

# Introduction to OpenTravel Schemas

Jessica Glace  
Specification Manager



**BEYOND  
THE TRAVEL  
TRANSACTION**



**2008 ADVISORY FORUM  
APRIL 14-16  
ATLANTA, GEORGIA**

# Prerequisite Knowledge

- Understand W3C Recommendations
  - XML
  - XML Schema including namespaces

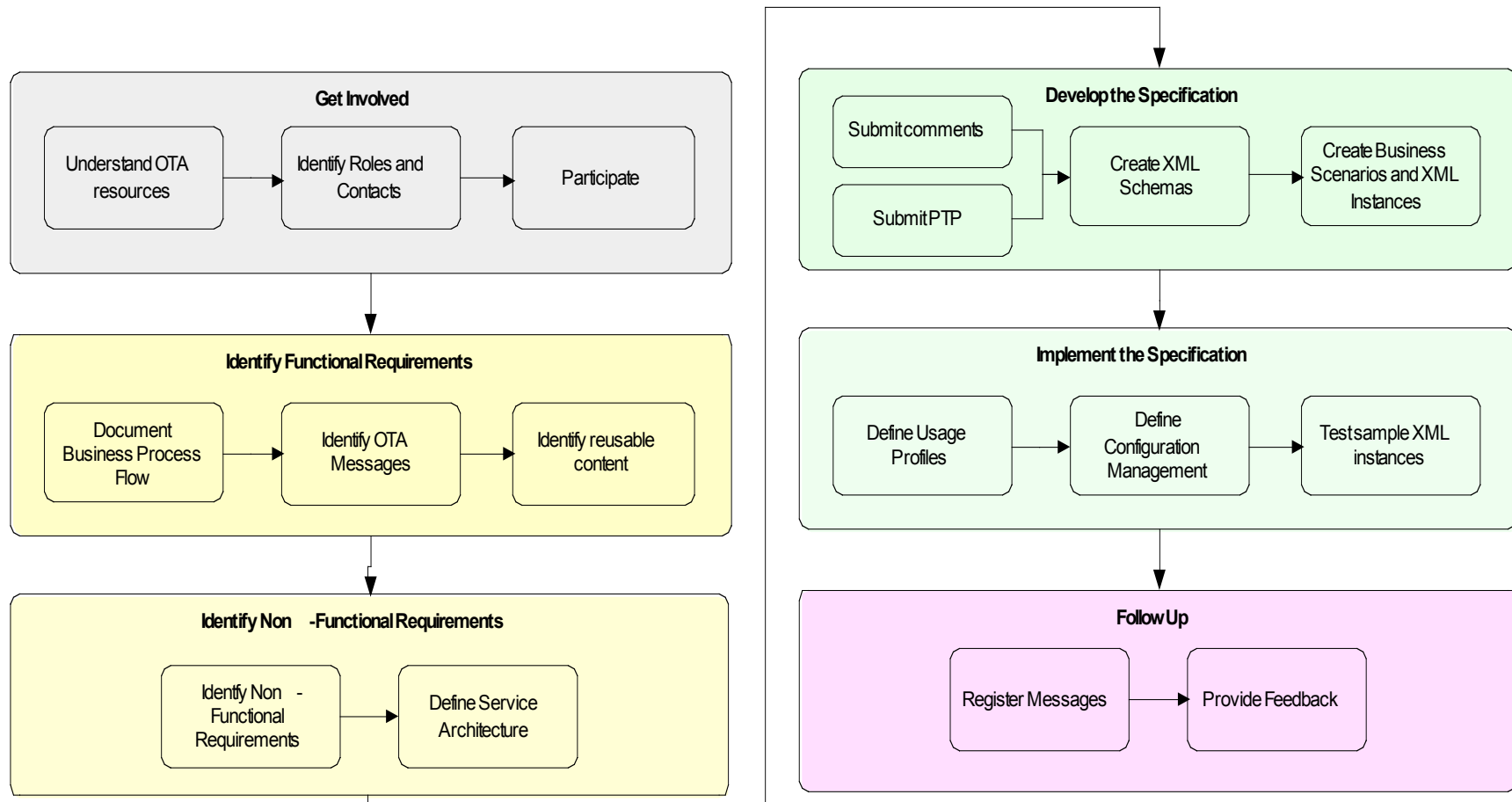


# Agenda

- OpenTravel within your Organization
- OpenTravel Schema Basics
  - Schema Architecture
  - Namespaces
  - File naming
  - Enumerations and code lists
  - Message exchange patterns



