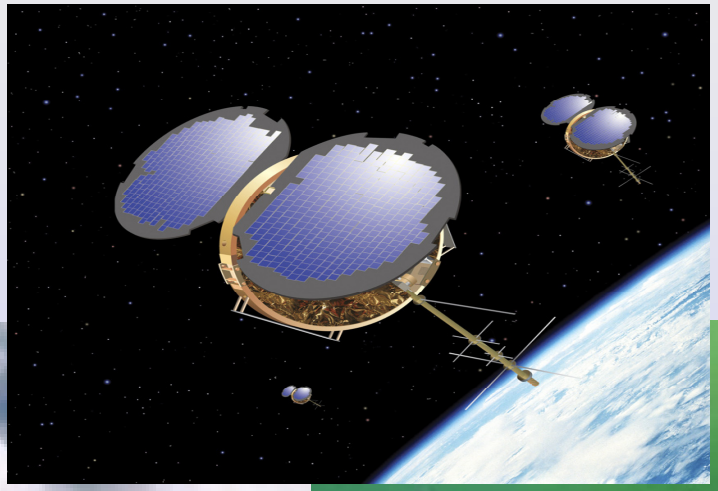
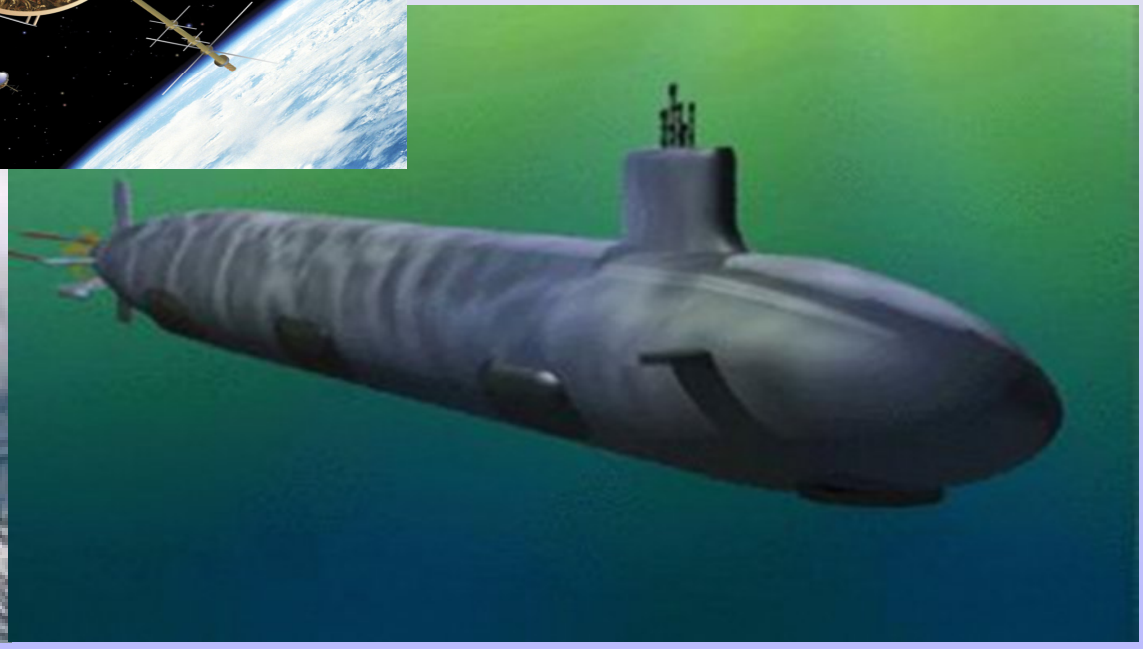
The WW Technology Group





Dependable Solutions & tools for Integrated Modular Avionics

Dependable Solutions for Integrated Modular Avionics

Dr. Chris J. Walter
cwalter@wwtechnology.com
410-418-4353

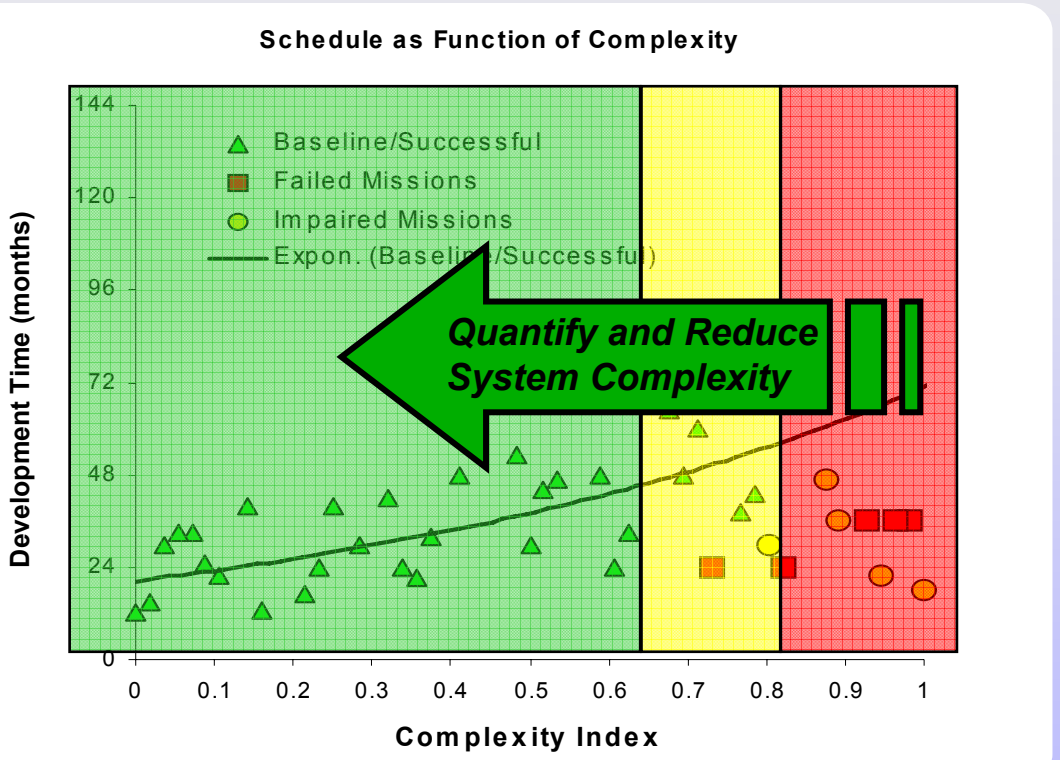
November 13, 2007

Lesson Learned

- Integrated
  - integration may reduce the physical connections but radically increase information/logical connections
  - not all connections are apparent, new failures can be happen due to
    - sneak paths
    - priority inversion
    - bus management
    - metastabilities and loading factors
  - what's on paper doesn't always translate easily to the platform
- Modular
- Avionics

