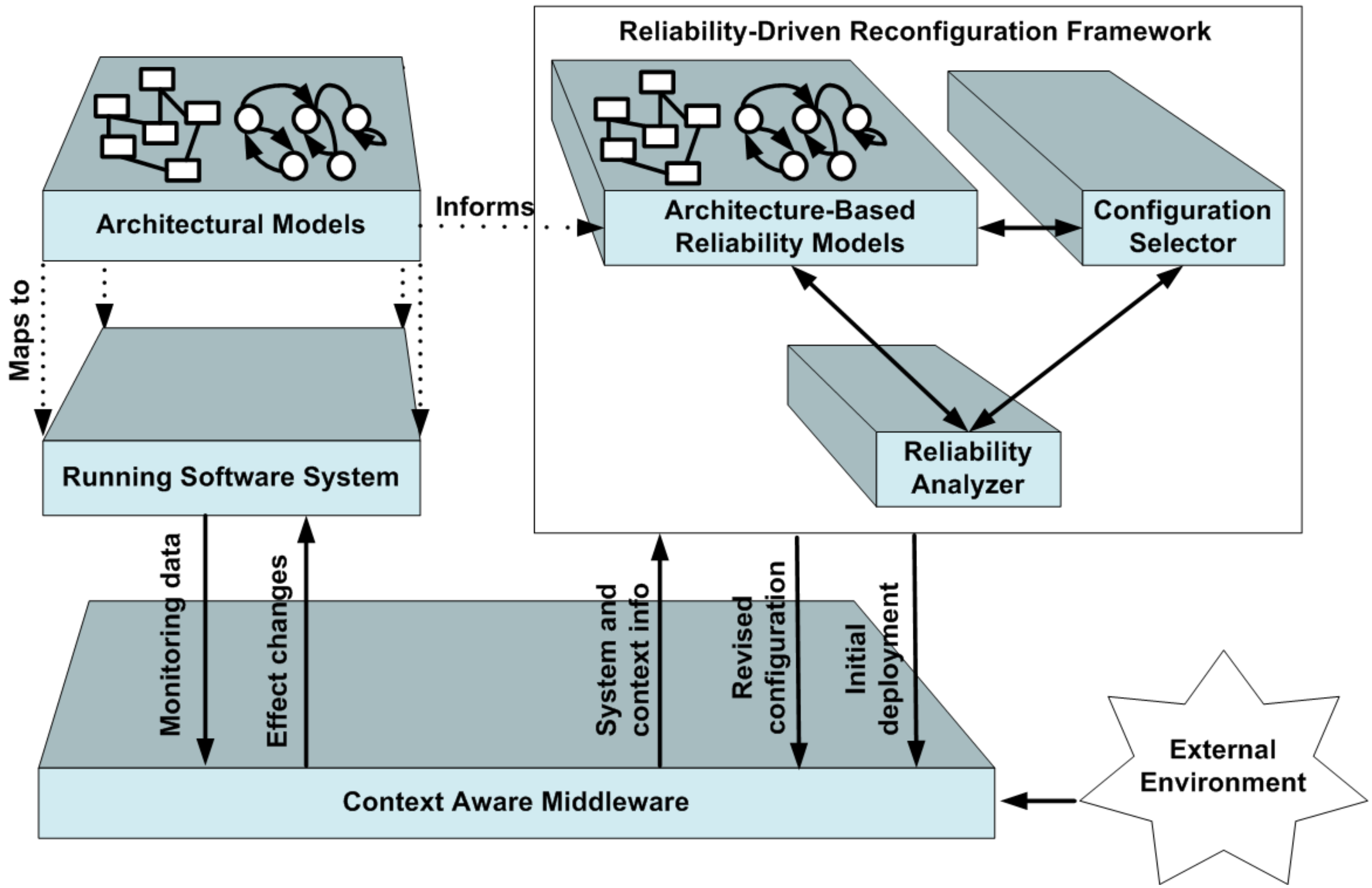
1. Impact of Context on Reliability
   – Internal vs. external faults
2. Impact of Dynamism on Reliability
   – Impact of adaptation on reliability
3. Difficulty of Predicting Reliability
   – Is system's past reliability indicative of its future reliability?
4. Granularity
   – Component-level as well as the system-level
5. Scalability
   – Efficient yet fine grained analysis

