Disparate Systems/Middleware

Vincent J. Bordo
Senior Consultant
Advanced Concepts Center, LLC

Wednesday, June 23, 2004
Dallas Convention Center
Today’s Topics

- Welcome and Introduction
- Incompatibility Issues
- Business Process Realignment
- Middleware Components
- Hospitality Software Standards
Welcome

Introducing your Presenter -- **Vince Bordo**

- Over 20 years experience in the IT industry
- Successfully integrated dozens of systems for GE Aerospace -- working on one of the largest projects in the United States
- Specializing in object-oriented technology, requirements and project management
- Road warrior...spending over 130 nights/year in hotels world-wide

Senior Consultant
Advanced Concepts Center, LLC
Headquarters: Roswell, Georgia
An Inhospitable Scenario

- Disparate systems can cause negative guest experiences
  - Missing reservations
  - Inaccurate guest folios
  - Inoperable key cards
  - Unrecognizable frequent guests
  - Incorrect room assignments
  - Misquoted room rates
  - Un-credited frequent guest cards
Back-Office Troubles Abound

• Disparate systems also affect operations
  – Inconsistent room inventory profiles
  – Disparate customer databases
  – Lack of consistent chain-wide guest information
  – Insecure links to booking websites
  – Multiple sales channels
  – Frequent guest programs not being credited
Incompatibility Issues

Q: Why do you think we have such an epidemic of incompatible systems throughout the hospitality industry?

A: Because of Conway’s Principle

Conway’s Principle *prov.* The rule that the organization of the software and the organization of the software team will be congruent; commonly stated as “Organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations.”

*The principle was named after Melvin Conway, an early proto-hacker who wrote an assembler for the Burroughs 220 called SAVE.*
Conway’s Principle

Three owners = three systems

One owner = one system

The structure of a system reflects the structure of the organization that built it.
Let’s Analyze the Problems

- Current systems are having difficulty keeping up with today’s increasing demands
  - Manual workarounds
  - Human errors
  - Incompatible data
- Interfaces not properly integrated with related systems
- Batch-oriented processing can cause timing problems
Reasons for Disparity

- System incompatibilities are sometimes caused by:
  - Different needs for automation
  - “In-place” systems incompatible with new technology
  - Investment constraints
  - Choosing non-integrated solutions

- Some vendors resist standards because of:
  - Little incentive to redesign proprietary interfaces
  - Perceived loss of competitive advantage
  - An unwillingness to believe standards will “fix” the problem
  - But...there are those who do develop to standards

Single-perspective solutions have caused enterprise-wide incompatibilities
Business Process Realignment

• It’s time to re-think our current business processes
  – Determine which processes need rework
  – Take-on an enterprise-view regarding the flow of activity
  – Redesign the inefficient processes
  – Utilize “new” technology (when possible) to rejuvenate outdated and/or outmoded operations
Locate Areas of Inefficiency

- Conduct an “as-is” process mapping
  - Trace the flow of current operations
  - Look for inefficient timing and control loops
  - Identify areas that require both manual and automated processes
- Trace the flow through the operations using an “activity (i.e., flow) diagram”
“As-Is” Activity Diagram

Produced by Hospitality Financial and Technology Professionals
Improve the Process

• Review the “as-is” process for areas requiring improvement
• Look at the current process with the “guest-experience” in mind
• Ask yourself “how can we make it better”?
• Identify all systems that could be improved (i.e., impacted) by the new process
• Document the “new” process by using a “to-be” activity diagram
“To Be” Activity Diagram

Guest
Arrive at Hotel
Accept Credit Card
Process Reservation
Issue Room and Key
Locate Reservation

Check-in Kiosk
Reservation System

Parking Attendant
Bellman

Improved Check-in Process

Property Management System

Telephone Management System

Keyless Entry System

Energy Management System

Go to Room

Issue Ticket
Park Car

Receive Notice
Deliver Bags

Check-In

Activate Phone

Activate Key Card

Turn HVAC on
Identify the Impacts

- Identify the interfaces which will require modification based on the “new” process
- Describe the interface specifications (i.e. data, frequency, volume, batch etc.)
- Determine which systems could change and which must stay the same
- Will this solution apply to every guest?
- Choose the most appropriate middleware technology for each interface
- Implement the solution
Improve the Process First...

