

# Enterprise Development Technologies and E-Commerce

Chris Geer,  
cgeer@flashmail.com

and

Kyle Lutes  
kdlutes@tech.purdue.edu

Computer and Information Systems Technology, Purdue University  
West Lafayette, IN 47906, United States

## Abstract

This paper contains excerpts from three distinct papers written during an independent study course. This course was intended to research enterprise technologies including Java, JINI, and LDAP. The focus of the majority of the research is the area of E-Commerce and how these technologies can be used and exploited for the purpose of E-Commerce. The project will also delve deeply into design strategies for a completely modular and flexible system. Its purpose will be to show how enterprise technologies being used today and in the future can be integrated to allow organizations to conduct business transaction over a public network (the Internet).

**Keywords:** E-Commerce, Java, XML, JINI, LDAP

## 1. THE OBJECTIVES OF E-COMMERCE

Over the past few years the structure of how business is conducted has been changing. There are many new time and money saving ways for individuals or businesses to conduct business. One of the most recognized new buzzwords in business today is E-Commerce, which stands for Electronic Commerce. E-Commerce is a term that is thrown around in conversations in which it has many different meanings. E-Commerce is really an umbrella term that covers a wide range of concepts and practices that allow individuals and companies to conduct their business.

The topic of E-Commerce can be broken down into two distinct camps. The first of these camps is business-to-consumer E-Commerce. Business-to-consumer is what many people think of when they hear *E-Commerce*. This idea includes the examples of Amazon.com or Dell.com where companies have a web page that allows typical consumers to browse a catalog and purchase products. These sites offer a wide range of services anywhere from a simple catalog with pictures allowing the customer to log on and buy a book, to sites allowing customers to design their own custom version of a product that will then be built and shipped to specification. Another aspect of business-to-consumer

E-Commerce includes financial and information services. These include online banking, online bill paying, and secure information distribution. (Microsoft, 1999) While these are what most people think of as E-Commerce applications, they are really just a small part of what E-Commerce really is.

E-Commerce is really anything that allows business to be conducted over an electronic medium, in most recent cases, the Internet. Using that broader definition there is a second camp of E-Commerce applications, which is business-to-business E-Commerce. These applications have been in use for many years by many businesses over their private networks but are now evolving to encompass the use of the Internet. These applications include corporate purchasing and value chain integration.

It's hard to be in business today without at least being introduced to some E-Commerce ideas, but the real question is why would a company want to invest in an E-Commerce solution. The reason to invest in a business-to-consumer solution is pretty self-evident: to reach more customers. But the reason to invest in a business-to-business application isn't so obvious. In fact there isn't any one overwhelming reason to invest in a business-to-business solution. There are different

reasons to invest depending on the function desired of the application.

Many companies are being enticed by the corporate purchasing application end of E-Commerce. This allows companies to set up websites that can enable their employees to directly order office supplies, for example. It allows for the setting up of business rules to be able to determine if an order is within company policies and automatically authorize and place or deny the order. (Microsoft, 1999) The objectives of a system like this are reduce turnaround time on office supply orders while also reducing the cost incurred in placing and managing them. This is accomplished by taking the person out of the equation. By fully automating a system like this the need for human interaction is removed, which saves time and money. Another objective of this type of system is accuracy. By placing an order electronically the potential of introducing error by having to pass along information from person to person is reduced.

Manufacturing companies on the other hand might be more interested in a value chain system. (Microsoft, 1999) This system would allow the company to connect to their suppliers, plants and distributors in real-time. This gives all parties the ability to know when, for example, the plants ran out of raw materials and thus allowing the supplier to ship the product in a just-in-time fashion. The objectives of a system like this are many -fold, but the obvious ones, again, are to save time, money and have more accurate orders. When a business looks at this type of system, it is the time saving that is of real interest. If a manufacturing facility is working under the "just-in-time" paradigm, then time is of the essence. When a distributor/customer places an order, a company has to be sure it can fill the order. It takes time (and money) to check inventory with suppliers to get what is needed. But using this type of system allows all parties to be connected in real-time so that when an order is placed, inventory is automatically checked and orders are placed to suppliers to ease shortages.

