SOA for Healthcare: Promises and Pitfalls

Dennis B. Smith
dbs@sei.cmu.edu
SOA in Health Care Conference: Value in a Time of Change
Chicago, IL USA
June 3, 2009
Agenda

Healthcare IT Challenges

SOA: The Promises

SOA: The Pitfalls

SOA: The Path Forward

Summary
Is Healthcare IT Unique?

Some business issues are unique

- Unique business processes and data
- Heavy regulation
- Large variety of stakeholders: hospitals, patients, providers, insurance companies, suppliers, research institutions, regulators, …

Some are general problems that are faced by every domain

- Defining and modeling essential business information and business rules
- Storing and accessing information in support of business processes
- Assuring security, performance, availability and usability of IT systems
- Applying appropriate technology to solve business needs

The challenge

- Apply general software engineering principles to specific problems of healthcare
- Determine applicability of SOA to healthcare business and IT demands
Healthcare IT Challenges

Current healthcare business and IT demands

- Create consistent lifelong patient health record that can be shared between multiple healthcare providers
- Assure security and identity management
- Enable vastly different systems to work together
- Enable evolution to new healthcare alliances and to rapidly changing regulatory requirements
- Maintain investments in legacy systems

A fundamental constraint in meeting these goals

- 70% of healthcare is conducted in offices of fewer than 10 people*

* Shahid Shah, “Why Healthcare IT is in the State That It’s In”,
http://www.ebizq.net/blogs/healthcare/2009/04/why_healthcare_it_is_in_the_state_that_its_in.php
Agenda

Healthcare IT Challenges

**SOA: The Promises**

SOA: The Pitfalls

SOA: The Path Forward

Summary
What is SOA?

Service-oriented architecture (SOA) is a way of designing, developing, deploying and managing systems, in which

- Services provide reusable business functionality via well-defined interfaces.
- Service consumers are built using functionality from available services.
- Service interface definitions are first-class artifacts.
  - There is a clear separation between service interface and service implementation
- An SOA infrastructure enables discovery, composition, and invocation of services.
- Protocols are predominantly, but not exclusively, message-based document exchanges.
Services

Services are reusable components that represent business tasks, e.g.

- Look up patient information
- Validate credit card
- Get test results
- Schedule appointment

Services can be

- Globally distributed across organizations
- Reconfigured into new business processes
Promise 1: Cost-Efficiency

A service with equivalent functionality can be implemented, and used by all three applications.
Promise 2: Agility

The new application can use available services.

New services can be used by other applications as well.

Scheduling Application

- Patient Lookup Service
- Insurance Lookup Service
- Physician Schedule Lookup Service

Training and Education Application

- Patient History Lookup Service
- Room Availability Service

The new application can use available services.

New services can be used by other applications as well.
Promise 3: Adaptability

The SOA Infrastructure provides a standard communication mechanism between consumers and services. Changes in services have potentially no impact on existing service consumers.
Promise 4: Legacy Leverage

Legacy platform diversity and complexity is irrelevant to consumers.

Consumers access the services in a standard way.

It is the service’s task to invoke the legacy system.

SOA Infrastructure

- Scheduling Application
- Physician Schedule Lookup Service
- Insurance Lookup Service
- Patient Lookup Service
- Patient History Lookup Service

Physician Record System

External System

Patient Record System
A Notional Service-Oriented System Architecture

Service Consumers

Patient Portal
Insurance Company System
Research And Public Health System
Outpatient System
Inpatient System

Service Consumer X

Service Consumers

Infrastructure

SOA Infrastructure

Service Interfaces

Get Patient Info
Get Physician Info
Get Test Info
Create Lab Test Order

Service X

Service Y

Service Implementation

Patient Record System
Physician Record System
Laboratory System
System Z
Agenda

Healthcare IT Challenges
SOA: The Promises
SOA: The Pitfalls
SOA: The Path Forward
Summary
SOA is Real

Many successful case studies, mainly in commercial enterprises
Main goal for adoption of SOA is internal integration and business process improvement
Main adoption barriers are lack of governance and finding people with the right skills
Currently the best option available for loosely coupled systems integration and leverage of legacy systems
The technologies to implement SOA will change over time, but the concepts are here to stay

- SOA is much broader than its most popular instantiation (Web Services)
However …

SOA is potentially being stretched beyond its limits

- What was initially an approach for asynchronous document-based message exchanges now has performance, availability, reliability, security and other expectations of traditional distributed systems

The required link between business goals and SOA value is not clear for SOA adopters—still seen as simply a new technology

- Lack of education
- Lack of people with both technical and business skills
- Large amount of vendors in the SOA market—each with their own discourse—does not help
SOA Does Not Provide the Complete Architecture for a System

SOA is an architectural pattern/style/paradigm and not the architecture of the system itself.

An architectural pattern provides guidance that embodies best practices.

- The concrete elements and their interactions are the architecture of the system.

Any number of systems can be developed based on an architectural pattern.

- An architecture based on SOA inherits both the good and the bad.

Corollary: SOA cannot be bought off-the shelf.

- System qualities have to be built into the architecture of the system.
- Decisions have to be made—service design and implementation, technologies, tradeoffs.
All Legacy Systems Cannot Be Automatically Integrated into a SOA Environment

Upfront hands-on analysis on the technical feasibility and cost-benefit must be performed on a per-system basis.

- Is it technically feasible to create a service from the legacy system or part of the system?
- How much would it cost to expose the legacy system as services?
- Is this cost plus the cost of maintaining the legacy system more than the cost of replacing it with a new one?
- What changes will have to be made to the legacy system?
- How much will these changes affect current users and other production systems?

It might just not make sense to migrate the legacy system to an SOA environment.
The Use of Standards Does Not Guarantee Interoperability

Interoperability needs agreement on both syntax and semantics.

