Failure Modes and Effects Analyses for Large Software Intensive Systems

Myron Hecht, Eltefaat Shokri, Alexander Lam, and Matthew Keegan

Presented to
2008 System and Software Technology Conference
Las Vegas, Nevada

THE AEROSPACE CORPORATION

Outline

- Motivation
- System Description
- Steps for Performing an FMEA for a Software Intensive System
- Conclusion
- Acronyms
- References

Motivation

- **Demonstrate the application of system safety analysis methods to a large software intensive system of national significance**
- **Required by Air Force Policy Directive 63-12 (1 February 2000)**
  - ❖ **The Air Force will**
    - – Assure the operational safety, suitability, and effectiveness of all systems and end-items currently in, or entering, the operational inventory.
  - ❖ **Definition of Operational Safety:**
    - – The condition of having acceptable risk to life, health, property, and environment caused by a system or end-item when employing that system or end-item in an operational environment. This requires the identification of hazards, assessment of risk, determination mitigating measures, and acceptance of residual risk.

System Characteristics

- Large ground control systems built on conventional Information Technology (IT) platforms
- Software:
  - OS platforms: COTS
  - Middleware (e.g., messaging middleware, DBMS)
    - Mostly COTS, Sometimes NDI, Rarely developed for the system
  - Applications
    - Mostly developed for the system, Sometimes NDI, Rarely COTS
  - Human Interfaces
    - Sometimes developed for the systems, Sometimes NDI, Rarely COTS
- Hardware (all COTS):
  - Computing nodes (consist of processors, internal interconnect, and memory)
  - Network devices (e.g., NIC, router)
  - Storage devices (e.g., RAID, NAS, SAN):
  - Interconnections (gateway, firewall, wires)

Steps for Performing an FMEA for a Software Intensive System

- **Determine the architectural level of analysis (level of indenture)**
- **Identify the units of analysis, their boundaries, and environments**
- **Define Effect Levels and Severity**
- **Define platform failure modes (to be included in each unit analysis)**
- **Define software failure modes**
- **Define hardware failure modes**

