Development of High-Quality Secure Service-based Software Systems

Stephen S. Yau
Information Assurance Center, and
School of Computing, Informatics, and Decision Sys. Eng.
Arizona State University
Tempe, Arizona, USA

ISSRE 2012
Outline

- Introduction of Service-based Systems (SBS)
- Challenges of Developing High-quality Secure SBS
- Current State of Art
- Our Research
- Future Research
Service-based Systems (SBS)

- Based on service-oriented architecture (SOA)
- Adopted in various application domains
  - E-business
  - Health care information systems
  - Homeland security
  - ...

- Services in SBS provide *standard interfaces* for users to access capabilities offered by various providers

- Services in SBS often need to be composed to form *workflows (business processes)* to provide functionality not provided by any individual service
Major Advantages of SBS:

- Facilitate *interoperation of heterogeneous systems* through standard protocols governing the interactions of services
- Enable *rapid composition* of applications from distributed services
- Make *adaptation* of applications possible through service discovery and late-binding
  - Satisfy dynamically changing requirements of users
  - Maintain or optimize performance in dynamic operational environments
Information Assurance in Service-Oriented Environments

- SOA is being adopted in critical information systems
  - Deployed over public networks - open to attacks from the world
  - Multi-party service interactions involving unknown entities
  - Lack of central control
  - Large-scale and cross-domain service collaborations
  - Traditional QoS management approaches not applicable in SBS
- Information assurance is of prime concern of SBS users
  - Defending SBS against various attacks to ensure continuous availability of services
  - Protecting confidential information of various parties, including service providers, infrastructure providers, and end users
  - Managing trust relationships among various parties to provide them high confidence in sharing and utilizing information services
Challenges of Developing High-Quality Secure SBS

- Interactions among services may have unforeseen consequences in security, QoS, trust and risk

  - Possible problems:
    - Untrusted/malicious services
    - Intermediate results generated during service interactions may reveal sensitive information
  - Need formal analysis to verify compliance to all security, QoS, trust and risk management requirements, especially in service interactions
Challenges of Developing High-Quality Secure SBS (cont.)

- SBS often has multiple QoS requirements from multiple users
  - Failure to satisfy any expected QoS requirements (e.g. throughput, service delay, security) affects users’ trusts
  - Tradeoffs among expected QoS requirements are often needed
    - Mechanisms providing security protection are computationally intensive and often requires certain sacrifice in other QoS (e.g. service delay, throughput)
  - System usability
Challenges of Developing High-Quality Secure SBS (cont.)

- Operational environments of services may dynamically change
  - Make assessing trust and risk difficult
  - Need situation awareness for dynamic assessment of risk and trust
  - Need adaptive enforcement of security policies

- Information needed for making decisions on security, QoS management, risk and trust is usually distributed on multiple services and organizations.
  - Need cooperative decision making (e.g. delegation, policy composition with multiple organizations, collaborative QoS management, risk assessment, trust evaluation)
  - Need efficient enforcement of distributed security policies
Challenges of Developing High-Quality Secure SBS (cont.)

- **Service selection and composition**
  - How to select *appropriate services* and *compose* them together to satisfy both the functional and QoS requirements of users while ensuring overall system security?
  - Need methods for analyzing *tradeoffs* among multiple QoS and security in SBS
  - Need *quantitative metrics for system performance* and *security* in SBS
  - Need a *service ranking* approach to help users find the best services that satisfy their security and QoS requirements
Challenges of Developing High-Quality Secure SBS (cont.)

- **Service adaptation**
  - How to *control and optimize the adaptation* of SBS intelligently and efficiently?
  - Need effective methods for *adapting resource allocation* and *selecting service instances* to avoid violations of users’ QoS requirements and improve system performance in run-time

- **QoS monitoring in run-time**
  - How to *monitor overall status* of service-based systems efficiently?
  - Design of an *effective architecture* for *QoS M/A* of SBS
**Current State of Art**

Automated Service Composition Based on Various Formal Specifications

- Process calculi; BPEL4WS

**QoS-aware Service Composition in SBS**

- Optimizing QoS attributes of services using the genetic algorithm during the service compositions (G. Canfora, et al., Univ. of Sannio, Italy).

- QoS provisioning for composed services, based on the Service Level Agreement (SLA) contracts of individual services (X. Gu, et al., Univ. of Illinois, Urbana)

- Developing QoS-aware middleware for web service composition to maximize users’ satisfaction expressed in utility functions over QoS attributes. (L. Zeng, et al., IBM)

- Policy-driven service composition with information flow control (W. She, et al., Univ. of Texas at Dallas & Purdue Univ.)
Current State of Art (cont.)

- Tradeoffs between Security and Multiple QoS in SBS
  - Development of a framework for quantifying the strength of system security (M. Satyanarayanan, et al., Carnegie Mellon Univ.)
  - An adaptive model for tradeoff between service performance and security in service-based environments (S. Yau, et al., ASU)
  - A comprehensive QoS model for service-based systems, (I. Jureta, et al., Univ. of Namur, Belgium)
Current State of Art (cont.)

