

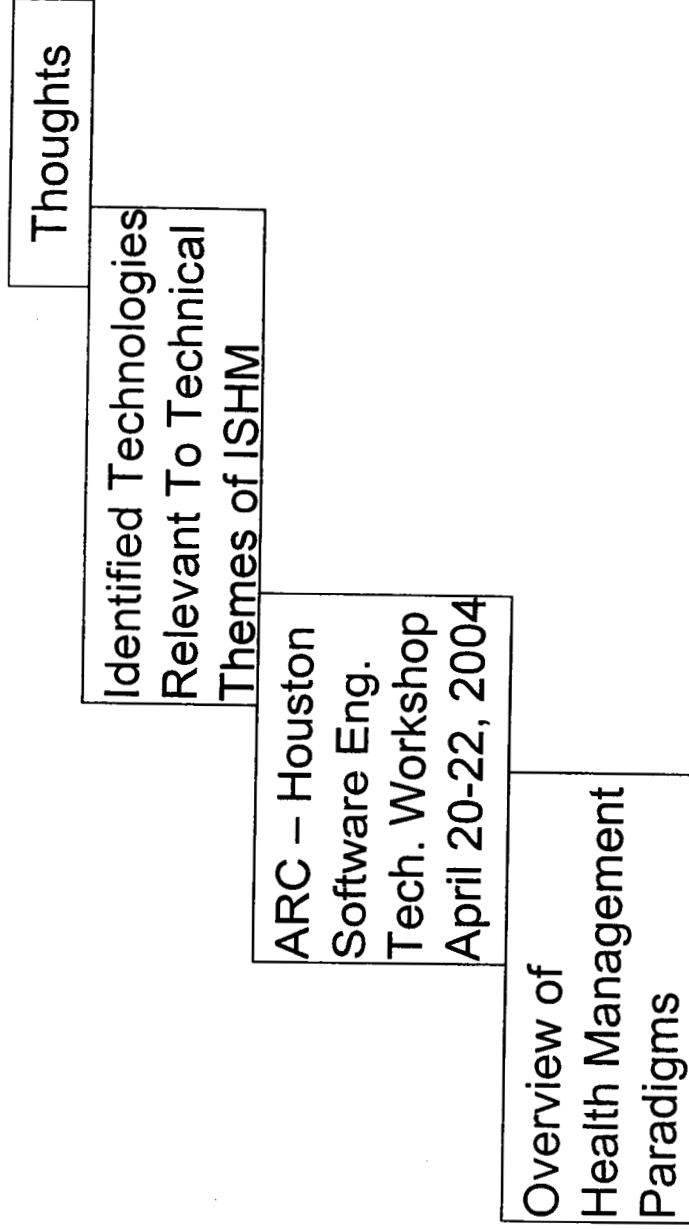


Software for Intelligent System Health Management (ISHM)

Briefing For
Georgia Tech, Intelligent Controls Laboratory
December 22, 2004

*Luis Trevino, Ph.D.
Advanced Sensors & Health Management Systems Branch
Spacecraft & Vehicle Systems Department
Engineering Directorate, MSFC*

Briefing Agenda





Overview of Health Management Paradigms

- Health Management (HM) technologies determine health of components / systems and subsystems for the purpose of informed-decision making either with humans in the loop or via Intelligent Autonomous Control
- Applicable to all aspects of space exploration – launch vehicles, CEV, upper stages, insertion/ascent stages, planetary habitats, etc:
- Technologies contributing to health management capabilities include:
 - 1) Advanced software algorithms, models, and software development technologies
 - 2) Fault Detection Diagnosis (including discrimination between component failures, sensor failures, internal software anomalies, actuator failures and nominal transients) , and recovery (or mitigation)
 - 3) Prognostics – the estimation of remaining life
 - 4) Information Fusion
 - 5) Degradation Management
 - 6) Smart Data Compression
 - 7) HM Technology Design Tools – HM needs to be incorporated as an integral part of the design process rather than an add-on. This will require a paradigm shift
 - 8) Software dependability (health management of software)



