

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

การพัฒนา Peer-to-Peer แอปพลิเคชันโดยใช้สถาปัตยกรรม JXTA  
A development of Peer-to-Peer Application  
Using JXTA architecture



รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน  
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ  
ภาคฤดูร้อน ปีการศึกษา 2547  
คณะเทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

วัน เดือน ปี.....	1.5 ก.พ. 2550
เลขทะเบียน.....	02246
เลขเรียกหนังสือ.....	อท. ส 231 ก 2547
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในห้องสมุดเท่านั้น การนำออกไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือเผยแพร่ข้อมูลใดๆออกไปใช้โดยไม่ได้รับอนุญาต  
หากต้องการนำเอกสารนี้ไปใช้ กรุณาติดต่อห้องสมุดเพื่อขออนุญาตทุกครั้งที่มีการนำไปใช้

**การพัฒนา Peer-to-Peer แอปพลิเคชันโดยใช้สถาปัตยกรรม JXTA**

**A development of Peer-to-Peer Application**

**Using JXTA architecture**



รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน  
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ  
ภาคฤดูร้อนปีการศึกษา 2547  
คณะเทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การพัฒนา Peer-to-Peer แอปพลิเคชัน โดยใช้สถาปัตยกรรม JXTA
นักศึกษา	นายชนากร แสงสุข
อาจารย์ที่ปรึกษา	ผศ.ดร.โชติพัทธ์ ภรณ์วลัย
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2547

### บทคัดย่อ

รูปแบบการประมวลสินค้าในอินเทอร์เน็ตในปัจจุบันโดยส่วนมากจะมีการทำงานอยู่ในรูปแบบของ โคลเอนด์/เซิร์ฟเวอร์ แม้ว่าจะมีข้อดีอยู่หลายประการแต่ในการทำงานรูปแบบนี้ก็ยังมีข้อจำกัดอยู่เหมือนกัน เช่นจำนวนของผู้เข้าใช้ หากมีผู้ใช้ในช่วงเวลาเดียวกันจำนวนมาก จะทำให้ระบบต้องรับภาระหนัก หากโครงสร้างระบบเครือข่ายและเครื่องคอมพิวเตอร์มีสมรรถนะไม่เพียงพอ ก็จะทำให้ระบบทำงานได้ช้ามาก การพัฒนาระบบประมวลในโครงการพัฒนาระบบงานฉบับนี้ จึงได้ทดลองเขียนการประมวลสินค้าให้อยู่ในรูปแบบของ เพียร์ทูเพียร์ (Peer-to-Peer) โดยใช้สถาปัตยกรรม JXTA ( JXTA Architecture) ในการพัฒนา ทำให้ระบบประมวลเป็นระบบที่ปราศจากเครื่องเซิร์ฟเวอร์ มีการกระจายการทำงานมากขึ้น ทำให้สามารถรองรับจำนวนผู้ใช้ได้มากและยังทำให้มีความเป็นส่วนตัวของผู้ใช้มากขึ้นอีกด้วย

<b>Title</b>	A development of Peer-to-Peer application using JXTA architecture
<b>Student</b>	Mr. Thanakorn Saengsuk
<b>Advisor</b>	Asst. Prof. Dr. Chotipat Pornavalai
<b>Level of Study</b>	Master of Science in Information Technology
<b>Major</b>	Information Science
<b>Academic Year</b>	2004

### Abstract

Presently, auctioning system designs on the Internet use the client/server form. Although there are several good points of using this design, there are also many factors which limit it. A good example would be the number of users. If there are several users online at the same time, it will cause the system to work harder and carry a heavier load. If the network infrastructure and computer does not have a sufficient capacity, it will greatly reduce the system's speed. In this report, development and improvement of the auctioning system design using a Peer-to-Peer system has been tested. JXTA Architecture has been used to develop the auctioning system free from the use of a server, which allows better decentralization. It will also allow the system to support high levels of users, and at the same time, providing more privacy to each user.

## กิตติกรรมประกาศ

โครงการพัฒนาระบบงาน การพัฒนา Peer-to-Peer แอปพลิเคชัน โดยใช้สถาปัตยกรรม JXTA นี้ ได้รับการสนับสนุนและได้รับการปรึกษาเป็นอย่างดีจากหลายฝ่าย ทำให้โครงการนี้ บรรลุผลตามเป้าหมายที่วางไว้ ผู้จัดทำจึงใคร่ขอขอบพระคุณ

1. บิดา มารดา ที่ให้ความช่วยเหลือ ให้กำลังใจในการศึกษาและทำงาน
2. ผศ.ดร. โชติพัทธ์ ภรณ์วลัย อาจารย์ที่ปรึกษาโครงการ ที่ได้ประสิทธิประสาทวิชาความรู้ แก่ข้าพเจ้า และให้คำแนะนำ คำปรึกษาในการจัดทำโครงการ
3. อาจารย์ทุกท่าน ที่สั่งสอนให้ข้าพเจ้าคิดเป็น ปฏิบัติเป็น และตัดสินใจในการแก้ไข ปัญหาต่างๆ จากองค์ความรู้ที่มีอยู่ได้

ธนากร แสงสุข

ผู้จัดทำ

# สารบัญ

	หน้า
บทคัดย่อ .....	I
Abstract.....	II
กิตติกรรมประกาศ .....	III
สารบัญ.....	IV
1. บทนำ.....	7
1.1 ความเป็นมา.....	2
1.2 วัตถุประสงค์ในการพัฒนาระบบ .....	2
1.3 ขอบเขตโครงการ .....	2
1.4 ขั้นตอนการศึกษา.....	3
1.5 เครื่องมือที่ใช้ในการพัฒนาระบบ.....	3
1.6 ผลที่คาดว่าจะได้รับ .....	3
2. ทฤษฎีที่เกี่ยวข้อง .....	5
2.1 เครือข่ายเน็ตเวิร์คแบบ Peer-to-Peer.....	5
2.2 JXTA.....	6
2.3 สถาปัตยกรรมเน็ตเวิร์คของ JXTA (JXTA Network Architecture).....	17
3. การประมูลสินค้า.....	19
3.1 ความหมายของการประมูล.....	19
3.2 ประเภทของการประมูลสินค้า.....	19
3.3 ข้อดีของการรูปแบบการประมูลสินค้าออนไลน์ .....	20
3.4 การพัฒนาระบบประมูลสินค้าแบบเพียร์ทูเพียร์ .....	20
4. ความรู้เบื้องต้นในการพัฒนาโปรแกรม P2P โดยใช้ JXTA.....	22
4.1 ความต้องการของระบบ .....	22
4.2 คำนวณโหนดและติดตั้ง JXTA ไบนารี .....	22
4.3 การคอมไพล์และรันโปรแกรมที่เขียนขึ้นโดยใช้ JXTA .....	22
4.4 คลาสและเมธอดที่สำคัญในการพัฒนาระบบงาน.....	23
5. การออกแบบระบบ.....	26

5.1 Use Case .....	26
5.2 activity diagram .....	29
5.3 Class Diagram.....	30
6. การพัฒนาระบบ P2P Auction.....	32
6.1 เครื่องมือที่ใช้ในการพัฒนา .....	32
6.2 หน้าจอในโปรแกรม.....	33
7. บทสรุป.....	40
7.1 ผลการพัฒนาระบบ .....	40
7.2 ประโยชน์ที่ได้รับ .....	40
7.3 ปัญหาและอุปสรรค.....	41
7.4 ข้อเสนอแนะ.....	41
บรรณานุกรม.....	43
ภาคผนวก.....	44
ประวัติผู้เขียน.....	59

## สารบัญรูป

	หน้า
รูปที่ 2.1 JXTA Virtual Network.....	6
รูปที่ 2.2 สถาปัตยกรรมของ JXTA .....	7
รูปที่ 2.3 Pipe Endpoint .....	16
รูปที่ 2.4 รูปการส่งข่าวสารผ่านไฟร์วอลล์.....	18
รูปที่ 4.1 การคอมไพล์และรัน โปรแกรม.....	24
รูปที่ 4.2 การสร้างคำประกาศ Auction Advertisement .....	24
รูปที่ 4.3 การ Register คำประกาศแบบกำหนดขึ้นเอง.....	24
รูปที่ 5.1 P2PAuction Use Case .....	26
รูปที่ 5.2 Activity Diagram ของ P2P Auction.....	29
รูปที่ 5.3 คลาสไดอะแกรมของ P2P Auciton.....	30
รูปที่ 6.1 JXTA Configurator .....	34
รูปที่ 6.2 JXTA Configurator (ต่อ).....	34
รูปที่ 6.3 หน้าจอหลัก.....	36
รูปที่ 6.4 หน้าจอผู้ขาย.....	36
รูปที่ 6.5 หน้าจอผู้ขาย(ต่อ).....	37
รูปที่ 6.6 คำประกาศการขาย (Auction Advertisement).....	38
รูปที่ 6.7 หน้าจอหลักของผู้ประมูล.....	39

# บทที่ 1

## บทนำ

อินเทอร์เน็ตและคอมพิวเตอร์ได้เข้ามาอยู่ในส่วนหนึ่งของชีวิตประจำวันของผู้คนมากขึ้น ซึ่งในปัจจุบันเทคโนโลยีทางด้านคอมพิวเตอร์ได้เจริญก้าวหน้าไปมาก ทั้งในด้าน ฮาร์ดแวร์ ซอฟต์แวร์ และการติดต่อสื่อสาร ทำให้สามารถใช้ประโยชน์จากคอมพิวเตอร์และอินเทอร์เน็ต ได้อย่างหลากหลาย จึงมีการนำบริการต่างๆเข้ามาบนอินเทอร์เน็ตมากมาย ไม่ว่าจะเป็นการซื้อขายผ่านอินเทอร์เน็ต การพูดคุยออนไลน์ รวมไปถึงการประมูลสินค้าแบบออนไลน์ หรือที่เรียกกันว่า “E-Auction”

เว็บไซต์ประมูลชื่อดังเช่น EBay หรือแม้แต่ในบ้านเราเช่น Pramool.com มีผู้นำสินค้าเข้ามาประมูลเป็นจำนวนมากอันเป็นสาเหตุให้เซิร์ฟเวอร์และเน็ตเวิร์กที่ให้บริการต้องมีประสิทธิภาพสามารถรับภาระการประมวลผลจำนวนมากได้ จากข้อมูลสถิติในเว็บ truehits.net ประจำเดือน พฤศจิกายน 2547 พบว่าเว็บไซต์ Pramool.com มีสถิติความนิยมเป็นอันดับหก เว็บไซต์ eBay ซึ่งเป็นเว็บที่เปิดให้การประมูลชื่อดังของโลก ที่มีคนเข้าจำนวนมาก ต้องรับภาระในการให้บริการต่อวันจำนวนมาก ซึ่งจะต้องมีค่าใช้จ่ายเกิดขึ้น ฉะนั้นผู้ที่นำของมาประมูลจะต้องเสียค่าธรรมเนียมในการนำของเข้ามาประมูลด้วย

ในปัจจุบัน เทคโนโลยี Peer-to-Peer (P2P) ได้เริ่มมีบทบาทกับการติดต่อสื่อสารในอินเทอร์เน็ตเป็นอันมาก มีรูปแบบการให้บริการหลายรูปแบบ ไม่ว่าจะเป็นการแบ่งปันไฟล์ (File Sharing) หรือ ร่วมกันประมวลผล (Collaborative Processing) ซึ่งช่วยลดเวลาในการประเมินผลลงไปได้เป็นอย่างมาก เนื่องจากการนำคอมพิวเตอร์หลายๆ เครื่องมาทำงานร่วมกัน

เทคโนโลยี P2P มีแนวคิดที่ว่าทุกๆ โหนดในอินเทอร์เน็ตมีความเท่าเทียมกัน ต่างจากโมเดล Client/Server ซึ่งมีการแบ่งภาระหน้าที่การทำงานกันอย่างชัดเจน คือมี เซิร์ฟเวอร์เป็นผู้ให้บริการ ส่วนไคลเอนต์เป็นผู้รับบริการ ซึ่งหากนำ P2P เข้ามาประยุกต์ในการทำระบบประมูลออนไลน์ก็จะสามารถทำให้ไม่จำเป็นต้องมีเซิร์ฟเวอร์คอยให้บริการและไม่มีปัญหาเกี่ยวกับ ความใหญ่ของระบบ (Scalability) ทั้งนี้จะเห็นได้ว่าความสำเร็จของ P2P เนื่องมาจากความสามารถของคอมพิวเตอร์ desktop ในปัจจุบันและความเร็วเทคโนโลยีอินเทอร์เน็ต บรอดแบนด์ ทำให้แต่ละโหนด มีความสามารถในการติดต่อสื่อสารและประมวลผลได้ดีขึ้น

## 1.1 ความเป็นมา

เนื่องจากการประมูลสินค้าผ่านทางอินเทอร์เน็ตเริ่มมีความนิยมมาก ทั้งนี้เนื่องจากความสะดวกและไม่ต้องหาลูกค้า เนื่องจากลูกค้าที่สนใจจะเข้ามาหาสินค้าในเว็บเอง ทำให้เว็บไซต์ประมูลบางเว็บไซต์เกิดปัญหาทางด้านขนาดของระบบ รวมไปถึงความปลอดภัยของระบบ เนื่องจากการประมูลทางเว็บนั้น การชำระเงินส่วนมากจะต้องชำระทางบัตรเครดิต ซึ่งถ้าเซิร์ฟเวอร์ที่ให้บริการไม่ปลอดภัยก็จะทำให้ข้อมูลบัตรเครดิตอาจจะถูกโจรกรรมได้

ดังนั้นผู้เขียนจึงได้พยายามนำข้อดีของเครือข่ายเน็ตเวิร์คแบบ P2P เข้ามาประยุกต์ใช้ในการพัฒนาระบบประมูลออนไลน์ โดยจะใช้ภาษา Java และสถาปัตยกรรม JXTA ในการพัฒนา โดย JXTA ได้กำหนดโปรโตคอลและรูปแบบการติดต่อสื่อสารแบบ Peer-to-Peer ไว้ให้อย่างมากมาย โดย JXTA นั้นเป็น Open Source Project และไม่ยึดติดกับภาษาใด โดยปัจจุบันมีภาษาที่สามารถใช้ได้คือ Java และ C

## 1.2 วัตถุประสงค์ในการพัฒนาระบบ

1. เพิ่มทางเลือกใหม่ในการประมูลสินค้าผ่านอินเทอร์เน็ต โดยระบบจะทำงานโดยปราศจากเซิร์ฟเวอร์ (Server-less)
2. สร้างระบบประมูลออนไลน์ที่ไม่มีปัญหากับจำนวนของผู้ใช้
3. สร้างความปลอดภัยให้เกิดขึ้นในการให้ข้อมูลที่สำคัญระหว่างผู้ใช้ระบบ
4. เพื่อสร้างระบบประมูลออนไลน์ ที่มีความพร้อมใช้งาน (Availability) สูง ระบบไม่ขึ้นกับองค์ประกอบใดองค์ประกอบหนึ่งอย่างเดียว (Single Point Of Failure)
5. ทดลองการพัฒนา โปรแกรมแบบ Peer-to-Peer เพื่อเป็นแนวทางในการศึกษาต่อไป

## 1.3 ขอบเขตโครงการ

1. ศึกษาระบบประมูลออนไลน์ ในปัจจุบัน พร้อมทั้งศึกษาหาแนวทางที่จะใช้ P2P เข้ามาใช้แทนระบบเดิมที่เป็น โคลเอนต์/เซิร์ฟเวอร์
2. ศึกษาการ JXTA เพื่อสามารถนำมาประยุกต์ใช้กับระบบประมูลออนไลน์
3. ศึกษาการเขียน GUI โดยใช้ Java Swing เพื่อนำมารวมเข้ากับ คลาสหลักของระบบ
4. ศึกษาการนำ Log4J เข้ามาใช้ ในการ Logging ของโปรแกรม

5. การทดลองจะทำในเครือข่าย LAN เท่านั้นเนื่องจาก เพราะเนื่องจากการทำผ่านเครือข่าย Internet จะทำให้ขั้นตอนของการเขียนโปรแกรม เช่น การตรวจสอบหาข้อผิดพลาด (Debugging) เป็นไปด้วยความยากลำบาก

#### 1.4 ขั้นตอนการศึกษา

1. ศึกษาระบบการประมวลผลออนไลน์ในปัจจุบัน
2. ศึกษาทฤษฎีต่างๆเกี่ยวกับเน็ตเวิร์คแบบ P2P
3. ศึกษาสถาปัตยกรรม JXTA
4. ทดลองติดตั้ง JXTA และศึกษาการทำงานของ JXTA
5. ศึกษาความเป็นไปได้ในการพัฒนาระบบประมวลผลออนไลน์โดยใช้สถาปัตยกรรมแบบ JXTA
6. วิเคราะห์และออกแบบระบบ
7. ศึกษาโปรแกรมต่างๆที่ต้องใช้ในการพัฒนาระบบ
8. พัฒนาระบบตามที่ได้ออกแบบไว้
9. สรุปผลการดำเนินงาน

#### 1.5 เครื่องมือที่ใช้ในการพัฒนาระบบ

ในการศึกษาพัฒนาระบบงานนี้มีเครื่องมือที่ใช้คือ

1. Operating System : Windows XP Service Pack 2
2. Programming Language : Java2Sdk 1.5
3. Java IDE : Eclipse
4. JXTA Platform : JXTA J2SE 2.3.2
5. Build Tool : Ant
6. Logging Tool : Apache Log4j
7. Deployment Tool : Fat Jar (Eclipse Plug-in)

#### 1.6 ผลที่คาดว่าจะได้รับ

1. เป็นระบบประมวลผลออนไลน์ที่ไม่ใช้เซิร์ฟเวอร์ ทำให้ไม่มีค่าใช้จ่ายในการดำเนินงาน

2. เมื่อไม่มีค่าใช้จ่ายในการดำเนินงาน ทำให้ไม่ต้องเก็บค่าธรรมเนียมกับผู้ที่จะนำสินค้ามา  
ประมูล
3. สามารถทำการประมูลสินค้าได้อย่างปลอดภัย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

