“OMG: Not your Father’s CORBA Organization Any Longer”

The OMG System Assurance Task Force’s SwA Ecosystem

Dr. Ben Calloni, P.E. CISSP, OCRES-AP

Lockheed Martin Fellow, Software Security

Lockheed Martin Aeronautics Company, FTW

OMG Board of Directors

Chair OMG System Assurance Task Force

The Open Group, former Vice Chair, Board of Directors

SMU Adjunct Professor, System Security Engineering

OMG
OBJECT MANAGEMENT GROUP

Who Is OMG?

Object Management Group (OMG) factoids:

- Founded in 1989
- Over 470 member companies
- The largest and longest standing not-for-profit, open-membership consortium which develops and maintains computer industry specifications.
- Continuously evolving to remain current while retaining a position of thought leadership.

OMG's Best-Known Successes

Common Object Request Broker Architecture

- CORBA® remains the only language- and platform-neutral interoperability standard

Unified Modeling Language

- UML™ remains the world's only standardized modeling language

Business Process Modeling Notation

- BPMN™ provides businesses with the capability of understanding their internal business procedures

Common Warehouse Metamodel

- CWM™, the integration of the last two data warehousing initiatives

Meta-Object Facility

- MOF™, the repository standard

XML Metadata Interchange

- XMI™, the XML-UML standard

Who Are OMG-ers?

Some of the hundreds of member companies;

ACORD	**Deloitte**	**Mega Practical**	**Progress**
Atego	**Fujitsu**	**MetaStorm**	**Red Hat**
BAE Systems	**General Dynamics**	**Microsoft**	**SAP**
Boeing	**HP/EDS**	**Navy UWC & SWC**	**Selex**
CA	**Harris**	**NEC Sphere**	**Software AG**
Capgemini	**Hitachi**	**Northrop Grumman**	**Sopra**
Cordys	**HSBC**	**No Magic**	**Sparx Systems**
CSC	**IBM**	**OIS**	**Tata**
DND Canada	**KDM Analytics**	**Oracle**	**Tibco**
FICO	**Lockheed Martin**	**Penn National**	**Vangent**

Liaison Relationships

OMG Organization

Architecture Board

Liaison SC
Object & Reference
  Model SC
Spec Mgt SC
MDA Users' SIG
Process
  Metamodels SIG
SOA SIG
IPR SC
Sustainability SIG
Architecture
  Ecosystems SIG
Business
  Architecture SIG

Platform TC

A & D PTF
ADM PTF
MARS PTF
SysA PTF
Agent PSIG
Data Distribution PSIG
Japan PSIG
Korea PSIG
Ontology PSIG
Telecoms PSIG

Domain TC

BMI DTF
C4I DTF
Finance DTF
Government DTF
Healthcare DTF
Life Sciences DTF
Mfg Tech & Ind. Systems DTF
Robotics DTF
S/W Based Comm DTF
Space DTF
Crisis Mgmt DSIG
Regulatory Compl. DSIG
SDO DSIG
Sys Eng DSIG

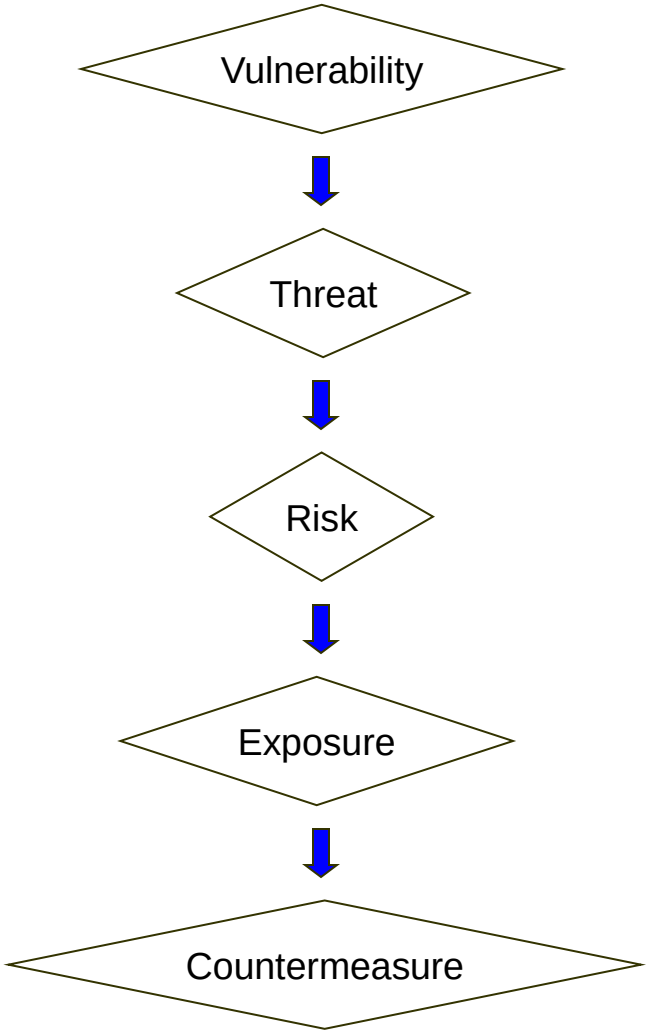
OMG System Assurance Task Force (SysA TF)

