OMG's First Software Assurance Workshop:
Working Together for Confidence

Fairfax, VA USA – March 5-7, 2007

Workshop Program

MONDAY – March 5, 2007

Tutorials – 2 Tracks

0930 - 1200 ***Introduction to Software Security and Assurance Cases***
Track 1 Samuel T. Redwine, Jr., Associate Professor, James Madison University

Intended for participants without backgrounds in software security or assurance cases but with an existing understanding of software, this tutorial provides the basic background on software security and assurance cases needed to understand other tutorials and presentations in the Workshop. Participants will first learn about the software security problem and its elements including how software is in danger all its life, not just during operations. The tutorial will define security and describe its place within software quality and dependability. Basic terminology will be supplied. The tutorial will highlight important issues in secure software engineering including establishing uncertainties and consequences related to software security and the associated assurance case.
An assurance case is central to software security risk management and provides the grounds for the degree of justified confidence in security-related claims made regarding the software. General structures for assurance cases will be outlined and kinds of evidence enumerated. Attendees will receive the charts and latest version of the software security body of knowledge edited by the presenter for the US Departments of Homeland Security and Defense.

0930 - 1200 ***Security in the Software Life Cycle***
Track 2 Karen Mercedes Goertzel & Theodore Winograd, Booz Allen Hamilton

This tutorial will start by describing the current state of software in terms of its vulnerability to the increasing number and intensity of security threats, focusing on the aspects of software that make it vulnerable to those threats, and how those vulnerabilities are exploited - and moreover, why this problem is so critical. The presenters will then describe the security properties that characterize secure software - i.e., software that remains dependable in the face of attack - and will explore how the way in which software is developed can increase its ability to resist, withstand, and recover from attacks. The presenters will also discuss the relationships and dependencies between security as a property of software and other desirable properties, such as reliability, safety, and quality. Key to producing secure software is the integration of risk management into the processes used to create and sustain that software.
This tutorial will thoroughly examine how risk management considerations and activities can be "injected" into all phases of the software development lifecycle. We will then present the principles and practices that will enable developers to build more secure software. We will introduce a number of methodologies, standards, and tools that are emerging to address the problem of how to establish a development process that will produce software that is secure. We will also discuss the education and training of developers that will enable them to use these practices, methods, and tools effectively.

1030 - 1045 Morning Refreshments

1200 - 1300 Lunch

1300 – 1515 *Software Safety Case Management*

Track 1 Tim Kelly, High Integrity Systems Engineering Group, Dept. of Computer Science, University of York

In recent years there has been a marked shift in the regulatory approach to ensuring software safety. Whereas compliance with development standards was the norm, the responsibility has now shifted back onto the developers and operators to construct and present well reasoned arguments that their software is acceptably safe (in a system context). These arguments (together with supporting evidence) are typically referred to as a software "safety case". This tutorial will be of interest to anyone with responsibility for managing, constructing, or reviewing software safety cases.
The following topics will be covered:

- The role and purpose of the software safety case
- Approaches to constructing and presenting clear safety arguments (including the Goal Structuring Notation)
- The current "state of the practice" in preparing software safety cases. In particular we will discuss the current emphasis on process-based standards and the use of general integrity claims (e.g. Software Integrity Levels).
- The typical architecture of a software safety case
- Current work on developing product-based approaches to software safety justification and shifts towards goal-based software safety regulation (as exemplified by the recently introduced UK Civil Aviation Standard SW01).

1300 – 1515 *An Introduction to Attack Patterns as a Software Assurance Knowledge Resource*

Track 2 Sean Barnum, Principal Consultant, Cigital

This tutorial will introduce the concept of attack patterns as a powerful representation of the attacker's perspective in software assurance, give an in-depth contextual grounding in transforming the concept into a useful knowledge construct and give specific guidance on how attendees can tactically and strategically leverage attack patterns as part of their software assurance activities. This tutorial should be of value to anyone involved in software assurance, including specifically: security analysts; requirements engineers & analysts; architects & designers; developers; QA engineers & testers; policy authors; software development managers; systems operations engineers, etc.

