P-7

FAULT-TOLERANT SOFTWARE EXPERIMENT OBJECTIVES AND STATUS

Dave E. Eckhardt, Jr.
NASA Langley Research Center

NASA
Computer Science / Data Systems
Technical Symposium

April 16-18, 1985

BACKGROUND

- Redundancy is an established principle for dealing with hardware faults.

- Software redundancy (modeled after hardware NMR and stand-by sparing techniques) has been advocated as a method to cope with residual software faults.

- Current use of redundant software includes:
  - Airbus Industries A310 slat and flap control
  - Swedish state railroads traffic control system
  - Boeing 737-300 critical flight control functions
  - Boeing 757 yaw damper and stabilizer trim systems
  - Atomic Energy of Canada nuclear reactor shutdown system
  - NASA Space Shuttle mission critical functions

OBSERVATIONS*

- No examples found where last phase of fault–tolerance, fault treatment, was an integral part of system.

- Trend in which regulatory agencies recommending redundant software to improve reliability.

- Empirical data needed to quantitatively evaluate performance has not been kept. Statements of improved reliability and cost trade–offs could not be made.

- No evidence that software redundancy degraded reliability.

- Instances of faults detected during code testing and faults in specifications as a result of redundant software.

*Study of Fault–tolerant software technology
NASA Contractor Report 172385

MAJOR ISSUES

RELIABILITY

- Little empirical data to assess reliability
- Independence assumption of current reliability models is questionable
- Prevalance of coincident errors in "independent" versions is unknown
- Cost of reliability gains is unknown

RELATION TO HARDWARE FAULT–TOLERANCE

- Layered or integrated hardware/software fault–tolerance
- Required hardware, operating system support for effective implementations

APPLICABILITY

- No examples of fault–tolerant operating systems
- Closed–loop systems present stability problems when certain fault–tolerant techniques introduced

EXPERIMENT OBJECTIVES

- Develop analytical methods to evaluate the general strategy of software redundancy to improve reliability.

- Gather empirical data to characterize fault distributions of coincident errors.

  - realistic application 3–5 mm effort
  - well tested, reliable software
  - software engineering practices
  - quality programmers

- Characterize coincident fault types and suggest methods to reduce their intensity coefficient.
  (speculate these are the residual faults, thus this effort should benefit software engineering in general)

CURRENT STATUS

- Grants to participating universities in place (UVA, UCLA, U Illinois, NC State)

- Hiring of programmers nearing completion

- Application selected

- Probabilistic framework for analysing multi-version software in the presense of coincident errors developed

- Definition of experiment protocol and development environment nearing completion

FUTURE ACTIVITIES

- Software specifications (Apr 10 - June 1)
- Software development (June 1 - Aug 15)
- Extensive life testing (Aug 15 ...)
- Reliability analysis