

E-Hermes: An Xml Tool for the Classroom

Andreas I. Nicolaou

Patricia A. Essex

Madhavarao Raghunathan

Department of Accounting and MIS

Bowling Green State University

Bowling Green, Ohio 43403

anicol@cba.bgsu.edu, pessex@cba.bgsu.edu, mrarghu@cba.bgsu.edu

Li Liu

Goose Island Beer Company

Chicago, Illinois

peaceliu2000@yahoo.com

ABSTRACT

This paper describes the development of an XML-based tool, called e-Hermes, suitable for use in a variety of instructional contexts. e-Hermes simulates the capabilities of a web-based system to handle customer order transactions and to apply both document structure (data format) validations and application (data content) validations. Students in Information Systems (IS) courses can use e-Hermes to support their learning about web-based database and system design principles, to practice using XML or to understand e-commerce transactions. Assignments are provided in this paper, along with an appendix for instructors who need more background in XML. This paper, therefore, presents important educational contributions in the field of information systems.

Keywords: Classroom Instruction, XML, Electronic Commerce Simulation

1. INTRODUCTION

The advent of modern technology has improved the ability of businesses to transact efficiently in electronic markets. Inter-organizational exchanges in electronic markets may involve transactions between a business and one or more other businesses (business to business e-commerce) with or without the use of an intermediary. The eXtensible Markup Language (XML) is a generic markup language that, by defining document content, can enable systems in remote locations to exchange and interpret business documents over the Internet (Kim 2001; Murray 2002). This ability to generate, send, retrieve, interpret, transform, and process the data in electronic messages automatically is critical to the conduct of web-based electronic commerce (EC) without human intervention. Information systems students need

to learn about the implementation details underlying web-based business activity so that they understand the importance of data validation to the internal control climate of their employer or clients. Students benefit from a study of XML, also, in order to understand how it affects the conduct

of web-based business activity. Additionally, they need to understand the underlying complexity of web-based databases that support concurrent online transactions. Past studies have explored the usefulness of XML as a medium of data exchange (Kim 2001; Murray 2002) and have developed instructional resources for learning XML syntax (Lunsford and Bizarro 2003), but have not examined the issues investigated in this paper.

This paper describes the development of an XML-based EC system, called e-Hermes, suitable for use in the undergraduate information systems class. e-Hermes simulates the capabilities of a web-based system to handle customer order transactions and to apply both document structure (data format) validations and application (data content) validations. The use of this system allows students to visualize and experience online ordering more thoughtfully than they might from personal ordering experiences and still relatively realistically. Suggested assignments are presented along with an appendix to provide instructors with basic XML concepts. The paper concludes with recommendations for further using e-Hermes in

experimental laboratory environments examining issues of continuous system assurance and online event data processing.

2. A MODEL FOR WEB-BASED EC

E-commerce transactions involve a number of separate processes, such as the transmission of data, the translation of exchanged documents using a shared schema, data processing, and updating of a database. Validation processes are critical to the successful implementation of web-based e-commerce and should be applied at all levels. Using XML, for example, a shared data schema can be defined for transacting parties and used to validate the structural format of documents submitted for processing. E-commerce systems also might apply real-time validation in order to ensure that data content included in transmitted documents meets business rules as defined in the firm's relational data model.

Figure 1 presents a general process for web-based EC exchange using XML techniques. Readers unfamiliar with XML might benefit from reading the Appendix A--XML Basics in order to aid with the following discussion. In this

general model, the originator of a transaction will use a schema from the global repository or a customized file to generate an XML instance file for that transaction. After this partner sends its document in XML format over the network, the receiving partner validates the XML instance file against the same XML schema from the global repository or against the customized schema file. These procedures of parsing and validation ensure that the structure of the instance file actually conforms to the schema. The two parties need to agree on the XML schema in advance, and the schema is transmitted with the instance file or, alternatively, the schema's location is defined within the instance file. Validation of the instance file against the XML schema is processed on the server (the receipt) side.

After the file's format is accepted by the recipient, the file contents should be examined to determine the correctness of the data, as defined by the recipient's application rules and standards. For this purpose, the data content from the instance file can be extracted and translated into SQL statements. Then, the validity of the data is checked against the types of elements and relationships that are defined in the recipient's data model. Once validation is completed, the data can be processed to update the recipient's database.

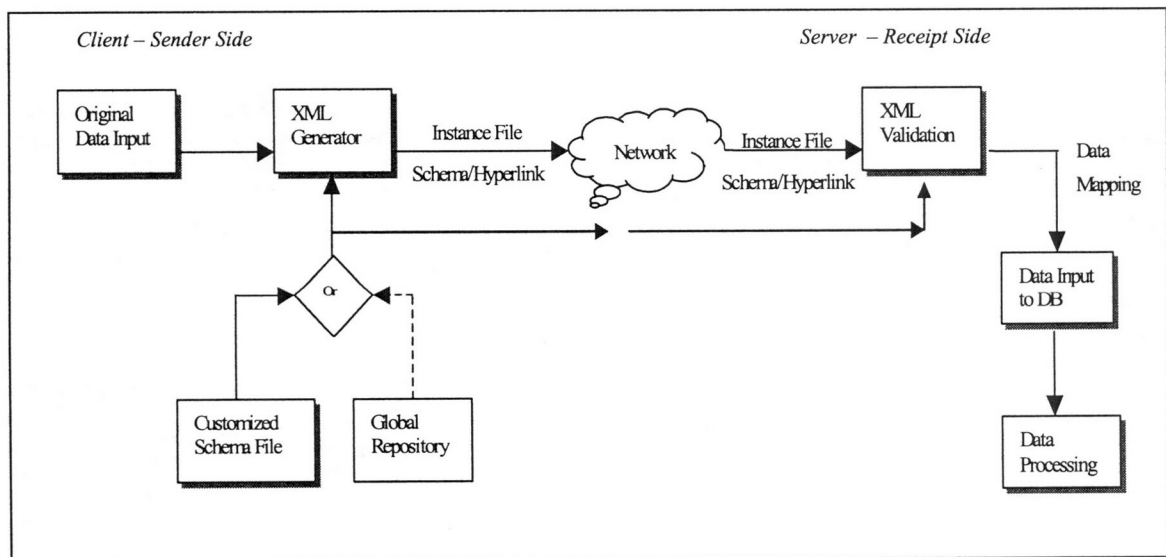


Figure 1: A Generalized Process for XML-based E-commerce

3. THE E-HERMES TOOL

3.1 Description and Purpose

Students at our school are presented with instruction related to web-based e-commerce in a number of accounting and information systems classes, both graduate and undergraduate. Information systems students undertake substantive relational database design projects. The undergraduate accounting students also are involved in systems design projects with a strong emphasis on internal controls, and our graduate students learn how XML can be utilized for business applications. Before e-Hermes was

developed, students learned underlying concepts primarily from lectures and some basic text materials, but had little hands-on experience to bolster that instruction. Thus, they often were struggling to design systems, controls, and applications with little real practice with web-based systems. In order to demonstrate concepts of web-based EC and to give students an opportunity to experiment, an information-exchange model to simulate the processing of such transactions was developed. This tool is called "e-Hermes" in honor of the ancient messenger of commercial transactions, and it can be accessed at:

www.cba.bgsu.edu/faculty_staff/Nicolaou/XMLproject/index.cfm.