1515 - 1530 Afternoon Refreshments

1530 – 1745 *Software Safety Case Management (Continued)*

Track 1 Tim Kelly, University of York

1530 – 1745 *MILS: Architecture Enabling High Assurance*

Track 2 Gordon Uchenick, Objective Interface Systems & W. Mark Vanfleet, National Security Agency

Multiple Independent Levels of Safety/Security (MILS) is a layering architecture that addresses the problem of obtaining and maintaining high levels of assurance certification. The basic principle of MILS is to dramatically reduce the size and complexity of safety or security critical code into a minimal Separation Kernel. Inspection of this smaller and simpler code body at the highest level of rigor, including mathematical proof of correctness via Formal Methods becomes practical and affordable.
The MILS Separation Kernel provides trustworthy enforcement of only four fundamental safety and security policies: Data Isolation, Control of Information Flow, Periods Processing, and Damage Limitation. All other traditional operating system functions such as device drivers, file systems, and communications stacks are moved into unprivileged address spaces into layers called Middleware and Applications. Instead of being able to violate safety and security policies, these components are now strictly subject to them. Components can physically interact only by their official and proven interfaces, eliminating all side effects when combined into a more complex system. Updating one component does not invalidate the certification of any other component. Attendees to this tutorial will develop an understanding of the requirements for protecting and separating data at multiple safety and sensitivity levels when they are integrated into a single system. The overlap and reinforcement of safety and security certification objectives will be discussed.

TUESDAY – MARCH 6, 2007

0900 - 0915 *Welcome & Opening Remarks*

Ken Berk, Vice President-Business Development, Object Management Group

0915 – 1000 *Keynote Presentation* - High Assurance Reliable Software

John P. Hopkinson
President, International Systems Security Engineering Association
Security Strategist – EWA Information & Infrastructure Technologies, Inc.

The need for High Assurance Reliable Software, while most often associated with software intended for security-related applications, is in fact universally applicable to all software. It can be argued that the industry does in fact know how to develop Secure, High Assurance Reliable Software, and has demonstrated its ability to do so. However, not all software produced fits the description of High Assurance Reliable Software, regardless of whether it is intended for security purposes or not.

Public and user trust in software, and in fact in Information and Communications Technology in general, is low due to multiple demonstrations of failure. The challenge then is not so much to be able to produce High Assurance Reliable Software, but to produce universally applicable High Assurance Reliable Software, demonstrate that assurance, rebuild public and user trust, and to find solutions to insulate new software from legacy systems that lack these properties.

1000 - 1015 Morning Refreshments

1015 – 1230 Session 1: *Assurance Cases*

Chair: Samuel T. Redwine, Jr., Associate Professor, James Madison University

How does one provide explicit, tangible grounds for justified confidence in software and systems? This session starts with two example approaches to answering this question followed by a presentation addressing what general forms satisfactory answers might take.

Manufacturing Software Interoperability and Assurance using Standardized Capability Profiling: An ISO 16100 Proposal

Michiko Matsuda, Professor, Kanagawa Institute of Technology - Qian Wang, Professor, Southeast University - Em delaHostria, Manager, Advanced Technology Group, Rockwell Automation - Eiji Arai, Professor, Osaka University

ISO 16100 provides a collective framework for plugging vendors' tools, methodologies and technologies together into solutions that dramatically reduce the time and cost associated with the development and integration of software capability and assurance. In ISO 16100, the interoperability and assurance of software units can be managed through their profiles, which describe their capabilities that are associated with the aspects of functionality, interface and structure of the software unit. This presentation will provide an explanation of the methodology, which is proposed in ISO 16100, and will introduce a trial implementation related to the ISO 16100 proposal.

Building High-Assurance Systems out of Software Components of Lesser Assurance Using Middleware Security Gateways

Sebastian Staamann, Director for Security Products and Solutions, PrismTech

The decomposition of application systems into software components ready to be deployed on different nodes enables the enforcement of detailed application-level security policies for the interactions between the single components. The evaluatable enforcement of application-level security policies on these interactions enables assurance levels for the overall system, which are higher than those of the single parts. The use of distribution middleware for the component interactions enables the building of generic reference monitors. The presentation discusses the building of high-assurance systems out of components of lesser assurance based on the use of middleware security gateways for CORBA, DDS, and Web Services.

Towards a Meta-model for Dependability Cases
George Despotou, D. Kolovos, R. Paige, F. Polack and T. Kelly, University of York

A dependability case communicates an argument that a system has acceptably met its objectives, addressing qualities such as safety, reliability, performance and security. Construction of a dependability case involves the elicitation and definition of dependability requirements and supporting assurance arguments, as well as the documentation of trade-offs and design rationale. A meta-model for an argument-based approach to developing dependability cases is presented. The meta-model defines and extends the Goal Structuring Notation (GSN) – an argumentation notation and method for constructing arguments that is already well established in the safety domain.