Determine the Architectural Level Of Analysis

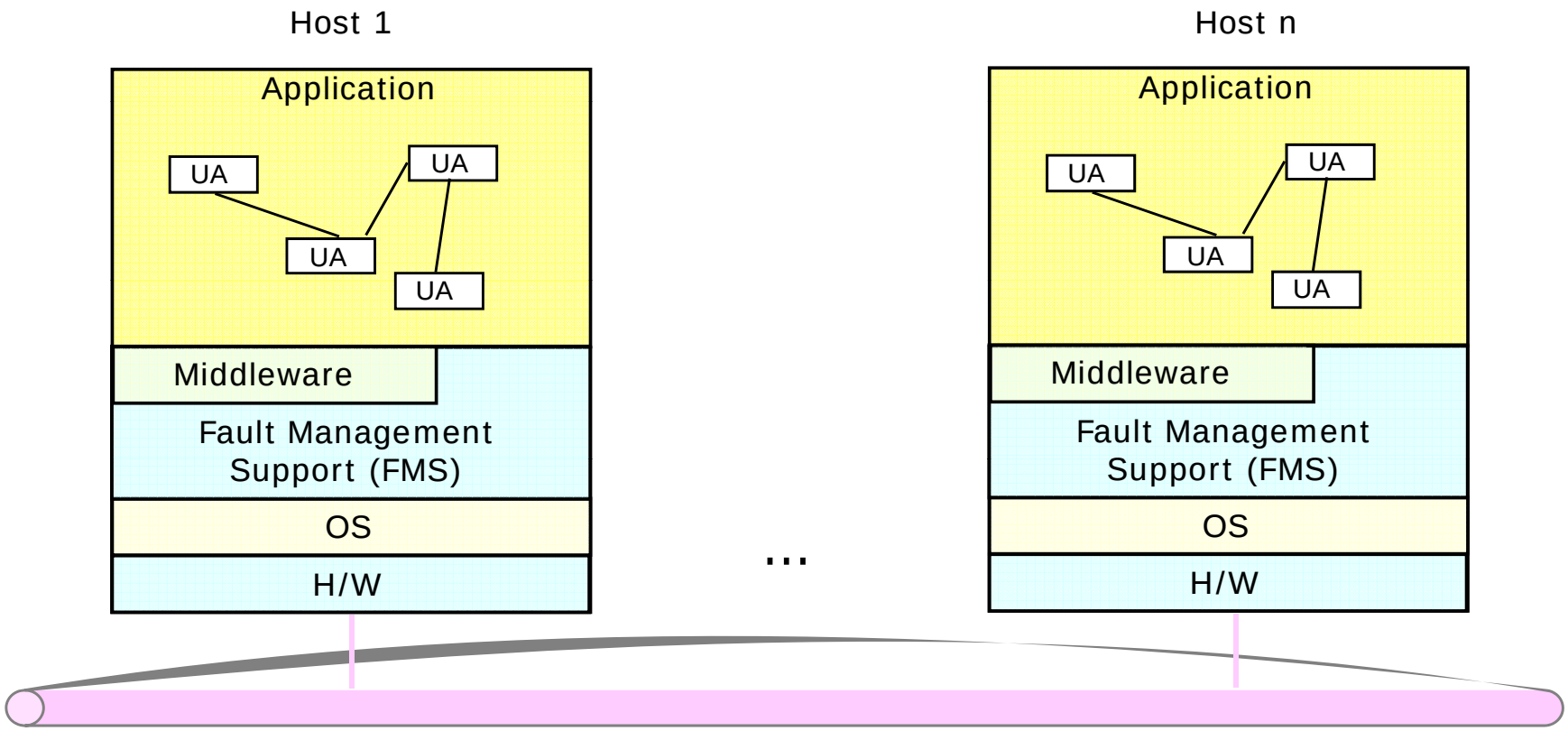
- **Unit of analysis (UA) - The unit of software at the level of indenture for which failure modes are identified and analyzed.**
  - Examples: processes/tasks, or collections of processes/tasks that interact with the rest of the system through common interfaces.
- **Tradeoffs for selecting units of analysis:**
  - Smaller units enable a more detailed analysis of the components and interactions.
  - Larger units of analysis may be more cost effective.
  - Limited by the detail and accuracy available in the design documents and code.
    - Example: a detailed analysis of the COTS elements of the system may be impractical.

Steps for Performing an FMEA for a Software Intensive System

- **Determine the architectural level of analysis (level of indenture)**
- **Identify the units of analysis, their boundaries, and environments**
- **Define platform failure modes (to be included in each unit analysis)**
- **Define software failure modes**
- **Define hardware failure modes**
- **Analyze failures for each unit of analysis**
  - ❖ Identify effects, severities, and mitigations for each failure mode

