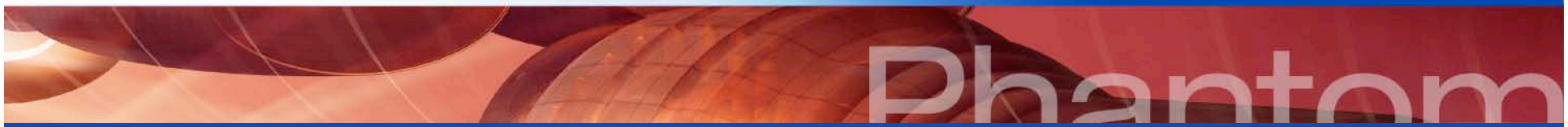
Boeing Technology
Phantom Works



Industrial Challenges in Composition of Embedded Systems

Dr. David Corman, Dr. James Paunicka

Technical Fellow – Network Centric Operations

Boeing Phantom Works

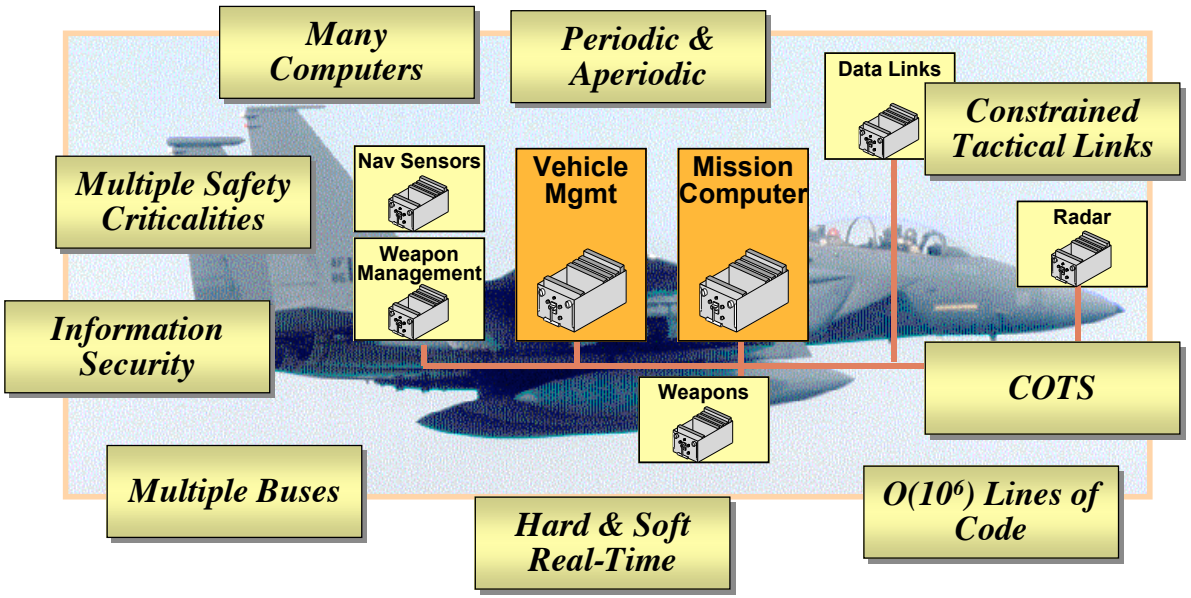
16 October, 2006

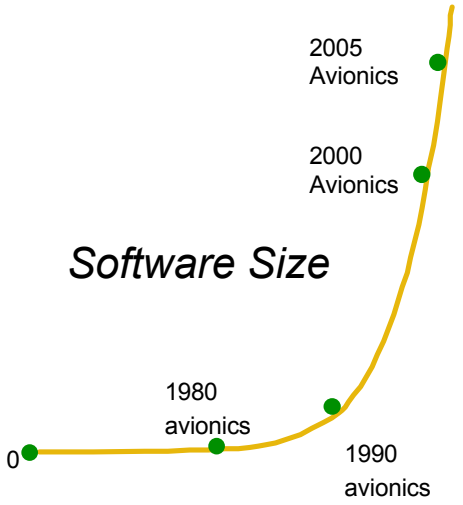
Who We Are

- The Boeing Company, working a rich set of application areas that will benefit from research focused on composition of real-time embedded systems
  - Air (manned and unmanned, commercial and military)
  - Space (ultra high-reliability applications)
  - Land (e.g., Future Combat Systems)
- Complexity of emerging system challenges is outstripping our capability to affordably develop and compose the next-generation of real-time embedded systems
- Virtually all new developments are focused on component based systems
  - All can benefit from additional research
  - Most are highly dynamic and require some level of “dependable” performance
- Aircraft platforms
  - Commercial
    - Primarily safety challenges
  - Military
    - Security and Safety challenges
    - Multi-service
    - Piloted and autonomous aircraft
    - Production and research customers

Embedded Systems Technology Challenges

- **Operational Complexity of Large-scale Embedded Systems Is Growing Exponentially**
  - **New capabilities: autonomous UAVs, Mixed Initiative, broadband aircraft networking,**
  - **Extended capabilities: real-time mission replanning…**
  - **Highly dynamic operating environments**
  - **Operation in a SoS Network**





- ***Avionics S/W challenges***
- ***Networking challenges***
- ***Information Management challenges***
- ***Application challenges***
- ***Verification, Validation and Certification challenges***
- ***Software Engineering challenges***
- ***Commercial vs military domain***

