Project Venezia-Gondola
(A Framework for P-Commerce)

Raymond Gao
Editor-in-Chief, P2P Journal
mailto:raygao@comcast.net
(972) 530-4562
5609 Harbor Town Drive
Garland, TX 75044

Abstract

This article presents the Project Venezia-Gondola (Project V-G), an application platform for enabling Peer-to-Peer commerce (P-Commerce). The author of the article proposes a new distributed computing pattern called the Inverted Model-View-Controller (IMVC) pattern. The article discusses IMVC's suitability for P-Commerce. And, the article outlines Project V-G's architecture and elucidates to readers on lessons learned from this experiment.

Article 1.0 – June 10, 2004

1 Introduction

The rapid growth of online commerce (more specifically E-Commerce) is one of the driving forces for Internet. E-commerce allows people to make transactions via the web. Some examples include retail sales to consumers, Electronic Data Interchange (EDI) networks between merchants and distributors, online exchanges for manufacturers and suppliers, etc. According to industry analysts, E-Commerce can be segmented into different market areas by looking at various attributes, i.e. market size, customer profiles (behaviors and other traits), as well as various technology drivers, etc.

1.1 Existing E-Commerce Models

B2C – The Business-to-Consumer market segment primarily deals with individuals. It directly sells goods and services to end-users. Personalization and content aggregation are some of the key strategies for this market segment.

B2B – The (Business-to-business) market segment deals with companies. B2B transactions often are more complex. Many enterprises also have stringent security policies. Additionally, businesses sometimes have legacy systems working in the background. As a result, good system integration capabilities between business partners are highly desirable.

Other e-commerce models - Other e-commerce models include Government to Business, Government to Citizen, Business to Employee (B2E), etc.

1.1.1 E-Commerce's Primary Enabling Technologies

The N-Tier architecture is widely accepted for e-commerce offerings. The N-Tier architecture separates presentation, business logic, process management, persistence (database), and integration into separate layers. HTML/XML, CSS, Javascript, and DHTML technologies have become staple technologies for E-Commerce in providing front-end and graphic user interface (GUI) services. Over a period of time, specialized tools have been developed for implementing N-Tier architecture, e.g. application servers, web server, portal servers, directory servers, workflow and integration engines, databases, etc. (See Figure 1)

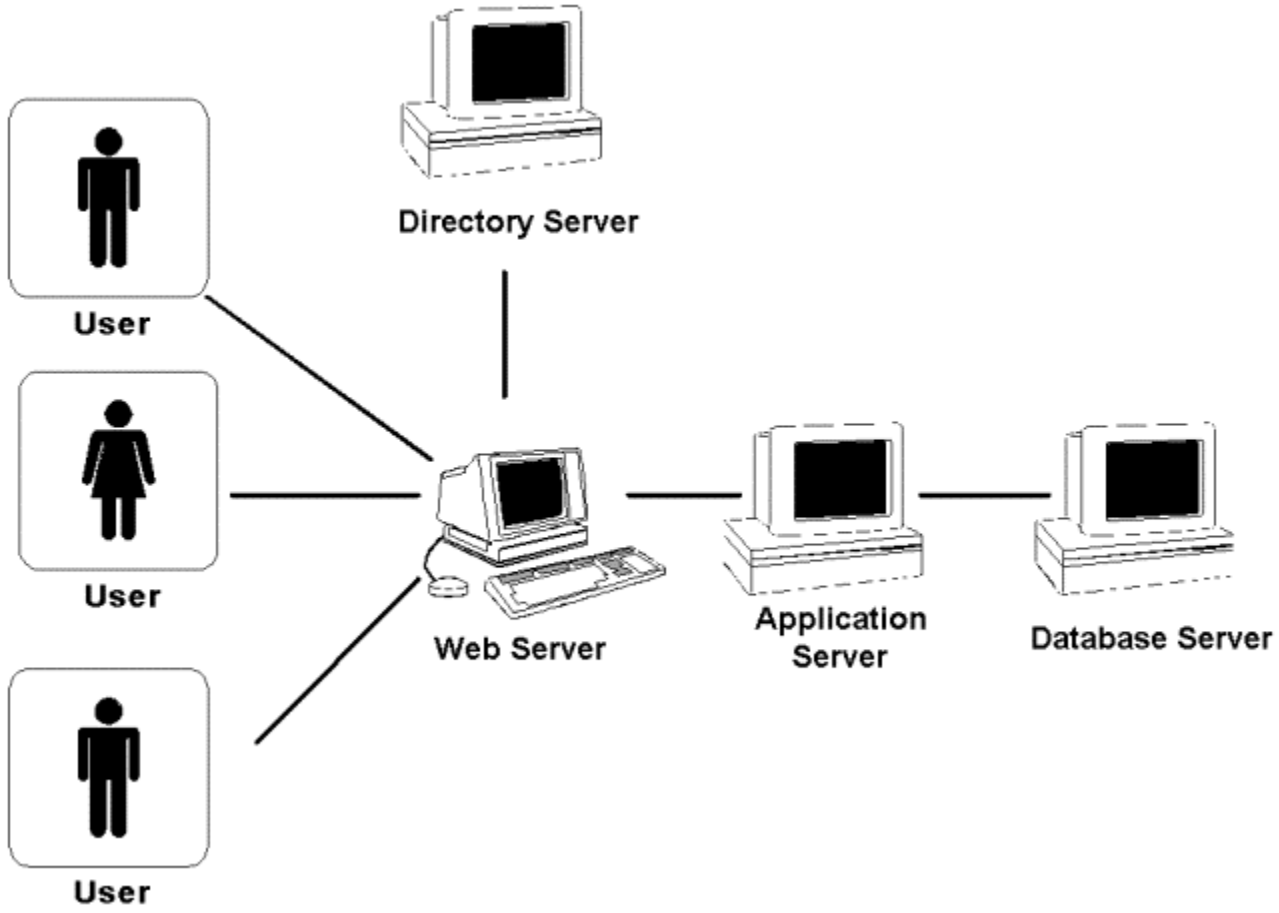


Figure 1 A Typical E-Commerce Model

1.1.2 Weakness

E-Commerce uses websites for the front-end. It inheritably depends on centralization, putting the content and the transaction logic on a central website or a set of server farms. However, centralization has drawbacks. Centralization means managing systems from a single point. That potentially leads to an opportunity for "a single point of failure". Additionally, centralization may mean that user's local settings can be overridden by central policies. That can cause inconvenience. Integrating is another

concern, particularly in dealing with complex systems between multiple business partners. No one wants to invest money on "brittle" legacy systems and inflexible IT infrastructure with negative Return On Investment (ROI).