#### 2.1 เครือข่ายเน็ตเวิร์กแบบ Peer-to-Peer

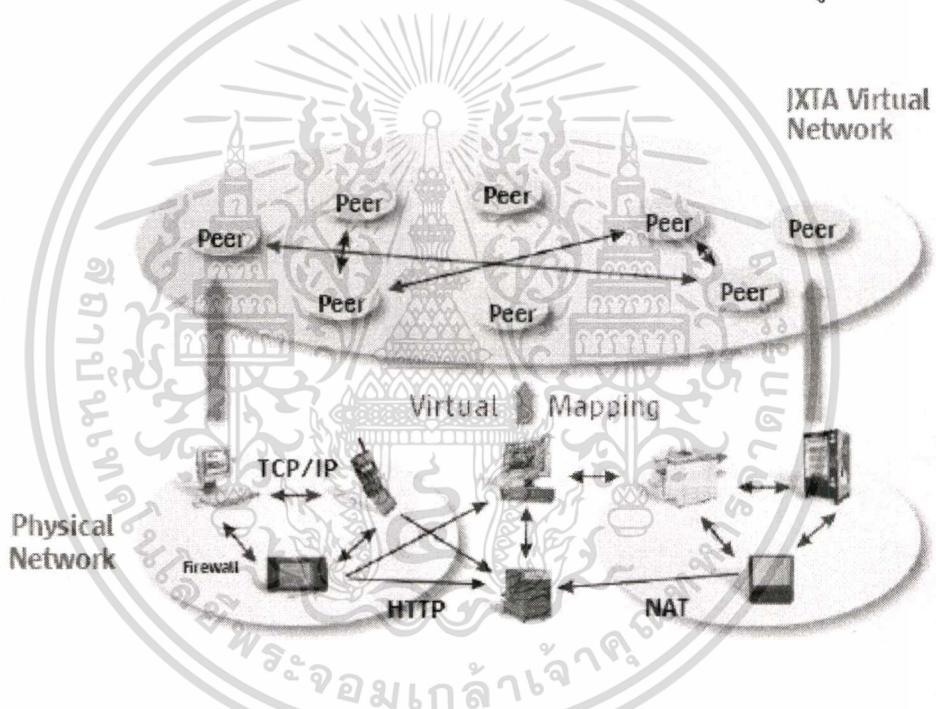
ในปัจจุบันบริการต่างๆ ในอินเทอร์เน็ตส่วนใหญ่แล้วจะมีสถาปัตยกรรมแบบ ไคลเอนต์/เซิร์ฟเวอร์ (Client/Server) ซึ่งในรูปแบบนี้ ไคลเอนต์จะทำการติดต่อเซิร์ฟเวอร์เพื่อขอรับบริการ โดยใช้โปรโตคอลในการติดต่อสื่อสารเช่น โปรโตคอลเอฟทีพี (FTP: File Transfer Protocol) ภาระในการขนส่งข้อมูล รวมทั้งการประมวลผลต่างๆ จะตกอยู่กับเซิร์ฟเวอร์ โดยที่ไคลเอนต์ใช้ทรัพยากรน้อยมาก บริการต่างๆ จำนวนมากในอินเทอร์เน็ต ไม่ว่าจะเป็น เว็บ, เอฟทีพี, จดหมายอิเล็กทรอนิกส์ ต่างมีรูปแบบการทำงานแบบ ไคลเอนต์/เซิร์ฟเวอร์ทั้งสิ้น อย่างไรก็ตาม สถาปัตยกรรมแบบนี้ก็มีข้อเสียอยู่เช่นกัน ยิ่งจำนวนไคลเอนต์มากขึ้นเท่าใด ภาระการประมวลผล และทรัพยากรต่างๆ ของเซิร์ฟเวอร์ก็มากขึ้นตามไปด้วย ทำให้เมื่อทรัพยากรต่างๆ ของเซิร์ฟเวอร์ไม่เพียงพอก็จะไม่สามารถให้บริการแก่ไคลเอนต์อื่นๆ อีกได้ แต่ข้อดีของสถาปัตยกรรมแบบนี้ก็คือ การประมวลผลที่ฝั่งไคลเอนต์มีน้อยมาก

ในขณะที่สถาปัตยกรรมแบบ ไคลเอนต์/เซิร์ฟเวอร์ได้รับความนิยม แต่ในปัจจุบันสถาปัตยกรรมแบบ Peer-to-Peer (P2P) กำลังได้รับความนิยมสูงมากเช่นกัน โดย P2P มีแนวคิดที่ว่าทุกโหนดมีความเท่าเทียมกันสามารถเป็นได้ทั้ง ไคลเอนต์และเซิร์ฟเวอร์ และไม่เหมือนแบบไคลเอนต์/เซิร์ฟเวอร์ที่พึ่งพาบริการจากเซิร์ฟเวอร์เพียงอย่างเดียว ใน P2P ทุกโหนดสามารถให้บริการและขอรับบริการได้โดยปราศจากเซิร์ฟเวอร์ศูนย์กลาง ข้อดีของ P2P ก็คือ การกระจายความรับผิดชอบในการให้บริการไปสู่ทุกๆ โหนด ทำให้ไม่มีโหนดใดโหนดหนึ่งต้องรับภาระอย่างเบ็ดเสร็จ ซึ่งทำให้ไม่มีปัญหาเกี่ยวกับขนาดที่เพิ่มขึ้นของจำนวนโหนดและไม่มีภาระพึ่งพาโหนดใดโหนดหนึ่งเป็นสำคัญ ฉะนั้นการขาดไปของโหนดใดๆ โหนดหนึ่งจะไม่มีผลกับการทำงานของระบบ อย่างไรก็ตามข้อเสียของ P2P ก็มีมากเช่นกัน เช่น การช้าช้อนของลักษณะโครงสร้างของเครือข่าย การร้องขอบริการของโหนดใดๆ สองโหนดไม่จำเป็นต้องได้รับบริการจากโหนดเดียวกัน อาจจะได้รับบริการจากโหนดที่แตกต่างกัน จึงทำให้ได้ผลลัพธ์ที่ไม่เหมือนกัน และการร้องขอใดๆ อาจจะมีการตอบกลับที่ล่าช้า หรือ อาจจะไม่ได้รับคำตอบเลย ทรัพยากรทุกประเภทในเครือข่ายแบบ P2P สามารถหายไปได้ทุกเวลาหากโหนดที่ถือทรัพยากรนั้นไม่สามารถติดต่อได้ แต่อย่างไรก็ตาม P2P ก็ยังสามารถกระจายเซิร์ฟเวอร์ต่างๆ ไปยังหลายๆ โหนดเพื่อให้เกิดความซ้ำซ้อน

ทำให้เหตุการณ์ที่ทรัพยากรขาดแคลนเกิดขึ้นได้ยากขึ้น ในปัจจุบันแอปพลิเคชันที่ใช้สถาปัตยกรรมแบบ P2P มีหลายตัวด้วยกัน เช่น การแชร์ไฟล์ทุกประเภท(File Sharing) , การส่งข้อความแบบทันที (Instant Messaging) , การประมวลผลร่วมกัน(Collaborative Processing) เป็นต้น

## 2.2 JXTA (Sun Microsystems. 2003)

เทคโนโลยี JXTA สร้างขึ้นโดยบริษัท ซัน ไมโครซิสเต็ม (Sun Microsystems) เป็นกลุ่มของมาตรฐานโปรโตคอล Peer-to-Peer ที่ทำให้อุปกรณ์ทุกชนิดในเครือข่ายเน็ตเวิร์คสามารถติดต่อสื่อสาร ทำงานร่วมกัน และ แบ่งปันทรัพยากร โหนดใน JXTA จะสร้างเครือข่ายเสมือน (Virtual Network) ไว้บนเครือข่ายเน็ตเวิร์คที่แท้จริง เพื่อซ่อนความสลับซับซ้อน (ดังรูปที่ 2.1)



รูปที่ 2.1 JXTA Virtual Network

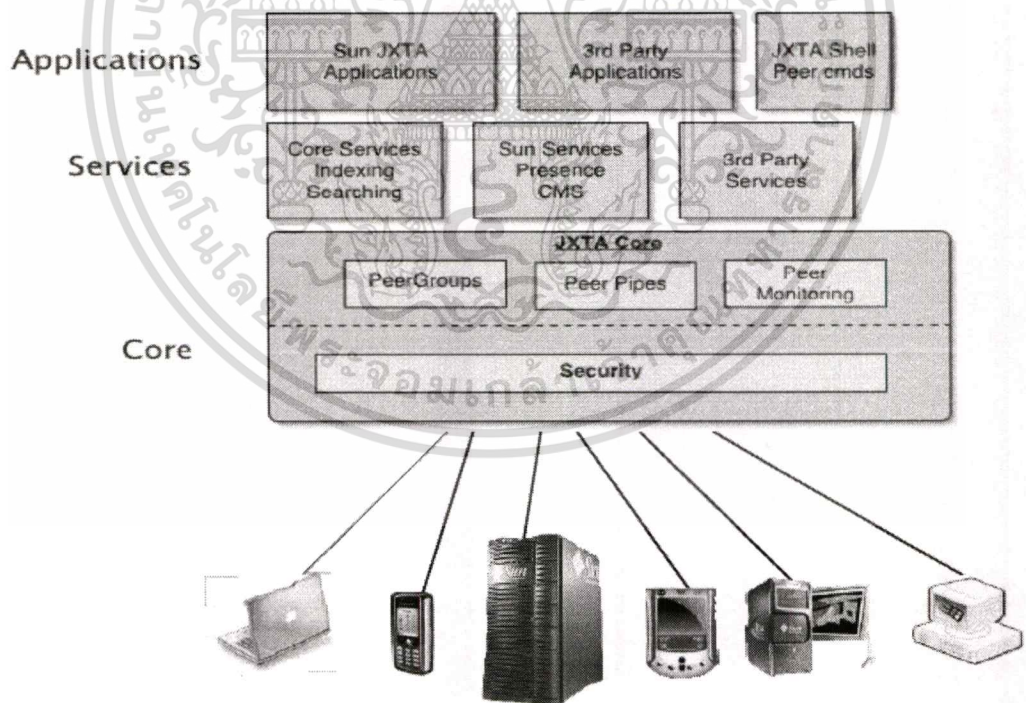
โดยในเครือข่ายเสมือนของJXTA โหนดทุกโหนดสามารถติดต่อสื่อสารกันได้ ไม่ว่าจะอยู่ที่ใด เป็นอุปกรณ์ชนิดใด ใช้ระบบปฏิบัติการใด ใช้โปรโตคอลในชั้นทรานสปอร์ตต่างกัน หรือแม้ว่า โหนดบางโหนด และ ทรัพยากรต่างๆ จะอยู่หลังไฟร์วอลล์ก็ตาม ดังนั้นการเข้าถึงทรัพยากรต่างๆ ในเน็ตเวิร์คจะไม่มีข้อจำกัดในเรื่องความไม่เข้ากันของแต่ละโหนด

การทำงานด้วยกันได้ (Interoperability) เป็นเป้าหมายสำคัญของ JXTA และ JXTA ได้มีการออกแบบให้โหนดที่ติดต่อกันสามารถค้นหาและติดต่อโหนดอื่นๆ สามารถเข้าทำกิจกรรมต่างๆ

ร่วมกัน และให้บริการต่างๆทั่วทั้งเน็ตเวิร์คแม้ว่าแพลตฟอร์มและเครือข่ายจะต่างกัน นอกจากนี้ในด้านความปลอดภัย JXTA กลไกรักษาความปลอดภัยในการติดต่อสื่อสารเช่น Transport Layer Security (TLS) , ใบรับรองดิจิทัล (Digital Certificate), หน่วยผู้ประกอบการรับรอง (Certificate Authority) เพื่อให้การติดต่อสื่อสารในเครือข่ายมีความปลอดภัยมากยิ่งขึ้น

สำหรับในด้านการพัฒนาแอปพลิเคชัน เทคโนโลยี JXTA ได้สร้างโครงสร้างพื้นฐานที่สนับสนุนการติดต่อพื้นฐานของ แอปพลิเคชันที่มีการการประมวลผลแบบกระจาย (Distributed Computing Application) และ สนับสนุนฟังก์ชันพื้นฐานของระบบโดยทั่วไปของ P2P โดยทั่วไปแล้ว แอปพลิเคชันที่อาศัยการประมวลผลแบบกระจาย ต้องอาศัยการเขียนโปรแกรมที่ยุ่งยาก ทำให้นักพัฒนาบางส่วนยากที่จะทำได้ การใช้เทคโนโลยี JXTA นักพัฒนาสามารถใช้ความคิดสร้างสรรค์ในการประดิษฐ์นวัตกรรมซอฟต์แวร์ โดยปราศจาก การสร้าง หรือ ปรับปรุง โครงสร้างพื้นฐานการทำงานของ P2P

### 2.2.1 JXTA Architecture (Sun Microsystems, 2005)



รูปที่ 2.2 สถาปัตยกรรมของ JXTA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การที่โปรโตคอล JXTA จะสามารถทำงานร่วมกันเพื่อทำงานกันอย่างเป็นระบบนั้น สถาปัตยกรรมจะต้องถูกกำหนด ดังรูปที่ 2.2

### 2.2.1.1 The JXTA Core

Core Layer เป็นเลเยอร์ที่ประกอบด้วย การอิมพลีเมนต์โปรโตคอลที่สำคัญของการทำงานในเน็ตเวิร์กแบบ P2P เซอร์วิสและองค์ประกอบที่เกี่ยวข้องในเลเยอร์นี้ ได้แก่ กลุ่มของโหนด (Peer Group), ความปลอดภัย, สถานะของโหนด (Peer Monitoring), การค้นหา (Discovery), การระบุตัวตนของโหนด (Peer ID), การประกาศของโหนด (Peer Advertisement), ไปป์ของโหนด (Peer Pipe)

### 2.2.1.2 Service Layer

ในเลเยอร์นี้ประกอบไปด้วยการบริการทางด้านเน็ตเวิร์กที่อาจจะไม่ใช่เป็นบริการที่สำคัญในเครือข่ายแบบ P2P แต่การที่มีไว้ก็จะทำให้ระบบมีประสิทธิภาพมากยิ่งขึ้น เช่น การค้นหา, ดัชนี (Indexing), การบริหารสมาชิก (Membership) เซอร์วิสต่างๆ เซอร์วิสที่ถูกสร้างขึ้นจะทำให้ P2P แอปพลิเคชัน ถูกพัฒนาได้รวดเร็วและมีประสิทธิภาพมากยิ่งขึ้น

### 2.2.1.3 Application Layer

ในแอปพลิเคชันเลเยอร์นี้จะมีส่วนประกอบคือ การพัฒนาแอปพลิเคชัน เช่น โปรแกรมสนทนา (Instant Messaging) โปรแกรมแบ่งปันเอกสารและทรัพยากร ระบบประมูลแบบกระจาย (Distributed Auction System) เป็นต้น

## 2.2.2 โปรโตคอล JXTA (Sun Microsystems. 2004)

โปรโตคอล JXTA เป็นกลุ่มของโปรโตคอล 6 โปรโตคอล ที่ออกแบบโดยเฉพาะสำหรับเน็ตเวิร์ก ที่เป็นลักษณะเฉพาะ มีลักษณะกระจาย และ การขนส่งข้อมูลต้องผ่านหลายโหนด (Multi-hop Peer-to-Peer) การใช้โปรโตคอล JXTA โหนดสามารถทำงานได้ด้วยตัวเอง และเข้าร่วมกลุ่มได้โดยไม่ต้องคำนึงถึงตำแหน่งของตนในเน็ตเวิร์ก (ปลายเครือข่าย (Edge), หลังไฟร์วอลล์, NAT, ไอพีสาธารณะ (Public IP), ไอพีส่วนตัว (Private IP)) โดยสามารถทำงานได้โดยปราศจากโครงสร้างพื้นฐานที่มีการจัดการส่วนกลาง

โปรโตคอล JXTA ได้ถูกออกแบบมาให้ไม่ใช้ทรัพยากรมาก และไม่ขึ้นกับ โปรโตคอล ในชั้นทรานสปอร์ต รวมไปถึงสิ่งแวดลอมรอบๆ โหนด โดยสามารถถูกนำไปใช้ในการพัฒนา แอปพลิเคชัน สามารถให้บริการได้แม้ว่าเน็ตเวิร์คมีลักษณะปรับเปลี่ยนอยู่ตลอดเวลา โหนดจะใช้ โปรโตคอล JXTA ในการประกาศทรัพยากรที่มันมีอยู่ และ ค้นหาทรัพยากรเน็ตเวิร์คจากโหนด อื่นๆ โหนดสามารถสร้างกลุ่มหรือเข้าร่วมกลุ่มเพื่อจะสร้างความสัมพันธ์กันแบบเฉพาะกิจ โหนด จะสนับสนุนการจัดส่งข่าวสาร (Message) ไปยังโหนดอื่นๆ เพื่อสร้างการติดต่อสื่อสารทั้งเน็ตเวิร์ค โดยโปรโตคอล JXTA จะอนุญาตให้ทุกๆ โหนดสามารถติดต่อสื่อสารกันโดยปราศจากความเข้าใจ และการจัดการเน็ตเวิร์คที่มีความสลับซับซ้อนซึ่งเป็นเรื่องปกติ

โปรโตคอล JXTA อนุญาตให้โหนดสามารถส่งผ่านข่าวสารโดยเลือกเส้นทางได้อย่าง อิสระ เพื่อไปสู่เป้าหมายซึ่งอาจจะอยู่หลังไฟร์วอลล์ ทุกๆ ข่าวสารจะเก็บข้อมูลเกี่ยวกับโหนดที่มัน เดินทางผ่านมา โดยโหนดระหว่างทางสามารถเลือกเส้นทางที่ดีที่สุดในการไปถึงเป้าหมายได้

โปรโตคอล JXTA ประกอบไปด้วยโปรโตคอล 6 โปรโตคอลที่ทำงานร่วมกันเพื่อใช้ในการ ค้นหา (Discovery) จัดการอย่างเป็นระบบ (Organization) ตรวจสอบสถานะ(Monitoring) และการติดต่อสื่อสารระหว่างโหนด(Communication)

