PICASSO'S INVENTORY SYSTEM:

A PROJECT THAT LEVERAGES XML TO CREATE, SAVE, QUERY, AND DISPLAY A SYSTEM'S INVENTORY

Fernando Zamora

Letitia Murdoch

Manfred Ayers

Mark Thompson

Department of Business

CIS 545 Extensible Markup Language

May 1, 2010

Abstract

We wrote this paper as part of a project for an **XML** class at the graduate level. The paper consists of three major parts; **XML Creation** (including XSDs, CSS, XML), **Data Entry Application**, and a **Paper describing the effort**. The project is broken down into two major subprojects; the actual inventory for a fictional business and the paper that describes the our process details, as well as the schema that this paper will ultimately describe the elements of this paper. This paper is only a portion of an entire project.

This project consists of the following artifacts that go along with this paper:

- Paper Related Artifacts
  - ProjectPaper.xml
  - ProjectPaper.xsd
  - ProjectPaper.html
  - PaperToHtml.xsl
  - Images Folder with the following images
    - chicagopaperschema.png
    - htmlscreenshot.png
    - inventoryschemahighlevel.png
    - inventoryscreen.png
    - vendorscreen.png
    - catalogscreen.png
- Inventory Related XML Artifacts
  - PicassosInventory.xsl
  - PicassosInventory.xml
  - PicassosInventory.xsd
  - PicassosInventory.html

- Images folder with the following:
  - PicassoLogo.png

Introduction

Picasso's Frame Shop

This paper is part of a project that will demonstrate how to use Extensible Markup Language **(XML)** to maintain a small inventory system for a small business. The objectives of this project are to create the data schemas, the XML documents as well as the data entry system to create and modify the XML document. Also to be included as part of this project, is a technical document that describes the project and process in creating the inventory system. A requirement for the technical document is that it must be written in a way that it can be treated as XML data. This way the XML document can be formatted by an Extensible Stylesheet Language **(XSL)** document an processing it using what is known as XSL Transformations **(XSLT)**. This paper will address how the users of this system will be able to create, share, modify and also query the inventory data. Additionally, this paper will address how the decisions were made the creation all of the artifacts. This paper will describe the small business that was targeted by the XML data for this project and why. Finally this paper will briefly described the schemas, content and XSL necessary to transform this paper into a properly formatted *Chicago Manual of Style* paper

Since XML is well suited to maintain a small hierarchical database, it has been chosen as the data persistence of choice for this project. Because this paper is a requirement for an XML class the requirements met by this paper were those provided to us. Nevertheless, XML allows for data definition in the form of Document Type Definition **(DTD)** documents and XML Schema Definition **(XSD)** files. We decided early on that it would be easier to stick with one type of schema language. Therefore the schema language

chosen was XSD over DTD. All of the schemas for this project are XSD files instead of DTD files.

Like any other school project we had some level of freedom in the decision-making process. Some of this freedom was the ability to choose a small business of our liking. Because Fernando has extensive experience in the frame making business, we decided to go with a hypothetical small frame shop as our business. Like all hypothetical business we needed a name for our business and since the business was somewhat related to the art world, we decided to call it Picasso's. Therefore, the inventory system was named the Picasso Inventory System.

The inventory system is a starting point, rather than a complete solution to the inventory system needs of Picasso frame shop. This is true of the schemas provided in this presentation as well. With that disclaimer out of the way, after careful analysis, the schemas results meet the factors important in maintaining adequate inventory within the shop. The design will allow the Picasso shop to minimize downtime by maintaining adequate inventory, determining pricing requirements, and maintaining our suppliers' contact information.