Increased Functionality and Complexity Are Stressing Design Capabilities

- Requirements for fault tolerance, safety and security drive up complexity of systems
- As system complexity grows certifying system interactions becomes more difficult
  - Cross Domain Analysis required to ensure competing system properties can be traded off
  - Quantification and early evaluation of desired system properties leads to reduced certification efforts and system development time/cost
- Architectural level analysis is a natural point to bring together many concerns



Data from D. Bearden, Fourth IAA International Conference on Low-Cost Planetary Missions

System Design Can Be Streamlined With Architectural Analysis

Modularity

- Modular
  - implies packaging of functionality and instantiations
  - how to assess coupling and cohesion?
    - information locality?
    - levels of security/risk?
    - operator locality?
    - maintenance locality?
  - providing modules that support additional goals
    - does the modularity support
      - performance
      - security
      - safety
      - maintainability

Avionics

- provides a context and meaning for integrated and modular
- Implies
  - safety
  - enough performance for real-time control
  - types of errors to anticipate
  - fault tolerance is required
  - methods must be certifiable

Perceptions and Reality

- Some perceived solutions
  - channelize functionality
    - input and voting plane issues
    - metastability
  - distribute
    - system splits
  - simplex
    - who's right? on-line controller or monitor?
- Reality
  - truth v. consensus
  - metastability

Understanding Limits

- Do we really understand limits of our proposed solutions?
  - points where solution breaks down and failure may be imminent.
- Formal methods very useful
  - implacable skeptic
  - doesn't carry biased perceptions
  - only as good as the model and checking procedures
  - therefore best if implacable skeptic is not restrained to benign or simple cases but is allowed to explore radical possibilities to ensure robustness across full problem space
    - this is a radical idea since it is expensive and requires more time
    - challenge is to make this less onerous

Strategy – Step 1

- First address modularity
- Define associated attributes
  - functions
    - performance
  - ilities
    - dependability
    - security and safety
- Identify relationships
  - establish “acceptability” for application domain
  - establish clear reasons why things outside this space are unimportant
    - program requirements
    - known policies

Strategy – Step 2

- Next address Integration
- what level of integration makes sense?
  - coupling factors
  - cohesion factors
  - complexity metrics
- Ensure no violations in policies or modularity strategy

Strategy – Step 3

- Address avionics domain needs
- Extend integrated modular design to be
  - fault tolerant
    - can it be compositionally constructed?
      - depends on nature of problem and modular elements
        - observability of errors
          - response time
          - error types
        - schedulability
    - are there asymmetric dependencies?
      - is one element more important than another
        - e.g. rad-hard processor or memory
      - is this asymmetry justified
        - is it economical (most likely more costly and another part to inventory)

Partitions

- Partitioning is a useful concept
  - can be used to contain errors (ECRs)
  - can be used to contain functionality (Virtual Machines)
  - can be used to restrict information flow (Mixed Criticality Levels for Safety/Security
  - can be used to restrict events (TDMA)
  - all the above
- Divide and Conquer approach useful for accelerating performance and improving fault tolerance
  - clusters v. ECRs

The Value Of System Model Based Analysis Techniques Has Been Established

- The Verification and Validation of Intelligent and Adaptive Control Systems (VVIACS) Project sponsored by the Air Force Air Vehicle Directorate
  - Identified key technologies that reduce system certification costs in high confidence software applications
    - System Model Based Design
    - Automated Verification Management
    - Rigorous Analysis for test reduction

- The WWTG Design for Certification approach applies similar techniques to a similar application domain
- The VVIACS study results are directly applicable to many submarine ship board systems

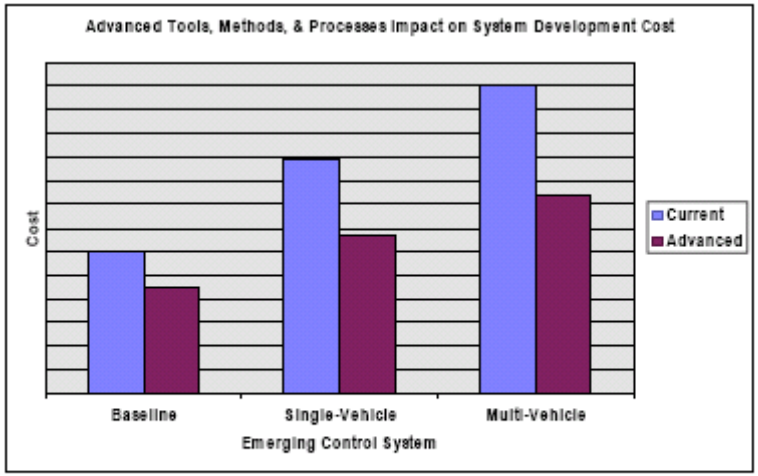


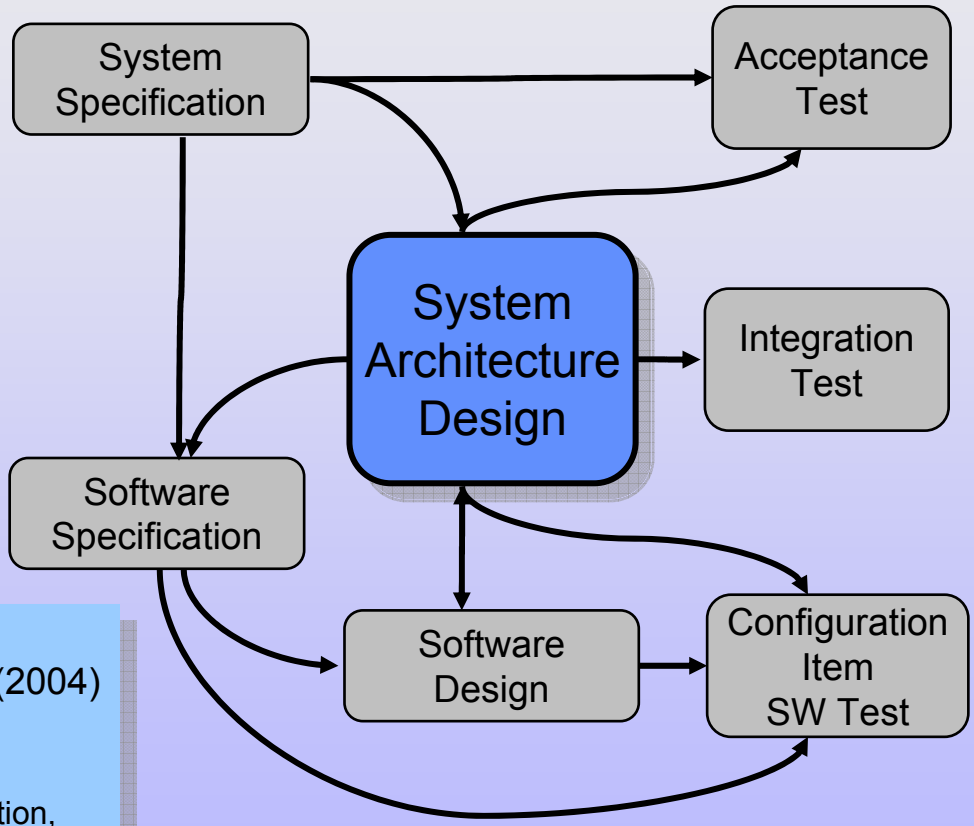
Figure 3 – V&V Technologies Impact on System Development Cost