Different outcomes and benefits result depending on the use of an E-Commerce solution. While E-Commerce is a term that is widely used, understanding in what sense it is being used is important before making judgments about how the system works. Since there are different possible solutions and applications there are different benefits and costs associated with those solutions. There is no one "E-Commerce" solution or package. A solution has to be developed around specific wants and needs. But as an overall generalization, E-Commerce solutions allow orders to be placed more accurately while saving manpower and time doing so.

## **2. TECHNICAL SUMMARY OF TECHNOLOGIES REVIEWED FOR PILOT APPLICATION**

(See Figure 1 for references to the pilot application)

Today's new systems are vastly different from those of the recent past. Previously, systems were frequently built by management describing analysts their needs, and then an in-house, custom solution was built from scratch. In the current state of the industry, that type of development is very rarely seen. The scenario today usually involves many stages of scouring the market to see what product or group of products will provide the needed functionality and then the developers' jobs are to integrate those products. While developers don't have to write as much code to make the products work, successfully integrating multiple companies' products can be a daunting task. E-Commerce projects are notorious for scenarios involving a pre-built billing system and inventory system that don't communicate, and a developer trying to design an E-Commerce system in the middle that has to communicate and schedule with both systems along with possibly a dozen other systems. These systems require software developers to understand and use a wide array of technologies.

Since the idea behind E-Commerce systems is to be able to allow multiple organizations to communicate and conduct business over an electronic medium, the technologies to facilitate that are of prime concern. For example, the project for this independent study course uses XML to transmit information over a standard HTTP connection.

### **XML**

XML or eXtensible Markup Language is one of the industry's hot technologies. It has been offered up as the silver bullet technology to solve all problems of communicating. While we don't think it is a cure for every technology problem, it does seem to lend itself well to E-Commerce scenarios.

Like E-Commerce, XML is one of those terms that has many different meanings depending on how it is used. To describe how it is being used here we think the Internet.com Corporation said it best: "Extensible Markup Language (XML) is a human-readable, machine-understandable, general syntax for describing hierarchical data, applicable to a wide range of applications." While that definition can seem a little overwhelming, all it really means is that XML is a way to define data that is easily understood by many heterogeneous systems.

XML was a good choice in this project because the communications protocol chosen was HTTP, which transfers information as ASCII text, which is human-readable and not in binary format. XML documents are easily transmitted over simple HTTP sessions since they are stored as simple ASCII text. Since there is no special encoding or compression in the XML format there are no special steps needed to transmit an XML document. It can be sent just like a typical HTML page. This allows for better security on external networks since no other

ports have to be opened, except for the already open web server port.

XML documents resemble standard HTML documents except that there are no predefined tags. The power of XML comes from being able to define your own tags that allow you to define your own data. The only requirement is that the sending and receiving parties should be aware of the schema.

The following example demonstrates the easy readability of XML:

Example:

Let's say that you need to send data about your child's little league team. Basically a team consists of players, each of whom has a name, age, address and such information. Maybe the team also owns some equipment that you need to store and the team's schedule. You could choose to write all of this in a simple document that would be easy to read to the human eye, but would be difficult to be read by a machine. XML offers an alternative. This document in XML could look like this:

```
<TEAM>
  <NAME>East side slayers</NAME>
  <PLAYER>
    <NAME>Joe Stevens</NAME>
    <ADDRESS>123 West Wing</ADDRESS>
  </PLAYER>
  <PLAYER>
    <NAME>Mike George</NAME>
    <ADDRESS>123 Main St</ADDRESS>
  </PLAYER>
  <EQUIPMENT>
    <TYPE>Bat</TYPE>
    <DESCRIPTION></DESCRIPTION>
  </EQUIPMENT>
  <EQUIPMENT>
    <TYPE>Glove</TYPE>
    <DESCRIPTION></DESCRIPTION>
  </EQUIPMENT>
  <SCHEDULE>
    <GAME>
      <TIME>12:00</TIME>
      <DATE>5/30/00</DATE>
      <OPPONENT></OPPONENT>
    </GAME>
    <PRACTICE>
      <TIME>1:00</TIME>
      <DATE>5/21/00</DATE>
    </PRACTICE>
  </SCHEDULE>
</TEAM>
```