Overview of Health Management Paradigms

Autonomy and Intelligence (Per H&RT SISM Team)

■ Autonomy is a combination of three attributes:

1. Task complexity
2. Robustness to unexpected circumstances
3. Level of human commanding

■ Any program or device that can perform complex tasks in changing or incompletely known environments with little human oversight is by this definition autonomous. Thus from a systems engineering point of view, autonomy should be considered for any task that is non-trivial, is performed in an environment that cannot be fully predicted or controlled, and for which human oversight is limited or unavailable.

■ This last criterion, the unavailability of human oversight, plus the finite speed of light, are the fundamental source of NASA-unique autonomy requirements – no other agency, and generally no private companies who are not working for NASA, need to perform complex tasks far enough from earth that detailed human oversight becomes impractical

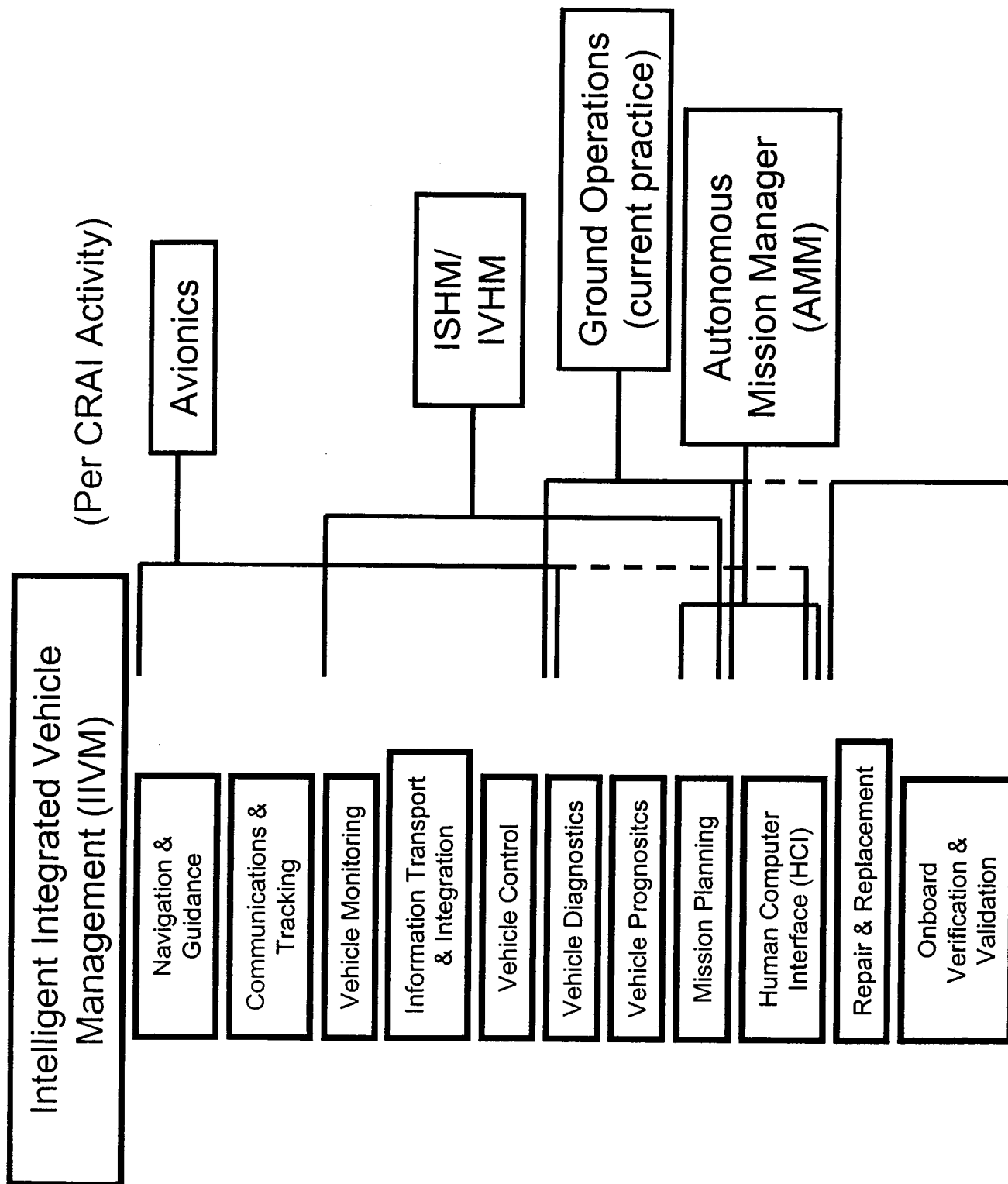
■ Intelligence pertains to the ability of devices and systems to be able to perform complex tasks robustly with limited human oversight (life support, power, propulsion, etc.)

