MAPPING UML TO XML: A METHOD FOR TEACHING AN OBJECT-ORIENTED APPROACH TO E-BUSINESS DEVELOPMENT

Jack T. Marchewka
Northern Illinois University

Chang Liu
Northern Illinois University

ABSTRACT

XML (extensible Markup Language) has been gaining notoriety and interest in electronic business applications. As a result, XML has become an increasingly popular topic in academia in order to better prepare students for their chosen careers. Recently, the authors worked closely with a team of four students developing an XML-based application for John Deere in Moline, Illinois. Although the students were able to learn XML code with relative ease, the real challenge occurred when they tried to apply this technology to an electronic business application. By using several tools associated with the Unified Modeling Language (UML), the students were able to develop a conceptual model that aided the students' understanding and provided a seamless application of this technology. Using a conceptual model for development and design provided a better real world abstraction, improved communication and understanding, and resulted in less rework, errors, and inefficiencies. This paper describes how the UML use case model and class diagrams can seamlessly aid an XML-based electronic business application. The experiences of this team provide a basis for action research that should be of particular interest to academics teaching system analysis and design-based courses and to organizations interested in developing XML-based applications.

INTRODUCTION

The growing popularity of electronic commerce has provided new and exciting opportunities for businesses. Technology has enabled organizations to extend their reach globally and to create new business models that extend their strategic visions and improve their competitive positions that were impractical only a few short years ago.

From a business perspective, e-commerce applications can be classified into three categories: business-to-consumer (B2C), intra-organizational, and business-to-business (B2B) (Applegate, Holsapple, Kalakota, Radermacher, andWhinston, 1996). Among these three categories, the impact of B2B marketplace is the largest financially compared to the other two sectors. According to Forrester Research (2001), the B2B economy will soar past $2.7 trillion mark in 2004. Compared with estimates for the B2C market over the same period, the ratio of revenues can be expected to grow more than 10 to 1. Interestingly, the growth of B2B e-commerce is larger than the gross national product of many small countries (Cunningham, 2001). As a result, many colleges and universities have literally jumped on the electronic commerce band wagon by offering programs and courses in electronic commerce and its related application development technologies.

With the current prevalence of using the Web as the platform for application development, XML (extensible Markup Language) has been gaining notoriety and interest for many electronic business applications. XML is a relatively new technology that provides a cross-platform method that can identify, or markup, data. An XML document allows for complete control over the

structure of the data. Subsequently, one only needs to define the tags or elements within the document.

The increasing popularity of XML stems from the ability to separate content from presentation, whereby the developer has the freedom to build an application with its own data structure using XML and HTML to create the user interface. By providing a common language for describing data, XML enables more precise searching, lets businesses share data more efficiently, and makes navigating data much easier (Gottesman, 1998). This allows organizations to not only use the same data definitions across different platforms, but also use the same data differently across different organizations.

Moreover, XML is based on UNICODE. This allows for XML to support not only the Western characters, which have been traditionally supported by ASCII, but Asian characters as well. As a result, XML has been of particular interest to organizations conducting electronic business globally. Since XML is designed as a tool for document-oriented applications, its flexibility is rapidly making it the tool of choice for B2B applications.

BACKGROUND FOR THIS RESEARCH

With support from the Business Information Technology Transfer Center (BITTC) at Northern Illinois University's College of Business, four students under the direction of two faculty members engaged in a project with John Deere in Moline, Illinois to develop an XML-based application during the fall semester of 2001. John Deere was interested in learning more about this relatively new technology and how it could support the relationships among their various business units and suppliers.

The Business Information Technology Transfer Center was founded in 1998 with the intent of creating a unique educational experience for College of Business students at the University. The mission of the Center is to:

- Enhance the educational experience of the University's students by exposing them to actual organizational problems and the application of the latest information technologies and methodologies;
- Cultivate, nurture, and sustain relationships between University's College of Business and organizations through the creation of mutual value;
- Enable College of Business faculty to improve the relevancy of their teaching and research by working closely with various organizations and small student teams; and
- Create and share knowledge from research initiatives with business partners and the academic community.

Although faculty members play the role of coaches, the student team is responsible for defining the goals, objectives, and deliverables associated with the project as outlined in the Center's project methodology. The coach's role is to serve primarily as a resource to the students and offer support and guidance when needed.