XML provides the syntactical framework in e-Hermes for the creation of data structures as well as for their exchange. Thus, the open architecture of the tool incorporates a generic database schema for the business process of sales order processing (ordering from the customer's standpoint) following the generalized model for XML-based e-commerce shown in Figure 1 above. Furthermore, e-Hermes

provides real-time validation both of the format and the content of transaction data, as well as an on-line real-time search function.

3.2 How e-Hermes Works

The basic workflow design of e-Hermes is depicted in Figure 2. Using Figure 2 as the point of reference, each workflow is described (using a numerical reference) to illustrate how the e-Hermes system applies the generalized process shown earlier in Figure 1.

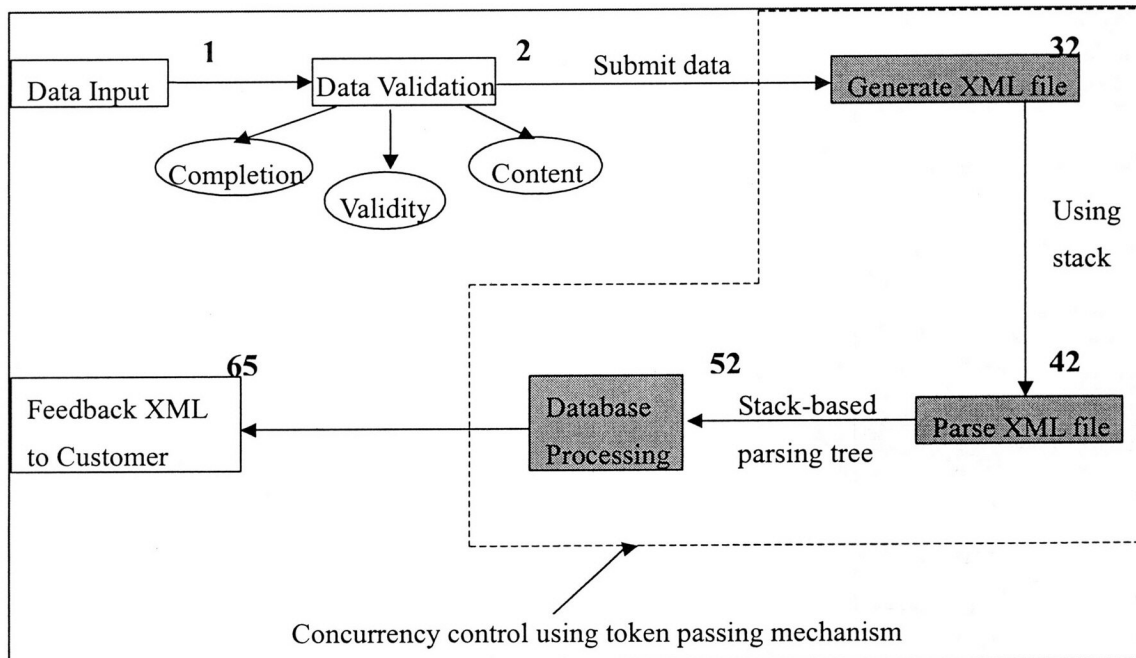


Figure 2: Workflow Design of the e-Hermes System

3.3 Data Input (Workflow 1)

New users register as customers of the simulated e-Hermes vendor. The system will generate and assign to the new user a random customer number and update the customer table in the server database with the customer information provided. To initiate an order transaction, the buyer starts with an online order form on which relevant details are entered. The system validates data entered in the various fields to ensure that they conform to the specified schema and that all required fields are present on the input form before it is submitted.

3.4 Data Validation (Workflow 2)

Once the user "transmits" the order, the transaction can be processed on the server side. e-Hermes validates the format of data entered on the order form as well as checking data content against application rules stored in the system. Table 1 below presents major characteristics of and validations that are carried out by the tool.

3.4.1 Input validation of data completion and validity:

Since e-Hermes was designed to simulate an actual EC environment and the on-line forms are maintained at the server side, the tool's workflows are modified from the

general process shown in Figure 1. In the generalized process, data validations would be executed after the XML instance file is generated by the client and sent to the server side. In the e-Hermes environment, data validation must precede the generation of an XML file. The XML validation step shown in Figure 1, however, is necessary when an XML instance file is transmitted from the client side to the supplier's server. Assuming no data error occurs during network transmission, the simplified approach that is adopted by e-Hermes works well although it does not reflect exactly the workflows observed in a real e-commerce data exchange.

When a new customer order is placed, e-Hermes verifies the correctness and completeness of the data input using a validation routine written in JavaScript to simulate validation against a schema file because e-Hermes is essentially an information exchange model in the context of e-commerce and not a data validation XML program. A corresponding XML schema is illustrated in Table 2. Table 3 presents the JavaScript code used in e-Hermes to validate the telephone number entered by the user. This validation demonstrates how JavaScript simulates validation to a schema, such as the one shown in Table 2 below.

Table 1: Significant Capabilities of e-Hermes

	Task	Examples
Control over input	Format validation	Date input in expected format. Zip code in expected format (5-digit). Phone number in expected format (10 digits). Field types are valid (numeric data where appropriate).
	Input Completeness	All required fields entered on the form.
	Content (application) Validation	Customer Number validated against customer table. Item Number validated against inventory table. Quantity ordered checked against inventory level.
Control over processing	Confirmation messages	Order transmittal occurred. Order acceptance by vendor.
Other	New customer	Registration process.
	Search	Check and report on status of a particular order.

Table 2: An XML Schema for e-Hermes

<pre> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <xsd:annotation> <xsd:documentation> This XML schema defines an order schema for demonstration. </xsd:documentation> </xsd:annotation> <xsd:simpleType name="CodeNum"> <xsd:restriction base="xsd:string"> <xsd:pattern value="\d{6}" /> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="zipCode"> <xsd:restriction base="xsd:string"> <xsd:pattern value="\d{5}([-]\d{4})?"/> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="phoneCode"> <xsd:restriction base="xsd:string"> <xsd:pattern value="\s*((\d{3}) (\d{3}[-]))?\d{3}[-]?\d{4}\s"/> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="emailCode"> <xsd:restriction base="xsd:string"> <xsd:pattern value="\w+\W*\w*@{1}\w+\W*\w+.\w+.*\w"/> </xsd:restriction> </xsd:simpleType> <xsd:element name="customerOrder"> <xsd:complexType> <xsd:sequence> <xsd:element name="orderNumber" type="CodeNum"/> <xsd:element name="orderDate" type="xsd:date"/> <xsd:element name="custNumber" type="CodeNum"/> <xsd:element name="custName" type="xsd:string"/> <xsd:element name="street" type="xsd:string"/> <xsd:element name="city" type="xsd:string"/> <xsd:element name="state" type="xsd:string"/> <xsd:element name="zip" type="zipCode"/> <xsd:element name="phone" type="phoneCode"/> <xsd:element name="email" type="emailCode"/> <xsd:element name="comments" type="xsd:string"/> <xsd:element name="item" type="orderItem" maxOccurs="unbounded"/> <xsd:element name="total" type="xsd:decimal"/> </xsd:sequence> </xsd:complexType> </xsd:element> </xsd:schema> </pre>

```

</xsd:element>
<xsd:complexType name="orderItem">
  <xsd:sequence>
    <xsd:element name="itemName" type="xsd:string"/>
    <xsd:element name="itemNumber" type="CodeNum"/>
    <xsd:element name="description" type="xsd:string"/>
    <xsd:element name="quantity" type="xsd:integer"/>
    <xsd:element name="cost" type="xsd:decimal"/>
    <xsd:element name="subtotal" type="xsd:decimal"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

Table 3: A Sample JavaScript Validation

