

Using CA Gen in a Service Oriented Architecture (SOA)

Don Pooley

Everware-CBDI

Session 15

Date: October 29, 2007

About Everware-CBDI

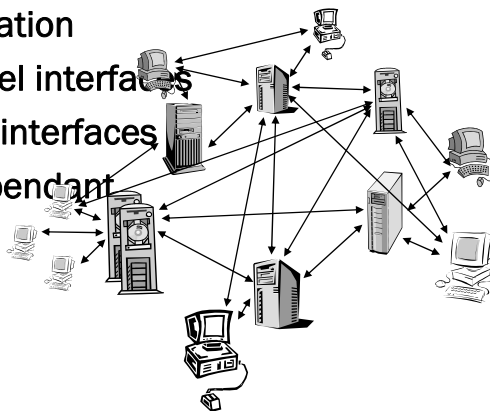
- **World Leading Independent SOA Consultancy and Knowledgebase**
 - CBDI Forum Knowledgebase (www.cbdiforum.com)
 - Monthly CBDI Journal (SOA Best Practices and Market Trends)
 - Public and in-house training
 - 22,000+ Subscribers Worldwide
- **Recognized World Leader in Component-Based Development**
 - Inventory of Component Products – CA Gen & J2EE
 - R & D to advance J2EE Components and Asset Reuse in SOA
 - Component Expertise to Implement SOA
- **Java/J2EE Expertise**
 - Service-Oriented Approach
 - Leverage Industry Best Practices & Design Patterns
 - Accelerated Development - Components, Templates, Generators
- **CA Gen Expertise (20+ years)**
 - Full Lifecycle Development
 - Legacy Application Transformation to SOA

Agenda

- **Services**
 - Gen Capabilities for Services
 - Which Option Do You Choose?
- **Service Architecture**
 - Gen's role in Service Architecture
- **SO Process**
 - Using Gen in the SO Process

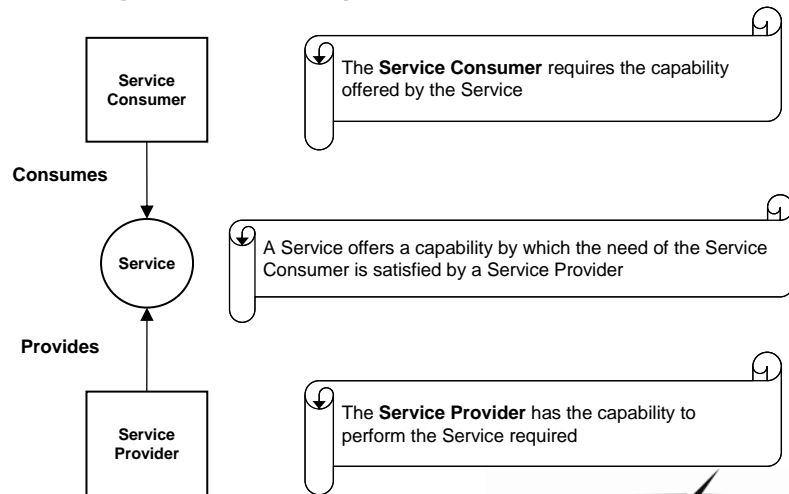
Tightly Coupled Applications

- Point-to-point integration
- Fine-grained low level interfaces
- Technology specific interfaces
- Implementation dependant

