Application Integration Framework for a Large Business Process Redesign Project

Practitioner’s Report
OOPSLA 2000
Minneapolis, MN
October 19, 2000

Jon Sholberg
Boeing Commercial Airplanes Group
Agenda

- The Information Technology situation
- DCAC/MRM overview
- Application Integration (AI) framework
- Lessons learned
- Future directions – for the industry
- Future directions – for The Boeing Company
- Summary
The Information Technology (IT) Situation

- We are at an unprecedented flashpoint in the IT industry.
- Growth of business-to-business (B2B) e-commerce will change the whole pattern of industry and the IT infrastructure forever.
- The biggest IT problem today is seamlessly integrating disparate legacy applications with Web portals, Internet servers and commercial exchanges.
- Large companies require electronic collaboration with suppliers, partners and customers. They need to be able to connect their computer applications with any other computer application via the Internet.
Define and Control Airplane Configuration/Manufacturing Resource Management (DCAC/MRM)

- Configurator
- Product Data Manager (PDM)
- Computer-Aided Process Planner (CAPP)
- Enterprise Resource Planning (ERP)

Architecture for DCAC/MRM’s application integration began implementing in 1994
Business Process Redesign (BPR)
Definition and Scope

Process
Data
Information Systems/Technology

DCAC/ MR M is Process-Driven Change
Business Process Redesign (BPR) Objectives

DCAC/MRM Mission

Fundamentally rethink and radically simplify the processes related to airplane configuration definition and production.

The Approach

- Single Source of Product Data
  - Single Bill of Material

- Simplified Configuration Management
  - Eliminate Effectivity from Drawings

- Tailored Business Streams
  - TBS 1 Basic & Stable
  - TBS 2 Previously Delivered Options
  - TBS 3 New/Unique Options

- Tailored Materials Management
  - Rate Schedule (Very Simple)
  - Kanban (Simple)
  - MRP II (Complex)
Pre-DCAC/MRM Integration Example

Mainline Computing Systems
(point-to-point)
"Looser" coupled computing environment, minimal dependencies
Critical IT Success Factors for DCAC/MRM

- Technology must not delay business process redesign
- Use best of breed Commercial-Off-The-Shelf (COTS) package
- Integrate COTS packages with existing production systems
- Establish and achieve system integration objectives
- Establish a solid foundation for the future
System Integration (SI) Objectives

- **Flexibility**
  - accommodate future changes from both external and internal sources

- **Scalability**
  - accommodate current and future growth in users, transactions and data

- **Reliability**
  - provide consistent, dependable delivery of services

- **Performance**
  - must support health of the system, plus address end-user service level expectations
Application Integration (AI)
A Simplified View

Application A

Data

Data Exchange

Data

Application B
But It’s Not So Simple
Adapters simplify data exchange . . .

. . . allow developers to use their native application technology . . .

. . . and hide middleware complexity.
How simple?
Language (Java) Based:
PDM.Part.Instance().update(partName, supplierCo, note, partNum);

File Based:
fprintf(file, "%s^A%s^A%s^A%d^B", partName, supplierCo, note, partNum);

COBOL IMS:
IMS TRAN
   GU
   ISRT IO-PCB
   ISRT ALT-PCB

Stub Based ("Fill in the blank."):
void PDM::Part::update
   (char *partName, char *supplierCo, char *note, int partNum)
{
    // Insert your code here.
    // Business logic only, no middleware code!
    return;
}
### Transaction Profile

<table>
<thead>
<tr>
<th>Category</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronicity</td>
<td>Synchronous, Asynchronous, Dual Async</td>
</tr>
<tr>
<td>Initiator</td>
<td>Push, Pull</td>
</tr>
<tr>
<td>Publishing</td>
<td>Publish/Subscribe, Store &amp; Forward</td>
</tr>
<tr>
<td>Activation</td>
<td>Trigger, Periodic, Persistent</td>
</tr>
<tr>
<td>Mode</td>
<td>Interactive, Batch</td>
</tr>
<tr>
<td>Latency</td>
<td>Guaranteed, Expires, Once Only</td>
</tr>
<tr>
<td>Sequence Sensitive</td>
<td>Sequence Sensitive</td>
</tr>
<tr>
<td>Security</td>
<td>Encrypt, Authenticate, Authorize</td>
</tr>
<tr>
<td>Response</td>
<td>Status, Exception, Error, No Reply</td>
</tr>
</tbody>
</table>