While looking at this might be a little confusing it is still readable. To a computer it is much easier to understand. With the starting and ending tags the depth of objects can be defined and attributes can be linked to each object. Even though this form isn't as readable to a human, there is a solution to that, too. Extensible

Stylesheet Language (XSL) allows developers to define a look and feel for each object so this information can be displayed in a form much more readable to a human.

In our pilot project XML allows us to store our orders in a format that is readable and easily transmitted across an already standard and existing infrastructure. XML also fits nicely into the Java architecture that is being used to build the pilot application. Using Java's XML framework, a developer can take incoming XML documents and convert them directly to true objects and the other way around. This allows for very easy transition from working with an order that happens to be a Java object to an XML document for easy transmission, and back into a Java object at the other end for further processing.

**Web Server/Application Server and Servlets**

As mentioned above, one of the reasons that XML was chosen for this project was easy transferability over the desired infrastructure. The infrastructure that is needed consists of web servers and application servers.

Web servers on both ends of the connection provide the support needed to create the HTTP connection to the other machine. They are the interface between the application and the outside network. Since most companies these days are already on the web, these are already in existence and can be leveraged pretty easily.

Application servers provide the ability to link the java code directly to the web server through the use of Java Servlets. "Java Servlets provide web developers with a simple, consistent mechanism for extending the functionality of a web server..." (Java Servlet API) These basically work by having the web server map a certain HTTP address to a specific Servlet and when that address is accessed the Servlet is called. This allows developers the full power of Java when developing for the web. This technology is similar in a way to calling Dynamic Link Librarys (DLL) from a Microsoft Active Server Page (ASP), except that it's tailored specifically for use in online processing.

In our project the orders are processed on the customer's end and then converted to XML that is pushed out on an outside network. The XML document is then received by the remote web server and mapped to the application server and a particular Servlet. The Servlet then takes the incoming XML document and converts it back into a Java object that is passed along to the processing components.

**Directory Servers and LDAP**

Directory servers are not a new concept. A directory server's function is to store information in a certain format that is optimized for retrieval. These can be anything from an employee directory to a product catalog storing any combination of data.

Many people are familiar with storing similar information in a typical relational database, and don't see the need to have a directory server. While the same effect can be achieved by storing information this way, some nice features of a directory would be missed. First, directory servers are optimized for searching and retrieving data. There can be significantly less time spent on searching for data stored in a directory server than in a typical relational database. Secondly, directory servers are accessible from a common LDAP interface.

LDAP stands for Lightweight Directory Access Protocol and was originally built to be used as a front end for the X.500 protocol. (LDAP Central) LDAP is a client-server protocol that is used to communicate with a directory service. LDAP provides a standard interface to any directory server.

LDAP is being used in the pilot project to be able to store customer information for the distributor. This is customer information, which needs to be accessed not only from this application, but also from the billing apps and many other applications

#### **JINI**

With previously mentioned technologies in place, a very efficient and powerful system can be built, but it lacks easy scalability. To solve this problem the Java JINI technologies were used. JINI is a set of technologies that allows the creation of services that dynamically join and leave communities when they are available. An example of this would be a typical office situation with printers and computers. This technology allows all of the printers that are online to register with a central lookup service and allow programs to search the lookup for a particular printer on which to print. Also, the technology can be configured so that when a new printer comes online and registers itself. Users could then be notified and given access to the new printer.

