Monitor Synthesis:
for software health management

Lee Pike leepike@galois.com
Alwyn Goodloe alwyn.goodloe@nianet.org
César Muñoz munoz@nianet.org
Where Are We?
Who Are We?

• Galois, Inc.
  - Galois’ mission is to create trustworthiness in critical systems. We’re in the business of taking blue-sky ideas and turning them into real-world technology solutions.
  - About 40 employees, including experts in functional programming, formal methods, and security.

• National Institute of Aerospace (NIA)
  - NIA is a non-profit research and graduate education institute created to conduct leading-edge aerospace and atmospheric research and develop new technologies for the nation.
  - Includes the NIA Formal Methods Group, working on critical systems of interest to NASA.
Project Staff

- Lee Pike, Galois (PI)
- César Muñoz, NIA (Co-PI)
- Alwyn Goodloe, NIA (Research Scientist)

Consultants:
- Joe Hurd, Galois
- John Matthews, Galois
Software Health Management

• What is **software health** for embedded control systems?
  - Functional correctness
  - Timing properties
  - Safety properties (capturing fault-tolerance)
  Under the environmental assumptions.

• Problem:
  - **testing** cannot ensure the absence of errors in ultra-reliable systems,
  - and **formal proof** does not yet scale.

• So “who watches the watchmen?”
  ...

© 2008, Galois Inc.
Software Monitoring

- Simplicity is the unavoidable price which we must pay for reliability. —C.A.R. Hoare

- Simple monitors analyze executions at runtime for software health.

- Monitors raise alarms or attempt to reset the system (into a known safe state).

- Research question: can software monitoring form a basis of software health management?
Research Contributions to IVHM

• Our research hypothesis: we can synthesize software monitors for *ultra-reliable systems* that are distributed, fault-tolerant, hard real-time.

• Our research challenges:
  - Distributed systems may require *distributed monitoring* (diagnosis without global information).
  - Monitors should not jeopardize hard *real-time requirements* of the monitored systems.
  - Monitors *themselves* need to be *reliable*, perhaps requiring *fault-tolerance*.
  - Formally synthesizing these monitors from requirements.
Key Research Contributions

• **Approach:**
  - Formal synthesis of fault-tolerant monitors from system specifications.

• **Systems characterization:**
  - Hard real-time
  - Fault-tolerant
  - “Small graphs”
  - “Fixed topology”

• **Properties to monitor:**
  - Validity
  - Agreement
  - Timing constraints
Proposed Monitoring Case Studies

• NASA’s SPIDER (Scalable Process-Independent Design for Enhanced Reliability)
  - Formally specified and verified fault-tolerant protocols.

• TTech’s TTEthernet
  - Allows hard real-time communication and services over ethernet.
  - Formally specified properties.
Proposed Plan of Work

• Year 1
  - Survey state-of-the-art approaches to software health management.
  - Research monitors for **hard real-time** temporal constraints.
  - Research **synthesis framework**.

• Year 2
  - Develop **synthesis framework**.
  - Design monitors for timing properties, agreement, and validity for our case studies.

• Year 3
  - Develop monitors for our case studies.
  - Research the synthesis of **fault-tolerant monitors**.