Distributed Execution Environment

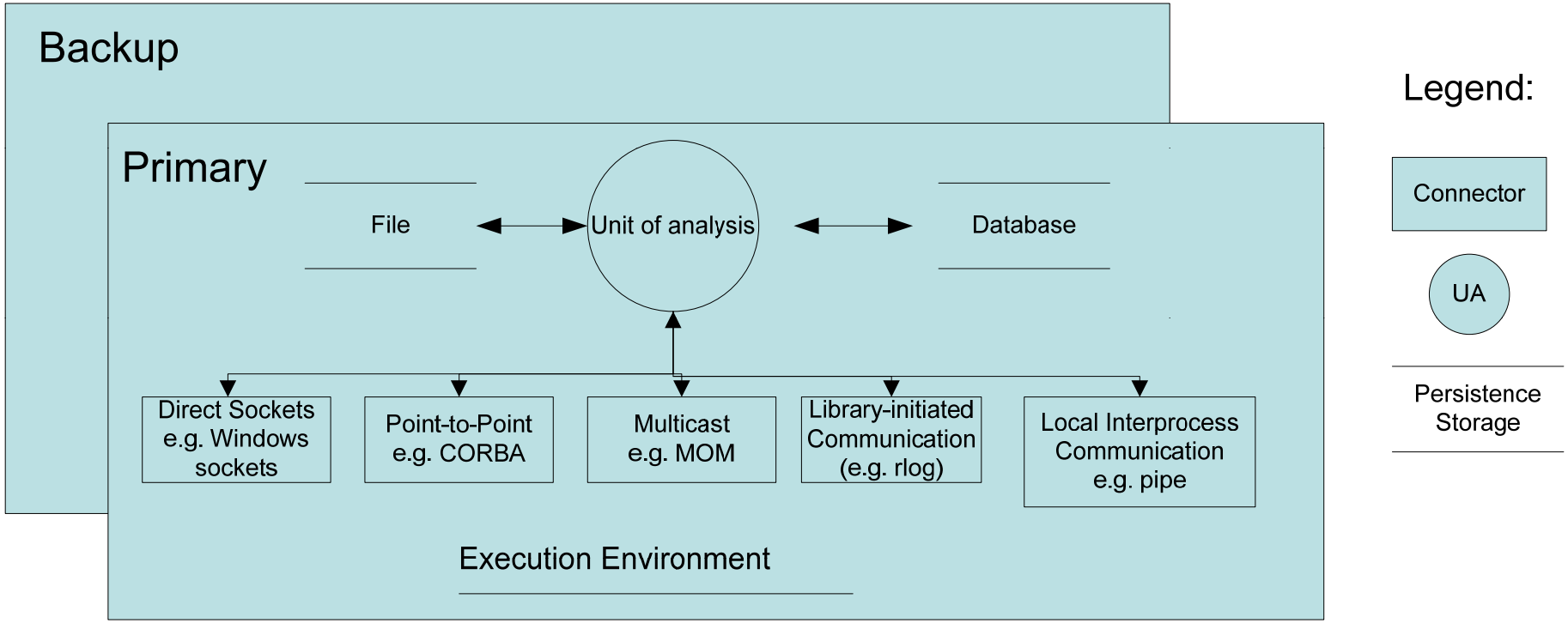
Sample Architectural Depiction (abstracted)



Note: Middleware, FMS, and OS units are either platform UA's or special cases of application UA's

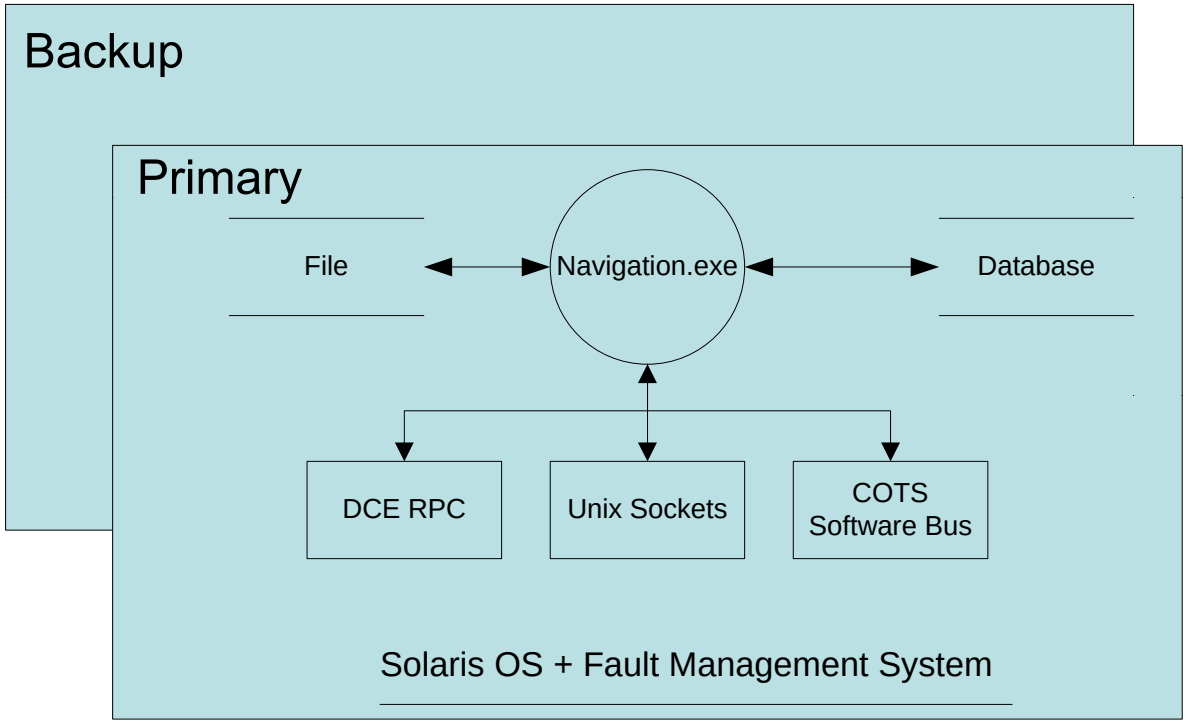
Resultant Boundaries, Environment, and Units of Analysis