1230 - 1330 Lunch

1330 – 1630 Session 2: *Providing a Standards-based Tool Foundation for Software Assurance*

Chair: Fred Waskiewicz, Director of Standards, Object Management Group

The challenges of providing software assurance to industry can only be met through a standards-based suite of tools capable of creating a full and comprehensive solution that dramatically reduces the time and cost associated with software assurance and modernization activities. The four presentations within this session provide an insight into existing standards-based tools and development efforts and describe the emerging software assurance standards landscape.

Certifying Applications for Known Security Weaknesses: The Common Weakness Enumeration (CWE) Effort
Robert A. Martin, Principal Engineer, MITRE Corporation

The Common Weaknesses Enumeration (CWE) initiative is a community effort focused on the development of a common dictionary of the underlying security weaknesses that can lead to exploitable vulnerabilities in software systems. Agreed and common definitions and technical understanding is foundational to the effective use of tool-based assurance arguments in reviewing software systems for weaknesses in code, design, or architecture. This talk will explore the CWE effort, the various participants in the effort, and the variety of companion and other activities that are leveraging it.

SOA ,Technical Risks, and Emerging Standards
Victor L. Harrison, Partner, Federal Consulting Practice, Computer Sciences Corporation

Building systems to satisfy current and future business goals is critical to the success of transformation and IT investment. Business transformation hinges on people, process, and technology considerations and at the core hinges on software architecture. The conceptual integrity of the associated attributes drive software architecture design. Choosing and designing an architecture for systems - one that satisfies the functional as well as the nonfunctional or quality-attribute requirements (interoperability, reliability, security, maintainability, etc.) are vital to the success of those systems. Service-oriented architecture (SOA) has gained widespread popularity for business transformation. Although some work has been done on analyzing how particular capabilities and associated attributes, such as security and interoperability, are handled within an SOA, this session will introduce a more thorough examination of the relationship between SOA, the conceptual integrity required of the attributes, and the associated impacts on standards, measurement, methods, and programmatics.

Software Assurance Ecosystem
Djenana Campara, Chief Executive Officer, KDM Analytics

As existing software systems get larger and more complex, they evolve into challenging and often conflicting designs that hinder system comprehension, compromise architectural integrity and decrease maintenance productivity. This creates severe problems moving forward. The system becomes more defect-prone, vulnerable to attacks and resistant to enhancements, which in turn reduces a level of confidence and trustworthiness of software systems. Lack of automated tools capable of fully analyzing end-to-end complex systems decreases the level of software assurance even further. Achieving a breakthrough in addressing this vicious circle of lower and lower levels of assurance requires the collaboration of all the key stakeholders, including tool vendors, software integrators and software suppliers. This collaboration has resulted in the formation of the Software Assurance Ecosystem.

Creating a Baseline Functional Specification for Source Code Analysis Tools
Michael Kass, Computer Scientist, National Institute of Standards and Technology

Static analysis tools, in particular source code analysis tools are becoming an integral part of today's software development environment. Industry recognition that "catching a bug before release" significantly reduces software maintenance costs is making integration of these tools commonplace in software development shops. With the proliferation of these tools come the questions:

- Do they share a common set of functions?
- Can their bug detection capability be measured?
- Are some tools "better" than others at identifying weaknesses?
- How should "false positives" and "false negatives" factor into tool measurement?

The U.S. Department of Homeland Security (DHS) would like to answer these questions, and is working with NIST through the SAMATE project address them.

1430 – 1900 *Demonstration Area Open*

1440 – 1510 Afternoon Refreshments

1630 – 1800 **Panel:** *Bringing Standards to Software Source Code Security Assessment*

Moderator: Robert A. Martin, Principal Engineer, MITRE Corporation

The need for standards within code security assessments is coming from the software acquirers, tool users, and researchers. Cooperation and collaborative research is going on to make this happen, with a specific focus on developing a dictionary of software flaw types. Panel members will discuss how this will shape and mature the source code security assessment industry, and dramatically accelerate the use and utility of these capabilities for organizations in reviewing the software systems they acquire, develop, and use.

