



*University of Nebraska's*

*The* PETER KIEWIT  
INSTITUTE

Technology Development Corporation

# Service Oriented Architecture Best Practices and Lessons Learned

**Rick Brown**

**Department of Computer Science**



# Overview

- Background
- PKITDC White Paper
- Seven Lessons Learned
- Recommendations
- Conclusion



# Background

- ISPAN A&I Chief Engineer asked PKITDC consultants for lessons learned research
- GNCI(GOC-CE)/C2SES Critical Event Web Service Issues:
  - Technical infrastructure
  - Simple issues that took a long time to figure out,
  - Work arounds had to be developed.
  - Tools, compiling, versioning, deploying, and operating the services themselves.
  - Incomplete standards
  - Service development methodology, cross-team interaction, etc



# PKITDC White Paper

- Reports findings from business/industry:
  - Successful approaches and motivation for SOA
  - Best practices in SOA (lessons learned)
  - Governance and management
  - Standards and specifications
  - USSTRATCOM perspective
- Offers:
  - Recommendations



# SOA Definition

- A service is software that performs work on behalf of some other entity, such as a user or another service.
- A service-oriented architecture (SOA) is the underlying physical and organizational structure supporting communications between services



# Lesson 1

## *Mission Objectives Must Drive the Adoption of a Service Oriented Architecture (SOA)*

- SOA is not an end in itself.
- Mission/Business objectives must motivate the adoption of SOA
- Insightful examples:
  - Retail organization: needed better infrastructure
  - Manufacturing organization: needed immediate functionality



## STRATCOM Perspective

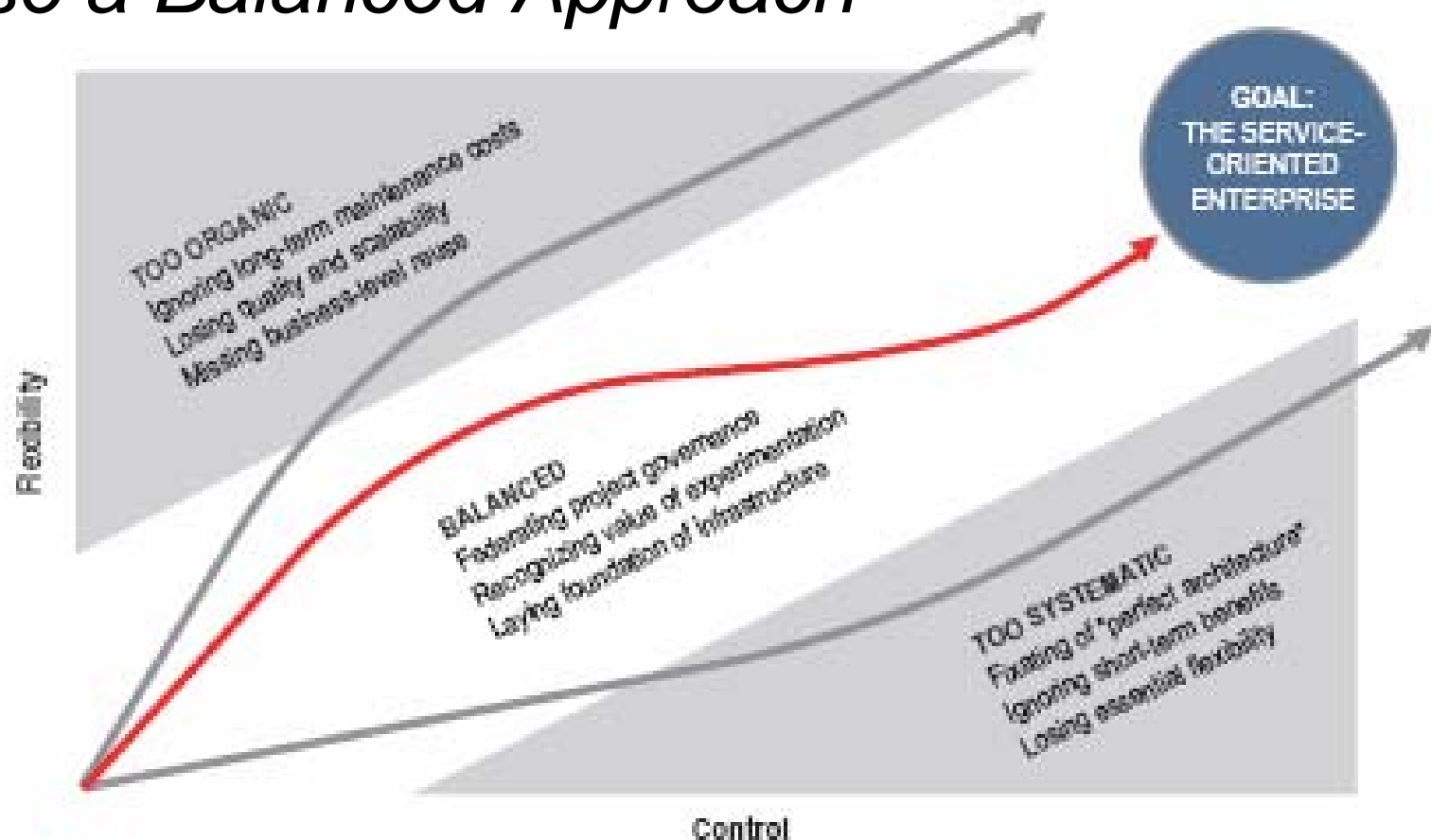
Mission/Business objectives must motivate the adoption of SOA