Typical Mission Computing Legacy Characteristics

System size of 10 – 30M not unknown

- 10-100 Hz Update Rates
- Up To 10-100 Processors
- ~1M Lines of Code
  - O(10^3) Components
- Proprietary Hardware
  - Slow CPU, small memory
  - Fast I/O
- Test-Based Verification
- Mil-Std Assembly Language
- Highly Optimized For Throughput and Memory
- Functional Architectures
  - Flowchart designs
- Frequently No Maintained Requirements or Design
  - Ad-hoc models used by algorithm developers
- Hardcoded Hardware Specific Single System Designs
- Isolated Use Of
  - Multi-processing
  - Schedulability analysis
    - Frequently overly pessimistic to be used

David Sharp, Boeing Phantom Works, HSCC Plenary Talk, Stanford, March 2002

Typical Vehicle Management Legacy Characteristics

Additional Characteristics

- **80/160 Hz Update Rates**
- **Single CPU System/ Quad Redundant**
- **Dual/Quad Redundant Sensors and Actuators**
- **<100K Lines of Code**
- **Extensive Built-In-Test**
  - **>50% of code**
- **Extensive Testing**
  - **Very conservative development culture**
  - **>50% of effort**
- **Control System Models Carefully Developed And Used**
  - **Matlab/MatrixX with auto code generation**

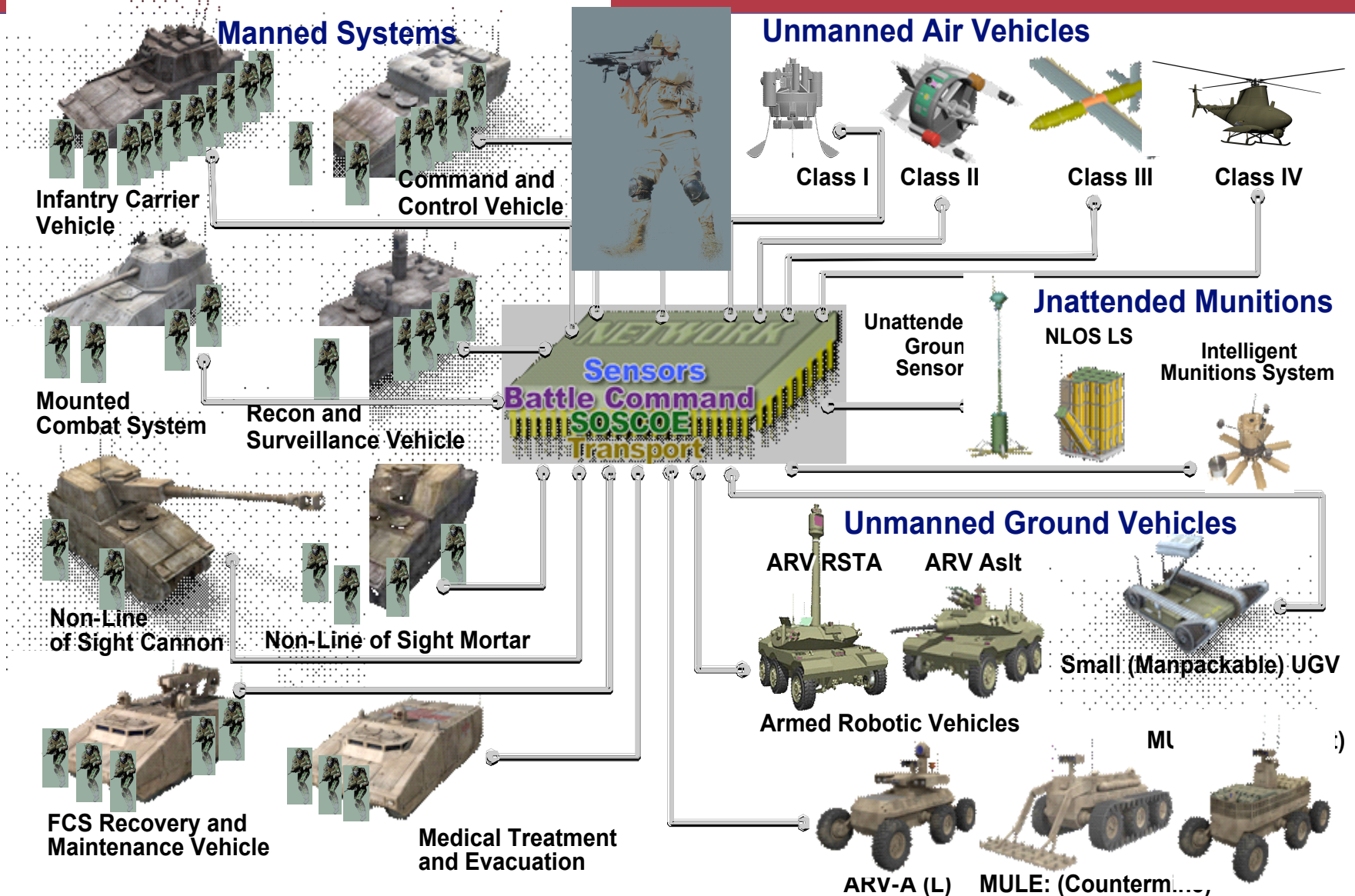
David Sharp, Boeing Phantom Works, HSCC Plenary Talk, Stanford, March 2002

Networked System of Systems Example:

FCS 18 Integrated Systems + Network + Soldier

FUTURE COMBAT SYSTEMS
FCS
One Team-The Army/Defense/Industry

NCO Thrust



Approved for Public Release, Distribution Unlimited, TACOM 14 March 2006, case 06-046.