Generally, projects are initiated and completed within one semester; therefore, the students are able to experience the entire life cycle of a project. Students are recruited the semester before a project is scheduled and go through a selection process similar to being hired for a position within a firm. In that past, an average of thirty students have applied for each project; however, team size has generally been limited between four and six students that include a mix of graduate and undergraduate students. Students receive three credit hours as elective credit and a $500 scholarship. Students invited to be on a project are chosen based upon:

- Competency of business and technology skills;
- Communication skills;
- Ability to contribute as a team member; and
- Ability to work in an unstructured environment.

Required team meeting times are scheduled as a three-hour class. Team members are expected to attend and participate actively. Although there is no syllabus, each team follows a project methodology to guide them throughout their project. Outside readings and meeting of key project deliverables generally require eight to twelve hours of outside work. Although the students do not have any quizzes or exams, their final grade is determined directly by the completion of key project deliverables. A final presentation to the sponsoring organization, project coaches, and any other interested stakeholders brings closure to the project.

The focus of the John Deere project was to explore electronic business models and, in particular, the electronic procurement of direct materials. Moreover, John Deere was especially interested in evaluating XML as a base technology. Subsequently, the student team was asked to develop a business model and XML-based

prototype application that would serve as a model and an impetus of ideas that would allow John Deere to develop a much larger application in the future.

After extensive research, the students recommended a reverse-auction model to support John Deere's business units and suppliers. Using Web technology, a reverse auction would allow John Deere's business units to post requests for quotations for various items. In turn, the company's vendors could then view these items and place bids over a specific period of time. This model is ideal for procurement because it makes the purchasing process more competitive and can reduce the overall costs of purchasing transactions.

In addition, the coaches taught the student the basics of XML, and several books and an online course provided additional support. However, it became apparent that the challenge was not in learning XML, but understanding how XML could be used to develop an application to support the reverse auction model. Therefore, the students had an understanding of what they wanted to accomplish and the tools to build it, but not the process for achieving it.

Action research provided an important and interesting approach to solving this particular challenge. Action research is becoming more accepted as an alternative to traditional, social sciences or empirical-based research that focuses on studying a particular phenomenon without changing it. Conversely, action research focuses on the practical concerns of individuals or an organization by solving an immediate and problematic situation while expanding scientific knowledge (Rapoport, 1970; Avison, Lau, Neilsen, and Myers, 1999). In a collaborative effort between the researchers and practitioners, action research allows for the simultaneous study of organizational change and the process leading to the creation of the new knowledge that is gained from the experience (Baburoglu and Ravn, 1992).

Under the direction of the faculty coaches, several basic UML models were developed to bridge this gap. This not only aided the students in developing a reverse auction application that would allow multiple vendors to bid on specific items, but allowed for a surprisingly seamless process for mapping UML design models to XML. To illustrate this approach, the remainder of this paper provides an overview of the UML models and how an XML-based application can be developed. The authors believe that this technique can be applied in MIS classes using XML and to organizations interested in developing XML-based applications for electronic business.

UNIFIED MODELING LANGUAGE (UML)

According to Booch, Rumbaugh, and Jacobson (1999), the Unified Modeling Language (UML) is a general-purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of a software system. UML is a product of object-oriented analysis and design (OOA & D) that first appeared in the late 1980s and early 1990s and unifies the methods proposed by Booch, Rumbaugh, and Jacobson (Fowler and Scott, 1997). OOA & D is a direct result of object-oriented programming languages such as SmallTalk and C++. UML has currently emerged as the software industry's dominant object-oriented modeling language (Siau and Cao, 2001).

As a modeling language, UML provides the graphical notation and syntax used to express a design, but does not include a method or process for steps in completing a design. However, UML provides a key technique for improving communication between developers and users and has a common vocabulary of object-based terms and diagramming techniques rich enough to model an applications system from analysis through design (Dennis and Haley, 2000).

An object-oriented approach to systems analysis and design has several benefits over the traditional process or data focused approaches. Moreover, information systems developed using these traditional approaches have been regarded as being more error-prone, expensive to build, and inflexible to maintain (Satzinger and Orvik, 2001). Since each object is small and self-contained, the object-oriented approach is more manageable since the complexity of systems development is reduced. In addition, object-oriented systems provide a higher degree of reusability and therefore a less expensive to build and maintain, while leading to a higher quality system (Satzinger and Orvik, 2001).