- For each unit being analyzed, consider all interactions.
- Primary and backup UA instances should be analyzed individually.



Boundaries, Environment, and units of analysis: An Example

The example UA, Navigation.exe, uses configuration files, stores/retrieves data in a database, and communicates with other units of analysis via RPC, Sockets, and a software bus.



Analyze:

1. Software executable units
2. Three Inter-process communication methods
3. Database operations
4. File input/output
5. FMS
6. Operating System

Steps for Performing an FMEA for a Software Intensive System

- **Determine the architectural level of analysis (level of indenture)**
- **Identify the units of analysis, their boundaries, and environments**
- **Define Effect Levels and Severity**
- **Define platform failure modes (to be included in each unit analysis)**
- **Define software failure modes**
- **Define hardware failure modes**
- **Analyze failures for each unit of analysis**
  - ❖ Identify effects, severities, and mitigations for each failure mode

Defining Effects

- **Distinguishing between local and higher level effects**
  - Local effect: impact on the unit of analysis
  - Next higher level effect: impact on the services and/or operations of the system
  - End effect: Impact on the end system.
- **Non-local effects are system specific**

Example: Defining Effects

- **Distinguishing between local and higher level effects**
  - Local effect: impact on the UA and immediate context
    - Process loss or degradation
    - Loss of hardware resource (disk, processor, network, interface)
    - Loss or degradation (late, incorrect) or loss of process output
    - Loss of database currency or integrity
  - Intermediate effect: impact on system operations
    - Impact on user of the UA (e.g., operator)
    - Impact on external network (e.g., high traffic, etc.)
    - Impact on database
    - Impact on recovery capability
  - End effect: Impact on end system
    - Satellite systems examples
      - Impact on telemetry, tracking or control
      - Impact on Navigation signal performance (accuracy, integrity, continuity, availability)

Assessing Severity

- **Category I**
  - Catastrophic – Failure which causes system loss
- **Category II**
  - Critical – Failure which causes major system damage resulting in mission loss
- **Category III**
  - Marginal – Failure which results in delay or loss of availability or degrades system operations
- **Category IV**
  - Negligible/Minor - Failure not serious enough to cause system damage; loss of partial functionality for short time

Note: Severity levels adopted from MIL-STD-1629A; When there are multiple effects base the severity on the most sever effect

Example: Assessing Severity

- **Category I**
  - Loss of Space Vehicle (SV)
- **Category II**
  - Undetected database integrity loss in SV
  - Jeopardize SV navigation message integrity
  - Jeopardize SV continuity
  - Undetected incorrect data through external output interfaces
- **Category III**
  - Degrades SV operations
  - Interruption of functionality that does not reach severity of Category II
- **Category IV**
  - Loss of partial functionality for short time