The Picasso's Frame Shop Business

Description of the Business

Picasso's Frame Shop is a small family owned and operated business. The business provides custom framing needs for the Temple, Killeen and surrounding areas. Custom

framing provides a higher quality and more elegant solution as well as higher variety for any framed artifact than traditional off-the-shelf frames and mats. Although off-the-shelf frames meet the requirements for most photographs and many posters, custom framing provides many benefits that off-the-shelf frames and mats cannot provide. Among these are:

- Non-Standard Frame Sizes
- Higher Selection of Moulding
- Necessary Protection of High Dollar Prints and Photographs
- Custom mat cuts not available with off-the-shelf mats
- Personalized Color Schemes to fit the framed artifact best
- Framing of non-traditional artifacts such as baseballs, flags, awards etc

Picasso's cuts every mat, moulding, and sheet of glass to customer desires specifications. For these reasons, the Picasso Frame Shop must be prepared to stock the most common supplies and be able to provide a high selection of custom ordered supplies. Because custom framing competition is quite fierce in the Killeen/Temple area, Picasso's must maintain the right quantities of the most common supplies. This is all in an effort to provide a quick and efficient turnaround for all orders.

Most orders have a common turnaround span of five days from order to delivery. In the case where a special order is required, the turnaround time can be as much as two weeks longer than normal. For that reason, Picasso's must maintain a high selection of items. This also helps reduce the number of custom vendor orders. This is all in effort to keep customers coming back to Picasso's. This lets the customers know that their needs will be met with a high degree of quality and punctuality. At the same time Picasso's must also identify those items that simply do not sell well or have been discontinued by the

vendor.

Custom framing requires a high degree of precision in pricing. This is so that store can be competitive, yet at the same time have the capability to price its services at a profit. This means that Picasso's must take all costs incurred into account in order to price the items adequately. The cost of these supply items is a significant consideration of those costs.

Framing Equipment and Supplies

The delivery of a custom framing order requires the use of several pieces of equipment and the consumption of several supplies. The equipment, composed mostly of machines, is maintained by employees. The adequate inventory of the parts and supplies consumed by these machines, must be maintained on hand to prevent any loss of productivity.

Picasso's equipment consists of the following machines:

- 3 Mat Cutters - Fletcher
- 1 Moulding Chopper
- 1 Moulding Miter Saw
- 1 Computerized Mat Cutting Machine – Wizard Model 8000
- 1 Oval Mat/Glass Cutter
- 1 Regular Glass Cutter
- Dry Mount Press
- 2 Miter Vice
- Other hand tools such as tape guns, multi-point drivers, wire cutters, screw drivers, utility knives, pencils, etc,

The equipment orders do not occur on a regular basis like the supplies, but the parts that are subject to heavy wear and tear are. Therefore, Picasso's must have the ability to maintain adequate inventory of the parts. These parts include but are not limited to the following:

- Brads for the miter saws
- Brads for the joiner (underpinner)
- Blades for the mat cutters
- Points for the multi-point drivers
- Wheel Blades for the glass cutter

Additionally Picasso's must maintain an adequate selection of Mats, Glass, and Molding. These items are usually the ones that the customer has the options to choose from. While there are no exact levels of prediction as to what items the customer will choose next, it is important to attempt to stock the items that are most likely to be chosen. Of equal importance is the need to maintain enough variety so that the customer has alternate options. The supplies to be maintained in inventory include the following:

- Mats varied by different colors and types
- Molding – This is the actual pieces used for the frame
- Glass – There are several types and sizes
- Foam Core Backing
- Glue
- Glass Cleaner
- Hanging Wire and Screws
- Pencils
- Other miscellaneous items

Maintaining these items in adequate quantities is necessary for successful operation of the shop. One caveat is that that not all purchases are individual item purchases. Some items sell individually while others sell in boxes and by different measures such as by the foot. These were all considerations in designing our system.

Vendors

In a perfect world it would be ideal to purchase all equipment and supplies from one vendor. This would allow us to maintain one contact and deal with only one vendor. However, in the real world not all vendors are created alike. Vendors differ in various aspects. These aspects include location distance, product types and pricing, just to name a few. As an example one vendor may carry a certain item that the other vendor carries but the price from one vendor to the other differs significantly. As another example, one vendor may focus on supplies while another may focus on equipment and equipment parts. Additionally one vendor may be located closer, thereby providing quicker and cheaper deliveries. At other times one vendor may be out of stock on a particular item while another vendor is plentifully stocked. All these examples and explanations are used only make the point that Picasso's needs to do business with several vendors and also maintain contact information for those vendors.

