Welcome

- Greeting
- Objectives
- Format
- Logistics
- Jump In!

“Why are we here”

I don’t know. Frankly, I’ve wondered that myself.
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
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
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*
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

- FASIS
- NUFin
- SES

Registry

Enterprise Service Bus

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 $$$
2. Are they authorized to do so? Are their login credentials ok?
3. User submits request for $$$
4. Approvers say yes
5. AP Voucher import service invoked
6. Nightly pay cycle jobs transmit ACH payments
7. Money in student’s account.
8. Student notified. Student 😊😊😊

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
Yes, You Can Buy That Valid Chart Strings

Brian Peters
Jason Schober
Use Case

• University Services provides over two dozen services to the NU community
  – Mail
  – Motor pool
  – WildCARD
  – Travel
  – Parking
  – Moving
  – Lab services
  – Shuttles/charters
  – Copiers
  – Vending
  – Fitness/recreation
  – Stockrooms
  – Lab laundry
  – Equipment calibration
  – ...and more
Use Case

• These services can be requested online, via phone or email, in person, or on paper
  – Varying application sitting behind each function

• NUFinancials requires a valid chart string
  – Fund – 141 values
  – Department – 3,509 values
  – Project – 20,096 values

Almost 10 billion permutations
…but they’re not all valid
…is this one? Just trust me?
What Could Go Wrong?

• In most cases, there is no real-time validation that a chart string is valid
• Transactions are compiled for billing periodically
• Invalid chart strings are discovered
  – Combination of Fund-Department-Project doesn’t exist
  – Combination does not have a budget for this FY
  – Combination does not have sufficient funds
  – Project is a grant that has been CLOSED
The Workload of Invalid Chart Strings

- Determine the root cause of the invalid chart string
- Find somebody from whom to get a valid chart string
- If not fixed prior to billing, post charges anyway and then do transfers
- Not productive...
Somebody should validate these things...

• Chart string are validated today
  – Several interfaces from NUFinancials directly to other major systems (FASIS, SES, FAMIS, CATracks, etc.)
  – Daily file of valid chart strings made available via means of differing technical sophistication
    • XML file for import
    • Excel/CSV file for import
    • Cognos output to download

• But this assumes upkeep on the consumer’s part...
Chart String Validation using SOA

• Multiple services may make sense to meet the varying needs of different applications:
  – Valid chart string value – transaction
  – Valid chart string value file
  – Tentative budget check – transaction
  – Valid ChartField value – transaction
  – Valid ChartField value file
Valid chart string value – transaction

• A consuming system would inquire as to whether a Fund-Department-Project combination is valid
  – Has a budget in the current fiscal year (could be $0)
• In this mode, the inquiry would expect a pass/fail or yes/no answer in the course of validation in the consuming application
  – Ideally at the time that a transaction is first entered/captured in real time
Valid chart string value file

• Similar inquiry by consuming application vs. universe of valid Fund-Department-Project combinations
  – In this mode, the inquiry would likely query against a data set with either a pass/fail answer for each queried combination or a list of invalid combinations
  – A use case for this mode would include a need to validate a set of monthly charges in a recharge system prior to passing them to NUFinancials via a journal entry upload process
    • The journal entry upload could be its own web service!
Tentative budget check – transaction

• In most cases, strict budgetary control is not enforced in NUFinancials
  – Feinberg chart strings, some grants are the exceptions
• PeopleSoft Financials delivers a function to perform a “what if” budget check
  – Same as “Valid chart string value – transaction” use case, but would return a result of Valid, Warning, Error
  – If called in real-time, could prevent budget error before it happens (e.g., when entering funding in FASIS)
Valid ChartField value – transaction

• Systems may also wish to consume information about the values of a single ChartField
  – Department: ID, description, manager, admin contact, default fund
  – Project: ID, description, manager, admin contact, default fund, default department, purpose, start date, end date
  – Account: ID, description, account type
  – Fund: ID, description, fund category
Valid ChartField value – transaction

• In real-time, an application might inquire about attributes of a ChartField
• For example, for a Project ID
  – To determine the title of the project id
  – To understand the intended purpose of the project
  – To locate an individual to contact for approval prior the use of the project (or verify that the requester is a known approved used), etc.
Valid ChartField value file

• Much the same as above, a system may wish/need to consume the information about each ChartField in a periodic fashion
  – Determining whom to direct inquiries to if transactions are found to be invalid when preparing to send a monthly recharge file to NUFinancials
  – Other units may find this data valuable for other reasons, such as an input to a report or a simple mail merge task
Opportunities for re-use

• We all use the same chart of accounts
• Opportunity to be more timely, more precise in our validation
  – Get expenses posted to the right place the first time!
• Several unique interfaces could be retired if moved to published services via enterprise architecture
SOA Use Case:
Introducing NU Learn

June 9, 2014
Introducing “NU Learn”

• A training management system for the University.
  – Faculty, staff and students

• Training programs that focus on business practice, compliance and staff development.
  – Replace existing solutions for research compliance

• A comprehensive, long-term solution that will evolve as an expanding circle into an enterprise wide system.
What is Saba?
Saba is an Island
Found in the Caribbean
Saba is also a Hosted TMS Solution
Why Now?

