# A Formal Model of Peer-to-Peer Digital Product Marketplace

**Sylvanus A. Ehikioya**[1]

*[1] Department of Computer Science, Baze University,*
*Abuja, NIGERIA*
*E-mail: ehikioya@gmail.com*

## Abstract

Peer-to-peer (P2P) electronic commerce has become increasingly popular over the last few years. This popularity of P2P is due mainly to the many advantages it offers, such as the ease of sharing of computer resources and reducing cost. This paper uses both semi-formal and formal specification methods to model a P2P digital product marketplace. First, models are written in UML, as this is expected to be relatively easy to understand and economic. Then, the models are transformed into a formal notation, object-Z notation. A formally specified and verified model of e-commerce transactions will provide a correct, fail-safe, and robust transaction processing environment for e-commerce. By using both semi-formal and formal methods in the design of the P2P system, we can reason about and create a correct and reliable system quickly and efficiently.

*Keywords*: Electronic commerce, Digital product, Formal model, Object-Z, Transactions, UML.

## 1. Introduction

With the development of the Internet and the continual growing sophistication of Internet services, electronic commerce (e-commerce) has become a new model of transacting business activities. The acceptance of the Internet as a medium for commerce is also increasing at a phenomenal rate so much that many people are more willing to do business using the Internet than ever before. The explosive growth in the adoption of e-commerce (and will continue to explode in the next few years) makes it fast becoming a trillion dollar market[1]. The prospect of e-commerce is very attractive. E-commerce is no longer the preserve of large companies. Small organizations, and even individuals, now take advantage of the ubiquity and pervasiveness of the Internet and e-commerce technologies to engage in business activities. Generally, e-commerce can be classified mainly into business-to-consumer (B2C), consumer-to-consumer (C2C), and business-to-business (B2B) models. Ehikioya and Hiebert[2] examined these models in detail. Other emergent models of e-commerce such as the government-to-government (G2G) and citizen-to-government (C2G), which are similar to the B2B and B2C respectively, and auctions are becoming popular.

E-commerce began with B2C transactions in 1995. B2C e-commerce involves transactions between a consumer and an online retail establishment. C2C e-commerce has also emerged, allowing unknown, un-trusted parties to sell goods and services to one another. In 1998, B2B e-commerce, the term used to refer to transactions between businesses conducted online, appeared and became more dominant in e-commerce.

Peer-to-peer (P2P) e-commerce has become increasingly popular over the last three years. P2P is defined as the sharing of computer resources and services by direct exchange between computers in a distributed environment that does not use a centralized control, allowing ordinary machines to perform server functions. A P2P business is one where consumers can buy and sell products and services to one another. It is a symbiotic relationship where consumers share value creation. In fact, P2P e-commerce is a form of C2C model of e-commerce. The music service provider Napster made file sharing the most well-known P2P application class. Napster attracted 38 million users within one year.

P2P e-commerce represents a technological feature in e-commerce rather than an interaction between participants. P2P B2B e-commerce or P2P B2C e-commerce is possible, in which B2B / B2C represents the participants of e-commerce. P2P e-commerce applications are characterized by autonomy of each peer; peers communicate and transact business with unspecified and often un-trusted peers; and the existence of commercial transaction mechanism to facilitate commerce. The following additional features[3] characterize P2P e-commerce applications: distribution, decentralization, self-composing and individualization, and symmetric communication by majority of the peers.

Recently, P2P-based publication and subscription systems appeared. The publish - subscribe system is a communication paradigm that supports many-to-many communication in a distributed environment, and it is often implemented on P2P infrastructure that enables information dissemination from publishers to subscribers. Liu, et al.[4] and Wang, et al.[5] discuss subscription mechanisms and the relevant security issues in the model.

P2P e-commerce takes advantage of resources available at the edge of the Internet. The edges can have the commercial resources such as physical products and their catalogs, digital products, computing powers, or storages. In this paper, we focus on P2P e-commerce on digital products marketplace. P2P digital products marketplace allows people use client software on their own server (could be common PC) to share, exchange, sell, and buy digital products and services with other peers who use the same client software. Digital products include music, movies, software, and digital documents including information products, and services. A digital product is characterized by the following properties[6]:

- Indestructibility: Once created, a digital product maintains its form and quality ad infinitum because of the lack of normal wear and tear.
- Transmutability: The content of digital products can be changed easily and instantly. They are extremely customizable and changes, whether accidental, intended, or fraudulent, can be irreversible. Thus, producers lose some control over the integrity of their products.
- Reproducibility: They can be reproduced, stored, and transferred at ease.

Additional common characteristics of all digital products are immateriality, borderless, interoperability, and right of withdrawal.

These properties lend P2P digital products marketplace to a broad set of new and emergent challenges. For example, integrity and security issues, intellectual property and copyright infringement issues, etc of digital products is a commonplace. Also, note that once a digital product is introduced as an inventory item, it remains there until it is explicitly removed from the system. There is no concept of out-of-stock with digital products. The time utility of the product may degrade over time (particularly in respect of knowledge-based / informational digital products). However, whenever a consumer needs more than the one copy downloaded, the digital product can easily be reproduced up to the number of licenses paid for. Digital products take the indestructibility, transmutability, and reproducibility properties to an infinite degree. A taxonomy of the different types of digital products and services (information and entertainment products; symbols, tokens and concepts; and processes and services) is available in the literature[6] and will not be discussed here in order to stay within the main focus of this paper.

This paper presents a formal model of a P2P digital product marketplace using both semi-formal and formal specification methods. First, models are written in unified modeling language (UML)[7], as this is expected to be relatively easy to understand and economic. Then, the models are transformed into a formal notation, Object-Z notation[8], which is formally verifiable. The combined design provides a correct, fail-safe, and robust transaction model for e-commerce.

