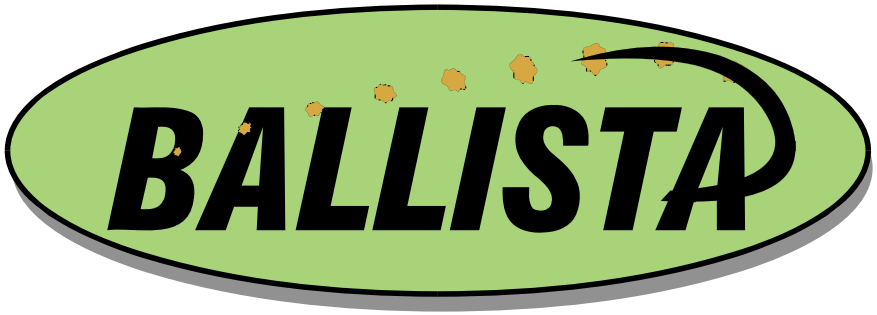
Session 1 Summary "Fault Tolerance & Autonomy"

Rapporteur

Jay Lala

Fault Tolerant Architectures For Space and Avionics: Towards More Autonomy



Dan Siewiorek

Priya Narasimhan

Carnegie Mellon University

(talk first presented in *International Forum on Integrated System Health Engineering and Management in Aerospace*, November 2005)

Electrical & Computer ENGINEERING

Fault-Tolerant Architectures for Space and Avionics -1

- Components of a spacecraft and generic fault detection and recovery techniques
- Progression of space probes and systems: from manual to autonomous over 40 yrs
- Early 60s vintage: DMSP
  - On-board ad hoc error detection & safing + ground-based diagnosis and recovery
- 90s vintage: Cassini-Huygens
  - Significant h/w redundancy with hot-backup config
  - Autonomous time-constrained onboard recovery
  - Recovery actions dependent on mission-mode

Fault-Tolerant Architectures for Space and Avionics -2

- Generic approaches to avionics fault-tolerance
- Commercial fly-by-wire flight control requirements (P(f) = 10^{-10} per hr)
- Contrast Airbus A3xx and Boeing 777 architectures to meet requirements
- TRENDS:
  - Increasing s/w size
  - Increasing redundancy, diversity, fault coverage

An Architecture for Robust and Fault Tolerant Autonomous Robots

Raja Chatila, Sara Fleury, Matthieu Gallien, Matthieu Herrb,
Felix Ingrand, Benjamin Lussier, David Powell, Fréderic Py

LAAS - CNRS
Toulouse, France

Arch for Robust & FT Autonomous Robots - 1

- Movies & animations of exemplar robots:
  - Companion, Service, and Tour Robots
- Focusing on reliability and safety aspects of dependability
- Motivated a need for an architecture
- Described the LAAS Architecture
  - Functional arch developed using GenoM which provides a software eng framework and an automaton for internal activities
  - Decision level arch (task planning) using OpenPRS and IxTeT
  - Execution Control using R2C

Arch for Robust & FT Autonomous Robots - 2

- Challenges of validating such architectures
  - Among others, hard to model environment and unforeseen evolutions
- Proposed solution: define constraining properties of modules (observable, controllable, real-time, etc)
- Examples of state checkers etc
- More movies and demonstrations of robot tests

Discussion & Issues

- Has increasing s/w complexity resulted in increased functionality/capability?
- What's a good language to program autonomous systems?
- Tension between new methods and risk averse managers.
- What are the biggest sources of undependability in autonomous robots?

DARPA UUV Program

- Circa 1986-89, DARPA funded Draper Lab to design, build & test two unmanned underwater vehicles (Office Dir: Dr. Tony Tether)
- UUV Mission & Characteristics:
  - 40’ long, 5’ diameter (yellow submarine)
  - Long duration mission: months unattended
  - Completely autonomous after launch from mother ship
  - Battery powered
  - Mission classified
- Dependability requirements:
  - Primary: UUV must not fall into enemy hands
  - Secondary: complete the mission successfully
- Vehicle swim-by-wire control, navigation, and system management
  - Triply redundant, Byzantine-resilient computer (Draper FTP)
  - No mission or systems failures during the program