<div align="center">

The Inventory System

</div>

Description

Picasso's inventory system consists of several artifacts that allow all of them, together, to operate as a single inventory system. The artifacts within the system are the picassosinventory.xsd, the XML data, picassoinventory.xsl stylesheet to enable viewing of the system through a browser. Also included, as part of the system, is the **Picasso Inventory System** which enables the **CRUD** (Create, Update, Read, Delete) operations on the data.

Rather than diving deep into the technical details of the data definition we will explain the schema in plain and clear English. We identified the need to maintain information on each significant element of the inventory system. We also identified their significant attributes. Below is a list that is structured in a hierarchical format of the significant elements with their attributes:

- Vendors
  - Id
  - Name
  - ContactPerson
  - Phone Number
  - Comments
  - Address
    - Street
    - City
    - State
    - Zip
- Catalog Item
  - Id
  - Vendor Id

- Unit Cost
- Category
  - Single Item
  - Boxed Item
    - Quantity in Box
    - Box Cost
  - Volume Item
    - Unit of Measure
    - Minimum Quantity
- Catalog Item Id
  - Comments
  - Inventory Location
    - Name
    - Description

With these needs in mind, we created the XSD schema located in appendix A.

In the tree-like structure of elements identified above, we aimed to keep our data definition simple and to the point. We also took into consideration that once in production the schema could possibly have different requirements. We deferred the finer detail schema decissions downstream in the schedule. We determined that every entity within our schema required an ID attribute. This is so that the system would avoid data conflicts based on other attributes. Therefore all of our entities identified above have an ID. Additionally each ID within the document is unique. This was also done to reduce the nesting of entities within other elements. Instead the entities are referenced by ID from the other entities.

Vendor Entity

The starting point for our data system is the vendor. The reason is that without a vendor there is no product. The vendor in our system carries some basic attributes like name and address. We also deemed it important to store the contact information. This led us down the path of placing the contact person and phone number. We did not think that it was necessary to break down the contact name any further than that. We did acknowledge that we would miss a significant attribute. Therefore, we decided to add the comments attribute. The comments attribute is important enough to exist in every entity within the system. That is why all of the entities have a comments attribute. The comments attribute is a catchall attribute that allows for entering comments about an entity.

Catalog Item

The catalog item is the actual product that we are concerned with in our system. Even though we acknowledged the possibility that two of the same catalog items can exist from two totally different vendors, we determined that each item has enough attribute differences to be separated by vendor. Among these differences are the product assigned vendor id, the packaging and the slight change in attributes. A possible consideration for the future is the concept of somehow tagging an item so that we can determine items by tag even if they are from separate vendors. This is a concept that would basically assign the same product id to an item even though the item can come from two different vendors. At this point this is only concept that requires further evaluation.

The catalog item has something known as the "vendor assigned product id". This vendor assigned product id is based on the fact that vendors identify items by id's within their

catalog. While it is unlikely, it is possible that two vendors may carry the same vendor assigned product id for one of Picasso's inventory products. For that reason the same vendor assigned product id can exist within different products so long as the products are from different vendors. The other important attributes for the catalog id are the description and the unit cost.

We also recognize that how an item is packaged can affect the category placement. Items can be sold individually, boxed or by some other measure which we decided to call volume. While the hierarchical category above identifies the category as an attribute of the catalog item, in actuality we decided to separate the catalog items into different elements: Catalog Item, Boxed Item, and Volume Item. We made this determination to avoid forcing one element to share non-required properties (i.e. properties that exist if within one category but not within another). The boxed item, for example, has a box cost and a unit cost. While the Catalog Item (single item) only has the unit cost requirement. Additionally the Volume Item has the requirement to determine the unit of measure such as feet, gallons, etc.

Inventory Item

Once we had the vendors and catalog items in place, the stage was set for our prime entity – the inventory item. For the inventory item we need to know all of it's catalog item related attributes as well as the physical location. Since the catalog item is tied to the inventory item this can also reveal, not only all catalog information, but also all of the vendor information. For the inventory items, Picasso's needs to know the quantity on hand as well as the Re-Order Point (ROP) and the Re-Order Quantity (RO). The ROP is

simply the point that, when reached, triggers a re-order action on the part of Picasso's staff. The RO is the quantity that must be ordered when that action is triggered.

