

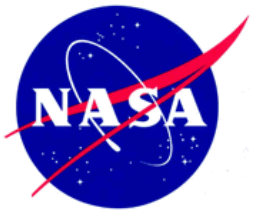

Contents

Experimenting with Architecture Evaluation to Improve Software Dependability

Sima Asgari, Victor Basili, Patricia Costa, Paolo Donzelli, Lorin Hochstein, Mikael Lindvall, Ioana Rus, Forrest Shull, Roseanne Tvedt, Marvin Zelkowitz


Fraunhofer USA
Center for Experimental Software Engineering Maryland

ICSE 2004 Workshop on Architecting Dependable Systems



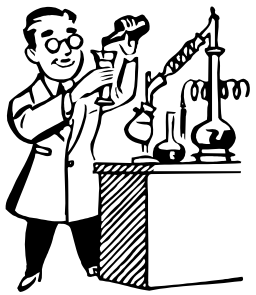
High Dependability Computing Project

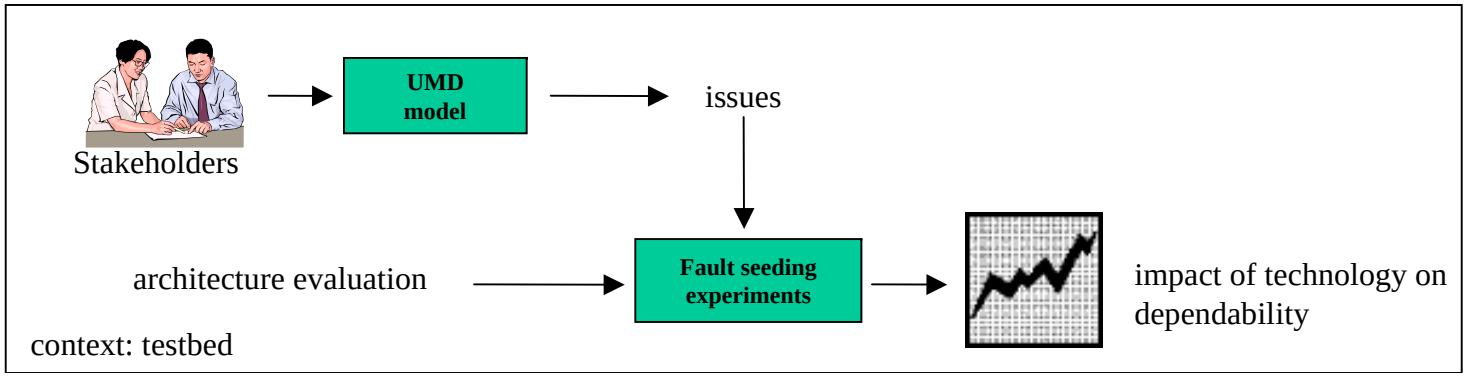
- Improve NASA's ability to build dependable software
- Investigate, foster, and transfer to practice new technologies developed by researchers at multiple universities
- Use testbeds for technology assessment: scaled-down versions of systems in domains which require high dependability
  - Autonomous rovers
  - Air traffic control systems
- Work in progress

