

Homework 8
1.264, Fall 2013
XML and Web services
Due: Wednesday, November 20

A. Overview

In this homework you will create and use XML files and Web services for your aircraft parts distributor. The steps in the homework follow lectures 21-23.

Create a new Web site, Homework8, with the following files/pages. You will also design but not implement some additional Web services within a service oriented architecture (SOA).

Your Web site will support “raw” XML document transfer and validation (steps 1 and 2), a Web page that displays XML data to a human user (step 3), a Web service that can be used by your customers (step 4), and a Web page that displays the results of a Web service to a human user (step 5). Usually a Web page displaying data for a human user will obtain the XML file or Web service result from a Web server operated by a different organization (e.g., your vendor or your customer), but you will use just your own organization’s XML and Web service in this homework; we/you don’t have a second Web server running to emulate the full business-to-business data exchange.

In step 6, you will design additional Web services that would be built to provide a full Web service/SOA implementation for your aircraft parts distributor. In step 7, you will draw a diagram that shows how your Web services interact with your customers’ Web services.

B. Assignment

1. Create an XML file to represent an original aircraft part and its replacement part. Name it airPart.xml. This will have several attributes. The XML must contain:
 - i. Original part number
 - ii. Manufacturer
 - iii. Part name
 - iv. Replacement part number
 - v. PMA holder for replacement part

Create a single XML file that contains at least two original aircraft parts, each with a replacement part. You may choose your own tag names, but they should correspond fairly well with the database column names. Make sure your XML file will display in Internet Explorer or Firefox. Make sure the + and – tree controls work to collapse and expand parts of the XML document.

2. Write a DTD to validate the XML file; name it aircraftPart.dtd. Create the file in VWD; the editor will help you a little as you write it.

3. Create an aircraftPart.aspx page to display the aircraft part data in the XML file using an XMLDataSource and a DataList, as in lecture 22. Use XPath queries similar to exercise 1 in lecture 22 to get the data from the XMLDataSource. (Or use the nicer format based on the BookMIT2.aspx file in the download.)

4. Modify the ProductService.aspx Web service in lecture 22 to return the aircraft parts associated with a MIDO. The MIDO is the parameter for the Web service. Call your Web service aircraftPart.aspx.

5. Following the example in lecture 23, use your Web service in an aircraftPart2.aspx page to display the aircraft parts in a gridview. The ObjectDataSource can use a default parameter (choose any MIDO), following the lecture example. For a nicer page, place a gridview or dropdown list with the MIDOs at the top of the page, and then use the selected value of MIDO as the parameter in the ObjectDataSource to show just the aircraft parts for that MIDO.

6. List the Web services you would include in your aircraft parts Web site. These would be registered (as you did for your parts service in step 5) and available for your customers to use in their systems that check your catalog, place orders, check status, and make payments. Most, if not all, will essentially be database queries. List the parameters that each Web service requires, and the result that it returns. For example:

Service: GetReplacementPartsService(OriginalPartNumber, QuantityNeeded).

Returns: ReplacementPartNumber, PMAHolder, QuantityAvailable, Price.

Logic: QuantityAvailable is maximum of QuantityNeeded, QuantityInStock.

Service: OrderFulfillmentService(OrderHeader, OrderDetail, Carrier, Mode)

Returns: OrderID, OrderStatus

Logic: OrderDetail as defined in OrderDetail table. OrderStatus gives detail on items in stock or back-ordered, whether carrier/mode are available, expected ship date, etc.

You do not need to write the SQL. The logic note is needed only if there is other processing beyond the obvious SQL queries in the Web service. You do not need to specify the data types (int, string, etc.) of parameters or return result. List no more than 15 services.

7. Draw the SOA diagram (similar to Figure 5.4 and/or Figure 5.15 in lecture 24) for your aircraft parts system, but **only for its interactions with customers (including their bank) and carriers. Do not include interactions with suppliers.** Create descriptive names for the Web services on the carrier and customer sides of the interactions that you will use on your drawing; you do not need to define the customer or carrier Web services, just name them. Use the list of Web services for the aircraft parts distributor that you defined in the previous step. You don't need to show WSDL verifications in your diagram.

Draw your Web services, and draw the Web services an aircraft maintenance company that buys parts would use to communicate with your Web services. Use the last 6 slides of lecture 24 (especially the one labeled 'Manufacturer information flow'), which describe the core components of an Enterprise Resource Planning (ERP) system, as a framework: make your Web services consistent with the ERP framework, as much as possible. Your drawing may be in PowerPoint or other common drawing tool, or you may draw on paper and hand it in as hardcopy.

C. Hand in

Create a zip file from the pages and files in your Web site, and the answers to parts 6 and 7, and upload it to the course Web site.

MIT OpenCourseWare
<http://ocw.mit.edu>

1.264J / ESD.264J Database, Internet, and Systems Integration Technologies
Fall 2013

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.