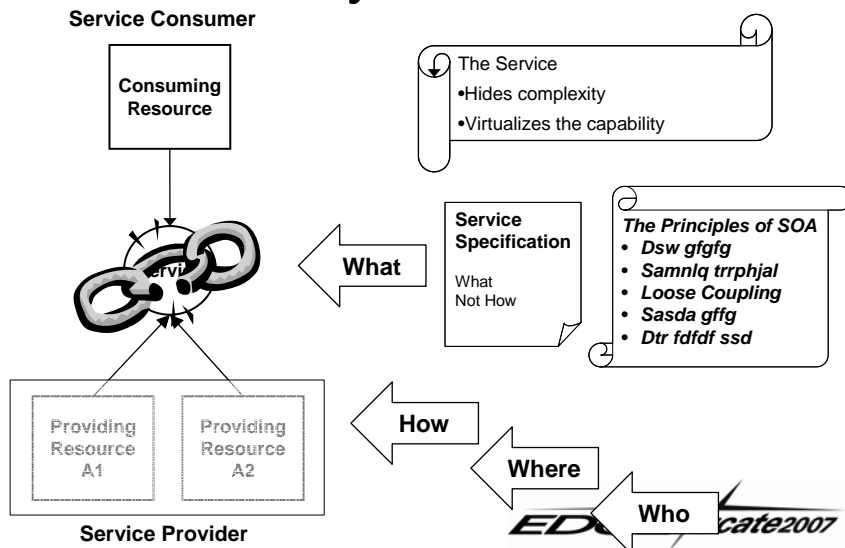


The "Accidental Architecture"

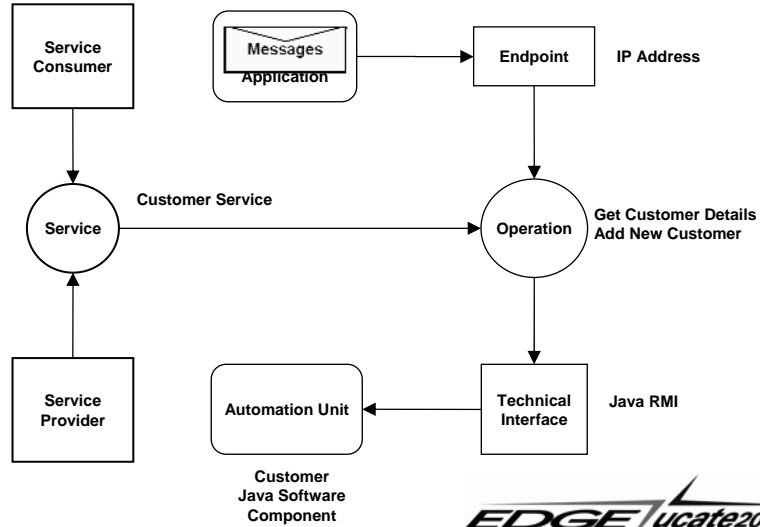
The Simple Concept of Service



Where or by Who

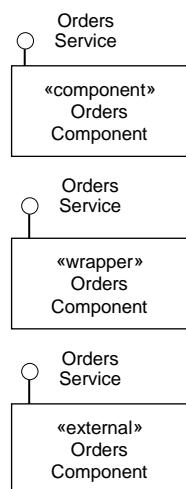


Services

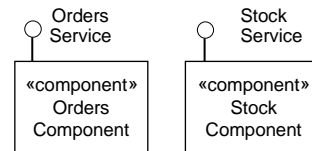


What is an Automation Unit?

Automation Unit: a collection of one or more executable modules or scripts that together provide the implementation for a service

