Service Oriented Architecture (SOA) Initiative: Kickoff Forum

SOA Technical Session

Harry Samuels
Kalpesh Patel
Brief SOA Overview
What is SOA?

**SOA is:** an acronym for **Service Oriented Architecture**

SOA is a method of “behind the scenes” interaction between independently functioning computer systems

- Initiating an action (event + action)
- Retrieving or providing data
- Notification of an event (publish/subscribe)

**A “service” is:**

- Software that delivers or consumes a system’s capabilities
- A service can be exposed to consumers through one or more methods
Why SOA?

• Enables real-time integration
• Enables faster integration of new systems
• Reduces spread of data (“multiple versions of truth”)
• Enables exposure of functionality in portals and distributed development of mobile apps
• Lower overall support requirements because the connections between systems are *loosely coupled* and *re-usable*
What Problems are We Solving?

- NU has a maze of custom built data interchange links which are costly to create and maintain, and which unnecessarily replicate information across applications.
- IT@NU cannot use this maze to support
  - real-time business functions
  - cloud-based applications
- We all incur costs when new software packages replicate what is already deployed.
Current Integration of Applications

- Messages are (large) collections of records
- Uni-directional flow; no confirmations
- Transfer is asynchronous and lags by minutes to hours
- Changes must be coordinated and tested
Future Integration of Applications

- Messages are transactional for single cases
- Responses indicate transaction success and may return result data
- Real-time integration
- Recommended practices isolate changes
What does this mean to applications?

• To realize the benefits of future application integrations, business system designers must:
  – Create a stable of exposed services to assist development of new business processes and workflows
  – Encode business rules to determine when and how
    • New users are provisioned and with what access permissions
    • User access is suspended or removed in response to notice of a change in status
What does this mean to applications?

• The future integration environment will require
  – Different software design approaches
  – Different thinking about the availability of data
  – Different thinking about solutions (buy, build, “crowd source”)
  – Different staff skills
Opportunities to Embrace SOA

• New system implementations
• System upgrades offer a value proposition of re-architecting integrations to use the new enterprise architecture
• Also need to proactively pursue those services that will be in highest demand
Does it Work? Yes!
2013 SOA Proof of Concept: Takeaways

✓ Successful deployment of infrastructure
✓ Successful development and testing of read and write services via both SOAP and REST
✓ Experience gained in architecting services to use ESB/MB, translate SOAP to REST, and vice versa
✓ Learning curve encountered: ~50-100% more effort. Gap will shrink as more experience gained (and offset by re-usability and loose coupling)
SOAP vs. REST Web Services

SOAP (Simple Object Access Protocol)
• It relies exclusively on XML to provide messaging services
• It has many extensions that can be used including one for security
• The difficulty of using SOAP depends to a large degree on the language you use

REST (Representational State Transfer)
• REST provides a newer, lighter weight alternative
• Instead of using XML to make a request, REST relies on a simple URL in many cases
• Unlike SOAP, REST can send/receive data as JSON, XML, or even plain text
SOAP vs. REST Web Services

• Consuming application systems may need a SOAP or REST service based on the application’s standards/compatibility, so enterprise teams should be prepared to support both standards

• In the PeopleSoft environment there is a movement toward REST services by Oracle (minimum of PeopleTools 8.52)
  – Oracle is investing in REST services in PT 8.53 and PT 8.54 and they are building all of their mobile apps using REST
SOA Design and Development Principles
Services at Northwestern Now

• The way NU currently uses services is tied to our current data architecture and will not scale

• A services architecture philosophy and mindset – with reusability and loose coupling as the default – will be *entirely new*
  – Use of enterprise middleware is essential
  – Designing for reuse is a paradigm shift and will require more effort up front
  – Developers and business analysts need to build new skills
Key SOA Design Principles

- Targeted Data
- Tight coupling
- Reusability
- Loose coupling
- Decomposition
Targeted Data = BAD

- Targeted data refers to the practice of producing or consuming a unique query, extract, or download of data from a source.
- A general data set may be made available to many consumers with each data set refined to meet the precise needs of the specific consumer.

Problem: redundancy increases our implementation and support burdens and leads to brittle connections.
Tight Coupling = BAD

- “Tight coupling” integrates two systems at a detailed level and forces coordinated changes and testing of both systems when one is modified
  - Database views
  - Passing structured data
  - Passing binary representation of numbers (byte, half-word, word)

**Problem**: cannot change without fear of breaking other components upstream or downstream
Reusability = VERY GOOD

• An existing service should be used rather than creating a new service
• New services can be created using existing services
• A service should implement at most one function
  – Composite services should be used to execute multiple functions in sequence/relation

**Objective:** minimize risk and development time
Loose Coupling = GOOD

• “Loose-coupling” integrates two systems at a higher level by:
  – Mapping representations to self-describing text messages
  – Never changing a service definition
    • Instead, a new version of the service is deployed
    • Then users of the current service are given time to migrate to the new service version

Objective: reduce risk and cost to modify business systems
Decomposition = HOW

• At the limit, all business processes can be decomposed into a set of individual services that are invoked by workflows.
• All applications should expose services to enable higher-level integration through workflows.