```

//Some Format Validations carried out by the e-Hermes System (in JavaScript)
//A function receiving a string and an element object to validate phone#
function isPhone(elementObj,str){
  var regexp=/^\s*((\d{3})|(\d{3}))[- ]?\d{3}[- ]?\d{4}\s*$/;
  if(!regexp.test(str))
  {
    var msg="Please enter phone number in the following format: "; msg+="\nLong distance:
(999)-999-9999, 999-999-9999,9999999999, 999 999 9999,etc";
    window.alert(msg);
    //clear user's wrong input
    elementObj.value="";
    elementObj.focus();
    return false msg+="\nLocal call: 999-9999, 9999999, or 999 9999";
  }
  return true;
}

```

3.4.2 Validation of data content: After all format validations are accepted, e-Hermes carries out validations of data content. It checks that the customer number and the order item numbers exist in the relevant database tables. Then, e-Hermes checks the database to ensure that the quantity ordered can be satisfied by the vendor's system. The system does not accommodate backorders at this time; as a result, quantity ordered of any one item is not allowed to exceed the quantity of that item available. To achieve these content validations, SQL statements are used to run a number of queries through the database. Such SQL statements are created by e-Hermes and are transparent to customers.

3.5 XML File Generation (Workflow 3)

After all format and content validations have cleansed the data, an XML file is generated at the server side and feedback is transmitted to the customer to report on the acceptance of the order. An example of an instance file that would result from this process is shown in Table 4.

3.6 XML File Parsing (Workflow 4)

A logical data structure implemented by a Document Object Model (DOM) tree subsequently parses the validated XML instance file. Table 5 above illustrates the data structure of the parser used to process the validated XML instance file.

3.7 Data Processing (Workflow 5)

Following data parsing, the flat file that is generated by the XML file parser is inserted into the simulated vendor's database tables using SQL statements. The relationships among the data at the vendor side are shown in Figure 3.

Another aspect of the workflow design of e-Hermes is that it can provide hands-on experience on the importance of concurrency control in database processing and how it works. Since multiple users could submit requests to the system simultaneously, mutual exclusion must be enforced in order to ensure data integrity for the whole database. Concurrency control is implemented using a token-based daemon, which runs in the background. When a user submits the on-line form, the submission process requests the token from the daemon. If no other process is accessing the database, the daemon will offer the token to the requesting process thus locking out other users. After the transaction is completed, the process will release the token back to the daemon.

3.8 XML Feedback to the Customer (Workflow 6)

As mentioned above, after validation is completed, the customer is provided with an online message about the acceptance of the order. The customer is also provided with an electronic copy of the XML instance file, which includes all submitted data and a system-generated order number for future reference. The customer can return to e-Hermes and

Table 4: XML Instance Example

```
<?xml version="1.0"?>
<customerOrder xmlns:mySchema="order.xsd">
  <orderNumber>000003</orderNumber>
  <orderDate>2004-03-15</orderDate>
  <custNumber>050871</custNumber>
  <custName>House Tools</custName>
  <street>251 North Grove Street</street>
  <city>Albany</city>
  <state>NY</state>
  <zip>62356</zip>
  <phone>423-874-4521</phone>
  <email>tony@housetools.com</email>
  <comments>Please ship the order ASAP.</comments>
  <item>
    <itemName>Copy Paper</itemName>
    <itemNumber>001056</itemNumber>
    <description>Letter HP</description>
    <quantity>23</quantity>
    <cost>80.99</cost>
    <subtotal>1862.77</subtotal>
  </item>
  <item>
    <itemName>Floppy disk</itemName>
    <itemNumber>002500</itemNumber>
    <description>Imation IBM 2HD</description>
    <quantity>100</quantity>
    <cost>4</cost>
    <subtotal>400.00</subtotal>
  </item>
  <total>2262.77</total>
</customerOrder>
```

Table 5: Data Structure of the Parser