Web Services enable syntactic interoperability.

- XML Schema defines structure and data types.
- WSDL defines the interfaces: operations, parameters and return values.
- Available information, technologies, and tool support.

Web Services do not guarantee semantic interoperability.

- XML and WSDL do not define the meaning of data.
- WSDL does not define what a service does.
- It is an active research area—unresolved issues.

With Healthcare, the HL7 v3 messaging standard defines electronic interface messages to support healthcare workflows

- This provides a starting point for getting away from unique interfaces
SOA Is Not All About Technology

SOA not only means a shift in technology but also changes in the organizational governance model.

<table>
<thead>
<tr>
<th>What life-cycle model should be followed for services?</th>
<th>What other governance mechanisms are required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Service requirements and definition</td>
<td>• Conflict resolution</td>
</tr>
<tr>
<td>• Service development, composition, and testing</td>
<td>• Deployment mechanisms</td>
</tr>
<tr>
<td>• Service evolution and change management</td>
<td>• Monitoring mechanisms</td>
</tr>
<tr>
<td></td>
<td>• Enterprise-wide policies</td>
</tr>
<tr>
<td></td>
<td>• Service-level agreements</td>
</tr>
<tr>
<td></td>
<td>• Service registries</td>
</tr>
</tbody>
</table>

SOA not only means a shift in technology but also changes in the organizational governance model.
Developing Applications Based on Services Remains Complex

It is relatively easy to build applications and services that work with a particular infrastructure . . . but designing a “good” service might not be that easy.

From a service provider perspective

- Not many best practices for designing services
  - What is the right granularity?
  - What is the right Quality of Service (QoS)? Can you guarantee it?
- Have to know and anticipate potential consumers and usage patterns
  - “If you build it they will come” – Can you afford this?

From a service consumer perspective

- Ease depends on tool availability for SOA infrastructure.
- Larger granularity may lead to larger incompatibilities.
- Most difficult part is composition—data and process mismatches.
Agenda

Healthcare IT Challenges
SOA: The Promise
SOA: The Pitfalls
**SOA: The Path Forward**
Summary
The Path Forward: Pillars of SOA Adoption

Successful SOA Adoption

- Strategic Alignment
- SOA Governance
- Technology Evaluation
- Change of Mindset

SOA Principles
Alignment with Business Goals

Any successful SOA strategy has to be aligned with business goals.
The first question any organization should ask itself is: “Why am I adopting SOA?”

Examples

- Reduce time-to-market for applications
- Increase information available to patients
- Integrate healthcare partners
- Decrease development cost by increasing reuse
- Reduce maintenance costs
- Improve patient
- Improve internal processes
# Different Business Needs and Goals Drive Different SOA Strategies

<table>
<thead>
<tr>
<th>Business Needs and Goals</th>
<th>SOA Strategy</th>
</tr>
</thead>
</table>
| Increase information available to patients | • Intuitive portals  
  • Creation of services related to patient information |
SOA Governance

SOA governance provides a set of policies, rules, and enforcement mechanisms for developing, using, and evolving SOA assets and for analysis of their business value.

It provides the **who**, that **what** and the **how** business, engineering and operations decisions are made in order to support a SOA strategy.

- Policies and procedures
- Roles and responsibilities
- Design-time governance
- Runtime governance

Examples of Governance Elements

Source: Governance elements adapted from Oracle’s SOA Governance Model
Match of Technologies to the Problem Domain

Need a realistic understanding on what technologies can do in the specific problem domain

How to understand and keep up with the “alphabet soup”?  
- XML, SOAP, WSDL, UDDI, HL7, WS-Security?

How to determine which standards and technologies to implement in specific situations?

How to build systems that are resilient to changes in standards and commercial products that implement them?

How to determine if selected technologies will meet QoS requirements?  
- Security  
- Availability  
- Performance

All the above questions suggest a need for contextual experimentation
## Service-Oriented Systems Require a Different Development Approach

### Change of Mindset

<table>
<thead>
<tr>
<th>Traditional Systems Development</th>
<th>Service-Oriented Systems Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tight coupling between system components</td>
<td>Loose coupling between service consumers and services</td>
</tr>
<tr>
<td>Semantics shared explicitly at design time</td>
<td>Semantics shared without much communication between developers of consumers and services</td>
</tr>
<tr>
<td>—In the future, even at runtime</td>
<td></td>
</tr>
<tr>
<td>Known set of users and usage patterns</td>
<td>Potentially unknown set of users and usage patterns</td>
</tr>
<tr>
<td>System components owned by the same organization</td>
<td>Systems components potentially owned by multiple organizations</td>
</tr>
</tbody>
</table>
Starting Point: Healthcare Community

Healthcare Services Specification Project (HSSP): The Practical Guide for SOA in Health Care*

- Identifies core SOA principles
- Develops a business case for SOA
- Provides guidance on steps for SOA implementation
  - Defining “as-is” state
  - Defining “to-be” state
  - Specifying architecture and services
  - Building transition and implementation plans
- A good starting point!

* Collaborative effort between Health Level seven (HL7) and Object Management Group (OMG)
Agenda

Healthcare IT Challenges
SOA: The Promises
SOA: The Pitfalls
SOA: The Path Forward

Summary
Summary

SOA is an approach to software development where

- Services provide reusable functionality with well-defined interfaces.
- An SOA infrastructure enables discovery, composition, and invocation of services.
- Service consumers are built using functionality from available services.

SOA can enable agility, legacy leverage, adaptability and cost efficiency for healthcare systems. However, success requires:

- Strong software engineering discipline
- A focused SOA strategy
- Effective governance processes
- Informed technology decisions
- Plans for addressing culture change
NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.