An important mention that cuts across all of our entities is that the ID's of each entity are and must be unique within the document. Also, we decided to not model our entities after real-world domain objects of the custom framing world. In other words we did not head down the path of creating a mat element, or a glass element or a molding element for that matter. We determined that it was better to make it generic and flexible enough to work with any inventory system. This decision was also influed by the numerous types of parts and supplies available in the framing business. This is also the reason that this data definition could work for many other business models.

Gluing It All Together

Therefore, everything in the schema is in place but the pieces remain disjointed. Driven by the fact that a well-formed XML document requires a root element we need some additional elements to put all the pieces together into one big orderly tree. For this, we created the additional elements. Below is a description of the tree as well as the XML Spy graphical view of the tree.

- Product Graph
  - Catalog
    - Catalog Items
      - Catalog Item
  - Boxed Items
    - Boxed Item

- Volume Items
  - Volume Item
- Inventory
  - Inventory Items
 - Locations
- Suppliers
  - Vendor
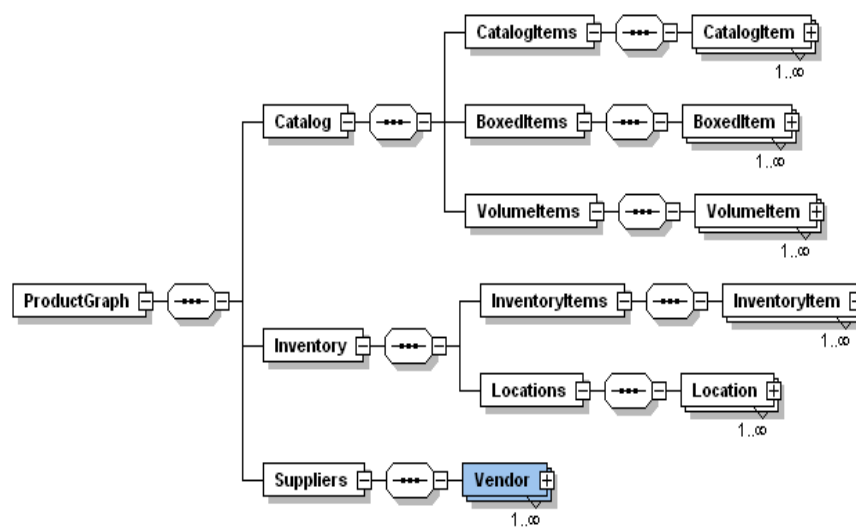
Figure 1. Inventory Schema High Level

The Picasso Inventory System

Data Entry Application

Once we created our database definition, we also had to tackle the concept of creating a simple data entry system. This was decided so that the user has an easy and efficient way of creating valid data. We will not delve into the fine-grained details of the system but we will point out that the database entry fully supports the data validation required by the products.xsd schema. The application code is in C# using the 3.5 .Net framework. The application design uses best practices, with a tiered approach. In this case, our three tiers included the following tiers:

- Database – In the form of an XML data file
- Data Access Layer – The mechanism to read the file was file IO
- The Model was in the form of Plain Old C-Sharp Objects (POCOs)
- The UI Layer was in the form of a windows form with three tabs.
- This mean one tab for each entry.
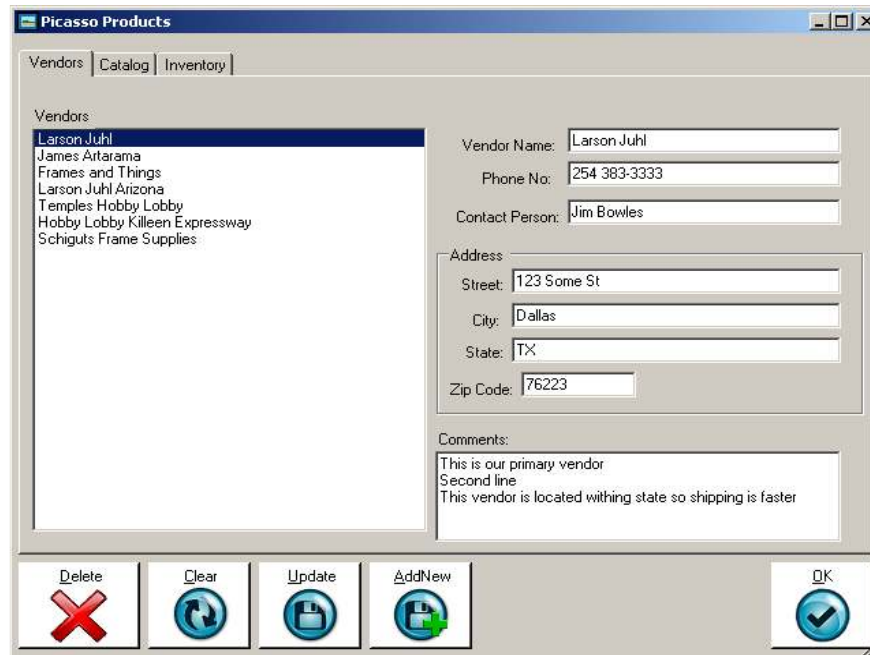
Figure 2. Vendor Screen

Figure 3. Catalog Screen

Figure 4. Inventory Screen

The Picasso Inventory System allows the creation of a new inventory database. It allows

all basic Create, Read, Update and Delete (CRUD) operations of each of the entities.For the sample output provided by the Picasso Inventory System please see appendix B.

The Inventory XSL Transformation

We decided to go with a simple transformation to transform our XML data into something readable that could display a report where all the vital information could easily be identified. The report would simplify the inventory process and it would allow the user to pull up a vendor's contact information. Below is a sample of this report.

Figure 5. Sample HTML Report after transformation

Presenting the information in this simple format did not come without its set of challenges. Iterating through the elements was not the difficult part. Amon these challenges were:

- Ability to Display the vendor, unit price, and description
- Alternating colors for rows

The reason these were challenges was that these attributes are stored in the catalog item receives only referenced by the inventory item. Adding to the problem is the dispersal of catalog items across three different paths of the XML tree:

- ProductGraph/Catalog/CatalogItems/CatalogItem
- ProductGraph/Catalog/BoxedItems/BoxedItem
- ProductGraph/Catalog/VolumeItems/VolumeItem

In retrospect, we would have been better off using a schema that wrapped the different catalog types for easier iteration like the list below:

- ProductGraph/Catalog/CatalogItems/CatalogItem/SingleItem
- ProductGraph/Catalog/CatalogItems/CatalogItem/BoxedItem
- ProductGraph/Catalog/CatalogItems/CatalogItem/VolumeItem

This meant that the use of variables and functions was vital in retrieving the correct value. In each of the cases there would always be one path of the tree that determined the value but in all cases it was unknown which one contained the value. To accommodate we grabbed all three values from each of them and then called a function to determine which one had the value.

We wanted alternating colors for our rows. To accommodate that capability we used the position() along with a mod to check for even rows. In addition, we formatted alternating rows with different colors.

The XSL Transformation for This Paper

The Schema

The content contained within this paper resides as XML data in the projectpaper.xml file. In order to fit the content into the XML document we had to come up with a XSD schema that would suit our needs. The schema definition resides in the file name ChicagoStylePaperSchema.xsd. That file contains all of the necessary elements to create a Chicago style term paper. We will point out that the Chicago Manual of Style is several hundred pages and I can safely guess that not even a professional writer will ever use everything found in the Chicago Manual of Style. The book includes everything from how to lay out a book for the printer, to proper documentation measurements. We set out to create a schema that met at a minimum the needs of this paper.