```
class element
{
  var $total_child = 0;           // If non_leaf_node, record number of children
  var $leaf       = 0;           // Token for leaf_node
  var $name;                 // Name of XML Tag
  var $value      = "";         // If leaf_node, record its value
}
```

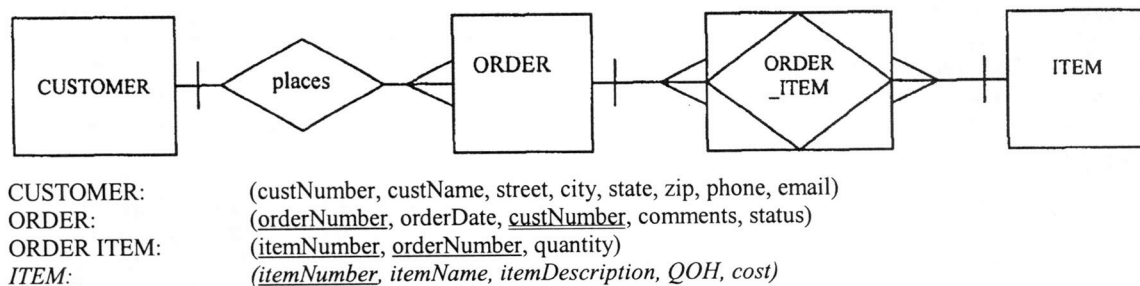


Figure 3: Underlying Database Structure

search for the status of orders by entering the customer number. The system will respond with one of three codes: 'A' if order is accepted, 'P' if order is in-progress, or 'S' if order has been shipped.

4. IDEAS FOR INSTRUCTIONAL USE

Because of its open architecture, the e-Hermes tool is adaptable for use by information systems classes in a number of situations. For classes that already are covering the design and implementation of databases, e-Hermes can be used to supplement and enhance that discussion without much new material and with little additional class time investment. Alternatively, with background instruction in web-based EC, e-Hermes can help the instructor focus student attention on procedural and security issues. This business and internal control focus is likely to require one or two class periods, depending upon how much depth is demanded by the chosen assignment(s). Finally, e-Hermes can be used to discuss XML coding, with assignments to improve, change, or expand it. Appendix A provides background details to support instructors who wish to use e-Hermes for XML instruction. Using e-Hermes as a vehicle to incorporate XML into the classroom could involve a significant time investment if XML is not already part of the curriculum.

In general, e-Hermes is a very flexible tool for educational use. It provides a capability for the analytic study of a system by in-depth examination of the individual parts. e-Hermes can be studied and used as a living B2B system, but it can also be decomposed to see how the programming interfaces with the database structures that exist on the server side of the transaction. A listing of a number of instructional usage suggestions is provided in Table 6, along with learning objectives and required background knowledge. The authors have used a number of these assignments (marked with an asterisk), and sample assignments are described in the next section.

5. SAMPLE ASSIGNMENTS USING E-HERMES COMPONENTS

The following section describes assignments that can be supported by one or more components of the e-Hermes system. For each assignment, information about the methods used and the preparation required is provided. Feedback was gathered from students about the assignments. On a nearly unanimous basis, the students reported that the assignments were helpful to their understanding of concepts and the application of these concepts to related projects. Some students offered suggestions, and where these seemed to be improvements, they have been incorporated into the revised assignments.

5.1 Instructional Focus: Database Design

The components of e-Hermes can broaden the database experience for IS students in a number of ways. Our undergraduate students in accounting are required to take a systems course in which they must develop an extensive Access database, based upon an entity-relationship (E/R)

diagram which they must create. The students benefit greatly from additional exposure to E/R diagrams and how these are interpreted in Microsoft Access, which is also used by e-Hermes. An assignment used to reinforce these concepts is shown in Figure 4 and is described in the following paragraphs.

5.1.1 Preparation: Before students are ready for this assignment, classroom discussion will have covered the process of extracting business entities from a scenario, determining which entities should be linked, formation of tables for these entities, identification of primary and foreign key fields, and how to accomplish the linking of related entities based upon the structural constraints (optionality and cardinality) of the relationship. Small exercises will have been completed to strengthen student skill in one or more of these items. Thus, students are ready to "work backwards" to build a small E/R diagram.

Teachers using this exercise will want to download the e-Hermes database from the e-Hermes website. Because this database is in frequent use by a number of classes and researchers, the data will change from time to time. Thus, before making the Figure 4 assignment, teachers will want to examine the data and be sure that it meets the needs of the assignment unless the teacher wishes to have students identify their assumptions. (If trying to eliminate assumptions, the data should show that at least one customer has never ordered, at least one product has never been ordered, some products have been ordered more than once, some customers have ordered more than once, and some orders include more than one item.) Also, upon downloading the database, teachers should remove the relationships from it so that students will have only tables with which to work.

5.1.2 Using e-Hermes: This assignment could be a homework assignment, with students acquiring their own copies of the e-Hermes database and working directly in Access to draw the relationships. In that case, the question about primary keys is a moot point as they can look at the tables. Alternatively, this assignment lends itself easily to an in-class, paper-and-pencil exercise. Students receive the directions, the Access Relationship page from the e-Hermes database that contains the de-linked tables, and a printout of the data contained in the five tables. Then, students can work in pairs to complete the assignment. Answers are reported to the class, and discussion of the results follows immediately.

5.1.3 Actual use: This project was assigned to undergraduate classes using database systems, both in accounting and IS. Entity relationship (ER) modeling was discussed in these classes prior to this assignment. The majority of students drew ER diagrams with accuracy levels of 90% or higher. The students were able to identify primary and foreign keys and both maximum and minimum cardinalities. The students wrote good critiques on the e-Hermes database, highlighting its strengths and weaknesses.

Table 6: Summary of Suggestions for Use of e-Hermes as an Instructional Tool

Focus	Assignment	Learning Objectives	Background Knowledge Needed
Database Design	Examine the e-Hermes system; draw the corresponding E/R diagram and outline the existing table structure.*	To identify cardinalities. To identify tables representing entities vs. those representing relationships.	Basics of database design, including E/R diagramming.
	Write SQL queries to provide management with useful information about customers and their ordering behavior, as captured by e-Hermes.	To practice SQL skills.	SQL procedures.
	Write SQL queries to provide management with useful information about other business processes (beyond e-Hermes data).	To identify managerial information needs.	General knowledge about web-based EC applications and the types of decisions faced by management.
	For each query developed, identify which data needed is already in the database and which data must be added to the database, including where it should be stored.	To analyze an information need and to reduce that need to the basic input data and processes required to produce the information.	
e-Commerce and Controls	Simulate buying behavior.*	To understand the notion of B2B commerce.	General knowledge about web-based EC applications.
	Discover as many validations built into the system as possible. Analyze whether these validations are appropriate, are in the correct location, and will work as intended. Classify each control as preventive, detective or corrective.*	To critique the efficiency and effectiveness of an existing internal control system.	Basics of internal control, particularly application controls. General knowledge about web-based EC applications.
	List the risks that are associated with a web-based EC purchasing application from the seller's point of view and from the buyer's point of view.	To consider the types of risks that can be minimized with good application controls.	Basics of internal control, in particular application controls.
	Consider the internal control objectives of input validity, input accuracy, and input completeness. Discuss whether reasonable assurance has been attained with respect to input errors in e-Hermes.*	To practice spotting internal control oversights.	General knowledge about web-based EC applications.
XML	Create an XML instance document for the customer order that matches the schema simulated by e-Hermes. When students are provided the XML schema for order generation as shown in Table 2, their responses in XML schema would appear similar to the document in Table 4. *	To understand clearly the data-tagging notion that underlies XML.	General knowledge about web-based EC applications. Basic knowledge of XML schema and instance documents.
	Adapt the simulated e-Hermes XML code, as provided in Table 2, so that additional validations can be handled by the system.	To understand how XML schema enforces certain kinds of restrictions and allows flexibility in other cases.	Knowledge of XML syntax.