Rocky 7 rover

Fraunhofer & UMD's Role in HDCP

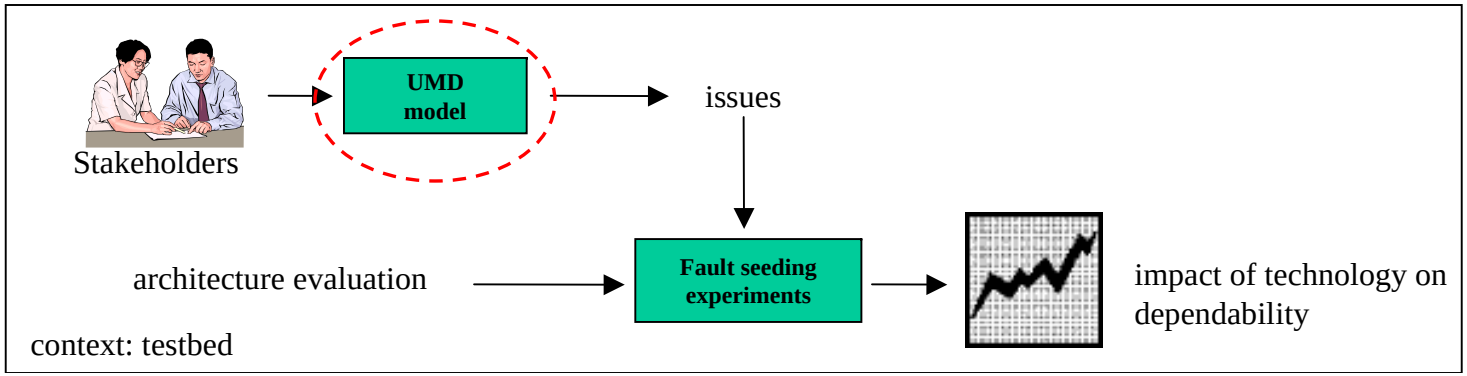
- Goal: Evaluate value of technology from stakeholder's perspective (e.g. system users, system developers)
- Empirical approach
  - Use experiments to understand & improve technologies
- Current work
  - Unified Model of Dependability (UMD)
  - Dependability-driven fault seeding
  - Architecture evaluation technique





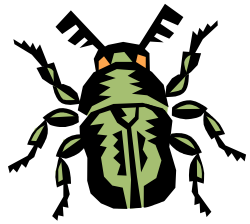
Unified Model of Dependability (UMD)

- Goal of UMD
  - Provide common language for discussing dependability
  - Help stakeholders define dependability needs as measurable system properties
- Stakeholders define undesirable **issues** that may affect the system, together with possible triggering external **events**
- Examples of issues: failures, hazards

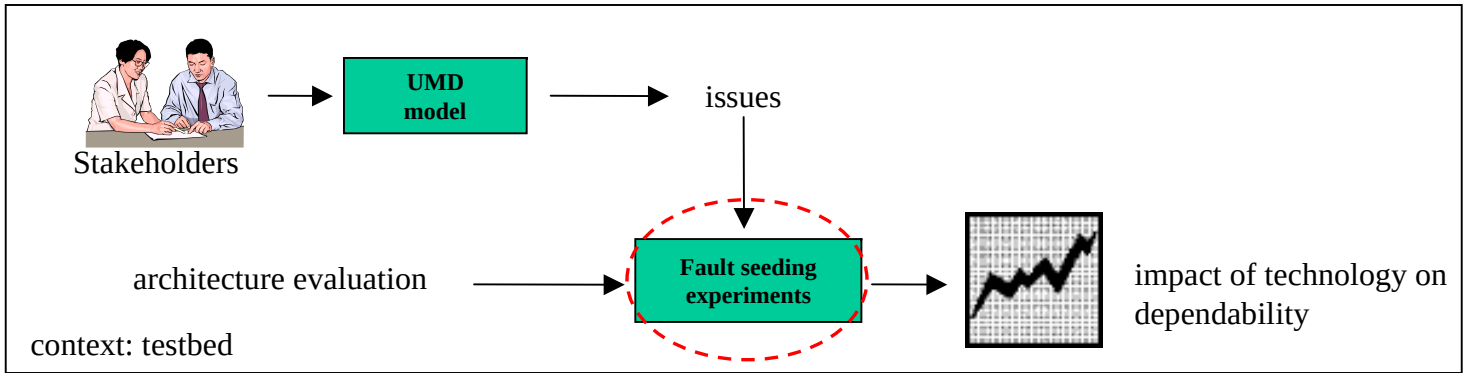


Dependability-driven Fault Seeding

- Overview
  - Seed faults in testbed artifacts
  - Apply technology to artifact
  - Identify impact of technology on faults, failures

