Enterprise Data Validation Architecture (EDVA):

Fixing Data Quality for Enterprise Interoperability

Cambridge, Massachusetts, USA
July 18-19, 2007

Bill McMullen
Outline

• Purpose
• Nature of the Problem
• Elements of a Solution
• Business Case Considerations
• Q&A

Business Enterprise Architecture (BEA) & Enterprise Transition Plan (ETP)  
September 2006

An Objective – “Transform the Department’s supply chain information environment by: 1) improving data integrity and visibility by defining, managing, and utilizing item, customer, and vendor master data; and 2) reducing complexity and minimizing variability on the supply chain business transactions by adopting standardized transaction and business rules.”
Purpose

• Purpose of Briefing
  Provides an overview of an Enterprise Data Validation Engine
  (This may be an important management option for certain business cases.)

• Purpose of an Enterprise Data Validation Engine
  – Aligns data across previously independent legacy system sources
  – Eliminates routine manual reconciliation efforts previously needed to coordinate legacy data sources
  – Improves decision making at the domain and/or enterprise levels through enhanced legacy data quality
  – Can be applied in many ways across the enterprise
Decision Making Capability for the Next Decade

• ERP promises highly accurate data and superior decision making ability, but a wait is involved
  – ECSS is expected to be deployed about 2011+

• It is not necessary to wait to increase data and decision integrity
  – Selected “legacy” systems, with minimal effort, can provide more integrity in the meantime
  – Global Logistics Support Capability (GLSC) requires reengineering both the business processes and their supporting systems

• It is possible to make improvements without distracting from ECSS-ERP
  – The proposed approach will actually facilitate the migration of valid data to ECSS
Nature of the Problem Regarding Enterprise Data Validation

- **The Business Problem**
  - Data from separate legacy systems frequently cannot reliably be combined at domain or enterprise levels
  - When combined, may not yield trustworthy results for quality enterprise decision making
  - Examples
    - **Case #1**: 29% mismatch among two requisition systems (re GLSC)
      - “Almost 30 percent of Air Force backorder data is inaccurate.”
      - “Forty-two percent of in-transit records ... were invalid. This equates to 4,627 transactions, valued at $325M.”
    - **Case #2**: 98% failure to integrate databases across domain
    - **Case #3**: 25% mismatch (est.) among several vehicle systems
  - Manual reconciliation to compensate is not satisfactory in today's environment (AFSO21)
    - Airman resources no longer will be available to “fix” data (re GLSC)

- **The Technical Issues**
  - Synchronization of multiple source system data
  - Association of multiple source data
The Systems Gap: Misaligned Data Leads to Questions

Source systems often describe different aspects of the same asset. Integration blends the data from the multiple source systems into a global “mismatched” view.

Data for Vehicles Within 3 Systems

Issue – Time and Instance Integrity Across Systems

Problem – Reporting Produces Inconsistencies (Q: How much can we trust this?)

Note: The Synchronization Issue shows up in a database when data from multiple source systems does not correspond across the sources in a logical manner, or as expected, given the business rules.
Enforce Business Rules & Link Records

- Check against enterprise business rules
- Make corrections at this point or they usually won’t (can’t) get done later

Cross-Reference Information (GFM-DI)

Note: The Association Issue shows up in a federated database when invalid associations among object is erroneously permitted. Validation against business rules must be performed prior to a source system allowing such a transaction to update its own database.
Problems Compound as More Systems Are Considered

Compounding of Missing, Mismatched, or Unsynchronized Data

(Due to improperly associated and serialized enterprise-wide data and enforcement of enterprise-wide business rules)
Some Elements of a Solution for Using an Enterprise Data Validation Engine

**Associations Across Systems**

Determine how different objects of two systems are associated.