# Approaching the Specification



# Get Involved

- Understand the OpenTravel resources
- Identify OpenTravel roles and contacts
- Participate
  - Lead a project team
  - Participate in work groups, project teams and committees



# Identify Functional Requirements

- Document business process flow
- Identify OpenTravel messages
- Identify reusable content



# Identify Non-Functional Requirements

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



# Develop the Specification

- Submit project team proposal
- Submit comments
  - Submit at any time so as you find issues, enter a comment
  - <http://www.opentravel.org/Specifications/CommentOnSpec.aspx>
- Create draft XML schema
- 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 will reflect organizational data needs with respect to a specific OpenTravel message, called a “usage profile”
  - Consider documenting usage profiles for distribution to trading partners



# Implement the Specification

- Define configuration management
- Test sample XML instances



# Follow Up

- Register messages
- Provide feedback
  - Identify needed functionality, create a project team proposal
  - TPA\_Extensions
  - Submit comments



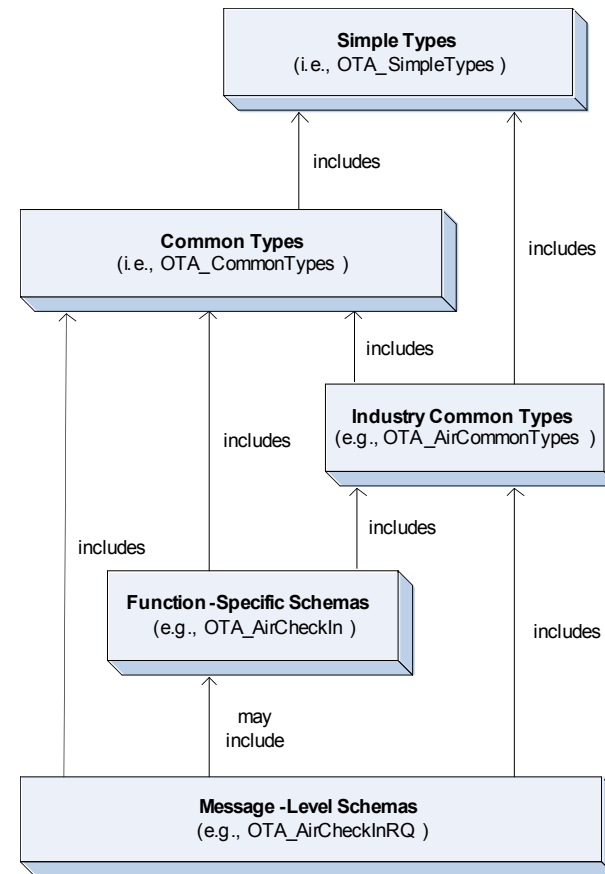
# Agenda

- OpenTravel within your organization
- OpenTravel Schema Basics
  - Schema Architecture
  - Namespaces
  - File naming
  - Enumerations and code lists
  - Message exchange patterns



# Schema Architecture

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



# Message-Level Schemas

Represent a particular type of business transaction

- [OTA\\_AirBookRQ.xsd](#)  
(Request airline booking)
- [OTA\\_AirBookRS.xsd](#)  
(Response to a request for airline booking)