Panelists:
Brian Chess, CTO, Fortify Software
Jack Danahy, CTO, Ounce Labs
Dr. Larry Wagoner, NSA
Michael Weider, CTO, Watchfire
Chris Wysopal, CTO, Veracode

1800 – 1900 *Demonstration Area Reception*

WEDNESDAY, March 7, 2007

0830 – 1200 Session 3: *Code Analysis Tools/Evaluation of Tools*

Chair: Djenana Campara, CEO, KDM Analytics

As software systems grow in scope and complexity and have an accelerating dependency on COTS and Open Source, they introduce greater variability in design subsequently hindering comprehension and reducing the ability to effectively "evolve" aging or respond to bugs. Ultimately, this results in the fielding of a growing number of potentially unstable and vulnerable applications in our operational environments. It is for these reasons we need tools and technologies that will support our efforts in building secure software products. This session will elaborate on tools and techniques currently employed within development organizations to overcome such important issues as well as the status and gaps of such tools and technologies that preclude their wider adoption within software industry.

Software Reviews at Ericsson
Dominique Toupin, Software Quality Assurance Manager, Ericsson

Electronic context sensitive review is one of the most efficient ways to improve software quality. Ericsson has developed an Eclipse based tool, framework and data model to do electronic context sensitive review. This type of review is done without leaving the context where the artifact was created. You will learn why reviewers are spending less time in a review, finding more anomalies and why more reviews are being performed. The presentation will show a combination of the tool/framework/data model, how the reviewing features integrate with the artifact creation context. This type of review is applicable to source code, UML, requirement, test cases, and methods.

High Fidelity Static Analysis for Secure Enterprise Software Requires Platform Knowledge
Nikolai Mansourov, CTO, KDM Analytics

The industry of static analysis tools is still quite immature, and has to evolve to catch up with the complexities of enterprise software development. In order to perform high fidelity security analysis of entire enterprise systems, the scope of static analysis needs to be increased to include the operating environments of software applications and the enterprise integration supported by these environments. We show how the static analysis community can benefit from standardization in the area of modeling enterprise applications driven by reverse engineering community.

Opportunities and Obstacles to Using Static Analysis for the Development of Safety-Critical Software
Redge Bartholomew, Engineering Manager, Rockwell Collins

This presentation will briefly describe static analysis and the requirements of DO-178B that static analysis could potentially address. This will include the potential role of the static analyzer in verifying safety critical software as well as the qualification criteria imposed on software verification tools. It will describe internal experience to date in using static analyzers in a DO-178B context, and in particular it will describe obstacles to their use. It will describe increasingly compatible government/industry working group efforts aimed at overcoming many of these obstacles, and will finish with a description of potential solutions.

Gaps in Static Analysis Tools Coverage
Pedro Vales, Principal Software Engineer, James Butler, Manager, Knowledge Solutions, David Rager, Sr. Software Engineer, Charles Stack, Principal Software Engineer and Christopher Telfer, Ph.D., Principal Software Engineer, Concurrent Technologies Corporation

This presentation describes key gaps of a sampling of state-of-the-art, commercially available, static analysis tools. It discusses the flaw detection coverage as well as ability to discern the root causes of flaws. The presentation also discusses the importance of tunability in both the analysis and the auditing processes. We, furthermore, describe how report generation, flaw browsing and detection accuracy affect the usefulness of a static analysis tool. We conclude that no single tool provides enough feature coverage for rigorous third party software assurance analysis. Finally, we suggest improvements for static analysis offerings including approaches to defining consistent metrics, terminology and interoperability standards.

1000 – 1600 *Demonstration Area Open*

1000 - 1030 Morning Refreshments

1200 - 1300 Lunch

1300 – 1500 Session 4: *MDA Tools & Technologies for Software Assurance*

Chair: David Chizmadia, Sr. Security Assurance Analyst, Promia

The earliest researchers and practitioners of computer security assurance in the US DoD have recognized that a key aspect of higher levels of security assurance is the use of formal, rigorous modeling techniques. Subsequent experience and research has confirmed this recognition. Until relatively recently such modeling techniques have been considered well beyond the realm of commercial practice, but with the advent of MDA standards and tools, modeling is moving into mainstream practice. The presentations in this session will show three ways in which MDA-based modeling tools can be used to increase assurance in software based systems.

Automatic Model-driven Security Policy Generation for High Assurance Systems

Ulrich Lang, CEO, ObjectSecurity - Rudolf Schreiner, CTO, ObjectSecurity
Ramesh Bharadwaj, High Assurance Center, US Naval Research Laboratory, Tom Ritter, Fraunhofer Institute FOKUS

New security tools for central security policy management and for software modeling are available that greatly simplify security policy management. The combination of these tools can automatically generate central security policies for complex IT environments, which can be managed in new central, consistent policy management products that go way beyond traditional IAM. This presentation will address case studies of air traffic control in Europe and work by the US Naval Research Laboratory.