**Validation Engine**

**New** business rules must be enforced by Systems A and B.

**System A**

**Object X**

**Associative Cross-Reference**

**System B**

**Object Y**

Note: EII usually requires that objects in separate database (i.e., Objects X and Y above) be logically related according to certain enterprise business rules.
Leveraging the Air Force Architecture
Source System Architecture Using the Enterprise Data Validation Engine

- Leads to greatly enhanced data agreement across systems that currently lack common standards or interoperability

- Enterprise Data Validation Engine ensures data integrity as it trickles in from the various systems

- Enterprise Data Validation Architecture
  - Components include
    - Cross-Reference for Associations
    - Enterprise Data Validation Engine
    - Change Identifier and Notifying
    - Interoperability Protocols (7 possible choices)
      [Database Snapshot for Non-Intrusion*]

* Can be non-intrusive to source system, but with critical tradeoffs
  - Performance
  - Data Quality

* Previous Snapshot for Non-Intrusion Systems

This architecture is an adaptation from "Protocols for Integrity Constraint Checking in Federated Databases", P. Grefen and J. Widom, 1997
Data Entry Scenario Where Legacy Modifications Allowed

Current Legacy Data Entry Process

1. Initial Entry Screen
   - ID
   - model
   - serial number
   - Address Line 1
   - Address Line 2
   - Address Line 3
   - xxx-xxx

Improved Legacy Data Entry Process

Cross-Reference

Check for existence

1. If already existing: Populate form
2. If not already existing: Continue as before
3. Finish data entry

1. Initial Entry Screen
2. Pre-Populate Screen
3. Remaining Data Entry

Local Store

ID model serial number
xxx-xxx

Address Line 1
Address Line 2
Address Line 3
yyy-yyy

ERP-new

Local Store

ID model serial number
xxx-xxx
Business Case Considerations

• Program-Centric Aspects
  – Development Effort and Time [Investment]
    • Added Effort
      – Systems Aspects (enhanced data integrity capabilities vs. system revisions)
      – Operational Aspects (elimination of reconciliation activities vs. data entry revisions)
    • Coordination Among Other Stakeholders (via a COI?)
  – Lifecycle Benefits (may accrue over time) [Return]
    • Improved Data Integrity
    • Improved Data Timeliness
    • Eliminate Redundant Data Entry
    • Eliminate Reconciliation Effort

• Enterprise-Centric Aspects (in addition to above)
  – Generic Solution Template
  – Prerequisite to ERPs (DEAMS, ECSS, DIMHRS) Migration
  – Improved Integrity of Enterprise-Level Decision Making
Considerations for Use of Enterprise Data Validation

Where would the Enterprise Data Validation Engine be considered for use?

After conducting Enterprise-Level Cross-System Data Quality assessments for given products, then perform the following Data Fixing Strategy as applicable.

<table>
<thead>
<tr>
<th>Rule</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Situation</td>
<td>Can Data Be &quot;Abandoned&quot;?</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Is Current Quality OK?</td>
<td>-</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Feasibly to Re-Entered?</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Feasible to Reconcile?</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Manually</td>
<td>Automatically</td>
<td>Automatically</td>
<td>Automatically</td>
<td>-</td>
</tr>
<tr>
<td>Can Apply Fix at Migration Time?</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>-</td>
</tr>
<tr>
<td>Fixing Sources Cost Acceptable?</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Yes*</td>
<td>No</td>
<td>-</td>
</tr>
<tr>
<td>Data Fixing Strategy</td>
<td>Do Nothing</td>
<td>Do Nothing</td>
<td>Manually Re-Entry</td>
<td>Manually Reconcile</td>
<td>Do Data Cleansing</td>
<td>Solve Enterprise Data Validation</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Note: The term "data" herein includes any data aggregation (element, record, file, cross-system composite, etc.).

Data Quality Improvements

This issue is similar to the Y2K problem in that you can't wait until the last minute to think about or act upon. It has the additional characteristic that the sooner it is addressed the better the data quality gets over time.

* If interfaces are being changed anyway, then these costs may be negligible to use EDVA template.

Indicates Inherent Government Responsibilities
Questions and Answers
Backup Slides
Associations Between Objects

Federating among systems can require that different objects of two systems be associated.

Thus, *new* synchronization business rules must be enforced on Systems A and B.

