

**An Object-Oriented Architecture for  
Business-To-Consumer  
Electronic Commerce On The Internet**

**A White Paper**

**Scott W. Ambler**  
Software Process Mentor

**Material for this White Paper has been taken from  
*Building Object Applications That Work*  
by Scott W. Ambler  
SIGS Books, 1997**

**This Version: April 2, 1999**

**Copyright 1997 Scott W. Ambler**

## Table Of Contents

<b>1. AN ARCHITECTURE FOR SUPPORTING ELECTRONIC COMMERCE.....</b>	<b>1</b>
<b>2. PAYMENT PROCESSING ON THE INTERNET.....</b>	<b>2</b>
<b>3. IT'S THE WORLD WIDE WEB – INTERNATIONAL ISSUES .....</b>	<b>3</b>
<b>4. YOU CAN SELL BOTH PHYSICAL AND VIRTUAL PRODUCTS INTERNATIONALLY .....</b>	<b>3</b>
<b>5. DON'T FORGET TAXES .....</b>	<b>4</b>
<b>6. AND THE GOOD NEWS IS.....</b>	<b>4</b>
<b>7. USING SMALLTALK AND JAVA TO DEVELOP ELECTRONIC COMMERCE APPLICATIONS.....</b>	<b>5</b>
<b>8. COMPARING JAVA AND SMALLTALK .....</b>	<b>6</b>
<b>9. IN SUMMARY.....</b>	<b>6</b>
<b>10. ABOUT THE AUTHOR.....</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>

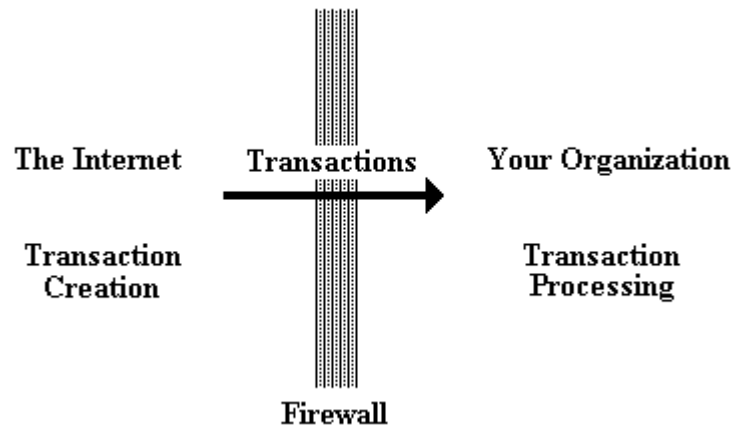
Most companies are either starting to do business on the Internet or soon will be, and because this environment is significantly different than what organizations are used to I want to provide a few words of advice. Organizations that understand the issues surrounding electronic commerce (e-commerce) are the ones that are going to succeed in the 21st century, and the ones that don't will flounder at best, and will fail at worst.

**Understanding e-commerce will prove to be the key to success in the 21st century.**

In this white paper we will discuss the following issues:

- An electronic-commerce architecture
- Payment processing
- International issues
- Selling physical and virtual products on the web
- Taxes on electronic commerce
- Using Java and Smalltalk to support e-commerce

## 1. An Architecture For Supporting Electronic Commerce



**Figure 1. A high-level architecture for electronic commerce.**

In Figure 1 we see a high-level overview that supports e-commerce on the Internet. The basic idea is that your potential customers go to your world-wide web (WWW) page, find a product or service that they wish to purchase from you, and then purchase it while they are at the web page. There are three components to this architecture: the Internet, a firewall, and your organization. The Internet is where you will interact electronically with your customers, your firewall will provide you with reasonable protection against people who wish you harm, and your organization's systems will process the business transactions generated on the WWW by your customers. Later in this white paper we will discuss a development approach that uses Java and Smalltalk to support this architecture.

To understand why we need to change our systems infrastructure we must first put the requirements of electronic commerce into perspective. We're talking about the WORLD wide web, and that means international commerce. Doing business internationally means handling multiple languages, multiple currencies, multiple cultures, multiple tax laws, and multiple shipping/customs rules. It's a whole new ball game folks, and we need to step up to the plate right now.

## Definitions

**Electronic cash (e-cash)** – A digital currency used on the Internet to buy and sell products.

**Electronic commerce (e-commerce)** – Any form of commerce in which the buyer of a product or service uses a computer to interact with the computer system of the seller of that product or service.