From:http://chess.eecs.berkeley.edu/hcssas/papers/Storm-HCSS_avionics_positon_paper.pdf

The projected cost savings are 25%-35% of development costs for software and test

Why Address These Issues At The Architectural Level?

- System Architecture plays a central role in system development
  - Architecture influences many of the key system development activities
  - Architecture is a natural repository for system characteristics that are most difficult to certify
  - Architecture is the natural place to perform analysis on these parameters



Excerpts from the National Academy of Science Workshop On Software Certification and Dependability (2004)

"Systems integration has remained a *vexing challenge* "

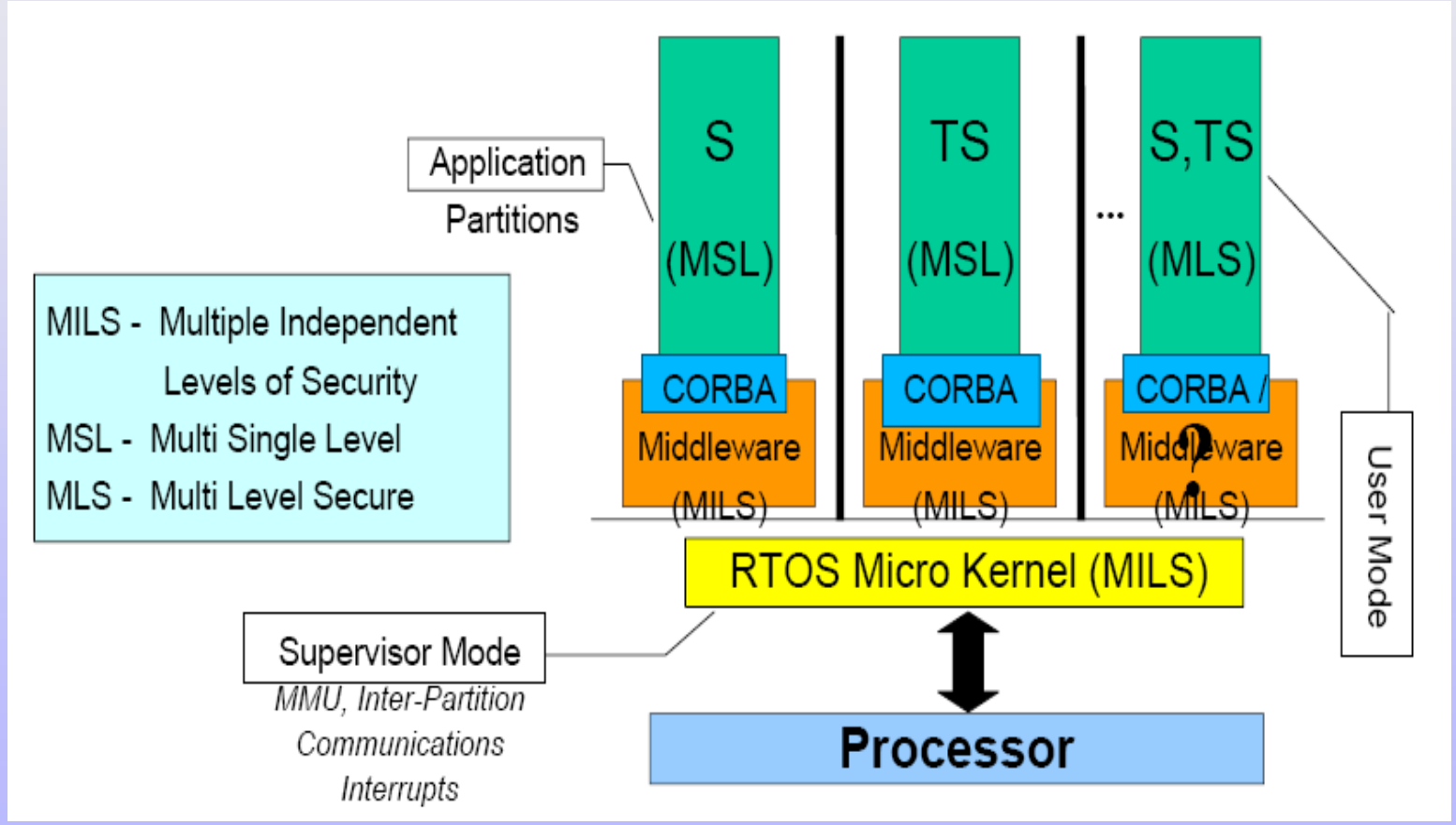
"There are many *integration problems* caused by unanticipated interactions between different technologies developed in isolation, especially in *aspects related to real-time properties, fault tolerance, security, and concurrency control*"

"How to *control and manage interactive complexity* is a key challenge for systems integration "

Multiple Independent Levels of Security/Safety (MILS/S)

- Goal is to protect the flows of information and guarantee that information assigned to different security levels is handled appropriately.
- Significant challenge to design MILS/S that is guaranteed to perform correctly with respect to security and safety.
  - John Rushby first introduced concept in the early 1980's for architecting secure systems using a separation kernel to reduce the security burden.
- Separation kernel mediates interactions between applications and enforces a security policy of information flow and data isolation on those interactions.

