Dependability and Architecture: An HDCP Perspective

Bill Scherlis
Carnegie Mellon University
ICSE Workshop on Architecting Dependable Systems
May 2002
scherlis@cmu.edu

Dependability and Architecture

- **Dependability**
  - ***Reliance that can justifiably be placed…***
  - Fault tolerance
  - API robustness
  - Code safety
  - Safe concurrency
  - Usability
  - Availability
  - Self-healing
  - *Etc.*

- **Architecture**
  - ***Structural constraint***
  - ***That which changes most slowly***
  - Dynamic monitoring
  - Robust APIs and exception mgt
  - Self-healing
  - Framework compliance eval'n
  - Managed adaptation

- **Generally Accepted Linking Principle**
  "Dependability designed in from the start"

Observation

- **Similar arguments for *from-the-start* are made for multiple dependability attributes**
  - Availability
  - Self-healing
  - Usability
  - Security

Questions

- **What are the concrete research steps?**
  - Beyond articulating precept on the basis of intuition and experience…
  - *What does it mean to "design in" dependability*?
- **What are the dependability measurables?**
  - For the various attributes
  - *How do we know if we are succeeding*?
- **What can be assured?**
  - On the basis of architectural commitment?
  - *What commitments can we make*?
- **How to reason about (trust) the add'l structure?**
  - Wrappers
  - Self-healing monitor/detect/log/mitigate
  - FT availability architecture

Exploring the Questions

The HDCP programmatic approach

- **Testbeds**
  - Experimentation at scale
  - Intervention
  - Measurement
  - Assurance

- **Scalable techniques**
  - Frameworks
  - Composable attributes and analyses
  - Horizontal approaches

Keep in Mind

- **Not much impact of 30-40 years of research in software dependability, broadly construed**
  - Some notable exceptions
    - Some critical systems
    - Fully embedded practices
      - Programming language types
    - Certain analyses
    - Conventional architectural practices
- **Measurement?**

The HDCP Approach

- **Focus**
  - Dependability at scale
  - Dependability and integration
  - Data, measurement, evaluation

- **Large-scale testbed projects**
  - Identify actual challenges in NASA mission projects
  - Undertake experimental interventions
    - Measurement, improvement, assurance
    - Multiple interventions: risk mgt for stakeholders
  - NASA stakeholders directly involved
  - Distance collaboration support

- **Diverse team**
  - CMU with USC, UMd, MIT, U Wash, U Wisc
  - Moffett campus

The HDCP Approach

- **Research areas**
  - Measurement and dependability (Boehm, Basili, Zelkowitz)
  - Analysis and assurance (Jackson, Koopman, Notkin, Scherlis)
    - Checking specifications
    - Concurrency and Java
    - Testing strategies
    - Robustness
  - Technological intervention (Garlan, Lee, Narasimhan, Reid, Shaw)
    - Self-healing architecture
    - Proof carrying code and mobility
    - Fault tolerance architecture
    - Secure dependable networking
    - Coalitions and anomaly detection
  - Usability and dependability (John, Bass)
    - Architecture and usability

HDCP Status

- **Scale of effort**
  - 5 years
  - 12 Lead investigators at 6 universities
  - Engineering team and collaboration infrastructure
- **Status**
  - Testbed proposals submitted by NASA organizations
  - Testbed selection decision to be announced shortly
- **Related effort**
  - NSF / NASA solicitation

Dependability in the mainstream?

- **Practices for critical apps**
  - Costly (orders of magnitude)
  - Significant sacrifices in capability and flexibility
  - Highly conservative (e.g., deterministic) architectures
  - Standards: rigor on surrogates (process, organization, *etc*.)

- **No trickle-down to mainstream**

Sustainability
- **Engineered-in** dependability
- **Evidenced** through measurement and assurance
- **Supported** by market and economic factors
- **Reachable** from the present environment

Dependability in the mainstream?

Sustainability
- **Engineered-in** dependability
- **Evidenced** through measurement and assurance
- **Supported** thru market and economic factors
- **Reachable** from the present environment

- **Elements**
  - Understand risk management challenges of users
  - Stakeholders: Users, Insurers, Auditors, Integrators, Vendors
  - Expertise: Technology, Economics, Markets, Law, Policy
    - Multi-university collaboration
- **Approach**
  - Sustainable Computing Consortium (SCC)
  - Build on HDCP, SWIC, and other efforts
  - Collaborate with open source and other engineering communities
- **Goal**
  - Engineering and market culture of dependability

Promising directions
(examples)

- **Architecture-level intervention**
  - Self-healing architecture
  - Transparent intervention
    - Application-transparent FT (CORBA, etc.)
    - Dynamic monitoring/logging
  - Structural transformation
    - Wrapping
  - Framework analysis
  - Mobile code architectures
- **Lightweight formal methods**
  - Model checking of specs
  - First-class encapsulation and types
  - “Narrow-band” assurance techniques
- **Usability-informed architecture design**
  - Robustness for person-in-the-loop processes
- **Program analysis**
  - API client compliance evaluation (protocol, threading, etc)
  - Buffer overflow detection, etc.
  - Annotation
  - Safe concurrency
- **Advanced testing**
  - Robustness and APIs (Windows, Linux)
- **Correlative measurement techniques**
  - CoQualMo, SecurityMM, ITsqc

Promising problems

- **Analysis and assurance for self-healing systems**
- **Policy and assurance for self-organizing systems**
- **Evaluation of dependability attributes for conventional architectures**
  - The “standard” configuration for high availability data centers
- **Architecture-level specification**
- **Formal linking of architecture specifications and low-level design / code**