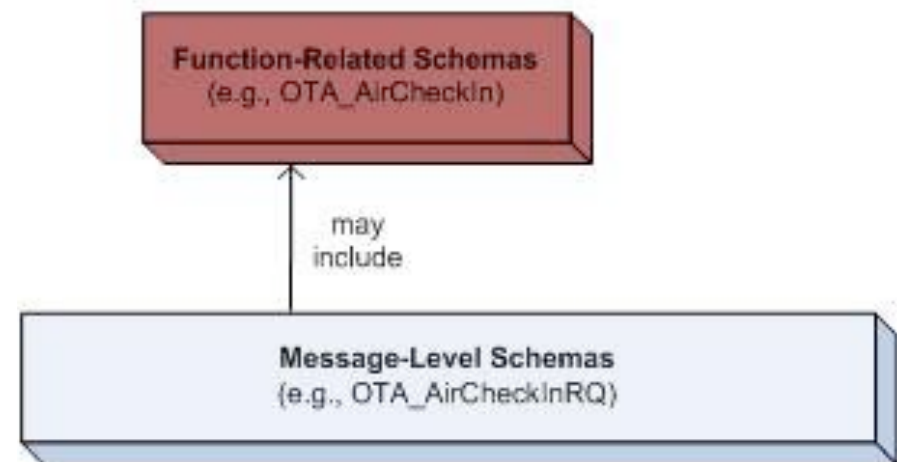
Message-Level Schemas  
(e.g., OTA\_AirCheckInRQ)



# Function-Related XML Schemas

Modules containing reusable structures used across messages of similar functionality

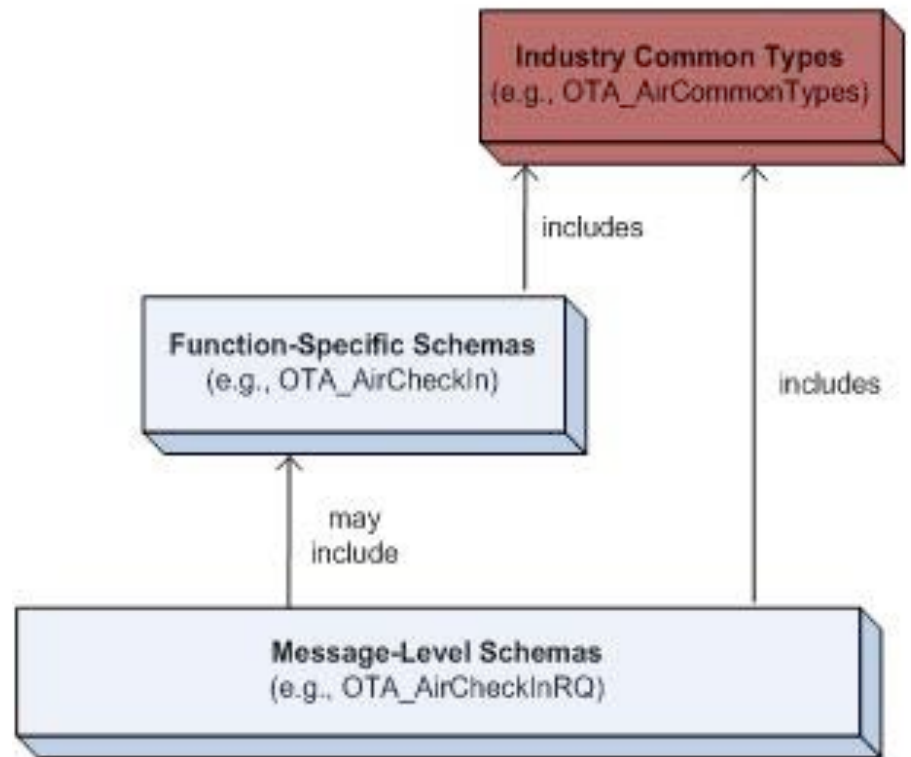
- `OTA_HotelReservation.xsd` is used in a number of message-level schemas for the hotel industry



# Industry Common Types Schemas

Provides a definition for complex types used in schemas for a particular industry