	Adapt the simulated e-Hermes XML code for a more complex change, for example, so that the customer could choose freight method from a given set of alternatives, or so that the vendor might offer discount pricing.	To be able to write XML schema and XML instances.	More advanced knowledge of XML syntax.
	Write XML code for another business cycle (for example, cash receipts).	To understand clearly the data-tagging notion that underlies XML. To be able to write XML schema and XML instances.	General knowledge about web-based EC applications and business processes. Knowledge of XML.

e-Hermes Project 1--Database Design

Learning Objectives:

- \$ To identify primary and foreign keys and the relationships among tables.
- \$ To identify tables that represent entities and those that represent relationships.

Directions:

- a. Examine the e-Hermes database, particularly the Access Relationships page. Identify key fields for each table, underlining the primary keys and placing an asterisk beside any foreign keys. Note that the diagram has tables but no relationships because they were removed for this exercise. Draw the relationships you would expect to find on the Relationships page.
- b. Draw the corresponding E/R diagram complete with as many structural constraints (optionality and cardinality) as you can identify. If you cannot complete a constraint from the Access Relationship page and your business knowledge, look at the printouts of the tables and the data in each. This data should resolve your assumptions.
- c. Evaluate this database. What are some of the good things that you can see about it? What needs improvement? Why?

Figure 4: A Sample Assignment related to Database Design

5.2 Instructional Focus: Electronic Commerce and Controls

Students can be given the opportunity to simulate buying behavior using e-Hermes. In our classes, we are concerned particularly about the design of input forms and controls over input. Figures 5 and 6 illustrate assignments that have been used.

5.2.1 Preparation: The instructor should discuss the process of web-based electronic commerce, in particular focusing on associated business risks and the types of controls that might be put in place to mitigate the risks. e-Hermes can be introduced as an example of the B2B ordering process. Before using the following exercises, students also should be introduced to the control objectives of input validity, input accuracy, and input completeness as well as the textbook definitions of a variety of application controls, such as field test, range test, valid combinations, and so on.

5.2.2 Using e-Hermes: An effective instructional procedure is to use the assignment in Figure 5 as the basis for in-class paired-partner work followed by discussion. Begin by showing the Customer Order form, either live or as an overhead, and provide an example of field, possible error, and effective control. For instance, for the Customer Number field, the error might be that the customer entered the wrong number, and one control would be a closed loop validation. Then, assign each pair two fields from the form, asking them to suggest an answer for each field. When the pairs are done (or after a few minutes), they can share their findings with the class with the teacher commenting as needed.

Following up on this exercise, the assignment in Figure 6 is given as homework. Students are to use e-Hermes and extend the controls discussion. Appendix B presents the

instructions that students use to navigate e-Hermes. Students seem to have no problems accessing and using the system without assistance.

After the assignment is due, class time is well spent in a discussion of validations that students noted as working or omitted. It is particularly interesting to hear students report their discovery that e-Hermes does not always verify data at the most convenient time for a user. Discussion should include questions such as: Where (in the event process) do particular validations belong in order to be most effective in preventing or detecting errors? What would the student change, if anything?

5.2.3 Actual use: In the undergraduate accounting and IS classes where we use this assignment, a major focus is database design. Students are required to self-learn database implementation using MS Access (beyond the minimal Access they have learned in a prerequisite class). An important requirement of student projects is the creation of multi-table data entry forms. This assignment provided our students with an opportunity to test a web-based data entry form. Students were able to understand the significant features of the order entry screen and to identify its shortcomings. They reported that the required critique of the e-Hermes system helped them achieve better forms design in their own projects.

5.3 Instructional Focus: XML

e-Hermes is a tool that allows students to examine the information processes involved in electronic commerce using XML, including the processes that occur outside of the system itself, as well as the processes that occur within the data exchange. The generalized process for XML-based EC can provide a comprehensive basis for the presentation of e-Hermes's capabilities. However, because the validation

e-Hermes Project 2--Internal Control

Learning Objectives:

\$ To evaluate the risks associated with an input document.

\$ To determine appropriate controls that might be used to minimize these input risks.

Directions:

Examine the e-Hermes customer order screen and consider where input might be entered incorrectly. For each field, state at least one error that could occur if the user were entering data directly into this online form. Then, suggest a corresponding control that could prevent or detect that error.

Field	Error	Control

Figure 5: A Sample In-Class Assignment Related to Internal Control

process has been simulated using JavaScript, current instructional capabilities must be based upon the schema and instance document presented in Tables 2 and 4, respectively, earlier in this paper. Figure 7 presents an assignment for use in classes that involve XML.

5.3.1 Preparation: While our undergraduate classes do not study XML, it is a topic in one of the graduate classes. Before the assignment is made, students have learned general information about XML and looked at some instance documents and schema. Also, the syntax of XML has been discussed, but students have not done any coding on their own. Preparation on the part of the instructor is to be relatively well versed in the basic XML syntax, which can be accomplished with one of the sources listed in Appendix A.

5.3.2 Using e-Hermes: With this homework assignment, the e-Hermes simulated XML instance document and schema are both provided to students along with the order screen. Students explore the code to learn how XML schemas are written and used, noting how the language they've already studied has been applied. As with our undergraduate class, however, the focus remains on controls and validations rather than the general use of XML. Students are required to state the specific changes that would be made to the given XML code if the system were to be validate the data in particular, relatively minor ways. This type of assignment could also include more complex changes, for example, updating the e-Hermes system to handle freight method choices, vendor backorders, or discount pricing.

5.3.3 Actual use: This project was used in both the undergraduate accounting and information systems classes, with some modifications to meet the needs of each class. As a part of their Access projects, students are expected to understand how data validation enforces business rules utilizing table and form properties. Students in both classes successfully identified:

- Existing data validations,
- Deficiencies in data validation, and
- Suggestions to augment data validation.

IS courses that take a more sophisticated and in-depth look at XML are likely to find e-Hermes a good vehicle for more

extensive projects. The e-Hermes system as it currently exists is limited to just one part of the B2B revenue cycle (for the vendor). Thus, e-Hermes is inherently a modular system. Students could be asked to work in teams to develop new modules—perhaps related to the cash collections cycle—to broaden the current focus on the customer's placing an order. Alternatively, student teams could increase the scope of the system to handle more extensive inventories and more complicated customer transactions.

Currently the system adopts the XML W3C 1.0 standard. In case the W3C updates this XML standard to a different version, it is expected that the existing 1.0 standard would be compatible with any new versions. In addition, the software modules included in the e-Hermes system are flexible and can be modified easily by instructors in order to incorporate any new XML standard. Moreover, instructors can provide assignments that could involve the identification of differences between the XML 1.0 and the newest standard and encourage students to modify the source code in the system itself.

5.4 Survey Outcomes

