

Software Health Management: A Short Review of Challenges and Existing Techniques

Knot Pipatsrisawat, UCLA

Adnan Darwiche, UCLA

Ole J. Mengshoel, CMU, NASA ARC

Johann Schumann, RIACS, NASA ARC

Overview

- **What is Software Health Management?**
- Challenges
- Structuring of the field and related work
- What is missing?

Software Health Management



health management for software

Search

[Advanced Search](#)
[Preferences](#)

Web [Show options...](#)

[Medical Office Software](#)

www.athenahealth.com Physicians: Get Web-Based **Software**, A Rules Database & More - Find Info

[EMR](#)

www.VisionaryMed.com Complete EMR **software**. E&M coding, differential diagnosis and more.

[Practice Management](#)

www.LeonardoMD.com Web-based, Multiple-locations, Messaging, Resources, \$150/mo.

[Healthcare Management Software Providers | Business.com](#)

Browse listings to find **healthcare management software** and **software** for **healthcare** administration. Research **software** requirements for **health management** ...

www.business.com/directory/health_care/health.../software/ - [Cached](#) - [Similar](#)

[Healthcare Management Software: hospital software administration ...](#)

Healthcare Management software. Free, interactive directory to quickly narrow your choices and contact multiple vendors.

www.capterra.com/healthcare-management-software - [Cached](#) - [Similar](#)

Software Health Management

The image shows a screenshot of the Windows XP Activation form. The window title is "Activation form". The Microsoft logo and "Windows XP" are at the top left. The main heading is "Activation of Windows." Below it, it says "Just 3 steps and you're done...".

Step 1: Select your location... (dropdown menu)

Step 2: Enter your contact information
Email [text box] Phone number [text box]

Step 3: Enter your billing information

Important: your card will NOT be charged.

Expiry date: Select Month [dropdown] Year [dropdown]

CVV2 code [text box]

To aid in the prevention of fraudulent credit card use, we now require the 3 or 4 digit code on the back of your credit card.

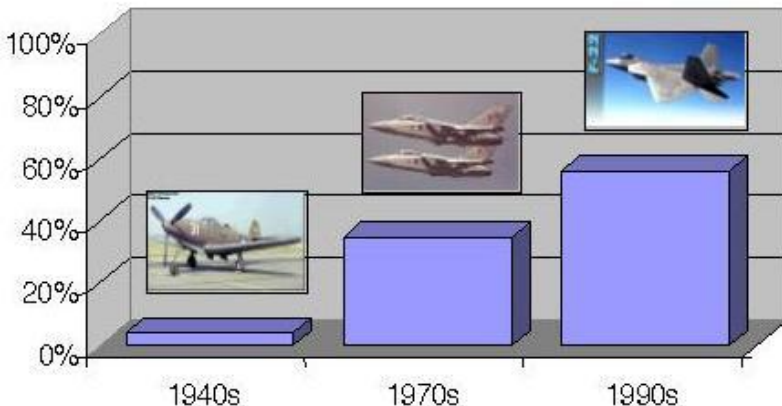
Fields for Name on card, Credit card number, and ATM PIN are also present. A "Back" button is on the bottom left and a "Next" button is on the bottom right.

Red circles highlight the "Credit card number" and "CVV2 code" fields.

Vehicle Health Management



- Modern aircraft, cars, etc.
 - have IVHM for major subsystems (engines, hydraulic, power, ...)
 - Important for safety, reliability, environmental impact, economical considerations
 - Rely heavily on Software
 - SW is getting more and more important



Why no Software Health Management?

Many software problems

K.I.S.S.: “Attach SW to IVHM”

not that simple...

- Software problems don't develop over time
 - occur instantaneously
 - come in during all phase of SW life cycle
 - “don't go away”
- SW failures mostly occur instantly—HW often fails gradually (e.g., an oil leak)
- SW usually is *hybrid* and consists of
 - discrete (mode logic, finite state machines) and
 - continuous components (e.g., control, navigation)
- ...

HW-SW Interoperation

Problematic Hardware-Software Interaction can lead to software problems and system failures

- HW (e.g., sensors) can behave differently than expected (and thus cause a SW failure)
 - “on purpose”: use same SW for different HW
 - Ariane V failure
 - accidentally during development
 - DART: new GPS system just before launch
 - HW failure or degradation during operation
 - broken cable or sensor
 - increased sensor noise
 - disabled sensor
 - novel/unexpected environment



SWHM is a piece of SW

Quis custodiet ipsos custodes?

Juvenal

- The HM system that monitors the SW system must be at least reliable as the SW under scrutiny
 - false alarms are not an option
 - undetected failures are a safety hazard
- Rigorous V&V of HM system necessary, state-of-art testing not sufficient

“We are not alone in the Universe...”