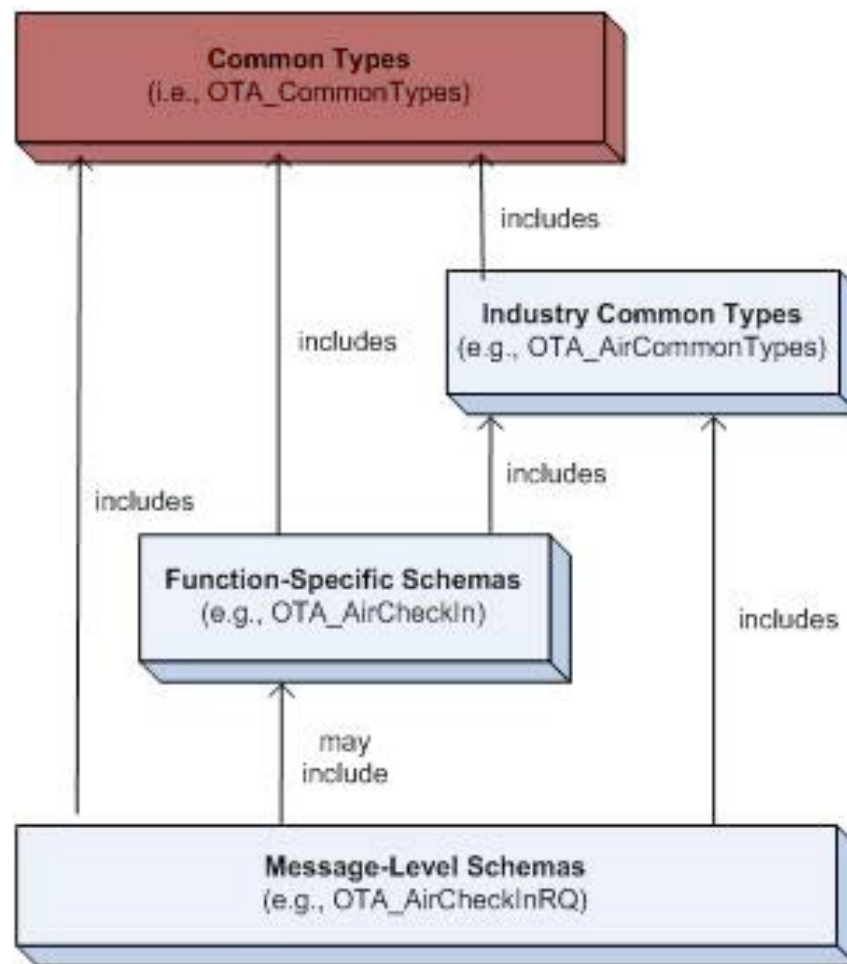
- OTA\_VehCommonTypes .xsd contains common types for schemas developed specifically for the car rental industry



# OpenTravel Common Types Schema

Provides a definition for data types available for use in schemas for all OpenTravel industries

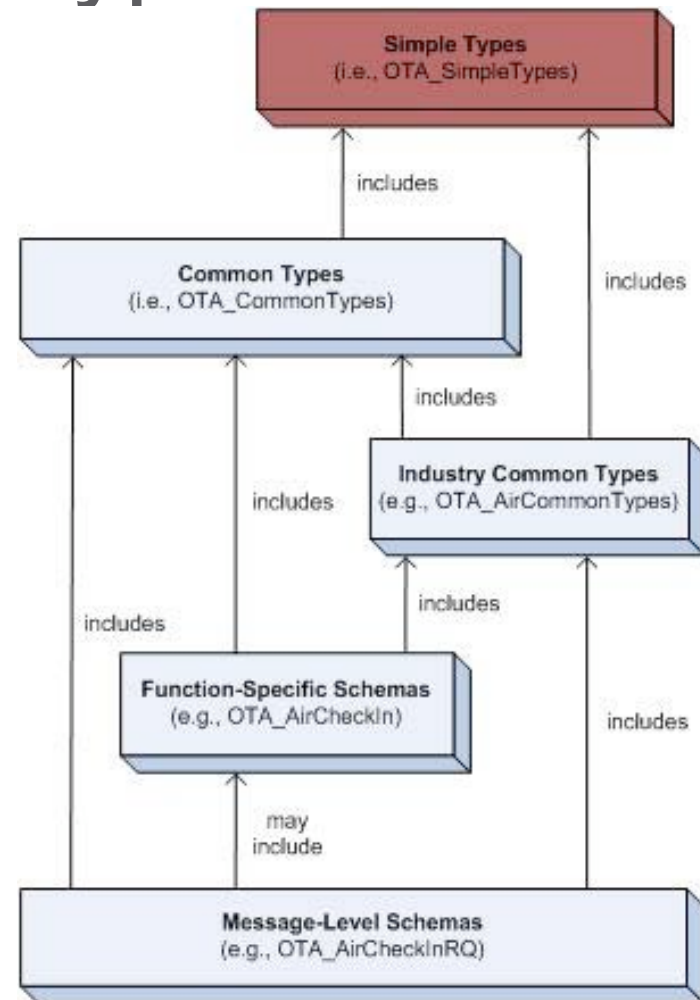
- `OTA_CommonTypes.xsd` contains common types for reuse and extension by industry common types, function-specific schemas, and message-level schemas