UML defines a set of nine object-oriented diagramming techniques that can be used to model a system (Dennis and Haley, 2000); however, only two of these modeling techniques were used to develop the reverse-auction application. These two techniques included the use case and class diagrams. The remainder of this section explains the two diagramming techniques and illustrates how they were used by the student team.

Use Case Diagrams

One of the first steps in designing the reverse-auction application involved the creation of a use case diagram. A use case diagram provides a simple, yet effective, high-level abstraction of the system to be developed. A basic use case diagram includes stick figures as actors – i.e., people, organizations, or other external entities who use or interact with the system. A box represents the system boundary and various use cases within the system boundary define the functionality or high-level features of the system. Lines connect the actors with the various use cases to define the relationships or interactions.

Figure 1 provides a use case diagram developed by the student team. Actors include the various business units and suppliers, as well as a John Deere administrator who would support the application. The various use cases within the system boundary were subsequently expanded as use case scenarios that basically "story-boarded" how each actor would use or interact with the system. The use case diagram and associated scenarios were reviewed by a manager at John Deere who acted as the project sponsor. After receiving his approval, the team then created a class diagram.

The Class Diagram

Objects are the heart of the object-oriented approach, since just about everything can be viewed as an object. One of the key diagramming techniques is the class diagram. The class diagram is a technique that can describe the types of objects (i.e., classes) in a system and the static relationships that exist among them. Relationships can be associations (e.g., a customer orders a product) or subclasses (a customer is a kind of person). In addition, class diagrams define the operations or methods (i.e., things an object or class can do), as well as the properties or attributes of a particular class.

In addition to associations, class diagrams can be useful for defining aggregation and composition. More specifically, aggregation allows the developer to model the "is-a-part-of" relationships (e.g., a wing is a part of an airplane or package contains items). By modeling object classes and their various relationships, the developer is able to create a better abstraction of the real world system. As a result, an application system can be developed that better reflects the needs of the user or organization.

Figure 2 provides the class diagram developed for the reverse-auction model. Each class includes the attributes and methods associated with the class. The methods in the class diagram were derived from the use cases specified in the use case diagram.

MAPPING UML TO XML DOCUMENTS

Although XML has been gaining interest and popularity, the design of a valid and well-formed XML document and XML-based e-business application remains a challenge. Therefore, an XML development process should be carried out in a rational, disciplined, effective, controlled, and uniform way to design e-business applications. Even though XML is not an object-oriented language, the opportunity to apply OOA& D techniques exists.

At the core of XML is a Document Type Definition (DTD) or a schema used to define the data contents of the XML document. A DTD specifics which tags or elements a document may have, what they are composed of, and how they are related to one another within the structure of the document in terms of contents, sequences, nesting, etc. (Gibb, O'Donnell and Leon, 2000). It proposes many mechanisms to define structuring rules for documents. In short, the purpose of a DTD is to provide a grammar for an electronic document.

For processing an XML-based electronic document, it must be "well-formed" and "valid." A well-formed XML document conforms to the syntax rules set up for XML by the World Wide Web Consortium (W3C) in the XML 1.0 specification. However, this really means that the XML document contains one or more elements, and one element, called the root, contains all the other elements. In addition, a "well-formed document" means each element also nests inside any enclosing elements properly. An XML document is "valid" if it references a DTD or schema, and the DTD or schema complies with the W3C standard. Although most XML browsers will check to ensure that the XML document is well formed, only a few have the capability to check to verify that it is valid.

To bridge the gap between objected-oriented system analysis and design and the development for XML-based B2B applications, we can use UML's class diagram to model XML's DTD structure.