The motivation for using both semi-formal and formal specification methods stems from the fact that most of the e-commerce applications developed and deployed are mostly based on ad hoc, intuitive, and informal development approaches that do not lend themselves to formal reasoning nor has any proof mandate / obligation[9]. This observation, although true over a decade ago, is still true today as most designers do not use formal specification methods. Although these ad hoc techniques / approaches enjoy widespread industry support because of their simplicity and ease of understanding, the determination of correctness of the requirements and design specifications are, however, based on human intuition. Thus, the requirements and design specifications are characterised by unproven artifacts, ambiguity, inconsistencies, and sometimes incorrect executions. The consequence is that this problem often permeates the final products such that they deviate from the initial requirements specifications and manifest unexpected failures during actual use. This is because informal methods do not explicitly capture the dynamic behaviour of e-commerce transactions, which is critical for their development, especially for the design phase. It is necessary, therefore, to model the dynamic characteristics of e-commerce transactions using formal specification approach in order to specify and verify the system behaviour. Our approach offers the following key benefits:

- The graphical nature of the UML diagrams can help in the easy visualization of the system and its components. The UML-based specification is easy to understand by potential users and domain experts, thereby providing opportunities for detailed discussions, which can identify ambiguous, incomplete, or inconsistent requirements and provides the basis for a detailed system description that can be used to generate a system implementation, and thus models have to be both correct and complete. Also, since UML is used as a tool for capturing, simplifying and specifying system requirements, therefore, the models have to be simple and easily understood to ensure correctness and completeness.
- Management of complexities of behaviours of the e-commerce system through their visual models. The use case analysis and the dynamic views in UML (e.g., statechart, collaboration, and activity diagrams) focus on the system's functional view. This permits the incremental study and development of the entire functionality of the system. This in turn supports collaborative development among the development team members.
- Separation of capabilities and using the most suitable aspects of a method to model the different aspects of the P2P model. For example, using Object-Z to model the pre-and post- conditions for the operations while using UML to capture the dynamic views of the system.

The contributions of this paper are:

- The Web-based P2P digital product marketplace provides an extensible infrastructure to e-commerce model. Although our sample specification appears simplistic, the system provides the opportunities to reason about specifying and designing a novel but unique e-commerce application using both semi-formal and formal notations, thereby intuitively and inherently providing and enforcing a correct, fail-safe, unambiguous, and robust transaction processing environment for e-commerce.
- It provides a practical case study for designing an e-commerce system while gaining skills on the application of formal notation to a realistic problem. The basic principles and thrust on which our design is anchored is applicable to solving real-world problems and provides a stimulating learning experience to other system developers.
- The implementation of our design provides a test-bed platform / environment for learning and gaining practical experience in Web applications architecture (software) design and implementation.

This paper is significant for the following reasons:

- First, by documenting the artifacts of the system, at different levels of abstractions and using complementary notations (Object-Z for the concrete definitions of objects and calculations of the pre- and post- conditions of operations, and UML diagrams to capture the relationships between, and interactions among, components), it makes the understanding of the relevant elements in e-commerce application clearer.
- Second, a formally specified and verified model of e-commerce transactions will provide a correct, fail-safe, and robust transaction-processing environment for e-commerce.

The rest of this paper is organized as follows: Section 2 briefly examines some of the key related research work in the literature. Section 3 describes the design of our model architecture of the P2P digital products marketplace and examines the key elements of the architecture suitable for e-commerce systems, which offers the critical link to understanding of the unique behaviour of the P2P digital products marketplace. Finally, in Section 4 we present our conclusions and briefly discuss our future work.

## 2. Background Literature

E-commerce has provided consumers with more options and opportunities than ever before. With online businesses developing very fast and available almost everywhere, consumers now have the ability to purchase goods and services using their own personal computer, or other mobile devices like cellular phones and PDAs. Generally, e-commerce is considered as the transactions between businesses and consumers. However, one of the most innovative developments in e-commerce is the rapidly growing P2P e-commerce market.

### *2.1. P2P Computing Paradigms*

A plethora of P2P applications have been developed in recent years, such as Gnutella, ICQ, Napster, KaZaA, and MSN Messenger. Other examples of P2P platforms are Airbnb, Aristotle Circle, eBay, Match.com, Uber, Lyft, and Zopa. Also, some famous IT companies have participated in developing P2P software. For example, Lyft[10], a ride-sharing program, launched in June of 2012 in San Francisco, is one of the latest in P2P transportation similar in services and concept to the popular Uber service. It matches individuals with vehicles and extra time to drive them (called the Drivers) with others in their areas that need rides (called the Riders).

P2P e-commerce is developed from peer-to-peer computing. The architecture of the P2P is contrasted with client-server architecture. P2P systems are built up around a collection of peers that are networked together in some fashion. Often, these peers are personal computers. Figure 1 shows the P2P conceptual model.

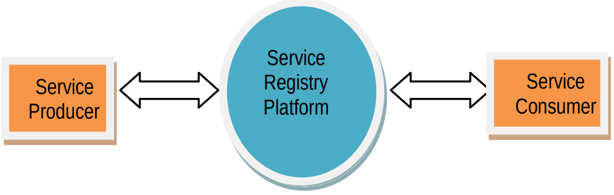


Fig. 1. A Conceptual Model of P2P System

The individual consumers provide the services and products offering on the platform for the consumption of the participant individual consumers. The service registry platform provides mechanism for consumers' interaction, match making, and arbitration ensuring safety of transactions, standards of quality, brand promise, protocols / rules, payments accounting, etc.

Our system architecture draws from the knowledge and theories of service oriented architecture (see Figure 2) well established in the literature[11, 12], whereby services can be provided locally or outsourced to external service providers. Ehikioya[13] uses this methodology to model real-time multi-agents interactions for e-commerce. Our model uses the simple object access protocol (SOAP) standard (a message exchange standard that supports service communication) and web service definition language (WSDL) standard that allows a service interface and its binding definitions. The binding maps the abstract interface to a set of protocols that specifies how (or rules) to communicate with a web service, a fundamental ingredient of web-based applications, including e-commerce systems.

In most cases, this request-and-reply communication is not done directly between requestConsumer and serviceConsumer. A broker is introduced to deal with the multiple requests and multiple services. The broker is a particular central peer entity that stores the index of all the resources for information (or IP address) of other peers holding a particular resource.