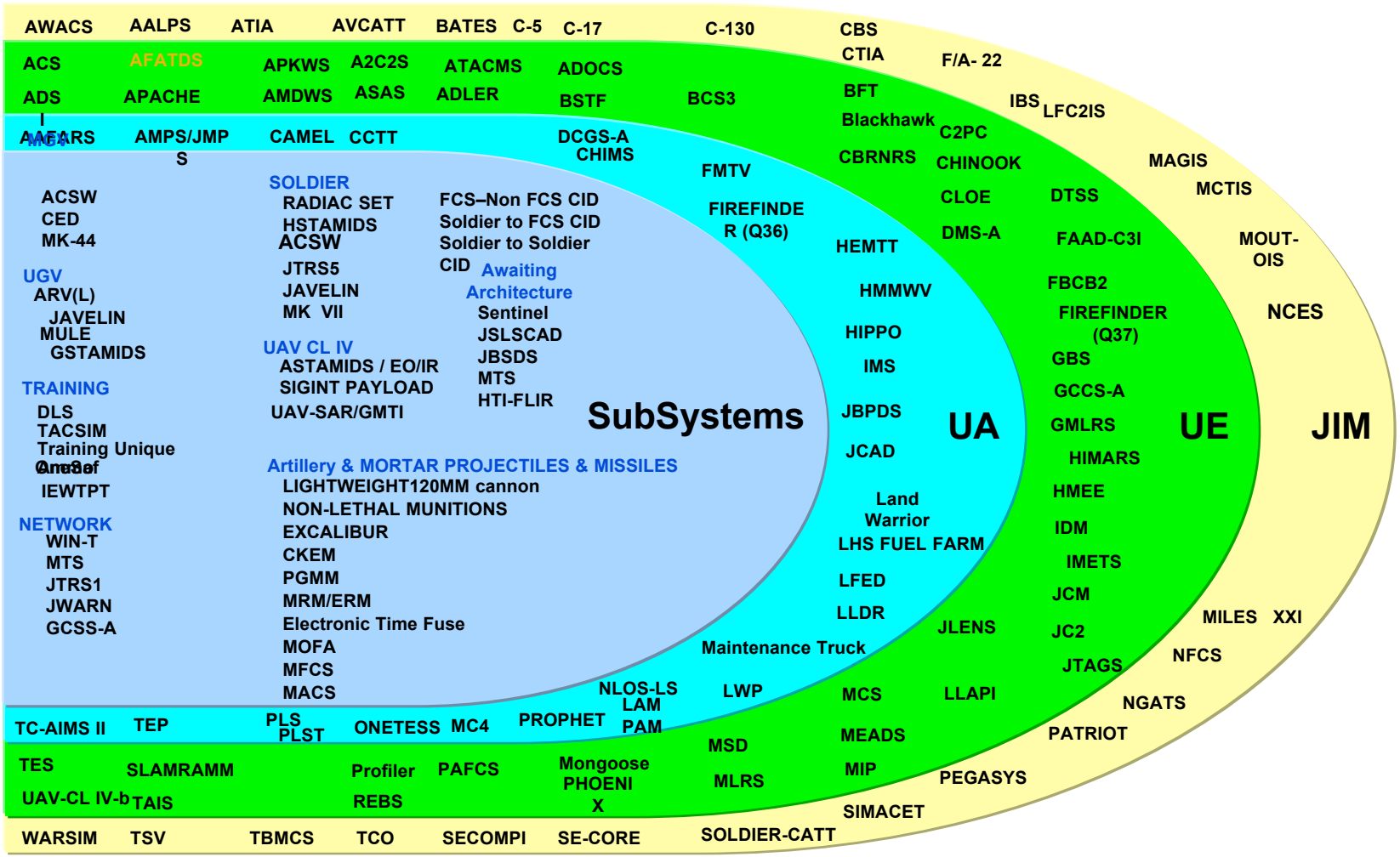
Networked System of Systems Example:

FCS Integration With Other Existing Systems

FUTURE COMBAT SYSTEMS FCS One Team-The Army/Defense/Industry

Boeing Technology | Phantom Works

NCO Thrust



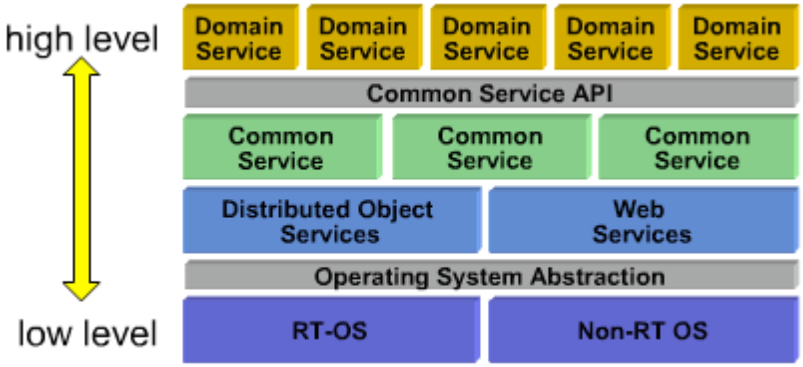
Subsystems in UA | Unit of ACTION Systems | Unit of EMPLOYMENT Systems | Programs Outside of UE

Approved for Public Release, Distribution Unlimited, TACOM 20 Jan 2005, case 04-171.