**2.2.2.1 Peer Resolver Protocol (PRP)** เป็นกลไกที่โหนดสามารถใช้ในส่งการค้นหา (Query) โหนดใดโหนดหนึ่งหรือหลายๆ โหนด และได้รับคำตอบ (Response) ที่มาจากแต่ละโหนดหรือ หลายๆ โหนด PRP อิมพลีเมนต์โปรโตคอล Query/Response โดยที่ คำตอบจะถูกอ้างอิงว่ามาจากการค้นหาใดโดยใช้หมายเลข ID ที่อยู่ในคำตอบ การค้นหายังสามารถกำหนดเป้าหมายในลักษณะ โหนดทุกโหนดในกลุ่มใดกลุ่มหนึ่ง หรือระบุเฉพาะ โหนดใดโหนดหนึ่งได้ด้วย

**2.2.2.2 Peer Discovery Protocol (PDP)** เป็นกลไกที่โหนดสามารถที่จะประกาศ ทรัพยากรของตัวเอง และ ค้นหาทรัพยากรของโหนดอื่นๆ ได้ ทรัพยากรที่ว่ามันก็เช่น ข้อมูลกลุ่มของโหนด ข้อมูล การบริการ ข้อมูลเกี่ยวกับไปป์ ทรัพยากรต่างๆ ของทุกๆ โหนดจะถูกอธิบายและประกาศโดยใช้ คำ ประกาศ (Advertisement) ซึ่งคำประกาศนี้จะถูกสร้างโดยใช้เอกสาร XML (Extensible Markup Language)

**2.2.2.3 Peer Information Protocol (PIP)** เป็นกลไกที่สามารถให้โหนดสามารถร้องขอ ข้อมูล สถานะของโหนดอื่นๆ ได้ ซึ่งจะเป็นข้อมูลประเภทสถานะ เวลาทำงาน (Uptime) ความคับคั่งของ เน็ตเวิร์ค (Traffic Load) ความสามารถของโหนด และข้อมูลอื่นๆ ได้

**2.2.2.4 Pipe Binding Protocol (PBP)** เป็นกลไกที่ให้โหนดสามารถสร้างช่องทางติดต่อสื่อสาร แบบเสมือนได้ หรือที่เรียกว่าไปป์ (Pipe) ระหว่างโหนดหรือหลายๆ โหนดก็ได้ โดยมีไปป์เป็น พื้นฐานในการติดต่อสื่อสารระหว่างโหนด

**2.2.2.5 Endpoint Routing Protocol (ERP)** เป็นกลไกที่โหนดใช้ในการค้นหาเส้นทาง (ซึ่งอาจจะประกอบไปด้วยโหนดระหว่างทางหลายๆ โหนด) เพื่อใช้ในการส่งข่าวสารไปสู่โหนดปลายทาง เช่น หาก โหนด ก. ต้องการส่งข่าวสารไปยังโหนด ข. และเป็นที่ยอมรับว่าไม่มีเส้นทางตรงจาก ก. ไป ข. ดังนั้น โหนด ก. จำเป็นต้องหาโหนดระหว่างทางซึ่งจะต้องเป็นผู้ที่เข้าร่วมในการส่งต่อข่าวสารไปยังโหนด ข. จึงใช้โปรโตคอล ERP ในการค้นหาเส้นทาง ซึ่งหากเครือข่ายมีการเปลี่ยนแปลง ทำให้เส้นทางเก่าไม่สามารถใช้งานได้ ก็สามารถใช้ ERP เพื่อให้ได้ข้อมูลเส้นทางใหม่มาใช้ในการแทน

**2.2.2.6 Rendezvous Protocol (RPV)** เป็นกลไกที่ทำให้โหนดสามารถเลือกได้ว่าจะเป็นผู้ให้บริการในการกระจายข่าวสาร (Propagation Service) หรือไม่ ในกลุ่มของโหนด แต่ละโหนดสามารถเลือกที่จะเป็น โหนดแบบ Rendezvous หรือเป็นโหนดธรรมดาก็ได้ โปรโตคอล RPV อนุญาตให้โหนดสามารถส่งข่าวสารไปยังอินสแตนซ์ที่ยังติดต่อกันได้ โดย RPV จะถูกใช้โดย PRP และ PBP ในการที่จะกระจายข่าวสาร

แต่ละโปรโตคอลของ JXTA เป็นอิสระต่อกัน ซึ่งหมายความว่า โหนดใดๆ ก็ตามจะไม่มีข้อกำหนดว่าต้องทำการอิมพลีเมนต์ ทุกๆ โปรโตคอล สามารถเลือกที่จะอิมพลีเมนต์เฉพาะโปรโตคอลที่ใช้ได้

## 2.2.3 แนวคิดของ JXTA (JXTA Concept)

### 2.2.3.1 โหนด (Peers)

คำว่า “โหนด” ใน JXTA จะหมายถึงอุปกรณ์เน็ตเวิร์ค เช่น เซนเซอร์ โทรศัพท์ พีซี เซิร์ฟเวอร์ ที่สามารถอิมพลีเมนต์โปรโตคอลที่สำคัญของ JXTA (JXTA Core) ซึ่งก็คือโปรโตคอล PRP และ ERP ทุกๆ โหนดจะถูกอ้างถึงโดยใช้หมายเลขประจำตัวที่ไม่ซ้ำกัน โหนดมีลักษณะปกครองตัวเอง ทำงานโดยอิสระ และทำงานแบบไม่ประสานจังหวะ (Asynchronous) กับโหนดอื่นๆ อย่างไรก็ตาม โหนดบางโหนดอาจจะต้องพึ่งพาโหนดอื่นๆ ในกรณีพิเศษ เช่น ความต้องการเกตเวย์ พรอกซี หรือ เร้าเตอร์ โหนดอาจจะประกาศบริการทางเน็ตเวิร์ค และ ทรัพยากรของตัวเอง เพื่อให้โหนดอื่นๆ สามารถนำไปใช้ได้ และ โหนดอาจจะ เก็บประกาศของโหนดอื่นๆ ไว้

โดยปกติแล้ว โหนดจะถูกกำหนดให้ค้นหาโหนดอื่นๆ ด้วยตัวเอง เพื่อสร้างความสัมพันธ์แบบชั่วคราวหรือถาวรกับโหนดอื่นๆ โหนดที่ให้กลุ่มของบริการเหมือนกันมีแนวโน้มที่เป็นโหนดที่สามารถแลกเปลี่ยนกันได้ ดังนั้น โดยทั่วไปแล้ว โหนดจำเป็นที่จะต้องปฏิบัติตามสัมพันธ์กับโหนดจำนวนไม่มากนัก และจะต้องไม่มีสมมติฐานเกี่ยวกับความเชื่อถือของโหนดอื่นๆ ทั้งนี้โหนดสามารถเข้าหรือออกจากเครือข่ายได้ทุกเวลา

โหนดอาจจะประกาศหลายๆเน็ตเวิร์คอินเตอร์เฟซ โดยแต่ละอินเตอร์เฟซที่ประกาศจะเป็นจุดปลายของโหนด (Peer Endpoint) ซึ่งจุดปลายของโหนดจะเป็น URI ที่จะสามารถเจาะจงเน็ตเวิร์คอินเตอร์เฟซของโหนดใดโหนดหนึ่งได้ เช่น URI อาจจะเป็น พอร์ตของโปรโตคอลที่ซีพีและ ไอพีแอดเดรส จุดปลายของโหนดจะถูกใช้โดยโหนดเพื่อทำการสร้างการติดต่อสื่อสารแบบจุดต่อจุดกับโหนดใดโหนดหนึ่งในเครือข่าย อย่างไรก็ตามการติดต่อสื่อสารของทุกๆโหนด อาจจะไม่จำเป็นต้องเป็นการสื่อสารแบบจุดต่อจุดเสมอไป โหนดบางโหนดอาจจะต้องการโหนดระหว่างทางเพื่อที่จะส่งผ่านข่าวสารไปยังโหนดที่ถูกแบ่งแยกโดยลักษณะข้อกำหนดของเน็ตเวิร์ค เช่น NAT, ไฟร์วอลล์ เป็นต้น

### 2.2.3.2 กลุ่มของโหนด (Peer Group)

โหนดจะจัดการตัวมันเองเข้าไปอยู่ในกลุ่มของโหนด กลุ่มของโหนดคือ กลุ่มของโหนดที่มีความสนใจเหมือนกัน โดยกลุ่มของโหนดจะถูกอ้างถึงโดย หมายเลขประจำกลุ่ม (Peer Group ID) ซึ่งแรงจูงใจในการสร้างกลุ่มของโหนดขึ้นมาก็คือ

เพื่อสร้างสภาพแวดล้อมที่ปลอดภัย กลุ่มของโหนดจะถูกสร้างเพื่อสร้างขอบเขตในการเข้าถึงทรัพยากรต่างๆในกลุ่มของโหนด กลุ่มของโหนดไม่จำเป็นที่จะสะท้อน โครงสร้างเครือข่ายที่แท้จริง ในที่นี้ก็คือการแบ่งกลุ่มของเน็ตเวิร์คให้เป็นขอบเขตที่ปลอดภัยโดยปราศจากขอบเขตที่แท้จริงของเครือข่าย

เพื่อสร้างขอบเขตของสภาพแวดล้อม กลุ่มของโหนดโดยทั่วไปจะถูกสร้างขึ้นและจัดการกันเองโดยมีพื้นฐานจากความสนใจที่เหมือนกัน ไม่มีข้อกำหนดในการสร้างกลุ่มของโหนด แต่อาจจะมีกฎในการเข้าร่วมกลุ่มของโหนด ขอบเขตของกลุ่มของโหนดจะเป็นข้อกำหนดขอบเขตของการค้นหา เนื้อหาต่างๆในกลุ่ม

เพื่อสร้างขอบเขตของการตรวจสอบสถานะ กลุ่มของโหนดอนุญาตให้โหนดสามารถตรวจสอบสถานะของโหนดอื่นๆได้ เพื่อวัตถุประสงค์ใดๆก็ตาม เช่น การตรวจสอบความสามารถในการติดต่อสื่อสาร เป็นต้น

กลุ่มของโหนดสามารถให้บริการที่เรียกว่า Peer Group Service โดยโปรโตคอล JXTA ได้ กำหนดบริการหลัก (Core Service) ดังนี้คือ

**บริการการค้นหา (Discovery Service)** บริการนี้มีไว้สำหรับ ให้สมาชิกในกลุ่มของโหนดสามารถค้นหาทรัพยากรต่างๆในกลุ่มได้ เช่น โหนด, กลุ่มของโหนด และ ไปได้

**บริการสมาชิก (Membership Service)** เป็นบริการที่ถูกใช้โดยสมาชิกในกลุ่มของโหนดที่จะอนุญาตหรือไม่อนุญาต การเข้าร่วมสมาชิกของโหนดใดโหนดหนึ่ง ซึ่งอาจจะมีการคัดเลือกโหนดใดโหนดหนึ่งเป็นตัวแทนในการจะที่ผู้แทนที่จะตัดสินใจการเข้ารับสมาชิก

**บริการการเข้าใช้ (Access Service)** เป็นบริการใช้เพื่อตรวจสอบความถูกต้องของการร้องขอของโหนดหนึ่งไปยังอีกโหนดหนึ่ง โหนดที่รับการร้องขอจะขอให้โหนดที่ทำการร้องขอยื่นหนังสือรับรอง (Credential) และข้อมูลการร้องขอ ไปสู่บริการการเข้าใช้ เพื่อตรวจสอบว่าจะทำการอนุญาตหรือไม่

**บริการไปป์ (Pipe Service)** บริการนี้ใช้เพื่อจัดการ และ สร้าง ไปป์ ระหว่างโหนดในกลุ่ม

**บริการค้นหาคำตอบ (Resolver Service)** เป็นบริการที่ไว้เจาะจงการร้องขอว่าต้องไปยังโหนดใดในกลุ่ม และรวบรวมคำตอบ

**บริการสถานะ (Monitoring Service)** เป็นบริการที่อนุญาตให้โหนดสามารถตรวจสอบสถานะของโหนดอื่นๆ

ไม่จำเป็นที่กลุ่มของจะต้องมีครบทุกบริการ โดยบริการอาจจะถูกเพิ่มขึ้นภายหลังก็ได้ ซึ่งก็จะทำให้เกิดประโยชน์มากขึ้น

### 2.2.3.3 บริการทางเน็ตเวิร์ก (Network Services)

โหนดสามารถประกาศ บริการต่างๆที่มันสามารถทำได้โดยไม่จำกัด โหนดจะค้นหาบริการทางเน็ตเวิร์กโดยใช้โปรโตคอล PDP โดย โปรโตคอล JXTA สามารถเข้าใจบริการทางเน็ตเวิร์กสองระดับคือ บริการของโหนด (Peer Service) และ บริการของกลุ่มของโหนด (Peer Group Service)

บริการของโหนดจะสามารถเข้าถึงได้โดยเข้าไปขอรับบริการจากโหนดที่ประกาศบริการนั้นเท่านั้น ถ้าโหนดนั้นขาดการติดต่อ บริการนั้นก็ไม่สามารถใช้ได้ไปด้วย บริการบางอย่างอาจจะมีหลายๆอินสแตนต์ก็ได้เพื่อเพิ่มความซ้ำซ้อน โดยมีข้อกำหนดว่า แต่ละอินสแตนต์จะต้องใช้คำประกาศ(Advertisement) ที่ต่างกัน

บริการของกลุ่มของโหนด จะเป็นกลุ่มของอินสแตนต์ของบริการ ที่ทำงานอยู่ในแต่ละโหนดในกลุ่ม ถ้ามีโหนดใดโหนดหนึ่งในกลุ่มขาดการติดต่อ บริการก็จะไม่เกิดผลกระทบส่วนมากก็จะมีโหนดอื่นๆในกลุ่มสามารถให้บริการแทนได้ โดยบริการของกลุ่มนี้จะถูกประกาศโดยเป็นส่วนหนึ่งของ คำประกาศของกลุ่ม (Peer Group Advertisement)

บริการต่างๆ สามารถถูกติดตั้งไว้ก่อน หรือ สามารถโหลดผ่านเน็ตเวิร์ก กระบวนการในการค้นหา ดาวน์โหลดและติดตั้ง เหมือนกับการค้นหาเวปเพจ ได้รับเวปเพจ ติดตั้งปลั๊กอินที่ต้องใช้

ในการจะทำให้บริการนั้นทำงานจริงๆอาจจะต้องหาการอิมพลีเมนต์ที่เหมาะสมกับสภาพแวดล้อมของโหนด

#### 2.2.3.4 IDs

โหนด กลุ่มของโหนด ไปป์ เนื้อหา (Content) และทรัพยากรอื่นๆสามารถอ้างถึงได้ในโปรโตคอล JXTA ซึ่ง JXTA ID จะไม่ซ้ำกัน

#### 2.2.3.5 คำประกาศ (Advertisement)

ทรัพยากรต่างๆในเน็ตเวิร์ค เช่น โหนด กลุ่มของโหนด ไปป์ และ เซอร์วิส ต่างๆจะถูกแสดงให้เห็นโดยคำประกาศ คำประกาศเป็น ภาษาของ JXTA ที่ลักษณะส่วนของข้อมูลที่ควบคุมข้อมูลที่เป็นกลาง เพื่ออธิบายทรัพยากรต่างๆ ในข้อกำหนดพื้นฐานของ JXTA ได้กำหนดชนิดของคำประกาศไว้ดังนี้

##### คำประกาศของโหนด (Peer Advertisement)

เป็นคำประกาศที่แสดงรายละเอียดเกี่ยวกับโหนด สิ่งสำคัญสำหรับการใช้คำประกาศนี้คือการเก็บรายละเอียดที่เป็นรายละเอียดเฉพาะของแต่ละโหนด เช่น ชื่อ รหัสประจำตัว และช่องทางการสื่อสาร

##### คำประกาศของกลุ่มของโหนด (Peer Group Advertisement)

เป็นคำประกาศที่อธิบายกลุ่มของโหนดเช่น ชื่อ รหัสประจำกลุ่ม รายละเอียดของกลุ่มบริการที่มีในกลุ่ม และ พารามิเตอร์ของการบริการนั้นๆ

##### คำประกาศของโมดูลคลาส (Module Class Advertisement)

เป็นคำอธิบายของ โมดูลคลาส หน้าหลักก็คือทำการ เก็บรายละเอียดของ โมดูลคลาส เช่น ชื่อของคลาส คำอธิบายของคลาส และ รหัสของคลาส

##### คำประกาศของโมดูลสเปก (Module Spec Advertisement)

เป็นการกำหนดรายละเอียดของโมดูลสเปก หน้าหลักก็คือบอกรายละเอียดให้กับผู้ที่จะนำโมดูลสเปกไปเขียนขึ้นจริง เพื่อให้สามารถทำงานกับโมดูลสเปกนี้ได้

##### คำประกาศของโมดูลอิมพลีเมนต์เตชัน (Module Impl Advertisement)

เป็นการกำหนดการอิมพลีเมนต์ของโมดูลสเปก ซึ่งจะมีชื่อ รหัสของโมดูลสเปกที่อิมพลีเมนต์ ไลค์ แพ็กเกจ และ พารามิเตอร์ ซึ่งทั้งหมดนี้ที่จะทำให้โหนดสามารถรับข้อมูลเพื่อทำงานได้

### คำประกาศของไปป์ (Pipe Advertisement)

เป็นการกำหนดช่องทางการสื่อสารโดยใช้ไปป์ ซึ่งจะใช้โดย ไปป์เซิร์ฟวิส เพื่อสร้างไปป์ขาเข้า และไปป์ขาออก คำประกาศของไปป์จะมีรหัสของไปป์ ชนิดของไปป์ และชื่อของไปป์

### คำประกาศของกองส์เดอวูส์ (Rendezvous Advertisement)

เป็นคำประกาศเพื่อแสดงว่าโหนดนั้นเป็นโหนดชนิดกองส์เดอวูส์ ซึ่งมีคุณสมบัติคือจะเป็นโหนดที่จะแคช คำประกาศต่างๆในกลุ่มเก็บเอาไว้

คำประกาศถูกประกาศโดยกำหนดช่วงชีวิต (Lifetime) เพื่อกำหนดช่วงเวลาที่สามารถใช้งานได้ ซึ่งคำประกาศสามารถถูกประกาศซ้ำได้เพื่อให้ช่วงชีวิตนานขึ้น