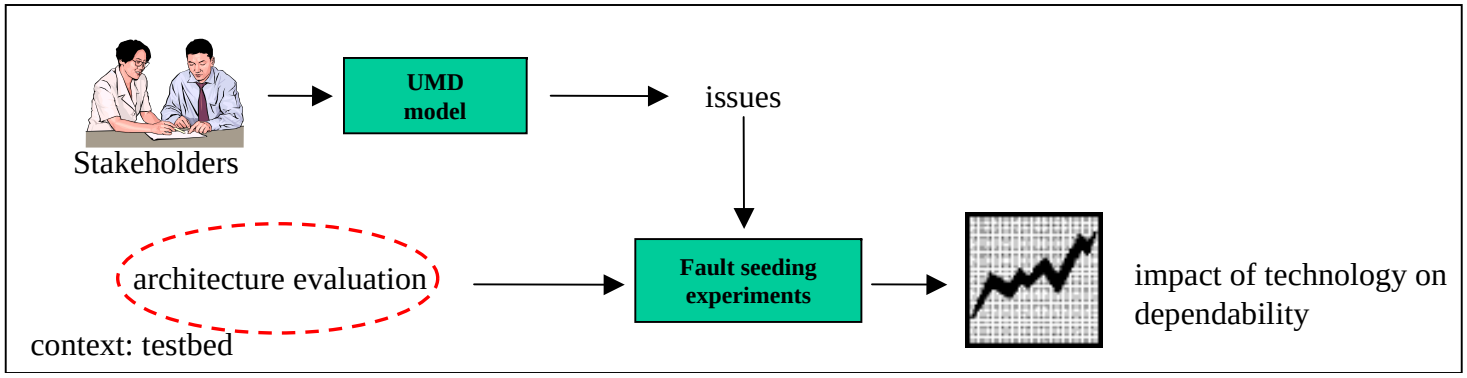


- Evaluate claims of researcher
  - Technology-specific faults
    - → Based on technology claims, seed types of faults that technology should be able to detect
- Evaluate impact of technology on **dependability** of testbed
  - Failure-related faults
    - → Identify failures of interest (output of UMD model)
    - → Hazard analysis to identify faults that could lead to those failures