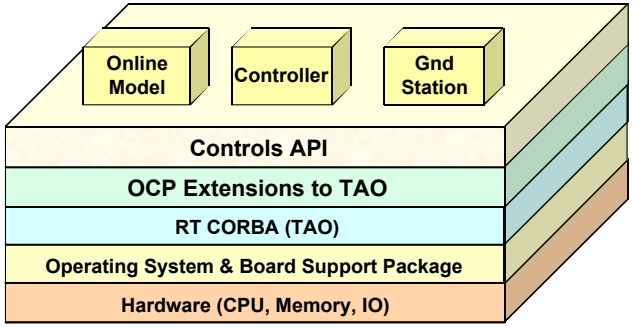
Boeing Leverages Middleware Based Architectures to provide Hardware and Operating System Independence

Boeing Technology | Phantom Works

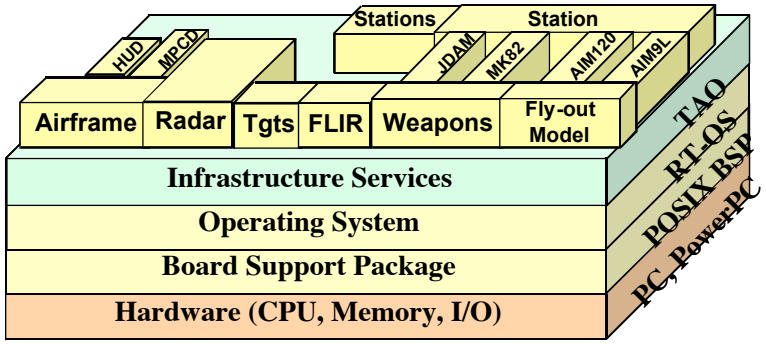
NCO Thrust



FCS SOSCOE Architecture

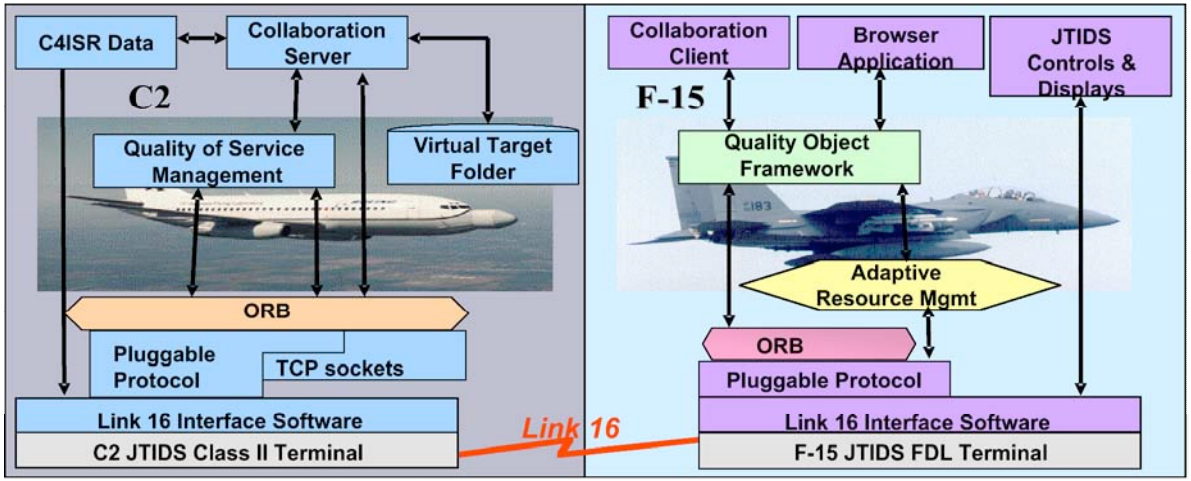


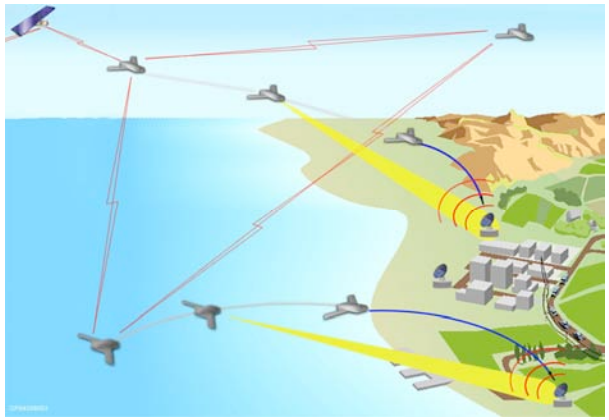
Open Control Platform for Autonomous Systems



Bold Stroke Product Line Architecture
(F/A-18, F-15, T-45)