### 2.2.3.6 หนังสือรับรอง (Credential)

ความต้องการที่สามารถสนับสนุนการกำหนดข้อตกลงเกี่ยวกับการเข้าใช้ทรัพยากรต่างๆ ในเน็ตเวิร์คแบบ P2P ที่มีลักษณะเฉพาะ ทำให้ต้องใช้กลไกควบคุมโดยมีฐานจากบทบาทหมายความว่า โหนดทุกโหนดจะต้องถูกอนุญาตโดยโหนดที่ได้รับการเชื่อถือแล้วเพื่อทำงานบางอย่างที่เฉพาะเจาะจง ความสัมพันธ์ระหว่างโหนดอาจจะเปลี่ยนไปอย่างรวดเร็วและนโยบายในการควบคุมการเข้าถึงต้องมีลักษณะที่มีความยืดหยุ่นที่จะอนุญาตหรือไม่อนุญาตการเข้าใช้ ความต้องการความปลอดภัยพื้นฐาน 4 ประการ คือ

**ความลับ (Confidentiality)** ต้องสามารถยืนยันได้ว่าเนื้อหาในข่าวสารที่ส่งไปจะต้องไม่ถูกเปิดอ่านโดยผู้ที่ไม่มีความเกี่ยวข้อง

**การอนุญาต (Authorization)** ต้องสามารถยืนยันได้ว่า ผู้ส่งจะต้องได้รับอนุญาตที่จะส่งข่าวสารแล้วเท่านั้น

**ความถูกต้องของข้อมูล (Data Integrity)** ต้องสามารถยืนยันได้ว่า ข่าวสารจะไม่ถูกเปลี่ยนแปลงขณะเดินทาง

**ความไม่สามารถปฏิเสธความรับผิดชอบ (Refutability)** ต้องสามารถยืนยันได้ว่า ข่าวสารที่ส่งไปจะต้องสามารถบ่งบอกผู้ส่งได้

โครงสร้างข้อมูลข่าวสารใน JXTA อนุญาตให้ แอปพลิเคชันสามารถเพิ่ม ส่วนของข้อมูลที่ใช้ควบคุมข้อมูล (Metadata) ได้ เช่น หนังสือรับรอง (Credential) คีย์สาธารณะ (Public Key) และ Digest โดย Digest ของข่าวสารจะสามารถยืนยันความถูกต้องของข่าวสารนั้นได้ นอกจากนี้ ข่าวสารนั้นยังสามารถถูกเข้ารหัสและเซ็น เพื่อให้ข่าวสารเป็นความลับและสามารถระบุผู้ส่งได้อีกด้วย

หนังสือรับรอง (Credential) เป็นชุดข้อมูลเล็กๆที่แสดงไว้ในข่าวสาร โดยจะใช้เพื่อบ่งบอกผู้ส่งและสามารถถูกใช้เพื่อเป็นการตรวจสอบสิทธิของผู้ส่งว่ามีสิทธิส่งไปยังปลายทางนั้นหรือไม่ หนังสือรับรองเป็นข้อมูลที่ต้องแนบไปทุกครั้งที่ส่งข่าวสาร ที่อยู่ของผู้ส่งที่อยู่ในข่าวสารจะถูกตรวจสอบกับที่อยู่ของผู้ส่งที่อยู่ในเอกสารรับรองว่าตรงกันหรือไม่

โปรโตคอล JXTA สามารถทำงานได้กับโปรโตคอลความปลอดภัยที่อยู่ในเลขอร์ทรานสปอด เช่น TLS ที่ทำให้การติดต่อสื่อสารระหว่างจุดสองจุดมีความปลอดภัยแม้ว่าจะเดินทางผ่านโหนดหลายๆโหนดก็ตาม

### 2.2.3.7 ไปป์(Pipe)

ไปป์เป็นช่องทางการติดต่อสื่อสารแบบเสมือนที่ใช้ในการส่งและรับข่าวสารระหว่างเซิร์ฟเวอร์หรือแอปพลิเคชัน ไปป์สร้างให้ง่ายต่อการใช้เสมือนผู้ส่งและผู้รับจดหมายซึ่งเป็นอิสระต่อที่อยู่ของโหนด และอิสระต่อของโครงสร้างของเน็ตเวิร์ก (Network Topology) ซึ่งคุณภาพในการบริการ (Quality Of Service) สามารถถูกอิมพลิเมนต์ เช่น

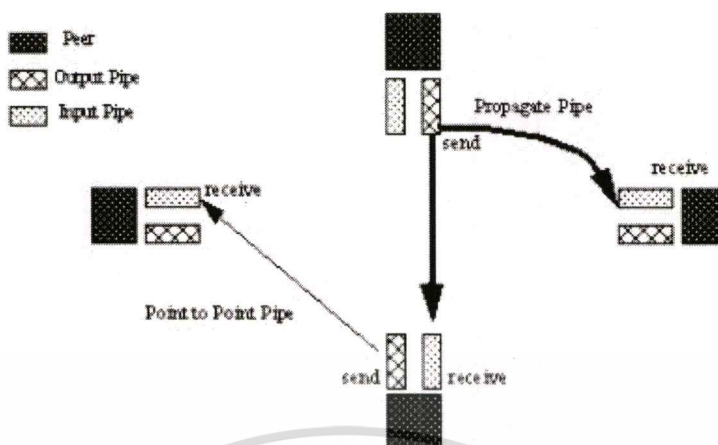
**Uni-directional asynchronous** จุดปลาย (Endpoint) จะส่งข่าวสาร โดยไม่มีการยืนยันว่าผู้รับได้รับหรือยัง

**Synchronous request-response** จุดปลายจะส่งข่าวสาร และ รับคำตอบกลับที่เกี่ยวข้องกัน การส่งข้อมูลขนาดใหญ่ (Bulk transfer) เป็นการส่งข้อมูลขนาดใหญ่ เช่น ข้อมูลที่เป็นไบนารี และสามารถเชื่อถือได้

**การส่งข้อมูลเป็นสาย (Streaming)** สามารถให้มีการควบคุมการไหลของข้อมูลให้มีประสิทธิภาพได้

**ความปลอดภัย (Secure)** สามารถทำให้เกิดความปลอดภัย ในการส่งข้อมูล

จุดปลายของไปป์ (Pipe Endpoint) จะถูกอ้างอิงถึงในลักษณะที่เป็น ไปป์ขาเข้า (Input Pipe) หรือส่วนรับ และ ไปป์ขาออกหรือส่วนส่ง (Output Pipe) ซึ่งจุดปลายของไปป์จะเชื่อมกับจุดปลาย (Endpoint) ตอนโปรแกรมถูกทำงานโดยใช้โปรโตคอล PBP ซึ่งขั้นตอนการเชื่อมกับจุดปลายประกอบไปด้วยการค้นหา และการติดต่อระหว่างจุดปลายของไปป์ 2 จุดหรือมากกว่า เมื่อข่าวสารได้ถูกส่งไปสู่ไปป์ขาออก ข่าวสารจะถูกส่งออกไปยังไปป์ขาเข้าของโหนดอื่นๆที่โหนดก็ได้ ทั้งนี้ การติดต่อสื่อสารของไปป์แบ่งออกเป็น 2 โหมดด้วยกัน ดังรูปที่ 2.3 คือ



รูปที่ 2.3 Pipe Endpoint

จุดต่อจุด (Point-to-Point) การติดต่อแบบจุดต่อจุด จะติดต่อจุดปลายของไปป์ 2 จุดเข้าด้วยกัน คือไปป์ขาเข้าโหนดหนึ่ง และ ไปป์ขาออกอีกโหนดหนึ่ง โดยไม่มีการยืนยัน (Acknowledgment) ข้อมูลอื่นๆที่ใส่เข้าไปในข่าวสาร เช่น ID จะถูกใช้เพื่อทำการเรียงข่าวสาร และ คำประกาศของไปป์ (Pipe Advertisement) มีไว้ใช้เพื่อผู้รับจะได้มีข้อมูลเกี่ยวกับไปป์ สามารถเปิดไปป์ส่งข่าวสารกลับไปหาผู้ส่งได้

### 2.2.3.8 ข่าวสาร (Message)

การส่งข้อมูลข่าวสารระหว่างไปป์ และ จุดปลาย จะส่งในลักษณะที่เป็น ข่าวสาร (Message) ข่าวสารจะมีการใส่ซอง (Envelope) เพื่อส่งชนิดของข้อมูลใดๆก็ตาม ข่าวสารจะสามารถบรรจุ หัวข้อย่อออกไปได้ไม่จำกัดซึ่งสามารถจัดการกับชนิดของข้อมูลแบบใดก็ได้

โปรโตคอล JXTA ถูกกำหนดในลักษณะที่เป็นกลุ่มของ ข่าวสาร XML แลกเปลี่ยนกันระหว่างโหนด การกำหนดข่าวสารให้อยู่ในลักษณะเอกสาร XML จะทำให้สามารถออกแบบให้โปรโตคอลสามารถใช้กับโหนดต่างชนิดในการติดต่อสารกัน และทุกๆโหนดมีอิสระในการอิมพลีเมนต์โปรโตคอลที่เหมาะสมกับความสามารถและบทบาทของมัน

### 2.2.3.9 Codats

โปรโตคอล JXTA ได้กำหนดว่าชนิดของเนื้อหา (Content) ที่แบ่งปันกัน สามารถเป็นชนิดใดก็ได้ เช่น ไฟล์เอกสาร (PDF, XML) หรือแม้แต่ ไลบรารีของภาษาจาวา ไม่มีการกำหนดขนาดของไฟล์

ไฟล์ต่างๆที่ถูกประกาศขึ้นและแบ่งปันกันระหว่างโหนดในกลุ่ม เมื่อเนื้อหาใดๆก็ตามถูกแบ่งปัน เนื้อหานั้นๆจะถูกอ้างถึงโดย JXTA ID ซึ่ง การรวมกันระหว่าง เนื้อหา และ JXTA ID จะ

ถูกเรียกว่า Codat ซึ่ง Codat จะเป็นของกลุ่มของโหนด หากเนื้อหาเดียวกัน ถูกประกาศไว้ต่างกลุ่มกันก็จะมี Codat 2 Codat

แต่ละ Codat จะถูกอ้างอิงโดยใช้ JXTA ID และสามารถถูกทำสำเนาไปสู่โหนดอื่นๆในกลุ่ม ซึ่งแต่ละสำเนาจะมี Codat และ เนื้อหาที่เหมือนกัน การทำสำเนา Codat ในกลุ่มก็เพื่อช่วยให้เนื้อหาสามารถใช้งานได้มากขึ้น เช่น ถ้าเนื้อหาถูกสำเนาอยู่ในโหนดสองโหนด โหนดเพียงโหนดเดียวที่จำเป็นที่จะต้องยังคงอยู่ในกลุ่มนั้น

## 2.3 สถาปัตยกรรมเน็ตเวิร์กของ JXTA (JXTA Network Architecture) (Sun Microsystems. 2005)

### 2.3.1 โครงสร้างของเน็ตเวิร์ก (Network Organization)

เน็ตเวิร์กของ JXTA มีลักษณะที่เกิดขึ้นแบบเฉพาะกิจ (Ad hoc) และสามารถปรับเปลี่ยนทำให้เหมาะสมได้ง่าย การเชื่อมต่ออาจจะเป็นการชั่วคราว (Transient) และ ข้อมูลที่ส่งกันระหว่างโหนดนั้นไม่สามารถคาดการณ์เส้นทางได้ โหนดสามารถเข้าและออกเน็ตเวิร์กได้ตลอดเวลา และเส้นทางก็สามารถเปลี่ยนได้ตลอดเวลาเช่นกัน โหนดสามารถมีรูปแบบได้หลายรูปแบบตามใดที่ยังคงเชื่อมต่อกับเน็ตเวิร์กของ JXTA ซึ่งโหนดแต่ละชนิดสามารถอธิบายได้ดังนี้

#### 2.3.1.1 Minimal edge peer

โหนดประเภทนี้จะสามารถส่งและรับข่าวสาร (Message) ได้ แต่จะไม่ทำการเก็บ (Cache) คำประกาศ หรือ ข่าวสารเกี่ยวกับเส้นทางสำหรับโหนดอื่นๆ โหนดที่ทำงานอยู่บนอุปกรณ์ที่มีทรัพยากรจำกัดส่วนมากจะทำงานในโหนดนี้ เช่น PDA โทรศัพท์มือถือ

#### 2.3.1.2 Full feature edge peer

โหนดประเภทนี้จะสามารถส่งและรับข่าวสารได้เช่นกัน แต่จะทำการเก็บคำประกาศต่างๆเอาไว้ เพื่อตอบหากมีการส่งคำร้องขอมาจากโหนดอื่นๆ

#### 2.3.1.3 Rendezvous peer

โหนดประเภทนี้จะคล้ายกับ โหนดที่เป็นแบบ Full Feature Edge Peer แต่จะส่งต่อคำร้องขอต่อด้วยหากตัวเองไม่สามารถตอบได้ เพื่อช่วยในกระบวนการการค้นหา (Discovery)

### 2.3.1.4 Relay Peer

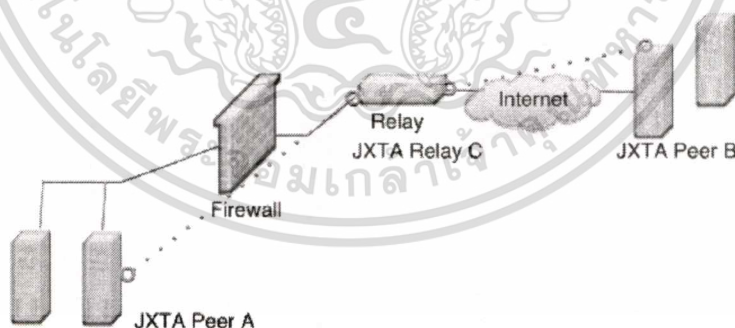
โหนดประเภทนี้จะเก็บข้อมูลเกี่ยวกับเส้นทางในการส่งข้อมูล หากโหนดในโหนดหนึ่งไม่สามารถทราบเส้นทางการส่งข้อมูลไปยังอีกโหนดหนึ่ง จะส่งการร้องขอมาถามโหนดที่ทำหน้าที่เป็น Relay peer นอกจากนี้ Relay peer ยังจะส่งข่าวสารต่อให้กับโหนดที่อยู่ หลัง NAT (Network Address Translation) อีกด้วย

### 2.3.2 Firewall and NAT

โหนดที่อยู่หลัง ไฟร์วอลล์ สามารถส่งข่าวสาร ไปยังโหนดที่อยู่ภายนอกไฟร์วอลล์ได้ แต่โหนดที่อยู่ข้างนอกไฟร์วอลล์จะไม่สามารถสร้างการเชื่อมต่อเข้ามายังโหนดที่อยู่หลังไฟร์วอลล์ได้ การที่จะทำให้โหนดสามารถติดต่อกันผ่านไฟร์วอลล์ได้ จะต้องมีลักษณะดังนี้

- ต้องมีโหนดในกลุ่มที่อยู่หลังไฟร์วอลล์สามารถติดต่อกับโหนดที่อยู่ภายนอกไฟร์วอลล์ได้
- โหนดที่อยู่ทั้งข้างหลังและภายนอกไฟร์วอลล์จะต้องสนับสนุนการใช้โปรโตคอล HTTP
- ไฟร์วอลล์จะต้องอนุญาตให้ทราฟฟิก HTTP ผ่านได้

ภาพที่ 2.3 จะแสดงการติดต่อสื่อสารในสภาพแวดล้อมดังที่กล่าวไปข้างต้น โดย โหนด A และ B จะทำการส่งข้อมูลข่าวสารระหว่างกัน แต่มีไฟร์วอลล์ที่ป้องกันการติดต่อสื่อสารแบบโดยตรง โหนด A จึงทำการติดต่อ Peer C โดยใช้โปรโตคอล HTTP เพราะสามารถวิ่งผ่านไฟร์วอลล์ได้ โหนด C ก็จะติดต่อกับ โหนด B ผ่านทางโปรโตคอล TCP/IP ทำให้เกิดช่องทางการสื่อสารแบบเสมือนระหว่าง โหนด A และ โหนด B



รูปที่ 2.4 รูปการส่งข่าวสารผ่านไฟร์วอลล์

## บทที่ 3

### การประมูลสินค้า

#### 3.1 ความหมายของการประมูล

การประมูลเป็นกระบวนการของการซื้อการขายสินค้า มีผู้ขายที่ต้องการนำสินค้ามาประมูลขายและมีผู้ซื้อที่สามารถเสนอให้ราคา (Bid) ผู้ซื้อที่ให้ราคามากที่สุดจะได้สิทธิในการซื้อสินค้านั้นไป อย่างไรก็ตามการประมูลบางประเภท ผู้ขายสามารถทำการกำหนดราคาสำรอง (Reserve Price) หากราคาสูงสุดที่ผู้ซื้อเสนอนั้นยังน้อยกว่าราคาสำรอง ผู้ขายสามารถยกเลิกการประมูลสินค้านั้นๆ ได้

#### 3.2 ประเภทของการประมูลสินค้า

การประมูลสินค้าในปัจจุบันไม่ว่าจะเป็นทางด้านธุรกิจวัตถุโบราณ (Antique) ของสะสม (Collectibles) สามารถแบ่งประเภทของการประมูลได้เป็นดังนี้

##### 3.2.1 การประมูลแบบอังกฤษ (English auction)

การประมูลในรูปแบบนี้เป็นรูปแบบการประมูลที่คนส่วนใหญ่เข้าใจ คือเมื่อเริ่มการประมูล ผู้ขายจะกำหนดราคาเริ่มต้นไว้ เรียกว่า ราคาสำรอง (Reserve Price) และ ราคาที่สามารถขายได้ (Buy-out Price) ผู้ซื้อจะทำการเสนอราคาต่อรองแข่งขันไปเรื่อยๆ ผู้ขายจะขายให้กับผู้ซื้อที่เสนอราคาที่ไม่ผู้ซื้อรายอื่นให้ราคามากกว่า หรือ ขายให้กับผู้ซื้อที่ให้ราคาที่สูงกว่าราคาที่สามารถขายได้แล้ว นอกจากนั้นผู้ขายมีสิทธิที่จะไม่ขายสินค้าหากผู้เสนอราคาที่สูงที่สุดมีจำนวนเงินน้อยกว่าราคาสำรองที่ผู้ขายได้กำหนดไว้

