Semantic Service Oriented Architecture

Case Study for
OMG SOA/MDA/WS Workshop
March 30, 2006
Lack of Awareness and Sharing of Available Capabilities (Services)

– Sheer Volumes of Data and Services Compounds the Problem
  • Word of Mouth “Discovery” Typical
– Available XML Web Services Solutions are ‘Pervasive,’ but…
  • Lack Ability to Easily Discover Services
  • Are Location Dependent; “Stale” References Possible
  • Are Protocol Dependent
  • Have Weak, or No, Semantics
  • Results in (at best) a Centralized Distributed Architecture
  • Include Ever-growing Multitude of Largely Unimplemented Standards (re: WS-*)
– When Found, Services…
  • Not Readily Interoperable
  • Not Described by “What They Provide”
  • Result in Human-Centric, Ad-hoc and Intermittently Repeated Processes
– Fragmented, Sub-Optimal Operations
  • Long Standing Problem – Analysts Aren’t Able to Focus on Core Competencies
SSOA Complements XML WS by Supporting a SOA that is:

- **Semantically Enabled**
  - Powering Efficient Publishing, Discovery, and Execution of all Available Services
  - Recommending Appropriate Services when New Services Come Online
  - Allowing Software Agents to Dynamically Construct Adaptive Workflows
  - Designed and Implemented Based on Current and Emerging Standards
  - Providing the Ability to Compose “Virtual Applications”

- **Powered By Sun Microsystems’ Jini™ Distributed Computing Model**
  - Location Independent, Protocol Agnostic Services
  - Autonomic (e.g. Spontaneous Networking, Self Healing, Self Synchronizing)
  - Allows Near-Real Time Collaboration and Capabilities Sharing
  - Relevant Services *Presented to* End Users
  - Distributed Event Model
  - Decentralized Distributed Computing Capability

- **Demonstrating Ability to Share Resources Across the Enterprise**
  - Enhancing Current SOA Projects by Acting as Risk Reduction/Complimentary Task
SSOA Foundations

- Service Oriented Architecture (SOA)
  - Separation of Concerns
  - Functionality Discovered, Used, Re-used
  - Standard Interfaces Abstract Impl’n Details

- Semantics Based Computing
  - Machine Interpretable Content
  - Structure + Epistemology + Logic

- Standards Based Design (SBD)
  - Presupposes Pervasive Heterogeneity
  - Integrate Existing Apps w/ New/Future Technologies

- Standards Involved w/ Prototype
  - *ISO 11179 Metadata Registries*
  - ISO 19763 Meta-Model Framework
  - ISO 24707 Common Logic
  - OMG Ontology Definition Meta-Model
  - W3C Semantic Web Services Framework

Diagram:
- Service Broker
  - Discover Services: Publish, Find, then Bind
  - Register Services
  - Invoke Services

Service Diagram:
- Resource
- ServiceProfile
- Service
- ServiceModel
- ServiceGrounding
- ServiceRequestor
- ServiceProvider

What the service does
How it works
How to access it

1. Register Service
   - Publish, Find, then Bind
2. Discover Service
3. Invoke Service

Model Driven Architecture

SCI...
Powerful Jini™-Based Model

Abstraction, Location Independence

Protocol Agnostic

Heterogeneity
• Logical (s/w)
• Physical (h/w)
• S3

From Valaran, Inc.
Prototype SSOA System View

Semantic & Agent Components

Services

Event Processing Agents

SSB_{effective}

SSOA Node 1

SSOA Node 2

SPR

SSR

EDR

SMDR

Dynamically Exchanged
(Sharing/Awareness)
Prototype SSOA Demonstration

Combat Search and Rescue (CSAR)
CSAR Process *(Greatly) Distilled*

IAW Doctrine JP 3.50-2

### Planning
- Manpower
- Assets
- Weather
- Imagery
- Enemy Intel
- Terrain

### Execution

### Adaptation
CSAR with SSOA

RCC Initiates Planning

Threat Analysis

Evaluate Units

Receive Intel Brief

Refine Plan

Rescue Plan

Launch Decision

Distress Indicator

RCC Initiates Planning

Receive Intel Brief

NO

YES

Execute Mission

Assessment

Lessons Learned

Requirements
SSOA Event Flow

1. Events
2. SSOA Event Listener
3. Process Coordinator Agent
4. SWR Query Agent
5. Process Agents
6. Service Execution Agent
7. Semantic Service Registry
8. Lookup Service
9. Business Service

Agent Architecture
**SSOA Event Flow – CSAR Demo**

1. **Distress Signal Service**
2. **SSOA Event Listener**
3. **Process Coordinator Agent**
4. **SPR Query Agent**
5. **Weather Agent**
6. **Imagery Agent**
7. **ThreatID Agent**
8. **Service Execution Agent**
9. **Active Services**

**Semantic Workflow Registry**

- **CSAR**
  - 1. Weather
  - 2. Imagery
  - 3. ThreatID

**Active Services**

- **Weather Service**
- **Imagery Service**
- **ThreatID Service**

**Semantic Service Bus**
Demonstration
SSR Functionality

• Key Concept: Every Service Type is a Collection of Tasks
  – Each Task Semantically Corresponds to a Specific Operation or Action

• Every Service is a Running Instance of a Service Type
  – Multiple Instances of a Specific Service Type May be Deployed to:
    • Build in Redundancy & Provide Load Balancing

• Task Selection Depends on a Semantic Description, Comprised of:
  – Input, Output & Action Types

• Given a Set of these Input, Output & Action Types, the SSR will:
  – Return the Candidate Tasks and Associated Service Types,
  – Provide Necessary Information to Discover and Execute Any of the Tasks Within the SSB

• Weather Service Example:
  – Inputs are: { Location, TimeStamp }
  – Outputs are: { TemperatureC }
    • SSR Matches 3 Potential Tasks:
      – TemperatureC getTemp(Location,TimeStamp)
      – TemperatureC temp(Location,TimeStamp)
      – TemperatureC getTempCelsius(TimeStamp,Location)
    – But Not:
      – TemperatureC getTempCelsius(TimeStamp,Location,Altitude)
Conclusions

- Competition
- Uncertainty
- Intelligence Collection
- Intelligence Failure Fraction
- (Data) Gross Collection Rate
- (Delay)
- Gross Data Collection
- Actionable Intelligence
- Decision Rate

DANGEROUS SHORT-CUT LOOP

- $+$
- $+$
- $+$
- $+$
- $-$
- $-$
- $+$
- $+$
- $-$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $-$
- $-$
- $-$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+$
- $+
Back ups
Jini™ Service Item

Service Item

Service ID

Proxy

Attribute

Attribute

Attribute
Publish, Find, then Bind

1. Service Item published in Lookup Service

2. Client downloads proxy object to access service

3. Client uses the proxy to communicate with the service
Publish, Find then Bind

- Client
  - proxy
- Lookup Service
  - proxy
  - proxy
  - proxy
- register
- register
- lookup
- Service
- Service
- register