Advanced Solutions Using Service Oriented Architecture

J. Eric Peele
Brian Olive
May 22, 2007

International Field Directors & Technologies Conference 2007
What do you think about SOA?

2007 Network Computing Reader Poll: Customer Dissatisfaction Guaranteed
Overview

1. What SOA is not
2. SOA Defined
3. Terms
4. From OO to SO
5. What is SOA?
6. 4 Tenets of SOA
7. SOA Benefits
8. Data Collection Drivers
9. What to Build
10. Pitfalls
11. Stated Benefits
12. Resources
What SOA is not

“Service Oriented Architecture (SOA) is widely forecast to be the most profound transformation in the history of business and IT”.

-- CBDI, an independent research company

- It is not a revolution, but an evolution
- It’s not about XML-in-your-face, late-bound programming, or Web services, although these are parts of an implementation strategy used in SOA
SOA Defined?

'A set of components which can be invoked, and whose interface descriptions can be published and discovered'

-- World Wide Web Consortium (W3C)
Coming to Terms with SOA

- Service
- Service Provider
- Service Consumer
- Message

- Interface
- Schema
- Contract
- Policy
From Objects to Services

1980s
- Object-Oriented
  - Polymorphism
  - Encapsulation
  - Subclassing

1990s
- Component-Oriented
  - Location Transparent
  - Tight Coupling
  - Runtime Metadata

2000s
- Service-Oriented
  - Message-based
  - Schema+Contract+Policy
  - Broad Interop

Compared to the Past

- Like objects and components, services represent natural building blocks that allow us to organize capabilities in ways that are familiar to us.

- Similarly to objects and components, a service is a fundamental building block that
  - Combines information and behavior.
  - Hides the internal workings from outside intrusion.
  - Presents a relatively simple interface to the rest of the organism.

- Where objects use abstract data types and data abstraction, services can provide a similar level of adaptability through aspect or context orientation.

- Where objects and components can be organized in class or service hierarchies with inherited behavior, services can be published and consumed individually or as hierarchies and or collaborations.
Four Tenets of Service Orientation

“A service is simply a program that one interacts with via message exchanges”

Don Box
Microsoft Corporation
January, 2004

Tenet 1: Boundaries Are Explicit

- Services provide a contract to define the public interfaces it provides.
- All interaction with the service occurs through the public interface.
- The interface consists of public processes and public data representations.
- The public process is the entry point into the service while the public data representation represents the messages used by the process.
Tenet 2: Services Are Autonomous

- A Service’s location can change at (almost) any time due to the fact that they’re dynamically addressable
- Neither the Service Provider or Service Consumer requires knowledge of each others’ internal workings in order to exchange data – they only need the published schemas & contracts
- Neither party make demands on each other as to how the work is carried out
- Services are entities that are independently deployed, versioned, and managed.
Tenet 3: Services Share Schema and Contract, Not Class

- Classes combine behavior and data (messages) into a single programming-language or platform-specific construct. Services break this model apart to maximize flexibility and interoperability.

- Schema defines the structure and content of the messages, while the service’s contract defines the behavior of the service itself.
Tenet 4: Service Compatibility Is Based Upon Policy

To interact with a service, two requirement sets have to be met:

- the functionality, syntax and semantics of the provider must fit the consumer’s requirements,
- the technical capabilities and needs must match.
Stated Benefits of SOA

- Attain greater business agility from existing IT investments
- Simplifies management of distributed resources across multiple platforms, requires less hardware, is more reliable, is standards-based, and is less costly.
- Enables development of a new generation of dynamic applications addressing a number of top-level business concerns that are central to growth and competitiveness
- Enhances business decision making by aggregating access to business services and information into dynamic, composite applications to provide more accurate and comprehensive information
- Flexibility in access to information in form and presentation (Web, rich client, mobile device)
- Greater employee productivity by providing streamlined access to systems and information and enabling business process improvement
Business Drivers in Data Collection

- Siloed applications with minimal integration between them
- Tightly coupled point-to-point integration
- Lack of real time business intelligence software that provides real-time key performance indicators (KPI) for decision making
- Duplicate case data—each channel maintaining its own case data
- Complex information integration
What Types of Services Do We Create?

- Infrastructure services.
- Data services—Simple atomic operations on an entity.
- Activity services—Coordinate data services for business process execution.
- Process services—Long-running business processes, possibly complex workflow, and human interaction.
- Event services—Notify subscribers of events.
Top 5 SOA Adoption Pitfalls

1. Not understanding SOA performance requirements
2. Not starting with an xml foundation architecture
3. Not creating a transition plan
4. Not standardizing SOA
5. Building SOA like traditional distributed architecture

Source: “Service-Oriented Architecture: Concepts, Technology, and Design” by Thomas Erl
Building SOA for the sake of SOA without reference to the business context is a project without organizing principles and guidance. The result is a chaotic implementation that has no business relevance.
The Shift to Service Orientation

General Advice:

• Be skeptical

• Start small

• Assume the worst and plan for it

• Give it time to mature
References and Resources


Questions

Download this presentation from:

http://www.rti.org/ifdtc