We focused our paper on those elements that are pertinent to a graduate paper. This includes definitions for the elements as seen in Figure 6:
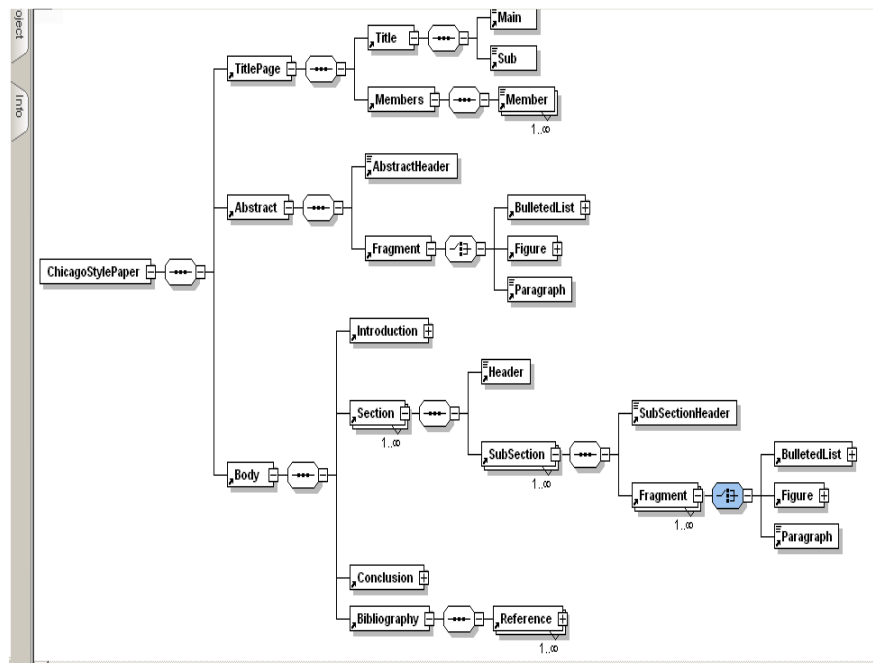
Figure 6. Chicago Style Paper Schema (Full Tree)

The Transformation

Having the schema in such a simplistic state, provided us with the capability to easily transform the contents into an HTML page. You can view the entire contents in appendix section of this paper. Perhaps it was due to all of the previous work that we have done for this class or possibly because the content is more sequential than anything we have previously done, it was rather easy to transform the content into HTML.

In short the transformation consisted of the XLS necessary to transform the XML, the Cascading Style Sheet configuration to properly format the HTML content. We did not stray too far from traditional HTML elements. We avoided HTML attributes largely in part so that we could delegate that responsibility to the CSS code. The CSS code enabled us to perform the following feats.

- Format the title appropriately in upper case

- Format all Level 1 headers

- Format all Level 2 headers

- Format our figures to a uniform size

- Provide the proper spacing between sections

- Properly size the paragraphs

- Properly indent the bulleted lists base on level

- Properly format the bibliography

Summary

The Picasso Inventory is far from being a complete system. It does demonstrate the ability to use XML through all parts of the system to create, update, and read the data effectively. Our sample HTML is nothing but one small sample of use data this data. With some more work, we can do more to display customized queries of the data along with other nice features like pagination, on-demand queries, and full catalog information.

Just like the inventory system, our Chicago Manual of Style schema is far from complete. It does provide a neat way of storing content. We are certain that with some more work and time, this schema can provide all of the required features that would go along with a term paper written in the Chicago Manual of Style format. Because our paper only used a small subset of the guidelines provided by the Chicago Manual of Style, we did not add every possible feature. Among the features missing are:

- Paragraph with mixed content to allow proper formatting of:
  - Italics
  - Bold items

- Other notations such as footnotes

- Level 3 and Level 4 headers

- Footnotes

- Block quotations

- Tables

- Proper indentation for Bibliography references

With a little more time, perhaps a week or two, completion of the implementation of most of those features is possible.

Bibliography

The Chicago Manual of Style, *The Essential Guide for Writers, Editors, and Publishers, 15th Edition* The University of Chicago Press, ***http://www.chicagomanualofstyle.org/home.html***

Mangano, Sal, *XSLT Cookbook,* O'Reilly Media Inc, and 2002, ***http://my.safaribooksonline.com***