Mapping to the DTD

The class diagram in Figure 2 shows the classes, attributes, and various associations. It provides a seamless guide for preparing the XML DTD. Figure 3

FIGURE 1
USE CASE DIAGRAM

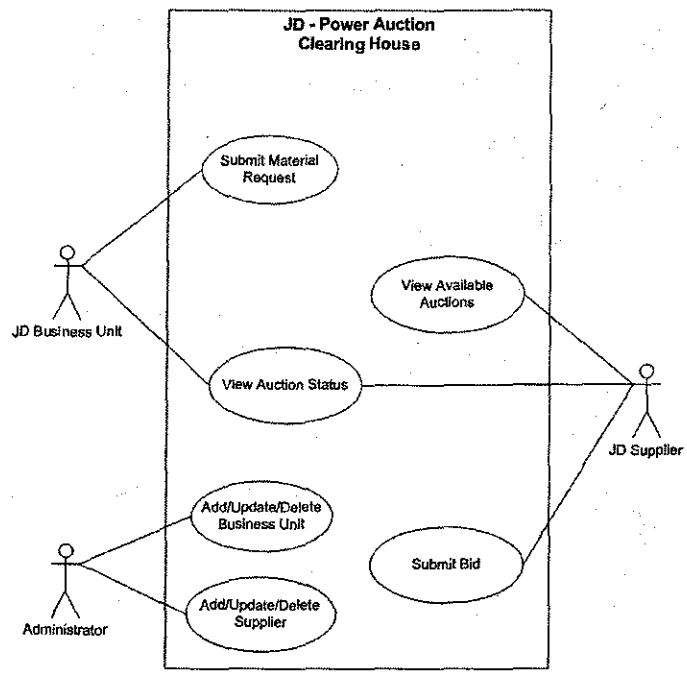

FIGURE 2
CLASS DIAGRAM

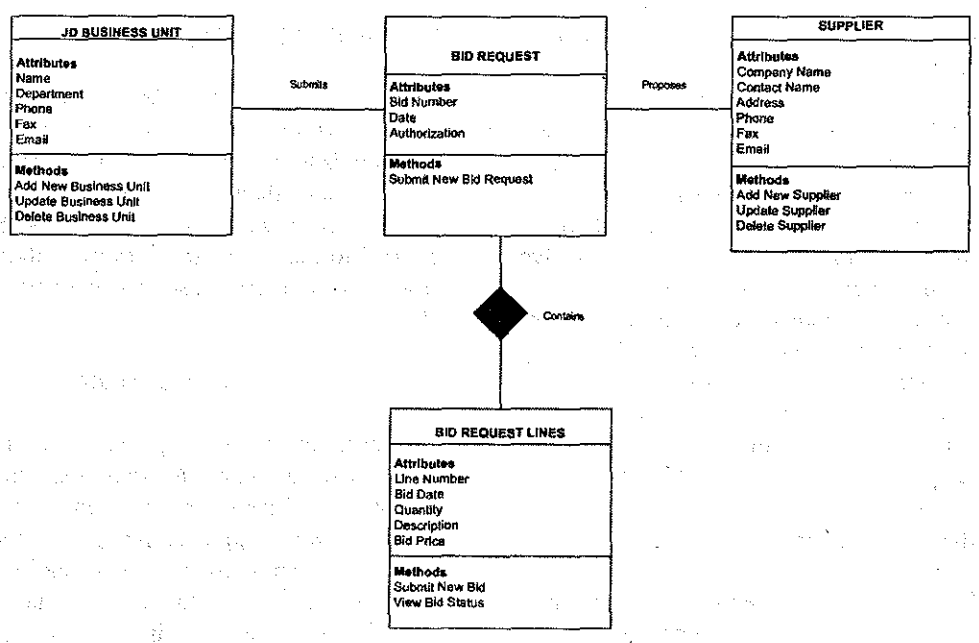

FIGURE 3
AN ELEMENT-BASED DTD MAPPED FROM THE CLASS DIAGRAM

```
<!ELEMENT XML_DOCUMENT (JD_BUSINESS_UNIT, BID_REQUEST,
SUPPLIER+, BID_REQUEST_LINE+)>

<!ELEMENT JD_BUSINESS_UNIT (Name, Department, JD_Phone, JD_Fax, JD_Email)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Department (#PCDATA)>
<!ELEMENT JD_Phone (#PCDATA)>
<!ELEMENT JD_Fax (#PCDATA)>
<!ELEMENT JD_Email (#PCDATA)>

<!ELEMENT BID_REQUEST (Bid_Number, Date, Authorization)>
<!ELEMENT Bid_Number (#PCDATA)>
<!ELEMENT Date (#PCDATA)>
<!ELEMENT Authorization (#PCDATA)>

<!ELEMENT SUPPLIER (Company_Name, Contact_Name, Address, S_Phone, S_Fax,
S_Email)>
<!ELEMENT Company_Name (#PCDATA)>
<!ELEMENT Contact_Name (#PCDATA)>
<!ELEMENT Address (#PCDATA)>
<!ELEMENT S_Phone (#PCDATA)>
<!ELEMENT S_Fax (#PCDATA)>
<!ELEMENT S_Email (#PCDATA)>

<!ELEMENT BID_REQUEST_LINE (Line_Number, Bid_Date, Supplier, Quantity,
Description, Bid_Price)>
<!ELEMENT Line_Number (#PCDATA)>
<!ELEMENT Bid_Date (#PCDATA)>
<!ELEMENT Quantity (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
<!ELEMENT Bid_Price (#PCDATA)>
```