The Process



Design-time reliability model
Design-time architecture model
Monitor system
Observations
Analyze
Runtime reliability model
Effect reconfiguration changes
Reconfigured system

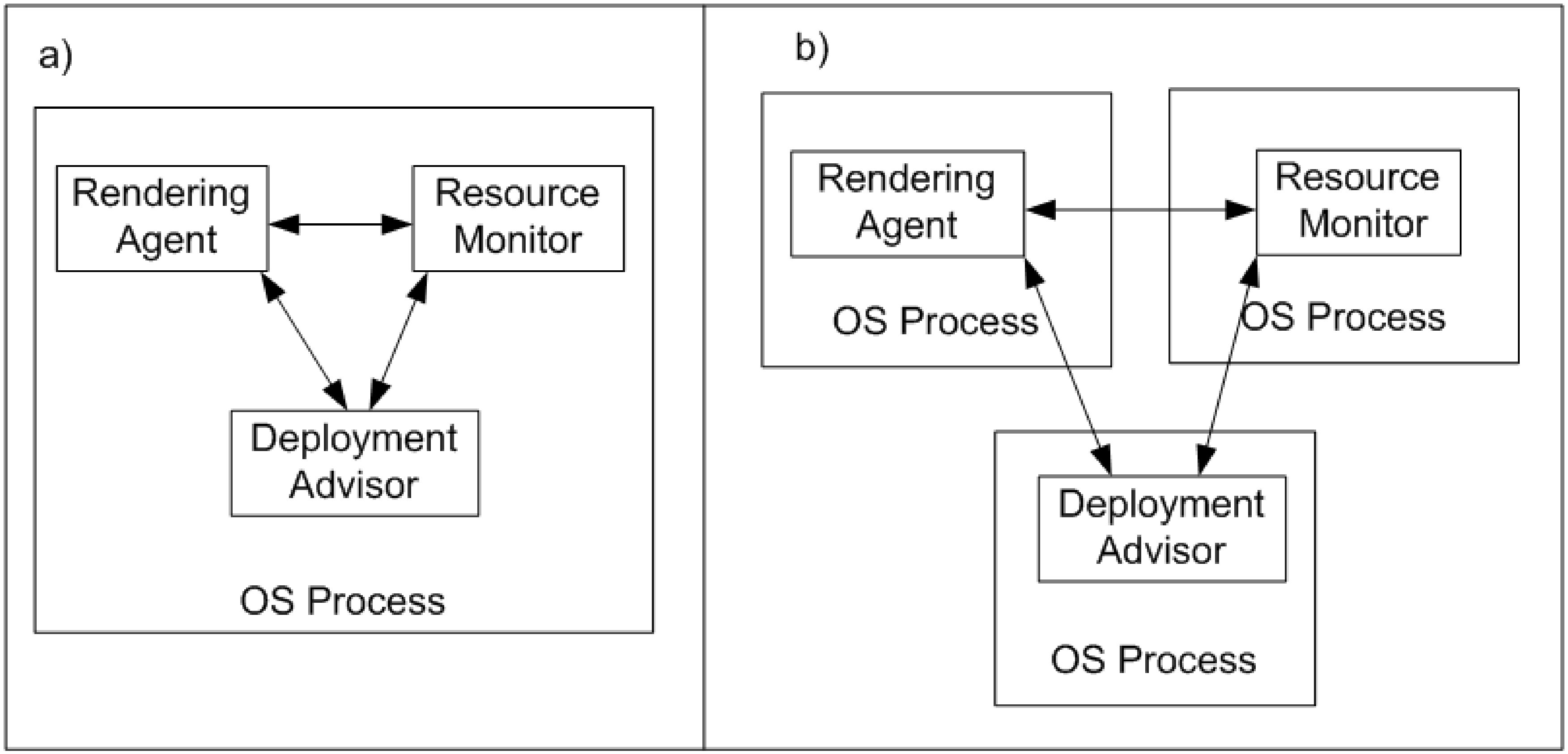
Reliability-Driven Reconfiguration Framework



Proactive Reconfiguration

- Infeasible to determine an optimally reliable architectural configuration for a mobile software system at design time
- Runtime reconfiguration may be necessary to achieve reliability requirements
  - E.g., Allocation of software components to OS processes