High Assurance MILS Architecture



W. Mark Van fleet, et al

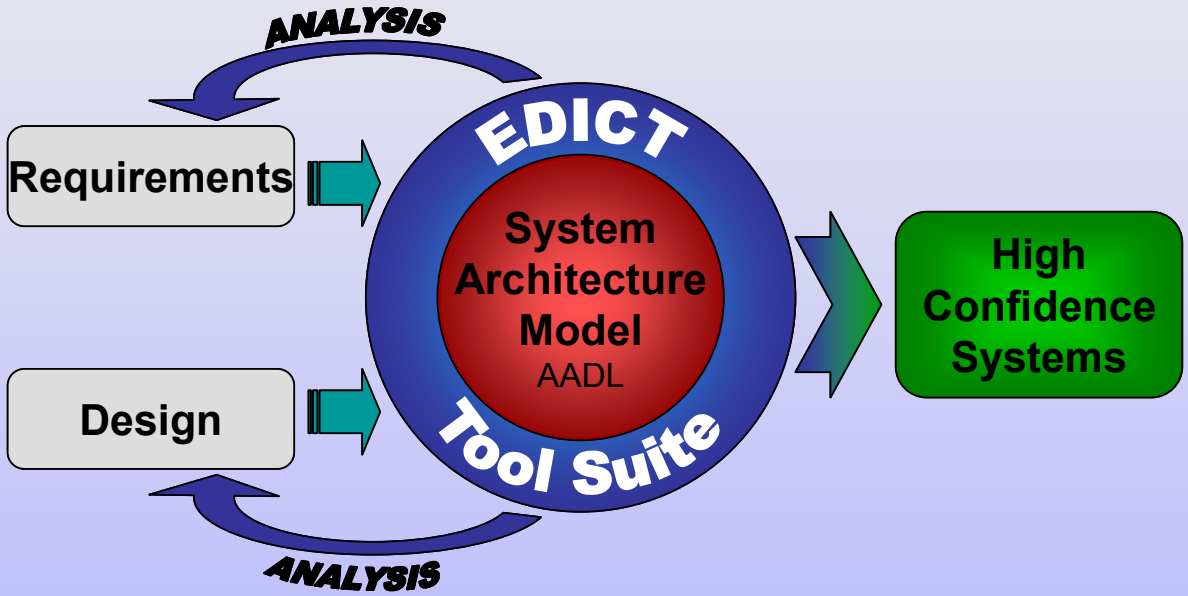
Analysis of MILS

- In analyzing MILS architectures we utilize a component-based analysis approach.
- In general, a good specification of a system component has two characteristics. [Van fleet, et al]
    1. It can be mapped to concrete component implementations using convenient and reliable methods. Such an approach enables the specification of a particular system component to be proved.
    2. A good specification encapsulates needed behavior so that the larger system can benefit from an assurance that the specification holds of the component. That is, the specification can be used in the larger system that contains the component about which the specification has been proved.

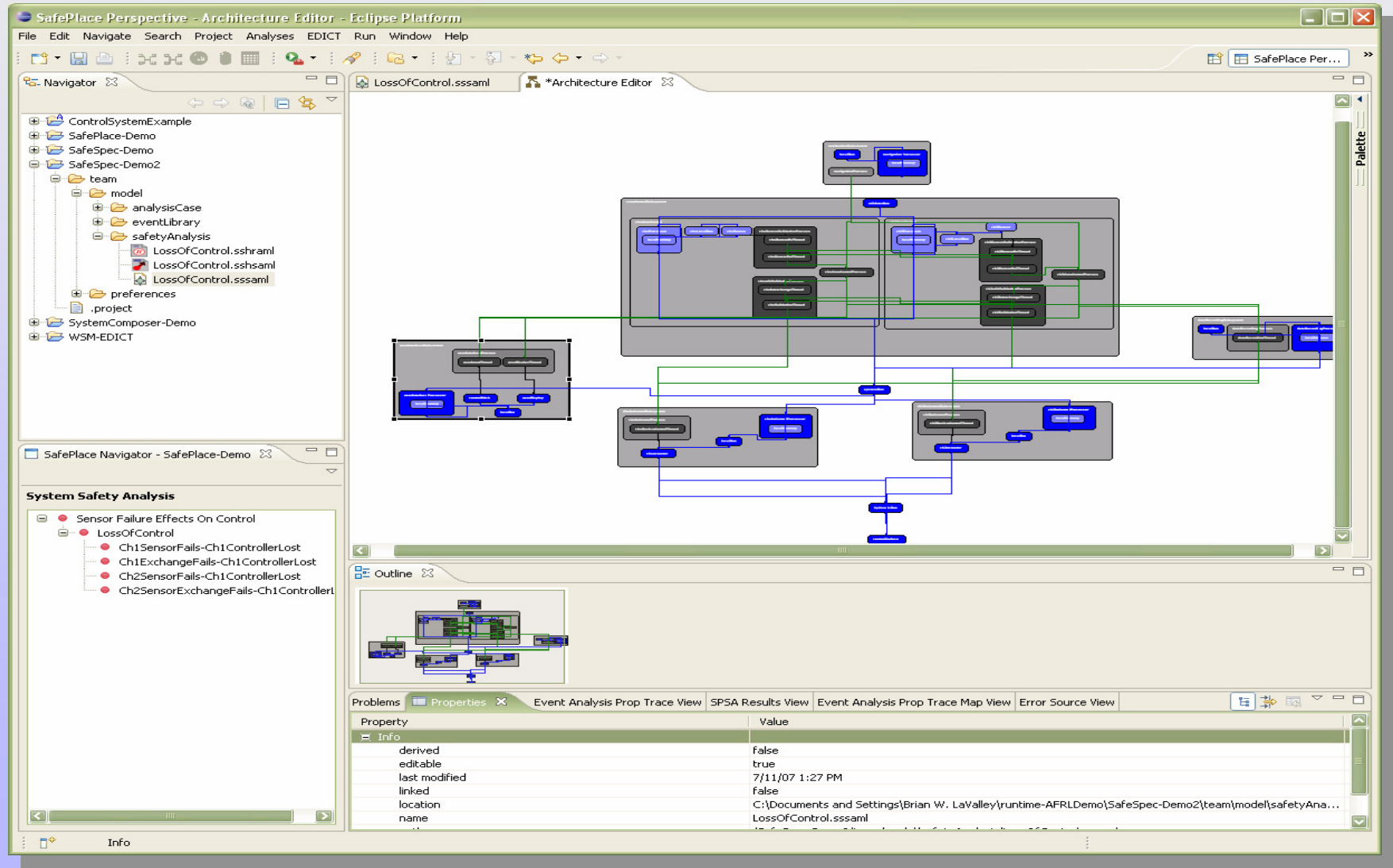
Information Flows

- Understanding the information flows in a MILS system is very important.
  - analysis of inter-component relationships
  - establishing robust partitions
  - understanding impact of errors on security mechanisms and system integrity
- Certain aspects are analogous to the analysis of error propagation that are available in current version of EDICT
  - these methods need to be adapted to model more general information flows