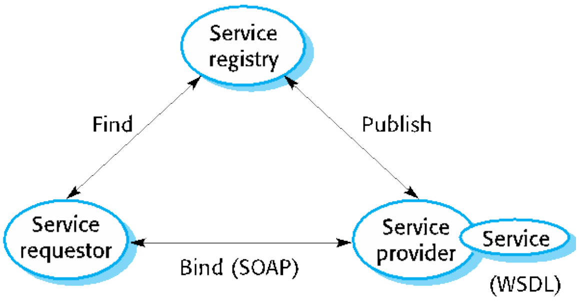
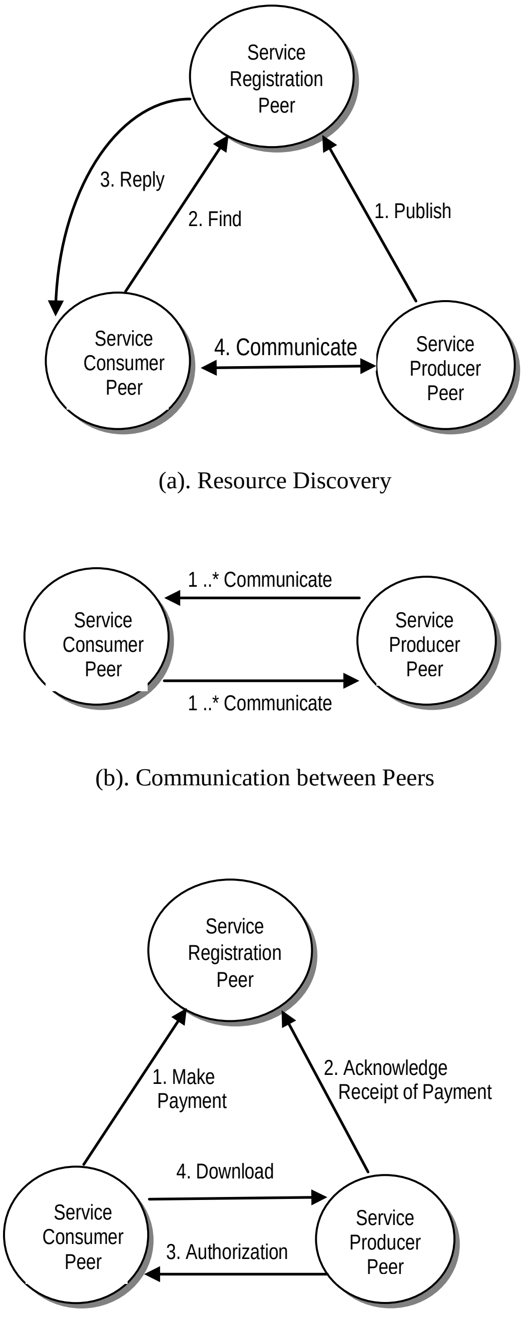


Fig. 2. Service Oriented Architecture[11]

At the beginning (see Figure 3(a)), every serviceConsumer registers its available digital products and services including name, OS, reliability, QoS, available time, etc. with a broker (Service Registry Platform in Figure 1). When requestConsumer needs a product or service (a task with some parameters (intended OS, reliability, etc.) to be executed), it will not send this request directly to multiple serviceConsumer, instead, it sends its request to the broker. The broker matches requestConsumer's request with serviceConsumer's registration information, and if it finds matched pair, it will forward the appropriate serviceConsumer's details to the requestConsumer. Thereafter, the requestConsumer could communicate with the selected serviceConsumer directly without going through the broker as shown in Figure 3(b). The communication between a connected pair of peers continues until disconnected. Figure 3(c) shows the procedure when a requestConsumer peer consumes a digital product or services offered by a serviceConsumer peer. Figure 3 encapsulates the interaction / communication procedure among the peers.

The introduction of the broker simplifies service peer discovery and the communication between requestConsumer and serviceConsumer, and reduces communication time and overhead. For a detailed examination of transactions in e-commerce environments, interested readers should see the following resources[14, 15, 16].

Currently, two types of P2P logical system architecture exists: decentralised and semi-centralised. In a decentralized P2P architecture, each peer within the architecture has equal status and no peer acts as a control node; while in semi-centralised architecture, there exists at least one control peer that performs an authoritative role within the network. Walkerdine et al.[17, 18] illustrate seven possible P2P logical architectures and these will not be repeated here.



(a). Resource Discovery

(b). Communication between Peers

(c). Purchase Transaction between Peers

Fig. 3: Interaction / Communication among Peers

However, it suffices to note that P2P architectures are mainly classified according to how information is routed within the network and how peers are organized.

The P2P architecture has been used extensively in several applications, such as energy trading and sharing[19], alternative funding financial services[20], media streaming[21], mobile networks[22, 23] and social networks[24, 25], because P2P systems are self-organized and distributed in nature in addition to their capability to store and retrieve information [without necessarily the need to have a central entity] in the discovery of peers and information exchange within the network. A P2P approach to digital products and services commerce and sharing promotes availability of digital products and services in the P2P community, and increases efficiency, flexibility and effectiveness in digital products transactions.

We note that in a decentralized architecture, service discovery is difficult and it often relies on perfect knowledge of the existence and location of a digital product / service. In addition, both peers may be untrusted, and transaction is dependent on the honesty of the participating peers. These challenges / problems make this model unattractive and unsuitable for e-commerce transactions. We adopt a semi-centralized architecture with a single centralized index server (Service Registry Platform) logical network architecture because of the high premium placed on safety, maintainability, and trust and accountability of transactions in the P2P e-commerce environment. Service discovery is fast and it is provided by the broker. These requirements outweigh the single point of failure which can greatly hinders the system's fault tolerance and survivability.

### *2.2. E-Commerce Applications*

There are many e-commerce applications on the Internet; however, most of them developed and deployed are based on ad hoc, intuitive, and informal development approaches[9]. The existing e-commerce system mostly offer a computer to computer communication based on common gateway interface (CGI) technology with proprietary formats and rules, and support special domains of interest with special features and possibilities. These systems support traditional business activities such as ordering, delivering, and financing.

Many researchers develop new technologies and methods on how to design these e-commerce systems. Some of them use some semi-formal specification such as UML, and formal specification such as algebraic signatures[26, 27] and Z Notation[28, 29] to specify and design e-commerce systems[2, 30, 31]. By specifying the systems formally, e-commerce systems become easier to understand, specify and design. Various methods and techniques suitable for modeling e-commerce transactions have been examined in the literature [32, 33, 34, 35], and these will not be repeated here. Interested readers should refer the relevant resources. However, it suffices to note that most implementations of e-commerce systems adopt mostly informal or semi formal methods, such as UML.

### *2.3 Formal Methods*

All e-commerce systems, including peer-to-peer e-commerce systems, could be designed and created by using formal methods. Using formal methods can help to avoid ambiguities, incompleteness, inconsistency, and keep cost down[36]. We use both semi-formal notation such as the UML and formal notation such as the Object-Z for our specification and design.