Avionics Software Challenges: Dynamic System Behavior

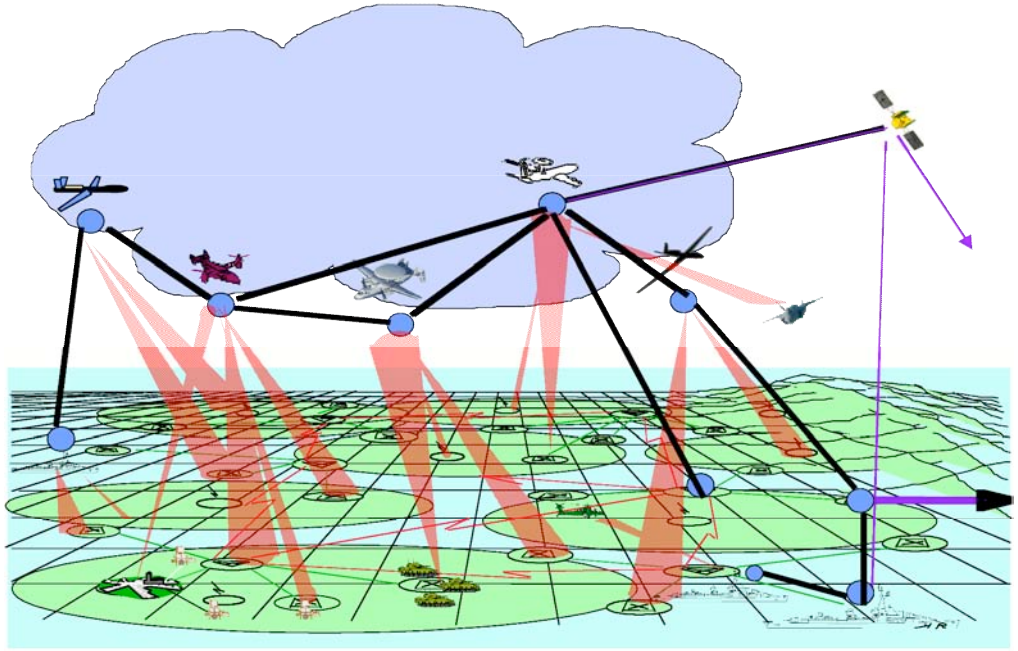




- New applications require dynamic behaviors
  - Mixture of hard and soft real-time tasks
  - Active resource management and dynamic scheduling
  - Mode changes with component configuration changes
  - Dynamic changes to system membership
    - Swarms
    - NCO
    - Power conservation
- Current component/system models favor static systems
  - CCM
  - OCP
- Must meet Embedded/Real-Time constraints in a dynamic setting
  - Need to handle during system execution things that were typically dealt with “out of band” at startup