WWTG's EDICT Tool Suite

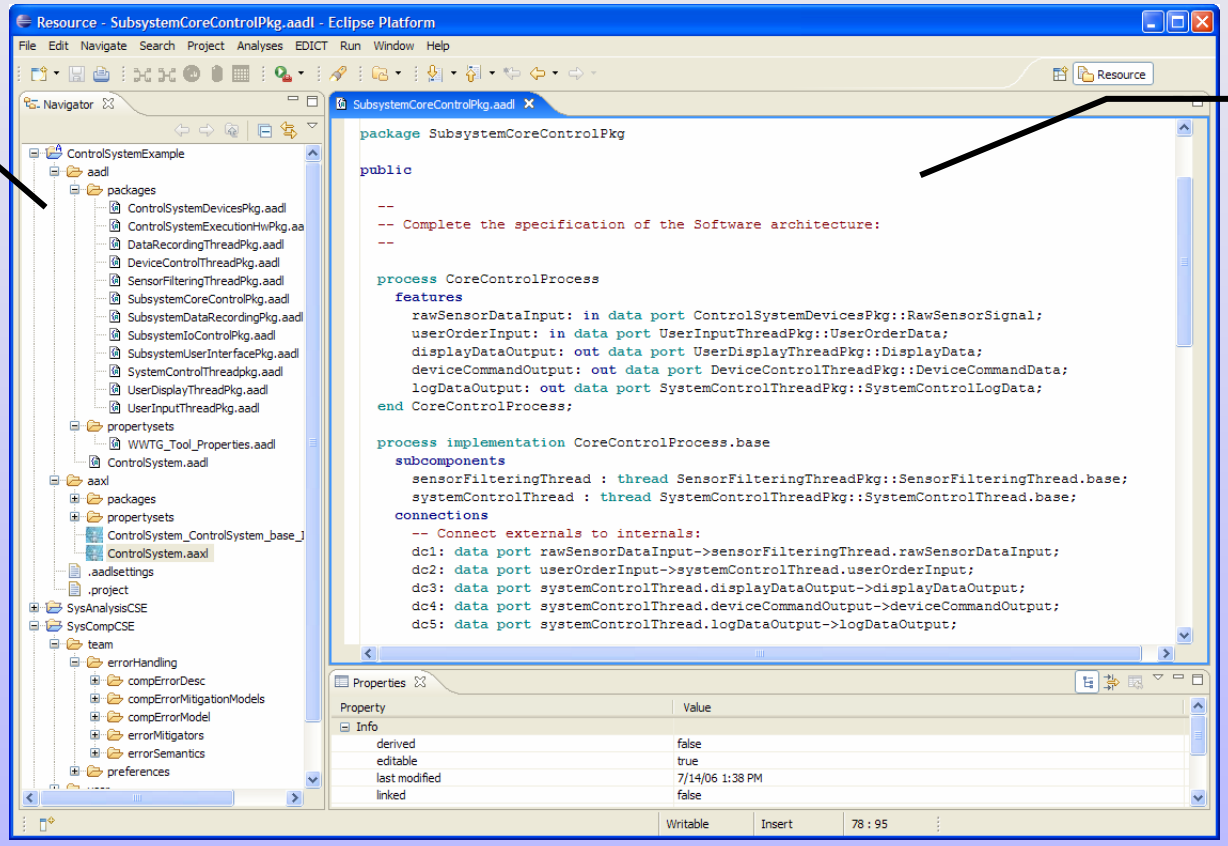


EDICT Tool Suite is Eclipse Based



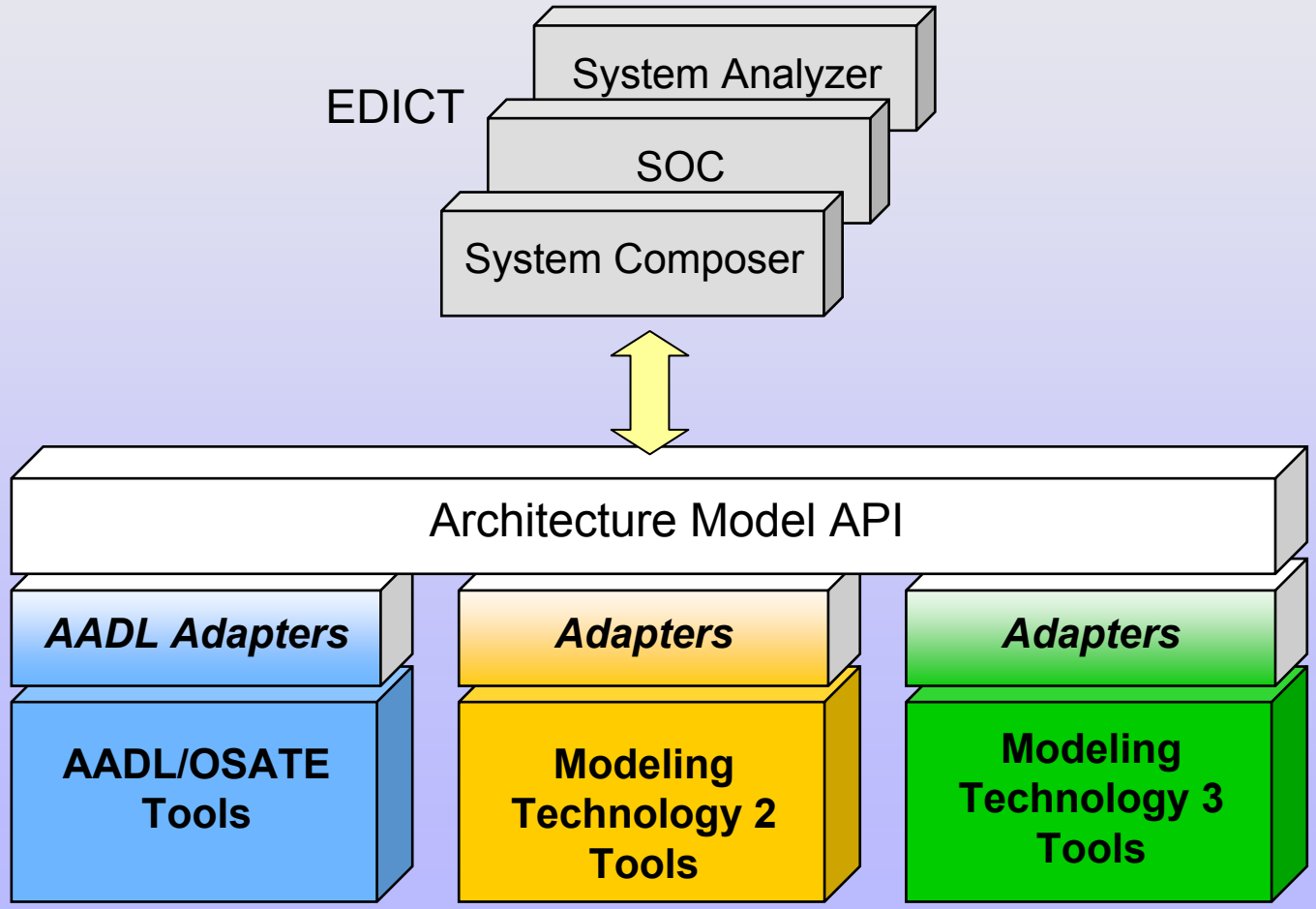
The OSATE tool provides a development environment for AADL

System architecture model information is stored in OSATE project structure.

