INTEGRATED SOFTWARE HEALTH MANAGEMENT FOR AIRCRAFT GN&C
Johann Schumann, Ole Mengshoel

Abstract
Modern aircraft rely heavily on dependable operation of many safety-critical software components. Despite careful design, verification and validation (V&V), on-board software can fail with disastrous consequences if it encounters problematic software/hardware interaction or must operate in an unexpected environment.

We are using a Bayesian approach to monitor the software and its behavior during operation and provide up-to-date information about the health of the software and its components. The powerful reasoning mechanism provided by our model-based Bayesian approach makes reliable diagnosis of the root causes possible and minimizes the number of false alarms. Compilation of the Bayesian model into compact arithmetic circuits makes SWHM feasible even on platforms with limited CPU power.

We show initial results of SWHM on a small simulator of an embedded aircraft software system, where faults can be injected.

Principle architecture for ISWHM

Software Can Fail
Despite careful SW development and V&V, safety-critical SW can fail.

F-22 Raptors crossing the date-line: SW bug caused loss of navigation and communication

Harrier Autolander: buggy radar-altimeter integration caused near-crash during landing (NASA)

SPIRIT: overfull on-board file system caused reboot-loop after landing

Ariane-V: SW reused from Ariane IV caused overflow and destruction of rocket

A Bayesian ISWHM
We are using Bayesian networks (BN) to construct a model of the software and its behavior in nominal and failure cases. BNs can be used to
• detect failure(s), and to
• perform detailed reasoning on the root cause of the problem

Example: low oil pressure and vibration indicates a likely problem with a bearing.

Modeling for ISWHM
ISWHM models are constructed from (software) sensor nodes, unobservable status nodes, and health nodes. Low posteriors of health nodes indicate problems and poor SW health.

Software Health Management monitors the system and software during operation to
• reliably detect faults
• diagnose most likely root cause(s) while minimizing the number of false alarms and missed adverse events

References

1 SGT, INC, NASA Ames 2 CMU, NASA Ames with contributions by Timmy Mbaya, UMass, Boston/USRP

NASA Aviation Safety Annual Technical Meeting, St. Louis, MO May 10 – 12, 2011