Steps for Performing an FMEA for a Software Intensive System

- **Determine the architectural level of analysis (level of indenture)**
- **Identify the units of analysis, their boundaries, and environments**
- **Define Effect Levels and Severity**
- **Define platform failure modes (to be included in each unit analysis)**
- **Define software failure modes**
- **Define hardware failure modes**
- **Analyze failures for each unit of analysis**
  - ❖ Identify effects, severities, and mitigations for each failure mode

Characteristics of Software Platform Failures

- Failures occurring in the operating system, fault management system, and other middleware will generally lead to application failures.
- A set of software platform failures should be identified and applied to each UA analysis.
- While the causes of these platform failures may be common, their effects, severity, detection methods, and compensating provisions may vary depending on the UA being evaluated.
- Reusing a set of pre-defined software platform failures will provide more consistent and complete evaluations with reduced effort.

Software Platform Failures List: Platform Example

	Failure Mode		Failure Mode
1	**OS Crash -- multiple causes**	7	**SNMP agent crashes or hangs**
2	**OS hang**	8	**System logging facilities crash or hang**
3	**Unacceptable delay caused by Solaris OS malfunctioning.**	9	**GUI not initializing, crashes, or hangs**
4	**Network File System Crash-multiple failures**	10	**OS & Infrastructure Initialization Error (device drivers, file system, networking)**
5	**Hardware-disk full – Fault management is able to detect this failure**	11	**COTS & NDI Initialization Errors (e.g., software bus)**
6	**Hardware-disk full - Fault management is not able to detect this failure**	12	**Application Initialization Error (.cshrc script, . login script, etc.)**

Example of a Software Platform Failure Mode Analysis

ID	**100**
Host	**TTC**
Function	**Solaris OS**
Failure Mode and Causes	**Crash -- multiple causes**
Mission Phase	**Operational/Primary**
Local Effect	**Loss of application functions**
Next Higher Level Effect	**Loss of local application functions and data**
End Effects	**Local application unable to provide services to users**
Failure Detection Method	**Detected by the Failure Management System**
Compensating Provisions	**Failover to backup host**
Severity Class	**IV**
Remarks	**Severity class assumes that failover from primary to backup host is successful.**

Note: Failure Mode Analysis fields adopted from MIL-STD-1629A

Steps for Performing an FMEA for a Software Intensive System

- **Determine the architectural level of analysis (level of indenture)**
- **Identify the units of analysis, their boundaries, and environments**
- **Define Effect Levels and Severity**
- **Define platform failure modes (to be included in each unit analysis)**
- **Define software failure modes**
- **Define hardware failure modes**
- **Analyze failures for each unit of analysis**
  - ❖ Identify effects, severities, and mitigations for each failure mode

Software Failure Modes

- **UA internal failure modes**
  - Examples: Crash, hang, incorrect result, untimely result
- **Inter-process communications failure modes**
  - Examples:
    - Failure of multi-cast communication (e.g., software bus, messaging middleware)
    - Failure of point-to-point communication (e.g., RPC)
    - Failure of socket communication (e.g., UDP, TCP)
    - Failure of local inter-process communication (e.g., pipe)
    - Failure of library based communication (e.g., remote login)
- **Database Reads and Writes failure modes**
  - Examples: Failure to connect, read, and write
- **File I/O failure modes**
  - Examples: Failure to open, read, and write
- **Fault management failure modes**
  - Failover/restart mechanisms
    - Example: Failure to restart or failover
  - Hot Standby mechanisms
    - Example: Failure of backup host/application to take over

Example of a Software Failure Mode Analysis

ID	**120**
Host	**Producer CSCI Host**
Function	**Producer CSCI Process**
Failure Mode and Causes	**Tracking software produces incorrect result**
Mission Phase	**Operational/Primary**
Local Effect	**Incorrect result output**
Next Higher Level Effect	**Incorrect result propagated into subsequent calculations resulting in cumulative error**
End Effects	**System integrity lost**
Failure Detection Method	**Reasonableness check on output**
Compensating Provisions	**None**
Severity Class	**II**
Remarks	**Not all errors will be detected; replace reasonableness check with inverse calculation**