Everyone is talking about Reliability, Availability, and Scalability (RAS). Maintaining a high performance websites for large number of users can be an expensive and challenging task. The infrastructure alone can cost a lot of money, e.g. to acquire many powerful servers, to sustain high bandwidth connectivity, and to manage a datacenter. The investment in human capital means additional cost. Often a team of specialists, system administrators and support people are needed to keep a website running smoothly. Security poses additional areas of concern, enterprise are very conscious about preventing computer virus spreading, professional hackers, worms, denial of service (DOS), network traffic sniffing, etc. Eventually, all those costs trickle into every business transaction. Users will have to pay for those services.

1.2 What is P-Commerce?

Peer-to-Peer Commerce (P-Commerce) is an alternative model to centralized commerce. It is based on the ad-hoc relationship between individual participants. Its uses Peer-to-Peer (P2P) network infrastructure.

A transaction by nature is transient. It records an event-in-time between two or more participants. Since a transaction is a temporary phenomenon, wouldn't it be more cost effective to focus on that event than investing large sums of money on a centralized website? By giving people autonomy and shifting control from the central authority to individuals, participants will be happier.

The P-Commerce's model is, "The network is the ultimate commerce channel".

1.2.1 Benefits

P-Commerce empowers people. In a business transaction, direct participants have the firsthand knowledge and understand current issues better than anyone else. Sometimes, they like to make their own rules rather than dealing with red tape policies set by an arbitrary entity. P-Commerce avoids inflexible business logic that is determined by a central website. Direct participants know what are appropriate transaction guidelines, operation etiquettes, and the amount.

P-Commerce is more flexible and has several technical advantages. It allows direct interaction between participants. It more accurately models a transaction where peers at the network edge interact; the "transaction logic" and "content" are distributed. P2P network can also scale better than the centralized architecture. It circumvents "the single point of failure" problem. Individual peers can make decision on whether to use a thin or a full-feature application.

P-Commerce saves money. Middleman takes a cut out of every business transaction. Peers who are price-conscious can directly deal with one another and avoid the middleman, thus reducing cost.

P-Commerce introduces creativity. It allows non-monetary based transactions, i.e. bartering (exchange & trade), goodwill (giving away merchandises and services based on certain criterions). It allows business to come up with imaginative new services and not feeling pressured by price wars. Quality and originality become more important bottom lines.

1.2.2 Current State

Currently, P2P technology is used in five main categories, file-sharing oriented, collaboration oriented, distributed computing oriented, distributed storage oriented, and distributed platform-oriented.[1] Telecommunication services such as voice-over-IP (VOIP) are emerging areas for P2P.

P2P technology has gone through several stages of evolution. The first stage was the Gnutella, Napster, and Freenet era. Those early applications' searching and listing functions were based on a combination of distributed querying, pinging, caching, and time-to-live (TTL) settings. The second-generation, identified by Overnet, eDonkey, RevConnect, Chord, CAN, focuses on performance improvement. For example, JXTA now uses the Distributed Hash Table (DHT). And, we are beginning to see the emergence of the third generation P2P applications that understand semantics and can intelligent route traffic. Additional areas for improvement include better peer-reputation management and solving the distributed trust issues.

1.2.3 Project VG – A New Direction

Using P2P technology for online commerce is a brand new idea. The Project Venezia-Gondola (Project V-G) is a bold new initiative. It is an experiment to verify author's belief that P2P technology has matured sufficiently and now has the necessary infrastructure for conducting intelligent and cost-effective business activities.

Project VG is a P2P yardsale application framework for P-Commerce. It adds "bartering" and "goodwill" functions and additional creative business processes. It has three components: a P-Commerce network called "Venezia Network", a P-Commerce engine called "Venezia", and a graphic user interface call "Gondola". Project VG proposes a new computing model called the Inverted Model-View-Controller (MVC) pattern. (See section 2)

Project V-G is also an open-source project. Developers can write their own user interfaces (UI), i.e. "skins" for "Gondola.

[1] Luca Caviglione, "The "dark" side and the "force" of peer-to-peer computing saga", P2P Journal, January, 2004

1.3 Key Concepts and Terminology

There are certain prerequisite concepts and terminologies.

Presence - Presence identifies a user's current status, e.g. online, busy (away), or offline.

Trust - Trust forms the cornerstone of business relationship. It is a gauge for keeping track of a users' reputation. It can also be used to measure the fidelity of a P2P marketplace. Trust maybe based on words-of-month from friends, search engine's result, and a summary of other peer's recommendations.

Reputation Tracking - It is a tool for determining the risk of doing business with a certain person. Under Project Venezia, it is an aggregate score from that user's previous business partners. A new user has an initial score of 1, meaning that the User A trusts himself or herself. A user's business partner can then rank that user with a score of -1 (not trustworthy), 0 (neutral), +1 (trustworthy). The score is tallied using a moving average method. Additional scorings factors include the nature of a transaction, quantity, and frequencies, etc.

Venezia Network – This is the backbone network for P-Commerce. Marketplaces are subgroups of the Venezia Network. A good analogy is to think how the "Grand Canal" of Venice links different districts & "marketplaces" together.

Gondola – That is the GUI interface.

Marketplace – It allows peers with same interest to band together and aggregate their listings. For example, in Venice "Piazza San Marco" is where people gather for lunch and coffee. A marketplace should have at minimum two peers present because one cannot trade with oneself. Project V-G uses JXTA's *Peer group* for implementation.[2]

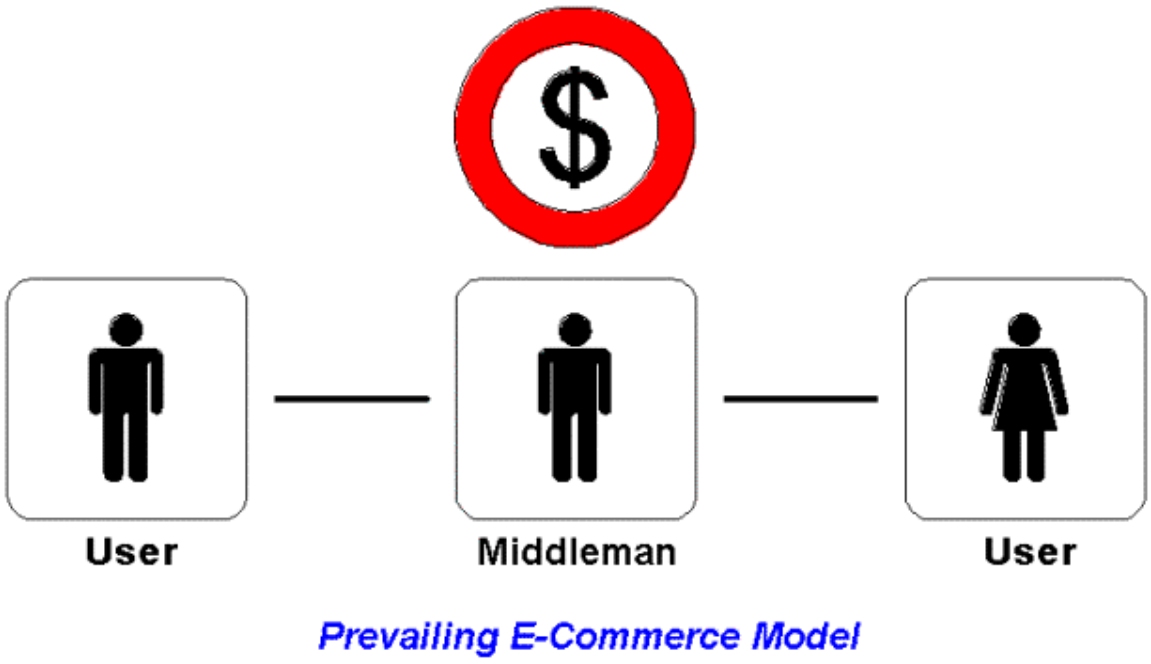
Caching - This is a mechanism used to help speed up and to improve the accuracy of search.