#### *2.3.1 Unified Modeling Language in E-commerce Design*

The UML is an industry-standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. It has become the dominant modeling language in the industry. Although UML is a semi-formal language, it is also a standard notation for objected-oriented modeling. By using UML, complex e-commerce system development could reason about the system and all ambiguities removed thereby making them more efficient. In fact, UML provides facilities to model the external, structural, behavioural, and interaction perspectives of an e-commerce system. An external perspective models the context or environment of the system. A structural perspective represents the organization of a system or the structure of the data that is processed by the system. A behavioural perspective models the dynamic behavior of the system and how it responds to events. Interaction model is used to represent the interactions between a system and its environment, or between the components of a system. Use case diagrams and sequence diagrams may be used to model interactions. This perspective is important for a number of reasons. First, modeling user interactions is important as it helps to identify user requirements, while modeling system-to-system interaction highlights the communication problems that may arise. Second, modeling component interaction helps us understand if a proposed system structure is likely to deliver the required system performance and dependability.

The popularity of UML is growing by the day as many web designers, as well as e-commerce systems designers, use UML to model their system before development begins. It is an object-based methodology to assist in the mapping of functional requirements to design specification. It breaks the architecture of any software, including e-commerce systems, into modules and each module is made up of one or several "use cases". He[36] used this methodology in designing a web-based bazaar system. Also, UML provide some other diagrams like class diagrams, collaboration diagrams, activity and sequence diagrams, and state diagrams for the modeling of behavioural and interaction aspects of an e-commerce system. Class diagrams show the structure of a system, collaboration diagrams shows the interaction between objects in use case, and state diagrams describes the local behavior of classes.

#### *2.3.2 Object Z Specification*

Object Z, an objected-oriented extension of the Z notation, is an object-oriented formal specification language. It adds to the Z notation notions of classes and objects, and inheritance and polymorphism. By extending Z's semantic basis, it enables the specification of systems as collections of independent objects in which self and mutual referencing are possible[8].

An Object-Z specification consists of a collection of class definitions and some global definitions, and each class definition is an encapsulated Z specification. In Object-Z, instantiation is used to model relationships between objects, which is almost same as association in UML. Objects, which instantiate other classes as their attributes, can refer to the objects of the instantiated classes.

In this paper, we used Object-Z to model the peer-to-peer digital product marketplaces. Object-Z could make the design easier and more efficient, and enable the programmers and users better understand the system. Readers unfamiliar with the Object-Z notation should see[8] for a gentle introduction.

## 3. P2P Marketplace Formal Design

To participate in the P2P digital product marketplace, each peer should install a client program, and specify a folder, which contains the digital products and services for sale. Through this client, a peer can search, buy, and sell digital products and services.

### *3.1 UML Design*

In the P2P digital product marketplace, each peer can be a seller, or buyer. The relationships and functions of a peer-to-peer digital product marketplace are not as complex as those of transactional e-commerce systems. The functions permissible on a digital product include Search, Establish Connection, Order Products, Pay for Products, and Release Products. Figure 4 illustrates the use case relationships between peers and functions.

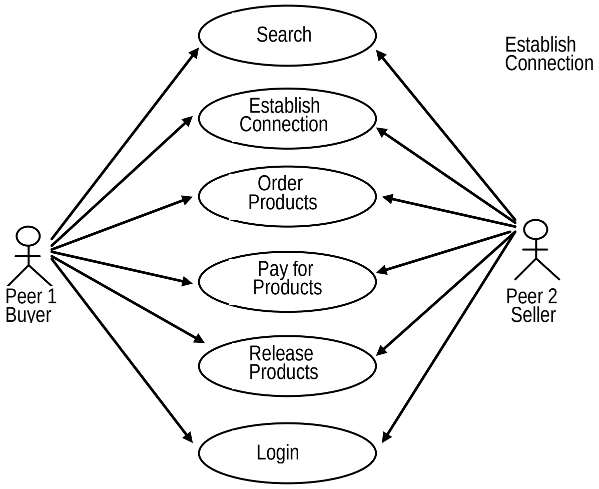


Fig. 4. Use Case Diagram of P2P Digital Product Marketplace

### Startup of Client Program

Every time a user (a peer) wants to do transaction with other peers, the user must start a client program, which is on own device / computer. A unique ID is required when a user starts the client program. In this system, we use e-mail address as ID, because every e-mail address is unique. The system has a check mechanism to ensure the correctness of the input e-mail address. However, the user should ensure the e-mail address is correct; otherwise the user will not receive any confirmation of order. Figure 5 shows the login operation.

### Search

After entering the system, a user can search for digital products on the computers of other active peers (i.e., peers that have started the same client program). Figure 6 shows the details of the *Search* method. A user can search for digital products [on the Internet] in several ways:

- Search by e-mail address. In this case, you should know the peer that has some products you want.
- Search by IP address. In this case, you also need to know the peer that has some products you want.
- Search by keywords. In this case, the system will return all the peers that have the digital products.

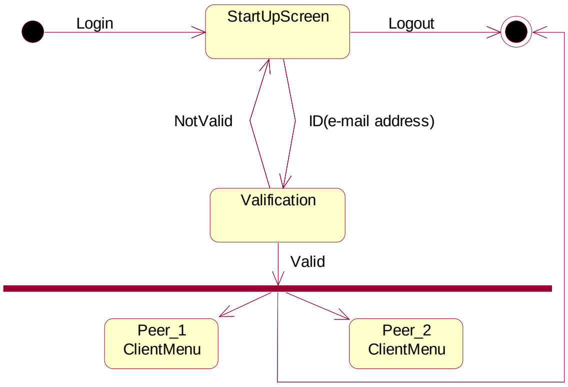


Fig. 5. Statechart for the Client Program Startup

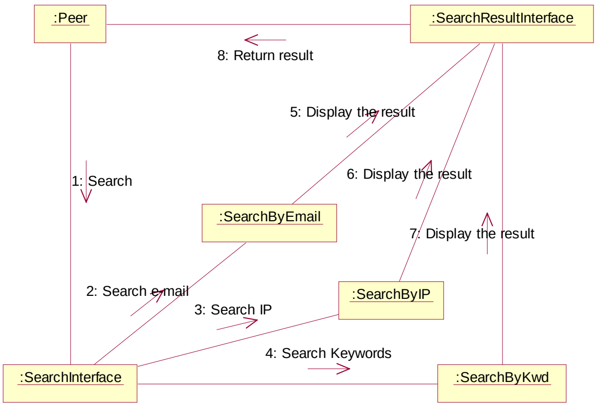


Fig. 6. Collaboration Diagram for Search

### Establish Connection

When a user finds another peer that has the digital products / services that the user wants, the requesting user peer establishes a secure connection with the service peer. At the same time, the bank verifies the bank account information or credit card information of both peers. Security issues are very important in this connection. However, these security issues are outside the scope of this paper, and will not be discussed here.

