Service Oriented Architecture (SOA) Initiative: Kickoff Forum

June 9, 2014
Welcome

• Greeting
• Objectives
• Format
• Logistics
• Jump In!
Forum Objectives

• **Educate** via brief overview about best practices in use of services/enterprise architecture
• **Inspire** and **catalyze** by providing examples of work to date or coming soon
• **Discuss** services that would add significant value
• **Call to action** to do it, including how
Today’s Format

• SOA overview and foundation
• Sample NU use cases for SOA
  – Discussion at tables between use case presentation
• How to move forward
• Lunch
• Technical sessions
Logistics

- Coffee, etc. at back of room
- A morning break + lunch
- Restrooms in hall
- Presentations will all be made available
- Please use discussion periods, Q&A, lunch and any other opportunities to network, brainstorm, more
- We’re going to keep moving...today is one day and what we’re discussing is the new normal!
Thank You

• SOA Adoption Working Group
  – Alan Wolff  
  – Ann Dronen  
  – Andrew Ludington  
  – Dan Johansson  
  – David Keown  
  – Harry Samuels  
  – Jason Schober  
  – Kalpesh Patel  
  – Michael North  
  – Tom Board

• Today’s presenters
• Event organizers

IT@NU
Transformation Time!

SOA

FROM...

TO...
Service Oriented Architecture (SOA): What is it? Why is it important?

Jason Schober
(my preferred title was already taken...)

Technical knowledge required for this presentation:

NEXT TO NONE
SOA in your daily life:
You want to book a plane ticket

So you drive to the travel agency
You want to book a plane ticket

So you call a 1-800 number

Your call is very important to us. So please enjoy this 40 minute flute solo
You want to book a plane ticket

So you go to one airline’s website

❌ ❓
You want to book a plane ticket

So you go to a 3\textsuperscript{rd} party website and search all airlines, all the fares, even special offers

YES?
You want to book a plane ticket

So you use your mobile device

YES
You want to book a plane ticket

And you expect the airline to tell you if anything changes

YES!
How does this all happen?

1. It might be magic

2. Gremlins have been speculated to be involved

3. You might say “that’s just how the Internet works”

4. And you’d be correct

And, you’re also invoking web services all along the way (it’s SOA...you just can’t see it)
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*
SOA in the Enterprise:
Imagine you are a large phone company

• You acquire a smaller company

• They have their own systems (e.g., procurement, accounts payable, general ledger, reporting)
Imagine you are a large phone company

- You already have these business systems, too
- You only want one GL for reporting
- Do you replace all their systems? $$$$$ NO!
Imagine you are a large phone company

You use SOA to integrate the applications!

... and then acquire more phone companies!
But, ... do we need this at Northwestern?
Drivers for Integration via Services

• Increased demand for integration of systems, data
• Desire to reduce duplicative data entry
• Demand for workflows that cross system boundaries
• Plans to move numerous new functions online via implementation of new modules or features
• Need to reduce burden on administrative systems teams of developing/maintaining data integration
• Increased partnerships on projects with IT@NU requires different methods for data access
NU Use Case Example

- **Problem**: Employee separation or move within organization must be discovered by system administrators and addressed on a case-by-case basis

- **Solution**: Use publish/subscribe web services to alert systems of employee status change and allow each system to manage its response
Publish Employee Status Changes to Subscribing Systems

1. Status change with emplid, etc.

2. Previously subscribed services are invoked

3. Each alerted system takes appropriate action

Registry

Enterprise Service Bus

FASIS
NUFin
SES

Msg Broker

Deans On Call

YOU ONLY NEED TO UNDERSTAND THE CONCEPT...
More event-driven process examples

- Student has matriculated
- Student has withdrawn
- Student has graduated
- Grant has been awarded
- Staff member has been hired
- Staff member changes jobs
- Faculty member has been appointed
NU Use Case Example

• **Problem**: Lag in propagating student standing from SES to learning management system (LMS) hampers student’s participation in class

• **Solution**: Web service call from student system to LMS can change student enrollment and registration data immediately upon committed change in student record
More real-time process examples

• Drop student registration in course
• Add student registration in course
• Check Wildcard is valid
• Check chart string is valid
• Authenticate NetID & password

Make information available in ways we haven’t even imagined yet!
Services at Northwestern Now