JINI allows for great flexibility in a network that changes. In the pilot application, JINI was chosen on the manufacturing side because it gives the manager the ability to control different factory equipment that might be brought up and down through out the day. This way when the equipment comes up, it is automatically registered with the manager and brought under control.

### **3. PILOT APPLICATION**

The pilot application for this independent study project had two distinct goals. The first goal was to develop an idea for how a business-to-business e-commerce application would work between a supplier and a manufacturer. Second, and probably the more interesting goal was to build this system completely modular to allow for seamless integration between all parts, yet allow for individual parts to be switched out at will without an impact on the system as a whole and no configuration changes to the other parts. (See Figure 1)

The design used to accomplish this was JINI technology along with Java's RMI (Remote Method Invocation) technology. JINI allows for parts of the system to advertise their services when they are available, giving us a means for each part to be able to find out about one another dynamically. RMI gives the ability for objects to communicate with other remote objects that provide services to them. Combining the two allows parts of the system to know when other parts are available and a way to communicate with them. For example using JINI a management console can monitor the network and know when equipment comes online and be able to use RMI to directly control it. This combination of technologies is the basis for the project.

Along with using JINI and RMI to communicate within the company's objects, a way to communicate between companies, over the Internet, was also needed. To solve that problem XML was proposed to be the communications medium. Using XML allows for the communications to take place over HTTP, a standard protocol that almost every company has handled. XML also makes communication easy because it has the ability to transfer data in a logical fashion that can be easily parsed at the other end. The actual communication takes place by one end taking the information that needs to be sent and putting it into an XML template. The template is then sent over a connection to the second company, which reads in the document and uses Java's Document Object Model to convert the data into a Java object, which then can be used by other processes.

The last section of the application was to use a LDAP (Lightweight Directory Access Protocol) server to store customer information. This gives fast and easy access to information such as addresses, phone numbers and the like. In this application, the server is used to verify orders.

#### **Findings**

While on its own, each one of these technologies is a very good and useful entity, trying to combine RMI and JINI proved to be a challenge. Taken separately, RMI provides an easy to implement and powerful way to provide remote method calls to an application and to allow computing over multiple computers. Similarly by itself JINI provides a very nice way to be able to have a service announce that it's available and provide a way to contact it.

On their own each work perfectly, but combining them caused major problems. As far as we can tell the problems lie within the realm of the code base of each object. The way Java provides for each technology to share objects is by the use of a code base where all of the classes that need to be transferred between objects are stored. JINI uses the code base to let each object that is trying to download a proxy know where the proxy is located and RMI uses the code base to store the stub files for remote objects. Since we are trying to combine

JINI and RMI to allow an object to download a JINI proxy that then uses RMI to communicate with a remote service, it seems as though the code bases get confused and won't work. At this point this is just an assumption because we haven't seen anything to prove this or anything that says otherwise.

To try and solve this problem we searched through documentation on all technologies involved, newsgroups on the topics, and discussions groups. During the search we didn't see any information on people trying to combine these two, or references to the problem we were encountering, so there was little help in solving it. The next step was to post our problem to these groups and hope someone would answer. At this time of this writing, no suggested solution has been posted.

In reference to the other technologies being used, LDAP is a tried and true technology that is currently being implemented in many companies across the world. It provides a very flexible means to store frequently used information that allows for quick retrieval. Also, using LDAP allows access to the data from many different applications.

While XML has been in the news for the past couple of years and people have questioned its usefulness, we think in this situation it is an appropriate technology. It allows for easy transport since it can be transported over already existent connections (Internet for example) and it has the nice feature of being in plain ASCII text so it's easily readable by computers and humans alike. Java has steadily increasing support for XML and with its current ability to take a XML document and convert it over to a Java object that can be further processed it is very powerful. While XML isn't going to replace RMI or

other similar technologies in our opinion, situations like this allow XML to shine.

