

Lessons Learned: How to Derive Business Value from a Service Oriented Approach

Jeff Hensley
CTO, The TriZetto Group





TriZetto Company Profile



- Public company (NASDAQ: TZIX)
- Headquartered in Newport Beach, CA
- Leading healthcare IT company, focused exclusively on the payer industry
- Comprehensive solutions for core administration, care management, network management, consumer health and government programs
- \$452M 2007 revenue, \$71M 2007 R&D
- 1,900 employees in 12 locations
- Strong management team, 20+ year experience



TriZetto's SOA Evolution

1999 began incorporating service oriented access via CORBA	2002 began development of standard Web services into products	2003 first published services available via Web services	2004 began deploying higher level business services	2005 SOA integration infrastructure technologies delivered	2006 integrated multi-product SOA solutions released to market	2007 delivered a new solution with services and an orchestration engine
---	--	---	--	---	---	--

- ▶ **Fielded service enabled solutions for 5 years +**
- ▶ **Many lessons learned along the way**
- ▶ **Today, all of our solutions are service oriented**



The SOA Journey

- ▶ **Initially developed solutions using CORBA**
 - Nice idea but not a widely adopted standard
- ▶ **Started producing fine grained services in 2002 and 2003**
 - Yeah, we have web services but usage was minimal
- ▶ **Realized that coarse grained services were the best way to interact with external systems and integration of solutions**
 - Finally began realizing benefits, additional services were enabled
- ▶ **Next we realized that we could provide new configurable solutions by orchestrating our coarse grained services**
- ▶ **We are now producing new solutions that are service enabled and include an orchestration engine to allow process configurability**



A natural combination

- ▶ **SOA + Orchestration enable Composite Applications**
 - The two most significant trends will continue to merge to support composite applications
 - Composite Applications lead to agile enterprises
 - One approach without the other will not succeed
 - To define future solutions, you must be able to offer multiple solutions
 - Core systems that can participate in a SOA environment
 - Component solutions to support emerging business trends
 - Infrastructure capabilities to blend components with core applications using the SOA approach

Composite Solution Example

Automated Claims Pre-processing Solution

- 1 EDI format transformed into internal format
 - 2 Clinical edits applied
 - 3 Re-price claim per network contract
 - 4 Transform claim back to EDI format
- E** Any **errors** routed to workflow for resolution

Inbound Claim Service

Outbound Claim Service

1

Transformation Service

4

Transformation Service

Transaction Manager

Enterprise Service Bus

BPM Engine

Business Rules Engine

Auditing Repository

2

Clinical Editing Service



Facets e2
PACE Subsystem

3

Pricing Service

NetworX™

E

Route To Service

Workflow



Service Orientation is an Enterprise architecture not an Application architecture

- ▶ **Many of industry “experts” are insisting Service Oriented applications have to be built on a loosely couple architecture and use Web Services inside the application**
- ▶ **This ignores the fundamental beginnings for SOA**
 - There are consumers and suppliers of services
 - Use a common protocol for interactions
 - Don't care about the suppliers' or consumers' platform or technology
- ▶ **We believe Service Orientation should be measured on how applications participate in an environment, not how they constructed internally**



Lessons Learned

- ▶ **Think business function when service enabling**
 - Very fine grained services are generally of little value and do not get adopted
 - Consider having business analysts specify what should be service enabled, developers seem to focus on fine grained services
- ▶ **Strive to adopt standard payloads**
- ▶ **Consider orchestration engines when interacting with multiple services**
- ▶ **Pay attention to exception handling – if it can go wrong.....**



Services versus data publication

- ▶ **As services are enabled, don't forget to refactor application integrations paying attention to legacy data publication**
 - Data publication is problematic....latency, initial loads, schema variations, etc.
 - Consider deprecating data publication in favor of services
 - Looser coupling
 - Less fragile
 - Better WAN performance
 - Data accuracy

QUESTIONS?

