Summary SOA (1)

- Software services similar to real-world services
- SOA: abstract architectural concept
- Architecture: publish/find/bind
- Technical benefits:
  - Efficiency
  - Reuse
  - Maintenance
  - Incremental adoption
- Business benefits:
  - Agility
  - Alignment
  - Customer satisfaction
  - Integration costs
  - Dependence

Summary SOA (2)

- Challenges:
  - Training/Skills
  - Discipline
  - Short-term costs
  - Legacy applications
- Possible solution: step-by-step adoption

Summary Web Services (1)

- One approach for SOA
- XML-based interface technology
- Properties:
  - Based on Web standards
  - Relatively simple
  - Platform independent
  - Pervasive
  - Standardized (W3C, OASIS, …)
  - Embraced by the IT industry

Summary Web Services (2)

- Core standards:
  - SOAP
  - WSDL
  - UDDI
- Extensibility
- Extended specifications:
  - WS-Adressing
  - WS-Policy
  - WS-MetaDataExchange
  - QoS: WS-Security, WS-ReliableMessaging, WS-Coordination, WS-Transaction
  - WS-CDL
  - BPEL4WS
- Adoption: step-by-step, core standards widely used

Summary (1)

- SOAP: XML-based messaging protocol
- Defines envelope with header and body
- Nodes: sender/receiver
- Nodes can act as different roles
- Processing model: defines how header blocks are processed by nodes
- Defines data model and encoding
- Interaction styles:
  - Document literal
  - RPC

Summary (2)

- Extensibility: features
  - Supported using modules (header blocks)
  - Supported by transport protocol
- Binding: how to use transport protocol
- HTTP binding
  - Request-response (POST), Response (GET)
- Binary attachments: SOAP MTOM, XOP
- Benefits
  - Standardized
  - Simple
  - Industry support

Summary (1)

- WSDL: XML vocabulary to describe Web services
- Separation in abstract and concrete parts
- WSDL 1.1
  - Types
  - Messages
  - Operations – port types
  - Bindings
  - Ports – Services
- Predefined bindings: SOAP, HTTP/MIME
- Problems and Limitations
  - Messages
  - SOAP binding
  - Services

Summary (2)

- WSDL 2.0
  - No message construct
  - 8 Message exchange patterns
  - Interface extensions
  - Include/import
  - Features and properties
- WSDL 1.1 widely used, adoption of 2.0 will take time but will be stable for a “long” time

Summary Composition

- Business process
  - Activity of related tasks
  - Specified sequence
  - Outcome
- Process model – real world
- Workflow model – technology interactions
- Composition
  - Creating new processes by combining existing services
  - Static or dynamic
  - Orchestration: “part-of”
  - Choreography: “sequencing”
- Web services workflows: quality of service

Summary WS-BPEL (1)

- Process-oriented composition language
- Relies on WSDL
- Block-structured programming
- Basic component: activities
  - Primitive activities
  - Structured activities
- Data handling: containers
- Abstract processes
  - Public interface
- Concrete processes
  - Implementation

Summary WS-BPEL (2)

- Service selection
  - Partner link types
  - Partner links
  - Endpoints
- Binding: static or dynamic
- Exceptions
  - Try-Catch-Throw
  - Exception handling per scope
  - Compensation
  - Default handlers
- Life cycle
  - Instances implicitly created
  - Correlation

Summary WS-CDL

- Declarative language for defining interaction patterns
  - Not executable
- Specifies interactions in B2B scenarios
- Intended to complement WS-BPEL

Summary (1)

- Transactions: fundamental concept
- Transaction properties: ACID
- Transactions in Web services: additional difficulties
- Long-running transactions
  - Compensation
  - Relaxed ACID properties

Summary (2)

- WS-Coordination
  - Foundation layer for coordination
  - Defines:
    - Activation
    - Coordination context
    - Registration
    - Extensibility for completion protocol
- WS-AtomicTransaction
  - Short-lived activities
  - 2-phase commit
- WS-BusinessActivity
  - Long-running activities
  - Allows nesting of operations
  - Compensation-based
  - Can be combined with atomic transactions

Summary Metadata

- Metadata: data about a software entity
- Web service metadata:
  - Data types/structures
  - Message exchange patterns
  - Addressing information
  - Requirements
  - Quality of service
- Web service metadata technologies:
  - XML Schema
  - WSDL
  - WS-Addressing
  - WS-Policy
  - UDDI
  - WS-MetadataExchange

Summary UDDI

- Universal Description, Discovery and Integration
- Service registry
  - White pages: business information
  - Yellow pages: categorization
  - Green pages: technical information
- Data structures:
  - businessEntity
  - businessService
  - bindingTemplate
  - tModel
- APIs for publishers, requestors, other registries
- Original vision: UBR
- Current state: mainly private/semi-private registries

Summary WS-Addressing

- Addressing mechanism required
- URIs insufficient for non-trivial cases
- Endpoint references
  - Address
  - Reference properties
  - Reference parameters
  - Metadata
- Headers
  - Mandatory: To, Action, reference properties/parameters
  - Optional: related endpoints, relationships between messages
- Use of WS-Addressing with SOAP and request-response pattern

Summary WS-Policy

- Describes nonfunctional service behavior
- Separate from WSDL
- Specifications:
  - WS-Policy
  - WS-PolicyAttachment
- Policy: assertions combined using operators
  - `ExactlyOne`
  - `All`
  - `optional` attribute
- Concrete assertions specified by other specifications
- Validation
  - Generic: intersection to get candidates
  - Domain-specific knowledge needed to check compatibility
- Attachment separated from definition

Summary WS-MetadataExchange

- Bootstrapping of interactions
- Extensibility of dialects
- Redirection
- GetMetadata operation
  - Generic request
  - Dialect-specific request
  - Definition-specific request
- Get operation
  - Implemented by metadata reference endpoint