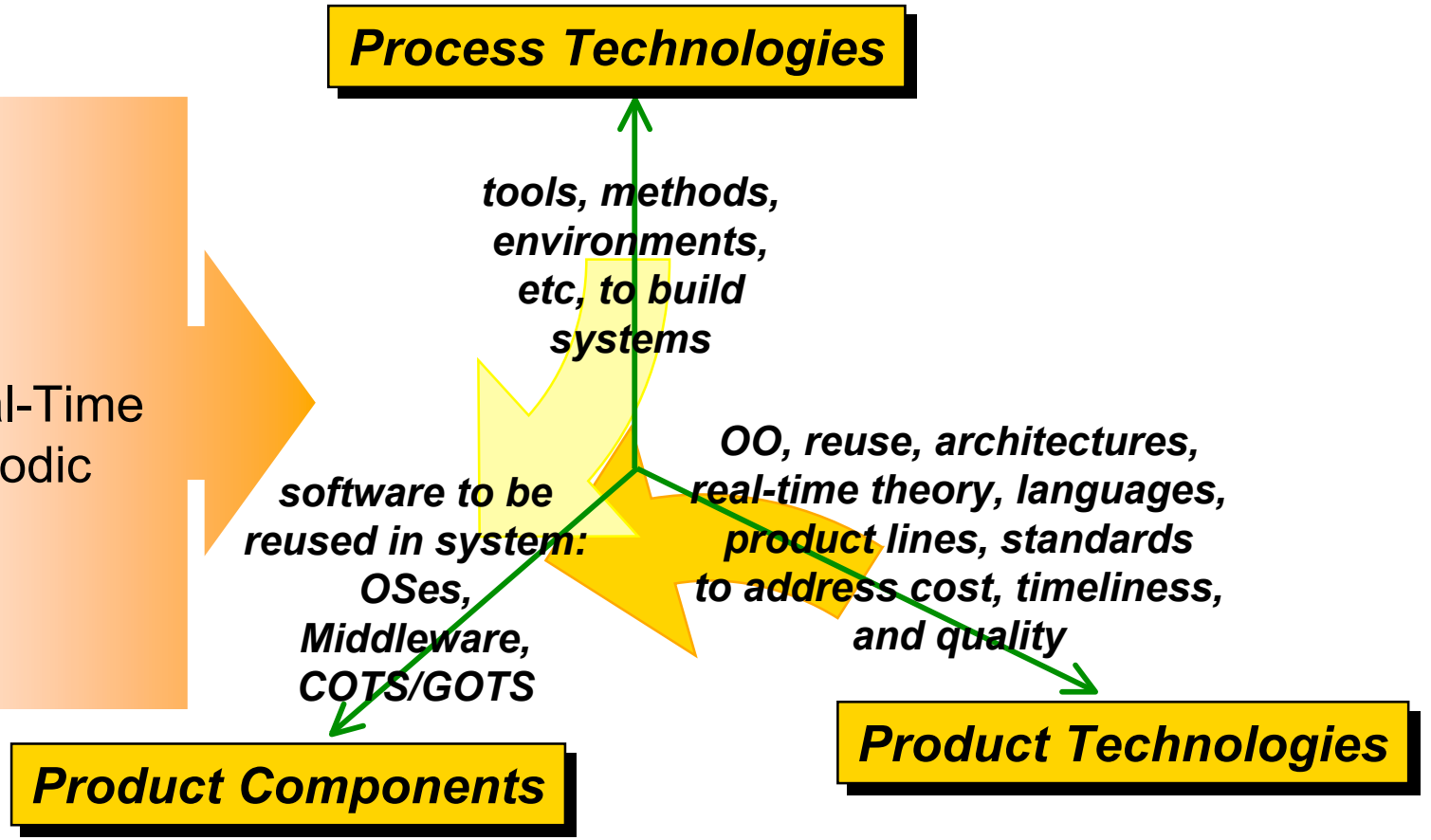
Networking Challenges

- **Smart routers**
  - **Help deal with massive amounts of data**
- **QoS in "open" networks**
  - **Latencies**
  - **Predictability**
  - **Guaranteed delivery**
- **Bandwidth**
  - **Wireless**
  - **Mobile**
  - **Satellite**
  - **Airborne**
- **Heterogeneous/federated networks**
- **Inherently unreliable networks**
- **Networked systems ranging from Motes upwards**



Multiple Competing Dimensions & Aspects Must Be Addressed, Balanced and Integrated

- Affordability
- Quality
- Timeliness
- Security
- Hard/Soft Real-Time
- Periodic/Aperiodic
- Safety
- Reliability
- TRL…



Process Technology Challenges

- **Design Technologies**
  - **Domain specific multi-view modeling & analysis**
    - **Capture cross-cutting systemic concerns (deployment, composition, messaging, fault tolerance, etc.) in native architectural terms**
  - **Integrated dynamic design space exploration & analysis**
    - **e.g. traffic analysis, system loading, operational latencies, path tracing, safety/security non-interference, etc.**
  - **Generative technologies for correct-by-construction synthesis**
    - **Architectural abstractions, common pattern application, synthesized composition configuration, middleware integration, etc.**

- **Verification & Validation Technologies**
  - **Flexible & reconfigurable SoS simulation & validation**
    - **Tailorable to specific mission context**
    - **Composable in number and type of participants, topology, etc.**
    - **Integrating real and simulated/surrogate/synthesized elements**
  - **Completeness in varying topologies & operational contexts**

Need *Scalable* technologies to design, analyze, verify and validate large Networked SoS.

Product Technology Challenges

- **Product Line Architectures**
  - **Contain product variability in heterogeneous security, criticality, timeliness & physical deployment concerns**
- **Component models & patterns for uber-scale systems**
  - **Enabling software variability across & within systems while addressing cross-cutting aspects**
- **QoS / Resource Management**
  - **Resource & behavioral adaptation to:**
    - **MANET topology, membership and capacity changes**
      - Including local vs. coordinated networked behavioral tradeoffs
    - **Intermittent communications availability (missed updates, dropped messages)**
    - **Changing mission context & commander's guidance**
  - **… while maintaining**
    - **Information separation**
    - **Multiple degrees of timeliness and criticality**
    - **Concurrent mission activities (e.g. operational & training)**
    - **Limited/competing communications throughput**
  - **… both at the system and SoS levels**

Need *Composable* approaches to enable heterogeneous system variability while addressing changing resource availability.

Product Technology Challenges (cont'd)

- **Information Management**
  - **Maintain coherent SoS situational awareness across possibly out-of-synch individual systems**
    - **Right information to the right user at the right time**
  - **Programming patterns, approaches and architectures for dealing with peer-to-peer ad hoc network communications**

- **Scheduling & Deployment**
  - **Hard, soft & non real-time, periodic & event-driven processing**
  - **Maintain optimal software deployment against changing user roles, mission context, systems modes and hardware availability**

- **Criticality**
  - **Intermixed safety critical, mission critical and non-critical processing**
    - **Deployment isolation, resource allocation, certification, etc.**

- **Information Assurance**
  - **Integrated IA architecture across GiG to the end-system user**
    - **Accounting for USI to multiple classification levels**