Overview of Health Management Paradigms



1. **Intelligence Enables Safe Vehicle Operation (Per CRAI & MSFC Activities)**
 - a. Greatly increase Crew Safety and Mission Success
 - 1) Vehicle can continue to sustain crew & meet mission objective during communications disruptions
 - a) Critical factor in going to Mars
 - b) Eliminates crew safety dependence on remote communications capabilities
 - 2) Automated functions respond to unexpected events in milliseconds, manual onboard functions respond in minutes, ground functions can take minutes to hours.
2. **Intelligence Minimizes Crew and Vehicle Size**
 - a. Autonomy allows small crew sizes to safely operate complex vehicle functions
 - b. Enables smaller vehicles
 - 1) Reduced crew size affects living space volume, consumable storage, life support systems
3. **Intelligence Minimizes Ground Operations Staff**
 - a. Autonomy increases crew safety and mission success
 - b. Autonomy reduces current ground staff which are expensive to operate
 - c. Autonomy reduces issues with variance in Martian vs. Earth day cycle
 - d. Ground based flight support
 - 1) Maintain status of mission for NASA and public
 - 2) Distribute Science Information
 - 3) Provide engineering support in the event of major vehicle failures

Overview of Health Management Paradigms



Overview of Health Management Paradigms



- Vehicle Level Management Functions
 - GN&C
 - C&T
 - Mission Planning
 - Vehicle Control
- Subsystem Focused Management Functions
 - Monitoring
 - Diagnostics
 - Prognostics
 - Subsystem Control
- Subsystems: Propulsion, Flight Control, Structural, Electrical Power, Thermal Management, Crew Environment, Robotics, & Payload

ARC – Houston Software Engineering Technology Workshop

April 20-22, 2004



- Participants from DFR, LaRC, ARC, MSFC, GRC, JPL, JSC,
& FAA
- Used IVHM as example to address 2010 & 2024 gaps and needs
 - Today primarily a mixture of subsystems of Health Management
 - Monitoring, diagnostics, prognostics, trending (dealing with degradation)
 - Rigorous semantics: be able to reason about behavior (SE Tools)
 - Verification & Certification
 - Big Issue / Gap: How to build, test, & trust (incremental process)
 - Standard components defined in a way that enable auto verification
 - Levels of abstraction must not inhibit depth of diagnosis
- “Technology will come up with a good idea” (Dan Cooke)

Identified Technologies Relevant To Technical Themes of ISHM



Real Time Intelligent Software Elements List

- Software health management: self-monitoring, self-configuring healing and recovery, self validating (usage of interlocks, fail-safe, self checking mechanism techniques, model-based approaches)
- Model-based software fault recovery and software fault avoidance
- Real Time Onboard data mining and software trend analysis
- Enhanced on-board data storage, processing, and data retrieval (including data compression techniques, data integrity and quality, spacecraft as a web server, IP data routing, secure access)
- Advanced architecture & frameworks for Software