Allocation of Components to Processes



More Efficient
Less Reliable

Less Efficient
More Reliable

Refinement of Reliability Analysis

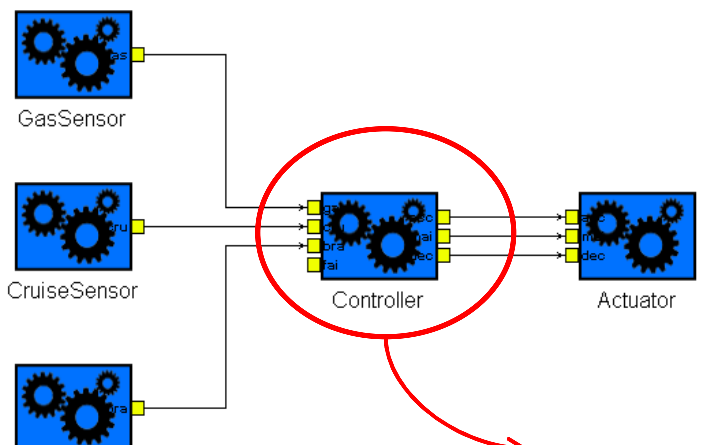
- Initial reliability *prediction* based on available sources of information at design time
- Runtime monitoring performed by the middleware is used to <u>refine</u> the initial prediction
  - internal software properties (e.g., frequency of failures, exceptions, and service requests),
  - external properties (e.g., network fluctuations, battery charge),
  - changes in the structure of the software (e.g., disconnection of components due to network drop outs, off-loading of components due to drained battery)
- Complementary sources of information

Reliability Analysis

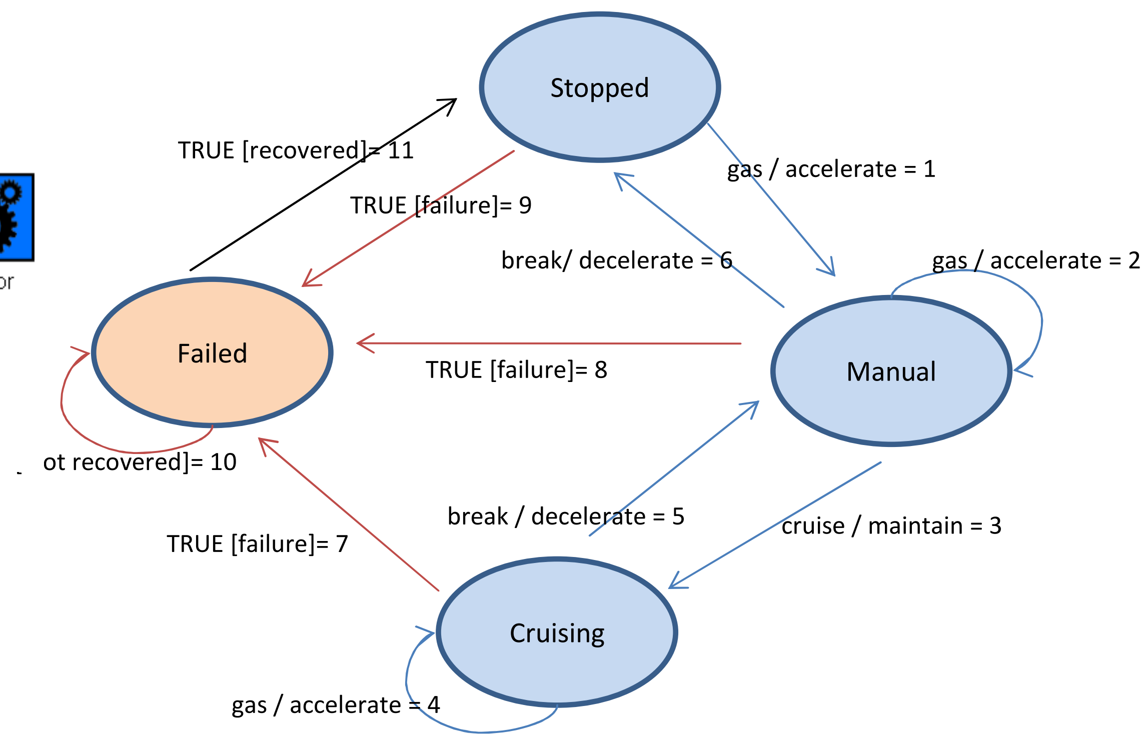
- Calculate Component reliability
  - Build HMM based reliability model using
    - Component's behavioral model
    - Training data from the running system
- Derive System reliability
  - Build HMM based reliability model using
    - System's structural model
    - Component level reliability