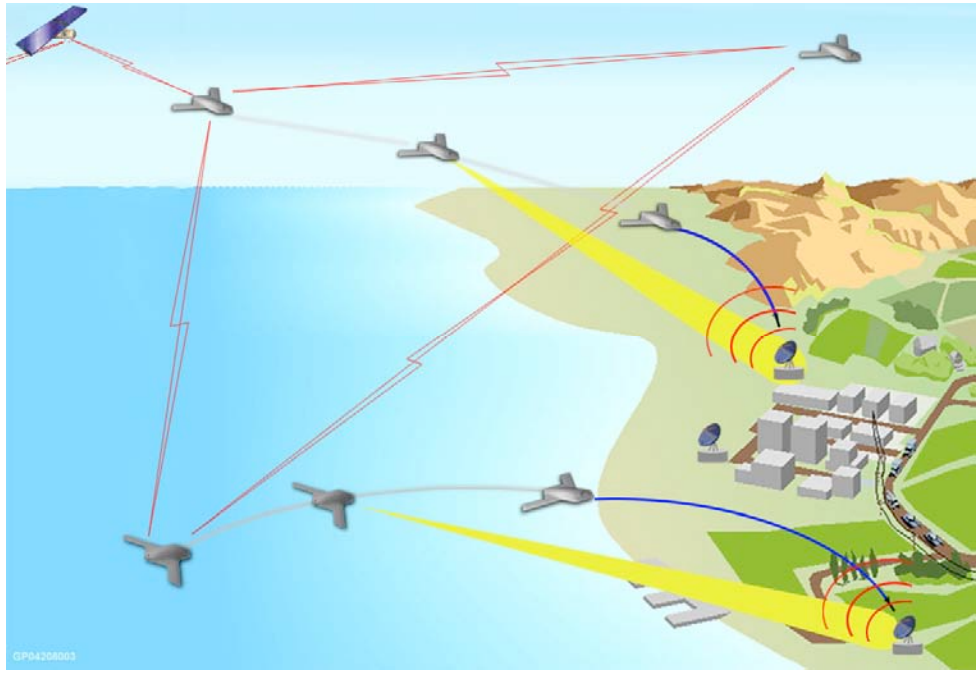
Product Component Challenges

- **COTS Standards-based Network-centric SoS Middleware**
  - **Composable from minimal safety certifiable profile to enterprise-level adaptive & reflective**
  - **Providing integrated network management for traffic flow reservation, priority negotiation and admission control**
  - **Supporting pushed & pulled, method-based & data-centric communications**
  - **Supporting static & dynamic peer-to-peer service discovery in a multi-network changing MANET topology**
  - **Optimized for limited network throughput**
  - **With reliable communications capabilities in support of safety critical operations**
  - **Tolerant of intermittent wireless comm failures**

- **IA components**
  - **High bandwidth trusted networking, cross domain and display components in support of tactical GiG to the soldier**

Multi-Entity Embedded System Challenges

- **Dynamic replanning and response to uncertainty**
- **Resource and task allocation**
- **Mixed initiative operations**
  - **Allocation of initiative to human operator or automata**
- **Fault-Tolerance**
- **Heterogeneous platforms**

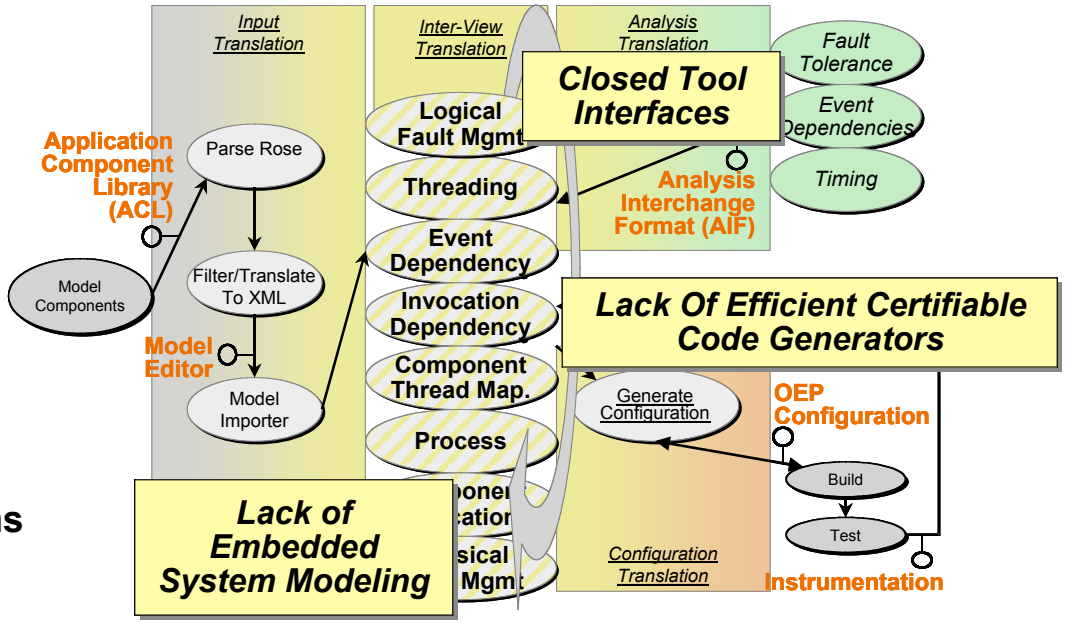


Software Engineering Challenges

- Creating component models/development environments for "legacy" systems
  - May not always be able to select the component model first
    - Have, instead, an existing infrastructure to create a component model and development environment for
    - Define component model consistent with legacy platform
      - Abstractions appropriate for
        - Component based development
        - Existing infrastructure
        - Functional and non-functional requirements
    - Generate development environment
      - Modeling tools
      - Code generators
      - Infrastructure adapters
      - Testing/analysis tools
- Data collection
  - Analysis techniques require data that:
    - Does not currently exist
    - Frequently is difficult to obtain
    - Too often impractical to create directly
  - Ex: timing and state transition data for legacy system/components

Avionics Software Challenges: Integration

- **Component/System integration is still hard**
  - **The MoBIES problems are still out there**
  - **Component integration satisfying QoS**
    - **Real-time**
    - **Concurrency**
    - **Distribution**
  - **Constrained by**
    - **Large, distributed development teams**
    - **Product line based reuse**
    - **Legacy systems, platforms and processes**
  - **With Efficient/Effective**
    - **Performance analysis, prediction and testing**
    - **Testing and iterative development**