- Obviously goals and challenges for Software Health Management are not new
- Many “HM” approaches for software exist
 - “the most trivial”
 - `if (x==0){ // if not healthy`
 - `exit(-1); // commit suicide`
 - `} ...`

Study on SWHM techniques

- We surveyed existing techniques and approaches that are related to the goals and challenges to SWHM
- Our survey on SWHM techniques
 - Identifies 18+ techniques and analyzes these according to 6+ major dimensions
 - Thus maps existing techniques related to SWHM
 - Can be used to determine the best approach(es) for given applications
 - Identifies strengths and weaknesses of current approaches
 - Can be used to develop better SWHM techniques

- **Phases in software life-cycle**
 - errors and faults in SW can show up in each phase of the software development cycle; from “early inception” to “after deployment”
 - techniques used during the various stages can help to
 - detect, isolate, and fix problems
 - robustify the software
 - in a SWHM system different techniques applied during different stages can work together synergistically
 - if you can *prove* that the code is free of “buffer overrun” errors, then you don’t need to monitor this during operation of the SW
 - The assumption “due diligence in software V&V has been taken” helps to focus SWHM.
 - What kind of problems *need* dynamic monitoring?

- **Fault handling approach**

- fault prevention
- fault removal
- fault tolerance
- fault forecasting

are the main subcategories of this dimension. Many approaches can be easily categorized according to this dimension.

Dimensions III

- **FDIR** (fault detection, isolation, recovery)
 - term often used in System Health Management
 - fault detection: identify that there is a fault in the system
 - isolation: identify source of fault and isolate it from the rest of the system
 - recovery: apply means to restore full functionality
 - similar: FDDR (Fault detection, diagnosis, and recovery)

Dimensions IV

- **Automation**

- manual
- semi-automatic
- automatic

depending on the application, technique, it can be necessary that the system must operate fully automatically and autonomously (no sys-admin is around for miles). Others allow for or require human interaction.

There is a trade space between level of automatic processing and power of the technique (e.g., manual proof vs. runtime monitoring)

- **Computational resources**

- CPU, memory, and bandwidth requirements

- **Completeness**

- Can the SWHM detect all faults?
- False alarms: ISHM goes off, but there is no fault
 - can cause missed opportunities (e.g., aborted mission), nuisance, degradation of performance
 - false alarms lower responsiveness to real alarms in human operators
- undetected failures: ISHM does not detect the fault
 - can produce a safety hazard
- most of the surveyed techniques are *not* complete
- many trade-spaces

SWHM Techniques Surveyed

Design & programming methodologies

Model-based design

Goal-based operations

Aspect-oriented programming

Recovery-based computing

Software configuration management

V&V techniques

Testing

Simulation

Debugging

Numerical analysis

Model checking

Theorem proving

Runtime techniques

Redundancy-based fault tolerance

Check-pointing and rolling back

Runtime monitoring

Trace analysis

Built-in tests

Software rejuvenation

Computer immunology

Self-healing software

Design & Development Phase

Technique	Fault handling	FDIR	Automation	Resources	Complete?
Model-based design	Fault prevention	n/a	Manual	n/a	No
Goal-based operations	Fault prevention	n/a	Manual	n/a	No
Aspect-oriented programming	Fault prevention	n/a	Semi-automatic	n/a	No
Recovery-based computing	Fault prevention, Fault tolerance	Recovery	Manual	n/a	No
Software configuration management	Fault prevention	n/a	Semi-autom.	n/a	No

Testing Phase

Technique	Fault handling	FDIR	Automation	Resources	Complete?
Testing	Fault removal	n/a	Manual, semi-automatic	Varied	No
Simulation	Fault removal	n/a	Automatic	Moderate-high	No
Debugging	Fault removal	n/a	Semi-automatic	Varied	No
Numerical analysis	Fault removal	n/a	Manual	Low	No
Model checking	Fault removal	n/a	Automatic	High	In certain cases
Theorem proving	Fault removal	n/a	Automatic	High	In certain cases

Post-Deployment Phase

Technique	Fault handling	FDIR	Automation	Resources	Complete ?
Redundancy-based fault tol.	Fault tol	Isolation, Recovery	Automatic	Varied	No
Checkpointing and rolling back	Fault tol	Recovery	Automatic	Varied	No
Runtime monitoring	Fault tol	Detection	Automatic	Minimal	No
Trace analysis	Fault tol	Detection	Automatic	Varied	No
Built-in tests	Fault tol	Detection	Automatic	Minimal	No
SW rejuvenation	Fault tol	Recovery	Automatic	Minimal	No
Computer immunology	Fault tol	Detection, Isolation	Automatic	Minimal	No
Self-healing software	Fault tolerance	Detection, Isolation, Recovery	Automatic	Varied	No

Conclusions

- many “HM” approaches for software exist, but they don’t meet all SWHM requirements
- Many approaches tailored for discrete software (state machines, mode logic) but not applicable for continuous or hybrid systems
- Weak area: *reasoning* about SW faults
 - How to identify the source of the fault?
- Weak area: *prognosis*
 - Can a SWHM system predict, when the software fails?
 - not in general: SW faults can occur instantaneously
 - behavior analysis of larger SW systems (e.g., OS with applications) can result in some predictions
 - file system at 98% capacity and continuous writes
 - increasing length of message queues, heap behavior, etc.
- Weak area: use of statistical/probabilistic techniques to monitor/analyze “quality” of calculation
 - how does a noisy input signal influence the calculation?
- Worthwhile to look out for synergies



ALARM: *Stop talking now!*

Questions?