Buddy List - This list organizes trustworthy business partners and friends.

Black List - People on this list should be avoided. (Perhaps they provide shady services / products or maybe they are unfriendly.)

---

[2] "A *Peer group* is a collection of peers that have agreed upon a common set of services. Peers self-organize into *peer groups*." (See JXTA Programmer's Guide.)



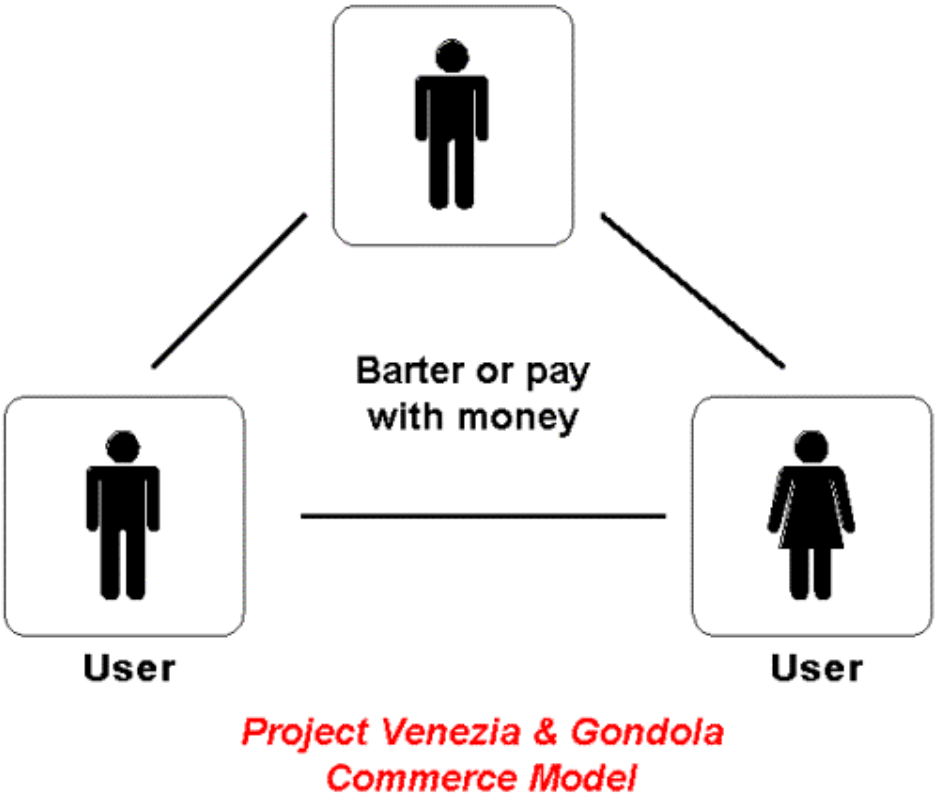


Figure 2 Comparing Two Online Commerce Models

2 P-Commerce, Model and Patterns

After analyzing several P2P networks and surveying general peers behaviors, we have decided to use the Inverted Model-View-Controller (IMVC) pattern for Project V-G.

2.1 The MVC Pattern

The Model-View-Controller (MVC) pattern has its roots extending back to IBM mainframe computing environments. The MVC pattern separates a complex application into three tiers. Those tiers are Model, View, and Controller. The Model represents the underlying data-store as well as access, update, manage, and delete functions. The View component displays those data in a useful format for the user. And, the Controller translates user actions and dispatches appropriate methods on the Model.

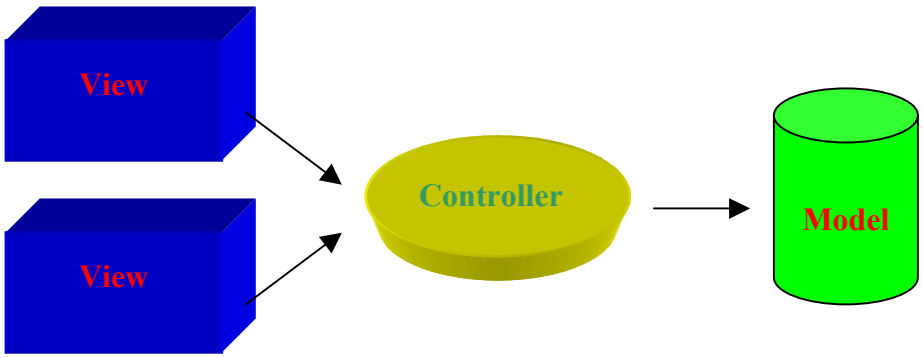


Figure 3 The MVC Pattern

Recently the MVC pattern is experiencing a Renaissance period. Open-source efforts, e.g. Struts and Turbine, are modeled after the MVC pattern. The J2EE Blueprint also makes an elaborate case for the MVC pattern.

2.2 Patterns observed from P2P network

The P2P network is based on ad-hoc relationships. Content is dispersed throughout the entire network and not aggregated on a single web-server. Looking at how to provide content management services, this becomes very clear, "The client-server model wants to centralize (content); the P2P model wants to push information to the edge."[3]

[3] Raymond Gao, "To P2P or P2P Too: A discussion of Peer-to-Peer and Related Technologies", P2P Journal, July, 2003

2.2.1 Reciprocal Relationship

Peers are independent entities. Each peer is a self-contained unit that is both a client and a server, e.g. "Servent" in Gnutella. Peers make individual decisions of whether to consume content, to add new content, to route information, or to observe activities in the system.