✔ Plus configuration & environment data
## Existing Adapter Types

<table>
<thead>
<tr>
<th>Client</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>File</td>
</tr>
<tr>
<td>FTP</td>
<td>FTP</td>
</tr>
<tr>
<td>C++ Library</td>
<td>C++ Library</td>
</tr>
<tr>
<td>Java Jar</td>
<td>Java Jar</td>
</tr>
<tr>
<td>SQL</td>
<td>SQL</td>
</tr>
<tr>
<td>SQLJ</td>
<td>SQLJ</td>
</tr>
<tr>
<td>JDBC</td>
<td>JDBC</td>
</tr>
<tr>
<td>C Library</td>
<td>C Library</td>
</tr>
<tr>
<td>TCL Procedure</td>
<td>TCL Procedure</td>
</tr>
<tr>
<td>IMS</td>
<td>IMS</td>
</tr>
<tr>
<td>MQSeries</td>
<td>MQSeries</td>
</tr>
<tr>
<td>CORBA</td>
<td>CORBA</td>
</tr>
<tr>
<td>C++ Stub</td>
<td>C++ Stub</td>
</tr>
<tr>
<td>Java Stub</td>
<td>Java Stub</td>
</tr>
</tbody>
</table>
Interface Agreement

Description of data
+ Transaction profile
+ Client & server adapter types
How to Handle all the Integration Needs?
Integration Strategies

Business Applications

ERP

PDM

NCM

Integration Strategies

N²

2N

Middleware
$N^2$ Integration

a.k.a Point-to-Point
2n Integration
Application Integration Capabilities

- Interoperability across technologies and applications
- Language, platform and application adapters
- No exposure of middleware, no CORBA coding required
- Highly distributed applications can run on any host
- Transparent transport of business data over middleware
- Server location
- Fanout
- Auditing and audit queries
- Transparent load balancing
- Fault tolerance with transparent failover
- Activation of transient processes
- System management
- Asynchronous and store-and-forward services
- File and resource management
- Public key infrastructure (PKI) security (soon)
- XML translators (soon)
Positive Lessons Learned

✓ Distributed, object-oriented integration was successful using CORBA and messaging
✓ Loosely-coupled, stateless integration enabled COTS applications to be managed autonomously
✓ Tough objectives were met: flexibility, scalability, reliability and performance
✓ Defining and developing an application integration framework enabled both abstraction and enterprise-wide system and performance management capabilities
✓ CORBA and UML standards enabled code generation and assembly of components
✓ Using standard components, skilled developers were easier to find
✓ Partnering with suppliers was essential to success
Less Positive Lessons Learned

- CORBA by itself was insufficient – messaging required
- Standards develop slowly
- Many services were not readily available (and still aren’t)
- COTS vendors didn’t prioritize application integration until lately
- Integration technology evolved quickly
- eBusiness has altered integration expectations dramatically
- Information broker capabilities became mandatory
- Demands for seamless integration of all types were everywhere
- XML has recently become wildly successful
Industry Future

✓ CORBA becomes infrastructure
✓ J2EE becomes the foundation for standards-based integration:
  • application-to-application (A2A) - internal
  • business-to-business (B2B) - external
  • collaboration - virtual
✓ Canonical form of data passed within the middleware (XML)
✓ Ability to interface to “everything” from “anywhere” is becoming possible
Next Steps

✓ Build upon the existing AI framework to embrace the new technologies
✓ Extend the existing architecture to easily facilitate eCommerce
✓ Unify integration both within The Boeing Company (A2A) and to outside entities (B2B specifically)
Future Application Integration Framework

Portal Server
- Scalability
- Routing
- Security Mgmt
- Profile Mgmt
- Policy

Application Server (J2EE)
- Business Logic
- Deployment
- Session and Process Management

Integration Server
- Data transformation
- Rules Engine
- Connectors

Browser
- HTTP
- WAP

Security
- System Mgmt
- Personalization
- Audit

Enterprise to Enterprise
- SOAP
- XML
- other

C++
- EJB
- COM+
- OS/390

PDM
- MES
- ERP
- MVS
Summary

- Distributed object computing works with CORBA, if it is based on a solid enterprise architecture
- The demands of eCommerce widen the integration requirements domain dramatically
- New development for the enterprise is happening on the J2EE application server platform
- Using both environments together gives the architectural foundation for integration that eCommerce demands
Thank you.

Questions?

jon.a.sholberg@boeing.com