First, the bank verifies the bank account information or credit card information of the Peer 1 (buyer). Then, Peer 2 (seller)'s bank account information or credit card information is also verified, because the bank will transfer the money into the seller's account, if the transaction succeeds. If information of both peers is correct, a connection is established. Figure 7 describes the establishment of a connection in great detail.

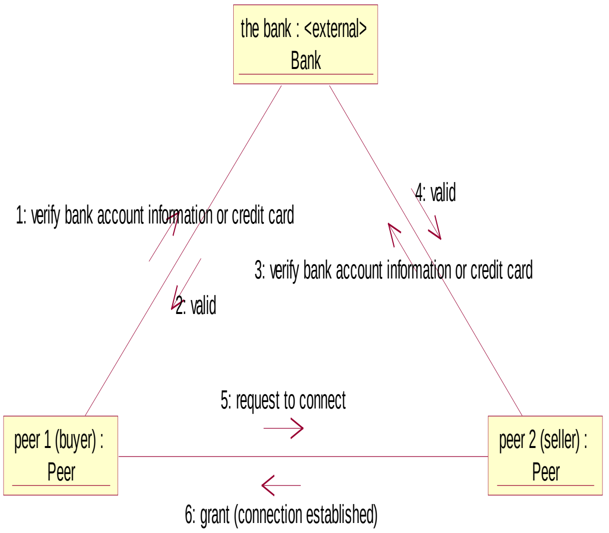

the bank : <external> Bank
1: verify bank account information or credit card
2: valid
3: verify bank account information or credit card
4: valid
5: request to connect
6: grant (connection established)
peer 1 (buyer) : Peer
peer 2 (seller) : Peer

Fig. 7. Collaboration Diagram for Connection Establishment

### Order Products

After the connection is established, the buyer can order digital products from the seller. The system allows a user to add order items, remove order items, cancel order, and confirm order. However, the system does not have the function to modify quantity of one kind of product, because digital products are different from other products and can be duplicated without any difficulty. Figure 8 shows the detail of order process.

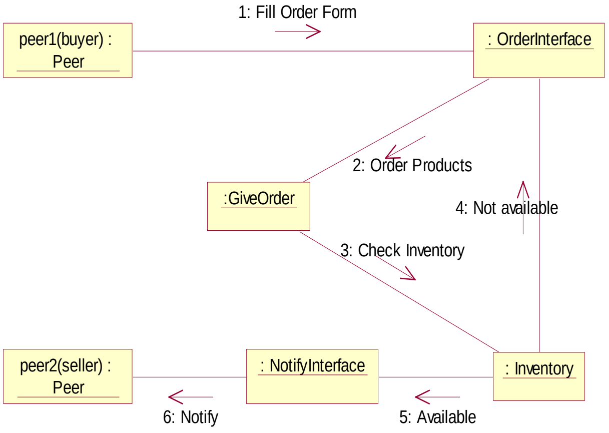


Fig. 8. Collaboration Diagram for Order Process

### Pay for Products

After an order is created, the buyer can select *cancel* the order or *confirm* the order and pay for the products ordered. The buyer fills out a form including payment amount, bank account number or credit card information, which are already verified by the bank in the peer connection process. The external system, the bank, will check the buyer’s account balance or credit limit of credit card to ensure it is enough to pay for the products. If the payment is authorized by the bank, the bank sends all the necessary information to the related seller and notifies the seller that the payment is OK. Figure 9 describes the details of paying for a digital product.

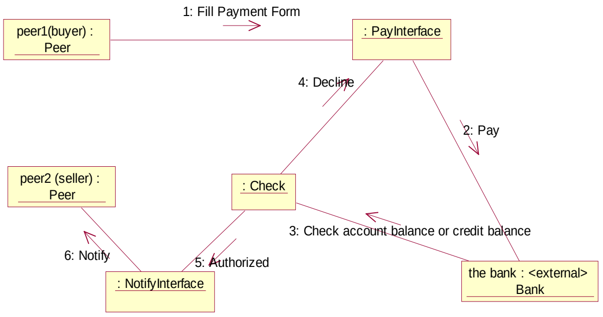


Fig. 9. Collaboration Diagram for Paying for Products

### Release Products

When the seller receives the payment confirmation from the bank, the seller releases the digital product, and allows the buyer to download. Figure 10 shows this process.

