EDICT-IA
Tools for Model-Based Information Assurance Analysis

Mr. Brian LaValley
blavalley@wwtechgroup.com
(401) 348 - 8525

December 2012
Security Challenges for Distributed Systems-of-Systems

Increasing integration of sensitive information systems coupled with rapid technology development is generating systems of expanding size and complexity.

- Increased information management security risks with higher levels of integration
- Information Assurance certification/recertification is a costly, burdensome process
- The lack of domain specific and common representations for IA concerns places large burdens on developers and program managers
- Rapid technology development and expanding system size and complexity renders current design and certification methods untenable

New methods for Design and Certification of Information Assurance intensive systems are required to meet these challenges
Property Based Information Assurance

Goals

- Reduce development, certification and accreditation costs
  - Break problems into manageable pieces
  - Analyze pieces independently
  - Provide a structured method for synthesis of pieces into a system that supports incremental certification/re-certification
- Support top-down specification/decomposition of problem
  - Establish critical property requirements based on system needs
  - Flow down properties to drive system architecture and component level selection
- Support bottom-up certification chains for full traceability
  - Flow up component level IA properties to system architecture and requirements
    - Integrate stand-alone component certifications (CC) into a system context
  - Provide property traceability for certification evaluation
IA Property Hierarchy

- A structured set of properties that support the composition and analysis of information assurance
- Three Tiers of properties
  - **Application**
    - Functional application level aspects for IA
  - **Architecture**
    - Structural system architecture aspects that support application
  - **Infrastructure**
    - Component aspects that support architecture
- Properties can be flowed down for refinement and requirement allocation
- Properties can be traced upward for certification and analysis
Property Based Information Assurance

Stakeholder Views

Modeling and Analysis

Certification

System Architect

System Security Engineer

System Certifier

Traceability

Abstraction

Evidence Assurance Cases

Analysis Results

Architecture Data

Test Results Documents

External Sources

EDICT Tool Suite

Eclipse Open Tool Framework

XMI/XML EMF

XML Based Model and Results Storage

Tool Integration - OSATE - TOPCASED

Stakeholder Views

Property Based Analysis

Information Domains
Application Properties

Logical Architecture
Information Flow / Partitioning
Architectural Properties

Property Based Analysis

System Architecture
Component Properties

Property Based Analysis

System Architecture
Component Properties
Our Perspective is Engineering Driven

- WWTG’s background is in mission critical system architectures
  - Not simply a language/tool specific approach
- Each language or analysis technique has its own strengths and weaknesses but complex system architectures must leverage all areas
- Makes our approach scalable - focus on integration of modeling technologies and tools
  - Pick the right representation for each aspect of the job

Utilize Complementary and Consistent Models and Analysis Techniques
Architecture Modeling

Architecture models are used to define the system under analysis
EDICT Architecture Framework Eases Integration of Different Modeling Technologies

- With a known common architecture representation, augmenting models and analyzers can be built once.
- Effort to integrate different modeling technologies is shifted out of core EDICT tools and into translation software.
- When a new source of architecture information is integrated, no change is required to any supporting EDICT modeling and analysis features… this greatly reduces time and cost!
System Architecture Models

- Hierarchical models define the system to the computing platform

**Systems** ➔ Subsystem decomposition

**External Interface** ➔ Users – Devices – Systems

**Hardware Architecture** ➔ Memory – Processors - Buses – Access

**Software Architecture** ➔ Processes – Threads – Data – Subprograms – Connections
Architecture Flows Are Used To Model Data and Control Flows
IA Architecture Defines Partitioning and IA Mechanisms

- Define the hierarchy of Sites, Enclaves and Partitions
- Allocate Components to Partitions
- Allocate Group-Level Mechanisms
- Allocate Mechanisms to communication path ports
- Allocate Logical Interfaces to Communication Paths
Diverse IA Services are Defined by IA Mechanisms

- IA Mechanisms are used to model the IA functions that are provided by the architecture
- These include
  - Partitioning
  - Encoding
  - Access control
  - Cross-domain
  - Redundancy
  - Physical security
- Mechanism sets are used to combine mechanisms into integrated IA services
Users can create new mechanisms to fit needs
Each mechanism type has a dedicated editor that allows users to define the attributes of the mechanism
IA Requirements models enable the specification of IA related requirements:
- Information domain model specified separation and allowed interfaces
- Domain attributes specify required IA properties
IA Requirements Modeling