Adaptive Resource Allocation in SBS

- A multi-layered resource management framework for dynamic resource management in enterprise systems. (P. Lardieri, et al., Lockheed Martin Corp.)

- Decentralized online resource allocation for dynamic web service applications (J. Stoesser, et al., Univ. of Karlsruhe, Germany).

- A regression based analytical model for dynamic resource provisioning of multi-tier applications (Q. Zhang, et al, HP Labs)
Current State of Art (cont.)

Design of SBS for QoS Monitoring and Adaptation

- A methodology for developing adaptive service-based software systems with distributed monitoring and adaptation of multiple QoS (S. Yau, et al., ASU)
- Comprehensive QoS monitoring of Web services and event-based SLA violation detection (Michlmayr, et al., Vienna Univ. of Tech, Austria)

Testing of SBS

- Dynamic reconfigurable testing of service-oriented architecture (W. Tsai, et al., ASU)
Situation Awareness in SBS

- Developing situation-aware SBS
  - Adaptable Secure Service-based Systems (S. Yau, et al., ASU)
  - A framework for context-aware adaptable web services (M. Keidl, et al., Univ. Passau, Germany)
  - Service-Oriented Context-Aware Middleware (SOCAM) (T. Gu, et al., Nat’l Univ. of Singapore, Singapore)

- Service discovery
  - Context-Aware Service Oriented Architecture (CA-SOA) for ubiquitous web service discovery and access (Y. L. Chen, Ching Yun Univ., Taiwan; J.H. Yang, National Central Univ., Taiwan; Jia Zhang, Northern Illinois Univ.)
  - Situation-aware semantic based service discovery (S. Yau, et al., ASU)

- Applications: ESCAPE – context management for teamwork and disaster management (Truong, et al., Vienna University of Technology); Omnipresent – context-aware location-based service system (De Almeida, et al., Univ. Federal de Campina Grande, Brazil)

*System support for monitoring context and situation changes, and adapting applications and systems in SCC environments needs to be developed*
Current State of Art (cont.)

- Applying Software Cybernetics for Developing Adaptive Software Systems:
  - Software cybernetics: Apply principles of control theory in development, deployment and operations of software systems for automation, optimization and adaptation (K.Y. Cai, Beihang Univ., etc.)
  - A performance guarantee approach for cloud applications (J. Shao & Q.-X. Wang, Peking Univ.)
Outline

- Introduction of Service-Based Systems (SBS)
- Challenges of Developing High-quality Secure SBS
- Current State of Art
- Our Research
  - Adaptable Situation-aware Secure Service-based (AS$^3$) Systems
  - Adaptive Service-based Systems (ASBS) with QoS Monitoring and Adaptation (M/A)
    - Motivation and Overview
    - Tradeoffs between Various QoS Aspects and Security in SBS
    - An Architecture for ASBS with QoS M/A
- Future Research
Our AS³ System Architecture

- Mission Planning Services (MP)
- Workflow Scheduling Services (WS)
- Directories
- Discovery Services (DS)
- Security Agents (SeA)
- Situation-Awareness Agents (SAA)
- Services
- Comp/Comm Resources
- Various Capabilities

Specification of Mission Goals

Feedback

Mission Planning Services (MP)

Workflow

Specification of Mission Goals

Mission Planning Services (MP)

Workflow Scheduling Services (WS)

Security Agents (SeA)

Situation-Awareness Agents (SAA)

Directories

Various Capabilities

ISSRE 2012, Yau
Our Approach to Rapid Development and Deployment of AS³ Systems

- NMR Axioms
- Coordination Axioms
- SAW Axioms
- Security Policy Axioms
- Failure Handling Axioms

Natural Deduction-Based Proof System Kernel

- Workflow & Backup Workflow Agents
- SAW Agents
- Security Agents

Workflow Scheduler

Resource Agents
- Annotated Workflow Agents
- Annotated SAW Agents
- Annotated Security Agents

AS³ Calculus Model Checker
- AS³Cal2Java Compiler
- Distributed SINS Agents

Goal

Static Feedback (Fail)

Dynamic feedback

ISSRE 2012, Yau
Key Features of Our Approach to $\text{AS}^3$ Systems

- Specification of SAW requirements for service-based systems
- Specification of flexible security policies by incorporating situation-awareness
- Rapidly generate SAW agents and security agents, which enable adaptive workflows and enforce security policies
Summary of Our Research on AS³ Systems

- Focus on a unified logic-based approach to synthesizing software agents for situation-aware workflows and security policy enforcement.

- Consider workflow QoS and allocate resources during the synthesis process, but do not provide sufficient runtime support for monitoring and controlling workflow QoS as well as adapting resource allocation.
Adaptive Service-Based Systems (ASBS) with QoS Monitoring and Adaptation (M/A)

- Satisfying *multiple QoS requirements*, including timeliness, throughput, accuracy and security, for multiple workflows *in dynamic service-based environments*.

