Evolution of Agile Enterprise Architecture

Presentation by Janne J. Korhonen, EDS
Contents

- EDS in Brief
- Background
  - Definitions
  - Dimensions of Agility
  - Brief history of enterprise integration
- From Fragile to Agile
  - From control to coordination
    - Workflow and Business Process Management
  - From imperative to declarative technologies
    - Orchestration and Choreography
  - Decoupling of entities
    - Objects, services, agents
- Agile Enterprise Architecture
  - Anatomy of Agile Enterprise Architecture
  - EDS Composite Application Framework
  - Vision of the future
EDS in Brief

- EDS is a leading global technology services company delivering business solutions to its clients:
  - ITO, AO, BPO
  - Application development
  - Transformation services
- EDS operates in 60 countries with 117,000 employees. The annual turnover exceeds $20 Billion.
- EDS Finland Oy is focused on professional services pertaining to Business Process Management (BPM), modern application architectures (J2EE/.NET) and Geographic Information Systems (GIS).
- EDS employs ~70 people in Finland.
Background
Definitions

• Enterprise Architecture
  – A means for describing business structures and processes that connect business structures. (CMU SEI Software Architecture Glossary)

• Agile Enterprise Platform
  – A network-based utility architecture that helps an organization respond more quickly to changing market dynamics, realize significantly improved economics and increase its competitiveness. (EDS)
Dimensions of Agility

**Examples**

- **Visibility**
  - Ability to have an **enterprise wide view** of markets and operations
  - Ability to see **key performance indicators** in a timely manner
  - Ability to conduct **root cause analysis**

- **Velocity**
  - Speed in serving existing markets and entering new markets
  - Speed of **decision making** and **organizational alignment** required to serve existing markets and enter new markets

- **Flexibility**
  - Adaptability of key business processes to needs of new market
  - High degree of **integration** across functional areas
  - **Financial flexibility** to invest in opportunities

- **Quality**
  - Level of excellence in executing key business processes
  - **Quality** of products and services **not compromised for operational excellence**

- **Efficiency**
  - Appropriate **balance of cost, quality and speed**
  - Management of **resources and costs** of achieving agility
  - Ability to **source** from the most **competitive suppliers**
Islands of Automation

- Information systems automate routine transactions and manual tasks
- Systems are isolated from each other and data are manually keyed in from one system to another
Point-to-Point Integration

- Systems are integrated with idiosyncratic one-to-one solutions
- The number of connections to integrate systems grows by the power of two as the number of systems grows
- Integration at endpoints is intrusive and requires specific technical knowhow
- Processes external to the integration scope
Enterprise Resource Planning (ERP)

- “One size fits all”
- Unified data source
- Best practice processes
- Often aligned with BPR projects
- Risky and costly
Package Software

- Data and process model of ERP too rigid and inadequate
- Domain specific solutions: SCM, CRM
- Back to square one: islands of automation
Message-Oriented Middleware (MOM)

- Queuing software as a mechanism to move information from point to point
- Relies on an asynchronous paradigm
- No direct coupling with the middleware mechanism and application required
- Message persistence
- Performance optimization
Enterprise Application Integration

- Message brokering
- Schema transformations
- Content transformations
- Technology adapters
- Application adapters
Business-to-Business Integration

- Leverages business protocols (e.g. RosettaNet, ebXML)
- Employs Web services technologies (SOAP, WSDL, UDDI)
- Loose document-based coupling
- Distinction between public and private process
- Collaboration
Workflow Management System (WFMS)

- Developed independently from integration technologies
- Workflow process
  - A single, sequential process
  - Document-/task-oriented
- Workflow technology does not support all types of processes
Business Process Management System (BPMS)

- **Orchestration**
  - A private executable process
  - Binds resources to a context through context-independent services

- **Choreography**
  - The observable public exchange of messages
  - Binds participants to roles within the business process
Brief History of Enterprise Integration Revisited

- Point-to-point
- MOM
- "Islands of Automation"
- ERP
- EAI
- Package Software
- B2Bi
- BPMS
- WFMS

---

Evolution of Agile Enterprise Architecture

12 April 2006

page 17 • EDS Confidential
From Fragile to Agile
## Fragile and Agile Architecture

<table>
<thead>
<tr>
<th>Fragile Architecture</th>
<th>Agile Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized control</td>
<td>Distributed coordination</td>
</tr>
<tr>
<td>Imperative paradigm</td>
<td>Declarative paradigm</td>
</tr>
<tr>
<td>Tightly-coupled entities</td>
<td>Loosely-coupled/decoupled entities</td>
</tr>
</tbody>
</table>
From Control to Coordination

Working definitions

• **Control** is power to direct the execution of a process within a context.

