A Reasoning Framework for the Architectural Design of Real-Time Fault-Tolerant Systems

Siemens Overview

- >400,000 employees worldwide
- Builds mission/safety critical software intensive systems
  - Power plant control systems
  - Power distribution systems
  - Telecommunications systems
  - Building control systems
  - Automotive systems
  - …
- Largest developer of software in the world

Large Scale Distributed Embedded Systems

- Huge capitol investment
- Long life expectancy
  - Often > 15 years
- Mission/safety critical
  - E.g. Power plant control systems
    - Downtime is very expensive

Reliability Impacts Revenue Directly

- Business model may rely on correct operations
  - Systems may be sold based on expected savings for consumer
    - Siemens gets a % of consumer's savings
  - Speaks to the degree of confidence in the system
    - Non-optimal execution may mean losses to consumer and Siemens

Designing for Dependability

- How to ensure design meets QoS requirements?
  - Analytic models exist for performance
    - Queuing
    - Scheduling theory
  - How do we practically determine behavior when faults occur?
  - How do we understand impact of design decisions?
- Difficult to balance competing concerns
  - E.g. performance, degree of fault-tolerance, resource utilization, …

Looking for Improvements

- Currently we rely on intuition of experienced architects
- Looking for practical structured methods for understanding connection between design decisions and requirements
  - SEI approach shows promise, but not yet suitable for QoS issues
  - Also looking into more “tolerant” architectures