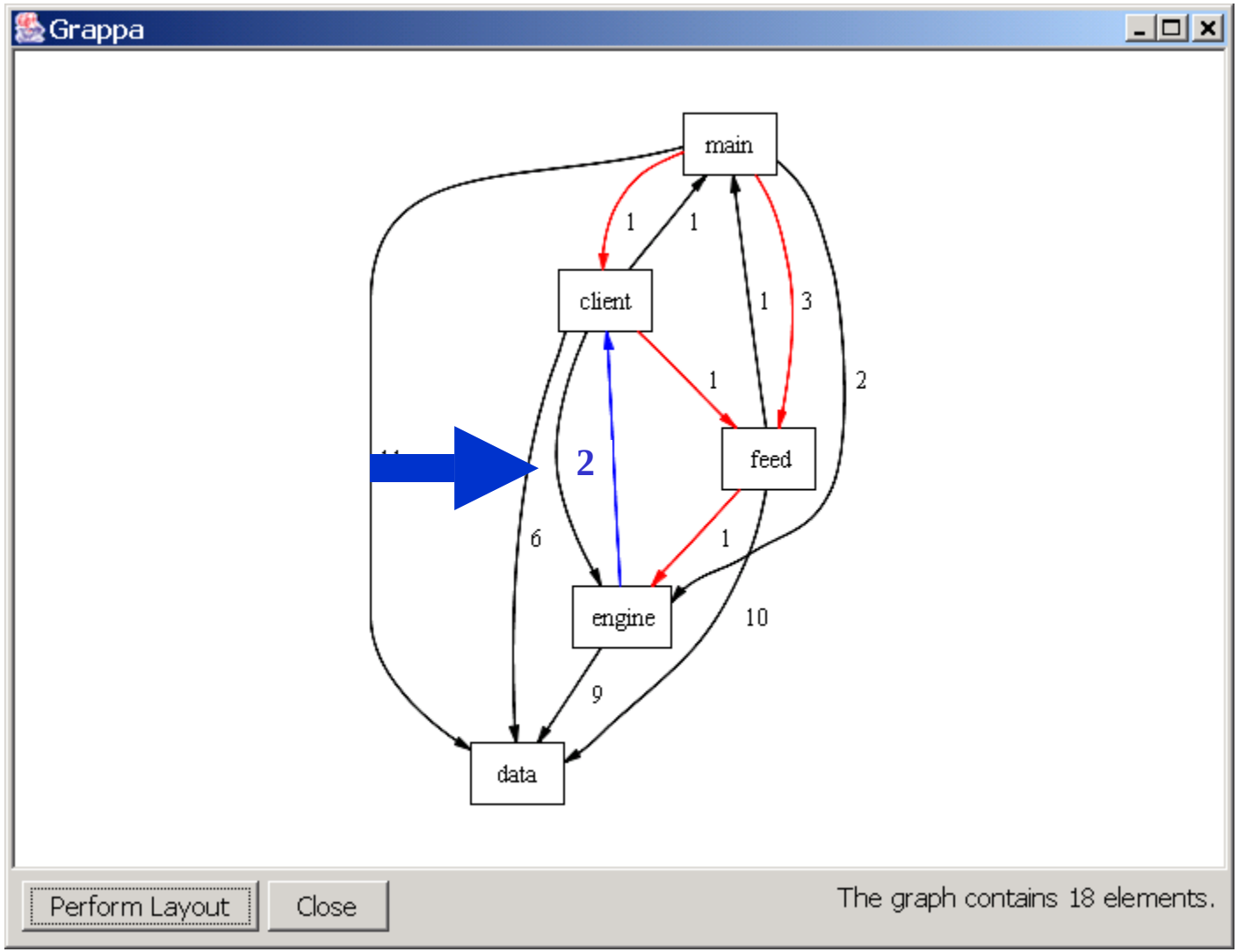


Sample Technology: Architectural Evaluation

- Architectural evaluation
  - Detect architectural violations and non-conformances between planned and actual architecture
- Has been used on several systems with encouraging results.
  - Architectural violations are common
  - Cause software to decay, hard to maintain
  - AE can quickly detect violations (keep architecture on track)