Verification, Validation and Certification Challenges

- **V&V and Certification is expensive, and getting more expensive, for fielded systems**
- **Future advanced manned and un-manned systems may not fit naturally under current V&V and Certification regimes**
- **Need approaches for efficient V&V and Certification for emerging technologies for them to be deployable**
  - **Multi-entity Systems**
  - **Human interaction with Autonomy.**
  - **Fused Sensor Systems**
  - **Adaptive Systems that change with environmental stimulus**
  - **Mixed Criticality- Functions dependent on information of varying confidence**

Program Composition of Embedded Systems (PCES) Program – Case Study in Unmanned Systems

- **Challenge - Compose UAV embedded software to support dynamic behaviors with network centric connectivity to a mission commander on a tightly resource constrained system**
- **Development and composition from new UAV components executing on Product-line Reusable Scalable eMbedded (PRiSm) framework**
- **Demonstration as part of DARPA / ARMY / Air Force joint military exercise at White Sands Missile Range, New Mexico**
- **On-board Autonomy**
  - **Enabling Software technology for hosting autonomy on ScanEagle**
  - **C++ and Real-Time Java software platforms**
    - **International award for Innovations in Use of Java**
  - **On-board autonomy allowed UA to map it own path in response to high-level commands from a mission commander controlling multiple UA**

Source: Boeing Press Release www.boeing.com/news/releases/2005/q2/nr_050420o.html

- ***Public-release video***

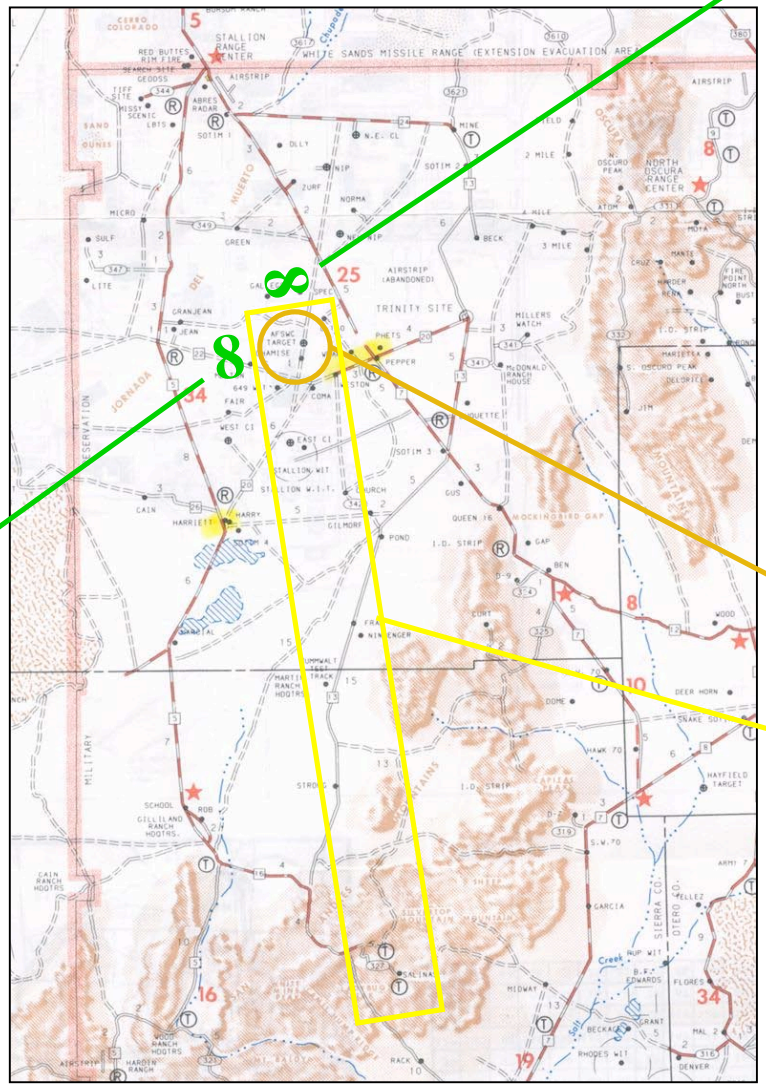
Technical Progress and Tech Transition

- **RT-Java middleware, event channels, and frameworks evaluated**
  - **Demonstrating real-time performance in embedded avionics application**
  - **Hosted on relevant embedded hardware in relevant vehicle (ScanEagle)**
  - **Realizing small footprint designs**
- **C++ PRiSm (Product-line Reusable Scalable Embedded) middleware exhibiting real-time performance**
  - **Relevant hardware / relevant vehicle**

Source: Boeing Public Release video for PCES demonstration

Technical Progress and Tech Transition

- **PCES RT-Java and C++ applications implementing autonomous maneuvering in ScanEagle to support time-critical target recce and monitoring of surface target weapon strikes**
- **Infospace leverage in integrated capstone demonstration**



Source: Boeing Public Release video for PCES Demonstration

Public Release Video

Boeing Technology | Phantom Works

NCO Thrust

Summary – Pervasive Industrial Strength Challenges in Composition of Embedded Systems

- **Dynamicism**
  - System of systems with changing participants
  - Changing modalities of individual systems
  - On-line code generation
- **Heterogeneity**
  - System of systems
  - Heterogeneous/federated networks
  - Heterogeneous collaborations
    - Ad hoc coalitions
  - COTS/GOTS components in an overall system
- **Fault Tolerance**
  - Unreliable networks
  - High confidence
  - Mission effectiveness in the presence of failures
  - IVHM
- **Scalability**
  - Massive data flows
  - Systems of many systems
  - Ever larger end systems
- **Certification and V & V**