• **Coordination** is a mechanism to manage a distributed control based on a multi-party contract that constraints the exercise of power between process participants.
Control-Centric and Coordination-Centric BPM

- **Control-Centric**
  - Traditional workflow
  - A single process
  - State-based
  - Central API
  - Straight-through processing model
  - Document- or task-oriented
  - Limited interoperability
  - Limited support for distributed transactions

- **Coordination-Centric**
  - A collection of processes
  - Message-oriented
  - Abstract service endpoint
  - Collaborative processing model
  - Emerging standards
  - Distributed transactions part of the core process definition
Process Perspective

Participant = Context
Control
Data
From Imperative to Declarative Technologies

<table>
<thead>
<tr>
<th></th>
<th>Imperative</th>
<th>Declarative</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Paradigm</strong></td>
<td>Describe <strong>how</strong> to solve the problem</td>
<td>Describe <strong>what</strong> the problem is</td>
</tr>
<tr>
<td><strong>Method</strong></td>
<td>Coding</td>
<td>Configuration</td>
</tr>
<tr>
<td><strong>Program</strong></td>
<td>A sequence of commands</td>
<td>A set of statements</td>
</tr>
<tr>
<td><strong>Artefacts</strong></td>
<td>Instructions</td>
<td>Goals, facts, and relations</td>
</tr>
<tr>
<td><strong>Examples</strong></td>
<td>C, Java</td>
<td>PROLOG</td>
</tr>
</tbody>
</table>
Orchestration and Choreography

Working definitions

- **Orchestration** is an imperative formal description of the sequence and conditions in which an executable process invokes services and interacts with other processes in order to achieve its design objectives.

- **Choreography** is a declarative formal description of the coordination between multiple participants, specifying their roles and observable message exchange.
Orchestration vs. Choreography

- Orchestration
  - An executable business process
  - Includes execution order of web services interactions
  - Perspective of a single endpoint
  - Describes process flow
  - Can be composed of both internal and external webservice
  - Imperative description

- Choreography
  - The observable public exchange of messages
  - Tracks the sequence of messages involving multiple parties and sources
  - Two or more business process endpoints
  - More collaborative in nature
  - Declarative description
From an Orchestration Perspective

Orchestration – A private executable business process

Adapted from Oracle
From a Choreography Perspective

Choreography – The observable public exchange of messages

Adapted from Oracle
Orchestration and Choreography Together

Two process models reflecting a business agreement

Adapted from Oracle
From Object-Orientation to Event-Driven Architecture: Decoupling of Entities

Granularity

Large

Small

EDA

Agents

SOA

Services

OOA

Objects

Coupling

Tightly-coupled

Loosely-coupled

Decoupled

1995 2000 2005 2010
Object

- Data abstraction
- Methods: encapsulation of behavior
- Reusability through inheritance
- Technology-specific
- Tightly coupled: object instances reference each other
Service

- Context-independent
- Composable
- Subordinate
- Reactive
- One-to-one request-response
- Loose-coupling via messages
- Connectionless

Diagram: Request → Response → Interface → Implementation
Agent

- Context-aware
- Communicative
- Autonomous
- Proactive
- One-to-one or one-to-many publish/subscribe
- Decoupled
- Usually stateful and connected
- Form multiagent systems
Agile Enterprise Architecture
Anatomy of Agile Enterprise Architecture
Agile Infrastructure: EDS Composite Application Framework

FrontPlane™
Personalized Visualization

CrossPlane™
Configurable Orchestration
- Real-Time Operational Data Stores
- ETL/Data Flows
- Portal Integration Framework
- BPM/BAM Flows
- Right Time Data Marts & Warehouses
- Business Rules Engine

BackPlane™
Enterprise Integration Model
- Processes (BPEL)
- Data (XSD)
- Services (WSDL)
- Meta-Data Repository
- Transformations (XSLT)
- O/JDBC C/S
- WS/SOAP C/S
- E/JMS

Point Solutions
Data and Tools In the “Silos of IT”
- ERP Adapters
- HR Mgmt Adapters
- Configuration Mgmt Adapters
- Service Engine

Visibility
Velocity
Flexibility
Quality
Efficiency

Evolution of Agile Enterprise Architecture
12 April 2006
Dizzy already? We are on the halfway up

Level of abstraction

Machine-centric computing  Application-centric computing  Enterprise-centric computing  Process-centric computing

1s and Os  Assembly language  OSs  3GL  OO  CBD  Distributed computing  EAI  Middleware  B2Bi  WS  BPM  MDA  Grid  Web  Semantic computing  Autonomous computing