Fault Tree Analysis of UML Designs

Christopher Harper, Director, Avian Technologies Ltd. & Alan Parkinson, AGP Micro Ltd.

UML is now being applied to safety critical systems. The definition of the semantic basis of UML has improved significantly in UML2.0, and the notation is now sufficiently mature to permit safety analysis techniques such as Fault Tree Analysis. We present a FTA technique for UML, which proceeds by constructing fault trees for UML behavioral diagrams, matching their graph structure systematically. The value of software FTA is gained by extracting candidate defects from the analysis, and writing test cases that can reveal their presence. The technique permits the generation of verification conditions written in OCL, derived directly from the analysis.

Harmonizing System Development and Test Development with MDA

Zhen Ru Dai, Fraunhofer FOKUS

Due to increasing complexity of today's software systems, the early integration of system development and test development processes becomes more and more important. By doing so, design mistakes and implementation faults can be detected in an early stage and the overall development time and costs can be reduced significantly. The Model-Driven Architecture (MDA) provides good means for system development, the framework does not address test development steps. In this presentation, we explore harmonizing system development and test development in the context of the MDA Framework to achieve high software quality.

1500 – 1530 Afternoon Refreshments

1530 – 1730 Session 5: *Software Assurance State of the Art*

Chair: Robert A. Martin, Principal Engineer, MITRE Corporation

The area of software assurance encompasses a rapidly evolving mix of efforts. This session will cover three separate examples of software assurance activities that are pushing the edges of practice into tomorrow. Between them, discussions about what works, what should be done, and ideas about what needs to be done will be balanced with practical experience and knowledge of things that have show success and failure in the past.

So You Have to Verify Software? A Quick Look at What You Should Expect

Frédéric Michaud & Frédéric Painchaud, Defence Research and Development Canada – Valcartier

This presentation surveys the results of an evaluation of automatic software verification tools for the C/C++ programming languages. The first section gives the most common usage scenarios by providing insight into which bugs can be detected, what kind of applications can be verified, and when should the verification take place. Then, we present our tool recommendations by discussing which are available and what attributes to consider. The third section pinpoints a few significant problems that arise when one tries to verify source code. Finally, we make a few observations with respect to analyzing the diagnosis of the verification process.

Using the Principle of Least Authorization to Improve Software Assurance
David Chizmadia, Sr. Security Assurance Analyst, Promia

This presentation will introduce Object Capabilities, which are a concrete implementation approach for the Principle of Least Authorization (POLA) policy, and the common OC design patterns used to create secure systems. The POLA policy is particularly powerful because it appears to enable every other enforceable security policy using only OC design patterns and very minimal (and generally useful) changes to the core OC design. The presentation will also cover existing efforts to develop operating systems, languages, and distributed systems middleware that implement the POLA model using Object Capabilities.

Full Cycle Real Time Information Assurance
Sumeet Malhotra, Global Director Of Advanced Research, UNISYS

In order to do a comprehensive job of information assurance analysis in any operational environment, it is important to look at the complete environment in which information resides - from the software in which content is available down through the various layers of APIs of frameworks that the original writers of the corresponding software leveraged, down to the OS and then to the BIOS, firmware and actual Hardware of the systems where information exist.

Also, during initial software development, vulnerabilities can become inherent at any stage of the complete software development lifecycle (SDLC). Vulnerabilities can appear during the inception and requirements gathering stage of the SDLC or during the architectural design and analysis stage or during the actual development stage or during the testing or debugging stage. This presentation will explain what kinds of vulnerabilities can crop up at what stages during the development of software and during actual runtime of that software. It will give an industry overview of the leading edge solutions that are available to combat those vulnerabilities.

1800 – 2000 ***Workshop Reception*** ***hosted by*** Promia

Program Committee

J.D. Baker*, BAE Systems*

Ben Calloni, *Lockheed Martin*

Djenana Campara, *KDM Analytics*

David Chizmadia, *Promia*

Kevin Loughry, *OMG*

Robert Martin, *MITRE*

Sam Redwine, *James Madison University*

Said Tabet, *RuleML Initiative*

Gordon Uchenick, *Objective Interface Systems*

Fred Waskiewicz, *OMG (PC Chair)*