- Training needs—driven by compliance obligations and organizational efficiency—are ubiquitous in a complex organization like NU
  - Our strengths have fundamental weaknesses--less than ideal efficiency in how we convey and document “the right way”
- Stand-alone solutions are awkward and create recordkeeping silos.
- The need today is acute—the technology is transformative & has enormous potential
But No System is an Island

Saba

Facilitates required training/job competencies
No System is an Island

But who needs to be trained?
No System is an Island

What roles do they fill?

Saba
No System is an Island

What unit(s) do they work in?

Saba
No System is an Island

What classes are they required to take?
No System is an Island

Can I get a view for that?
No System is an Island

No, you can have a service.

Saba
Service or Event Message?

That is the question!
It’s a question of “Fit”

• Saba is architected as a silo
• BUT, it supports event messaging
• Given the high use potential of NULearn, messaging looks appealing:
  – Less traffic to enterprise data providers
  – Less exposure to downtime
It’s a question of fit

Data
- Stored in NULearn
- Sourced from Authority
- Updated in real time
Challenges

Disambiguation of identity

SES

FASIS
Challenges

Primary role
Solution

SES → FASIS → ISIS → Integration App → Saba

Enterprise Service Bus
Solution

Enterprise Service Bus

SES  FASIS  ISIS  Integration App  Saba
Integration App

• Applies business rules
  – Specific to NU Learn
  – Disambiguates ID OR notifies for action
  – Sets role “trumping rules”
• Notifies NU Learn of changes
• Is loosely coupled
  – Changes will not impact other consumers of the event messages
Timing

• A partnership among friends & allies
  – NUIT
  – HR Information Systems
  – Office for Research/ ORIS

• Phase 1 is a commitment to stand up research safety training & create a common look to OR training.
• Phase 2 and beyond include a wider application of the system across campus with more enable features.
SOA Use Case: Online Forms

How SOA can enhance the Online Forms application

SOA Forum

June 9, 2014
Introductions

Dan Johannsson
Manager of Technical Solutions
Weinberg IT

Adam Finlayson
Manager of Functional Solutions
Weinberg IT
What is Online Forms?

Web application developed by Weinberg IT through the Federated Service Model, to solve a university-wide paper forms problem.
What does Online Forms do?

- Designed with multi-tenancy in mind
- Multi-section forms with workflow
- Sections can be assigned to people or groups
- Workflow assignment based on selected or entered data
- Optional sections
- Per-section visibility rules
- Email notifications
- File attachments
- Auto-population of certain fields from LDAP directory
- Archival

But, it intentionally doesn’t do everything!
What is Online Forms currently used for?

- Interschool Transfer
- Dual Bachelor’s Degree

More forms are in the making:

- Transfer Credit
- Study Abroad Credit (Weinberg)
- Add / Drop Major
Solved one problem, but created another

Once forms were online:

- Easier to submit and fewer barriers
- More submissions (double the usual volume for IST form)
- Increased human transcribing work to system of record (SES for IST form) at the end of workflow, due to increased submissions
Service enabling Online Forms to reduce manual transcription and errors

This sort of transcribing data from one system to another is the perfect problem to solve with services.

Once we add the ability to initiate service calls from Online Forms we can push data to other systems or initiate actions in other systems.

This will solve the problem of additional workload on people from the increased demand for forms, and should eliminate errors from a person transcribing data from one system to another.
Add/Drop Major Form: Initiating action in SES

We will be using the Add/Drop Major form as the first use case where we initiate an action in an external system (SES) using a web service call.

At the end of the Add/Drop Major form workflow, one or more of a student’s majors should be dropped in SES, and one or more majors should be added.