**Internet** – A collection of interconnected computers that people can log onto to share information, to communicate, to be entertained, and to perform electronic commerce transactions.

**Intranet** – A network internal to your organization that is built either partially or completely from Internet-based technology.

**World Wide Web (WWW)** – A component of the Internet that provides users with the ability to move from computer system to computer system by following predefined links among those systems.

## 2. Payment Processing on the Internet

Without a doubt accepting multiple currencies is the easiest issue to deal with when doing business electronically. The American dollar isn't the only currency in the world, and it isn't unreasonable to expect that people will want to do business in German marks, English pounds, French francs, and even Canadian dollars. The easiest solution to this issue is to quote prices on the net in your currency of choice, probably US dollars, and then convert to the other currency at the time of the sale.

**You have to accept multiple currencies.**

In addition to various types of currency you should also be able to accept multiple payment types. Credit-card transactions are fairly easy to process electronically and have the advantage that credit-card processors will do the currency exchange for you automatically. The big issue with credit cards is security, but by the time this white paper goes to print the major credit-card companies should have addressed this issue.

**Credit-card transactions will prove popular in the short term.**

Credit cards, however, aren't the only game in town. Just as I can go down to the grocery store and use my bank debit card to buy food, I should also be able to use it to purchase products on the net. Processing bank debit cards is a little harder than processing credit cards, but as the American financial industry consolidates itself over the next few years this will become less and less of a problem.

**Bank debit cards will grow in importance over time.**

I also see wire transfers being a significant payment type for electronic commerce, especially for large orders between companies. I hope financial institutions will be able to get a mechanism in place to take advantage of the huge opportunity presented by the Internet. In addition to wire transfers I also see electronic purchase orders, promises by known customers to pay later, as a popular payment mechanism.

**Electronic purchase orders will also prove popular.**

You should expect to see electronic/digital cash (e-cash) evolve over time, ultimately becoming the dominant payment type on the net. The main issue is one of acceptance, and I fear that it will take a generation or two for e-cash to truly catch on. Just as it was difficult centuries ago to convince people to start trading their cows and chickens for these new things called coins, it is just as difficult today to convince people to trade their

**E-cash will potentially dominate e-commerce in the long run.**

hard-earned money for a collection of bits called e-cash.

### 3. It's the WORLD Wide Web – International Issues

Language is another issue that is fairly easy to get our minds around – everyone doesn't read, write, and speak English. Although English is arguably the most popular language in the world, it isn't universal. German is quickly becoming the language of business in Europe, in large part because Germany is much more proactive in integrating Eastern European countries than are its trading partners. Spanish and Portuguese are the languages of business in South America, with Japanese and various dialects of Chinese used in most of Asia.

The point to be made is that if you truly want to do business on the web you're going to have to be able to support several languages. To do this you need a robust user interface that is easy to modify, a requirement that is well-supported by the class-type architecture presented in *Building Object Applications That Work* (Scott Ambler, SIGS Books 1997). Having an interface class layer puts you into a position to support several languages in your user interface without forcing you to change your underlying business classes.

**You'll need to support multiple languages.**

Another issue that needs to be considered when doing business on the web is culture. There are thousands of unique cultures around the world, each of which has its own ways of doing business and its own taboos. Did you know that in some Middle Eastern countries red is a forbidden color? Did you know that color is spelled "colour" in Canada and England? Make a small error in the design of your user interface and you risk offending millions of potential customers. The moral of the story is that if you want to be able to effectively do business in these countries you had better take the time to understand them and learn their unique nuances.

**You need to understand the culture of the people you are trying to sell to.**

A very important point to be made is that internationalization is more than a technical issue – Just because you use the Unicode character set in your application doesn't mean that you're ready for prime time.

### 4. You Can Sell Both Physical and Virtual Products Internationally

Electronic commerce also opens up opportunities for selling virtual products – software, online newspapers and magazines, and electronic libraries to name a few. Unlike selling physical products such as widgets to customers you'll find that selling virtual products is a significantly different endeavor. How do you price a virtual product? How do you ensure that it isn't copied and resold unbeknownst to you (should you even care)? How do you predict how many copies you'll sell? Interesting issues that we simply don't have answers to yet.