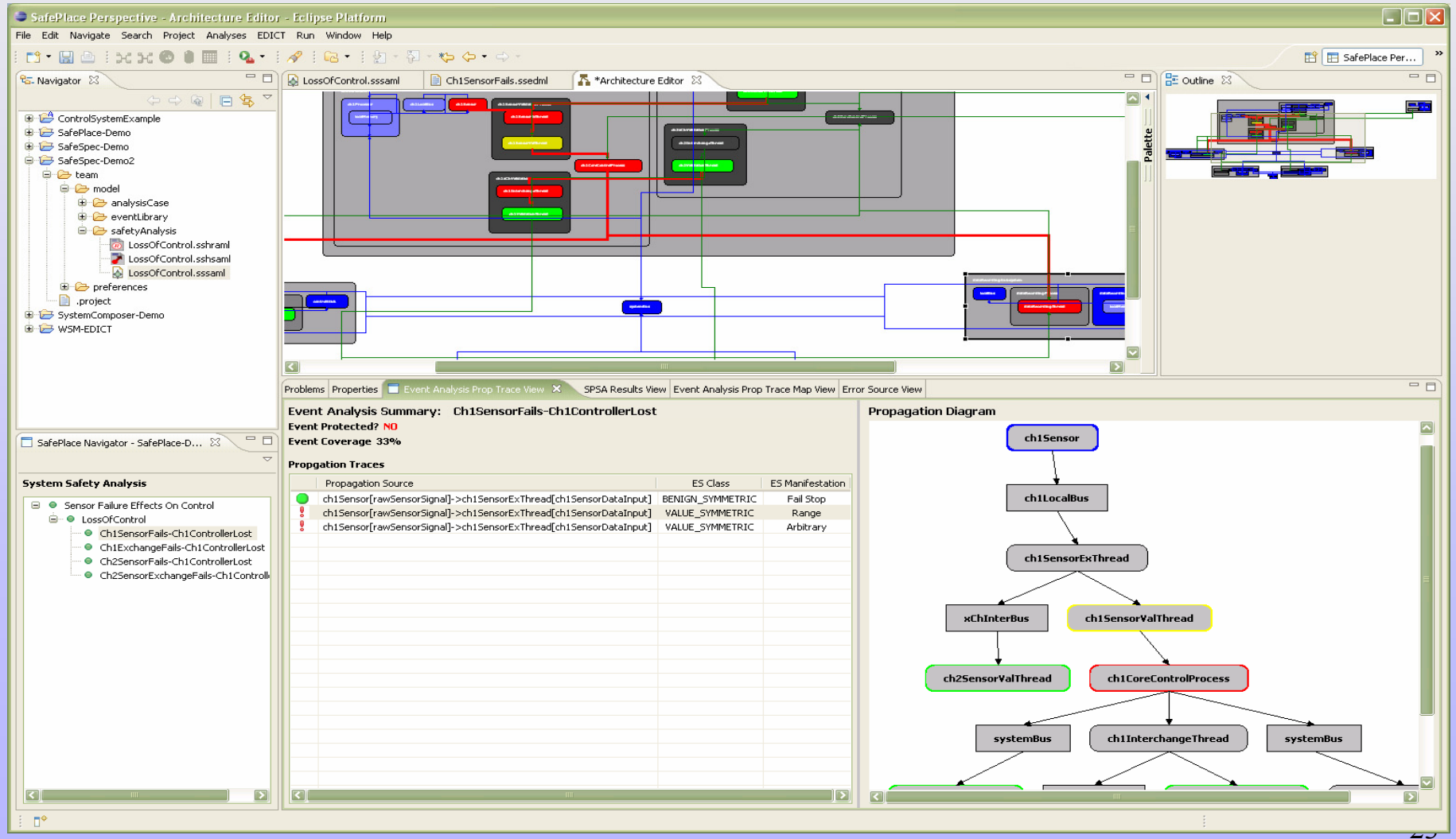


AADL editor and underlying compiler capabilities facilitate AADL specification of system architecture composition and component properties.

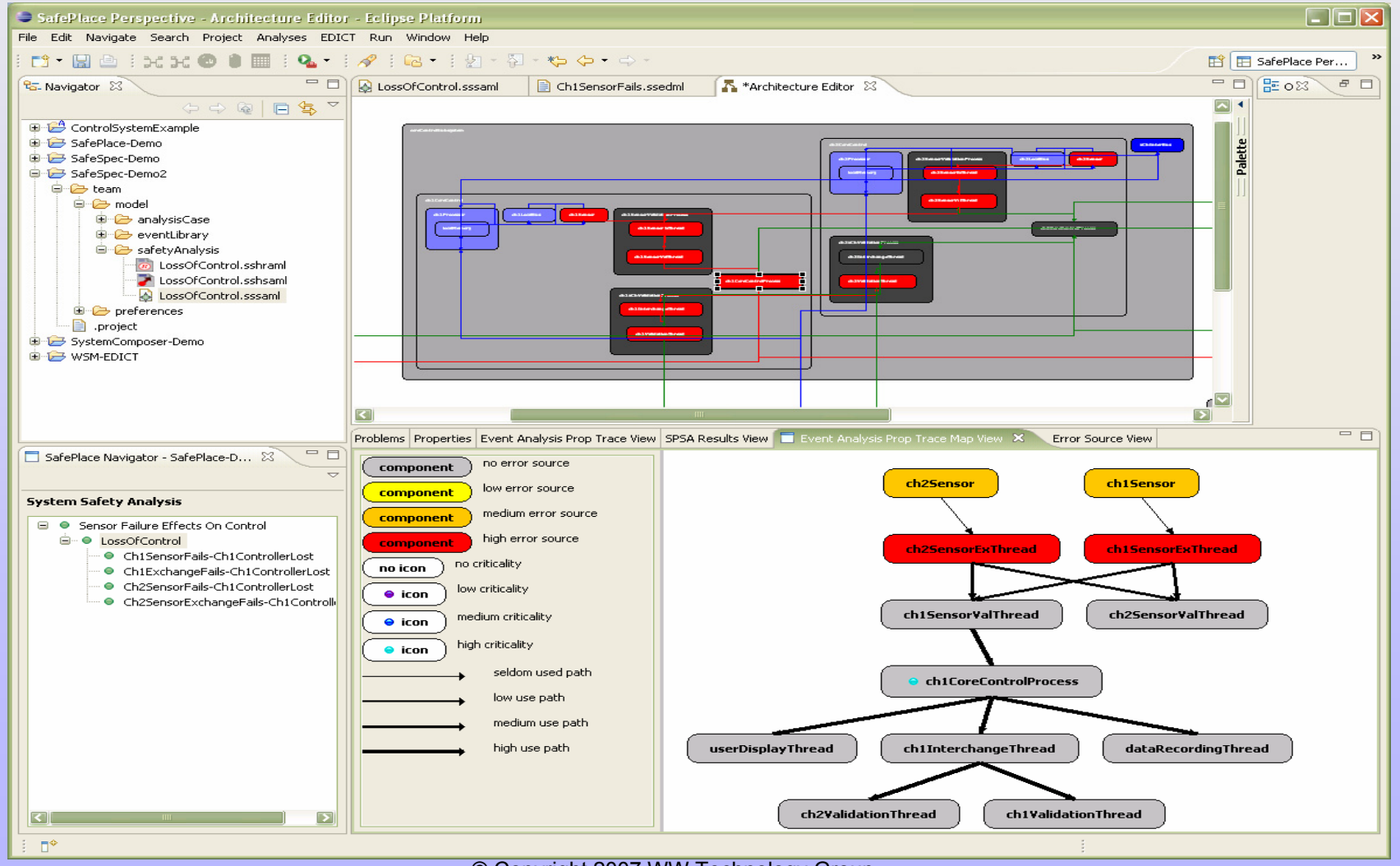
Architecture Model API and Adapter Framework Insulate Tools from Technology Specific Descriptive Modeling Solutions