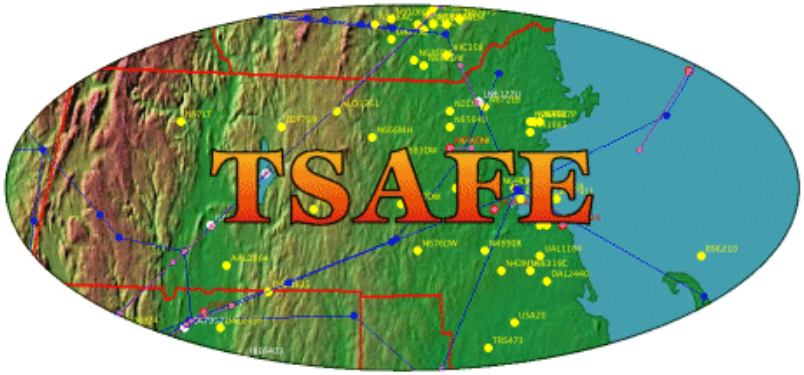


Example – Planned/Actual Comparison

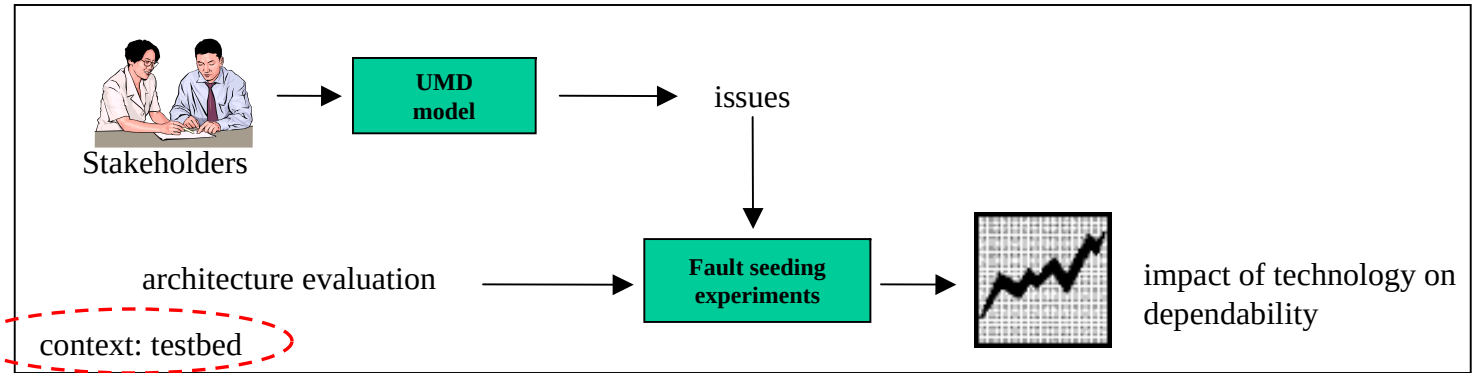


A link between client and engine is missing.

Testbed: TSAFE
Tactical Separation Assisted Flight Environment

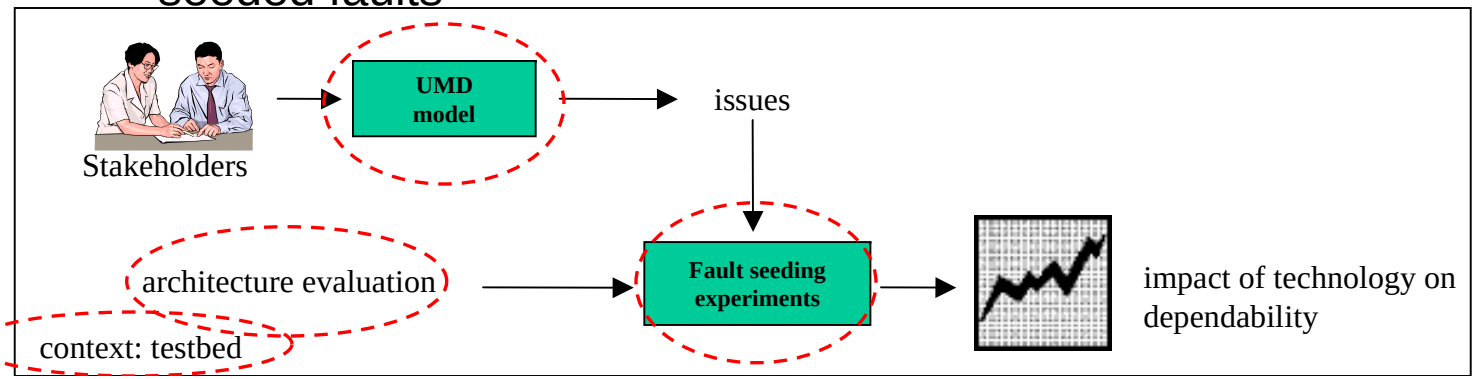


- Aids air-traffic controllers in detecting short-term aircraft conflicts
- Proposed as principle component of larger Automated Airspace Computing System
- MIT TSAFE testbed; a partial implementation
  - Determines if plane is conforming to flight plan
  - ~ 20 KLOC of Java



Using TSAFE as a Testbed

- Applying UMD model to TSAFE
  - Determine feasibility of approach
  - Evaluate usefulness of web-based support tool
  - Identify failures of interest
- Applying fault seeding to TSAFE
  - Using failures from UMD model, identify possible source code faults which would cause those failures
  - Identify possible architecture-related faults that architecture evaluation method should catch
- Applying architectural evaluation to TSAFE
  - Use independent expert to apply technique on TSAFE with seeded faults