Calculating Component reliability

- Build HMM based reliability model
- set of states $S = \{S_1, S_2, \ldots S_N\}$, a transition probability matrix $A = \{a_{ij}\}$
- set of observations $O = \{O_1, O_2, \ldots O_M\}$, an observation probability matrix $E = \{e_{ik}\}$



E.g.: Analyze behavioral model of controller



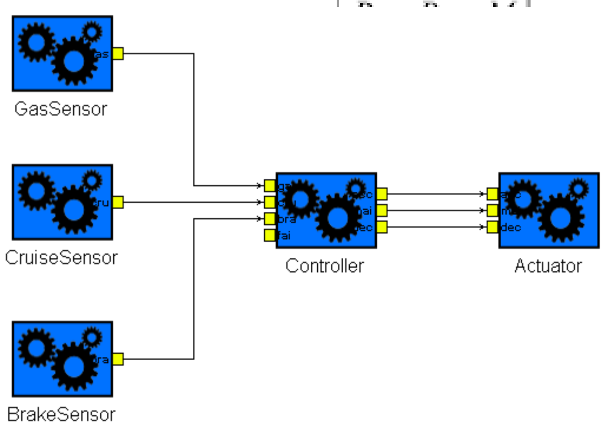
Calculating System Reliability

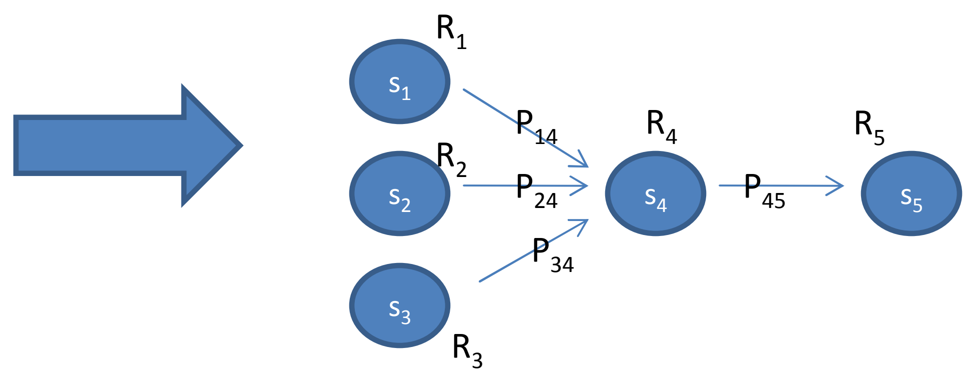
- Build Discrete Markov Chain based reliability model
- *S* is successful output state, *F* is failure state. $D_1 = [1]$, $D_2 = [1]$
- The inner matrix M is a k * k matrix with only transient states, in which s_1 is the entry state and s_k is the exit state (where k is the number of states)
- R_k is the probability of successful execution of state k

$$T = \begin{array}{c} \\ S \\ F \end{array} \begin{array}{c} \begin{array}{ccc} S & F & s_1 \cdots s_k \end{array} \\ \begin{bmatrix} D_1 & 0 & 0 \\ 0 & D_2 & 0 \end{bmatrix} \end{array}, \quad B_1 = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ R_k^+ \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 - \sum_{j=1}^{k} M(1,j) \\ \vdots \\ 1 - \sum_{j=1}^{k} M(k-1,j) \\ 1 - R_k^+ \end{bmatrix}$$