The first students to use a version of the database-oriented assignments were 32 undergraduate students in an accounting information systems class. These students were asked to respond to an open-ended survey designed to evaluate the assignments shown above in Figures 4, 5, and 6 and to solicit suggestions for future use. Table 7 below summarizes the responses. (Numbers do not add to 32 because two students did not do the survey, and occasionally a student did not answer one of the questions.)

These assignments were modified, and then students in MIS database classes were asked to respond to a short survey at the end of the three assignments. Eighteen students both completed the assignments and responded to the survey.

This survey asked the students to rank the utility of the three assignments (taken as a whole) on their ability to attain the following learning objectives:

- To identify primary keys and foreign keys in a relational database, and to identify relationships among the tables.

- B. To evaluate the design of an online data entry form.
- C. To evaluate the design of a database.
- D. To evaluate the presence and absence of adequate data validation rules.

down to 0 for "very poor". Student responses to individual learning objectives are summarized in Table 8. Overall the MIS students rated the assignments to be at least somewhat helpful. Interestingly, the assignments were found to be most helpful in evaluating the design of an online data entry form and in evaluating data validation rules, which was our main purpose in using the assignments.

Assessments were reported on a scale of 5 for "very good"

e-Hermes Project 3--Internal Control

Learning Objectives:

\$ To critique an existing system with respect to its control efficiency and effectiveness.

\$ To practice spotting internal control oversights.

Directions:

Following the provided directions, use the e-Hermes system to enter the two specified transactions with at least three more of your choosing. Whenever you see that the system is validating your entry, make a note of that validation rule. Whenever you find that the system did not implement a control that you believe should have been present, make a note of that, too.

For at least four rules that you have spotted as present, answer the following questions, placing your answers in the table below.

What type of application control is the validation rule that you spotted? (That is, is the control a field test, range test, length test, validity test, valid combinations test, closed loop verification, completeness test, or system-generated data?)

Is the control handled in the most logical way and in the most efficient spot in the ordering process? If not, what would be an improvement?

Is the timing of the rule that you found considered preventive, detective or corrective?

Is the objective of this control input validity, input accuracy, or input completeness (or a combination)?

	Rule that you found to be <u>present</u>	Type of validation	Most logical location? (If not, why not?)	Timing			Objective		
				P	D	C	IV	IA	IC
Ex.	Order quantity is checked against the quantity on hand in the database.	Range test	Yes, seems efficient and effective	x				x	
1									
2									
4									

Find and describe below at least three controls that seem to be absent and that you believe are advisable.

	Rule that you found to be <u>absent</u>	Type of validation	Timing			Objective		
			P	D	C	IV	IA	IC
1								
2								
3								

Figure 6: A Sample Hands-on Assignment Related to Internal Control

e-Hermes Project 4: XML and Internal Control

Learning Objectives:

- \$ To emphasize the relationship between an XML document and its related schema.
- \$ To practice coding XML Schema so that validation rules enhance control over input.
- \$ To consider the usage of various computerized application controls in the Order context.

Directions:

On the following pages, you will find printed copies of an order screen, an XML Schema, and an XML document that uses that schema. These items relate to a system called e-Hermes which simulates business-to-business e-commerce activity. Familiarize yourself with the order screen and how it relates to the XML document. Then, examine the schema and satisfy yourself that it will work with the XML document to capture the order data.

1. What are some good things about the coding that you have examined? Why?
2. What are some things about the coding or system design that seem to need improvement? Why?
3. Extend or improve the XML schema coding so that it contains more validations of the data that a user of the system will enter. In particular, revise the code so that the following validation checks could occur:
 - a. Quantity of item ordered must be greater than zero.
 - b. State must be entered as a standard (upper-case) two-digit abbreviation.
 - c. Suppose that the customer coding system is changed such that the customer number must begin with C and be followed by exactly five numeric digits.
 - d. The number of items ordered is not limited.
 - e. Customer phone number must be present.

Figure 7: A Sample Homework Assignment Related to XML

Table 7: Accounting Students' Evaluation of e-Hermes Database Assignments

	Yes	Somewhat	No
Was Assignment 1 (see Figure 4) helpful to you for improving your understanding of course material?	24	2	4
Was Assignment 1 (see Figure 4) helpful to you for understanding how to apply course material to your group project?	24	2	1
Was Assignment 2 (see Figure 5) helpful to you for improving your understanding of course material?	25	1	4
Was Assignment 2 (see Figure 5) helpful to you for understanding how to apply course material to your group project?	24	3	3
Was Assignment 3 (see Figure 6) helpful to you for improving your understanding of course material?	28	1	0
Was Assignment 3 (see Figure 6) helpful to you for understanding how to apply course material to your group project?	26	0	3

Table 8: MIS Students' Evaluation of e-Hermes Database Assignments

Learning Objective	Mean Rating	Median Rating
A	3.38	3.5
B	3.47	4.0
C	3.08	3.0
D	3.72	4.0

6. CONCLUSION

Both IS instructors and researchers often find great difficulty gaining access to application systems of large business organizations that could have developed such systems. The simulated tool described in this paper can provide the necessary flexibility to demonstrate important concepts in the classroom and the functionality to simulate a real business process, as it might occur in an EC web-based environment. We have used the tool with both undergraduate and graduate students and in both accounting and information systems courses. Results of surveys, typically conducted after a series of assignments, have provided nearly unanimous support for the use of the tool in the classroom. As students recommend useful changes, they have been incorporated in the assignments reported here.

The high flexibility of the tool also permits its use for research purposes. The tool can be customized in order to implement a variety of tasks in an experimental laboratory setting. For example, the tool can be adapted in order to manipulate the provision of different levels of assurance in an online exchange environment. Furthermore, it can be adapted to emulate event processing in open web-based exchanges versus mediated-data exchanges that often exist in business exchanges that occur through electronic data interchanges assisted by value-added network providers. Such applications of e-Hermes, for example, are reported in Nicolaou [2004] and Nicolaou, Liu and Lord [2003]. These uses of e-Hermes and its adapted laboratory versions allow subjects to comprehend abstract concepts about differences between web-based (open) and mediated electronic commerce exchanges, which may be difficult to grasp due to the lack of access to real-world applications.

7. ACKNOWLEDGEMENTS

This manuscript has benefited from valuable comments from conference participants at the 2003 Mid-Year Conference of the Information Systems Section of the American Accounting Association.

8. ENDNOTES

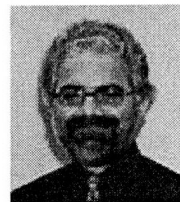
eHermes tool is available at: http://www.cba.bgsu.edu/faculty_staff/Nicolaou/E-Hermes_Zip/XMLProject.zip

9. REFERENCES

- Armstrong, Eric. (2001). *Code Fast, Run Fast with XML Data Binding*. Available Online: <http://java.sun.com/xml/jaxp/dist/1.0.1/docs/binding/DataBinding.html>
- Goldfarb, C.F. and P. Prescod (2002) *Charles F. Goldfarb's XML HandbookTM*, 4th edition, Upper Saddle River, NJ: Prentice-Hall PTR.
- Kim, H. (2001) "An XML-based modeling language for the open interchange of decision models", *Decision Support Systems*, 31, 429-441.
- Kobielus, J.G. (2001) *BizTalk: Implementing Business-to-Business E-Commerce*, Upper Saddle River, NJ: Prentice Hall PTR.
- Mercator (2001) *Simplifying E-Business Integration for SAP Customer*, available at: <http://www.mercator.com/products/sap.html>
- Lunsford, D.L. and P. Bizarro (2003) "XML alphabet soup", *Information Systems Section of the American Accounting Association: Compendium of Classroom Cases and Tools for AIS Applications*, available at: <http://aaa-is.byu.edu/Publications/C3/title.htm>
- Murray, M (2002), "Using the extensible markup language (XML) as a medium for data exchange", *Communications of the Association for Information Systems*, 9, 105-119.
- Nicolaou, A.I. (2004). "Information Quality in Data Exchanges: Effects of Risk, Trust and Performance", working paper, Bowling Green State University.