##### 3.2.2 การประมูลรูปแบบเนเธอร์แลนด์ (Dutch auction)

การประมูลแบบนี้เป็นการประมูลแบบเก๋าคั้งเดิม ผู้นำสินค้าเข้ามาประมูลจะบอกราคาสูงในตอนเริ่มต้น จากนั้นจะค่อยลดลงมาเรื่อยๆ การเสนอขายจะมีขึ้นเมื่อ มีผู้เสนอซื้อเข้ามา หรือราคาที่เสนอได้ลดลงต่ำกว่าที่ได้กำหนดไว้ตั้งแต่เริ่มประมูล การประมูลชนิดนี้เหมาะกับการประมูลสินค้าจำนวนมาก ทำให้เกิดความรวดเร็วมากยิ่งขึ้น

##### 3.2.3 การประมูลแบบปิดราคาสูงสุด (Sealed first-price auction)

การประมูลแบบนี้ผู้ร่วมประมูล จะทำการเสนอราคาแก่ผู้ขายพร้อมๆกัน โดยที่ไม่มีผู้ใดทราบการเสนอราคาของผู้ซื้อรายอื่น ผู้ที่ชนะการประมูลคือผู้ที่ให้ราคาสูงสุด

### 3.2.4 การประมูลแบบปิดราคาอันดับสอง (Sealed second-price auction)

การประมูลในลักษณะนี้เหมือนกันกับการประมูลแบบปิดราคาสูงสุดทุกประการยกเว้น ผู้ชนะในการประมูลไม่ต้องจ่ายราคาสูงสุดตามที่ตัวเองชนะการประมูล แต่จะจ่ายเงินจำนวนเท่ากับจำนวนเงินที่มีผู้เสนอราคาเป็นอันดับที่สอง

## 3.3 ข้อดีของการรูปแบบการประมูลสินค้าออนไลน์

### 3.3.1 ไม่จำกัดเวลา

การประมูลสินค้าออนไลน์สามารถมีขึ้นได้ตลอด 24 ชั่วโมง สินค้าที่นำเข้ามาประมูลโดยมากจะมีระยะเวลาการประมูลระหว่าง 1-10 วัน(หรือแล้วแต่ผู้ขาย) ทำให้เพิ่มเวลาให้กับผู้ซื้อในการค้นหา ตัดสินใจ และทำการเสนอราคา ความสะดวกนี้เองทำให้สินค้ามีจำนวนผู้สนใจซื้อมากขึ้น

### 3.3.2 ไม่จำกัดสถานที่

ผู้ซื้อและผู้ขายสามารถเข้าร่วมประมูลได้จากทุกๆที่ในโลก ที่สามารถเข้าสู่อินเทอร์เน็ตได้ สิ่งเหล่านี้ทำให้การประมูลสามารถเข้าถึงได้ง่าย และลดค่าใช้จ่ายในการประมูล จำนวนผู้ขายและจำนวนผู้ซื้อก็มากขึ้นตามไปด้วย ผู้ขายไม่จำเป็นต้องส่งสินค้าไปยังสถานที่ประมูล จึงทำให้ลดภาระค่าใช้จ่าย ทำให้จำนวนเงินที่ต้องการของผู้ซื้อน้อยลงไปด้วย

### 3.3.3 มีผู้ซื้อจำนวนมาก

เนื่องจากการประมูลออนไลน์ทำให้ได้สินค้าในราคาที่ต่ำกว่าท้องตลาด ความหลากหลายในสินค้าที่มีการนำเข้ามาประมูล และการเข้าถึงได้ง่ายของการประมูลในรูปแบบออนไลน์นี้ ทำให้มีผู้ซื้อจำนวนมากเข้ามาค้นหาสินค้าที่ต้องการ

### 3.3.4 มีผู้ขายจำนวนมาก

เนื่องจากมีผู้ซื้อจำนวนมาก โอกาสที่จะสามารถขายสินค้าได้ก็มีมากขึ้นตามไปด้วย รวมทั้งค่าใช้จ่ายในการขายก็น้อยกว่าหากต้องนำสินค้าไปประมูลเพราะต้องมีค่าเดินทาง ค่าดูแลสินค้าต่างๆ ทำให้เกิดผู้ขายจำนวนมาก

## 3.4 การพัฒนาระบบประมูลสินค้าแบบเพียร์ทูเพียร์

การประมูลสินค้าแบบเพียร์ทูเพียร์ที่ผู้เขียนได้พัฒนาขึ้น กระบวนการประมูลนั้นมีลักษณะคล้ายกับการประมูลสินค้าแบบอังกฤษ คือ ผู้เข้าประมูลที่เสนอราคามากที่สุดจะเป็นผู้มีสิทธิ์ได้

สินค้านั้นไป จะต่างกันตรงที่การเสนอราคานั้น จะต้องเสนอราคามากกว่าจำนวนที่ผู้ขายได้กำหนดเอาไว้ตั้งแต่ต้นแล้ว ทั้งนี้ทำให้กระบวนการประมูลเป็นไปได้อย่างรวดเร็วมากยิ่งขึ้น ส่วนระยะเวลาในการประมูลนั้น เป็นแบบไม่จำกัดทั้งนี้เนื่องจากการตัดสินใจของตัวผู้ขายเอง เนื่องจากข้อจำกัดทางด้านเทคนิคในการพัฒนา ทำให้การเก็บสถานะของการประมูลทำได้ยาก เพราะการทำงานของระบบมีลักษณะแบบกระจาย (Decentralize) ลักษณะของการประมูล จึงต้องเป็นการประมูลที่เป็นลักษณะแบบต่อเนื่อง มีการเห็นการเปลี่ยนอย่างทันทีหากมีผู้ประมูลเข้ามา ซึ่งลักษณะเช่นนี้คล้ายกับการประมูลจริงเป็นอย่างมาก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ความรู้เบื้องต้นในการพัฒนาโปรแกรม P2P โดยใช้ JXTA

#### 4.1 ความต้องการของระบบ

ระบบที่สามารถพัฒนาและติดตั้งโปรแกรมที่ใช้ JXTA ในการพัฒนาได้ จะต้องสนับสนุนการทำงานของ Java Runtime Environment (JRE) หรือ Software Development Kit (SDK) 1.4.1 หรือรุ่นที่สูงกว่า การทำงานในสถานะแวดล้อมนี้มีในระบบปฏิบัติการ ตั้งแต่ Sun Solaris , Microsoft Windows 95/98/2000/ME/NT/XP/2003/NT 4 , Linux และ Mac OS ซึ่ง JRE สามารถดาวน์โหลดได้ที่ <http://java.sun.com/j2se/download.html>

#### 4.2 ดาวน์โหลดและติดตั้ง JXTA ไบনারี่

ในการพัฒนาโปรแกรมจะต้องใช้ไลบรารี JXTA ซึ่งจะต้องดาวน์โหลดที่ <http://download.jxta.org/index.html> โดยรูปแบบของไลบรารีของ JXTA จะมีด้วยกัน 2 แบบคือ

##### 4.2.1 Release Build

รูปแบบของไลบรารีแบบนี้ก็คือ จะเป็นไลบรารี ที่มีการทดสอบอย่างแน่นอนแล้วว่ามีประสิทธิภาพในการทำงาน และเหมาะกับนักพัฒนาโปรแกรมที่เพิ่งเริ่มหัดพัฒนา JXTA หรือนักพัฒนาที่ต้องการความเสถียรและแน่นอนของระบบ

##### 4.2.2 Nightly Build

รูปแบบของไลบรารีแบบนี้จะเป็นไลบรารีที่ มีความใหม่ที่สุด เหมาะกับนักพัฒนาที่ต้องการทดลองความสามารถใหม่ๆ ของ JXTA

#### 4.3 การคอมไพล์และรันโปรแกรมที่เขียนขึ้นโดยใช้ JXTA

การคอมไพล์และรันโปรแกรมโดยใช้จาวาคอมไพเลอร์นั้น จะต้องมีการระบุตำแหน่งที่ตั้งของ ไลบรารี JXTA ที่ใช้ ดังเช่นในรูปที่ 4.1

```

Compilation Command
C:\javac -classpath .\lib\jxta.jar HelloJxta.java

Run Command
C:\java -classpath .\lib\jxta.jar;.\lib\log4j.jar;.\lib\bcprov-jdk14.jar;
HelloJxta.jar

```

รูปที่ 4.1 การคอมไพล์และรันโปรแกรม

เมื่อโปรแกรมที่เขียนขึ้นโดย JXTA ถูกสั่งให้ทำงาน ก็จะมีการสร้างไคลเรททอรีที่จะต้องใช้เก็บคำประกาศต่างๆ ซึ่งโดยดีฟอลท์แล้วจะถูกเก็บในไคลเรททอรี .jxta แต่เราสามารถกำหนดให้เป็นอย่างอื่นได้ โดยให้เพิ่มออปชัน `-DJXTA_HOME="Alternate_DIR"` ต่อท้าย โดย `"Alternate_DIR"` เป็นไคลเรททอรีที่ถูกกำหนดขึ้น

#### 4.4 คลาสและเมธอดที่สำคัญในการพัฒนาระบบงาน

##### 4.4.1 คลาส PeerGroupFactory และ PeerGroup

Statement ที่จะทำให้ JXTA แสดง Configurator ออกมาคือ `PeerGroupFactory.newNetPeerGroup ()` ซึ่งจะได้ Reference ของ World Peer Group โดย Reference นี้จะเป็น Reference ของ คลาส PeerGroup ซึ่งจะต้องนำไปใช้ในการขอบริการต่าง ซึ่งจะอยู่ในคลาส PeerGroup เช่น บริการในการค้นหา (Discovery Service) สามารถขอได้โดยใช้เมธอด `getDiscoveryService ()` และบริการไปป์โดยใช้เมธอด `getPipeService ()` เป็นต้น

##### 4.4.2 คลาส AdvertisementFactory

คลาสนี้เป็นคลาสที่เกี่ยวข้องกับการสร้างคำประกาศทุกประเภทใน JXTA โดยคลาสนี้จะมี static เมธอดชื่อ `newAdvertisement` ใช้สำหรับสร้างคำประกาศทุกประเภท โดยการเลือกจะทำให้เป็นคำประกาศใดจะใช้การ casting เช่น หากจะสร้างคำประกาศ `AuctionAdvertisement` ก็จะต้องใช้คำสั่งดังในรูปที่ 4.2

```
AuctionAdv = (AuctionAdvertisement) AdvertisementFactory.newAdvertisement
(AuctionAdvertisement.getAdvertisementType ());
```

#### รูปที่ 4.2 การสร้างคำประกาศ Auction Advertisement

หากเป็นคำประกาศที่ผู้พัฒนากำหนดขึ้นเอง (Custom Advertisement) จะต้องมีการ register เพื่อให้ JXTA รู้จักและบริการการค้นหา (Discovery Service) สามารถใช้การได้โดยใช้คำสั่งดังรูปที่ 4.3 โดยตัวอย่างการเขียนคลาสที่เป็นการสร้างคำประกาศที่กำหนดขึ้นเองจะแทรกอยู่ในภาคผนวก ก

```
AdvertisementFactory.registerAdvertisementInstance
(AuctionAdvertisement.getAdvertisementType (), new AuctionAdvImpl.Instantiator ());
```

#### รูปที่ 4.3 การ Register คำประกาศแบบกำหนดขึ้นเอง

##### 4.4.3 คลาส Discovery Service

คลาสนี้เป็นคลาสที่เกี่ยวข้องกับการการประกาศ (Publish) คำประกาศ และการค้นหา คำประกาศ เมธอดที่สำคัญอีกเมธอดหนึ่งในการพัฒนาระบบงานนี้คือ เมธอด flush ซึ่งเนื่องจากระบบประมวลจะต้องมีการอัปเดตของคำประกาศอยู่เสมอ ดังนั้น การจะเก็บคำประกาศไว้ในแคชของเครื่องนั้นเป็นสิ่งที่ต้องหลีกเลี่ยง เพราะการค้นหาคำประกาศของโหนดอื่นๆ อาจจะถูกโหนดที่มีคำประกาศที่ล้าสมัยตอบกลับไป ทำให้คำประกาศที่ได้ไปนั้นเป็นคำประกาศที่ล้าสมัย ฉะนั้นเมื่อโหนดใดโหนดหนึ่งทำการค้นหาคำประกาศ เมื่อได้รับคำประกาศที่ต้องการแล้วจะต้องทำการลบทิ้งทันทีเพื่อป้องกันปัญหาข้างต้น

##### 4.4.4 คลาส SwingWorker

คลาสนี้ไม่ได้อยู่ JXTA แต่เนื่องจากมีความสำคัญในการพัฒนา JXTA เนื่องจากว่าโปรแกรมที่พัฒนาโดย JXTA มีลักษณะที่เป็น Multi Threading ซึ่งการเขียนให้เข้ากับ GUI เช่น Swing ใน java ไม่ว่าจะเป็นการ ค้นหา การส่งข่าวสารหากัน โอเปอเรชันทั้งหลายเหล่านี้ซึ่งไม่เกี่ยวกับการทำงานของ GUI และเป็นงานที่ต้องใช้เวลาในการประมวลผลพอสมควร หากเขียนขึ้นธรรมดาตรงเมธอดที่รับ Event ของ Swing จะไม่มีการทำงานเกิดขึ้น วิธีการแก้คือ ต้องสร้าง Thread

ขึ้นมาเพื่อรองรับโอเปอเรชันทั้งหลายเหล่านี้ โดยทางเว็บไซต์ของจาวา ได้มีการสร้างคลาส SwingWorker เพื่อสร้าง Thread ที่ไม่ยุ่งยากเหมือนการสร้าง Thread ธรรมดา และทางเว็บไซต์ได้แนะนำให้ใช้คลาสนี้แทนการสร้าง Thread แบบธรรมดาเพื่อลดข้อผิดพลาดที่อาจจะเกิดขึ้น

#### 4.4.5 คลาส StructuredDocumentFactory และ StructureDocument

สองคลาสนี้เป็นสองคลาสหลักที่จะสร้างเอกสารที่มีลักษณะเป็นโครงสร้าง เช่น เอกสาร XML เนื่องจากการส่งข่าวสารในระบบเครือข่ายที่สะดวกที่สุด เช่นในระบบงานนี้ รูปแบบที่ใช้ในการส่งข่าวสารในการเสนอราคา หรือ เรียกดูรูปภาพ ก็จะใช้ข่าวสารในรูปแบบของ XML ในการติดต่อสื่อสารเป็นหลัก

#### 4.4.6 คลาส JxtaSocket และ JxtaServerSocket

สองคลาสนี้เป็นคลาสที่สำคัญเพราะเป็นกลไกหลักที่ใช้ในการติดต่อสื่อสารของระบบงานนี้ เนื่องจากการส่งรูปภาพและข่าวสารที่ใช้ติดต่อกันมีขนาดใหญ่ คลาสสองคลาสนี้ในการติดต่อสื่อสารจะมีการแบ่งเป็นแพคเกจย่อยๆ โดยอัตโนมัติ ทำให้ผู้พัฒนาสะดวกไม่ต้องกังวลว่าขนาดของข้อมูลที่มากที่สุดที่สามารถส่งได้ในระบบเครือข่ายเป็นเท่าไร

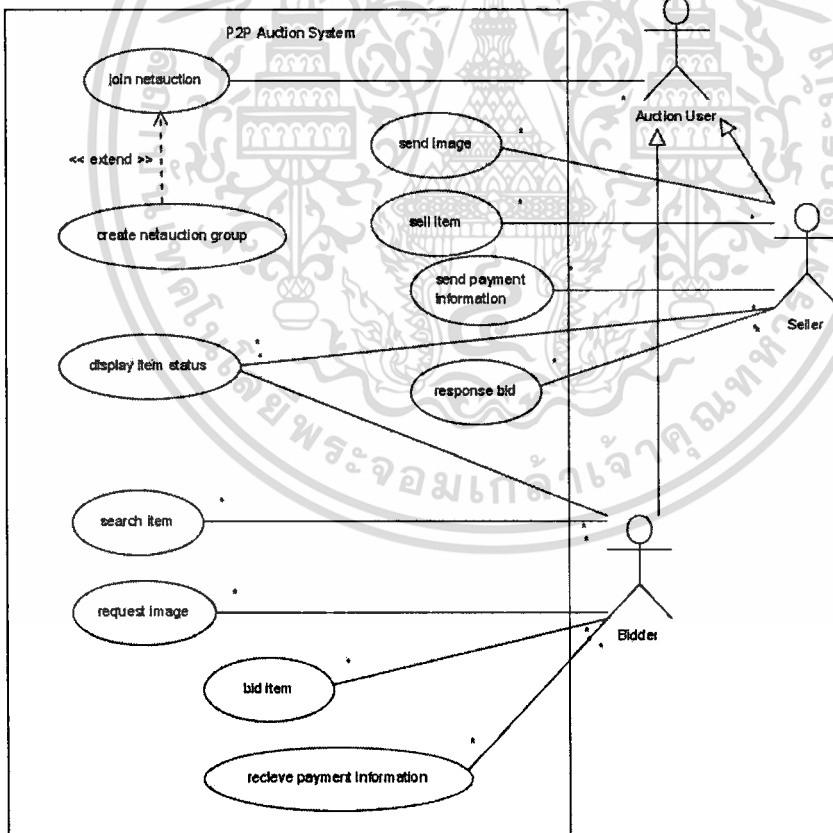
# บทที่ 5

## การออกแบบระบบ

ในการออกแบบระบบ P2P Auction System มีการออกแบบระบบโดยสามารถแสดงโดยใช้ โมเดลต่างๆของ UML เพื่อให้เห็นในต่างมุมมองของระบบ

### 5.1 Use Case

การออกแบบระบบประมูลสินค้าแบบ P2P โดยใช้โมเดล Use Case เป็นไปดังรูปที่ 3.1 โดยมีรายละเอียดดังนี้



รูปที่ 5.1 P2PAuction Use Case

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.1.1 Actor

#### 5.1.1.2 ผู้ขาย (Seller)

ผู้ขาย หรือ ผู้ที่นำของมาประมูล สามารถนำของมาประมูลได้โดยให้รายละเอียดสินค้าที่ต้องการขาย ได้