You also need to address the issue of shipping and customs – what can you sell to Cubans? How about Russians? Japanese? Canadians? Brazilians? Americans? Depending on the nation that your customers live in you are able to sell and ship different products to them as defined by the trade agreements your country has with theirs. Forget the complexities of understanding the ever-changing trade agreements between nations, how do you even know what country a person is in? How do you know you aren't illegally selling a virtual product to someone? For example, what's stopping a Cuban from getting an Internet address through a Canadian provider and purchasing virtual products from an American

company? You'd never know you committed a felony until the police came knocking on your door. Don't underestimate shipping and customs issues.

## 5. Don't Forget Taxes

As a Canadian I'm probably a little more aware of taxes than a lot of other people, and without a doubt this is to my detriment. Taxes on electronic commerce are going to be a huge issue, one that probably won't be resolved quickly. If an American company sells a product to an Italian consumer via an electronic order made on a web server they operate in Germany, what taxes get applied? American, German, or Italian? What rates do we charge? Who do we submit them to?

Taxes are collected at international, federal, state, and municipal levels. Tax rules are incredibly complex and constantly changing. Can you imagine trying to keep track of tax rules from thousands of countries, states, and cities? The point to be made is that tax collection is going to be a huge issue for anyone wanting to do business on the web.

As more and more business is done on the Internet I suspect that online tax-calculation services that organizations can access for a per transaction fee will do very well. Calculation and remittance of sales taxes on an international scale are far too complex for most organizations to handle, although will prove to be a lucrative business for anyone who wants to provide these services online.

**Online tax services on the Internet that encapsulate the complexities of tax calculation and remittance should do very well.**

At the present time the United States is pushing to declare the Internet a "duty-free" zone and have every country agree to this. Personally, I wouldn't hold my breath. It sounds like a good idea, but the potential for e-commerce is too large for most nations not to impose a tax on it, not to mention every state/province and municipality. I fully support the idea of a duty-free Internet, but I simply don't see it happening. I hope that I'm wrong about this.

## 6. And The Good News Is...

You don't have to tackle the world all at once, in fact you can start out small and then build from there. At a minimum you need to be able to process credit card transactions, support American English, and collect applicable Federal and State sales taxes. Many organizations start by giving access to their "best" customers at first and then roll it out from there. The road to long-term success with e-commerce is to start small but think big.

## 7. Using Smalltalk and Java to Develop Electronic Commerce Applications

It is clear that supporting electronic commerce, the issues of which are summarized in the table below, will prove to be more difficult than many people think. I believe that to successfully do so your organization will need to develop a system architecture that is robust enough to deal with the constantly changing complexities of international electronic commerce. It isn't bad enough that you need to support multiple languages, multiple currencies and multiple payment mechanisms but you also need to support tax laws and trade agreements that change daily. To put it bluntly doing business on the web is a complex and constantly changing environment. Oh, and by the way, now that you're doing business

internationally you now have six billion potential customers, so expect to receive a few more orders than usual once you're up and running on the web.

At the time of this writing it is realistic to expect that Java will be used to create relatively simple applets that collect basic information from your customers, package the information into a transaction, and then send the transaction through your firewall to be processed by the applications within your organization. Although simple, Java applets are the key to doing business on the Internet.

Applets will be simple for several reasons: First, Java is still immature as a language and still needs to grow before it can be used for complex applications. Second, the bandwidth on the Internet is limited and that effectively limits the size of the Java applets that you can reasonably transmit. Third, you want to limit the amount of information that you reveal to your competition on the Internet. Few organizations today hand their mission-critical applications over to their competition and it won't be any different with Java on the Internet (remember, anybody can go to your web page, even your competition). The short story is that you'll provide basic editing and validation logic in your applets, as well as whatever logic is needed to package the data into a packet to be transmitted through your firewall. Anything more and you're playing with fire.

If your Java applets don't encapsulate the complexities of doing business on the web that we discussed earlier, then where does all of that go? The answer is that it belongs on your side of the firewall. The basic idea is that your Java applets create and transmit standard transactions to your mission-critical applications and are processed there. You handle the complexity of electronic commerce within your internal applications where you can control and maintain it. As shown in Figure 2, Smalltalk is probably your best choice for creating the mission-critical applications that you need to support the needs of electronic commerce.