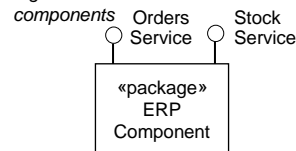


One service, one component



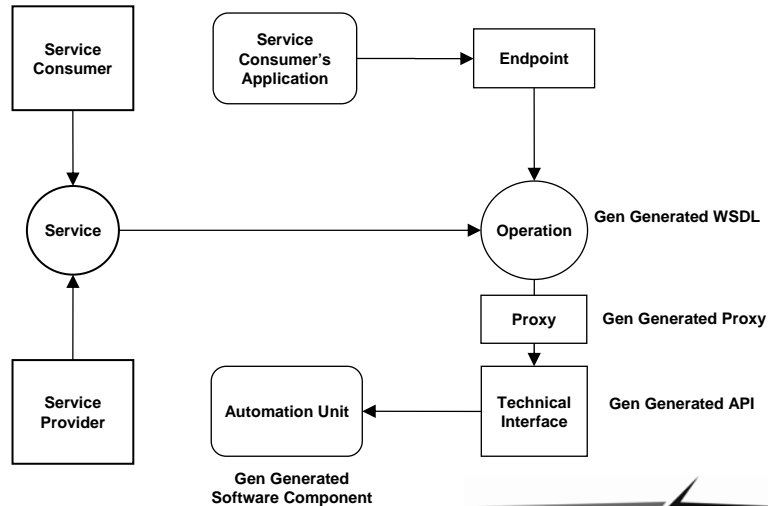
Several services, one component

e.g. well formed J2EE or .NET components



e.g. Large Software Package or Mainframe Application

Where Do Gen Apps Fit in?

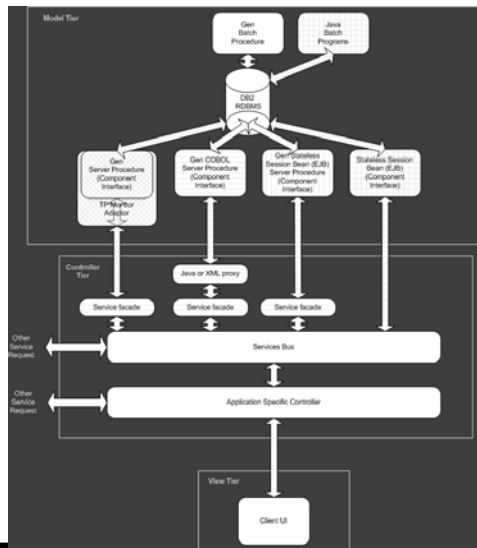


CA Gen Capabilities for SOA

- **Proxy Enablement**
 - .NET
 - JAVA
 - XML
- **EJB / .NET Generation**
 - Stateless Session Beans
- **Web Services Wizard (Gen R7)**
 - Generates WSDL
 - Leverages XML Java Proxy

Which Option do You Choose?

- TP Monitor adapter wrapped Gen Server
- Proxy exposed Gen Server
- Gen generated EJB or .NET



ate2007

Which Option do You Choose? ... Pros

- **TP Monitor Adapter**
 - Potential to use Gen transactions "as is"
 - TP Monitor might provide native support for Web Services
- **Proxy Exposure**
 - Supports multiple technical interface implementations from the same Gen server
 - Generates the distributed interfaces (no hand coding)
- **Gen Generated EJB or .NET servers**
 - Enables Gen servers to be deployed to new technical environments (i.e., Linux)
 - Potential to homogenize the architecture

Which Option do You Choose?

... Cons

• TP Monitor Adapter

- It's still transactional
- May be effort to build each adaptor
- Additional version control requirements

• Proxy Exposure

- Gen implementation specific, rather than business process / workflow service oriented (semantic definition)
- Additional licensing cost
- Additional version control requirements

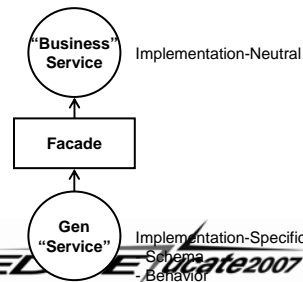
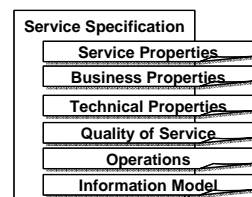
• Gen Generated EJB or .NET servers

- Risk associated from moving to a new platform
- Additional skills required
- Additional licensing cost
- What about the External Action Blocks?