If the submitted Add/Drop Major forms grow to 1,000 once put online, then this will be a huge savings in time. Even if they stay at 500, then it is still a worthwhile savings.
Other service enabled possibilities

The possibilities are endless for service enabled Online Forms:
• At any stage of a workflow data can be pushed to an external system, data can be pulled from an external system, or an action can be initiated in an external system.
• A follow-up form could be triggered in Online Forms.
• Forms can be initiated by calls from external systems, triggered by events or data changes. User is notified and then completes form.
• Online Forms system could offer data to other external systems, perhaps custom dashboard or reporting tools.
• An archival copy (TIFF, PDF, other) of a form could be pushed to an imaging archival system.
Small Group Discussion

• Please use the paper provided at your table
  – What services would you wish to consume?
  – What services would you be able to produce?
  – What questions do you want to raise during Q&A?
• Document your major takeaways thus far
• Please leave paper at your table as an artifact of the discussion
BREAK
A Collaboration Project: Mobile Student Data Retrieval Application for Student Affairs “Deans on Call”

By:
Stephen Brawn
Christopher Walker
Anthony (Tony) Kirchmeier

NORTHWESTERN UNIVERSITY
Dean on Call (DOC) staff members serve in emergency/crisis management situations regarding NU students 24/7/365.
Dean on Call (DOC) - rotation staff

- All DOC work in the Division of Student Affairs:
  - 2 Assistant Directors
  - 6 Directors
  - 1 Executive Director
  - 4 Assistant Deans
All DOC directly report to:

- AVP/Dean of Students, Todd Adams
- AVP-Student Engagement, Burgwell Howard
- AVP-Student Auxiliary Services, Julie Payne-Kirchmeier

These three AVPs report to Patricia Telles-Irvin, Student Affairs-Vice President.
STUDENT INFO APP

- Dean-on-call
- Rotation

NORTHWESTERN UNIVERSITY
DEMO
Developer Takeaways

• A simplification of web services allows us to focus on the issues we really care about: the end-user experience

• A concentration of reusable components allows for flexibility in form factor (mobile, desktop, etc)

• The adoption of a SOA platform presents the possibility of greater collaboration between back-end and front-end engineers
Takeaways

• An agile development attitude and direct contact with the customer allowed us to use our knowledge of SES data points to design an effective product that really serves the customer’s needs.

• Listening to Tony give scenarios where this product would be useful to them in emergency situations helped us really understand what they needed and helped us become invested in working together to get the best streamlined product available to solve their problem.
Outcomes

- Decreases time to retrieve enterprise student data (decreases DOC stress).
- Helps DOC focus on the crisis situation and coordination with stakeholders.
- Ultimately helps to increase student safety.
SOA at NUL, Weinberg and McCormick

Michael North – University Library
Dan Johannsson – Weinberg
Alan Wolff – McCormick
SOA at NUL

• We can give you data!
  – Collections of audio, video, images and catalog records.
  – Content from other parts of campus could be published in the library’s central search.
  – Name, Place, Subject Authority Control Services could be consumed by various applications on campus.
  – Subject-based lists of databases, journals and subject guides could be consumed by departmental or research web sites.
SOA at Weinberg

• **We can give you data!**
  
  – Weinberg undergraduate interests and career goals could be used by departments outside of Weinberg for targeted communications for special interest events.
  
  – Weinberg academic awards are sometimes granted to non-Weinberg students. Their schools may want to know about this.
  
  – Advising notes from Insight and Dossier could make it into SES.
  
  – External systems could get the status of and results of a poll in Online Elections system.
  
  – Online Forms could expose data for custom reporting and dashboards.
SOA at McCormick

Faculty Research Areas, CVs, Publications:

- Birmson, L. Catherine
  - James B. Cleen Professor of Mechanical Engineering
  - clemson@northwestern.edu
  - Multiscale mechanics of advanced materials, including shape memory alloys, polymer nanocomposites, and tribotribial synthesis, characterization and modeling.

- Broadbelt, Linda J.
  - Chair and Professor of Chemical and Biological Engineering
  - Sarah Rebecca Roland Professor
  - brob@northwestern.edu
  - Multiscale modeling of complex reacting systems; design of nano-structured catalysts, discovery of novel biochemical pathways, polymerization and depolymerization kinetics.

- Burgardt, Wesley R.
  - Charles Deering McCormick Professor of Teaching Excellence
  - Professor of Chemical and Biological Engineering
  - Polymer science, rheology, non-linear fluid mechanics.

360° Leadership Assessments:

Assessments for classes with group projects:

- 360: Overcoming Adversity and Failure
- Self Assessment
- Outside Rater
- Difference

McCormick Seminars:
Small Group Discussion

- Please use the paper provided at your table
  - What services would you wish to consume?
  - What services would you be able to produce?
  - What questions do you want to raise during Q&A?
- Document your major takeaways thus far
- Please leave paper at your table as an artifact of the discussion
Message from Sean Reynolds, VPIT/CIO

David Keown
I’m Sold, What’s Next?

Harry Samuels
And Why Are We Here Today?