- **Services are not** entirely new to Northwestern
  - Online alumni community. Our Northwestern leverages real-time integration between CATracks, other systems.
  - NUFinancials integration to iBuyNU for transmission of orders, intake of invoices, transmission of payments.
  - FSM faculty database communicates with FASIS.
  - PeopleSoft administrative systems use messaging to send incremental changes (e.g., new user, name change, COA values) multi-directionally.
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
An example of orchestration via SOA

Problem: Students need to be reimbursed for club expenses, but process is paper-based and requires manual routing for signatures, receipt review

Potential Solution: Use a platform we own or build a custom to bring process online with workflow

- OnBase as alternative to PeopleTools due to ease of deployment, desire to avoid customization of delivered software
What are the goals?

Business requirements:

– Eliminate paper forms and intercampus mail
– Reimburse students within shorter duration
– Be able to track status of reimbursement requests throughout process
– Preserve step of Dean’s Office review for appropriateness
– Retain supporting documentation to substantiate reimbursement
– Maintain sufficient internal controls to prevent fraud, etc.
Prior process

1. Student incurs expenses
2. Student fills out paper reimbursement request form
3. Student assembles paper receipts
4. Student hand delivers or inter-offices paperwork to approver
5. Repeat if there is more than one approver
6. Approved requests sent to “AP” for disbursement
7. Transaction rekeyed into financial system
8. Student gets their reimbursement
9. Bank reconciliation validates check/payment/etc.
Ripe for overhaul

1. Student incurs expenses
2. Student fills out paper reimbursement request form
3. Student assembles paper receipts
4. Student hand delivers or inter-offices paperwork to approver
5. Repeat if there is more than one approver
6. Approved requests sent to “AP” for disbursement
7. Transaction rekeyed into financial system
8. Student gets their reimbursement
9. Bank reconciliation validates check/payment/etc.
New and improved process

1. Student incurs expenses
2. Student initiates reimbursement request online
3. Student attaches electronic receipts to request
4. Student routes request to approver (online workflow)
5. Approver routes if there is more than one approver
6. Approved requests sent to “AP” for disbursement
7. Transaction rekeyed into financial system
8. Student gets their reimbursement
9. Bank reconciliation validates check/payment/etc.
Has this process been optimized? What might we be missing?

1. Student incurs expenses
   – Are they an appropriate consumer of this service?

6. Approved requests sent to “AP” for disbursement
   – This is done in Excel...is that the best we can do?

7. Transaction rekeyed into financial system

8. Student gets their reimbursement
   – Happy to get money! Check? Direct deposit? Are they notified?

9. Bank reconciliation validates check/payment/etc.
   – Did we add another source against which to reconcile?
What If???

• We could validate authorized consumers of service? Or, at least authenticate them with NetID?
• We could eliminate the manual data entry?
• Funds were sent via direct deposit?
• Requester was kept informed throughout...maybe even on his/her mobile device?
• We could bring more similarity to the consumer Internet to this user experience?
An optimized architecture and flow

1. User initiates request for $$$
3. User submits request for $$$
4. Approvers say yes

2. Are they authorized to do so? Are their login credentials ok?

5. AP Voucher import service invoked
6. Nightly pay cycle jobs transmit ACH payments

7. Money in student’s account.

8. Student notified.

REAL TIME

OnBase -> IAM -> NUFin

Chase Bank

Registry

Enterprise Service Bus
There will be a learning curve

• We should not simply replace a paper form with an online form
  – Attachment of imaged receipts, incorporation of workflow big wins in this example (beyond moving process online)
• Need to look upstream and downstream, too
• Need to think about whether solution could meet more than one units’ needs
• What is optimal state? (Phasing may be needed)
Recap

• SOA will be transformative in its impact to our administrative systems landscape:
  – No more need to replicate data across systems (transfers)
  – No more need to wait 24 hours or more to use data from another system (overnight batch)
  – No more fragile database links
  – No more one-off projects to do the above

• Today’s forum is about expanding awareness, not announcing a final product

• We’re already underway!
On to the Use Cases!

A series of use cases will be presented to help inform you of current demands and activities underway

- Three use cases
- Then discussion at your table
- 15 minute break
- Two more use cases
- More discussion at your table