Objective: Rapid creation, deployment and modification of business processes.
Enterprise SOA Infrastructure
Enterprise SOA Infrastructure

• The backbone of an SOA environment is an enterprise infrastructure that:
  – Provides security
  – Insulates systems from changes to other systems
  – Keeps track of the locations of services
  – Provides the means to register web services so that they can be found and used by others
  – Can transform protocols
Enterprise Use of SOA

- There are three components to an enterprise architecture for SOA:
  - Enterprise Service Bus (ESB)
  - Message Broker (MB)
  - Web Services Registry (WSR)
Web Services Registry (WSR)

The Web Services Registry (WSR) lists all web services so they can be found by others and used by their own systems

– Complete specifications for the service as needed by designers and programmers
– Location of the service and security requirements
– Service contract (e.g., hours available, max invocations per minute/second, expected response time, etc.)
Enterprise Service Bus (ESB)

The Enterprise Service Bus (ESB) provides an active hub through which systems communicate via services

- The ESB is the Post Office, but it also adds a layer of functionality, e.g., it can enforce security, transform message data, etc.
- The ESB enables a consistent, central URL to be used as the destination/target for services
- As services change (new versions, etc.), endpoints do not have to migrate to a new location
Enterprise Service Bus

• The Enterprise Service Bus is the component of the SOA infrastructure that:
  – Acts as a proxy for web services requests and responses
  – Routes service requests and responses
  – Can transform protocols
    • SOAP -> REST and REST -> SOAP
    • Web services -> Messages
  – Can transform data
Enterprise Service Bus

• The Enterprise Service Bus is the component of the SOA infrastructure that:
  – Can present one web service as multiple web services
  – Can present multiple web services as one web service
  – Can support non-SOA protocols from legacy systems and present them as web services
  – Can monitor and record the usage of services
  – Can enforce SLAs
  – Can throttle usage of services
What is a Message?

• Messages are asynchronous as opposed to web services, which are synchronous
• A message gets published to a message queue
• Subscribers to the message queue, pick up the message when it’s convenient
• Messages use different protocols than web services – they use JMS and AMQP
• Messaging is perfect for event notifications
The Message Broker (MB) provides a hub for event notifications

- If one system knows that a registered event has occurred, it publishes the event to the MB and the MB passes the information on to all systems that have subscribed to the event, i.e., asked to be informed
Message Broker

- The Message Broker is a set of message queues, each with its own topic.
- An application system publishes a message to a message queue and other systems that have subscribed to that message queue retrieve the message whenever they want.
- Messages are durable and remain in the queue until all subscribers have picked them up.
Example: Employee status change

- FASIS event (new appointment, removed appointment, department change, etc.)
- MB “publish” event triggered
- MB invokes listener WS “subscribed” to that event
Security

- HTTPS must be used for all web services requests and responses
- Each consuming application system needs an LDAP service account that must be presented to the ESB when making a request
- Producing application systems may also require the ESB to present another set of credentials (e.g., LDAP service account) when making a request on behalf of the requester
Security

• The ESB can validate that requests are made by approved consumers
• The ESB can check the IP range of a requesting application
NU Implementation of Enterprise SOA Infrastructure
Our Infrastructure & How We Got Here

• We did a Proof of Concept in the fall
• In order to do so quickly and without making a financial commitment we chose 3 products from a company called WSO2
  – WSO2 ESB
  – WSO2 Message Broker
  – WSO2 Governance Registry
Our Infrastructure & How We Got Here

• And we learned a few things in the process about these kinds of products. We learned:
  – Companies package the key functional components differently
  – Some vendors offer special connectors to enable you to present web services where none actually exist
  – Products have different capabilities for transformations, mediations, routing, composition, and decomposition
  – Ease of administration differs widely
  – Integration across products differs widely
Our Infrastructure & How We Got Here

• In the end, we chose to invest in two products from IBM: IBM Integration Broker and WebSphere Services Registry & Repository

• Because of price, Gartner ratings, relationship, better administrative features, and better integration between the two functional components
Our Infrastructure & How We Got Here

• Right now, we are in a state of transition
• The WSO2 products are up in a production high availability mode (we are open for business)
• The IBM products are being implemented now and will be finished by August
• We will cut over with minimal disruption. URLs and endpoints will remain the same. No additional work will have to be done by the service providers or consumers.
The Registry

• The Service Registry may include the following service information:
  – Service Name
  – Description
  – Version
  – Service Provider Owner and Contact Information
  – Service Provider Organization
  – Location of the service and Endpoints
  – Available Operations
The Registry

- Service Level Agreement (hours available, maximum invocations per minute, expected service response time, etc.)
- Security requirements
- Service Consumers and Contact Information
- Service Documentation and specifications for the service needed by system designers and programmers (For example, WSDL, XSD and other artifacts)
### Service Versions

This is the collection of Service Versions present in the registry. A Service Version represents a specific version (or release) of a Service and provides a range of functional and non-functional specifications that hold for that version of the service. The Service Version exposes its capabilities as service level definitions. It may also (in the case of a composite service) identify the services it depends on by defining Service Level Agreements to the Service Level Definitions provided by the consumed service.