- Gen Cartwright –
  - Maneuver faster, access to data, rapid acquisition, gain an advantage
- Like the Retail organization: need a better infrastructure
- Some of the pilot projects seem like the Manufacturing organization: need immediate functionality



# Lesson 2

## *Use a Balanced Approach*





## STRATCOM's Approach

- *Pilot projects, programs are implementing services*
- *Governance processes have been created*
- *We have not identified a comprehensive architecture for implementing or managing the services*



## Lesson 3

*SOA Adoption Is a Journey of Best Practices  
involving Technology and Culture*

*Common Phases:*

- *Education*
- *Experimentation*
- *Execution*
- *Embrace*



## Lesson 3 continued

### SOA Adoption Checklist:

- **Choose and prioritize services to develop**
- **Plan to incrementally develop valuable services**
- **Focus on business processes.**
- **Define the formal interfaces between service providers and service consumers (not just end users)**
- **Design for Reuse - Effective communication and collaboration**
- **Recognize that as the scale grows, formal governance is essential.**
- **Establish ownership of services**



## Where is STRATCOM on the Journey?

- **Challenge: Incompatible Tools**
  - **Best Practice: Verify the compatibility of toolsets.**
- **Challenge: The Implementation Environment Is Difficult to Test**
  - **Best Practice: Adopt a multi-tiered implementation approach**



## Where is STRATCOM on the Journey?

- Experimentation
  - GOC-CE(GNCI) pilot
  - Other pilots (Blog input service)
- Education
  - USSTRATCOM is engaged in educational venues such as the developer's forum, SEA&IT meetings, etc.



# Lesson 4

## *Recognize Unique Challenges of Adopting SOA In Government*

- **Understand the mission processes deeply**
- **Coordinate the implementation and architecture teams closely**
- **Design for re-use (but don't expect > 30% - 40%)**
- **Focus on SOA training, education and skill-refreshment**
- **Select strategic, long-term projects**
- **Recognize importance of governance across organizational boundaries**
- **Use pilot implementations to increase skills**



# STRATCOM's Unique Challenges

- Financial issues –
  - Who will fund the shared requirement that comes from a COI?
  - How do we change from an industrial process for acquisition to a more flexible, less time-consuming process?
- Cultural issues –
  - Should there be separate directorates in charge of application development, operations and management?
  - Can different COIs communicate and negotiate common requirements across organizational and even Service boundaries?