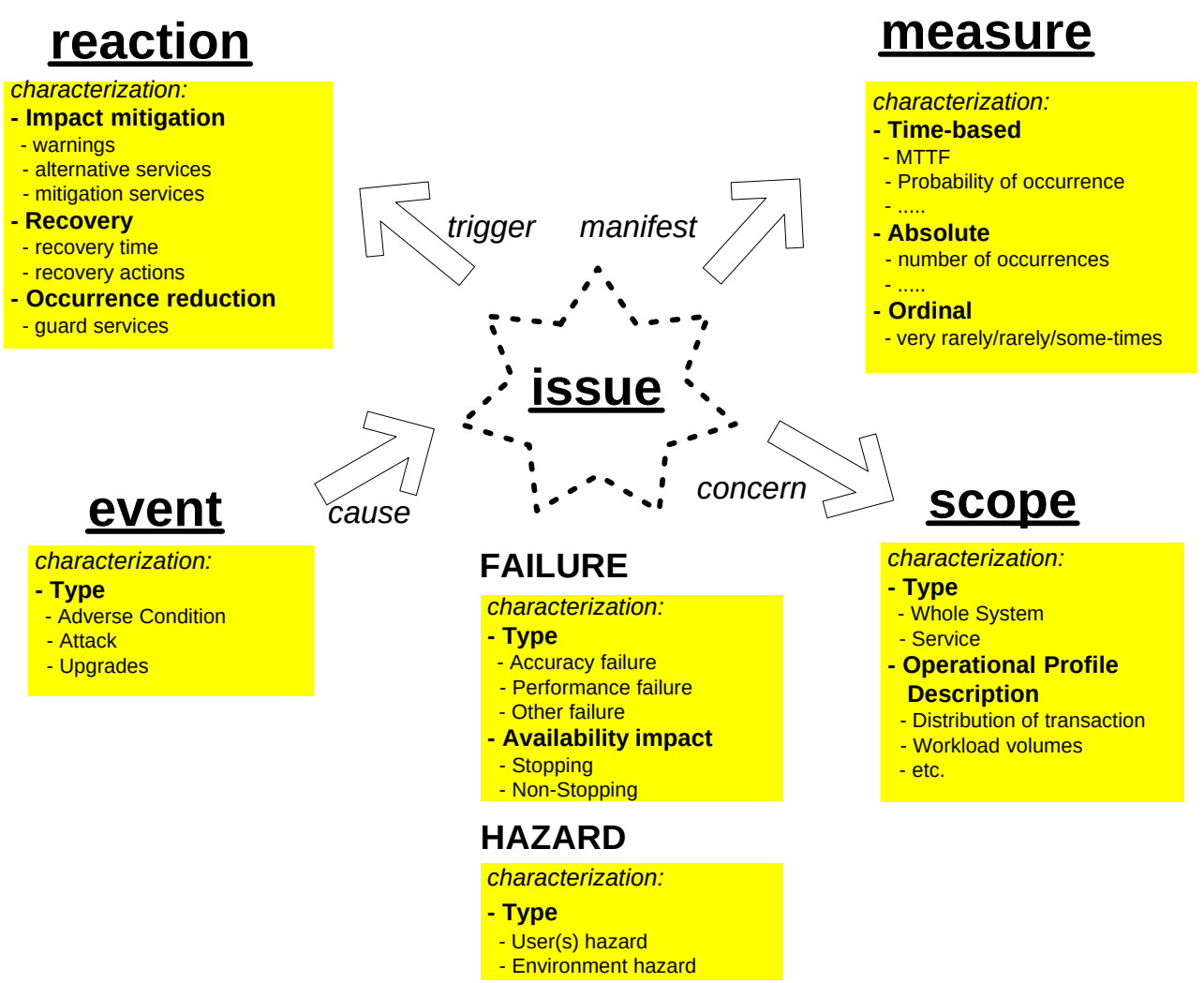

Current Status

- Current status
  - Expert is applying architecture evaluation to fault-seeded version of TSAFE
  - Building a database of potential TSAFE faults for use in future experiments
  - Evolving UMD model
- Future work
  - Analyze results of architecture evaluation experiment
  - Evaluate other technologies using TSAFE (e.g. code inspections)
  - Perform experiment using other HDCP testbeds (e.g. USC Full-Service Robot)

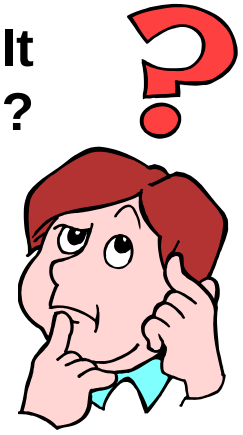
Contents

Thank you

Unified Model of Dependability (UMD)



What Is Dependability and How Is It Defined?



University of Maryland Dependability Framework (UMD framework) defines system dependability based on Issues, e.g. assume Issues = Failures then

Less Failures ⇦ More Dependable