2.2.2 Browsing vs. Providing Information

"A study of Gnutella network has shown that 70 % of peers were free riding."[4] It is fairly obvious that browsing for content is the primary activity for many users. Only 10% of altruistic individuals actively add new material. This is quite similar to the online commerce situation. Most people are shoppers. Only few entities and organizations are merchants.

2.2.3 Distributed Content and Transaction Logic

A recent technical article says, "popular file-sharing applications amounts to a considerable fraction (up to 60%) of the total Internet traffic."[5] Decentralizing content leads to decentralized transaction logic. In P2P networks, there is no single entity that controls the content. Likewise, it is through peer collaboration and popularity, new contents get passed between people. Again, one can draw a parallel to commerce in our daily life. There are many stores. People make their purchase decision based on price, product quality, service, accessibility, public opinion, and/or their previous experience with a merchant. Consumers manage their pocketbooks and make smart purchase decisions. To model their behavior, a commerce system needs to be equally versatile at mimicking peers activities.

2.3 The Invert MVC Pattern

The Inverted MVC pattern is a derivative of the MVC pattern.[6] While the data organization scheme is decentralized, the P2P network behaves like a large data source. One can think that peers are "tables"; peergroups are "schemas" that group several peers together; and the P2P network is a giant "database". Each user has a "window" into that system via his/her application. (See Figure 3)

For system performance, stability, and transparency reasons, some peers are promoted to super-peers, (rendezvous hosts and proxies). Super peers maybe more powerful computers, have high bandwidth, or bridge intranets and extranets. Super peers can be used to rein in control from chaos, to enhance the network performance, and to improve system management. Additionally, super peers can joint chart business rules and help define contexts for transactions.

---

[4] Ben Strulo, "Middleware to Motivate Co-operation in Peer-to-Peer Systems", P2P Journal, March, 2004

[5] Dimitrios Tsoumakos, Nick Roussopoulos, "Probabilistic Knowledge Discovery and Management for P2P Networks", P2P Journal, November, 2003

[6] The MVC pattern buffers the data-store and the transaction logic from end-users and is more suitable for a website.

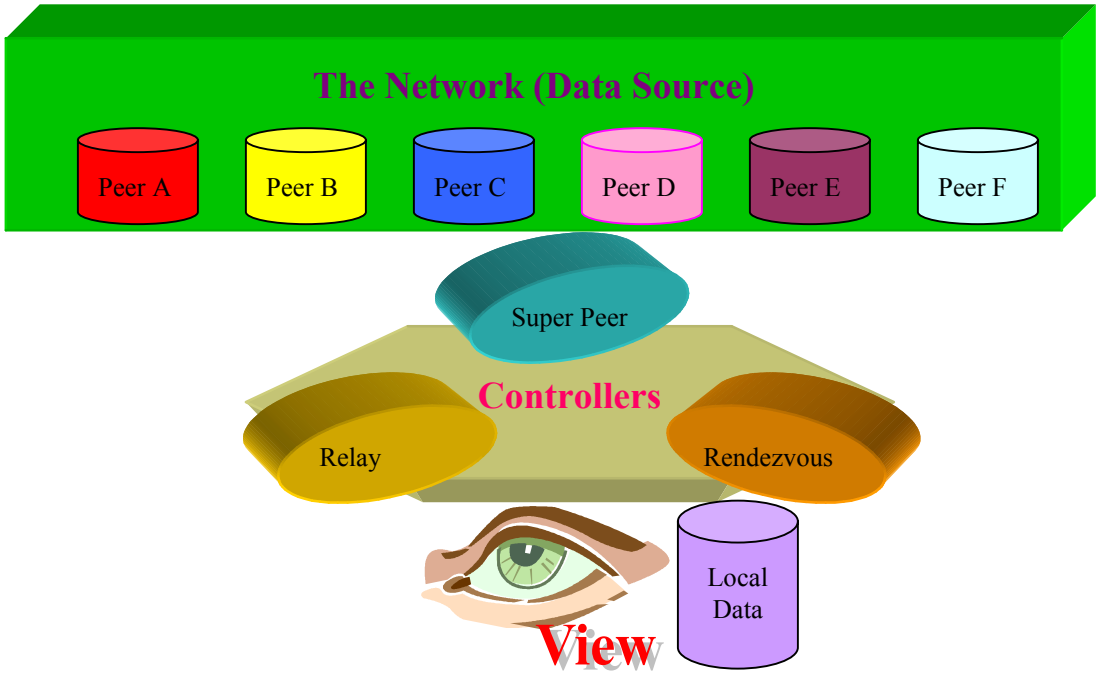


Figure 4 The Inverted MVC Pattern

2.3.1 Suitability of Invert MVC Pattern

The Inverted MVC Pattern helps reinforcing the separation between the data storage, the presentation, and the process flow. If P2P networks are designed to eliminate the hierarchy, then the IMVC pattern provides a strong model for developing P2P applications. The modeling process has two aspects, structure and behavior. The structure mapping process was discussed earlier. The behavior mapping process can be further refined to the "Observer Pattern", an IMVC sub-pattern.

"When we implement the Observer pattern, we usually refer to the data as the Subject and each of the displays as Observers. Each of these observers [listening peers] registers its interest [with] the subject [content provider]. Then, each observer has a known interface that the subject calls when the data change. Observers promote abstract coupling to Subjects. A subject doesn't know the details of any of its observers."[7] This notion of separating the content provider and the listening peer models the P2P network very well.

[7] James W. Cooper "Java™ Design Patterns", Addison-Wesley, 2001

In fact, we have observed such a pattern in our experiment. The rendezvous peer is an area controller. We have set up a private rendezvous host for Project V-G. We can use the "tail" command to read the "log file" of our "rendezvous host" and control the number of active clients using that service.

3 Project Venezia-Gondola's Goals and the Design

3.1 Solution Statements

Project V-G is a P-Commerce application framework. It allows participants to conduct various decentralized business activities, e.g. buy, sell, haggle, barter, bid, and goodwill. Project V-G believes in that self-governing marketplaces are very efficient. Different products and services have different characteristic. Likewise, trading and/or bartering should have different guidelines. Having a single marketplace for trading everything may cause excess regulations and impede normal business activities.

Project V-G provides tools (network services and toolkits) that encourage establishing multiple P-Commerce marketplaces. Those marketplaces can have competing, cooperative, inter-dependent, and consolidated relationships, as well as be self-standing. It provides an alternative to the mono-marketplace approach because it is too complex to for a single website to write and monitor all business rules.

3.2 Architecture Overview

Project V-G three modules. The Gondola module functions as the Graphic User Interface (GUI), a façade to the underlying engine. The Venezia module is the core that enables P-Commerce. And, there is a third module for importing and parsing business rules.