#### Preferences

<table>
<thead>
<tr>
<th>Select</th>
<th>Name</th>
<th>Graph</th>
<th>Namespace</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Eligibility service</td>
<td></td>
<td></td>
<td>1.0</td>
<td>Service to determine customer account eligibility</td>
</tr>
<tr>
<td></td>
<td>Get Academic Groups</td>
<td></td>
<td></td>
<td>1.0</td>
<td>This service allows you to enter a term and then it returns the academic groups for the term. It only returns the schools that are participating in the Class Descriptions, so LAW and Medical School for example are not included.</td>
</tr>
<tr>
<td></td>
<td>Get Class Detail</td>
<td></td>
<td></td>
<td>1.0</td>
<td>This service is a little different because it returns two methods (service operations). The method named NWCD_ALLCLS_SERV_OPR.v1 returns all classes for a subject regardless of whether the class has a class description attached. The method NWCD_DESCR_SERV_OPR.v1 returns only classes that have class descriptions attached. The request is the same for both.</td>
</tr>
<tr>
<td></td>
<td>Get Freshman Seminars</td>
<td></td>
<td></td>
<td>1.0</td>
<td>This Service returns all Freshman Seminars for a term. Freshman Seminars have no associated classes so we don’t code for that in this Service. Special characters do come through. Encoding is UTF 8 or 16.</td>
</tr>
</tbody>
</table>
Get Academic Groups

Description
This service allows you to enter a term and then it returns the academic groups for the term. It only returns the schools that are participating in the Class Descriptions, so LAW and Medical School for Version
1.0
Consumer Identifier
Version Availability Date
Wednesday, June 4, 2014
Version Termination Date
Tuesday, June 30, 2015
Version Requirements Link
1.0
Asset Web Link
Remote State
Owner Email
v-povle@northwestern.edu

Graphical View
Applied Policies
Applied Policy Attachments

None
SES_acad_groups
None
Student Enterprise Systems
None

None
None
None

Chartered Business Capability(s)
None
Consumer DOU(s)
None
Provider DOU(s)
None

Identified

None
Producing & Consuming Web Services
Producing Services

• There will be an approval process for registering services that will include the data steward
  – Producers can decide if they want their service visible to end-users
  – But, it will be important to have services registered

• There will be a web form to make the request
  – The CoE will follow up to gather more technical information (e.g., SSL cert info, endpoints, required credentials, etc.) and help facilitate firewall changes
Producing Services

• There may be follow-up about adding content to make the service more re-usable

• Producers can request transformations to another format

• Over the long-term, producers will be responsible for maintaining the services, versioning the services, ensuring they are available, announcements on removal of services or versions of services.
Consuming Services

• There will be an approval process for consuming registered services
• There will also be a process to request services that do not yet exist
• Both can be requested using a web form
• Consumers of services will have to have a service account for their system to authenticate with ESB
Consuming Services

• Data stewards will have to approve the consumers of a service
  – Consumers may be required to sign a Terms of Service agreement

• The CoE will help facilitate the acquisition of service accounts, firewall rules, SSL cert configurations, etc.
  – The CoE will register the consumer as a certified consumer of the service so that the ESB will honor its requests
More on the Center of Expertise

• The CoE will run the infrastructure
• The CoE will register services
• The CoE will register certified consumers
• The CoE will provide transformations, service compositions and decompositions
• The CoE may be able to provide special connectors where web services are not possible
More on the Center of Expertise

• The CoE can assist with advice on best practices, coding techniques, and configuration
• The CoE can provide code samples
• The CoE will be able to assist on system selection to ensure that new systems can participate in the SOA ecosystem
Q&A
Wrap Up
True Opportunity for Federation

- Consumers will exist in many forms in the application ecosystem, including:
  - Enterprise applications consuming the services of other enterprise applications (e.g., FASIS and NUFinancials)
  - Enterprise applications consuming the services of unit-developed solutions (e.g., FAMIS and Housing)
  - Unit-developed solutions consuming the services of enterprise produced services (e.g., Wildcard and Library)
  - Mobile/portal developers using services to expose features
  - Public web services may also be consumed
In Closing

• SOA is about Community
• Success will depend on everyone in the Community
• We will provide services to each other
• We will help each other
• The whole will be greater than the sum of the parts