- Strategy
  - Establish a common framework for analysis and exchange of information related to system assurance and trustworthiness. This trustworthiness will assist in facilitating systems that better support Security, Safety, Software and Information Assurance

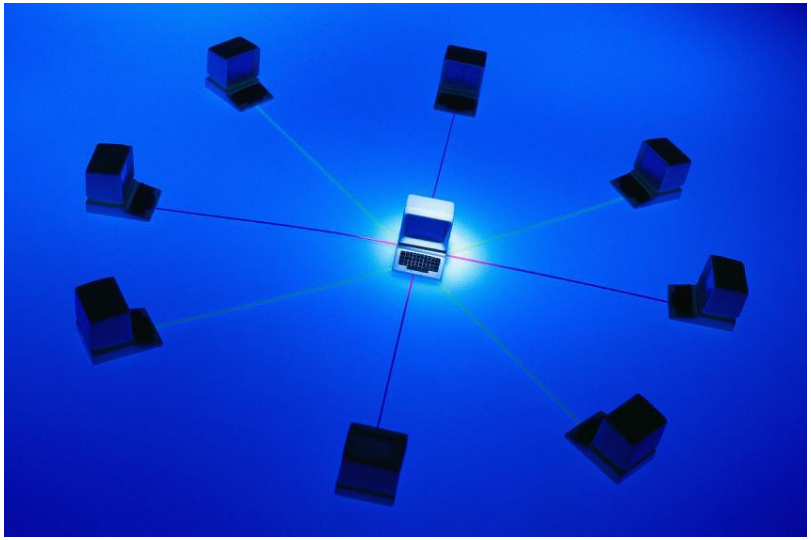
- Immediate focus of SysA TF is to complete work related to
  - SwA Ecosystem - **common framework for presenting and analyzing properties of system trustworthiness**
    - leverages and connects existing OMG specifications and identifies new specifications that need to be develop to complete framework
    - provides integrated tooling environment for different tool types
    - architected to improve software system analysis and achieve higher automation of risk analysis

Security Definitions$_1$(CISSP BoK)

- Reduction of “attack surface” important process within information security
  - Security management uses key definitions to identify areas of concern and how to protect them
    - Vulnerability
    - Threat
    - Risk
    - Exposure
    - Countermeasure