#### 4. SUMMARY

This project has been a challenge because it has been hard to implement a system that is completely flexible and modular while also seamless. It's also been frustrating working with these technologies and running into "brick walls" with technical issues.

While the technical working of the application can't be completely proven to work together at this point, the theoretical potential of this application is immense. Providing the technical problems can be fixed with more research this system would provide industry with a completely integrate and flexible system. The system would still function even if systems came up and down throughout the day. It would also allow for parts to be switched out as long as they still implement the same interfaces without causing any problems.

This type of system design is perfect for an E-Commerce application because it has so much flexibility. Since an E-Commerce application is required to communicate with so many outside and different systems, the flexibility is a must. This design allows for each outside system to be modified and only affecting the component that is the interface to the E-Commerce system.

With system design becoming more and more modular it is becoming more of a challenge, but is also creating easier upkeep.

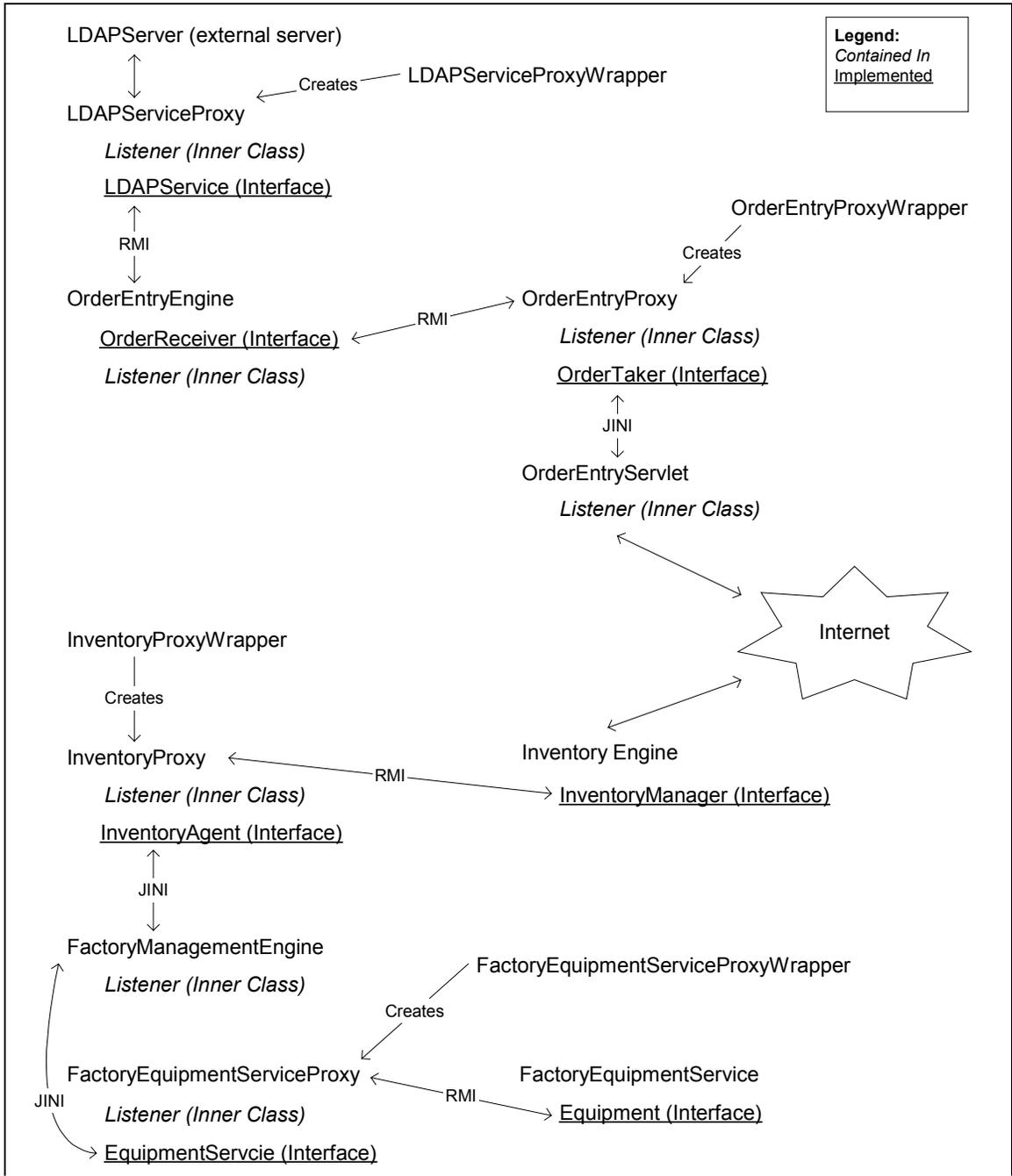


Figure 1

## 5. REFERENCES

- Berg, C. (1999) Advanced Java 2 Development For Enterprise Applications (2<sup>nd</sup> ed.). Prentice Hall PTR
- Edwards, K. (1999) Core JINI. Upper Saddle River, NJ: Prentice Hall PTR
- Golomb, K, Sorgie, T. (1998). How Do we Ensure Communications From a Java Applet?. Dr Dobb's Journal, 23 (6), 107-9+
- Horstmann, C., & Cornell, G. (1999) Core Java 1.2 Volume we – Fundamentals. Palo Alto, Ca.: Sun Microsystems Press
- Horstmann, C., & Cornell, G. (2000) Core Java 2 Volume II – Advanced Features. Palo Alto, Ca.: Sun Microsystems Press