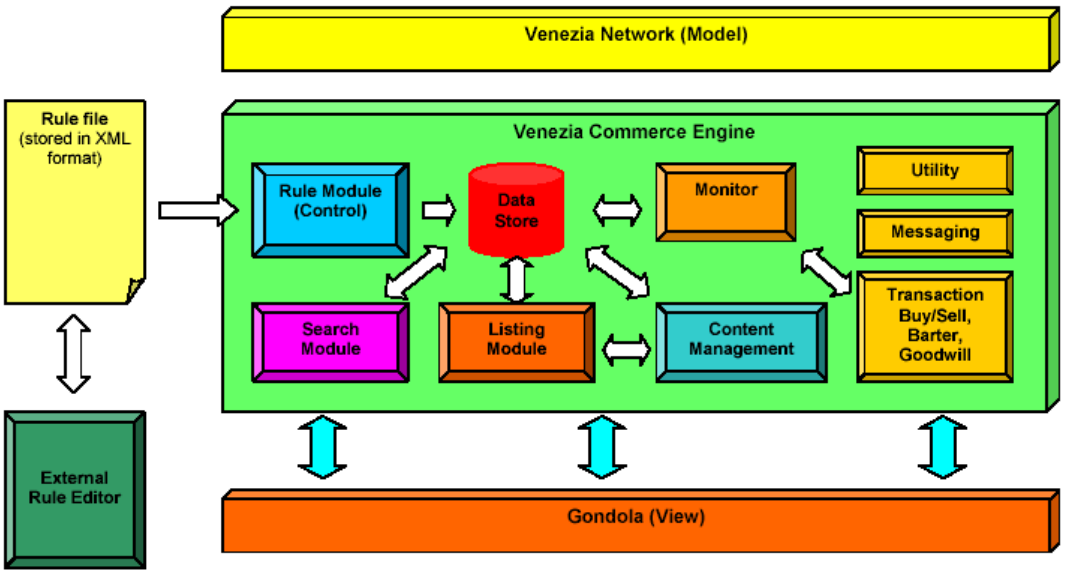


Figure 5 Project Venezia-Gondola Technical Architecture Model

The architecture follows the Inverted MVC pattern, whereby the "Model" is the Venezia P-Commerce Network. Many users will spend majority of their time browsing rather than adding new listings. The "View" is the Gondola module, which acts as the "windows" into the network. And, business rules are used to control the flow of information and guide the user's interaction with the system. Business rules can be jointly edited and are stored in the XML format. Applications create "monitors" from importing business rules.

3.3 Typical P-Commerce Processes

Certain processes are recurring across most online commerce activities. Those processes can be grouped into following categories - Registration, Searching & Browsing, Listing, Transaction, and Feedback.

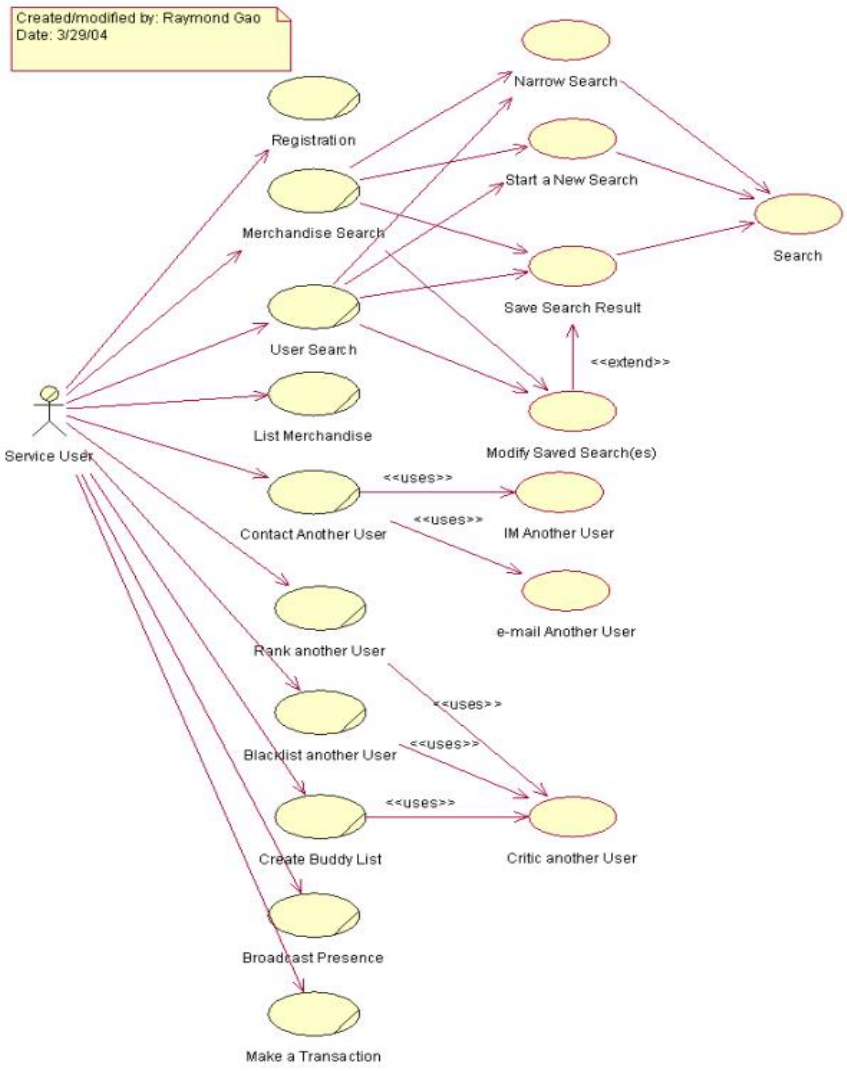


Figure 6 Project Venezia-Gondola's Common Capabilities

Additionally, there are utility functions to support above processes, e.g. Peer communication, Logging of transactions, Post-transaction critic and other miscellaneous activities.

3.3.1 The Registration Process

Before a *User A* can participate in the Venezia Commerce Network, he/she must establish his/her identity. To register, *User A* needs to have Name (First-name plus Last-name), E-mail address, Instant Message ID, Phone number, Postal Address, and Zip Code.

Project V-G intends on implementing that function using the self-identification and the quorum process. It involves generating a pair of asymmetrical keys for encryption and decryption from U*ser A*'s e-mail address, node-name, node's Internet Protocol (IP) address, etc. The process hashes *User A's* e-mail address (*username@URL*) to produce a Unique User Identification number (UUID) and then ask peers to insert that UUID in the community listing. The success and failure of that insertion process depends on tallying the number of confirmations against a minimum acceptable number P_{min} .