# OpenTravel Simple Types Schema

Defines simple types used throughout the specification

- `OTA_SimpleTypes.xsd`
  - contains reusable simple types built on W3C XML data types
- `simpleType="StringLength1to32"`
  - Any attribute of this type can contain a string value ranging from 1 to 32 characters (including spaces)



# Flattened Files

- Reduced processing
- Easier version compare
- Part of spec download



# OpenTravel Namespaces

- All message-level schemas are part of the same target namespace
  - <http://www.opentravel.org/OTA/2003/05>
- Member Review
  - <http://www.opentravel.org/OTA/2003/05/alpha>
- Public Review
  - <http://www.opentravel.org/OTA/2003/05/beta>
- Common schemas do not have a namespace



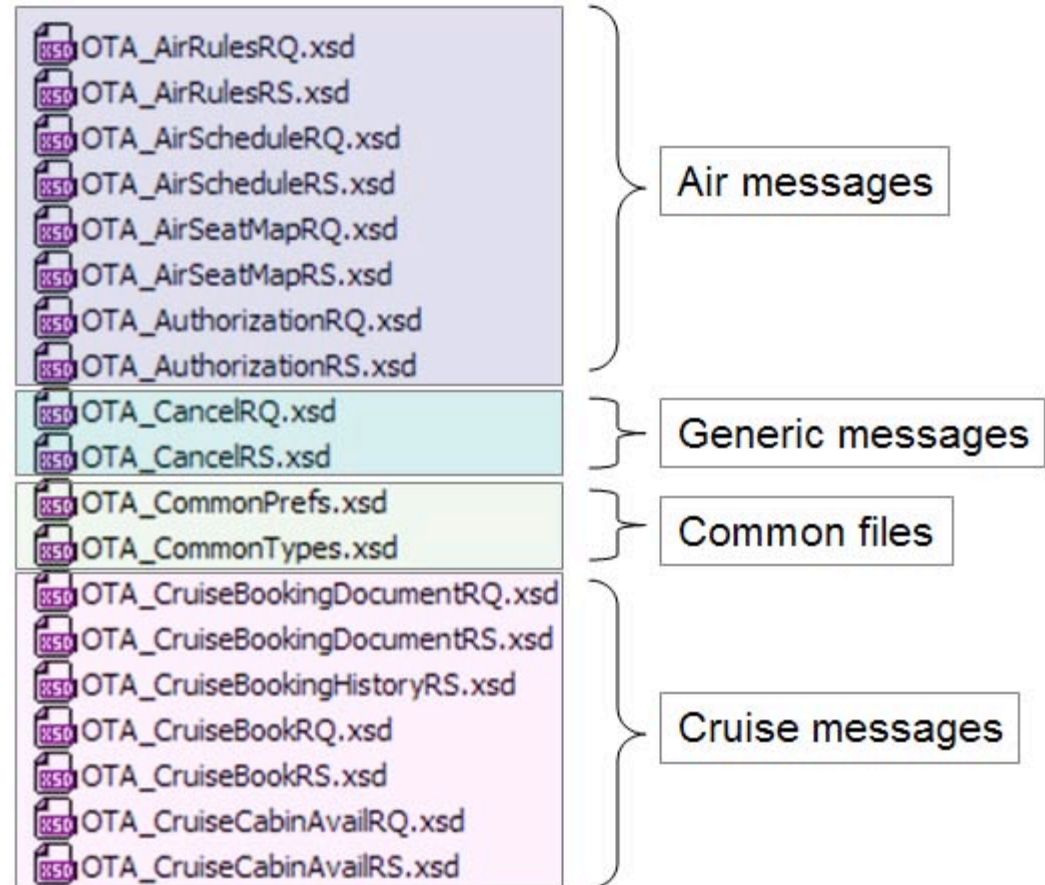
# Versions

- Major
- Minor (Backward compatible)
- version="1.009"
- id="OTA2008A"



# File Naming Convention

- Identical to root element
- Industry specific schemas are identified in the name
- Message-level schemas note the transaction type (e.g. “RQ” for a request)



# Enumerations & Code Lists

- Enumerations are codified in an element or attribute when values are expected to be static, or rarely modified
  - E.g. Days of the week
- Code lists, that are referenced by elements and attributes, are preferred where modifications are likely
- OpenTravel and external code lists are supported in the specification



# OpenTravel Code Lists

- References to code lists are of the simple type: “OTA\_CodeType”
- OpenTravel code lists are available in spreadsheet and XML format



# Example OpenTravel Code List Reference

```
