OpenTravel Schemas

Getting Started

John Lambe, CTO

OpenJaw Technologies

OpenTravel Body of Work

- OpenTravel XML Schema Design Best Practices
  - Contains OpenTravel Schema naming and design rules
- OpenTravel Message Users Guide
  - Description of each OpenTravel message
  - Sample use cases
  - Sample XML instance documents

OpenTravel Body of Work

- OpenTravel XML Schema Definition files
  - Formal definition of the specification in W3C XML Schema
  - Download from OpenTravel public website at http://www.opentravel.org/
  - The XML Schemas are cumulative and contain all of the business messages defined by the OpenTravel

- Release Notes
  - The release notes detail the latest information and changes for any given release

Specification Philosophy

- ## Greatest Common Denominator
  - Specification built according to needs of numerous OpenTravel members
  - Many optional data fields exist
    - Implement only what you and your trading partners need need
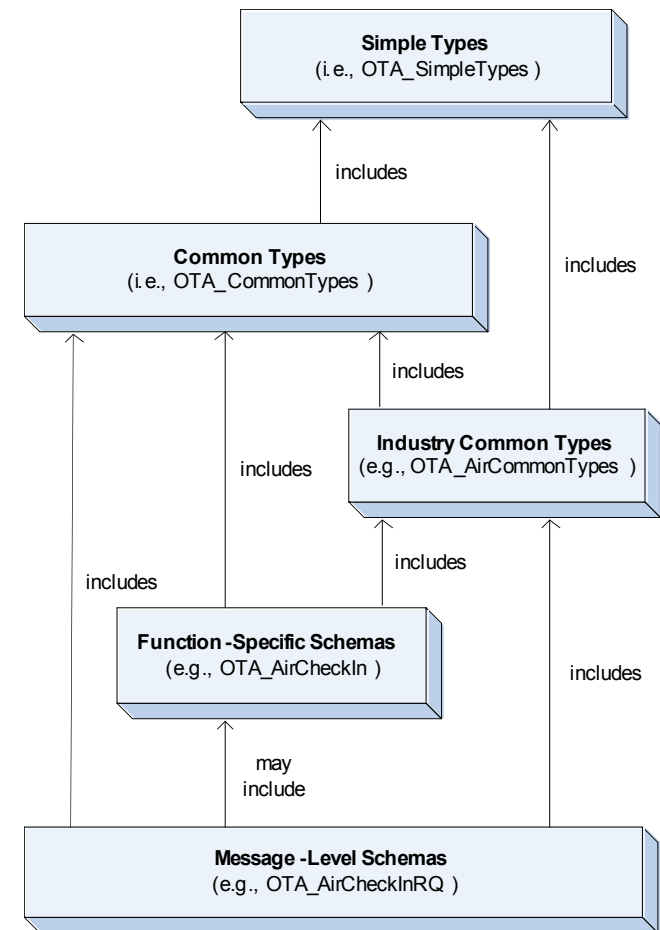   - Create internal documentation to support your implementation

Schema Architecture
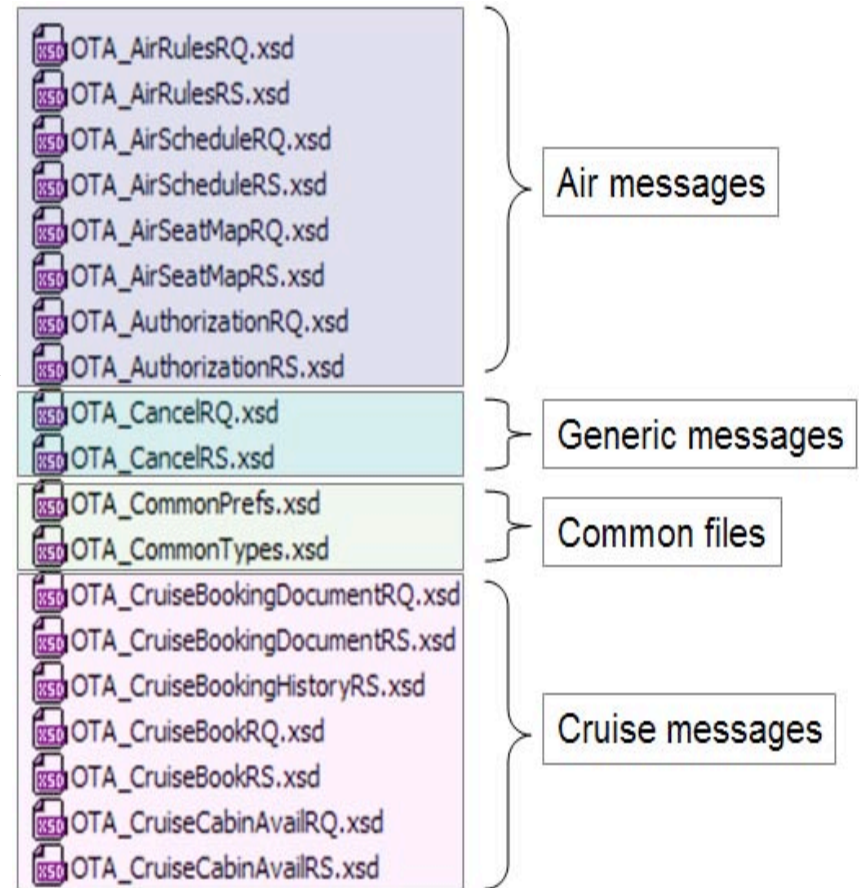
Hierarchical collection of schemas
that builds from reusable simple
structures into business messages