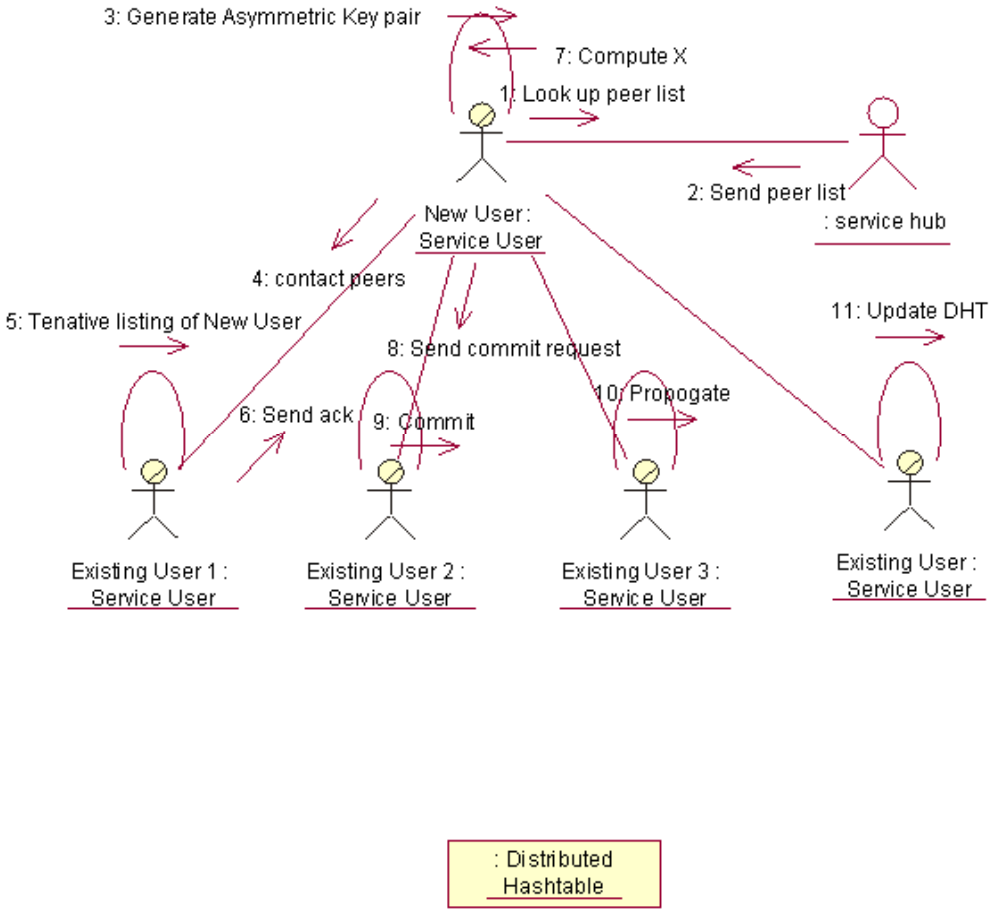


Figure 7 The Registration Process

3.3.2 The Browsing Process

The “browsing” process requires understanding how the listing process works. In a P2P network, product entries can have overlapping names. Occasionally, this produces a “name collision” problem. To address this issue, each item will have a Global Unique ID (GUID) that acts as its primary identifier. That GUID is created base on owner’s UUID, a hash function, item name, and nodename.

The “browsing” process involves finding a GUID and retrieving its information, i.e. description, name, quantity, price. The “Iterator” pattern is highly suitable for this purpose.

3.3.3 The Searching Process

There are two directories. One is a list of merchandises. The other is a list of users. There is also a "map", linking merchandise listings to owners. (See Data Model)

Searching is done in the typical P2P fashion, by either contacting a central rendezvous server or broadcasting search queries in the network with a time-to-live (TTL) number. The individual user determines searching criterions. A user can either search for “user ID” or “item listings”.

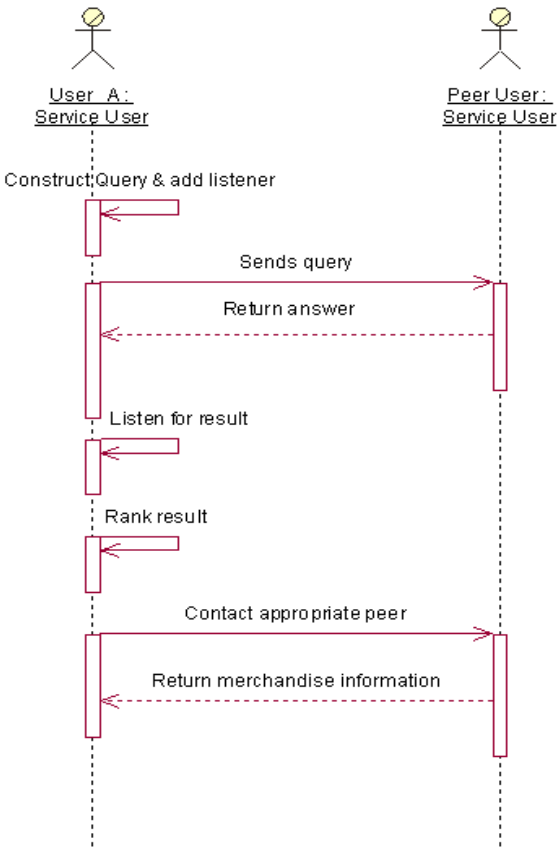


Figure 8 Searching for Merchandise

The “search” process has several sub-processes, including.

- Start a new search[8]
- Save search result
- Modify and refine search criterions, e.g. Keywords, Category, Cost, Method of transaction (Price range, barter, goodwill), etc
- Comparing search results
- Merge two or more searches into one

Searching is related to the "Wish list" function. “Searching” uses the query method. And, the “Wish list” function is proactive advertising.

The P2P searching process is different from searching on a traditional websites. It may not be exhaustive and is constrained by peers going on and offline, preset propagation settings on rendezvous servers, time-to-live (TTL), network segmentation issues, etc.

3.3.4 The Listing Process

Adding a new listing involves generating a GUID for an item, confirming that ID, and propagating that listing in the network. The GUID creation is based on owner’s UUID, item name, and a custom hash function. Each item will have following information.

Product Specific

1. Name
2. Category
3. Keywords
4. Detailed description
5. Images
6. Available Quantity
7. Price
8. Payment Options (Sell, Barter, Goodwill)

Support Information

1. Documentation, user guide, etc.
2. Shipping information
3. Contact information, e.g. phone number, e-mail, IM, etc.
4. Reference

[8] The “Interpreter” pattern is very useful in understanding a user’s input and for constructing queries.