- Nicolaou, A.I., L. Liu, and A.T. Lord (2003), "Demand for data assurances in electronic commerce: An experimental examination of a web-based data exchange using XML" In S. J. Roohani, Editor, *Trust and Data Assurances in Capital Markets: The Role of Technology Solutions*. PriceWaterhouseCoopers LLP Research Monograph, pp. 32-42.
- Peregrine (2001) *TrustLink TM: AS/400 Edition*, available at <http://www.peregrine.com>
- S. de la Fe, Jr., C. Hoffman, and E. Huh (2000) "XBRL Taxonomy: Financial Reporting for Commercial and Industrial Companies, US GAAP", available at <http://www.xbrl.org/us/gaap/ci/2000-07-31/us-gaap-ci-2000-07-31.doc>
- W3C (2004) *W3C Technical Reports and Publications - XML Word Wide Web Consortium*. Available at: <http://www.w3.org/TR/2004/REC-xml-20040204>
- W3C (2001) *XML Schema Part 0: Primer*. Available at: <http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/>
- W3C (2001) *Document Object Model Activity Statement*. Available at: www.w3.org/DOM/Activity
- XML Conference (2002) *About XML 2002*. Available at: <http://www.xmlconference.org/xmlusa/2002/about.asp>

AUTHOR BIOGRAPHIES

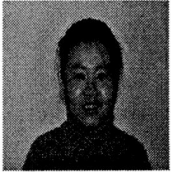
Andreas I. Nicolaou is Associate Professor of Accounting at Bowling Green State University. He received his B.A. in Business Administration from the Athens University of Economics and Business, and his M.Acc. and Ph.D. in Accounting from Southern Illinois University-Carbondale. He is a CPA (non-practicing). His research has appeared in such journals as the *International Journal of Accounting Information Systems*, *Journal of Information Systems*, *European Accounting Review*, *Advances in Management Accounting*, among others. He has also published two textbooks and has contributed in two research monographs on trust and data assurance and on business intelligence and accounting.



Patricia A. Essex is an Associate Professor in the Department of Accounting and MIS in the College of Business Administration at Bowling Green State University. She was awarded her Ph.D. in accounting from Michigan State University in 1993 and has been a CPA since 1986. Her articles have been published in a variety of journals, including *Advances in Accounting Information Systems*, *Ohio CPA Journal*, and *Journal of Management Information Systems*.



Li Liu received the B.S. degree in International Economics and Trade from Nankai University, P.R.China in 1996 and the M.S. degree in Accounting from Bowling Green State University, OH in 2002. She is a Certified Public Accountant in Illinois and currently works as an accountant in Goose Island Beer Company at Chicago.



Madhavarao Raghunathan is an Assistant Professor in the Department of Accounting and MIS in the College of Business Administration at Bowling Green State University. He obtained his Ph.D. in Management Information Systems from Kent State University in 1999. He has published in the *International Journal of Electronic Commerce*.



APPENDIX A: XML BACKGROUND

In the business world, appreciation for the ability of XML to customize tags that describe the data structure of files continues to grow. Many software vendors have incorporated XML into their web-based EC offerings, including Microsoft's BizTalk server [Kobiellus 2001], Peregrine's TrustedLink [Peregrine 2001] and Mercator's E-Business Broker Suite [Mercator 2001]. Such programs serve an intermediary role by translating data from X.12, EDIFACT, and flat files, for example, into XML format using data parsers and by helping to integrate the translated data with the organization's internal information systems. Thus, XML offers small businesses an ability to transact with larger companies, those that were early implementers of EDI technology, without installing traditional EDI systems.

XML is not so much a single standard as it is a set of specifications centered on the core standard, XML 1.0, which is an "official recommendation" of the World Wide Web Consortium [W3C 2004]. Currently many organizations are facilitating the use of XML in business exchanges and business reporting. OASIS and UN/CEFACT are organizations that are building industry consensus for a technical framework called ebXML to enable consistent use of XML for the exchange of business data [Goldfarb and Prescod 2002]. XBRL (eXtensible Business Reporting Language), based on and an extension of XML, has been developed for the financial reporting domain. According to the IDG Research Service Group, nearly 80% of organizations surveyed are using XML currently or planning to use XML in their business activities soon [XML Conference 2002].

Terminology

The metadata (that is, data to describe a particular set of data) of the XML file is defined by either XML Schema or Document Type Definition (DTD) code. While the terms schema and DTD are often used interchangeably, they do have specific technical meanings. *Schema* is a generic term for a document/data structure with a prescribed set of rules. A DTD is one method for coding a schema; XML Schema is another. In this paper, the term "schema" will be used whenever a user might choose to employ either a DTD or an XML Schema. The schema, thus, defines the allowed data structures and data types in a document and includes required constraints on each data element. In a customer order document, for example, the schema would define the data elements to be included in the document (such as customer information, delivery information and ordered items) as well as any required validations that the data must meet (for example, a numeric format or specified length). To illustrate, consider the following customer data:

Customer Code	Customer Name—English	Customer Name—Spanish	Customer Name—Greek	Phone 1	Phone 2	Average Purchase	Reference Name	Reference Phone
B110	The Bull	El Toro	Taurus	215-000-1122	215-000-1123	\$800	-----	-----
B112	Bell Library	-----	-----	608-782-5191	000-000-0000	\$320	Susan White	608-995-1618

An XML *instance document* is one that contains data tagged with descriptors defined by its related schema. Table 9 provides an example of a short DTD and an instance file that, together, encode the data presented above.

While XML bears a strong resemblance to the markup of HTML (and both are derived from Standard Generalized Markup Language), they differ in purpose. In general, XML describes data content while HTML focuses on data presentation. In this example, the XML instance file can be understood much more easily due to its use of customized tags while the HTML file necessarily uses standardized tags to present information with repeated instances of blank fields. For the sake of comparison, Table 10 demonstrates the HTML code corresponding to the XML code included in Table 9.

XML documents can be manipulated to suit the needs of users. One or more different "extensible stylesheets" (XSL) can be applied to an XML document in order to present various information subsets or to manipulate content. In the example above, for instance, a user might want to examine the average purchases made by all customers who were referred by the same person. By applying a stylesheet to the XML document, this type of view can be easily derived. Users can produce nearly any view of the document content that they wish, thus allowing data re-use. XML supports and promotes these types of derivative processes by being flexible and modular.

A schema in XML is similar to a blueprint; it should be used to check whether the contents of a document (the instance file) follow predetermined rules in terms of required document structure (established in the schema). This schema is transmitted from one trading partner to the other along with the instance file; or, alternatively, the location of the schema can be specified in the instance file. In addition, an XML document can declare one or more public namespaces, in the form of unique URLs, as a mechanism for resolving XML tag and attribute name conflicts in the DTDs and schemas utilized by different document originators and recipients. For example, XBRL documents can cite the already-developed taxonomy for the creation of XML-based instance documents for business and financial reporting of commercial and industrial companies according to US

Table 9: A Simple DTD and Related Instance File

Document Type Definition	Instance File
<pre> <!ELEMENT customers (customer*)> <!ELEMENT customer (cname+, cphone*, average_purchase, reference?)> <!ATTLIST customer code ID #REQUIRED> <!ELEMENT cname (#PCDATA)> <!ATTLIST cname language #REQUIRED> <!ELEMENT cphone (#PCDATA)> <!ELEMENT average_purchase (#PCDATA)> <!ELEMENT reference (rname, rphone*)> <!ELEMENT rname (#PCDATA)> <!ELEMENT rphone (#PCDATA)> </pre>	<pre> <?xml version="1.0" ?> <customers> <customer code="B110"> <cname language="English">The Bull</cname> <cname language="Spanish">El Toro</cname> <cname language="Greek">Taurus</cname> <cphone>215-000-1122</cphone> <cphone>215-000-1123</cphone> <average_purchase>\$800</average_purchase> </customer> <customer code="B112"> <cname language="English">Bell Library</cname> <cphone>608-782-5191</cphone> <average_purchase>\$320</average_purchase> <reference> <rname>Susan White</rname> <rphone>608-995-1618</rphone> </reference> </customer> </customers> </pre>

Table 10: HTML Code of XML Instance Document