- Message Level Schemas
  - Represent a business transaction
  - E.g. OTA_AirBookRQ/RS
- Function Specific Schemas
  - Reusable functional structure
  - E.g. OTA_HotelReservation
- Industry Common Types
  - Complex structures for a specific vertical
- OpenTravel Common Types
  - Complex structures common to multiple verticals
- OpenTravel Simple Types
  - Simple structures common to multiple verticals

Simple Types
(i.e., OTA_SimpleTypes)

includes

includes

Common Types
(i.e., OTA_CommonTypes)

includes

includes

Industry Common Types
(e.g., OTA_AirCommonTypes)

includes

includes

includes

Function -Specific Schemas
(e.g., OTA_AirCheckIn)

may
include

includes

Message -Level Schemas
(e.g., OTA_AirCheckInRQ)

Schema Architecture (2)

- Namespace
  - Common to all messages
  - http://www.opentravel.org/OTA/2003/05
- File Naming Conventions
- Enumerations
  - Codified within the schemas
  - Expected to be static e.g. Days of Week
- Code Lists
  - Available as XML or Spreadsheet
  - Expected to be modified
- Success / Warnings / Errors
- Message Exchange Patterns
  - Request/Response (RQ/RS)
  - Notifications (Notif)

OTA_AirRulesRQ.xsd
OTA_AirRulesRS.xsd
OTA_AirScheduleRQ.xsd
OTA_AirScheduleRS.xsd
OTA_AirSeatMapRQ.xsd
OTA_AirSeatMapRS.xsd
OTA_AuthorizationRQ.xsd
OTA_AuthorizationRS.xsd
} Air messages

OTA_CancelRQ.xsd
OTA_CancelRS.xsd
} Generic messages

OTA_CommonPrefs.xsd
OTA_CommonTypes.xsd
} Common files

OTA_CruiseBookingDocumentRQ.xsd
OTA_CruiseBookingDocumentRS.xsd
OTA_CruiseBookingHistoryRS.xsd
OTA_CruiseBookRQ.xsd
OTA_CruiseBookRS.xsd
OTA_CruiseCabinAvailRQ.xsd
OTA_CruiseCabinAvailRS.xsd
} Cruise messages

Approaching the Specification

- Identify Functional Requirements
- Identify Non-Functional Requirements
- Find (or Create) the Message
- Implement the Specification
- Follow up

Identify Functional Requirements

- Document business process flow
  - Specification is primarily a set of data specifications that enable the automated exchange of data
  - Typically each data exchange will be executed within the context of a particular business process
  - Business process may affect the information exchange requirements
  - OpenTravel implementers should consider how new or existing OpenTravel messages operate within the context of a given process
  - To date, OpenTravel has not issued any formal business processes with the specifications

opentravel
setting the standard

European Conference
5 February 2008
Earls Court 2, London

Identify Functional Requirements

- Identify OpenTravel messages
  - Once the business process context is established, a company can identify the OpenTravel messages to be used
- Identify reusable content
  - When a company identifies the need for a new OpenTravel message, it will propose a new project