Steps for Performing an FMEA for a Software Intensive System

- Determine the architectural level of analysis (level of indenture)
- Identify the units of analysis, their boundaries, and environments
- Define Effect Levels and Severity
- Define platform failure modes (to be included in each unit analysis)
- Define software failure modes
- **Define hardware failure modes**

Hardware Failure Modes

- Startup Failure
- Operations Failure (unannounced cessation of function)
- RAM data corruption
- Persistent data corruption
- Communication data corruption
- LAN channel babbling
- Power supply/conditioning short
- Power supply/conditioning open
- Discrete contact stuck at 0
- Discrete contact stuck at 1

Hardware Elements

- Standalone Processors
- Blade server
- Equipment cabinets
- SANs
- NASs
- Routers
- Crypto units
- Firewalls
- CSU/DSU
- LAN Cables
- Wiring Hubs

Failure Modes: Standalone Processors (Servers and Workstations)

- **Startup Failure**
  - Effect: no output
  - Example Mitigation: retry; switch to alternate; repair/replace
- **Operations Failure (unannounced cessation of function)**
  - Effect: no output
  - Example Mitigation: retry; switch to alternate; repair/replace
- **RAM data corruption**
  - Effects: no output (if causes processor crash); corrupted data
  - Example Mitigation: Error correcting codes
- **Persistent data corruption on Direct Attached Storage**
  - Effects: corrupted data
  - Example Mitigation: CRCs, multiple writes for critical data
- **Communication data corruption**
  - Effects: See LAN, NAS, SAN
- **Power supply/conditioning short**
  - Effect: power surge causes overheating; loss of output
  - Example Mitigation: fuses and circuit breakers
- **Power supply/conditioning open**
  - Effect: no power; loss of output
  - Example Mitigation: switch to alternate; repair/replace power supply

Conclusions

- Software focused analysis is necessary in large software intensive systems
- Software architecture provides a basis for defining units of analysis.
- Generally accepted failure modes and local effects can be identified for many categories of units of analysis
- Higher level effects are system specific
- Methodology presented here has been applied to a large scale ground system
- Other methodologies are applicable to smaller scale systems

Acronyms (1)

- CMOS - Complementary Metal–Oxide–Semiconductor
- CORBA - Common Object Request Broker Architecture
- COTS - Commercial Off-The-Shelf
- CRC - Cyclic Redundancy Check
- CSCI - Computer Software Configuration Item
- CSU - Channel Service Unit
- DB - Database
- DBMS - Database Management Software
- DSU - Data Service Unit
- FMEA - Failure Modes & Effects Analysis
- FMECA - Failure Mode, Effects and Criticality Analysis
- FMS - Fault Management Support/System
- GPS AEP - Global Positioning System Architectural Evolution Program
- GUI - Graphical User Interface
- H/W - Hardware
- LAN - Local Area Network

Acronyms (2)

- MOM - Message-Oriented Middleware
- NAS - Network Attached Storage
- NDI - Non-Development Item
- NIC - Network Interface Card
- NFS - Network File System
- OS - Operating System
- RAID - Redundant Array of Independent Drives
- RAM - Random Access Memory
- RPC - Remote Procedure Call
- S/W - Software
- SAN - Storage Area Network
- SNMP - Simple Network Management Protocol
- SV - Space Vehicle
- TCP - Transmission Control Protocol
- UA - Unit of Analysis
- UDP - User Datagram Protocol

References

- **MIL_STD-1629A: www.uscg.mil/hq/gm/risk/E-Guidelines/RBDM/html/vol4/Volume4/Tool-spec_Rec/FMEA/MIL-STD-1629A.pdf**
- **William Greenwell, Gail Haddock, Myron Hecht, Steven Meyers, Eltefaat Shokri, and Elisabeth Nguyen, "Safety Analysis Methods for Software Intensive Satellite Ground Systems", *Space Systems Engineering and Risk Management Conference,* Los Angeles, CA, February, 2008, available from http://www.aero.org/conferences/riskmgmt/**

- **E-mail: myron.hecht@aero.org**