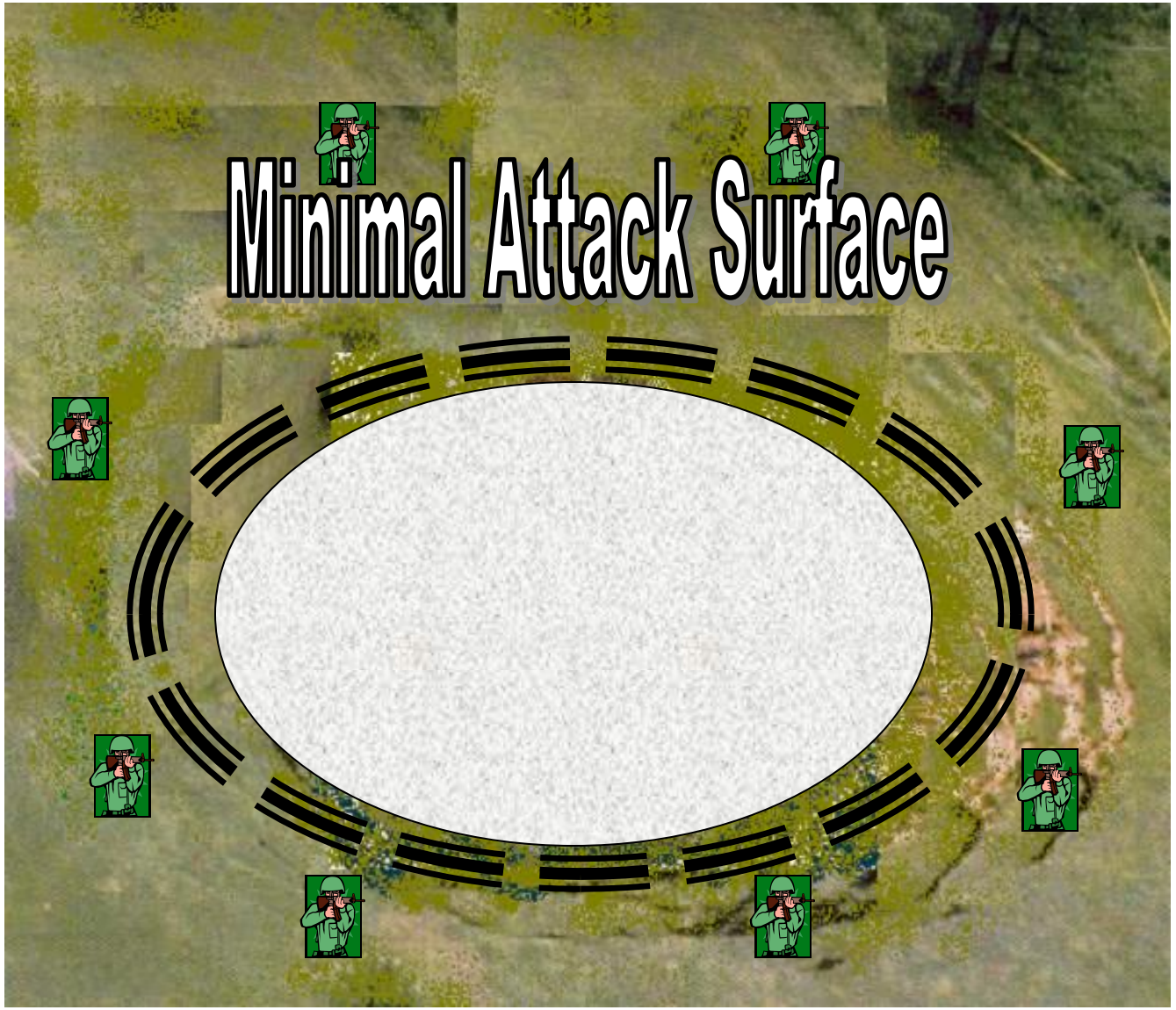
Security Definitions$_2$



- Attack surface
  - Area or parts of the system or network that are available to an assailant to compromise an environment
    - Might include multiple channels of entry
    - Services
    - Software
    - Physical Access
  - Fundamental objective in INFOSEC is constantly reducing the “attack surface” to better secure the environment

Extremely Robust Confidentiality (Low Risk)

Extremely Robust Confidentiality (Low Risk)



Provide Access (Availability)


The Microsoft Windows
"Attack Surface"

Security Definitions$_3$

- Vulnerability
  - A flaw
    - Implementation
    - Design
    - Requirements
  - Attacker access to the flaw, and
  - Attacker capability to exploit the flaw

Security Definitions$_4$

- Threats
  - Any potential hazard or harm to the data, systems or environment by leveraging a vulnerability
  - Individual taking advantage of a vulnerability is consider a threat agent

Security Definitions$_5$

- Risk
  - Risks are the probability of the threats using the vulnerabilities
  - Higher risks come with more vulnerabilities and increased threats

Security Definitions$_6$

- Exposure
  - The damage done through a threat taking advantage of a vulnerability
    - Examples of exposure
      - Data deletion or modification, the loss of integrity
      - Malicious code deployed in a private network and stealing sensitive customer information
      - Unauthorized viewing of private data
        - » SSN, Banking, Medical

Security Definitions$_7$

- Countermeasures (aka Controls)
  - Processes and standards that are used to combat and mitigate the risks
    - Examples
      - Keeping up-to-date on service packs, hotfixes
      - Maintaining current virus definitions
      - Hiring a security staff to monitor the facilities
      - Access control systems inside the operating systems
      - Biometric devices to provide higher assurance of authentication
      - Educating users on managing passwords and/or sensitive information
- Countermeasures are implemented only if they cost less than exposure (loss)!

Ford Pinto Cost Benefit Analysis
aka "The Bean Counters"

- Benefit (Estimated $49.5 Million)
  - 180 burn deaths, 180 serious burn injuries, 2,100 burned vehicles each year
  - $200,000 per death, $67,000 per injury, and $700 per vehicle
- Cost of recall ($137 Million)
  - Sales: 11 million cars, 1.5 million light trucks.
  - Unit Cost: $11 per car, $11 per truck.
- Lawsuit Damages:
  - Jury: $2.5M Compensatory: $125M Punitive
  - Judge: Reduced Punitive to $3.5M
  - Total Actual : $6.0M

From a Risk Analysis POV Ford made the right decision!

- Ultimately, 27 people were determined to have been killed in rear-end-crash explosions involving Pintos

http://www.engineering.com/Library/ArticlesPage/tabid/85/articleType/ArticleView/articleId/166/Ford-Pinto.aspx

Delivering System Assurance:

Delivering System Predictability and Reducing Uncertainty

- Software Assurance (SwA) is 3 step process
  1. **Specify Assurance Case**
     - Enable supplier to make **bounded assurance claims** about safety, security and/or dependability of systems, product or services
  2. **Obtain Evidence for Assurance Case**
     - Perform software assurance assessment to justify claims of meeting a set of requirements through a structure of sub-claims, arguments, and supporting evidence
     - Collecting Evidence and verifying claims' compliance is complex and costly process
  3. **Use Assurance Case to calculate and mitigate risk**
     - Exam non compliant claims and their evidence to calculate risk and identify course of actions to mitigate it
     - Each stakeholder will have their own risk assessment – e.g. security, liability, performance, compliance

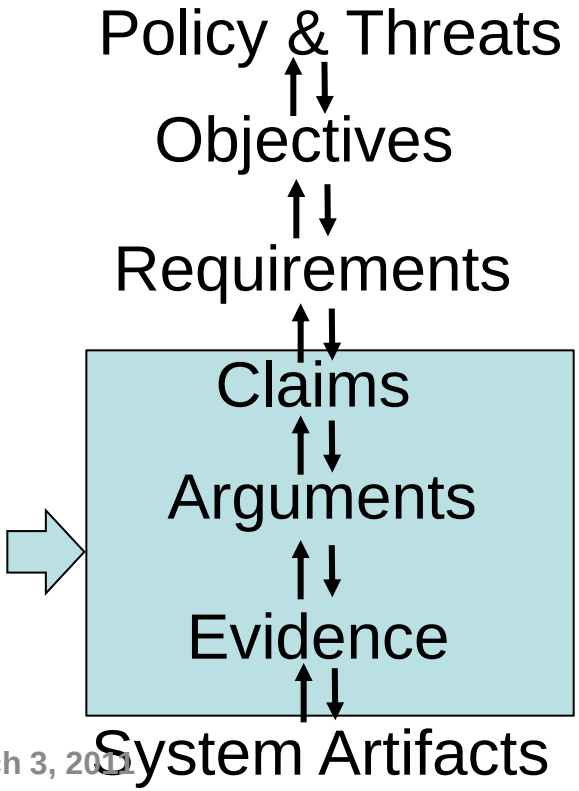
Currently, SwA 3 step process is informal, subjective & manual

Current Assessment Approaches - Limitations

- Lack of formalized methodology between high level policy claims and evidence means a laborious, unrepeatable (subjective), lengthy and costly certification process
- Current assessment approaches resist automation

Claim:
"It rained last night!"