shows an element-based DTD with a root element called "XML_DOCUMENT." Each class is mapped as a group element within the XML DTD. Subsequently, the attributes of each class in the class diagram are mapped as PCDATA with their associated group element.

As illustrated in Figure 3, each element and sub element of the XML DTD maps directly to the super classes and subclasses of the class diagram. However, one could use the XML attributes rather than elements and sub elements for the DTD mapping. Figure 4 shows this alternative way to define and map external DTD structure by using attributes. It appears that elements and sub elements are better suited to an object-oriented approach while using attributes is better suited to a relational database perspective. Regardless, using either elements or attributes will produce a well-formed and valid XML document when performing the UML/XML mappings.

Generating the XML Document

Once the DTD is created, it may be inserted directly into the XML document or referenced from within the document. Figure 5 shows partial codes for the XML document that references an external DTD called BidDTD.dtd.

Mapping the Methods

Although the example XML documents provide a structure for the reverse-auction data, they do not include any methods for interacting with the data. Using the use case and class diagrams, a combination of scripting languages, such as Java Script and Active Server Pages, was used to implement the various methods that would allow the user to create, search, and update the auction bids.

DISCUSSION

Using an action research approach, this paper provides a means for developing XML-based electronic business applications. From our experience, we found that building an XML-application requires more than just a good-working knowledge of the technology. A process and technique for developing an accurate, real-world abstraction is required to improve understanding and communication among the development team. Although

FIGURE 4
AN ATTRIBUTE-BASED DTD FROM THE MAPPED FROM THE CLASS DESIGN

```
<!ELEMENT XML_DOCUMENT (JD_BUSINESS_UNIT, BID_REQUEST,
SUPPLIER+, BID_REQUEST_LINE+)>

<!ATTLIST JD_BUSINESS_UNIT EMPTY>
      Name        CDATA    #REQUIRED
      Department CDATA     #REQUIRED
      JD_Phone          CDATA    #REQUIRED
      JD_Fax            CDATA    #REQUIRED
      JD_Email          CDATA    #REQUIRED>

<!ELEMENT BID_REQUEST EMPTY>
<!ATTLIST BID_REQUEST
      Bid_Number        CDATA    #REQUIRED
      Date        CDATA    #REQUIRED
      Authorization     CDATA    #REQUIRED>

<!ELEMENT SUPPLIER EMPTY>
<!ATTLIST SUPPLIER
      Company_Name  CDATA    #REQUIRED
      Contact_Name      CDATA    #REQUIRED
      Address           CDATA    #REQUIRED
      S_Phone           CDATA    #REQUIRED
      S_Fax       CDATA    #REQUIRED
      S_Email           CDATA    #REQUIRED>

<!ELEMENT BID_REQUEST_LINE EMPTY>
<!ATTLIST BID_REQUEST_LINE
      Line_Number       CDATA    #REQUIRED
      Bid_Date          CDATA    #REQUIRED
      Quantity          CDATA    #REQUIRED
      Description CDATA    #REQUIRED
      Bid_Price         CDATA    #REQUIRED>
```