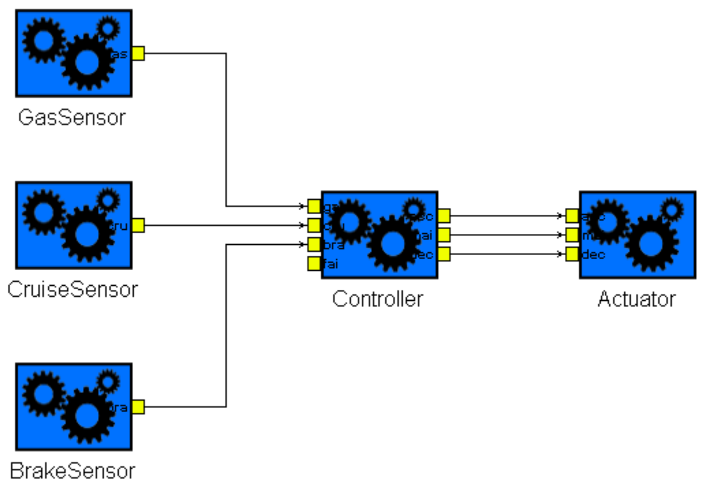
$$\text{reliability} \quad R = (-1)^{k+1} R_k^+ \frac{|E|}{|I - M|}$$

E is the matrix obtained by excluding the last row and first column of (*I-M*)

