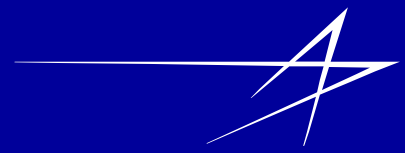
LOCKHEED MARTIN

Emerging DoD Need for "Trustworthy" Software Systems

"Software for the Real World" Panel Briefing
Software Design and Productivity Workshop
April 18th & 19th, 2001

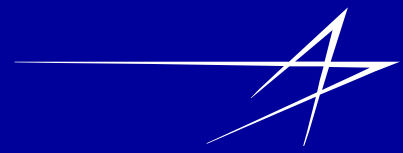
Patrick Lardieri
Patrick.J.Lardieri@lmco.com
Principal Member, Engineering Staff
Lockheed Martin – Advanced Technology Laboratories

A Sampling of Forces Driving Future DoD Systems



- **Economic**
  - Expanding Mission Areas and Competing Policy Initiatives are Creating Pressure in the DoD to
    - Reduce Manning
    - Leverage Open Source/Commercial Products
- **Technological**
  - Commercial Market Forces are Driving
    - HW performance & scale improvements
    - More Complex SW Infrastructure Technologies and Frameworks
  - To Enable More Complex Systems the DoD Is Driving Enhanced Information Infrastructures
    - Semantic Web
  - Increased Potential for Autonomous Behavior
- **Social**
  - Eliminate people from hazardous environs
  - Ubiquitous, secure , reliable access to information and services
  - Dynamic and adaptive collaboration with foreign national coalitions

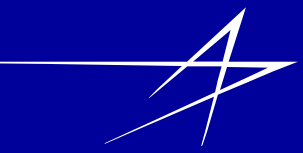
Implications for Future DoD Systems



- **Aggressive Increase in Automation**
  - — Replace people with software
- **HMI Shift from Operators to Managers**
  - — Evolution from a master-apprentice to a leader-staff relationship
- **Allocate More Complex Behaviors to Software**
  - — People and software will possess different skills
- **Capabilities of Both People and Software Will Evolve**
  - — Both Will Require Training and Certification
  - — People and Systems Will Exhibit Emergent Behavior

Migration of Trust from People to "People and Software" (e.g. FCS seeks policy based decision making at the leaf node)

Implications for Future Software Technology



- **Improve Software Predictability**
  - Capture Formal, Computable Descriptions of non-Functional Requirements
    - Timeliness, Security, Reliability, Concurrency
  - Ubiquitous, Hierarchical, Policy-Based, Resource Specification and Scheduling Mechanisms
  - Domain Level (vice software level) System Description Languages
    - Formalisms with provable properties enabling automated pattern based design and code generation (e.g. want to generate CORBA code)
- **Analysis, Testing and Certification of Software Behavior**
  - Patterns for Software Governors
  - Techniques to Categorize Bounds on Emergent Behavior
  - Standards for Certification
- **Legal Advocacy**
  - Establish Clear Lines of Responsibility

Example: State-of-the-Market in COTS Based Real-Time Distributed Computing

- **Today we cannot deliver predictable, discriminated real-time performance using commercial IT infrastructure**
  - **Requires QoS mechanisms in the application, middleware, OS and network to achieve**
  - **Sporadic offerings across numerous products**
  - **No end-to-end scheduling tools to coordinate numerous configurations and/or reservations**
- **Hand tailored, static scheduling of individual elements yields excellent soft-real time behavior, in the presence of congestion but outside of failure conditions**
- **Technologies to support hard-real time distributed computing with automate static scheduling and bounded failure recovery is the required ante for COTS based systems with trustworthy real-time systems**