Identify Non-Functional Requirements

- Identify non-functional requirements
  - Be aware of how trading partners expose and maintain their services
  - Address security, performance, and service levels
- Define service architecture
  - Leverage OpenTravel as appropriate within a service architecture

Find (or Create) the Message

- Does a message exist that meets your requirements?
- If yes, move forward with implementation
- If not, work with OpenTravel to create the message
  - Submit project team proposal
  - Submit comments
    - Submit at any time, so as you find issues, enter a comment
  - Create draft XML schema
    - For new messages, a project champion may find it useful to submit a draft schema to jump start a project
  - Create business scenarios and instances

Implement the Specification

- Identify usage profiles
  - Each message may be used (at runtime) in different ways by different trading partners
  - Variation reflects the particular data needs that each company has with respect to a specific, published schema referred to within OpenTravel as a "usage profile"
  - To decrease the time required to develop service interfaces, companies should consider documenting their own usage profiles for distribution to their trading partners

Implement the Specification

- Define configuration management
  - OpenTravel publishes two specifications per year, each of which may provide incremental and useful functionality
  - Many companies find themselves maintaining multiple versions of the same XML schemas
  - Maintaining precise awareness and configuration control of schema versions running within a particular environment will be critical to success

- Test sample XML instances

Follow Up

- Register messages
  - OpenTravel provides an online site for OpenTravel implementers to register their messages
  - Registration captures a number of data points related to a company's use of OpenTravel
- Provide feedback
  - Identify data requirements not provided in OpenTravel messages
  - OpenTravel schemas provide temporary extension points by way of TPA_Extension elements
  - TPA_Extensions are intended to serve as a provisional means to exchange data
  - Implementers should submit comments to incorporate their requirements into future publications

Further Technical Guidance

- The Implementation Guide contains further advice on implementing the OpenTravel specification
  - Payload Transaction Management
  - State Maintenance
  - Message Transport
  - SOAP Messaging
  - HTTP Messaging
  - Web Service Description
  - Authentication
  - XML Data Binding

opentravel
setting the standard

European Conference
5 February 2008
Earls Court 2, London

Some Other Considerations

Before you start…..

- ## What are you trying to achieve?
  - Replace Existing Partner Connectivity?
    - Lower Operational Costs
    - Reduce Implementation Support
  - Support new distribution channels?
    - 3rd party websites
    - MetaSearch engines
  - Existing CRS / distribution technology does not support new products/pricing?
    - 'Add ons'
    - Taxes
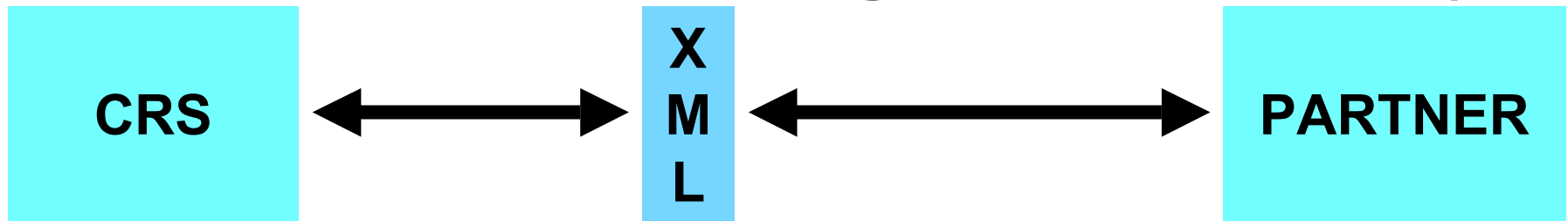   - Rate types (prepaid etc.)

opentravel setting the standard

{ European Conference
5 February 2008
Earls Court 2, London }

Two Approaches

- XML Interface to existing CRS functionality

- Updated functionality in Middleware tier

What functionality might you change?

- Hotels
  - Multiple room type search & book
- Flights
  - Flight & Fare Attributes
- All
  - Tax information
  - Service fees
  - Advanced Searches
  - Alternate results

What do your Distribution Partners want?

- Good Quality Descriptive Content
  - Coded, not UCASE TXT FR AGNTS
  - Multiple Language Content (Coded content enables translation)
  - Notification of content changes and errata
    - Distributed in a format that allows automated processing & import (ideally XML)
  - Standard Image Sizes with option to specify custom sizes

What do your Distribution Partners want?