```

<HTML>
<HEAD>
<TITLE>XML to HTML Example</TITLE>
</HEAD>
<BODY BGCOLOR="#CCCCCC">
<table>
<tr>
<td> Customer Code </td>
<td> Customer Name - English </td>
<td> Customer Name - Spanish </td>
<td> Customer Name - Greek </td>
<td> Phone 1 </td>
<td> Phone 2 </td>
<td> Average Purchase </td>
<td> Reference Name </td>
<td> Reference Phone </td> </tr>
<tr>
<td> B110 </td>
<td> The Bull </td>
<td> El Toro </td>
<td> Taurus </td>
<td> 215-000-1122 </td>
<td> 215-000-1123 </td>
<td> $800 </td>
<td> ----- </td>
<td> ----- </td> </tr>
<tr>
<td> B112 </td>
<td> Bell Library </td>
<td> ----- </td>
<td> ----- </td>
<td> 608-782-5191 </td>
<td> 000-000-0000 </td>
<td> $320 </td>
<td> Susan White </td>
<td> 608-995-1618 </td> </tr>
</table>
</BODY> </HTML>

```


GAAP [S. de la Fe, Hoffman and Huh 2000]. Different elements included in the DTD can utilize a specified taxonomy by including its namespace prefix before each element name or tag. The use of namespaces enables the use of a broadly shared semantic context and promotes accountability in all documents and transactions that reference the same namespaces.

The process of comparing an instance document to its schema represents the validation step in a document exchange. To validate an XML instance file against its schema, the instance file is read and interpreted in a process called parsing. A typical parser today will handle the following tasks:

- Extraction of data from a submitted document or file so as to create event or new data structures from the original data.
- Examination of the submitted document to see if it meets XML standards.
- Comparison of the submitted document to the structure defined by its related schema to see that the document contains all necessary elements as defined by the schema and that the elements are in the proper order.

Some parsers are *non-validating* in that they do not check the document against a schema but determine only whether the XML instance file is a well-formed file according to standards set by W3C [W3C 2004]. Beyond that, a *validating parser* also tests the message against the predetermined rules of the schema to provide additional assurance about the validity of data format and the integrity of transmission. Note that, while current parsers can validate an instance file with respect to such characteristics as data type and data constraint, neither the parser nor the schema ensure the correctness of the actual data content [Armstrong 2001].

A number of open sources provide XML parsers, including MSXML 4.0, XML Spy and Oracle SDK (Software Development Kit). The XML parser exposes the document structure and content to external applications through a Document Object Model (DOM) programming interface. MSXML 4.0, for example, is a kind of *tree-based parser*, which can describe the relationships among data items using a tree structure, and it uses the DOM as its interface. Developed by W3C, the purpose of DOM is to provide software programmers with a standard connection between the message and the parser, one that is independent of software languages or computing platforms [W3C 2001]. DOM also allows programs and scripts to possess the characteristics of object-oriented programming, facilitating the processing of document content, structure and style.

An XML instance does nothing more than describe the data used in a transaction. Application Program Interfaces (APIs) are used to process the data. W3C has worked to approve and disseminate particular APIs so that people writing in different languages or using different platforms can still acquire the XML-tagged data and manipulate it with any number of applications via standardized API routines. Thus, the behavior (the processing) and the data can be developed independently, which helps realize the goal of data reuse. In addition, by defining a document's logical structure in a common application program method, software packages are more likely to manipulate messages consistently. For example, both leading web browsers (Microsoft's Internet Explorer Version 5.x or above and Netscape Navigator Version 6.x or above) are XML-enabled in that they incorporate XML parser software and can display an XML document's hierarchy in a tree-like structure.

Instructors desiring more material for their own or their students' use might consider the resources listed in Table 11; these particular references have proven useful to this article's authors. Many more books are available at bookstores, and a great deal of additional material can be found online.

Table 11: Suggested Reference Materials for XML

	Resource	Comments
Online Resources	www.xml.org	Lots of background material for new and experienced XML users.
	www.w3schools.com	Free XML and DOM tutorials.
	www.ucc.ie:8080/cocoon/xmlfaq	A good FAQ on XML; some foreign language materials for international students.
	wikibooks.org/wiki/XML	Free XML e-book that can be edited by users of the site.
Hard-copy Resources	Castro, Elizabeth. <i>XML for the World Wide Web, Visual Quickstart Guide</i> . Peachpit Press, 2001.	Easy to understand and inexpensive primer for both instructor and student.

APPENDIX B: INSTRUCTIONS FOR STUDENT USE

Introduction to the e-Hermes system

Go to www.cba.bgsu.edu/faculty_staff/Nicolaou/XMLproject/index.cfm where you will be presented with an Order form for the business-to-business system.

Step 1. Register as a customer of the vendor company. Follow the link in the Order Form to access the appropriate input screen (Customer Registration Form). Then enter an imaginative business name that uses your own recognizable name as part of the business name. For example, Sam Julius might enter as a business customer named AJulius Jewelry@. Continue entering suitable information as requested on the form, but do not use your own private phone number, e-mail address or other personal data.

Once your information is submitted, the system will respond with a customer number. *Make a note of your customer number.* You will need this number when processing transactions.

Step 2a. Go back to the order form and enter the following order information:

Item Number: 001056

Quantity: 5000 (five thousand) cartons

Customer Number: *Your system-assigned number*

Step 2b. You should notice that the system will not accept your transaction because the quantity ordered is greater than the quantity available in the vendor's database. Reenter the order as only 5 cartons.

Step 3. On the second line of the form, enter the following:

Item Number: 002500

Quantity: 4 boxes

Customer Number: *Your system-assigned number*

Check the other validations provided by the system, including format (e.g., date, zip code, etc.) as well as content on item number and customer code. Try submitting the form with some missing values or values that do not meet specified criteria and note the results.

At this time, the only item numbers in the system are the following: 001056, 002002, 002500, 008658, and 102345. Please do not enter huge quantities as the system currently has no re-order programming.