EDICT Evaluates Error Propagation And Impacts



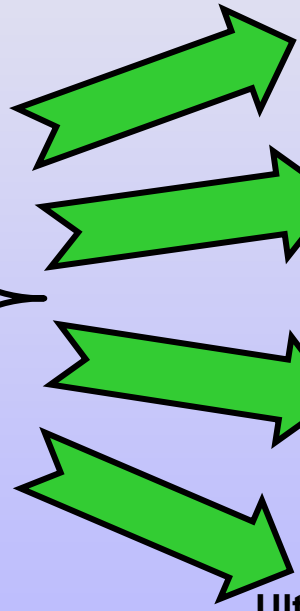
EDICT Provides Metrics And Visualizations To Aid In Run-time Mitigator Placement



Design For Certification Applies To A Wide Range Of Systems

Complementary Techniques

- Safety
- Dependability
- Complexity/Risk
- Certifiability

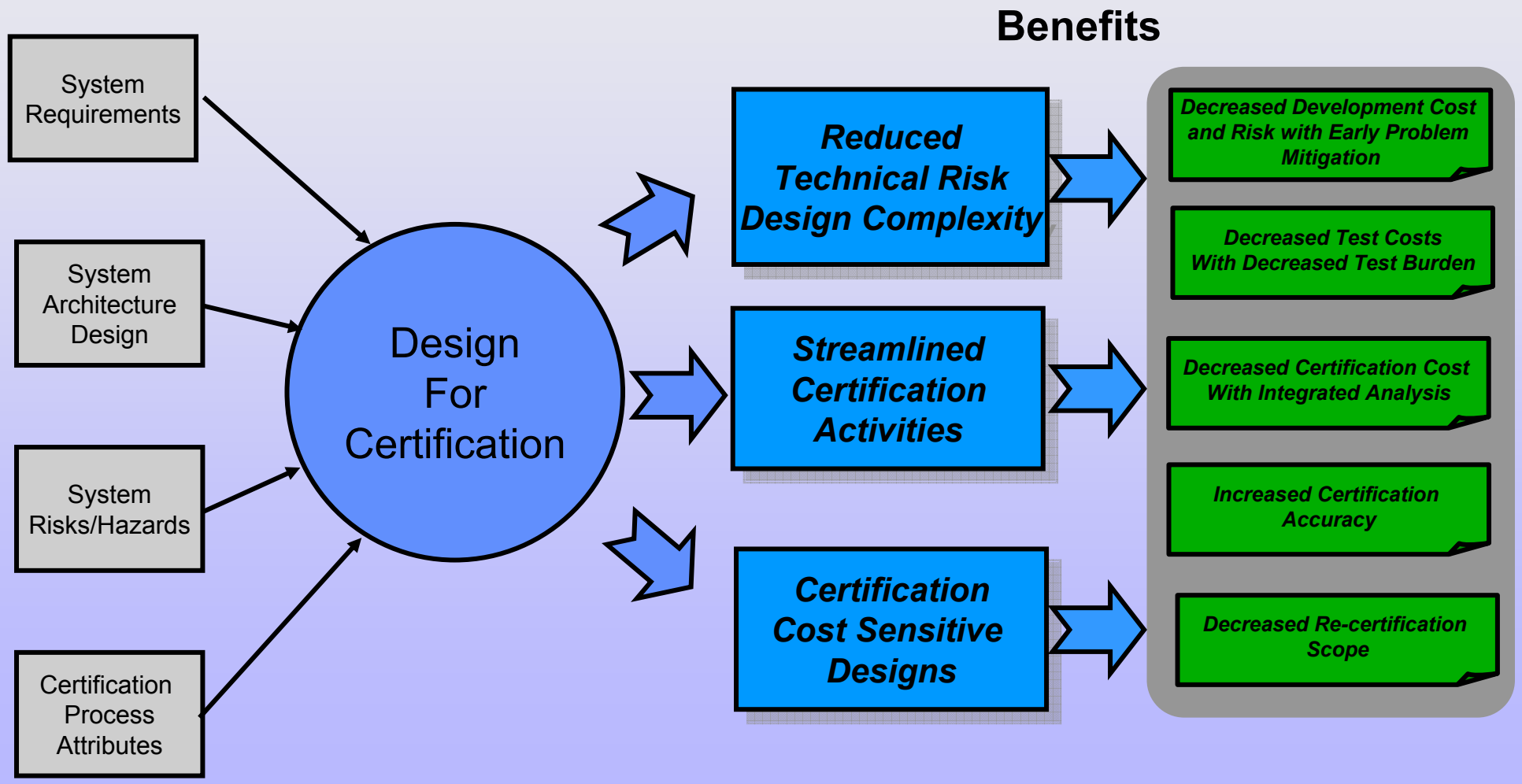


High Confidence System Categories

Mission Critical	High Availability Soft Real-time
System Critical	Safety Critical Fail Safe
Dependable	Safety Critical High Reliability Fail Operational
Ultra-Dependable	Safety Critical High Reliability Fail Operational