Identified Technologies Relevant To Technical Themes of ISHM

Real Time Intelligent Software Elements List

- Onboard mission and maneuver planning, execution, attitude control, and collision avoidance
- Intelligent machine / human relationships: Natural language high level task uploads, interface to direct science and spacecraft goals & priorities
- System Real-time and health monitoring and automated fault detection, avoidance, isolation, & recovery
- Advanced software techniques to address Single Event Upsets

Identified Technologies Relevant To Technical Themes of ISHM



Real Time Intelligent Software Elements List

- **Dynamic on-board reconfiguration of flight software**
- **System and Software Real-time performance tuning**
- **Enhanced software voting techniques**
- **Automated sensor & actuator calibration and integration**
- **Partitioning between mission and non-mission critical applications**

Identified Technologies Relevant To Technical Themes of ISHM



Intelligent Software Engineering Tools

- Software analysis tools including software reverse engineering tools, static software analysis tools, and real-time software analysis and verification tools (e.g., prediction of software defects and of future software system trajectories & validation envelope penetration)
- Software practices for COTS certification and integration
- Utility & certification of auto-generated code tailored to NASA software from design specs
- Generic software simulators / test beds
- Methods for V&V of software systems (Model-based autonomous, intelligent, adaptive flight control, etc.)



Identified Technologies Relevant To Technical Themes of ISHM

Intelligent Software Engineering Tools

- Methods to automate the verification & regression testing of software, its interfaces, and its test procedures
- Device independent interface software
- Software assurance practices for reused / heritage software
- Life cycle robustness, especially for new applications (emerging paradigms and algorithms): need better or combined lifecycle models for reliable software development and test indicators and metrics
- Graphical and readable software representation tools (graphical modeling languages)



Identified Technologies Relevant To Technical Themes of ISHM

Intelligent Software Engineering Tools

- Software risk assessment tools
- Software requirements hazard analysis – fault tree analysis
- Software requirements capture
- Rapid prototyping to explore mission software requirements and design specifications
- Personnel management: process, tools, etc. offset software personnel turnovers
- New software languages & techniques (e.g., objective oriented, real time applications, Java, etc.)
- Libraries of standard components for development & reuse

Identified Applications Relevant To Technical Themes of ISHM



Agency Wide Activities:

- CRAI (Capabilities Requirements Analysis Integration)
 - Concepts: IAHM&C, IIVM, IVM
- NGLT
- ISHM Agency Wide Working Group
- NASA SWG – Software Technology Infusion Strategy Three
 - Collaborations Involving ARC, JSC, JPL, IV&V, MSFC, LaRC
- Collaboration with ARC on automated software analysis & verification tools
- Others (e.g.): JSC Architecture Study, Mars Reference Mission, OASIS

Space Shuttle Main Engine Avionics Technologies

- Improved methods for software testing of mission critical software systems
 - Automated Offline Test Generation Technology
 - Improved methods for software testing of mission critical software systems
 - Generic Simulation Technologies
 - Software Analysis Technologies
 - Expansion of the parameter simulation/patching Technologies
 - Real-time Data Reduction/Analysis Technologies
 - Real-time Data Display/Analysis Technologies
- AHMS Phase IIB



Thoughts

■ **Software**

- Intelligent Software Engineering (ISE) is needed: Tools, Test & Verification Platforms, Efficient Development Processes, automation, results interpretations, risk mgmt, requirements, etc.
- Ties health management systems all together

■ **Intelligent Systems (IS)**

- Needed for more autonomous operations, crew safety, and mission success
- Autonomous operations includes reconfigurability, diagnostics, & prognostics

■ **Modeling**

- Improved formal methods & mathematical model development needed for further supporting ISE for IS

■ **Universal Theory (Standard) of ??IVHM?? And corresponding discipline needed**

- HM paradigms must exhibit situational awareness & docile features

■ **Verification & Validation (Certification) Technologies need to progress to accommodate larger state space of possible test scenarios**



Questions ???