Methodology Gap



Improving System Assessments:
Systematic, Objective and Automated

Key Requirements:

1. Specified assurance compliance points through formal specification
2. Transparency of software process & systems
3. End-to-end Traceability: *from code to models to evidence to arguments to security requirements to policy*
4. Standards based Integrated tooling environment

Together, these requirements enable the management of system knowledge and knowledge about properties, providing a high degree of transparency, traceability and automation

CC Assurance Requirements Example (Separation Kernel Protection Profile: EAL-6+)

6.1 Configuration Management (ACM)

6.1.1 CM Automation (ACM_AUT)

6.1.1.1 Complete CM Automation (ACM_AUT.2)

ACM_AUT.2.1D The developer shall use a CM system.

ACM_AUT.2.2D The developer shall provide a CM plan.

ACM_AUT.2.1C The CM system shall provide an automated means by which only authorized changes are made to the TOE implementation representation, and to all other configuration items.

ACM_AUT.2.2C The CM system shall provide an automated means to support the generation of the TOE.

ACM_AUT.2.3C The CM plan shall describe the automated tools used in the CM system.

ACM_AUT.2.4C The CM plan shall describe how the automated tools are used in the CM system.

ACM_AUT.2.5C The CM system shall provide an automated means to ascertain the changes between the TOE and its preceding version.

ACM_AUT.2.6C The CM system shall provide an automated means to identify all other configuration items that are affected by the modification of a given configuration item.

ACM_AUT.2.1E The evaluator shall confirm that the information provided meet all requirements for content and presentation of evidence.

The Software Assurance Ecosystem: Turning Challenge into Solution

- SwA Ecosystem is a formal framework for analysis and exchange of information related to software security and trustworthiness

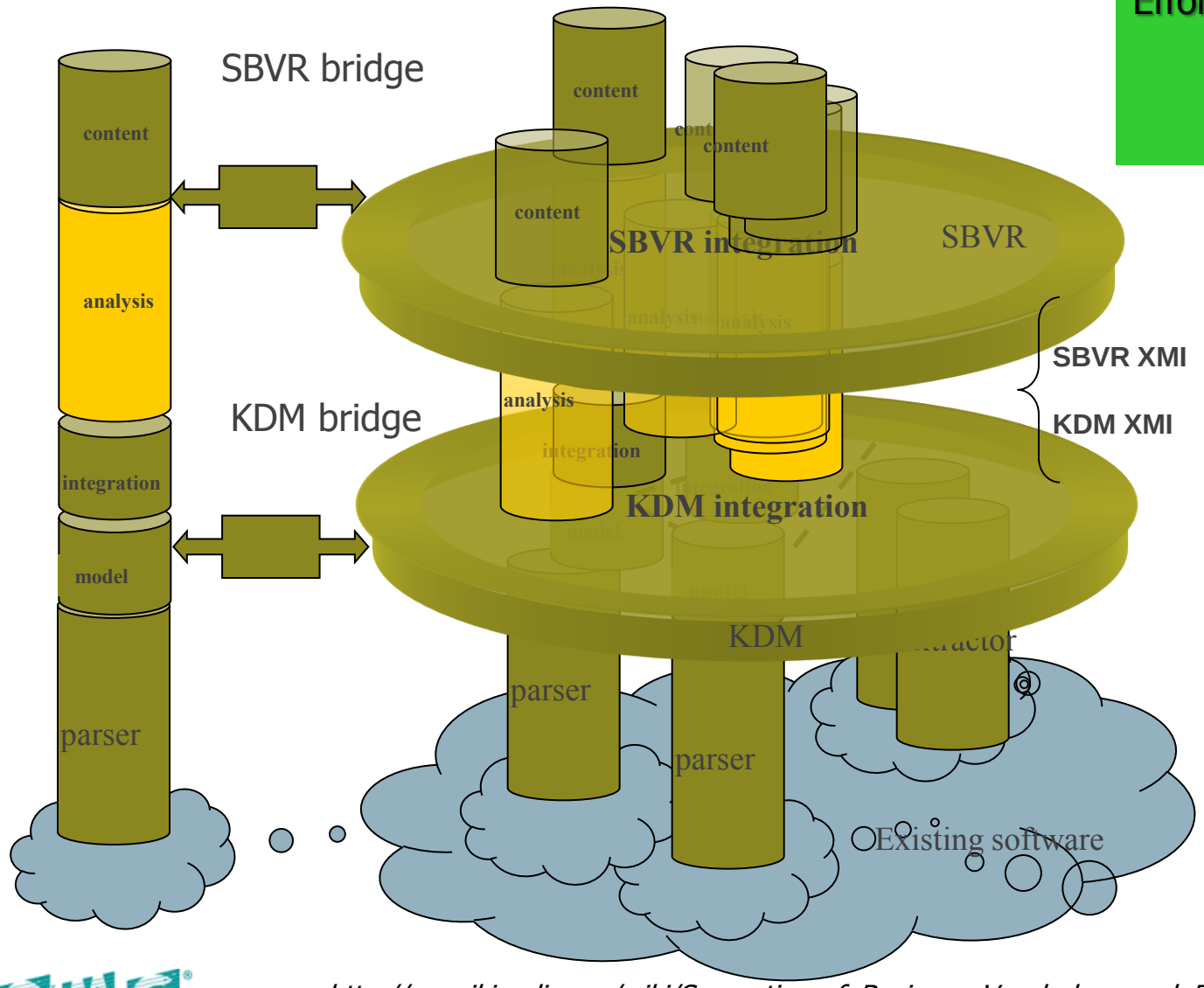
- Provides a technical environment where formalized claims, arguments and evidence can be brought together with formalized and abstracted software system representations to support high automation and high fidelity analysis.