FIGURE 5
THE XML DOCUMENT BASED ON THE DTD

```
<?xml version="1.0" ?>
<!DOCTYPE XML_DOCUMENT SYSTEM "BidDTD.dtd" >
<XML_DOCUMENT>

    <JD_BUSINESS_UNIT>
        <Name>Joe Smith</Name>
        <Department>Accounting</Department>
        <JD_Phone>233-737-2323</JD_Phone>
        <Fax>233-737-7600</Fax>
        <JD_Email>jsmith@john_deer.com</JD_Email>
    </JD_BUSINESS_UNIT>

    <BID_REQUEST>
        <Bid_Number>1001</Bid_Number>
        <Date>5/23/2001</Date>
        <Authorization>Robert Rollins</Authorization>
    </BID_REQUEST>

    <SUPPLIER>
        <Company_Name>Reynolds Metals</Company_Name>
        <Contact_Name>Mary Jones</Contact_Name>
        <Address>2000 First Street, Dekalb, IL 60115</Address>
        <S_Phone>815-758-8888</S_Phone>
        <S_Fax>815-758-7777</S_Fax>
        <S_Email>maryjones@reynolds.com</S_Email>
    </SUPPLIER>

    <BID_REQUEST_LINE>
        <Line_Number>1001_A</Line_Number>
        <Bid_Date>5/23/2001</Bid_Date>
        <Quantity>200 tons</Quantity>
        <Description>Steel</Description>
        <Bid_Price>$10,000</Bid_Price>
    </BID_REQUEST_LINE>

    <BID_REQUEST_LINE>
        <Line_Number>1001_B</Line_Number>
        <Bid_Date>5/23/2001</Bid_Date>
        <Quantity>10 tons</Quantity>
        <Description>Rubber</Description>
        <Bid_Price>$5,000</Bid_Price>
    </BID_REQUEST_LINE>

</XML_DOCUMENT>
```

this notion is commonly-held, the interesting lesson learned was how well UML modeling techniques map to XML.

At the end of the project, the faculty coaches met separately with the team and the John Deere sponsor. From the student's perspective, the use of the techniques described in this paper provided much needed direction at a critical point in the project. In addition, the project sponsor was extremely pleased with the quality of the application and was anxious to demonstrate the system to several key business unit managers.

The lessons learned from this project are based on a collaborative effort by the BITTC coaches and student team members to solve an immediate problem at hand. Although no empirical evidence can support these claims, this paper documents the tools, processes, and mental thought processes that were used. These same techniques may be replicated and generalized to the teaching of systems analysis and design-based classes and for practitioners interested in developing XML-based applications. Therefore, the action research method utilized in this paper should provide the beginning and foundation for future empirical research that may look at the effectiveness of this approach in the classroom or by organizations.

Finally, the approach used by the project team utilized only two UML diagramming techniques. Other techniques, such as state-transition diagrams, interaction diagrams, and so forth should also be considered and evaluated empirically.

REFERENCES

Applegate, L. M., Holsapple, C. W., Kalakota R., Radermacher, F. J., and Whinston, A. B. (1996). "Electronc Commerce: Building Blocks of New Business Opportunity." *Journal of Organizational Computing and Electronic Commerce* 6(1): 1-10.

Avison, D., Lau F., Neilsen, P. A., and Myers, M. (1999). "Action Research." *Communications of the ACM* 42(1): 94-97.

Baburoglu, O. N. and Ravn I. (1992). "Normative Action Research." *Organization Studies* 13(1): 19-34.

Booch, G., Rumbaugh, J., and Jacobson, I. (1999). *The Unified Modeling Language User Guide*, Reading, Massachusetts: Addison-Wesley.

Cunningham, M. J. (2001). *How to Build a Profitable E-commerce Strategy.* Cambridge, MA: Persus Publishing.

Dennis, A. and Haley, W. B (2000). *Systems Analysis and Design: An Applied Approach.* New York: John Wiley & Sons, Inc.

Fowler, M. and Scott K. (1997). *UML Distilled: Applying the Standard Object Modeling Language.* Reading, MA: Addison-Wesley.

Gibb, F., O'Donnell R., and Leon, R. (2000). "The Integration of Information Retrieval Techniques Within a Software Reuse Environment." *Journal of Information Sciences* 26(4): 211-226.

Gottesman, B. Z. (1998). "Why XML Matters." *PC Magazine*: 217-238.

Rapoport, R. N. (1970). "Three Dilemmas in Action Research." *Human Relations* 23(4): 499-513.

Satzinger, J. W. and Orvik, T. V. (2001). *The Object-oriented Approach: Concepts, System Development and Modeling with UML.* Boston, MA: Course Technology.

Siau, K. and Cao, Q. (2001). "Unified Modeling Language (UML)—A Complexity Analysis." *Journal of Database Management* 12(1): 26-34.