- Hussain, B. (1999). Understanding LDAP. Dr Dobb's Journal, 24 (3), 78+
- Laurent, S. (1999) XML: A Primer. IDG Books Worldwide
- Lewis, J. (1996, Dec) LDAP Unites The Internet. Byte, pp 121-2+
- Severance, C. (1997). Could LDAP Be The Next Killer DAP. Computer, 30, 88-9
- Sikorski, R, Peters, R. (1999). Digital Security, part II. Science, 283 (5405), 1133-4
- Wong, H. (1999). JINI and network-enabled devices. Dr Dobb's Journal, 24 (7), 21-2+
- FreeS/WAN Project [online]. Available: <http://www.xs4all.nl/~freeswan/>. (February 2000).
- Java Servlet API [online]. Available: <http://java.sun.com/products/servlet/index.html>. (March 2000).
- Internet Standards. (1997, Feb) Byte, p. 72
- Lightweight Directory Access Protocol [online]. Available: <http://www.umich.edu/~dirsvcs/ldap/>. (February 2000).
- Linux Directory Services [online]. Available: <http://www.rage.net/ldap/>. (February 2000).
- Mod\_ssl: The Apache Interface to OpenSSL [online]. Available: <http://www.modssl.org/>. (February 2000).
- Network Working Group (December 1997). Rfc2251 [online]. Available: <http://www.cis.ohio-state.edu/htbin/rfc/rfc2251.html>. (February 2000).
- OpenSSL: The Open Source toolkit for SSL/TLS [online]. Available: <http://www.openssl.org/>. (February 2000).
- Secure Sockets Layer (SSL) resources [online]. Available: <http://www.computerworld.com/home/features.nsf/All/980720ssllinks>. (February 2000).
- Sun in Electronic Commerce [online]. Available: <http://www.sun.com/solutions/third-party/ecommerce/>. (February 2000).
- What Is... [online]. Available: <http://www.whatis.com>. (February 2000).
- XML: Extensible Markup Language [online]. Available: <http://w3c.com/Authoring/Languages/XML/>. (March 2000).
- (1999) Microsoft's Electronic Commerce Strategy. Microsoft Corporation.
- (1999) LDAP Central – Your Central Resource For Directory Service & LDAP [online]. Available: <http://www.ldapcentral.com>. (February 2000).