- Based entirely on ISO/OMG Open Standards
  - Semantics of Business Vocabulary and Rules (SBVR)
  - Knowledge Discovery Metamodel (KDM)
  - Structure Metrics Metamodel (SMM)
  - Structured Assurance Case Metamodel (SACM) (Adopted June 2010)
    - Software Assurance Evidence Metamodel (SAEM)
    - Argumentation Metamodel (ARM)
- Architected with a focus on providing fundamental improvements in analysis

Benefits of SwA Eco System standard-based approach

Content is unlocked from "siloed" tools

Effort is focused at creating common, reusable content that can be used by multiple tools



- **Rules are higher-level than procedural or object-oriented code;**
- **Rules are better suited for the task of capturing assurance content;**
- **Natural-language interface to rules;**
- **Rules are written against standard ontology for software**
- **Software engineering content can be formalized independently**
- **Executable SBVR+KDM rules are queries to KDM repository that is populated by language-specific KDM knowledge extraction tools**

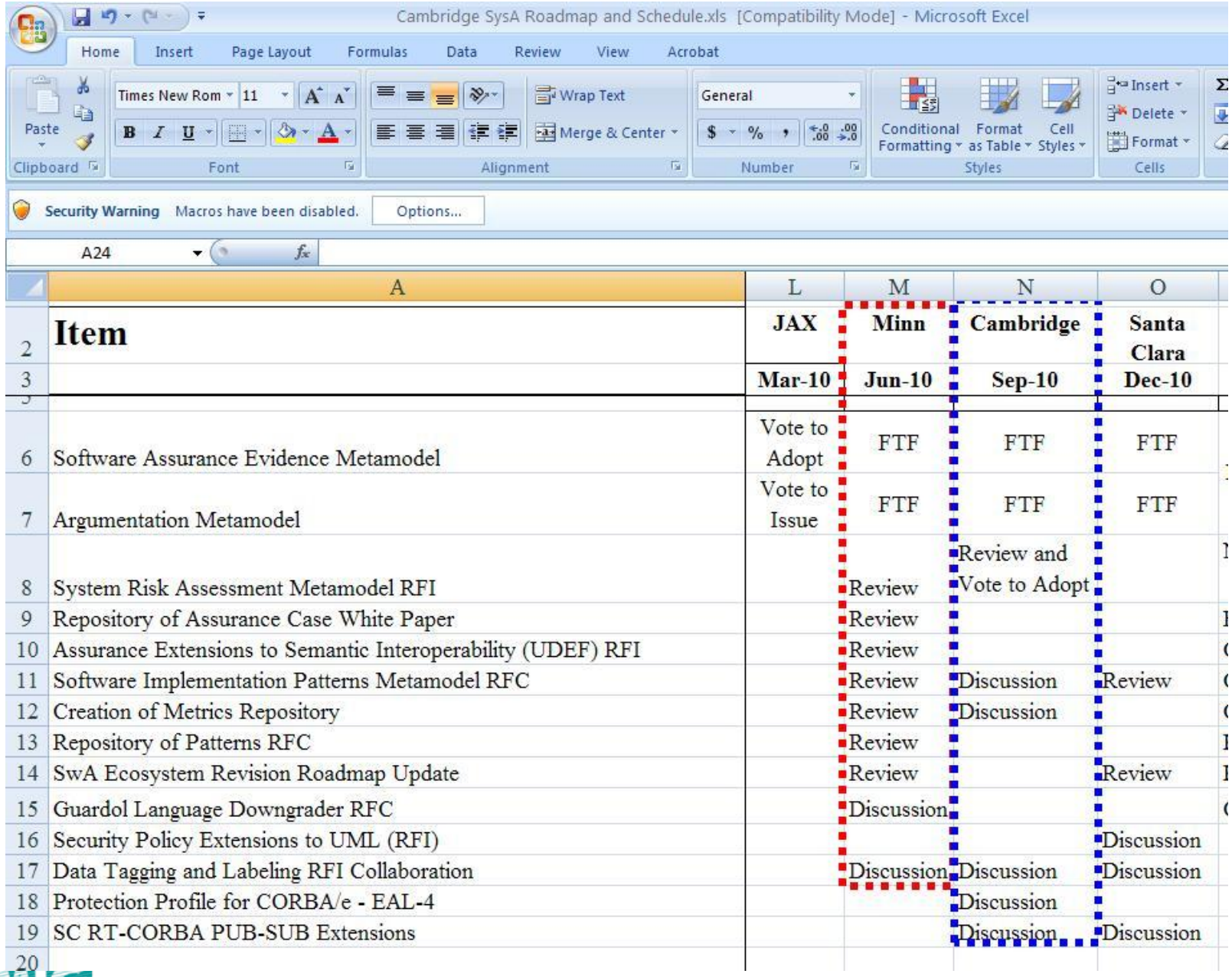
http://en.wikipedia.org/wiki/Semantics_of_Business_Vocabulary_and_Business_Rules