3.3.5 The Transaction Process

There are several types of transactions[9], including straight purchase, haggle, auction, barter, goodwill, etc. Each transaction may have additional subtypes and be inter-linked. For example, the auction process has the Dutch auction as a sub-category.

Straight Purchase - Buying a product without negotiation. The buyer agrees to pay the seller's advertised price. (See Figure 9)

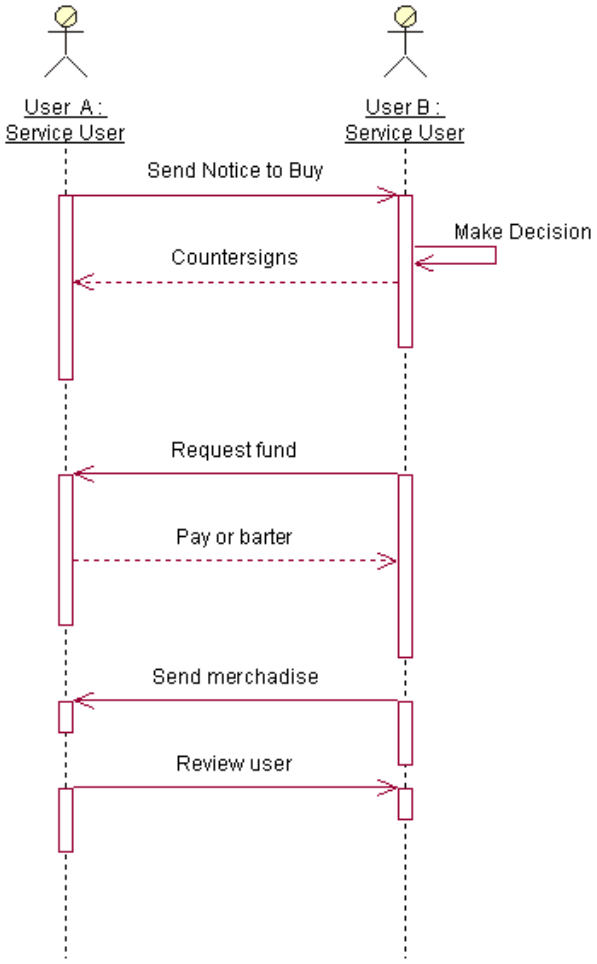


Figure 9 Making A Transaction

Haggling – During the bargaining process, the buyer sends the seller his/her price. The seller replies with either a "yes "or "no" answer and a new asking price. The buyer adjusts the bid and repeats the process until both parties agree. If no agreement can be reached, the transaction is abandoned.

Bartering - Both parties choose to exchange items instead of using money.

Goodwill – The seller gives away some free stuff to generate new commercial interest.

[9] The "Momento" pattern is suitable for keeping track of the transaction process.

3.3.6 Post Transaction Activities

After completing a transaction, both parties have an opportunity to review each other by giving a score of -1 (not trustworthy), 0 (neutral), +1 (trustworthy). A cumulative score is used to keep track a user's reputation. The party being reviewed will provide the reviewer a temporary session key to record those data. Those keys will be saved in a "key ring". And, a user's reputation will be propagated in the network.

Blacklist A User - Blacklisting a user is valid for 30 days in the system. The time limitation allows a user to recover from his/her previous lapse in the judgment.

Buddy List - The buddy list helps to keep track friends and provides the basis for the referral service.

4 The Data Model

Project V-G currently uses HSQL for the local data-store. There are two primary tables. The "User" table stores the user specific information, such as "User ID", "Name", "E-mail", "Interests", etc. The "Item" table keeps track information about the merchandise. The "Item" table is linked to the "User" table via owner as a foreign key.

Secondary tables include "Review", "Transaction", and "SearchResult" tables. The "Review" table is used to keep track a user's reputation. And, the "SearchResult" table stores searching information. Finally, the "Transaction" table is a temporary means to store the transaction information.