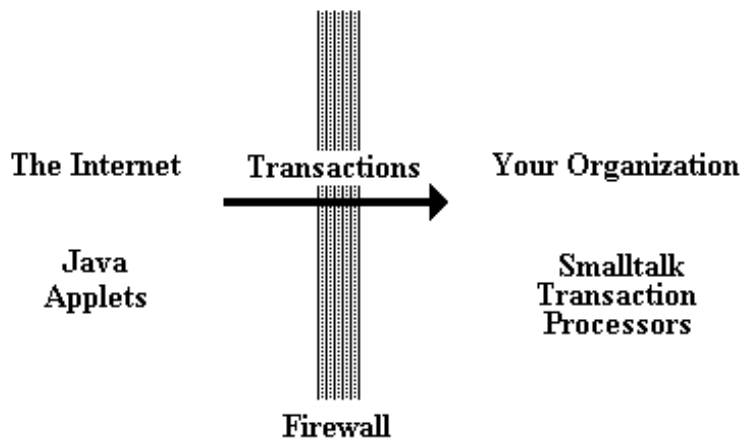


Figure 2. Using Java and Smalltalk together to meet the needs of electronic commerce.

## 8. Comparing Java and Smalltalk

Java applets provide a small, secure mechanism that run on a wide variety of client machines, making it the ideal language for implementing interface logic. However, at the time of this writing Java still suffers from run-time performance problems and it is still maturing as a language. One only has to look at the issues involved with porting from the Java Development Kit (JDK) 1.0.2 to 1.1 to see what I mean. Smalltalk on the other hand is a mature, proven language for mission-critical applications. Its high-level nature makes it a much more productive environment to develop in than Java or C++, and contrary to

popular opinion Smalltalk runs quite fast. Java and Smalltalk in combination make a formidable toolset that is ideal to handle the complexities of e-commerce. In Table 1 the strengths and weaknesses of each language are summarized.

<b>The Strengths and Weaknesses of Each Language</b>	
<b>Java</b>	<b>Smalltalk</b>
<ul style="list-style-type: none"> <li>• Multi-platform support (ideal for client machines)</li> <li>• Small, secure applets</li> <li>• Well supported by the industry</li> <li>• Multi-vendor</li> <li>• Tools/libraries still maturing</li> <li>• Runtime performance is still slow but improving</li> </ul>	<ul style="list-style-type: none"> <li>• High-level language ideal for business logic</li> <li>• Mature and proven</li> <li>• Robust toolset</li> <li>• Multi-vendor</li> <li>• Runtime performance near that of C++</li> <li>• Much higher developer productivity compared to C++ or Java</li> </ul>

**Table 1: Comparing Java and Smalltalk.**

## 9. In Summary

Many organizations are trying to either modify their existing legacy applications, or are trying to put an object-oriented wrapper around them so that newer programs can access them. Although some of these efforts will prove to be successful, I wouldn't put any money on them. Think of it like this: Do your old applications truly support the needs of electronic commerce, as summarized in Table 2? Probably not. Just like you can't create a silk purse out of a sow's ear you can't create a modern application out of the ancient legacy code of yesteryear. If you want to do business on the web you're going to have to rewrite your mission-critical applications using a combination of Java and Smalltalk it's as simple as that.

<p>Have you taken into consideration being able to</p> <ul style="list-style-type: none"> <li>• Process multiple currencies?</li> <li>• Process multiple payment types?                             <ul style="list-style-type: none"> <li>• Credit/debit cards</li> <li>• Wire transfers</li> <li>• Electronic/digital cash</li> <li>• Electronic purchase orders</li> </ul> </li> <li>• Do business in several languages?</li> <li>• Support various cultures and beliefs?</li> <li>• Identify what you can and cannot sell to people?</li> <li>• Support sale of both physical and virtual products?</li> <li>• Collect and submit the appropriate federal, state, and municipal taxes?</li> <li>• Limit the information you reveal to your competition?</li> <li>• Process orders made by six billion potential customers?</li> <li>• Leverage existing legacy applications?</li> </ul>
---

**Table 2: Electronic Commerce Development Checklist**



## 10. About the Author

Scott W. Ambler is Software Process Mentor living in Newmarket, Ontario, which is 45 km north of Toronto, Canada. Scott is the author of *The Object Primer* (1995), *Building Object Applications That Work* (1998), *Process Patterns* (1998) and *More Process Patterns* (1999) all published by SIGS Books/Cambridge University Press. He has worked with OO technology since 1990 in various roles: Process Mentor, Business Architect, System Analyst, System Designer, Project Manager, Smalltalk Programmer, Java Programmer, and C++ Programmer. He has also been active in education and training as both a formal trainer and as an object mentor. Scott is a contributing editor with *Software Development* (<http://www.sdmagazine.com>) and writes feature articles for *Component Strategies* (<http://www.sigs.com>) and *Computing Canada* (<http://www.plesman.com>). He can be reached via e-mail at [scott@ambysoft.com](mailto:scott@ambysoft.com) and you can visit his personal web site <http://www.ambysoft.com>.