• Most frequently asked questions on RSVP form were:
  – What’s being developed and when?
  – When will you require us to switch to services?

• This forum is not about announcing a final solution with hard deliverables and time lines.
  – It’s a milestone in the process
  – It’s about expanding awareness and engagement
  – It’s about expanding the SOA community
  – It’s about rallying the community
For the 5 W’s and the H
Why?
Multiple Business Reasons

• Reduce effort over time
• Enable more business to move online
• Enable real-time interaction
• Make it easier for services to move into the cloud, and the institution to function at scale without hands-on intervention
• Enable more efficient and more productive business processes
What?
New Functionalities

- Eliminate data integration via batch data file exchanges
- Keep data consistent between systems
- Connect workflows to enterprise systems of record
- Use composite apps to combine data in new, easier to use ways
- Expose functions in NUPortal
Who?
IT@NU

• Services will be built, maintained, and supported by the unit who owns the application being connected via a service, i.e... the IT@NU community

• Defining key services will be a joint effort between producers and consumers

• Creating a SOA Community of Practice will be key
  – Share SOA best practices and code samples
  – Articulate ideas and common needs
  – Provide mutual assistance
Who?
The Role of NUIT

• Provide support for the SOA infrastructure
• Provide a “Center of Expertise” for the University (NUIT’s Application & Development Integration team - led by Kalpesh Patel - will consult with the community on:
  – Application system product selection for its fit within the SOA environment
  – Programming techniques and configuration issues
  – Best practices for writing services (including code samples)
  – System architectures that will most effectively leverage the SOA environment
How?
Northwestern Roadmap for SOA

1. Convey the vision for future application integration, data sharing, and business process improvement
2. Build distributed skills and collaboration
3. Create a framework to ensure and service sustainability, service reusability, and data integrity
4. Find ways to grow services architecture while still addressing Line of Business needs
5. Have services be the default method of data exchange by end of FY2015
How?

Build Awareness, Alignment, Timelines

• Work has already started
  – Engagement with governance committees, infrastructure, service design, administrative controls, training options

• Now building awareness of goals and timelines, with:
  – System implementers
  – Data stewards
  – The IT@NU community
  – Any units buying or building new software
How?
Build distributed skills and collaboration
How?

Training

• Prospective training providers are being vetted now by a multi-unit working group
  – Funding for training comes from participants

• Because of the diversity of programming languages and frameworks at NU
  – Classes to be offered on SOA concepts and methodology, not programming “how to”. Target: starting in July.
  – Will use the SOA Community of Practice to ramp up our programming skills within the community
How? Build Collaboration

• Create and grow a SOA Community of Practice.
  – SOA website and mailing list for communications about the initiative

• Continue to use multi-unit working groups with connections to the IT@NU community
  – SOA Initiative Steering Group
  – Training Partner Working Group
  – Identifying key strategic services to be built

• Remain closely connected to IT Governance
How?
Create administrative processes and controls around the production of services

• Goals:
  – Ensure that a contract is understood between the publisher and consumer of a service
  – Services are reused rather than one-offs just multiplying

• There will be an approval processes for, and a web form to request:
  – Registering services
  – Using services that have been registered
  – Requesting services that do not yet exist
How?
Find ways to grow SOA while still meeting LOB needs

• Focus on Business Opportunities
  – Leverage tactical opportunities.
    • Do projects differently. Where possible, prioritize projects that further the SOA strategy.

• New enterprise system upgrades
  – Instead of investing effort in testing data interfaces, invest it in building services to replace them.

• New systems
  – Look for systems with services and APIs already built into them.
When?
This won’t happen overnight, but it’s already started

GOAL: Have services be the default method of data exchange by end of FY2015

Projects already in Progress

• Student system requests
• Research systems
  • eIRB re-implementation
  • Training Management System
• Sponsored award data service (SESP request)
• Library ALMA implementation
  • Identity management
  • Purchases/payments (NUFinancials)
  • SES/FASIS
• NUFinancials chart of accounts data
When?
Next Steps

• Finalize administrative controls processes with the data stewards

• Identify key strategic services for enterprise systems
  – Collaboration between producers and consumers

• Develop roadmap for getting these services into production
  – Which ones will come from the leveraged tactical opportunities?
  – Which ones need to targeted independently?
What Should I Do Next?

• Understand that this is the new normal
• Continue this discussion...over lunch, back at your office, within IT Governance, invite us to visit
  – What services can I make available to the rest of the University community?
  – What services would the University community like for me to make available?
  – What services would I like to have?
  – How can I use SOA to create better business processes for my unit?

• Look for a Community of Practice announcement
Q&A
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
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
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.)
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
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
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.
Message Broker (MB)

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
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.

<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>
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