Leveraging what we already have through SwA Ecosystem

- Software Assurance Ecosystem enables industry and government to **leverage** and **connect** existing policies, practices, processes and tools, in an affordable and efficient manner

- The key enabler is the Software Assurance (SwA) Ecosystem **Infrastructure**
  - an open standard-based integrated tooling environment that dramatically reduces the cost of software assurance activities
    - Integrates different communities: Formal Methods, Assurance Case, Reverse Engineering and Static Analysis, and Dynamic Analysis for a System Assurance solution
    - Enables different tools to interoperate
    - Introduces many new vendors to ecosystem because they each leverage parts of the tool chain

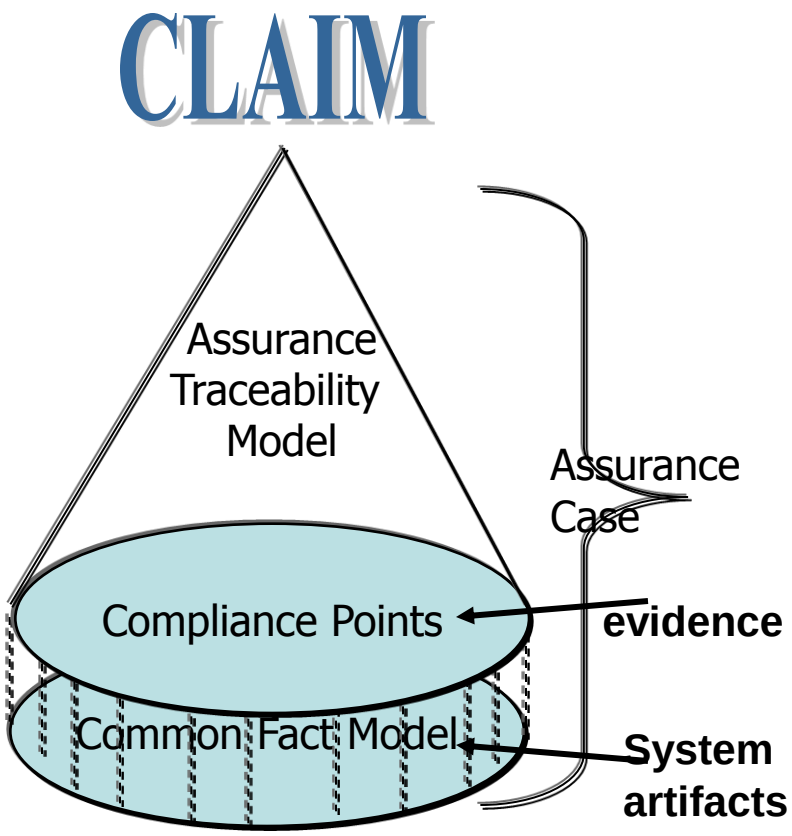
Where We are Going – Expanding SwA Ecosystem



	A	L	M	N	O
2	**Item**	**JAX**	**Minn**	**Cambridge**	**Santa Clara**
3		**Mar-10**	**Jun-10**	**Sep-10**	**Dec-10**
6	Software Assurance Evidence Metamodel	Vote to Adopt	FTF	FTF	FTF
7	Argumentation Metamodel	Vote to Issue	FTF	FTF	FTF
8	System Risk Assessment Metamodel RFI		Review	Review and Vote to Adopt	
9	Repository of Assurance Case White Paper		Review		
10	Assurance Extensions to Semantic Interoperability (UDEF) RFI		Review		
11	Software Implementation Patterns Metamodel RFC		Review	Discussion	Review
12	Creation of Metrics Repository		Review	Discussion	
13	Repository of Patterns RFC		Review		
14	SwA Ecosystem Revision Roadmap Update		Review		Review
15	Guardol Language Downgrader RFC		Discussion		
16	Security Policy Extensions to UML (RFI)				Discussion
17	Data Tagging and Labeling RFI Collaboration		Discussion	Discussion	Discussion
18	Protection Profile for CORBA/e - EAL-4			Discussion	
19	SC RT-CORBA PUB-SUB Extensions			Discussion	Discussion
20					