### About *The Object Primer*

*The Object Primer* is a straightforward, easy to understand introduction to object-oriented analysis and design techniques. Object-orientation is the most important change to system development since the advent of structured methods. While OO is often used to develop complex systems, OO itself does not need to be complicated. This book is different than any other book ever written about object-orientation (OO) – It's written from the point of view of a real-world developer, somebody who has lived through the difficulty of learning this exciting new approach. Readers of *The Object Primer* have found it to be one of the easiest introductory books in OO development on the market today, many of whom have shared their comments and kudos with me. Topics include CRC modeling, use cases, use-case scenario testing, and class diagramming. Visit <http://www.ambysoft.com/theObjectPrimer.html> for more details.

### About *Building Object Applications That Work*

*Building Object Applications That Work* is about: **architecting** your applications so that they're maintainable and extensible; analysis and design techniques using the Unified Modeling Language (UML); creating applications for stand-alone, client/server, and distributed environments; using both relational and object-oriented (OO) databases for persistence; OO metrics; applying OO patterns to improve the quality of your applications; OO testing (it's harder, not easier); user interface design so your users can actually work with the systems that you build; and coding applications in a way that makes them maintainable and extensible. Visit <http://www.ambysoft.com/buildingObjectApplications.html> for more details.

Uses the  
  
 Unified  
 Modeling  
 Language

### About *Process Patterns* and *More Process Patterns*

*Process Patterns* and *More Process Patterns* are ground-breaking texts, describing proven, reusable techniques for developing large-scale, mission-critical object-oriented software that is robust and extensible. The focus of the book is The Object-Oriented Software Process (OOSP), presented as a collection of process patterns that are geared toward medium to large-size organizations that need to develop software that support their main line of business. Process patterns are the reusable building blocks from which your organization will develop a tailored software process that meets its exact needs, and have been shown to be ideal for enhancing the industry-standard Unified Process. Visit <http://www.ambysoft.com/processPatterns.html> and <http://www.ambysoft.com/moreProcessPatterns.html> for more details.

Uses the  
  
 Unified  
 Modeling  
 Language

### About *The AmbySoft Inc. Coding Standards for Java*

*The AmbySoft Inc. Coding Standards for Java* summarizes in one place the common coding standards for Java, as well as presents several guidelines for improving the quality of your code. It is in Adobe PDF format and can be downloaded from <http://www.ambysoft.com/javaCodingStandards.html>.

**Index**

<b>A</b>	
Applet.....	5
Architecture	
for electronic commerce .....	1, 6
Author	
contacting .....	7
<b>D</b>	
Debit card .....	2
Duty-free zone.....	4
<b>E</b>	
E-cash... <i>See</i> Electronic cash. <i>See</i> Electronic cash	
E-commerce..... <i>See</i> Electronic commerce	
Electronic cash.....	2, 3
Electronic commerce.....	1
architecture .....	1, 6
definition.....	2
international issues.....	3
payment processing .....	2
taxes.....	4
virtual products .....	3
<b>I</b>	
International issues .....	3
Internet .....	1
definition.....	2
Intranet	
definition.....	2
<b>J</b>	
Java	
and electronic commerce .....	4
vs. Smalltalk.....	5
Java applet.....	5
<b>P</b>	
Payment processing .....	2
<b>S</b>	
Scott Ambler	
contacting.....	7
Smalltalk	
and electronic commerce .....	4
vs. Java.....	5
Start small, think big.....	4
<b>T</b>	
Taxes.....	4
<b>U</b>	
UNICODE.....	3
<b>V</b>	
Virtual products.....	3
<b>W</b>	
Wire transfer .....	2
World-wide web	
definition .....	2
electronic commerce .....	1
WWW .....	<i>See</i> World-wide web