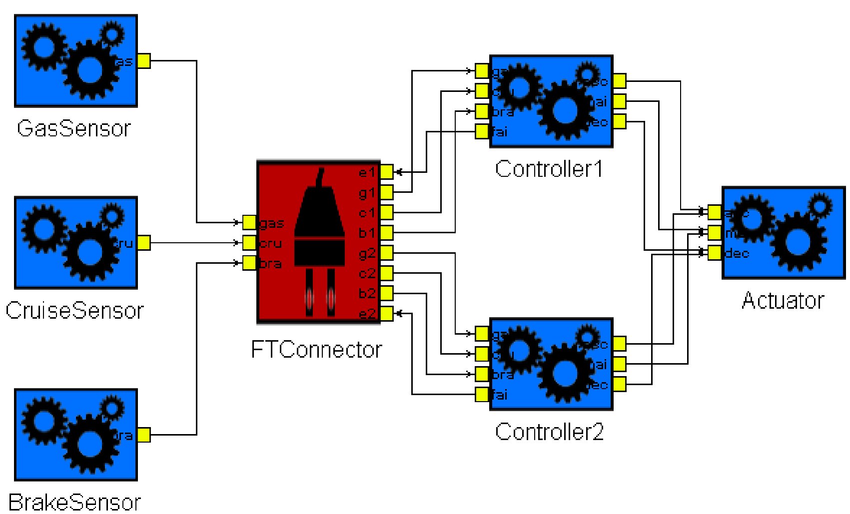




Proactive Reconfiguration

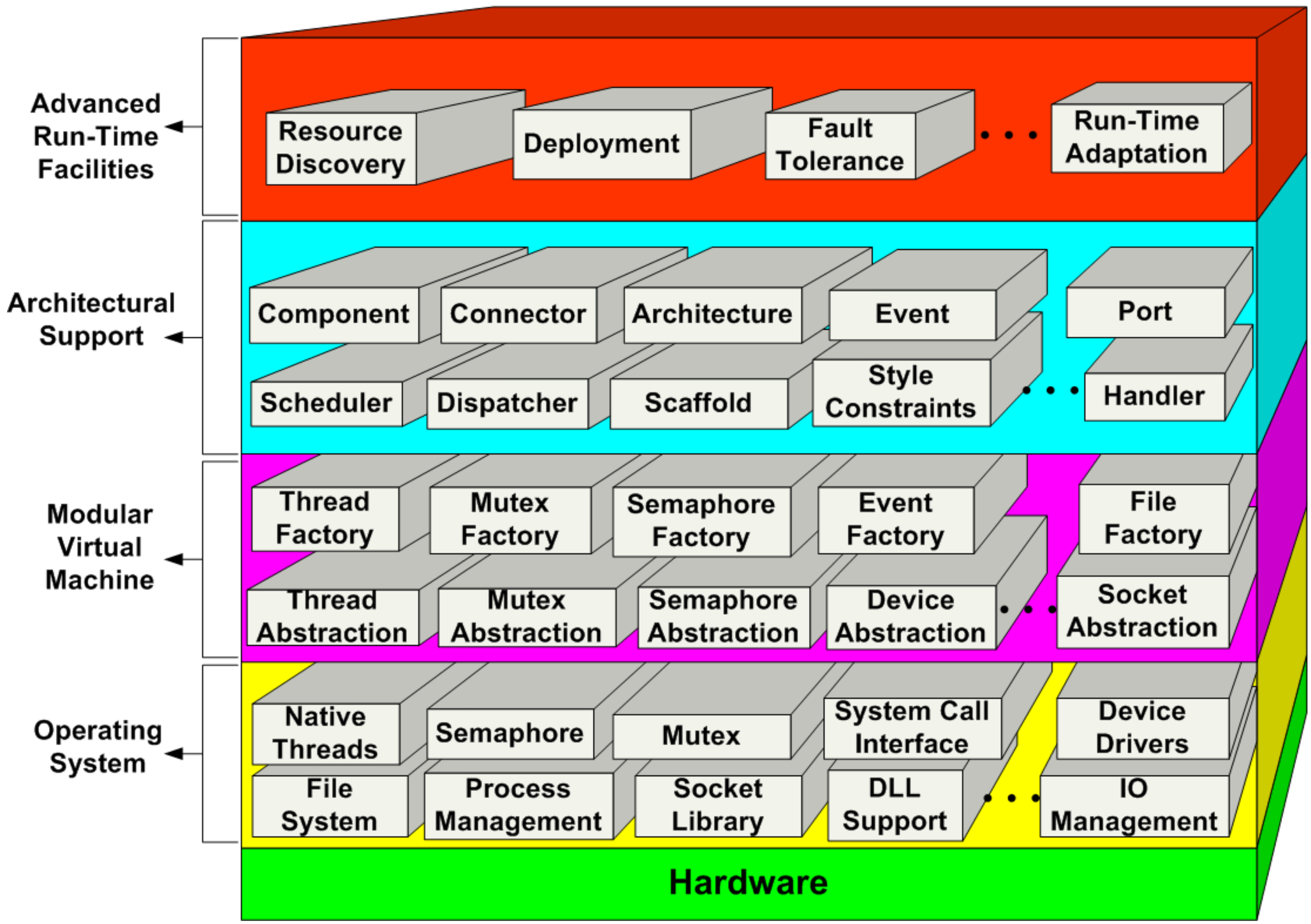


$$M(i,j) = \begin{cases} R_i P_{ij}, & \text{state } s_i \text{ reaches state } s_j \text{ and } i \neq k, \\ 0, & \text{otherwise}, \end{cases} \quad \text{for } 1 \leqslant i,j \leqslant k$$