System A

Object X

Validation Engine

Associative Cross-Reference

System B

Object Y

Tx means transaction.
AF GLSC Data Adaptors connect the authoritative sources to the ESB and provide various mediation services including format and protocol translation, mapping and translation.
Source System Architecture (expertise)
Source System Architecture
(some needed pre-requisites)
Source System Architecture
Variant Where Source System A Is ESB-Enabled

However, existing messages from source system A to ESB must be sufficient for enterprise-wide needs, both in its timing and in its elements.
Source System Architecture Variant where Source System A Is AFKS-Enabled

However, existing messages from source system A to ESB must be sufficient for enterprise-wide needs, both in its timing and in its elements.

These are either/or, not both.

Not part of these discussions.
About 200 Sites
Distributed Data
Stand Alone Operations

System of Record for Equipment Management

Rejecteda Transactions Require Human Intervention

AFEMS
Air Force Equipment Management System

Vehicle Records

AS-IS Data Sharing Example from Vehicle Domain

Rejected Transactions Require Human Intervention

OLVIMS
On-Line Vehicle Information Management System

Vehicle Operations Management Data

Summary Financial Reporting

SBSS
Standard Base Supply System

Vehicle Records

$\$\$
Systems

1. Tx are edited for syntax by SBSS
2. Tx are edited for syntax and semantics by AFEMS
3. Some tx are simply forwarded after syntax check (no other processing done by SBSS)

Core vehicle data copies with additional domain data

Interfaces with limited automation; mostly abandoned in favor of dual data entry

Rejected tx have limited notification processes and must each be cleared by workers

Rejected transactions require human intervention

Tx Types of Equipment Management Interest

Tx Types of Financial Interest and Equipment Management Interest

Tx Types of Financial Interest

Summary Financial Reporting

Rejected Transactions Require Human Intervention

Proceedings of the MIT 2007 Information Quality Industry Symposium

PG 820
Vehicle Life Cycle

- Need: One Pickup
- Acquire
- Wait for
- Receive

<table>
<thead>
<tr>
<th>Change Highlights</th>
<th>Generating Parties</th>
<th>Interested Parties</th>
<th>Participating Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait for (SPR)</td>
<td>Acquisition</td>
<td>Vehicle Management, Equipment Management, Inventory Management</td>
<td>OLVIMS, AFEMS, SBSS</td>
</tr>
<tr>
<td>Receive (REC)</td>
<td>Vehicle Management</td>
<td>Vehicle Management, Equipment Management, Inventory Management</td>
<td>OLVIMS, AFEMS, SBSS</td>
</tr>
<tr>
<td>Operate (mission capable)</td>
<td>Vehicle Operations</td>
<td>Vehicle Management, Equipment Management</td>
<td>OLVIMS, AFEMS</td>
</tr>
<tr>
<td>Maintain (non-mission capable)</td>
<td>Vehicle Maintenance</td>
<td>Vehicle Management, Equipment Management</td>
<td>OLVIMS, AFEMS</td>
</tr>
<tr>
<td>Maintain (awaiting parts)</td>
<td>Vehicle Maintenance</td>
<td>Vehicle Management</td>
<td>OLVIMS</td>
</tr>
<tr>
<td>Turn-In (salvage)</td>
<td>Vehicle Management</td>
<td>Vehicle Management, Equipment Management</td>
<td>OLVIMS, AFEMS, SBSS</td>
</tr>
</tbody>
</table>

- Operate and Maintain
  - Mission Capable
  - Mission in Progress
  - Needs Preventive Maintenance
  - Work Order in Progress

- Turn-In
Characteristics of The Situations

- Multiple object creation responsibility
- Cross-Reference contains
  - Object identifiers (with association rules)
  - Data common to the systems
- Systems can
  - Check object association validity
  - Eliminate cross-system data conflicts
  - Reduce redundant data entry

Attribute Synchronization Across an Object

System A
- Object X
- Account Information

System B
- Object Y
- Customer Information

System C
- Object Z
- Branch Information

Associations Among Objects

System A
- Object X
  - piece 1
  - Acquire Data

System B
- Object X
  - piece 2
  - Finance Data

System C
- Object X
  - piece 3
  - Inventory Data