# Lesson 5

## *SOA Intensifies Management and Governance Challenges*

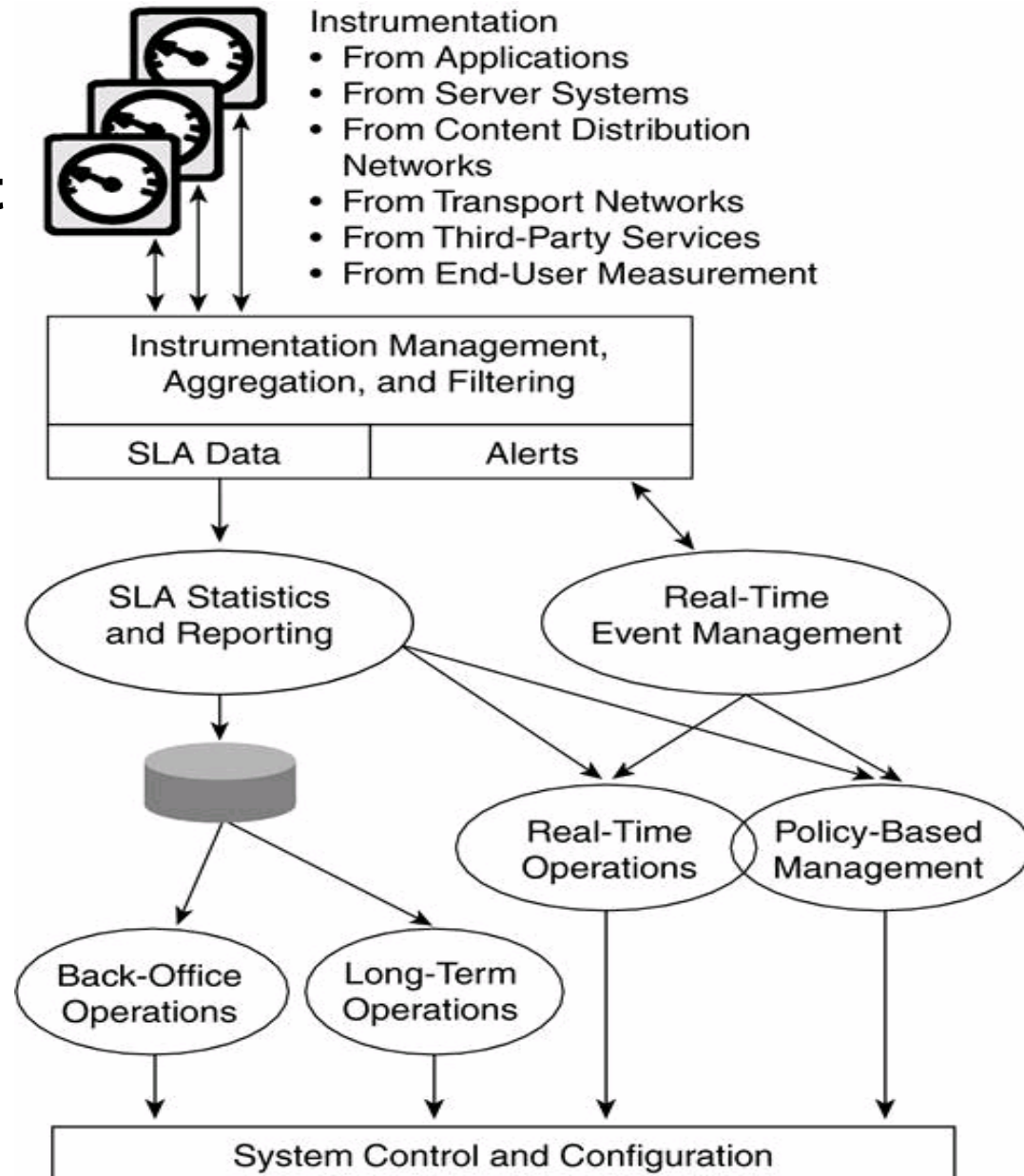
### **SOA Service Level Management (SLM):**

- **Services can support many applications**
- **Service Level Agreements (SLA)**
  - for services provided
  - for services depended upon



## Lesson 5 continued: Service Level Management Architecture

from *Practical Service Level Management: Delivering High-Quality Web-Based Services*. by McConnell & Siegel (2004).