- Organize around outcomes, not tasks
- Process modernization *precedes* automation
- Use of benchmarking and other techniques to regularly assess the costs and benefits of functional processes
- Establish process ownership (responsibility, accountability, and authority)
- Subsume information-processing work into the real work that produces the information
- Put decision points where work is performed, and build control into the process
- Standardize similar processes
- Be customer focus
- Capture information once, and at the source
- Move toward standard data definitions by your entity
- CHANGE NOW, do not wait for a "perfect" solution
- Build new systems only as a "last resort"

...The Technology Will Follow
Upgrade the Technology

- After the processes have been improved, identify areas of automation requiring change
- Identify the mis-match of interfaces between systems and identify problem areas
  - Lack-of or incompatible data exchanges
  - Incompatible languages/systems
- Choose the appropriate middleware components to resolve the differences
Move Towards Real-Time

- Today’s hospitality systems will benefit from event-driven behavior
  - A room reservation is booked
  - A guest checks into a hotel
  - A bar tab is paid
  - The guest leaves their room
  - A telephone call is made
- These events will result in real-time messages to alert/inform other systems
What is Middleware?

- Middleware is a layer of software between the network and the applications.
- Provides services such as identification, authentication, authorization, directories, and security.
- Promotes standardization and interoperability between disparate systems.
What is XML?

- eXtensible Markup Language (XML) is a simple text format used to exchange messages between systems
- “Tags” are assigned to each element of the message
- XML is:
  - an open standard
  - human readable
  - platform independent

```xml
<?xml version="1.0" ?>
<Reservation GuestName="Mr. Jim Smith">
  <GuestAddress>
    <GuestStreet>123 Peachtree St.</GuestStreet>
    <GuestCity>Atlanta</GuestCity>
    <GuestState>GA</GuestState>
    <GuestZip>30030</GuestZip>
    <HomePhone>(404) 577-1234</HomePhone>
  </GuestAddress>
  <Payment Method="Credit Card">
    <CardType>Visa</CardType>
    <CardNumber>4388 1234 5678 8901</CardNumber>
    <ExpDate>2006-05-31</ExpDate>
  </Payment>
  <ReservationId>1654739</ReservationId>
  <Property>NoTel Motel</Property>
  <CheckinDate>2004-06-21</CheckinDate>
  <CheckoutDate>2004-06-24</CheckoutDate>
  <NumOfNights>3</NumOfNights>
  <BedType>King Size</BedType>
  <Smoking>Non Smoking</Smoking>
  <PillowType>Polyester Fill</PillowType>
  <SpecialRequest>Room with view</SpecialRequest>
</Reservation>
```
Middleware as a Conduit for XML

- Property Management System
- Room Service
- Sales & Catering
- In-Room Movie
- Point-of-Sale
- Call Accounting
- Voice Mail
- PBX
- Keyless Entry
- Central Reservation System
- Housekeeping
- Mini-Bar
- Energy Management

Produced by Hospitality Financial and Technology Professionals
What About Uncooperative Databases?

- Proprietary databases have caused some of the most significant interoperability problems
- Data is often stored in ways which make it difficult to access or share with other systems
- One technique to avoid the multi-vendor database incompatibility issues is to use Open Database Connectivity (ODBC)
What does ODBC Offer?

- Open Database Connectivity (ODBC) drivers allow programs to access any database using the same interface.
Getting our Systems to “Talk”

- Many incompatibility problems occur when different vendor’s systems try to communicate
- The platform differences cause communication problems
- Middleware can provide the conduit for messaging.
What is CORBA Used For?

- Common Object Request Broker Architecture (CORBA) enables interoperability between disparate systems.

IDL = Interface Design Language
Standards, Standards, Everywhere

HITIS = Hospitality Industry Technology Integration Standard
OTA = Open Travel Alliance
RMSIG = Reference Model Special Interest Group
WHIS = Windows Hospitality Interface Specification
HTNG = Hotel Technology Next Generation

Produced by Hospitality Financial and Technology Professionals
Standards: The Key to Future Success

- Technology standards need to be adopted throughout the hospitality industry to control the disparity among new systems.
- The Internet has become popular for allowing systems to share information and provide services.
- As systems become capable of using the same “protocol”, the problems with disparity will be reduced or eliminated.
- Purchasers must demand systems be compatible!

Standards will allow systems to be interoperable
Service-Oriented Architectures

• The Internet is...
  - A low-cost mechanism to allow applications to communicate, share data and process requests anywhere in the world

• Application Service Providers (ASP’s) rent web-based services to hoteliers for a fee. These services include:
  – HR Functions
  – POS Functions
  – Sales and Catering Functions
  – Reservation Functions
Summary

- Review current manual and automated processes
- Look for areas of improvement and optimization
- Identify the systems that will be impacted
- Choose a standard messaging format to exchange data
- Implement an open database standard to keep applications vendor-independent
- Integrate middleware to allow disparate systems to communicate
- Adopt industry standards wherever possible
- Demand that vendors follow these standards