Tips for Delivering Services from Gen Transactions

- Create rich service specifications using semantics that reflect business process service requirements
 - Use of Gen Component Specification 3.1. Specifies the operations and interfaces (signature) from a business perspective without regard to the target implementation.
- Look for affinity in data and operations to uncover higher level service groupings
- Create service facades to enforce "loose coupling" ... What Gen requires on one side vs. what the business needs on the other.
 - Could be implemented using an ESB or similar mediation capability

Rich Service Specification



Focusing on the A in SOA - Service Architecture

*Lots of Disparate Consuming Solutions
Driven by Pressing Needs Of Individual
Business Sponsors*



*Lots of Services delivered with good
intention, but failing to deliver the full
benefits of SOA*



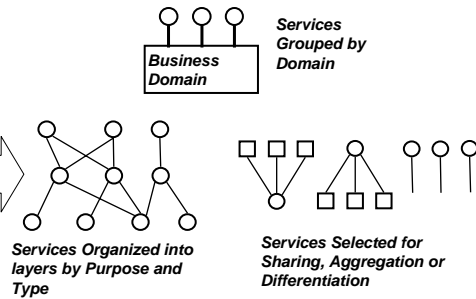
*Minimal Sharing, still silo'ed, disparate, and
meeting only the requirement of individual
business sponsors!*



*Lots of Duplicated, Silo'ed, Disparate,
Disparate Approaches*

Typical Approach to Services Today

Application of Architectural Concepts

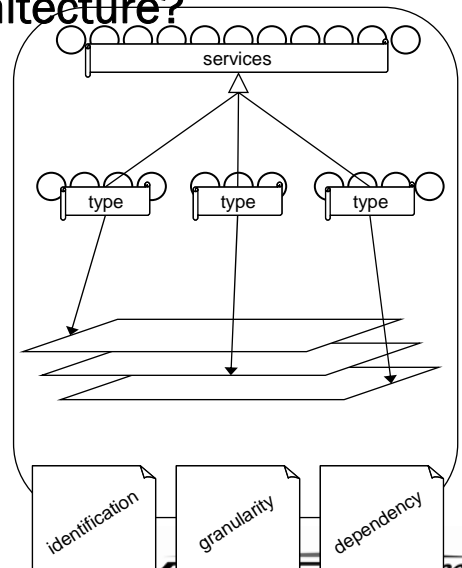


**Service
Portfolio
Plan**

EDGE Educate2007

What is Service Architecture?

- A collection of services
- classified into types
- arranged into layers

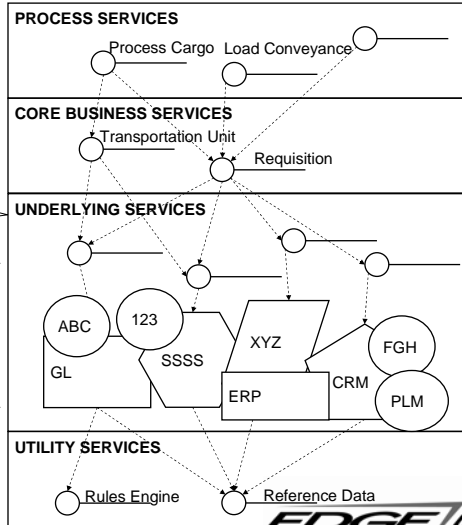


EDGE Educate2007

Where Do Gen Apps Fit in a Layered Services Architecture?

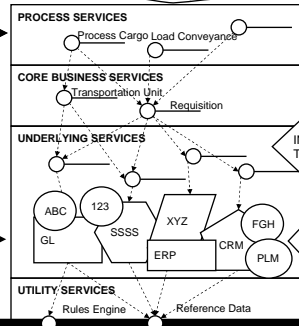
Service Facades that Expose a Gen Proxy

Gen Apps

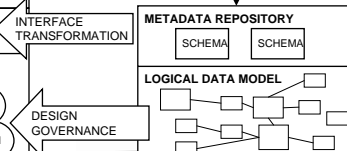
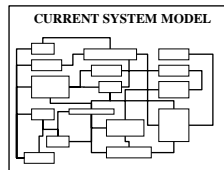
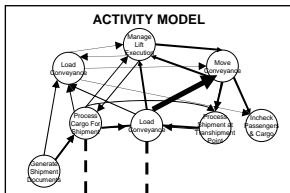
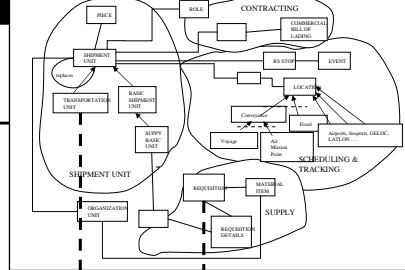