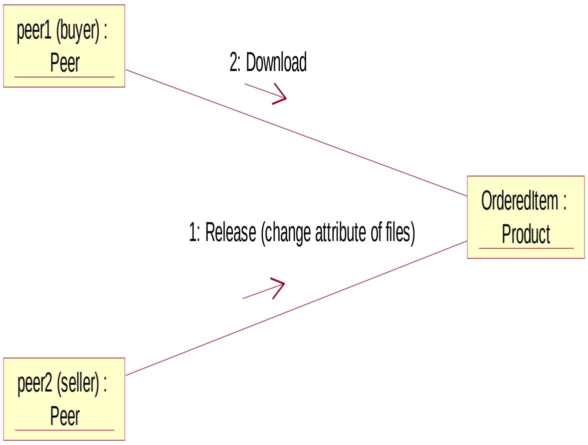


Fig. 10. Collaboration Diagram for Release Products

## *3.2 Object-Z specification*

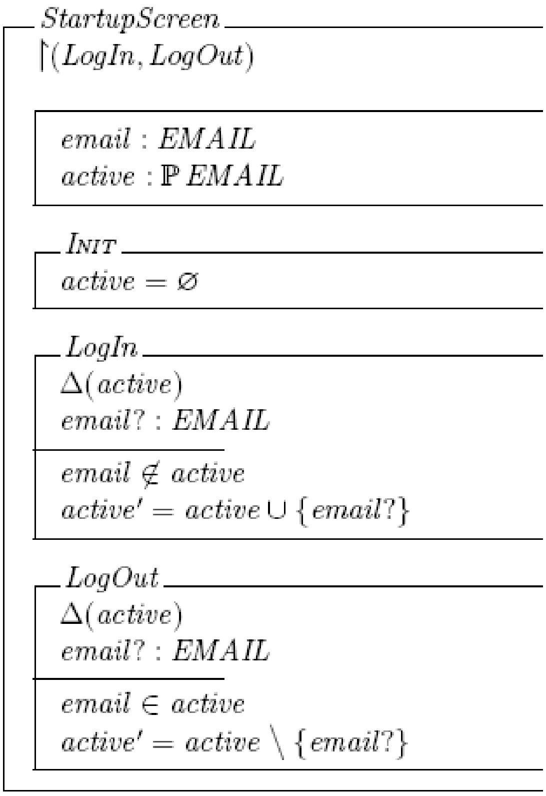
We provide the Object-Z specification of our P2P model. However, we annotate the specification, with clear and understandable commentary to assist the reader, in order to improve its understanding and readability.

**Startup of Client Program**

When the client program starts, the system requires the user to input a UserID, which must be a valid e-mail address. The system then checks if the input e-mail is correct, and check if other peers use the same e-mail address to login into the system. In the *StartupScreen* class, there are two operations (or methods):

- ***LogIn***: Before Login, the system should ensure that no other peers in the system have the same e-mail address in the active accounts.
- ***LogOut***: After Logout, the system should remove the e-mail address in the active accounts.

The following schema shows the formal specification of the class *StartupScreen*.



*StartupScreen*

$\upharpoonright(LogIn, LogOut)$

$email : EMAIL$
$active : \mathbb{P}\, EMAIL$

INIT

$active = \varnothing$

*LogIn*

$\Delta(active)$
$email? : EMAIL$

$email \notin active$
$active' = active \cup \{email?\}$

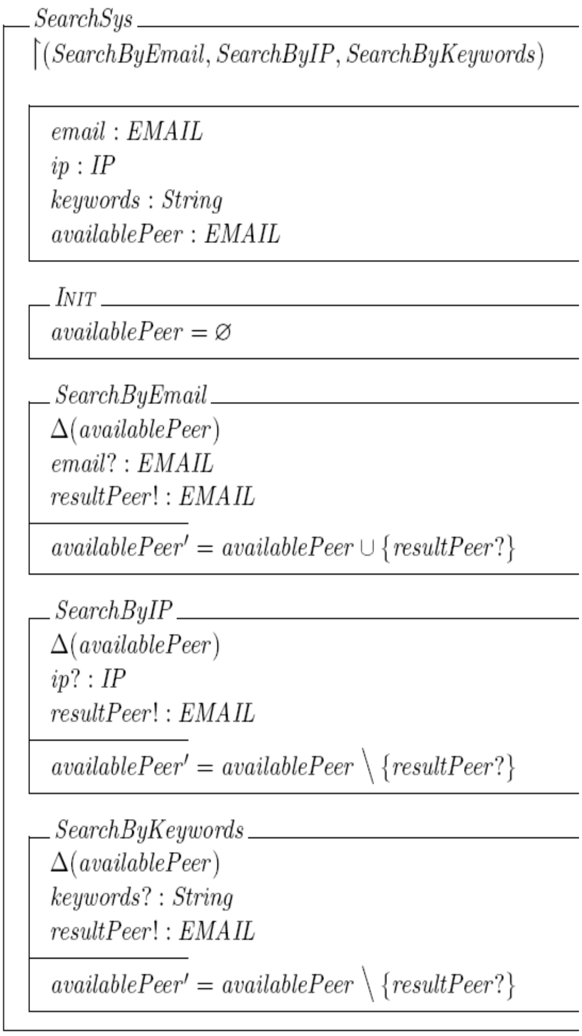
*LogOut*

$\Delta(active)$
$email? : EMAIL$

$email \in active$
$active' = active \setminus \{email?\}$

**Search**

There are three ways for a user to search digital products on the Internet, search by email address, search by IP address, and search by keywords. The following schema shows the formal definition of operations of class *SearchSys*.



*SearchSys*

$\upharpoonright(SearchByEmail, SearchByIP, SearchByKeywords)$

$email : EMAIL$
$ip : IP$
$keywords : String$
$availablePeer : EMAIL$

INIT

$availablePeer = \varnothing$

*SearchByEmail*

$\Delta(availablePeer)$
$email? : EMAIL$
$resultPeer! : EMAIL$

$availablePeer' = availablePeer \cup \{resultPeer?\}$

*SearchByIP*

$\Delta(availablePeer)$
$ip? : IP$
$resultPeer! : EMAIL$

$availablePeer' = availablePeer \setminus \{resultPeer?\}$

*SearchByKeywords*

$\Delta(availablePeer)$
$keywords? : String$
$resultPeer! : EMAIL$

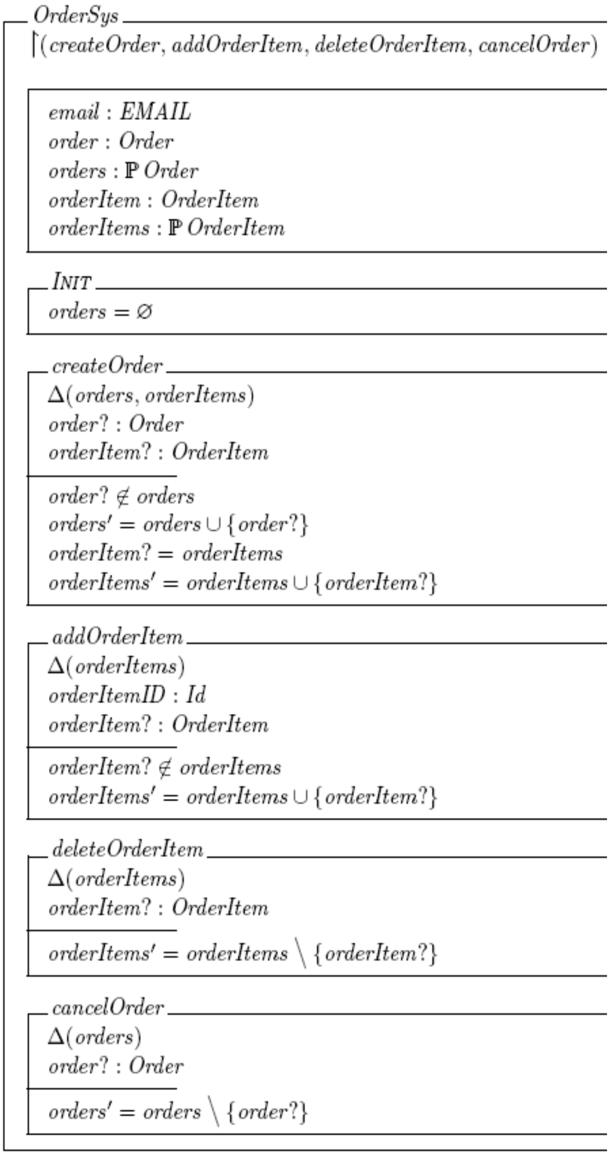
$availablePeer' = availablePeer \setminus \{resultPeer?\}$