- Accurate Pricing
  - Total price with full price breakdown (daily rates?)
  - Full Tax Information at search stage
  - Multiple currency support
  - Information on discounts applied (savings)
  - Cost of add-ons
  - Cancel and Modify costs

What do your Distribution Partners want?

- **Transactions**
  - Booking/Cancel/Modify are typically multi-stage processes
    - Payment (3d Secure?)
    - Itinerary Storage
    - Accounting handoff
    - Other Travel Products (packaging)
    - Cross Sell items (destination activities, insurance etc.)
  - Implement 2 Phase Transaction Capability

Best Practice Rules

- If something is complex, do it once on your side of the connection. Don't force every partner to implement the complex logic.

  - Connection Pools
  - Session Handling
    - Error Recovery
  - Searching / Booking Multiple Products
  - Cancel & Modify Cost Calculation

Accor OpenTravel Implementations

Olivier Lamy

OpenTravel Interfaces Techical Lead,
Accor Hotels

Overview

- Number and type of trading partners
- Connectivity models
- Functional implementation (availability, booking, etc.)
- Technical Overview
- Why open standards vs. proprietary schema?

OpenTravel Connectivity

- 1 GDS in full connectivity
- 2 Travel Web Sites
- 5 Travel Web Sites (Price Comparators, Geographical Search Engines ..) with the reservations done on the accorhotels Web Site
- Several projects in pilot phase, development or in project

Connectivity Models

We are currently supporting different connectivity models

- Push Model : all inventory details are send to the partner. Accor receive only booking requests.
- Shopping Model : nothing in the partner database except the hotel description.
- Hybrid Model : the partner handle on his side the multi property requests.

Messages Supported Receive Mode

Availability Search (OTA_HotelAvail)

- Multi properties search with different response modes (only prices range or detailled response)
- Single property with different response modes (all property offers or offer detail)

Messages Supported Receive Mode

Booking/Modification/Cancellation

- Two phase commit supported (type A) (OTA_HotelRes/OTA_HotelResModify/OTA_Cancel)
- One phase commit (type B) supported too (OTA_HotelResNotif/OTA_HotelResModifyNotif /OTA_Cancel)

Messages Supported Send Mode

- Availability Update : OTA_HotelAvailNotif

- Price Update : OTA_HotelRateAmountNotif

- Sales Conditions : OTA_HotelAvailNotif (FPLOS) Minimum/Maximum Stay and closed to arrival

Messages Supported Send Mode

- Hotel Content : OTA_HotelDescriptiveContentNotif

- Pre payment message with PSP : OTA_PurchaseItemRQ

Technical Overview

- Software written with Java language
- Open Source Technologies
- Transport layer for receive mode is http/s
- Transport layer for send mode http/s, ftp and scp

Conclusions…

The specifications are so huge that there are some different
ways to say the same things (IT guys called it : TIMTOWDI).

We have to build specific connectors based on a core
implementation due to partners implementations.

The real benefit : most of the time we can re-use messages
implementations with different partners.