- Requiring the following capabilities:
  - *Monitor resource status and QoS, detect changes in execution environments*, and *determine whether adaptations are needed*
  - *Allocate resources* to services dynamically to maximize service performance and resource utilization
  - *Dynamically select proper service instances* based on their estimated QoS and resource consumption, and *adapt service composition* to provide satisfactory QoS
Our Approach to ASBS with QoS M/A (cont.)

Layered-view of Existing Service-based Systems

- Presentation: Portlets, WSRP
- Business Processes
- Services
- Enterprise Components
- Operational Systems: ERP, CRM, data warehouse...

ASBS

- Users
  - Have QoS expectations
  - Request services
  - Adaptation commands
  - Situations & QoS measurements
- QoS Adaptation
  - Adaptation commands
- QoS Monitoring
  - Measure changes of resource states
  - Produce events

Functional Services

- Produces events
- QoS Monitoring
- Consume resources
- Affect QoS

Resources

Situations & QoS measurements
Our Approach to ASBS with QoS M/A (cont.)

Our Approach

I. A system modeling approach to constructing Activity-State-QoS (ASQ) models for QoS estimation in SBS

II. Techniques for QoS M/A of SBS in dynamic environments

III. SOA-compliant simulation techniques to support validation of our research results
Our Approach to ASBS with QoS M/A (cont.)

- A *system modeling approach* to facilitating QoS estimation in SBS
- Techniques for *adaptation* of SBS in dynamic environments
  - *Dynamic resource allocation for multiple workflows* to optimizing system resource utilization
  - Techniques for *handling tradeoff between security and service performance*
  - Design of *service-/workflow-level QoS M/A components*
- *SOA-compliant simulation* techniques to support validation of our research results
  - An SOC-DEVS simulation environment for SBS
Tradeoff Between Performance and Security

- Services in SBS are adaptable and can provide different QoS for various users
- Tradeoff between performance and security
  - Find service instances for best tradeoff between performance and security according to users’ preferences on performance and security
  - Satisfaction guarantee on users’ minimum QoS requirements on both performance and security
- Challenges
  - Need to develop quantitative metrics for performance and security
  - Need to develop the tradeoff objective function
QoS-based Service Ranking

- Help *users* to find most suitable service instances that satisfy their QoS requirements
- Users have QoS requirements on various QoS aspects
- QoS-based service ranking
  - Compute *satisfaction scores* for each QoS aspect to measure how well service instances’ QoS satisfy users’ requirements on this aspect
  - Compute *overall satisfaction scores* for each service instance by aggregating their satisfaction scores on all QoS aspects with weighting factors
  - *Rank service instances* based their overall satisfaction scores
Requirements on QoS M/A in SBS

- Support monitoring data collection and processing
  - Server-level
    - Status of *system resources*: memory, CPU, network bandwidth, etc.
    - Status of *service instances*: service availability, QoS measurements (service delay, throughput, etc.), usage statistics (request arrival rate, service rate)
  - Workflow-level
    - Status of *workflow execution*: aggregation of workflow-level QoS, detection of situation changes requiring adaptation (service/host/network failures, unsatisfied QoS requirements)
Requirements on QoS M/A in SBS (cont.)

- Support workflow execution and QoS adaptation
  - Controlling workflow execution
    - Intercepting and parsing user requests
    - Starting, suspending/resuming workflows
    - Dynamic service replacement (binding new service instances) and reconfiguration (changing service parameters)
  - Coordinating distributed decision-making (QoS optimization) processes
    - Processing data I/O for distributed QoS optimization processes
    - Providing communication support
General System Architecture of Distributed QoS M/A Components
A Workflow Monitoring, Execution and Control Architecture

- Separation of service-level and workflow-level components
  - Avoid redundancy at the service-level since services are shared among different workflows
  - Distribute M/A tasks properly (also consistent with distributed QoS optimization processes)

- Introducing a *QoS adaptor directory* to facilitate the discovery of distributed QoS adaptors
Workflow agents and workflow-level QoS adaptors are dynamically generated

- Based on workflow and service specifications provided by application developers when users have new application requirements
- Each workflow has its corresponding workflow agent and workflow-level QoS adaptor
- Multiple instances of workflow agents can be deployed for a workflow, one workflow-level QoS adaptor for each workflow
Future Research

- Distributed QoS optimization approach to provide better service for users
- Security metrics to measure more security features
- More flexible service ranking approaches to support users’ QoS requirements
- Dynamic generation and deployment of QoS M/A components, including workflow agents and workflow-level QoS adaptors
- Automated and efficient verification and testing techniques for situation-aware applications in SBS
Future Research (cont.)

- Innovative ways of integration of fuzzy logic and semantic technology to provide
  - Easy to use methods for specifying situation awareness requirements
  - Capability of handling imprecise requirements involving situation awareness
  - Capability to handle uncertainty of users’ behavior
  - Capability to analyze and resolve conflicting requirements on situation awareness from multiple users sharing computing resources
- Innovative ways of incorporating software cybernetics to support system monitoring and adaptation in SBS
Thank You!