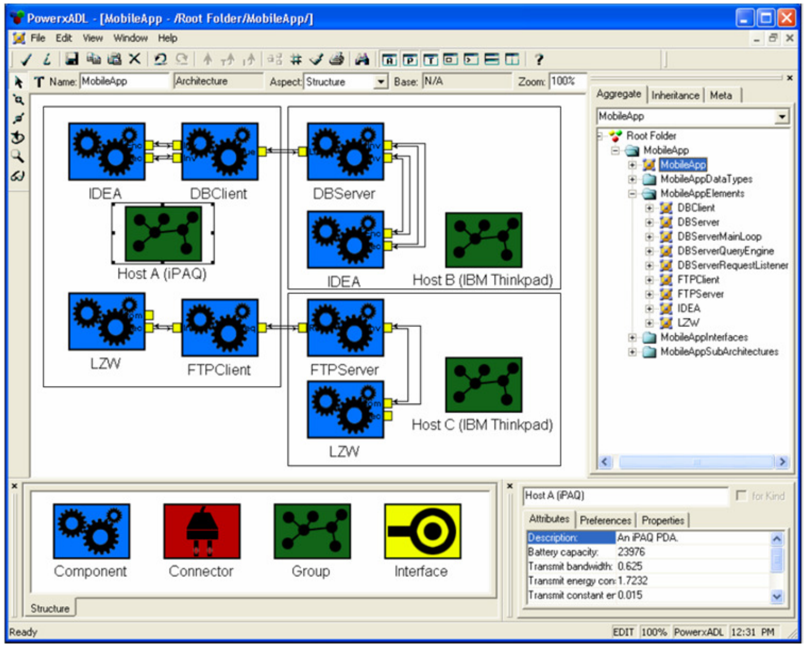


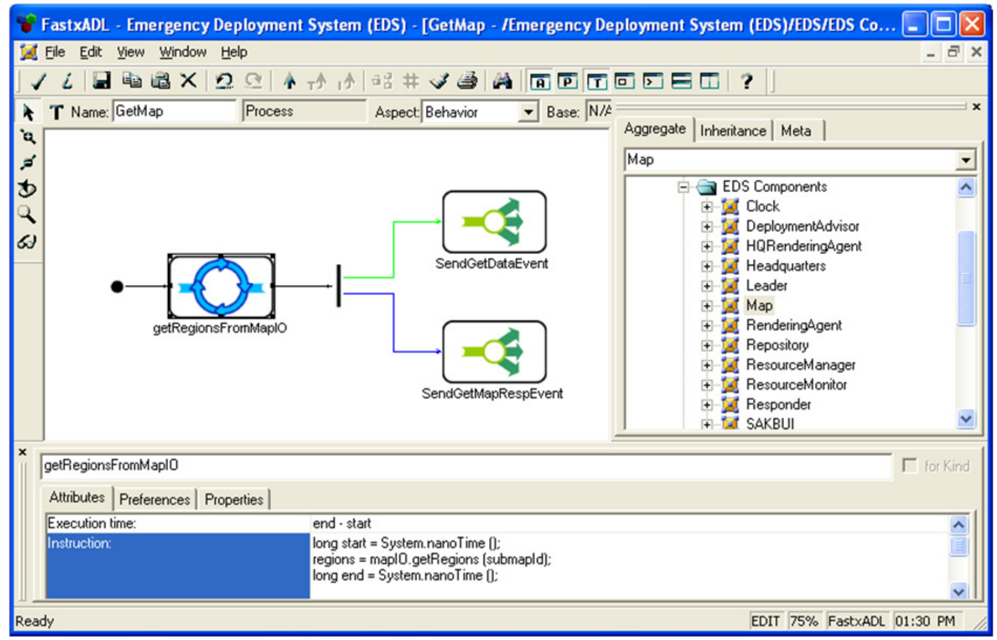
$$R_i = \begin{cases} r_\alpha, & \text{only } c_\alpha \text{ in } s_i, \\ 1 - \left(\prod_\alpha (1 - r_\alpha) \right), & \forall c_\alpha \text{ in } s_i, \end{cases} \quad 1 \leqslant i \leqslant m$$

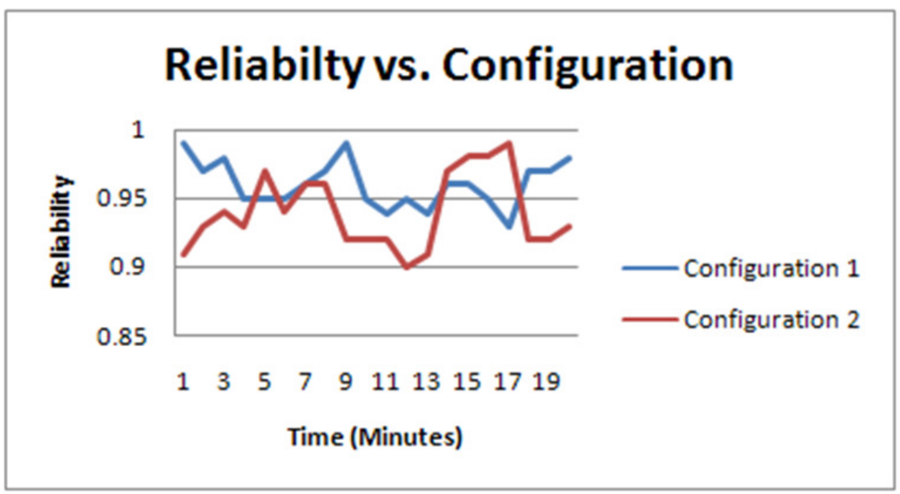
Prism-MW: Architectural Middleware for Mobile Systems



XTEAM: Modeling and Analysis Tool







Conclusion and Future Work

- **Problem:** architecture-based reliability analysis for mobile and adaptive software systems
- **Approach:** assess and improve the reliability of mobile and dynamic software systems through dynamic reconfiguration
  - Initial framework development, and preliminary evaluation [completed]
  - Incorporation of contextual information into reliability analysis, and evaluation of mobile software systems [TBD]