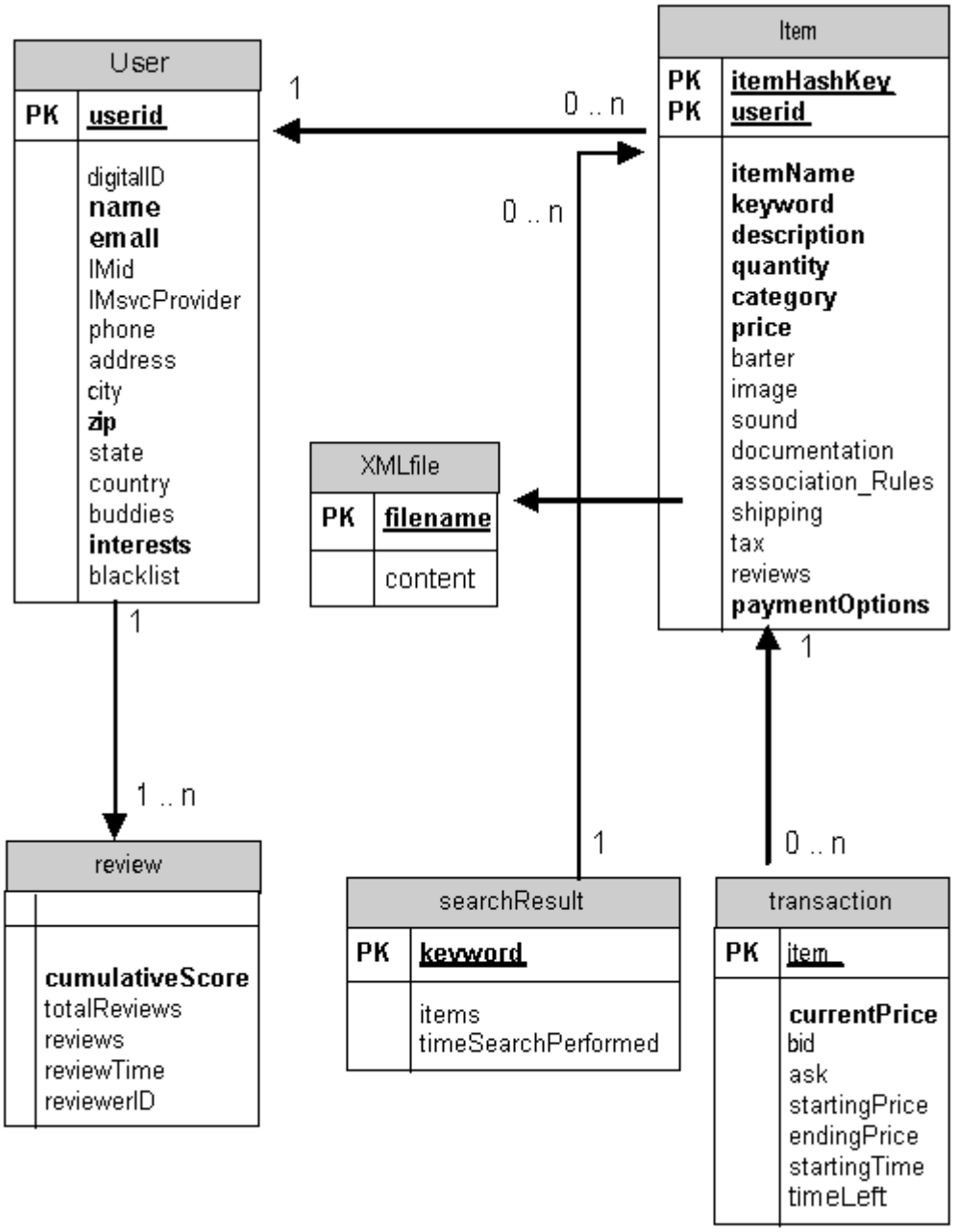


Figure 10 Project V-G Data Model

5 Observations

Project V-G is an ongoing project. During the implementation process, several observations were made.

1. Business processes for online commerce can become very complex. Section 3.4 has shown several typical P-Commerce processes. In reality, those processes barely meet the "tip of the iceberg". There are many other types of processes and variations.
2. It is important to keep the client binary distribution to as small as possible. We are currently building this project on Sun's JXTA platform. We have found that JXTA carries many

dependent libraries (JAR and ZIP files). The size of the JXTA platform alone exceeds 6 megabytes. While broadband has become quite common nowadays, for international and dial-up users size remains an issue. We hope to strip down the JXTA platform and eliminate unused library files.

3. "Rendezvous pollution" poses a performance problem. Currently, the JXTA's configuration utility allows any user to become a rendezvous host. Because a peer can disconnect anytime, it may leave its clients waiting for an extended time. Additionally, response from JXTA's public Rendezvous hosts maybe slow at times. (It could take up to several minutes to resolve a connection.) To work around those issues, we have set up a private Rendezvous host that restricts access to a predefine peergroup. That private box runs BEA's JRockit as its Java Virtual Machine on a Red Hat Linux 9.x platform.
4. The conventional P2P searching method, broadcasting queries, is inefficient for a large network. It could potentially cause an avalanche of network activities. We are working on a referral model. Hopefully, that searching process will become more intelligent.
5. In the age of cyber-threat, many people have personal firewalls installed or are using network address translation (NAT) devices. Occassionally, they pose problems for the discovery process.
6. JXTA is a research-oriented project. From time to time, we have observed that JXTA APIs can change without clear documentation. However we hope to see that situation improve.
7. "Security and trust are key topics for peer-to-peer (P2P) computing. And, P2P computing is volatile."[10] We intend to develop a Pluggable Authentication Model (PAM) approach, allowing third parties add-on security modules.

6 Summary

Project V-G is an ongoing experiment to demonstrate the validity of P-Commerce. It uses the IMVC model. Currently, we have two part-time developers working on this project. Given the complexity of P-commerce, we certainly feel that our project will benefit from community involvement. Perhaps, we can align our project with other open-source efforts.[11] Taking that into consideration, we have designed the "business rule" module as a separate component allowing third party, technical and non-technical people to improve our application.

Based on this and other experiments, we feel that in a P-Commerce setting, broadcasting queries is not the most efficient mechanism for searching. We are working on a referral model, i.e. Stanley Milgram's "Small World" experiment has shown that referral can be very highly efficient - after several iterations of referrals, you can build a huge number of people involved in a social network. Plus, semantic referral may add intelligence to the searching process.

---

[10] Raymond Gao, "P2P Security and Trust", P2P Journal, September, 2003

[11] For example, "JOSE – A Java Open Source Exchange", P2P Journal, January, 2004

Performance metrics can become quite valuable for management. For a P-Commerce service provider, that may add great value. We hope that in phase three of our project, we will work on adding those functions, i.e. modeling with a unified mathematical and UML modeling techniques.[12] Finally, we have a long list of functions that we would like to have. We hope that by opening up the source code, it will encourage people to get involved in this project. Following items are on our wish list.

- Skins for the GUI application
- Enhanced security and business transaction modules
- Transaction certification modules between buyer and seller of a merchandise
- Transaction booking and logging tools that can interface with and save results to accounting software like Microsoft Money or Intuit Quicken
- Tracking new merchandise listings of friends and acquaintances
- Enhanced searching and listing capabilities, i.e. efficiently walk through the Distributed Hashtable (DHT), probabilistic filters that understand synonyms of keywords for merchandise listings, and statistical tracking of popular items, etc.
- P2P yardsale aggregation services, caching of merchandise listings, etc.

References

1. Project Venezia-Gondola's website (http://venezia-gondola.net)
2. "Project JXTA 2.0: Java ™ Programmer's Guide", Sun Microsystems, Inc. May, 2003
3. Daniel Brookshier, Darren Govoni, Navaneeth Krishnan, "JXTA: Java™ P2P Programming", SAMS Publishing, March 2002
4. Denis Urusov, "JOSE – A Java Open Source Exchange", P2P Journal, January, 2004
5. Luca Caviglione, "The "dark" side and the "force" of peer-to-peer computing saga", P2P Journal, January, 2004
6. Ben Strulo, "Middleware to Motivate Co-operation in Peer-to-Peer Systems", P2P Journal, March, 2004
7. Dimitrios Tsoumakos, Nick Roussopoulos, "Probabilistic Knowledge Discovery and Management for P2P Networks", P2P Journal, November, 2003
8. Joseph J. Bambara, Paul R. Allen, "J2EE Unleashed", SAMS publishing, 2002
9. White, K. Peterson, B. Lheureux, "New P2P Solutions Will Redefine the B2B Supply Chain", Gartner Research Note, 1 February 2003
10. R. Batchelder, "Peer Spaces: The Web Services Desktop", Gartner Research Note, 16 September 2002

[12] Raymond Gao, "Editor's Note", P2P Journal, January, 2004

11. Carl Lehmann, “P2P In B2B”, Meta Group Research Report, 19, June, 2001
12. Raymond Gao, “To P2P or P2P Too: A discussion of Peer-to-Peer and Related Technologies”, P2P Journal, July, 2003 (http://p2pjournal.com)
13. James W. Cooper “Java™ Design Patterns”, Addison-Wesley, 2001
14. Bo Leuf, “Peer to Peer: Collaboration and Sharing over the Internet”, Addison-Wesley, June 2002
15. Mark Buchanan, “Nexus”, W. W. Norton & Company, 2002