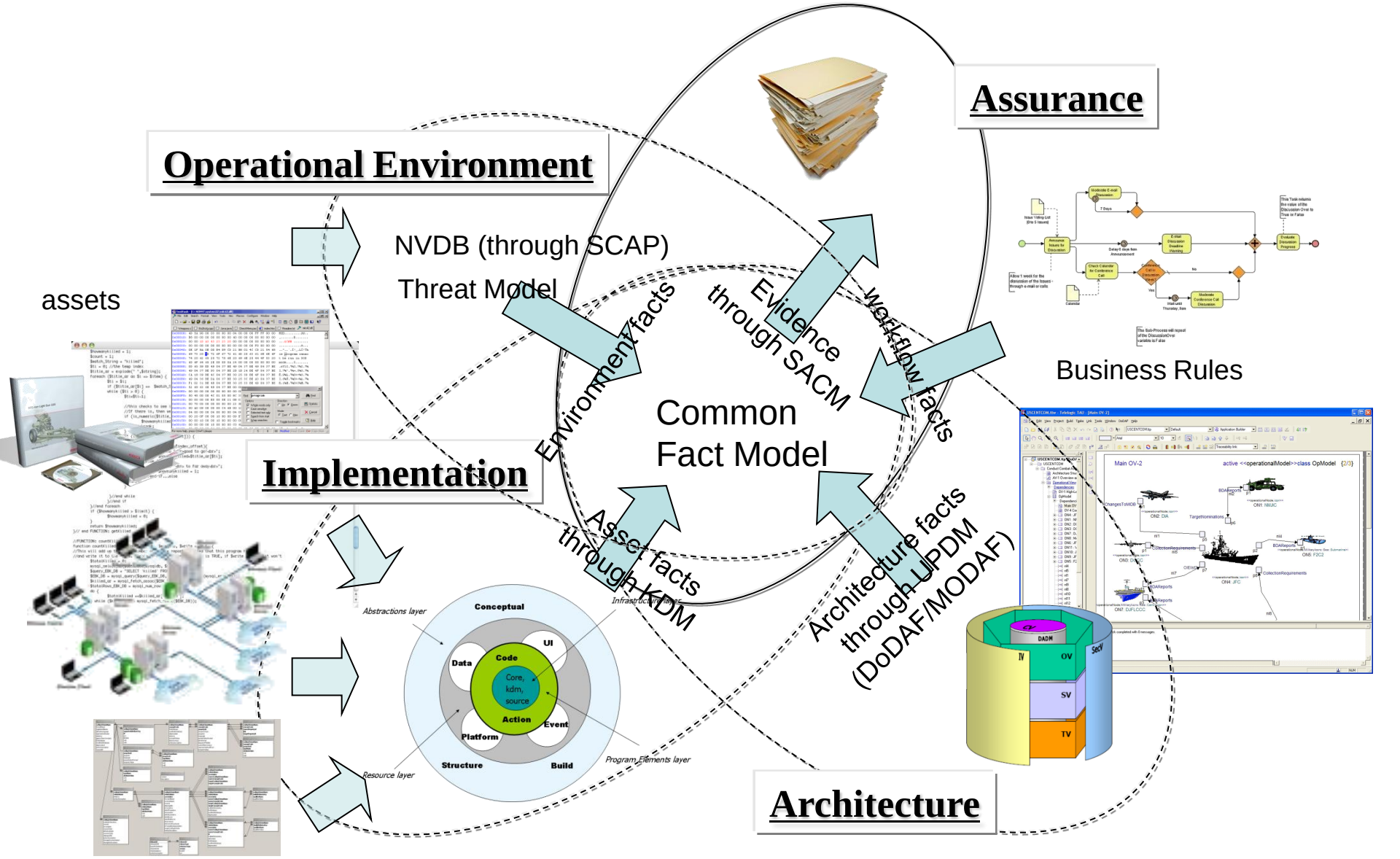
System Assurance Process: Achieving Automation

- Two key models in Assurance Case
  - Assurance Traceability Model (ATM)
    - Connects evidence to high level policy
  - Common Fact Model (CFM)
    - Connects system artifacts to evidence
- ATM defines *compliance points* that the system will be evaluated against
  - Set of formal knowledge models within System Security Domain
  - Defined at the lowest level of ATM
  - Serve as query to Common Fact Model
- CFM defines unified, precise and normalized System Model
  - Set of formal knowledge models within System Engineering Domain populated by artifacts from system under evaluation



Unified and formal knowledge models of the System Security and Engineering Domain are fundamental requirements for supporting and enhancing risk management approaches

Common Fact Model



Software Assurance Ecosystem: *The Formal Framework for System Assessments with Focus on Automation*

Tools Interoperability and Unified Reporting Environment

Process Docs & Artifacts

Requirements/Design Docs & Artifacts

Reports, Risk Analysis, etc

Process, People & Documentation Evaluation Environment

- Some point tools to assist evaluators but mainly manual work
- Claims in Formal SBVR vocabulary
- Evidence in Formal SBVR vocabulary
- Large scope requires large effort
- **Supported by The Open Group's UDEF***

Process, People, documentation Evidence

Formalized Specifications

Assurance Case Repository

- Formalized in SBVR vocabulary
- Automated verification of claims against evidence
- Highly automated and sophisticated risk assessments using transitive inter-evidence point relationships

Supported by the following standards:
- ISO/IEC 15026
- ISO/TC 37 / OMG SBVR
- OMG ARM
- OMG SAEM
- Software Fault Patterns (Target late 2011)
- UML Security Policy Extensions (planned)

Software System / Architecture Evaluation

- Many integrated & highly automated tools to assist evaluators
- Claims and Evidence in Formal vocabulary
- **Combination of tools and ISO/OMG standards**
- Standardized SW System Representation In KDM
- Large scope capable (system of systems)
- Iterative extraction and analysis for rules