#### 5.1.1.3 ผู้ประมูล (Bidder)

ผู้ประมูลจะเป็น ผู้ที่ค้นหาสินค้า เพื่อจะประมูลให้ได้สินค้านั้นๆ

#### 5.1.1.4 ผู้ใช้ระบบ (Auction User)

ผู้ใช้ระบบ เป็น Generalization ของ ผู้ประมูล และ ผู้ขาย

### 5.1.2 Use Case Description

#### 5.1.2.1 join netauction

Join netauction เป็น use case ที่มีหน้าที่ในการรับ user name และ password รวมทั้ง ข้อมูลต่างๆ เช่น ไอพีแอดเดรส พอร์ตที่สามารถใช้ติดต่อ จากนั้นนำผู้ใช้เข้าสู่กลุ่มที่ได้สร้างไว้ ซึ่งการเข้ากลุ่ม จะสามารถหาทรัพยากรต่างๆของกลุ่มในเครือข่ายได้

#### 5.1.2.2 create netauction group

ในกรณีที่ยังไม่มีกลุ่ม net auction อยู่ในแคชของโหนดนั้น ก็จะต้องมีการสร้างกลุ่มชื่อ net acution ขึ้นมาใหม่ โดยเพื่อให้การสร้างกลุ่มจะมีรหัสประจำของกลุ่ม ซึ่งได้ถูกกำหนดไว้ตายตัว เพราะการติดต่อสื่อสารใน jxta อิงตามกลุ่มเป็นสำคัญ เปรียบเสมือนขอบเขตของการติดต่อสื่อสาร

#### 5.1.2.3 sell item

Sell item เป็น use case ที่ใช้ในการรับข้อมูลสินค้าที่ต้องการขาย จากผู้ขาย โดยผู้ขาย จะต้องกรอกรายละเอียดของสินค้าที่ต้องการนำมาขาย เช่น ชื่อสินค้า คำอธิบายสินค้า ราคาเริ่มต้น รูปภาพสินค้า จากนั้น ก็ต้องนำไปสร้างคำประกาศขาย (Auction Advertisiement) และส่งคำประกาศออกไป

#### 5.1.2.4 display item status

Use case นี้มีไว้เพื่อ แสดงรายละเอียดต่างๆของ สินค้าต่างๆ เช่น รูปภาพ ราคาเริ่มต้น ราคาที่กำลังประมูลกันอยู่ ผู้ที่ให้ราคาмаกที่สุดในขณะนี้ เป็นต้น

### 5.1.2.5 request image

Request image เป็น use case ที่ใช้สำหรับ ร้องขอรูปภาพของ ผู้ซื้อ จากผู้ขาย

### 5.1.2.6 search item

Search item เป็น use case ที่ใช้สำหรับ รับคำค้นหาจากผู้ซื้อ เพื่อนำมาค้นหาใน เครื่องข่าย JXTA โดยการค้นหานั้น จะมีการส่งคำค้นหาออกไปเฉพาะในกลุ่ม Netauction ที่สร้างไว้ ก่อนหน้านี้แล้วเท่านั้น

### 5.1.2.7 send image

Send image เป็น use case ที่ใช้สำหรับส่งรูปภาพจาก ผู้ขายไปสู่ผู้ซื้อ

### 5.1.2.8 bid Item

Bid item เป็น use case ที่ใช้สำหรับส่งการเสนอราคาของผู้ซื้อ ไปยังผู้ขาย

### 5.1.2.9 response bid

Response bid เป็น use case ที่จะต้องตรวจสอบการประมูลของผู้ซื้อ โดยต้องตรวจสอบว่า มีค่ามากกว่าราคาเริ่มต้น หรือ มากกว่า ราคาที่มีผู้เสนอสูงสุดขณะนี้หรือไม่ หากมีค่ามากกว่า ก็ จะต้องทำการ ปรับปรุงค่าประกาศขายใหม่ หากมีค่าน้อยกว่า ก็จะต้องส่งข้อผิดพลาดแจ้งให้ผู้ซื้อ ทราบ

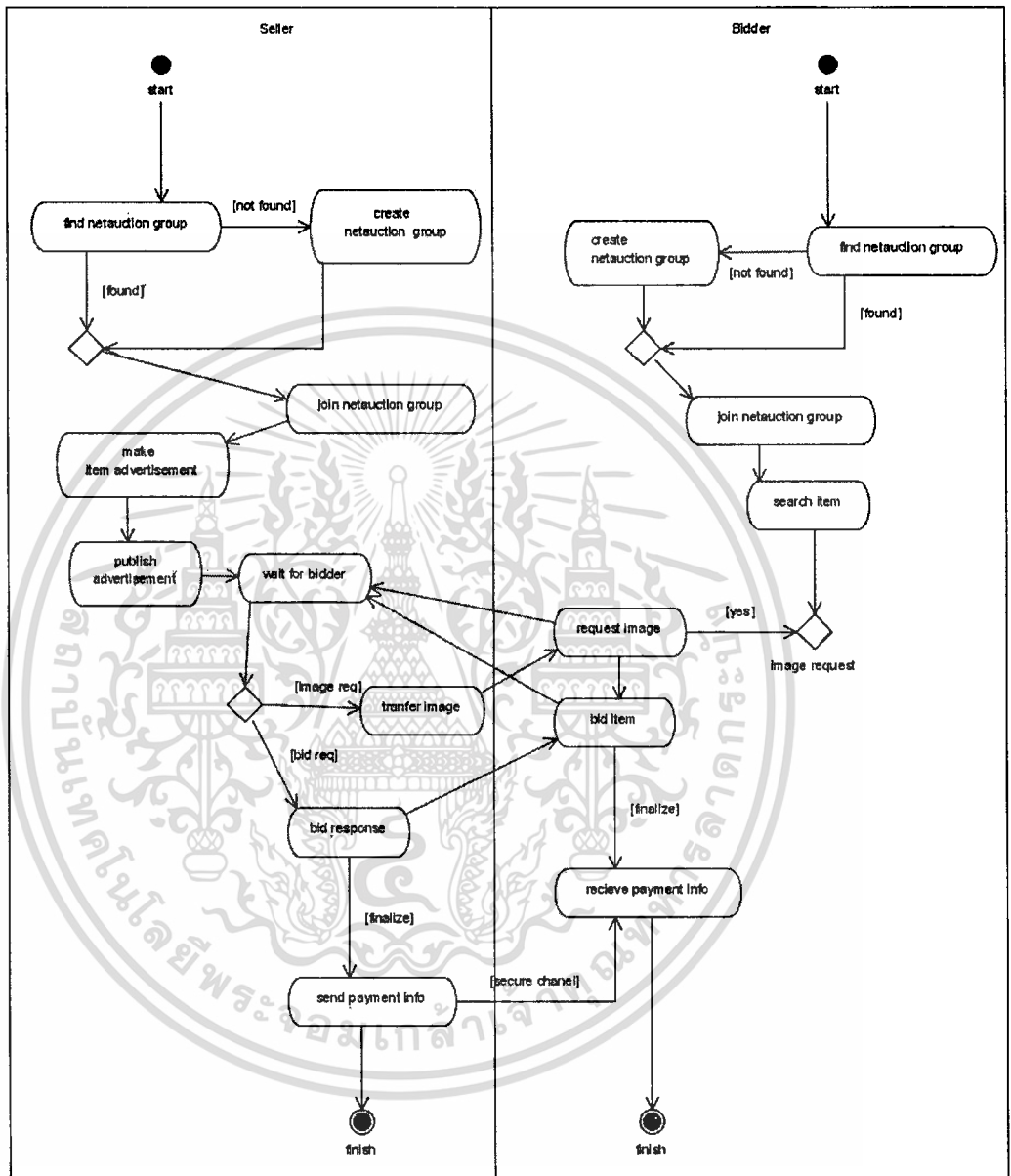
### 5.1.2.10 send payment information

Send payment information เป็น use case ที่จะใช้เมื่อผู้ขาย ได้พอใจ กับราคาที่ได้ใน ปัจจุบันแล้ว ก็จะทำการลบ คำประกาศของสินค้าชนิดนั้นออกไป เพื่อไม่ให้ผู้ซื้อรายอื่นๆสามารถ ค้นคำประกาศของสินค้าชนิดนี้ได้ จากนั้นก็จะต้องมีการส่งรายละเอียดการชำระเงินไปยังผู้ซื้อที่ให้ ราคามากที่สุด และการส่งรายละเอียดอันนี้จะต้องส่งไปยังช่องทางที่ปลอดภัย

### 5.1.2.11 receive payment information

Receive payment information เป็น use case ที่ ผู้ซื้อจะได้รับข้อมูลเกี่ยวกับการชำระเงิน และจะต้องแจ้งกลับไปหาผู้ซื้อว่า ยืนยันการซื้อ เป็นอันเสร็จสมบูรณ์

## 5.2 activity diagram



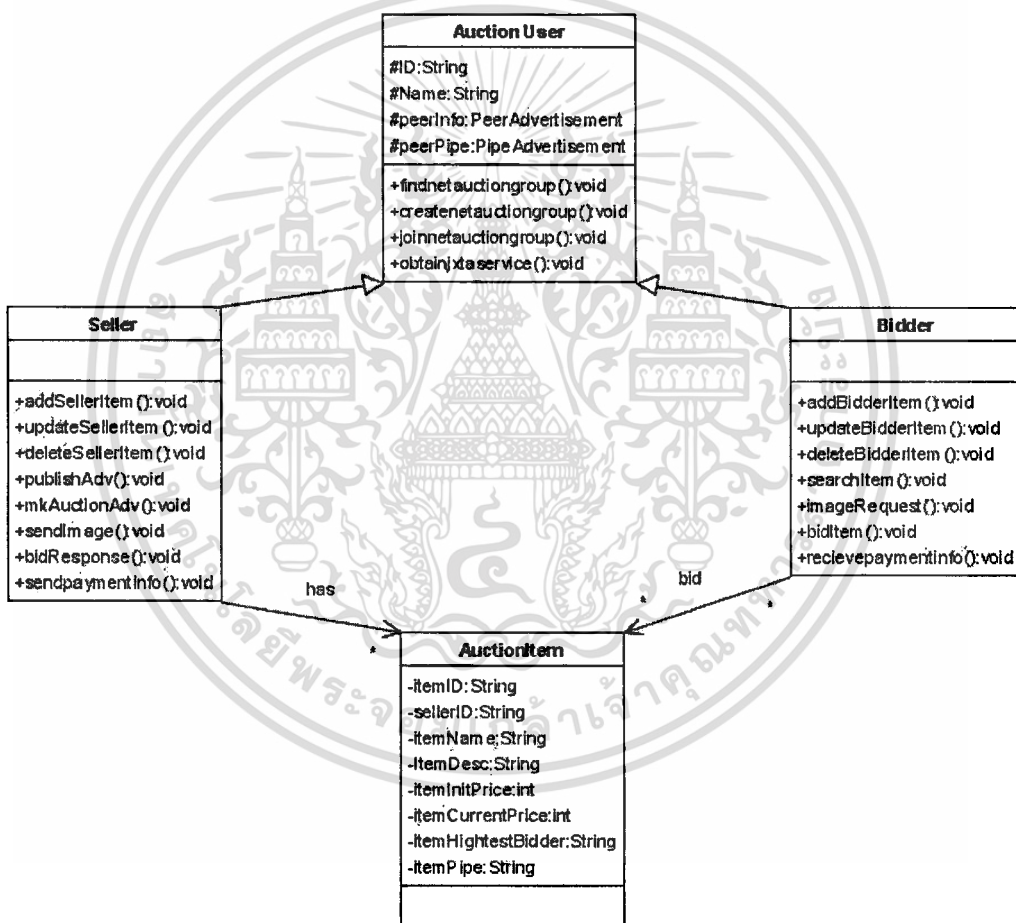
รูปที่ 5.2 Activity Diagram ของ P2P Auction

จากรูปที่ 5.2 จะเห็นได้ว่าการทำงานของโปรแกรม ของผู้ขายและผู้ซื้อจะเหมือนกัน คือ เริ่มต้นจะต้องมีการค้นหากลุ่ม netauction ก่อน หากไม่มีต้องสร้างขึ้นใหม่ และเข้ากลุ่ม ในส่วนของ ผู้ขายก็ สามารถสร้างคำประกาศการขายขึ้นมา จากนั้นก็ต้องส่งคำประกาศออกไป เมื่อส่งคำ

ประกาศออกไปแล้ว ผู้ขายก็อยู่ในสถานะรอ คือ เมื่อมีผู้สนใจสินค้าและส่งคำร้องขอรูปภาพเข้ามา ผู้ซื้อจะต้องส่งรูปภาพกลับไปให้ หรือ หากผู้ซื้อส่งการเสนอราคามา ก็จะต้องทำการตรวจสอบว่าผู้ซื้อเสนอราคามาให้ถูกต้องหรือไม่ แล้วจึงส่งคำตอบกลับกลับไป เมื่อผู้ขายตัดสินใจขายให้กับผู้ซื้อ ก็จะต้องทำการส่งรายละเอียดเกี่ยวกับการจ่ายเงินค่าให้กับผู้ซื้อ

### 5.3 Class Diagram

สำหรับระบบ P2P Auction นี้มีการออกแบบ คลาสไดอะแกรม ดังรูปที่ 4.3



รูปที่ 5.3 คลาสไดอะแกรมของ P2P Auction

#### 5.3.1 คลาส Auction User

คลาสนี้เป็นคลาสที่เก็บข้อมูลเกี่ยวกับโหนดนั้นๆ ไม่ว่าจะเป็นชื่อ รหัสประจำตัว และ ข้อมูลการติดต่อต่างๆ(Peer Advertisement) นอกจากนั้นยังรวมถึง โอเปอเรชันหลักของระบบคือ

การค้นหากลุ่ม การสร้างกลุ่ม การเข้าร่วมกลุ่ม และการขอบริการต่างๆจากกลุ่มที่ JXTA มีให้เช่น บริการการค้นหา (Discovery Service) บริการไปป์ (Pipe Service)

### 5.3.2 คลาส Seller

คลาสนี้เป็นคลาสที่สร้างไว้สำหรับเก็บโอเปอเรชันที่ผู้ขายสามารถทำได้ ไม่ว่าจะเป็นการสร้างคำประกาศของสินค้า การลบคำประกาศ การประกาศคำประกาศ การส่งรูปภาพ การตอบกลับ เมื่อมีการเสนอราคา เป็นต้น

### 5.3.3 คลาส Bidder

คลาสนี้เป็นการสร้างไว้เก็บโอเปอเรชันที่ผู้ซื้อสามารถทำได้ เช่นการค้นหาสินค้า การเสนอราคา และการร้องขอรูปแบบ เป็นต้น

### 5.3.4 คลาส AuctionItem

คลาสนี้เป็นคลาสที่เก็บข้อมูลต่างๆของสินค้าไม่ว่าเป็นชื่อ รหัสสินค้า คำอธิบายสินค้า ราคาเริ่มต้น ราคาปัจจุบัน รูปภาพ ผู้ที่ให้ราคามากที่สุด ณ ขณะนี้ และ ช่องทางการติดต่อของผู้ขาย เป็นต้น



## บทที่ 6

### การพัฒนาระบบ P2P Auction

เมื่อได้มีการออกแบบระบบ เพื่อรวบรวมฟังก์ชันหน้าที่ที่ต้องการแล้ว ขั้นตอนถัดมาก็จะเป็นการพัฒนาระบบจริงๆ ซึ่งจะต้องการศึกษาการใช้ภาษาจาวาและ java swing เพื่อใช้เขียน GUI และการใช้ไลบรารีของ JXTA รวมทั้งเครื่องมือต่างๆที่ต้องใช้ในการพัฒนาโปรแกรม ซึ่งในการพัฒนาระบบงานนี้ จะต้องคำนึงถึง JAVA IDE (Java Integrated Development Environment) เป็นสำคัญ เพราะเป็นหัวใจหลักในการพัฒนาโปรแกรม ผู้เขียนจึงได้พิจารณาอย่างรอบคอบแล้วจึงเลือก Eclipse เป็นเครื่องมือหลักในการพัฒนา เพราะเป็นซอฟต์แวร์ Open Source และมีความสามารถหลายอย่างที่น่าสนใจ

#### 6.1 เครื่องมือที่ใช้ในการพัฒนา

**6.1.2 Eclipse (Java IDE)** สาเหตุที่เลือกใช้โปรแกรมนี้ในการพัฒนาก็เพราะเป็นโปรแกรมที่ไม่ต้องเสียค่าใช้จ่าย และ มีความสามารถมากตัวหนึ่ง แม้จะไม่มี GUI Editor ก็ตาม ผู้เขียนเห็นว่า ใช้โปรแกรมลักษณะนี้จะดีกว่า เพราะสามารถควบคุมและเข้าใจการทำงานของโปรแกรมทั้งหมด เนื่องจากการเขียนด้วยตัวเองและไม่มีการให้โปรแกรมช่วยสร้างโค้ดใดๆเพื่อช่วยการทำงาน อีกทั้งโปรแกรมติดตั้งได้ง่ายและไม่เข้าไปเปลี่ยนแปลงรีจิสตรี ของวินโดวส์อีกด้วย ซอฟต์แวร์ตัวนี้สามารถดาวน์โหลดได้ที่ <http://www.eclipse.org>

**6.1.3 Fat Jar (Eclipse Plug in)** เนื่องจากการพัฒนา JXTA มีการอ้างไลบรารีจำนวนมาก การที่จะนำโปรแกรมไปติดตั้ง (Deployment) จึงเป็นเรื่องลำบากเพราะต้องมีการนำไลบรารีเข้าไปติดตั้ง และเวลารันโปรแกรมทุกครั้งจะต้องกำหนดค่า Class path ให้กับตัวแปลภาษาของจาวาซึ่งจะต้องใช้คำสั่งยาวมากจึงไม่เป็นการสะดวกหากต้องใช้วิธีนี้ ผู้เขียนจึงได้ใช้ Fat Jar ซึ่งเป็น ส่วนเสริมของโปรแกรม Eclipse เพื่อใช้ในการรวบรวมคลาสทั้งหมดที่ต้องใช้ในระบบ ทั้งคลาสของตัวเองโปรแกรมเอง และ คลาสไลบรารีที่เรียกใช้ บีบอัดรวมกันเป็น ไฟล์นามสกุล .jar เพียงไฟล์เดียว สามารถนำไปรันได้เลย โดยใช้คำสั่ง `java -jar p2pauction.jar`