**Order**

After the connection is established, the buyer can order digital products through a secure connection. The order subsystem has a class called *OrderSys*, and its sub-classes *Order* and *OrderItem*. The operations of the class *OrderSys* include *createOrder*, *addOrderItem*, *deleteOrderItem*, and *cancelOrder*. The specification of the class and its operations is shown in the following schema, *OrderSys*.

## 4. Conclusion and Future Work

P2P blends technology, innovation, and disruption to provide an improved individual consumers buying and selling experience with its characteristic lightweight online operating models. This paper formalizes requirements of peer-to-peer digital product marketplace using UML and Object-Z notations.



*OrderSys*

$\upharpoonright(createOrder, addOrderItem, deleteOrderItem, cancelOrder)$

$email : EMAIL$
$order : Order$
$orders : \mathbb{P}\ Order$
$orderItem : OrderItem$
$orderItems : \mathbb{P}\ OrderItem$

*INIT*

$orders = \varnothing$

*createOrder*

$\Delta(orders, orderItems)$
$order? : Order$
$orderItem? : OrderItem$

$order? \notin orders$
$orders' = orders \cup \{order?\}$
$orderItem? = orderItems$
$orderItems' = orderItems \cup \{orderItem?\}$

*addOrderItem*

$\Delta(orderItems)$
$orderItemID : Id$
$orderItem? : OrderItem$

$orderItem? \notin orderItems$
$orderItems' = orderItems \cup \{orderItem?\}$

*deleteOrderItem*

$\Delta(orderItems)$
$orderItem? : OrderItem$

$orderItems' = orderItems \setminus \{orderItem?\}$

*cancelOrder*

$\Delta(orders)$
$order? : Order$

$orders' = orders \setminus \{order?\}$

Theoretical and practical considerations of the system were discussed. In comparison to traditional e-commerce systems, this system is novel and will become more and more popular in the next few years. First, we employ use case views and dynamic views of UML to model this special e-commerce system. Then, we use another formal language, Object-Z, to give a more formal specification of this system. The examples presented in this paper were small, easily modeled with UML diagrams and the Object-Z specification. By integrating the two different approaches for the specification, we derive easy visualization of the system and its components, containment of complexities of the behaviours of the e-commerce system via their visual models, and concentrate on using different aspects of a method / notation that are most suitable to model the operations and capture the dynamic views of the system.

The design presented is complete and effective. However, there are some other aspects we should consider in the future extension of this paper. These include security issues between peers, and payment issues with the financial institutions. The most vulnerable part of any electronic commerce transaction is the communication between the server and the client. The transaction data / information travel through transmission media, routers and possibly hundreds of miscellaneous devices. The seven threats[37] of malicious code, hacking, credit card fraud / theft, spoofing, denial of service attacks, sniffing and insider jobs are perpetuated here. Encrypting the information transmitted is necessary to reduce the risk of revealing the content of the message to malicious individuals who may wire-tap the links. One of the most widely used protocols to encrypt communications between the customers' browsers [and the electronic commerce site] is Secure Sockets Layer (SSL).

The need to present different aspects of security for electronic commerce is crucial. Generic solutions for securing servers and communications, including some of the most common cryptographic security solutions, will be examined. Clearly, an integrated solution that includes cryptography is part of the solution. Failure to recognize this risk would be catastrophic for a large number of sites at the same time, thereby putting in danger the economy of countries dependent on e-commerce businesses.

## References

1 M. Norris, S. West, and K. Gaughan, *eBusiness Essentials*, John Wiley and Sons Ltd., 2000.

2 S. A. Ehikioya and K. Hiebert, "A Formal Model of Electronic Commerce", *First International Conference on Software Engineering, Networking, and Parallel and Distributed Computing,* May 19-21 2000, Champagne-Ardenne, France, 2000.

3 R. Rajkumar, N. Ch. S. N. Iyengar, and D. Saikrishna, "Architecture for Mobile P2P Auction using JXTA / JXME in M-Commerce", *International Journal of Advanced Engineering Sciences and Technologies (IJAEST),* Vol. 4, Issue #2, 2011, pp. 004-009.

4 S. M. Liu, K. R. Lang, and S. H Kwok, "Participation Incentive Mechanisms in Peer-to-Peer Subscription Systems", In *Proceedings of the 35th Annual Hawaii*

*International Conference*, Volume 9, January 2002. , page 302b.

5 C. Wang, A. Carzaniga, D. Evans, and A. L. Wolf, “Security Issues and Requirements for Internet-scale Publish-Subscribe Systems”, In *Proceedings of the 35th Annual Hawaii International Conference*, Volume 9, January 2002. page 303

6 S. Y. Choi, D. O. Stahl, and A. B. Whinston, “Chapter 2: Characteristics of Digital Products” in *The Economics of Electronic Commerce*, Macmillan Computer Publishing, 2003.

7 *Unified Modeling Language (UML) Specification, Version 1.3a*, Object Management Group, March 1999. (Available at http://www.rational.com/uml/)

8 G. Smith, *The Object-Z Specification Language*, Kluwer Academic Publishers, 2000.

9 S. A. Ehikioya, “A Formal Perspective to Modelling Electronic Commerce Transactions”, *Colombian Journal of Computation*, Volume 2, Number 2, 2001.

10 R. T. Marshak, “Peer-to-Peer Business Models: Facilitating “Matchmaking” between Individuals for Fun and Profit”, Patricia Seybold Group, USA, July 12, 2013. (Available at: Direct link: http://dx.doi.org/10.1571/bs07-12-13cc). Accessed on April 6, 2018.