- Define the IA Domains, as well as domains associated with Actors and Externals
- Define the connectivity between domains
- Associate IA properties and attributes with domains
- JAFAN certification models enables domain properties to be specified using JAFAN nomenclature
- Each domain can have a unique characterization
  - Eliminates “float high” effect of certification requirements for a system
Domain Traceability

Property Based Information Assurance

Provides direct relationships between levels of abstraction
Ensures IA Requirements model address all functions and data flows
IA Domain to Architecture Traceability

- IA Requirements elements
  - Information Domains
  - Cross Domain Interfaces
  are traced to the architecture models through a traceability model and editor
- Components can be allocated to domains to support traceability of processing
- Cross Domain Interfaces are defined by allocating:
  - Interfaces – direct interfaces between components that are allocated to
  - Flows – flows that cross domain boundaries must also be allocated
- Model verifier is used to ensure that all elements are fully traced
  - Ensures all components are allocated
  - Ensure all data paths are accounted for
Property Based Analysis

Architectural Analysis used to verify that IA properties hold across the architecture
## IA Property Analysis Matrix

<table>
<thead>
<tr>
<th>Application Property</th>
<th>Integrity</th>
<th>Confidentiality</th>
<th>Access Control</th>
<th>Availability</th>
<th>Non-Repudiation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture Property</td>
<td>IA Architecture Analysis Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information Flow</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flow Confidentiality</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Flow Integrity Protection Strength</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Identification and Authentication</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Security Threshold Protection</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Persisted Data Utilization</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Data Isolation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Domain Partitioning</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Mixed Security Partitioning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Partition Integrity Strength</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Persisted Data Confidentiality</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PartitionConfidentiality Strength</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Mixed Security Designation</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Damage Limitation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Persisted Data Containment</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Persisted Data Hot-Spot</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Persisted Data Integrity</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Persisted Data Criticality</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Least Privilege</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Persisted Data Access</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boundary Flow Tracing</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Boundary Interface Allocation</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
IA Analyzers

- Each of the architectural analysis methods has a dedicated analyzer
- Each of these provides configuration options and results display
IA Application Property Analysis Summary

- Provides access to a summary of all error and warning conditions uncovered by analyzers for all IA Application Properties.
- Arranges analyzer results by IA property.
- Strives to narrow the focus to a particular problem area to avoid overwhelming the user.
- Short-cut mechanisms are provided to facilitate navigation from the problem area to the analyzer the generated the individual concerns.
Certification and Reporting

Support generation of evidence from the models and analysis methods
Organize and monitor evidence in context of certification process
JAFAN Compliance Modeling Analysis and Reporting

EDICT Certification

Specification Features
- JAFAN 6/3 Views - Management

Compliance Features
- JAFAN 6/3 Views - Analyzers - Management

Common IA Specs

IA Requirements

IA Architecture

IA Deployment

Traceability

IA Analysis Results

External Sources

Reports
- Assurance Cases
- Certification Artifacts

Test Results
- Certification Results
- Documentation/Plans
JAFAN Certification Package Editor
Feature Set Tab - Evidence

- Feature Assurance Sets
- Available Evidence updated to include system architecture data
- User defined external evidence
- Assurance Case
- Selected Evidence Summary Table

WW Technology Group
© Copyright 2012 All rights reserved.
Assurance Case Evidence Also Updated For System Architecture Data

- Solution elements can have evidence associated with them
- There are several possible sources
  - Analyzers
  - Verifiers
  - Reports
  - External Data
- Status of evidence is updated live
Wrap up

- Increased reliance on integrated distributed systems poses significant challenges for assurance of desired system properties.
- The need to be innovation in the areas of specification, modeling techniques, analysis methods to represent information assurance aspects.
- The EDICT IA tools begin to address these challenges through:
  - A structured, property-based approach to evaluating system architecture for information assurance.
  - A framework that supports traceability and decomposition from requirements down to detailed design models.
  - A suite of tools that implement the framework using open tool platforms and modeling languages.
  - Generation of certification evidence from model-based analysis for JAFAN compliance.
  - A suite of analysis tools that can evaluate system architectures for Confidentiality, Integrity and Access Control properties.
Questions?