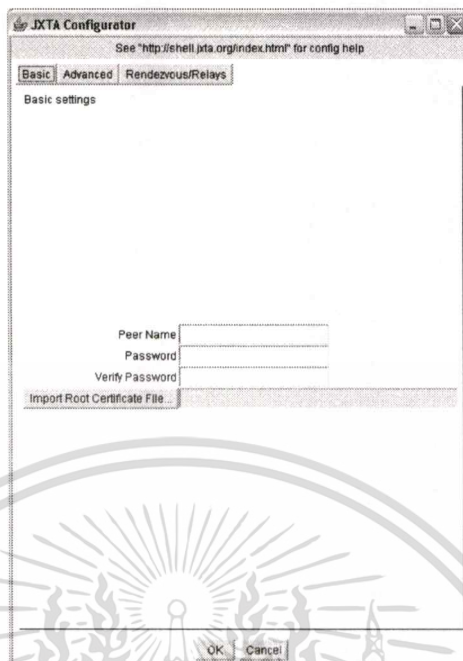
**6.1.4 Java J2SE 1.5** การพัฒนาระบบงานนี้ ได้ใช้ คอมไพเลอร์และตัวแปลภาษา Java J2SE 1.5 ซึ่งเป็นเวอร์ชันล่าสุดในขณะนี้

**6.1.5 Apache Log4j** เนื่องจากผู้เขียนได้ศึกษาการเขียนโปรแกรมของ JXTA มาพอสมควร จึงได้เห็นการใช้ Logging Management System ของ JXTA ที่มีการใช้ Log4j ผู้เขียนจึงได้นำมาประยุกต์ใช้เข้ากับระบบงานนี้ การใช้ Log4j นั้นมีการแบ่งชั้นของการแสดงผลและเก็บ log อย่างมีระเบียบ โดยจะมีระดับของ log คือ ดีบั๊ก(debug) ข่าวสาร(info) แจ้งเตือน(Warning) เกิดข้อผิดพลาด (Error) และ เกิดข้อผิดพลาดร้ายแรง (Fatal) นอกจากนั้นยังมีกลไกในการเลือกที่จะให้ log ไปแสดงผลที่ใด ซึ่งอาจจะเป็น Command Prompt หรือดังเช่นที่ระบบงานนี้ใช้ คือให้ log ไปปรากฏอยู่ที่ Command Prompt และ JTextArea ซึ่งเป็นส่วนที่แสดงสถานะของโปรแกรม การใช้ Log4j ผู้เขียนเห็นว่ามีประสิทธิภาพกว่าการใช้คำสั่งในการพิมพ์ข้อความธรรมดา

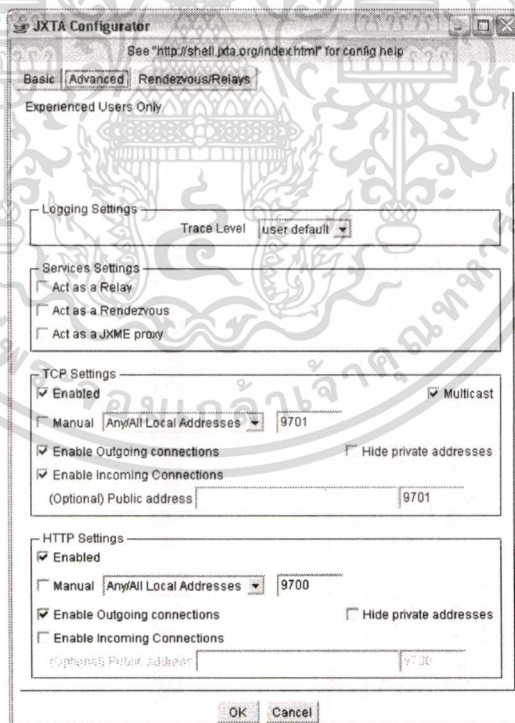
## 6.2 หน้าจอโปรแกรม

### 6.2.1 JXTA Configurator

ในการเริ่มต้นเข้าสู่เน็ตเวิร์คของ JXTA นั้นจะเกิดขึ้นเมื่อมีโค้ด `PeerGroupFactory.newNetPeerGroup()`; อยู่ในโปรแกรม โดยในการกำหนดค่าจะกำหนดค่าในกลุ่ม Basic Setting และ Advance Setting โดยใน Basic Setting (รูปที่ 6.1) นั้นจะต้องใส่ Username และ Password โดย Password ต้องมีความยาวไม่น้อยกว่า 8 ตัวอักษร ส่วนใน Advance Setting (รูปที่ 5.2) จะต้องกำหนด ในส่วนของ TCP Setting โดยเลือกหมายเลขไอพีแอดเดรสของเครื่องที่ต้องการใช้ และ เลือกพอร์ตที่จะให้รองรับการเชื่อมต่อ ส่วนในส่วนของ Http Setting สามารถปิดการใช้งานได้ หากเป็นการทดลองในเครื่องเดียวกัน หรือภายในแลนเดียวกัน การที่จะทดลองโปรแกรมที่เขียนขึ้นโดย JXTA นั้นจะต้องอยู่คนละไครเรทอรีและจะต้องเลือกพอร์ตที่ต่างกันจึงจะใช้งานได้ เมื่อกำหนดค่าเรียบร้อยแล้วจะสังเกตได้ว่า จะมีไครเรทอรีเกิดขึ้น ชื่อ `jxta` ซึ่งภายในจะมีไฟล์ชื่อ `PlatformConfig` ซึ่งเป็น คำประกาศของโหนด (Peer Advertisement) ไฟล์ `jxta.properties` ซึ่งเป็นการกำหนดค่าบางอย่างในระบบ และ ไครเรทอรี `cm` ซึ่งเป็นไครเรทอรีที่เก็บคำประกาศ (Advertisement) ของแต่ละกลุ่มของโหนด(Peergroup)



รูปที่ 6.1 JXTA Configurator



รูปที่ 6.2 JXTA Configurator (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6.2.2 หน้าจอหลักของโปรแกรม

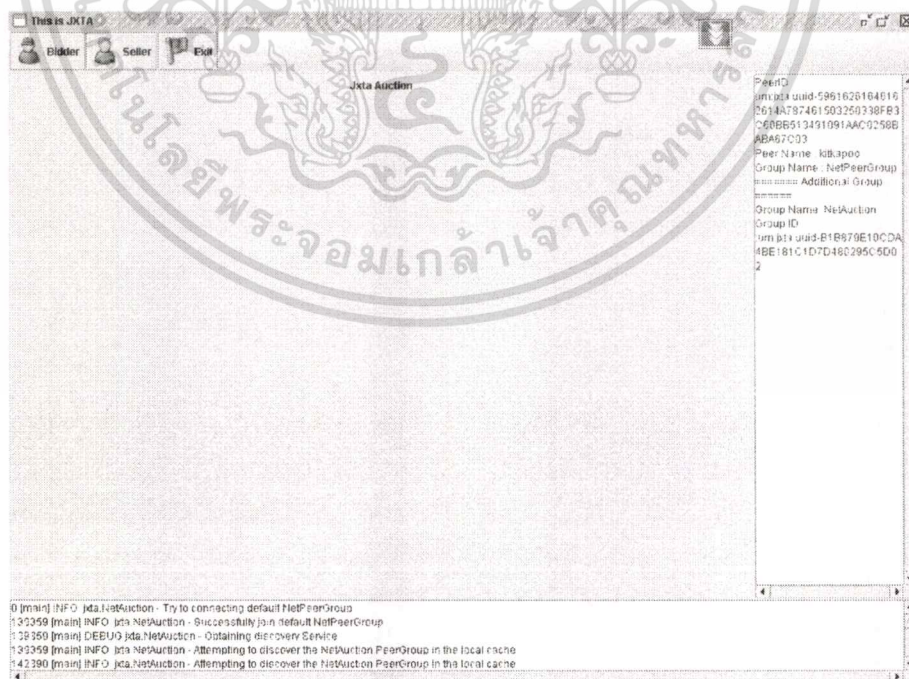
เมื่อผ่านการกำหนดค่าใน JXTA Configurator แล้วหากกำหนดค่าถูกต้อง ระบบก็จะนำมาสู่หน้าจอหลักดังรูปที่ 6.3 โดยจะมีส่วนประกอบหลักสี่ส่วนด้วยกันก็คือ

### 6.2.2.1 ส่วนทูลบาร์ ประกอบไปด้วย ปุ่มสามปุ่ม ก็คือ Seller Bidder และ Exit

เนื่องจากการพัฒนาเป็นการพัฒนาระบบ P2P ดังนั้นโปรแกรมจะต้องสามารถทำหน้าที่เสมือนเป็นได้ทั้งเซิร์ฟเวอร์และไคลเอนท์ในเวลาเดียวกัน ฉะนั้น ปุ่ม Seller ก็จะเป็นปุ่มที่เปลี่ยนหน้าจอแสดงผลหลักให้เป็นการแสดงถึงสินค้าที่กำลังขาย และการเพิ่มหรือลบสินค้าที่กำลังขาย

**6.2.2.2 Info Pane** เป็นส่วนที่แสดงรายละเอียดที่สำคัญเกี่ยวกับ โหนด เช่น รหัสของโหนด (PeerID) ชื่อของกลุ่มของโหนดที่โหนดนี้สังกัด รวมทั้งรหัสของกลุ่มของโหนด (PeerGroup ID)

**6.2.2.3 Status Pane** เนื่องจากการพัฒนาโปรแกรมบนเครือข่าย ผู้เขียนได้ตระหนักถึงความสำคัญในการ Log เนื่องจากช่วยในการดีบั๊กโปรแกรมให้มีประสิทธิภาพมากยิ่งขึ้น ดังนั้นผู้เขียนจึงได้ ใช้ Log4j โดยให้มี log ออกไปทางคอนโซลและ Status โดยรูปแบบของ Log จะเป็นดังนี้ time [thread] level class - message เช่น 148609 [main] DEBUG jxta.NetAuction - Not found Pipe.adv try creating new one จะหมายถึง เวลา 148609 (หน่วยเป็น millisecond) อยู่ใน Main thread คลาส NetAuction ใน แพคเกจ jxta มี log level คือ debug และมีข้อความว่า Not Found Pipe.adv try creating one



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

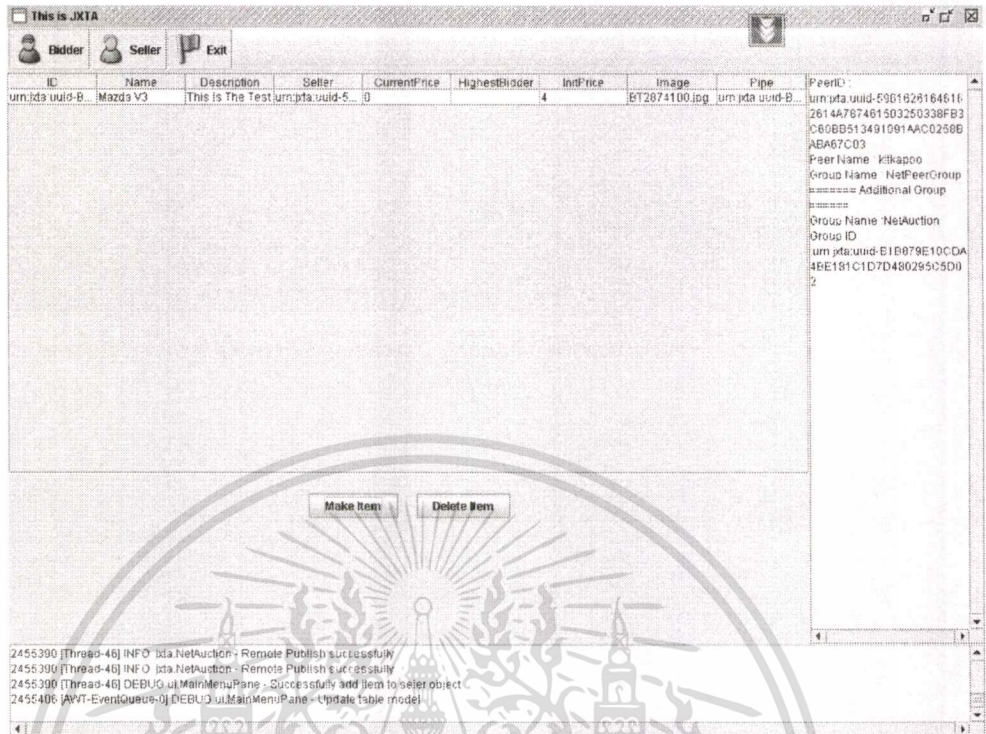
## รูปที่ 6.3 หน้าจอหลัก

### 6.2.3 หน้าจอผู้ขาย

ผู้ขายเมื่อจะเข้าไปขายสินค้า จะต้องกดที่ปุ่ม Seller เพื่อนำมายังหน้าจอผู้ขายดังในรูปที่ 6.4 และจะต้องมีการสร้างการขายขึ้นมาใหม่โดยกดที่ปุ่ม New Item โดยเมื่อกดแล้วจะมี Dialog ให้ใส่รายละเอียดสินค้าที่นำมาทำการประมูล คือชื่อสินค้า รายละเอียดสินค้า ราคาเริ่มต้น และรูปของสินค้า โดยเมื่อใส่ข้อมูลเรียบร้อยแล้ว ข้อมูลจะไปปรากฏบริเวณตาราง ดังรูปที่ 6.5 และ โปรแกรม จะทำการสร้างคำประกาศขาย (Auction Advertisement) ขึ้นมา ดังรูปที่ 6.6



รูปที่ 6.4 หน้าจอผู้ขาย



รูปที่ 6.5 หน้าจอผู้ขาย(ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<?xml version="1.0"?>
<!DOCTYPE jxta:AuctionAdv>
<jxta:AuctionAdv xmlns:jxta="http://jxta.org">
  <Id>
    urn:jxta:uuid-
    59616261646162614E5047205032503326FC566492FB4A08AC05D5CBA1DF
    947301
  </Id>
  <Seller>
    urn:jxta:uuid-
    59616261646162614A787461503250331698475EDC754DE3A19E916B235A
    043403
  </Seller>

  <Pipe>
    urn:jxta:uuid-
    B1B879E10CDA4BE181C1D7D480295C5D162B1AF412554500872A8FF00F7A
    D3E104
  </Pipe>
  <Name>
    Masda
  </Name>
  <Desc>
    Very Very Good Car
  </Desc>
  <InitPrice>
    500
  </InitPrice>
  <Image>
    Mazda.jpg
  </Image>
</jxta:AuctionAdv>

```

รูปที่ 6.6 คำประกาศการขาย (Auction Advertisement)

#### 6.2.4 หน้าจอผู้ประมูลราคา

หน้าจอนี้เป็นการให้ผู้ประมูลราคาทำการค้นหาสินค้าที่สนใจ ดังรูปที่ 6.7 โดยใส่คำค้นหา แล้วกดปุ่ม Search โปรแกรมจะส่งคำถามไปยังทุกโหนดที่อยู่เน็ตเวิร์ค JXTA หากมีผู้ตอบกลับมา โปรแกรมจะทำการแสดงบนตารางซึ่ง ผู้ประมูลราคาสามารถเลือกแล้วกด view เพื่อดูรายละเอียดเกี่ยวกับสินค้าได้ ปุ่ม image request มีไว้สำหรับการขอรูปกับผู้ชาย ปุ่ม bid เป็นปุ่มที่ใช้ในการเสนอราคา โดยใส่ราคาไว้ที่ textbox ด้านหน้า ซึ่งหากราคาที่เสนอไปเป็นราคาที่ไม่ถูกต้องเช่นน้อยกว่าราคาปัจจุบัน หรือ น้อยกว่าราคาเริ่มต้น ก็จะมีข้อความที่ status pane บอกว่าราคาที่เสนอไม่ถูกต้อง

The screenshot shows a NetAuction application window. The main area contains a table with the following data:

ID	Name	Description	Seller	CurrentPrice	HighestBidder	InitPrice	Image	Pipe
um.jkta.uid-B...	Cisco	This is a good	um.jkta.uid-5...	0		40	E12074100.jpg	um.jkta.uid-B...

The sidebar on the right displays peer information:

```
PeerID:
um.jkta.uid-5661620164616
2614A787461503250339BC0
7225840344ABBFD0D777EF
DCA3BC03
Peer Name : Buyer1
Group Name : NetPeerGroup
===== Additional Group
=====
Group Name : NetAuction
Group ID
um.jkta.uid-618379E10CDA
4BE181CID7D480295C5D0
2
```

At the bottom, there is a console window with the following log entries:

```
335563 [Thread-18] DEBUG jkta.NetAuction - Price is valid...OK
335563 [Thread-18] DEBUG pta.NetAuction - Price is valid...OK
335563 [Thread-18] DEBUG jkta.NetAuction - Price is valid...OK
335563 [Thread-18] DEBUG pta.NetAuction - Price is valid...OK
```

Buttons for 'Search', 'Image Request', 'View', and 'Bid' are visible at the bottom of the main window.

รูปที่ 6.7 หน้าจอหลักของผู้ประมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 7

### บทสรุป

ในโครงการพัฒนาระบบการประมวลแบบ P2P โดยใช้สถาปัตยกรรม JXTA ได้อธิบายถึงทฤษฎีหลักที่เกี่ยวข้องกับ JXTA ซึ่งเป็นส่วนสำคัญในการที่จะทำให้เข้าใจถึงกลไกการทำงานของ JXTA ซึ่งผู้เขียนได้ทดลองทำระบบการประมวลขึ้นมาเพื่อประกอบการศึกษา โดยการทำระบบประมวลครั้งนี้ไม่ได้ใช้โมเดลแบบไคลเอนท์/เซิร์ฟเวอร์เหมือนกับที่มีในปัจจุบัน

#### 7.1 ผลการพัฒนาระบบ

การพัฒนาระบบ ผู้เขียนได้ศึกษาทฤษฎีเกี่ยวกับการเขียนภาษาจาวา และ การใช้ JXTA ตั้งแต่หลังวิชาสัมมนาสอง และทดลองเขียนโปรแกรมสั้น เพื่อลองใช้งานดูมาโดยตลอด

เนื่องจาก JXTA เป็น รูปแบบ ซึ่งไม่ได้ยึดติดกับภาษาใดภาษาหนึ่ง แต่ในปัจจุบัน ภาษาที่นิยมใช้ในการเขียน JXTA มากที่สุดก็เป็นภาษาจาวา ผู้เขียนจึงได้ใช้ภาษาจาวาในการทำโครงการครั้งนี้ทั้งระบบ

จากการพัฒนาระบบโดยใช้ Java Swing ในการเขียนส่วนที่ติดต่อกับผู้ใช้ ก็พบว่าในจาวาเวอร์ชันปัจจุบัน (J2SE 1.50) ได้มีการพัฒนารูปแบบให้มีความสวยงามมากยิ่งขึ้น และไม่จำเป็นต้องติดตั้งทดลองใช้ในเครื่องคอมพิวเตอร์ในปัจจุบัน

การทดลองโปรแกรมที่พัฒนาขึ้น ผู้เขียนได้เลือกที่จะทดลองภายในเครือข่ายแลน เนื่องจากง่ายในการหาข้อผิดพลาดของโปรแกรม หากโปรแกรมนี้อาจใช้ในเครือข่ายอินเทอร์เน็ต ต้องมีการพัฒนาเพิ่มเติมปรับเปลี่ยนการติดต่อเพียงเล็กน้อย ซึ่งผลการทดลองในเครือข่ายแลนก็ได้ผลเป็นที่น่าพอใจ

#### 7.2 ประโยชน์ที่ได้รับ

ประโยชน์ที่ได้รับจากการพัฒนาระบบงานครั้งนี้ คือ ผู้เขียนได้เรียนรู้การใช้ภาษาจาวา ซึ่งต้องยอมรับว่าเป็นภาษาต้องอาศัยการเรียนรู้มาก หากเทียบกับภาษาเช่นวิซวลเบสิก ยิ่งการเขียน Java Swing ด้วยแล้ว การวางเลเอาท์ต่างๆ ไม่สามารถทำได้อย่างง่าย ผู้เขียนเลือกที่จะเขียนโค้ดเองทั้งหมด โดยไม่ใช่เครื่องมือใดๆ ช่วยเลย ทำให้ได้เข้าใจถึงตัวโค้ดโปรแกรมทั้งหมด

นอกจากนี้จากการศึกษา JXTA พบว่ามีความน่าสนใจมาก มีมหาวิทยาลัยในต่างประเทศจำนวนมากที่เข้าร่วมกับทางเว็บ JXTA แต่จากการค้นหาในประเทศไทยพบว่ามีผู้ที่สนใจเรื่องนี้ อยู่จำนวนไม่มาก และ JXTA ก็ยังไม่ได้รับการพัฒนาถึงจุดสูงสุดของมัน แต่การพัฒนาก็ยังมีไปอย่างต่อเนื่อง

### 7.3 ปัญหาและอุปสรรค

ปัญหาและอุปสรรคที่เกิดขึ้นจากการพัฒนาระบบประมวลแบบ P2P โดยใช้ JXTA มีดังนี้

- ผู้เขียนต้องใช้เวลาศึกษาการเขียนโปรแกรมในแบบที่เป็น Object Oriented โดยใช้ภาษาจาวา อยู่นานพอสมควร
- การศึกษาการเขียนโปรแกรมเพื่อติดต่อกับผู้ใช้โดยใช้ Java Swing มีรายละเอียดมาก จึงทำให้การพัฒนาระบบล่าช้า
- เนื่องจากหนังสือที่เขียนเกี่ยวกับ JXTA ก่อนข้างล้ำสมัย ประกอบกับ JXTA มีการพัฒนาอย่างต่อเนื่องดังนั้น ความรู้ที่หาได้เกี่ยวกับ JXTA ส่วนใหญ่จะอยู่ที่เว็บไซต์ [www.jxta.org](http://www.jxta.org) ซึ่งเป็นการเขียนโดยโปรแกรมเมอร์ทั่วโลกที่รวมตัวกัน และ ใน mailing list ของเว็บ ทำให้ยากต่อการศึกษา
- ในระบบนี้มีการใช้คำประกาศการขาย (Auction Advertisement) ซึ่งเป็นการกำหนดคำประกาศขึ้นเอง ในการกำหนดคำประกาศขึ้นเอง มีขั้นตอนการทำยากกว่าคำประกาศที่มีอยู่แล้วใน JXTA มาก ผู้เขียนต้องค้นหาใน CVS Tree ของในเว็บ เพื่อดูรูปแบบของโค้ดที่ใช้เขียนคำประกาศเพื่อนำมาประยุกต์ใช้กับระบบนี้ ซึ่งเป็นขั้นตอนที่ใช้เวลามาก เนื่องจากไม่มีเอกสารในเว็บที่จะแสดงให้เห็นถึงการเขียนโปรแกรมในลักษณะนี้
- การพัฒนาระบบให้มีลักษณะแบบ P2P ผู้เขียนพบว่าเป็นการยากพอสมควร เนื่องจากต้องเขียนให้โปรแกรมเป็นได้ทั้งไคลเอนท์ และ เซอร์ฟเวอร์ การหาข้อผิดพลาดของโปรแกรม ก็ทำได้ยากมากเช่นกัน บางครั้งจะต้องใช้โปรแกรมจำพวกวิเคราะห์โปรโตคอล (Protocol Analyzer) เพื่อหาข้อผิดพลาด

### 7.4 ข้อเสนอแนะ

ในปัจจุบันจะเห็นได้ว่า โปรแกรมที่มีลักษณะแบบ P2P ได้เกิดขึ้นจำนวนมาก และเป็นที่ยอมรับสำหรับผู้ใช้อินเตอร์เน็ต แต่การศึกษาการทำงานของ P2P อย่างจริงจัง ยังมีน้อยมาก หากไม่ได้ศึกษาในระดับของการค้นคว้า (Research) ก็ยากที่จะเข้าใจ การทำงานที่แท้จริงของมัน

นอกจากนี้การเรียนการสอนเกี่ยวกับ การเขียน โปรแกรมเน็ตเวิร์ค ส่วนใหญ่จะเน้นอยู่กับ การเขียนโปรแกรมแบบ โมเดลไคลเอนท์/เซิร์ฟเวอร์ ทำให้ไม่สามารถประยุกต์เข้ากับการเขียน โปรแกรมในโมเดล P2P ได้เพราะแตกต่างกันในหลายองค์ประกอบ

โปรแกรมที่ผู้เขียนได้พัฒนาขึ้น สามารถนำไปพัฒนาต่อให้มีความสามารถมากขึ้นได้ เช่น พัฒนาในส่วนของการความปลอดภัย และการเก็บข้อมูล ซึ่งต้องอาศัยความชัดเจนของ JXTA ในส่วนนี้ มากขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

Joseph, D. Gradecki and Joe, Gradecki 2002. **Mastering JXTA**. USA: Wiley.

Sun Microsystems. **JXTA v2.0 Protocol Specification** [online] Available:

<http://spec.jxta.org/nonav/v1.0/docbook/JXTAProtocols.html>. 2004.

Sun Microsystems. **JXTA v2.3.X: Java Programmer's Guide** [Online] Available:

[http://www.jxta.org/docs/JxtaProgGuide\\_v2.3.pdf](http://www.jxta.org/docs/JxtaProgGuide_v2.3.pdf). 2003.

Sun Microsystems. **JXTA Technology Creating Connected Communities** [online] Available:

<http://www.jxta.org/docs/JXTA-Exec-Brief.pdf>. 2005.



## ภาคผนวก

### กฉฉ AuctionAdvertisement

```

package jxta;

import net.jxta.document.ExtendableAdvertisement;
import net.jxta.id.ID;
import net.jxta.peer.PeerID;
import net.jxta.pipe.PipeID;

/*
 * Created on 3 1/4. 2548
 *
 * TODO To change the template for this generated file go to
 * Window - Preferences - Java - Code Style - Code Templates
 */

/**
 * @author kitkapoo
 *
 * TODO To change the template for this generated type comment go to
 * Window - Preferences - Java - Code Style - Code Templates
 */

public abstract class AuctionAdvertisement extends ExtendableAdvertisement {

    private String itemDescription = null;
    private ID itemID = null;
    private PeerID seller = null;
    private String itemName = null;

```

```

private int itemInitPrice = 0;
private int itemCurrentPrice = 0;
private PeerID itemHighestBidder = null;
private String itemImage = null;
private PipeID pipeID = null;

// Define TAG to be used

//private static final String advertisementType = "AuctionAdv";

protected final static String descTag = "Desc";
protected final static String idTag = "Id";
protected final static String sellerTag = "Seller";
protected final static String nameTag = "Name";
protected final static String initPriceTag = "InitPrice";
protected final static String currentPriceTag = "CurrentPrice";
protected final static String highestBidderTag = "HighestBidder";
protected final static String imageTag="Image";
protected final static String pipeTag="Pipe";

public PipeID getPipeID() {
    return pipeID;
}

public void setPipeID(PipeID pipeID) {
    this.pipeID = pipeID;
}

public ID getID() {
    // TODO Auto-generated method stub
    return (ID) itemID;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

/**
 * @return Returns the itemCurrentPrice.
 */
public int getItemCurrentPrice() {
    return itemCurrentPrice;
}

/**
 * @param itemCurrentPrice The itemCurrentPrice to set.
 */
public void setItemCurrentPrice(int itemCurrentPrice) {
    this.itemCurrentPrice = itemCurrentPrice;
}

/**
 * @return Returns the itemDescription.
 */
public String getItemDescription() {
    return itemDescription;
}

/**
 * @param itemDescription The itemDescription to set.
 */
public void setItemDescription(String itemDescription) {
    this.itemDescription = itemDescription;
}

/**
 * @return Returns the itemHighestBidder.
 */
public PeerID getItemHighestBidder() {

```

```

        return itemHighestBidder;
    }
    /**
     * @param itemHighestBidder The itemHighestBidder to set.
     */
    public void setItemHighestBidder(PeerID itemHighestBidder) {
        this.itemHighestBidder = itemHighestBidder;
    }
    /**
     * @return Returns the itemID.
     */
    public ID getItemID() {
        return itemID;
    }
    /**
     * @param itemID The itemID to set.
     */
    public void setItemID(ID itemID) {
        this.itemID = itemID;
    }
    /**
     * @return Returns the itemImage.
     */
    public String getItemImage() {
        return itemImage;
    }
    /**
     * @param itemImage The itemImage to set.
     */
    public void setItemImage(String itemImage) {

```

```

        this.itemImage = itemImage;
    }
    /**
     * @return Returns the itemInitPrice.
     */
    public int getItemInitPrice() {
        return itemInitPrice;
    }
    /**
     * @param itemInitPrice The itemInitPrice to set.
     */
    public void setItemInitPrice(int itemInitPrice) {
        this.itemInitPrice = itemInitPrice;
    }
    /**
     * @return Returns the itemName.
     */
    public String getItemName() {
        return itemName;
    }
    /**
     * @param itemName The itemName to set.
     */
    public void setItemName(String itemName) {
        this.itemName = itemName;
    }
    /**
     * @return Returns the itemOwnerID.
     */
    public PeerID getSeller() {

```

```

        return seller;
    }

    /**
     * @param itemOwnerID The itemOwnerID to set.
     */
    public void setSeller(PeerID itemOwnerID) {
        this.seller = itemOwnerID;
    }

    public static String getAdvertisementType() {
        return "jxta:AuctionAdv";
    }
    public final String getBaseAdvType() {
        return getAdvertisementType();
    }
}

```

#### คลาส AuctionAdvImpl

```

package jxta;

import java.net.URI;
import java.net.URISyntaxException;
import java.util.Enumeration;

import net.jxta.document.Advertisement;
import net.jxta.document.AdvertisementFactory;
import net.jxta.document.Attributable;
import net.jxta.document.Document;

```

```

import net.jxta.document.Element;

import net.jxta.document.MimeMediaType;

import net.jxta.document.StructuredDocumentFactory;

import net.jxta.document.StructuredDocument;

import net.jxta.document.StructuredTextDocument;

import net.jxta.document.TextElement;

import net.jxta.id.ID;

import net.jxta.id.IDFactory;

import net.jxta.peer.PeerID;

import net.jxta.pipe.PipeID;

/*
 * Created on 3/4/2548
 *
 * TODO To change the template for this generated file go to
 * Window - Preferences - Java - Code Style - Code Templates
 */

/**
 * @author kitkapoo
 *
 * TODO To change the template for this generated type comment go to
 * Window - Preferences - Java - Code Style - Code Templates
 */

public class AuctionAdvImpl extends AuctionAdvertisement {

    private static final String [] fields = {nameTag, idTag};

    public AuctionAdvImpl()
    {

```

```

}

public AuctionAdvImpl(Element root)
{
    initialize(root);
}

void initialize(Element root)
{
    if(!TextElement.class.isInstance(root))
    {
        throw new IllegalArgumentException(getClass().getName() +
"only support" +
        " text element");
    }

    TextElement doc = (TextElement)root;
    if(!doc.getName().equals(getAdvertisementType()))
    {
        throw new IllegalArgumentException("Could" +
" not construct :" + getClass().getName() +
"from doc containing a" + doc.getName() );
    }

    Enumeration elements = doc.getChildren() ;
    while (elements.hasMoreElements()) {
        TextElement elem = (TextElement) elements.nextElement();
        String nm = elem.getName() ;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(nm.equals(idTag))
        {
            try {
                URI pid = new URI(elem.getTextValue() );
                setItemID((ID)IDFactory.fromURI(pid));
            } catch (URISyntaxException e) {
                // TODO: handle exception
                throw new IllegalArgumentException("Bad ID
in advertisement");
            }
        }
        continue;
    }
    if (nm.equals( nameTag))
    {
        setItemName(elem.getTextValue());
        continue;
    }
    if(nm.equals(descTag))
    {
        setItemDescription(elem.getTextValue());
        continue;
    }
    if(nm.equals(initPriceTag))
    {
        setItemInitPrice(Integer.parseInt(elem.getTextValue()));
        continue;
    }
    if(nm.equals(currentPriceTag))
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setItemCurrentPrice(Integer.parseInt(elem.getTextValue()));
        continue;
    }
    if(nm.equals(imageTag))
    {
        setItemImage(elem.getTextValue());
        continue;
    }
    if(nm.equals(sellerTag))
    {
        try {
            URI Pid = new URI(elem.getTextValue());
            setSeller((PeerID)IDFactory.fromURI(Pid));
        } catch (URISyntaxException e) {
            throw new IllegalArgumentException("Bad
Peer ID");
        }
        continue;
    }
    if(nm.equals(highestBidderTag))
    {
        try {
            URI Pid = new URI(elem.getTextValue());

setItemHighestBidder((PeerID)IDFactory.fromURI(Pid));
        } catch (URISyntaxException e) {
            throw new IllegalArgumentException("Bad
Peer ID");

```



```

    /* (non-Javadoc)
    *
    net.jxta.document.Advertisement#getDocument(net.jxta.document.MimeMediaType)
    */
    public Document getDocument(MimeMediaType encodeas) {
        // TODO Auto-generated method stub
        //StructuredDocument adv = null;

        //adv =
        (StructuredTextDocument)StructuredDocumentFactory.newStructuredDocument(encodeas,getAd
        vertisementType());
        StructuredDocument adv = (StructuredDocument) super.getDocument(
        encodeas );
        if(adv instanceof Attributable)
        {
            ((Attributable)adv).addAttribute("xmlns:jxta","http://jxta.org");
        }

        Element e;
        e = adv.createElement(idTag,getItemID().toString());
        adv.appendChild(e);

        e= adv.createElement(sellerTag,getSeller().toString() );
        adv.appendChild(e);

        if(getItemHighestBidder() != null)
        {
            e=
            adv.createElement(highestBidderTag,getItemHighestBidder().toString() );

```

```

        adv.appendChild(e);
    }

    if(getPipeID() != null)
    {
        e = adv.createElement(pipeTag,getPipeID().toString());
        adv.appendChild(e);
    }

    if(getItemName()!=null)
    {
        e = adv.createElement(nameTag,getItemName());
        adv.appendChild(e);
    }
    if(getItemDescription()!=null)
    {
        e = adv.createElement(descTag,getItemDescription());
        adv.appendChild(e);
    }
    if(getItemInitPrice()!=0)
    {
        e =
        adv.createElement(initPriceTag,Integer.toString(getItemInitPrice()));
        adv.appendChild(e);
    }
    if(getItemCurrentPrice()!=0)
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

e
=
adv.createElement(currentPriceTag,Integer.toString(getItemCurrentPrice()));
    adv.appendChild(e);
}
if(getItemImage()!=null)
{
    e = adv.createElement(imageTag,getItemImage());
    adv.appendChild(e);
}
return adv;
}
public static class Instantiator implements AdvertisementFactory.Instantiator
{
    /* (non-Javadoc)
    *
    * @see
    net.jxta.document.AdvertisementFactory.Instantiator#getAdvertisementType()
    */
    public String getAdvertisementType() {
        // TODO Auto-generated method stub

        return AuctionAdvImpl.getAdvertisementType();
    }

    /* (non-Javadoc)

```

```

*
net.jxta.document.AdvertisementFactory.Instantiator#newInstance()
*/
public Advertisement newInstance() {
    // TODO Auto-generated method stub
    return new AuctionAdvImpl();
}

```

```

/* (non-Javadoc)
*
net.jxta.document.AdvertisementFactory.Instantiator#newInstance(net.jxta.document.Element)
*/
public Advertisement newInstance(Element root) {
    return new AuctionAdvImpl(root);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้เขียน

ชื่อผู้เขียน	นายธนากร แสงสุข
วันเกิด	17 ธันวาคม 2521
สถานที่เกิด	ระยอง
วุฒิการศึกษา	วารสารศาสตร์บัณฑิต มหาวิทยาลัยธรรมศาสตร์
ตำแหน่งหน้าที่	ผู้ดูแลระบบคอมพิวเตอร์
สถานที่ทำงาน	The American School of Bangkok



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้