Getting Started with OpenTravel Schemas

Darrin Talbot

Platform Technical Architect

(ba.com Selling),

British Airways

British Airways PLC

- International Airline
  - 36 million passengers
  - 850,000 + tonnes cargo
  - Number of subsidiaries, BA Holidays, Open Skies
- Producer and Consumer of data services
  - Providing data to meta-search engines
    - Flight availability - OTA_AirLowFareSearch
  - Consuming data from distributors (hotels, car hire, etc) for BA Holidays
  - Internal data services, OpenSkies/ BA Holidays
    - Payment – based on OTA base types
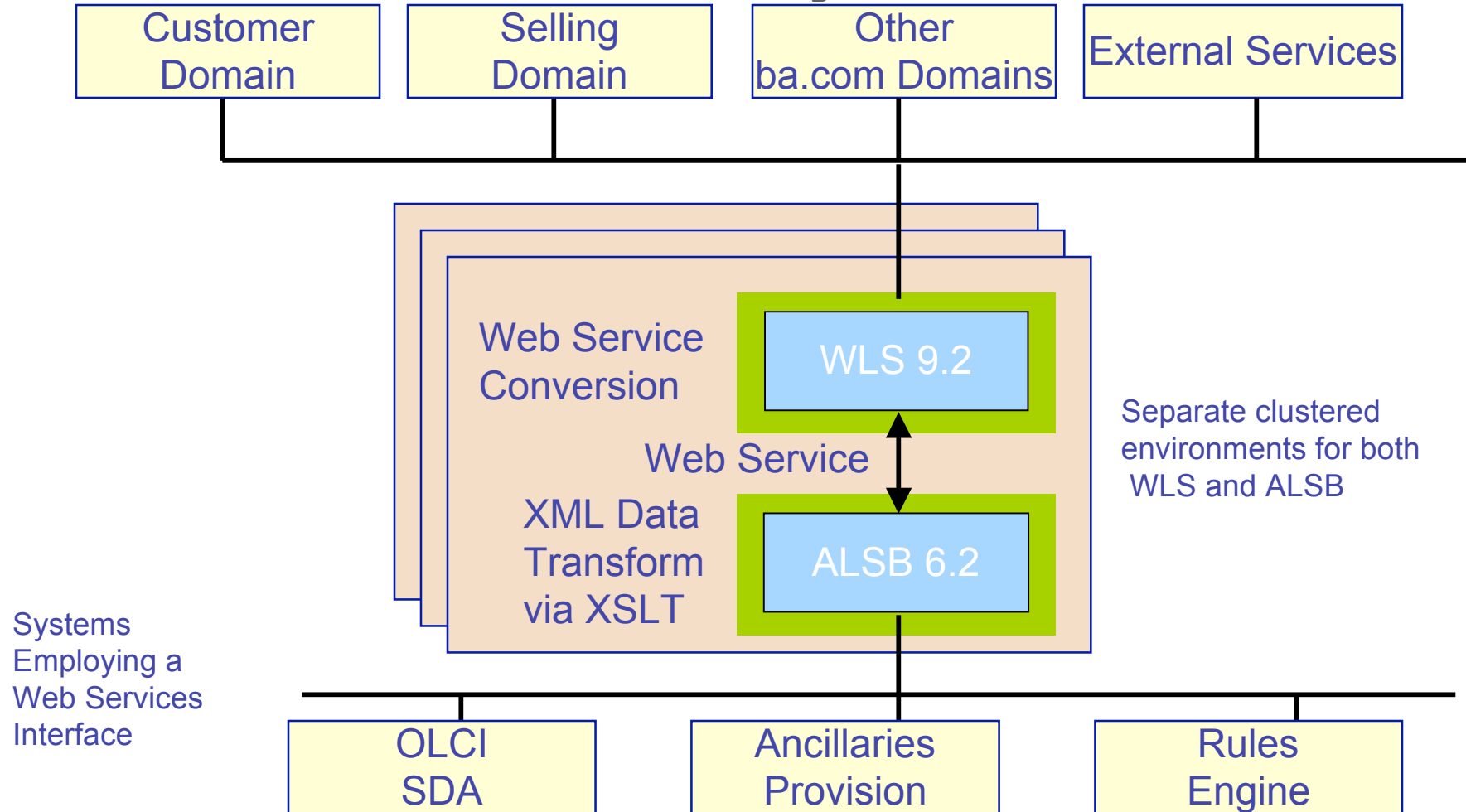   - Executive club member information
    - Booking creation OTA_AirBook

Technical Overview

ESB - SOA -Message Mediation

Customer Domain	Selling Domain	Other ba.com Domains	External Services

Web Service Conversion

WLS 9.2

Web Service

XML Data Transform via XSLT

ALSB 6.2

Separate clustered environments for both WLS and ALSB

Systems Employing a Web Services Interface

OLCI SDA	Ancillaries Provision	Rules Engine

Technical Overview

- SOA based architecture
- ESB – BEA (Oracle) Aqualogic
  - Web services with mediation
  - Java based platform
  - Services available of HTTP/HTTPS
  - Throttling provided by F5 load balancers
  - Message security provided by LDAP services
- 20 OTA based messages in use

Conclusions…

- Benefits of using OpenTravel schema
  - Lower development costs
  - Faster implementation time
- Lessons learned from implementations
  - Get to know the schemas
  - Governance and controls
- Future plans
  - More use of OpenTravel messages especially in the Ancillary arena
- It pays to adopt OpenTravel messages