# SOA Service Level Management at STRATCOM

- Trouble-ticket architecture
  - Static
- Need a well-designed, well-documented infrastructure



## Lesson 5 Continued

### Governance Challenges:

“In 2006, lack of working governance mechanisms in midsize to large ... post-pilot SOA projects will be the most-common reason for project failure.” Paolo Malinverno, The Gartner Group

#### Who decides

- What services?
- What service levels?
- Who develops, maintains?

#### How do you

- Avoid duplication?
- Increase reuse?
- Meet consumer requirements?



## Lesson 5 Continued

### Governance Policies:

- Define who makes decisions
- Define the framework of processes and organizations in which the decisions are to be made.
- Include a process for defining, designing, executing and maintaining reusable services while avoiding duplication.
- Determine service ownership, including cost-allocation

### SOA without Governance:

- Wild West SOA – No Flexibility
- Duplicated SOA
- Shelfware SOA



## SOA Governance at STRATCOM

- STRATCOM Information Services Committee (SISC)
  - Challenge: seen as unnecessary overhead.
  - The intent of the SISC is to pull parties with interest in common requirements together to find a solution that meets their needs, and other needs within a Community of Interest.
- The challenge of creating *buy-in* or *demand* for governance may be one of the most significant.



# Lesson 6

## *Avoid Death by Governance*

- *Governance is valuable, but costly*
- *Many services may never be reused (local services) outside the local domain*
- *There must be a process to differentiate between globally available, highly reusable services and local services*



## Local vs. Global Services at STRATCOM

- **Threshold of Reusability**
  - This could help project level developers to make objective decisions about the applicability of the governance mechanism for a proposed function
- **Keeping Metrics such as Reuse Levels.**
  - It could also give incentive to make the distinction between global and local services.



# Lesson 7

## *Standards Allow Interoperable Services*

- *W3C and OASIS publish standards*
  - *WS-I Basic Profile (DoD Mandate)*
- *Some specifications are not yet published standards*
  - *Stay abreast of the technology (still immature)*
  - *Select these with care*
  - *Plan for migration to eventual standard*



## Interoperability at STRATCOM

- WS-I compliance and careful selection of emerging standards, though necessary, will not be enough to ensure interoperability.
  - Each vendor has its own WSDL generating and parsing implementation (for example)
  - There are bugs and incompatibilities between them, some of which have not yet been discovered
  - This issue can arise between any two vendors
- Interoperability will require good schema design practices (including Standards compliance) and experimental verification, not merely vendor selection.



# Recommendations

## *Architecture*

- *Design an enterprise-wide architecture for implementation, management and governance*
- *Utilize a multi-tiered implementation approach to allow development, staging and evaluation, and production environments for services*

## *Incremental Adoption*

- *Implement the most valuable priorities for pilot projects*
- *Pilot the governance and management*
- *Implement utility services*
- *Define and model high-level mission processes for implementation as services*

## *Governance*

- *Distinguish between globally available, highly reusable services and local services*

## *Development Aids*

- *Establish a Collaborative Tool (Wiki or Discussion Forum)*
- *Utilize a Web Services Lab to experimentally verify interoperability, security and functionality of proposals*



# Conclusion

- Use an incremental approach
- Choose services to implement based on value to the mission
- Design a comprehensive architecture early
  - implementation
  - governance
  - management
- Reap benefits
  - Lower maintenance
  - higher flexibility
  - cohesion of IT with the overall mission



# Contact Information

**Eric Brown**

[ewbrown@mail.unomaha.edu](mailto:ewbrown@mail.unomaha.edu)

554-2746

**John de Groot**

[jdegroot@mail.unomaha.edu](mailto:jdegroot@mail.unomaha.edu)

554-2097

**Dorest Harvey**

[dorestharvey@mail.unomaha.edu](mailto:dorestharvey@mail.unomaha.edu)

554-3161