<xs:attribute name="UseType" type="OTA_CodeType"  
  use="optional">  
  <xs:annotation>  
    <xs:documentation xml:lang="en">Describes the use  
    of the address (e.g. mailing, delivery, billing, etc.).  
    Refer to OTA Code List Address Use Type  
    (AUT).</xs:documentation>  
  </xs:annotation>  
</xs:attribute>
```



# Example OpenTravel Code List

Code list name

Code list 3-letter code

<b>Content Format Code</b>		<b>CFC</b>	
	1	mpeg	Standard internet movie platform
	2	mpg	MPEG digital video format
	3	avi	Audio video interleaved
	4	wmv	
	5	jpeg	Joint photographic experts group
	6	jpg	24 bit graphic format

Code value

Code name

Code definition

# OpenTravel Code List Additions

- E-mail the appropriate work group distribution list:
  - the code list name (e.g., Address Use Type)
  - the code list 3-letter code (e.g., AUT)
  - the new code name (the actual code value will be assigned later by the code list manager)
  - the OTA message and element path where the new code is to be used

**All information is required for request to be considered.**

# OpenTravel Code List Additions

- Allow two weeks for questions and comments from the work group.
- If no objections, forward request to [<codelist@opentravel.org>](mailto:codelist@opentravel.org) and indicate that there have been no objections.



# OpenTravel Code List Additions

- Code list manager reviews request and, if no issues, assigns a code value to the new code and distributes new code to the requestor.
- If there is an issue, the code list manager may query the work group.