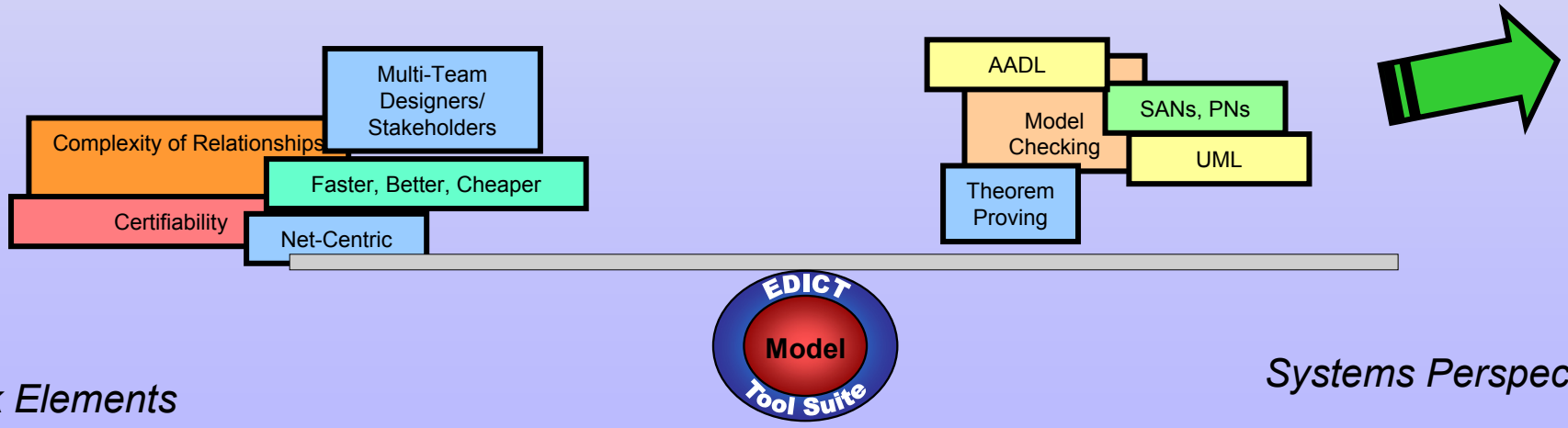
A Tailored Set of Techniques Are Applied Based On The System Needs

Design For Certification Benefits



Design of Complex Safety Critical Systems

- Challenging task where the complexities and nuances of a design can have significant impacts
  - Difficult to uncover problematic relationships
  - Faster, better, cheaper is (by itself) a negative pressure that can induce more design errors
  - Net-centric designs expose error propagation paths throughout the system (internal/external)

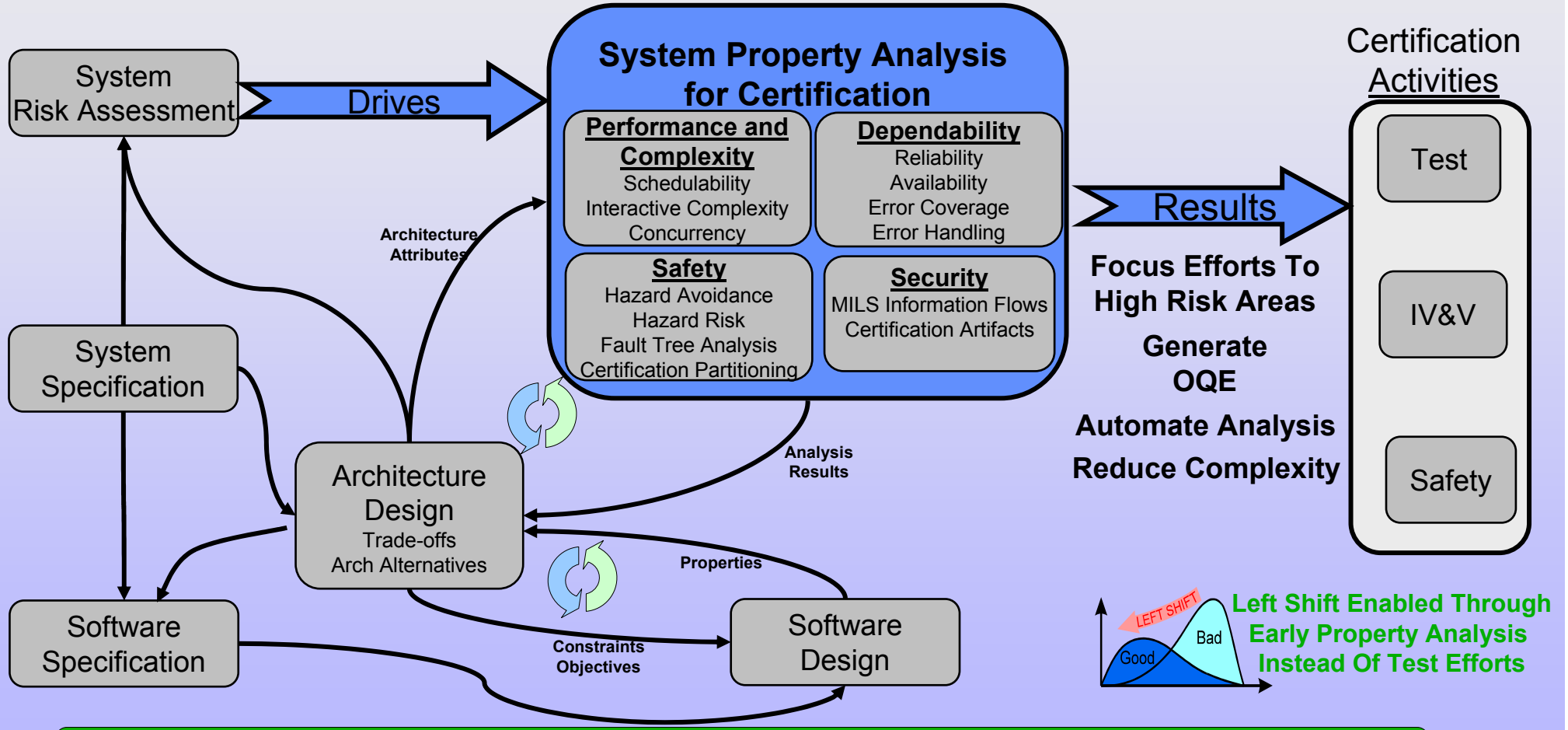


Risk Elements

Systems Perspective

Need to Extend Leverage of Modeling & Analysis Methods to Balance Challenges

Analysis Tools Provide For Early An Incremental Analysis That Build Confidence and Directs Certification



Increase Certification Efficiency By Focusing Efforts To Highest Risk Areas In System

QUESTIONS?

Dr. Chris J. Walter
cwalter@wwtechnology.com
410-418-4353