11 I. Sommerville, *Software Engineering*, 10th Edition, Addison-Wesley, 2015.

12 D. Georgakopoulos and M. P. Papazoglou, “Overview of Service-Oriented Computing”, Chapter 1 in *Service-Oriented Computing*, edited by Dimitrios Georgakopoulos and Michael P. Papazoglou, The MIT Press Cambridge, Massachusetts, USA, 2009.

13 S. A. Ehikioya and C. Zhang, “A Real-Time Multi-Agents Architecture for E-Commerce Servers”, *International Journal of Networked and Distributed Computing (IJNDC)*, Vol. 6, #2, April 2018, pp: 88-98.

14 S. A. Ehikioya and K. Hiebert., “Agents Negotiation in Electronic Commerce Transactions”, *1st Annual International Conference on Computers and Information Science (ICIS’01)*, The Grosvenor Resort, Orlando, Florida, U.S.A., October 3-5, 2001

15 S. A. Ehikioya, "An Agent-based System for Distributed Transactions: A Model for Internet-based Transactions". *IEEE Canadian Conference on Electrical and Computer Engineering*, Edmonton, Alberta, Canada, May 9-12, 1999.

16 S. A. Ehikioya and T. Walowetz, "A Formal Specification of Transaction Systems in Distributed Multi-Agents Systems". *ISCA 14th International Conference on Computers and Their Applications*, Cancun, Mexico, April 7-9, 1999. pages 378-383.

17 J. Walkerdine, L. Melville, and I. Sommerville, “Dependability Properties of P2P Architectures”, In *Proceedings of the Second International Conference on Peer-to-Peer Computing*, Linkoping, Sweden, September 5-7, 2002. page 173.

18 L. Melville, J. Walkerdine, and I. Sommerville, “Ensuring Dependability of P2P Applications at Architectural Level”, D5 - Report on the Dependability Properties of P2P Architectures, 0305F01_*DependabilityReport No. IST-2001-32708*, Information Societies Technology (IST) Programme, P2P ARCHITECT, July 31, 2002.

19 O. Jogunola, A. Ikpehai, K. Anoh, B. Adebisi, M. Hammoudeh, H. Gacanin and G. Harris, “Comparative Analysis of P2P Architectures for Energy Trading and Sharing”, *Energies*, Vol. 11 Issue #1, 62, 2018; doi:10.3390/en11010062

20 “The Peer-to-Peer (P2P) Marketplace”, *White Paper*, Happiest Minds Technologies Pvt Ltd, 2014.

21 N. C. Barwar and B. Rajesh, “Network performance analysis of startup buffering for live streaming in p2p vod systems for mesh-based topology”, In *Proceedings of the International Congress on Information and Communication Technology*, Springer: Singapore, 2016; pp. 271–279.

22 F. Chowdhury and M. Kolberg, “Performance evaluation of structured peer-to-peer overlays for use on mobile networks”, In *Proceedings of the Sixth International Conference on Developments in eSystems Engineering (DeSE)*, Abu Dhabi, UAE, 16–18 December 2013; pp. 57–62.

23 K. Tutschku and A. Nakao, “Towards the Engineering of Dependable P2P-Based Network Control: The Case of Timely Routing Control Messages”, *IEICE Trans. Commun.*, Vol. E92-B, No.1, January 2009.

24 M. H. Tran and S. V. U. Ha, “Decentralized Online Social Network Using Peer-to-Peer Technology”, *REV J. Electron. Commun.*, 2016, 5, 1–2.

25 B. Yuan, L. Liu, and N. Antonopoulos, “A Self-organized Architecture for Efficient Service Discovery in Future Peer-to-Peer Online Social Networks”, In *Proceedings of the IEEE Symposium on Service-Oriented System Engineering (SOSE)*, Oxford, UK, 29 March – 2 April 2016; pp. 415–422.

26 C. Adiele and S. A. Ehikioya, “Evolving a “Wise” Integration System for E-Commerce Transactions”, *Journal of Electronic Commerce Research and Application*s, Volume 6, No. 2, 2007, pp: 219 - 232.

27 C. Adiele and S. A. Ehikioya, “Algebraic Signatures for Scalable Web Data Integration for Electronic Commerce Transactions”, *Journal of Electronic Commerce Research*, Vol. 6, #1, 2005, pp.56-74.

[28 J. M. Spivey, *The Z Notation: A Reference Manual*, 2nd Edition, Prentice Hall International Series in Computer Science, 1992.

29 J. M. Spivey., *Introducing Z: A Specification Language and Semantics*, Cambridge University Press, 1988.

30 S. A. Ehikioya, "A Formal Characterization of Electronic Commerce Transactions", *International Journal of Computer and Information Science*, Volume 2, Number 3, 2001.

31 S. A. Ehikioya, "Modelling Electronic Commerce Transactions in Z", *First International Conference on Internet Computing*, Monte Carlo Resort, Las Vegas, Nevada, USA, 2000, pp. 245 – 252.

32 B. Ola, “A Comparative Evaluation of Specification Techniques for Electronic Commerce Systems”, *Master’s Thesis*, University of Manitoba, Department of Computer Science, Winnipeg, Manitoba, Canada, September 2002.

33 S. A. Ehikioya, and B. Ola, “A Comparative Study of Specification Methods for Electronic Commerce Systems”, *The 3rd ACS/IEEE International Conference on Computer Systems and Applications*, 2005, Page(s):108

34 S. A. Ehikioya and B. Ola, “A Comparison of Formalisms for Electronic Commerce Systems”, *Alternative Approaches in Software Engineering; A special session at IEEE International Conference on Computational Cybernetics*, Vienna, Austria, 2004, pp.253 – 258.

35 S. A. Ehikioya, B. Ola, and J. Akinyemi, “A Taxonomy of Specification Techniques for E-commerce Systems”, *3rd International Conference on Computer Science and its Applications (ICCSA-2005)*, San Diego, California, USA, 2005.

36 J. He, “A Formal Specification and Design of an Online Bazaar System”, *Master’s Thesis*, University of Manitoba, Department of Computer Science, Winnipeg, Manitoba, Canada, September 2002.

37 K. C. Laudon and C. G.. Traver. *E-commerce*. Addison Wesley, 2002.