Bigger Picture

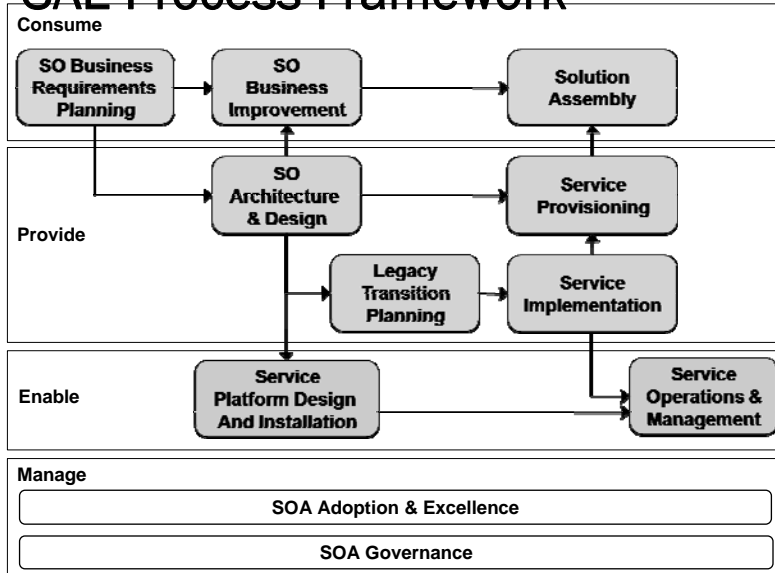
DOMAINS



BUSINESS TYPE MODEL



SAE Process Framework



e2007

Using Gen in the SOA Process...

- **SOAD**
 - Identify different Service types using appropriate models
 - Core Business Services
 - Process Services
 - Utility service
- **Legacy Transformation Planning**
 - Map legacy application functionality to Underlying Services
- **Service Provisioning**
 - Specifying Services
 - Specify the 'what' of the candidate service in terms of technology neutral interoperation semantics
 - Produce a rich service specification that is palatable to the service consumers (UML / Gen / Excel / Word)
- **Service Implementation**
 - Designing Services
 - Document a service design that supports the service specification within the defined architecture ... Focus is how and not what
 - Construction activities to create functionality defined during design
 - Implementation may extend beyond CA Gen

Do's and Don'ts

Do

- Understand and resolve architectural items early
- Institute and follow a methodical approach
- Try to design larger grain services that focus on Business Domains

Don't

- Confuse technical loose coupling with semantic loose coupling
- Confuse individual transactions with services
- Assume that replacement or re-development is easiest

Preparing for SOA

- Commitment
- Think 'services' and service-oriented architectures versus a transaction orientation
- SOA is a software architecture not a technology
- Organization
 - Teams should be constituted around a service rather than around the technical tiers
 - Twin track – separate solution assembly from provision
 - New sets of skills are required
 - Service orientation
 - New organizational roles
 - SOA Champion
 - SOA Custodian
- Active architectural support for the service teams
- Create a reference implementation (a model for others to follow)
- Proven supporting infrastructure

How Everware-CBDI Can Help

- SOA Guidance and Support
 - Subscription Services
 - Knowledgebase of SOA methodology and best practices
 - Education
 - Consulting
- Transition planning
 - SOA Adoption
 - CBD + SOA
 - System



Contact:
Don Pooley
Everware-CBDI
dpoolley@everware-cbdi.com

EDGE Needs Your Feedback!

- Please complete the conference evaluations
- Your feedback/suggestions are necessary to continue to bring you top-notch events
- Thank You for Coming – See you in 2008!