# Message Exchange Patterns

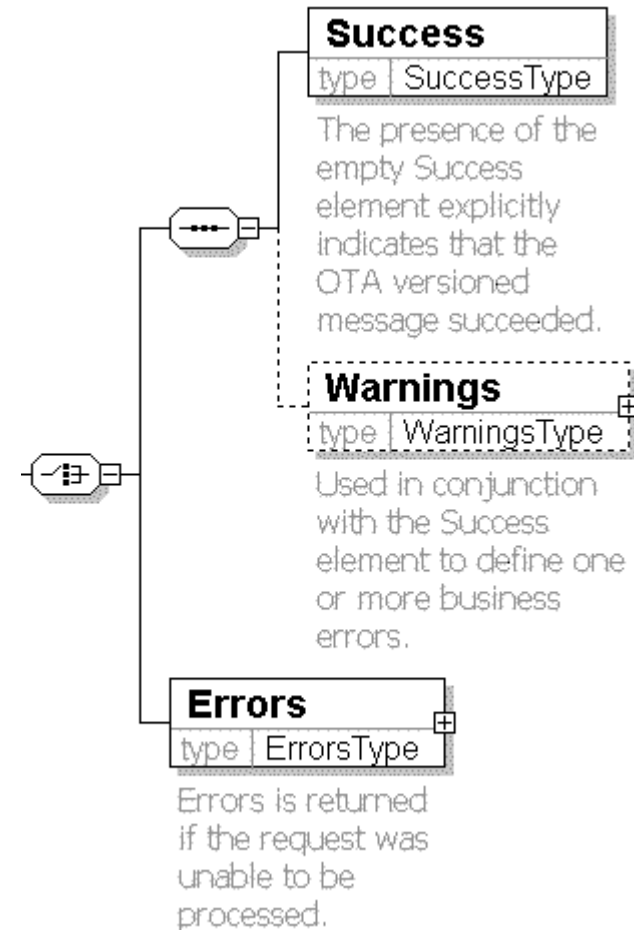
- Request/Response (RQ/RS)
  - Starts with a “pull” request for information or action (e.g. OTA\_CruiseBookRQ.xsd), with the expectation of a response appropriate to the request (e.g. OTA\_CruiseBookRS.xsd)
- Notification (Notif)
  - Starts with a “push” message to send unsolicited data (e.g. OTA\_HotelInvNotifRQ.xsd)
  - A “Notif” response message provides processing status information (e.g. OTA\_HotelInvNotifRS.xsd)



# Success/Warning/Errors

Common structures for reporting back processing status to a trading partner

- Success: An empty element, whose presence indicates the message was readable
- Warnings: Accompanies a Success element to indicate business issues with the message (e.g. requested data not found)
- Errors: Indicates a message could not be processed



# OTA\_PayloadAttributes

- The root element of each message-level schema refers to the OTA\_PayloadStdAttributes attribute group
- This attribute group is defined in the OTA\_CommonTypes file and contains attributes that, collectively, support common transaction management requirements
- At runtime, these attributes operate within the XML payload rather than the envelope layer



# OTA\_PayloadStdAttributes

attributeGroup		OTA_PayloadStdAttributes		ann:The OTA_PayloadStdAttributes defines		
Attributes   Identity constraints						
Name	Type	Use	Default	Fixed		
EchoToken	StringLength1to128	optional				
TimeStamp	xs:dateTime	optional				
Target	xs:NMTOKEN	optional	Production			
Version	xs:decimal	required				
TransactionIdentifier	StringLength1to32	optional				
SequenceNmbr	xs:nonNegativeIntege	optional				
TransactionStatusCode	xs:NMTOKEN	optional				
grp PrimaryLangID_Group						
PrimaryLangID	xs:language	optional				
grp AltLangID_Group						
AltLangID	xs:language	optional				
RetransmissionIndicato	xs:boolean	optional				

# OTA\_PayloadStdAttributes Example

```
<?xml version="1.0" encoding="UTF-8"?>  
<OTA_VehLocDetailsNotifRQ  
  
  xmlns="http://www.opentravel.org/OTA/2003/05"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05 OTA_VehLocDetailsNotifRQ.xsd"  
  
  EchoToken="1234567"  
  TimeStamp="2006-05-17T09:30:47-05:00"  
  Target="Production"  
  Version="1.000"  
  PrimaryLangID="en-us">
```



# Further Resources

- Websites
  - <http://www.opentravel.org>
  - <http://wiki.opentravel.org> (some content is member only)
- Mailing list
  - Each work group maintains an electronic mailing list
  - Only available for member use
- Advisory Forum
  - Annual meeting to learn and discuss issues around OpenTravel specifications



# Presentation Basis

- OpenTravel Implementation Guide
  - Version 1.0, August 2007
- Available to OpenTravel members on the Resources page of the wiki:
  - <http://wiki.opentravel.org/index.php/Public:Resources>



# OpenTravel Documents

- Specifications
- Design Best Practices
- Message Users Guide
- Implementation Guide



# 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



# Introduction to OpenTravel Schemas

Jessica Glace  
Specification Manager



**BEYOND  
THE TRAVEL  
TRANSACTION**



**2008 ADVISORY FORUM  
APRIL 14-16  
ATLANTA, GEORGIA**