Supported by ISO/IEC 19506

Software system Technical Evidence

Executable Specifications

SFP(CWE)

Hardware Environment

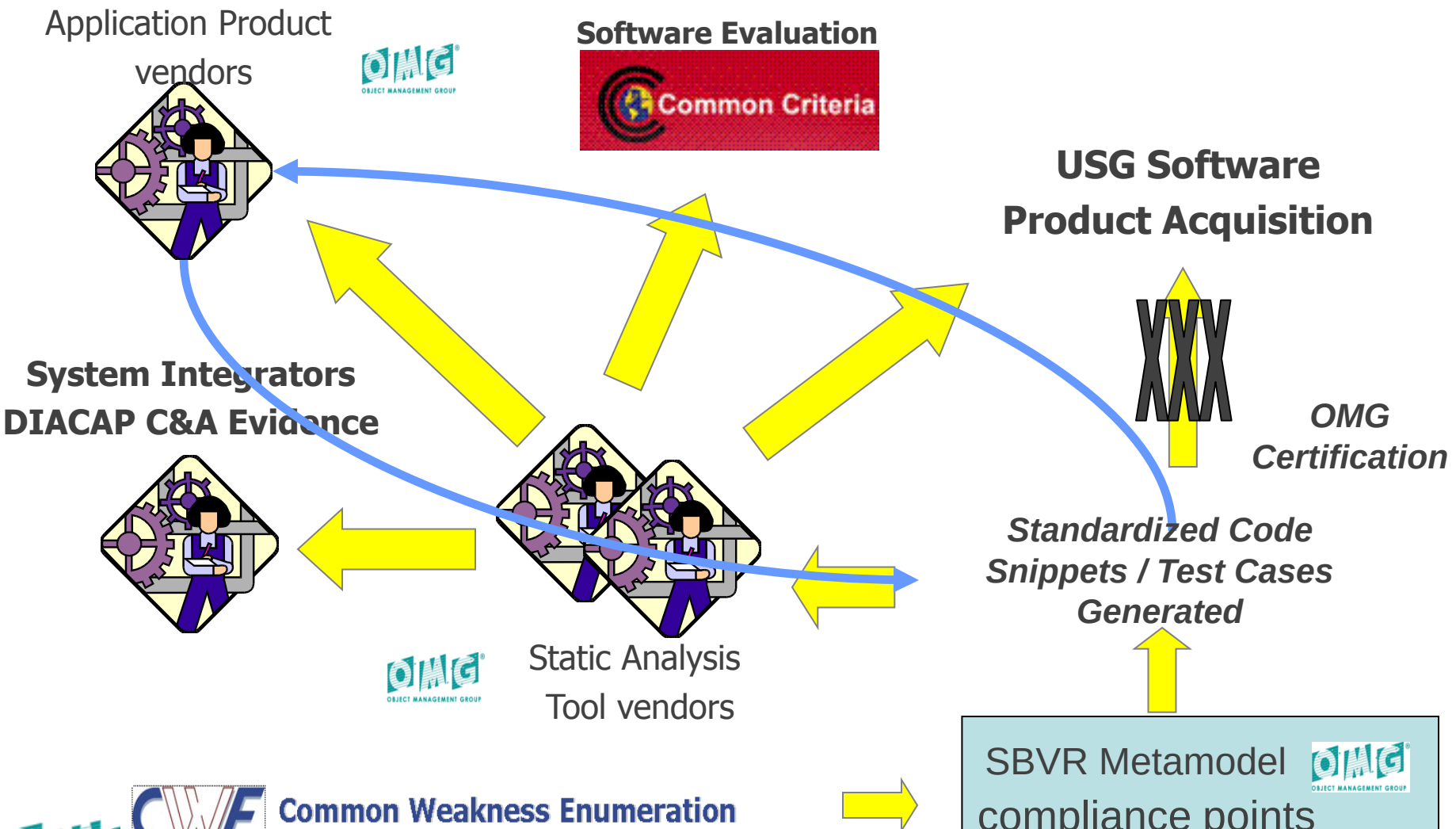
Software System Artifacts

Data Structures

IA Controls

Protection Profiles

SFP Component of Ecosystem in Standards Process and Tool Certification



Common Weakness Enumeration
A community-developed dictionary of common software weaknesses

Cyber Security

"Cyber Security is a 'National Security Crisis'. We have accepted the myth that software is too difficult and complex so we accept poor quality"

- John Gilligan, CEO The Gilligan Group
  - Former US Air Force CIO
  - DHS-OSD Software Assurance Forum
  - March 12, 2009

It is not <u>too</u> difficult but *<u>Assured Software</u>*** is hard and requires extra effort and diligence!**

Summary of the SwA Ecosystem Approach

- Normalized uniform common fact model
  - Separation of data feeds from reasoning
  - Standards-based
- Assurance case and SBVR
  - Representation of substantive reasoning
  - Natural language
- End-to-end multi-segment Traceability models
  - Code to state diagrams
  - Code to architecture
  - Code to conceptual model
  - Code to evidence determined by arguments
  - Evidence to arguments
  - Arguments to policy
- Focus on polynomial path-based properties
  - Instead of exponential state-based properties
- Arguments are “executable” queries to the fact model

Key Value of SwA Ecosystem Approach: End-to-end Traceability: *from code to models to evidence to arguments to security requirements to policy*

sysa-chair@omg.org