รายงานโครงการวิจัยประจำปีงบประมาณ 2545

เรื่อง

สถาปัตยกรรมแมทช์เมคกิ้งเอเจนท์แบบกระจาย

Distributed Matchmaking Agent Architecture

โดย

ภัทรชัย ลลิตโรจน์วงศ์

คณะเทคโนโลยีสารสนเทศ

ทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ 10520

พ.ศ. 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง
ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

รายงานโครงการวิจัยประจำปีงบประมาณ 2545

เรื่อง

สถาปัตยกรรมแมทช์เมคกิ้งเอเจนท์แบบกระจาย
Distributed Matchmaking Agent Architecture

00956
Lทส17445
1 2721815

โดย

ภัทรชัย ลลิตโรจน์วงศ์

วัน เดือน ปี 10 ต.ค. 2549
เลขทะเบียน H0006916
เลขเรียกหนังสือ R QA 76.76 .I58 ภ 3655 ฉ.2
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."

คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ 10520
พ.ศ. 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาปัตยกรรมแมทช์เมคกิ้งเอเจนท์แบบกระจาย

Distributed Matchmaking Agent Architecture

ภัทรชัย ลลิตโรจน์วงศ์

บทคัดย่อ

งานวิจัยนี้นำเสนอการปรับปรุงเทคโนโลยีแมทช์เมคกิ้งเอเจนท์ ซึ่งเป็นเอเจนท์ที่ทำหน้าที่ค้นหาเอเจนท์ที่สามารถให้บริการที่เหมาะสมกับเอเจนท์หนึ่งซึ่งกำลังมองหาบริการที่เหมาะสมจากเอเจนท์อื่นได้ โดยเสนอสถาปัตยกรรมใหม่ที่ช่วยบรรเทาปัญหาการขาดความสามารถในการปรับขยายขนาด (scalability) ซึ่งทำให้เอเจนท์ประเภทนี้ยังไม่สามารถนำไปใช้งานในระบบเอเจนท์ใดๆ ได้อย่างมีประสิทธิภาพ หากแต่สามารถนำไปใช้ได้เฉพาะกับระบบเอเจนท์ขนาดย่อม แนวทางแก้ไขที่ได้นำเสนอได้แก่การพัฒนาแมทช์เมคกิ้งเอเจนท์เป็นระบบกระจาย โดยแบ่งงานตามโดเมนของบริการที่ร้องขอเข้ามาแทน เพื่อให้ได้รับประโยชน์จากโครงสร้างการทำงานแบบลำดับชั้น สถาปัตยกรรมที่พัฒนาขึ้นนี้จะถูกนำไปทดสอบด้วยการพัฒนาเอเจนท์ขึ้นมาทดสอบจริง เพื่อแสดงให้เห็นว่าสถาปัตยกรรมนี้จะช่วยทำให้สามารถนำเทคโนโลยีแมทช์เมคกิ้งเอเจนท์ไปใช้กับระบบที่ใหญ่ขึ้นได้

Abstract

This research work proposes an improvement of matchmaking agent technology. A matchmaking agent is an agent that finds provider agents who can give a requester agent looking for some services appropriately. A new architecture is introduced in order to relieve the lacking in scalability of the existing approaches. Currently, the scalability problem makes this type of agent unable to be efficiently implemented in any agent systems; only a small group of agents can employ it. The proposed solution is to develop a distributed matchmaking system by partitioning work based on request and service domains. This architecture will be validated by the implementations of agents in simulated environment. The experiment results demonstrate that we are able to apply the matchmaking agent technology in large systems with this architecture.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทที่ 1 บทนำ 1

1.1 ปัญหาและความเป็นมา 1

1.2 วัตถุประสงค์ในการศึกษา 2

1.3 ขอบเขตการศึกษา 2

1.4 ประโยชน์ที่คาดว่าจะได้รับ 2

บทที่ 2 แนวคิดและทฤษฎีที่เกี่ยวข้อง 3

2.1 เทคโนโลยีเอเจนท์ 3

2.2 โพรโทคอลแลกความรู้ 3

2.3 ระบบหลายเอเจนท์ 5

2.4 เทคโนโลยีแมทช์เมคกิ้งเอเจนท์ 7

2.5 เทคโนโลยีที่ใช้ในการปรับปรุงแมทช์เมคกิ้งเอเจนท์ 17

2.6 สรุป 22

บทที่ 3 ขั้นตอนการศึกษา 23

3.1 การกำหนดปัญหาและแนวทางการแก้ไข 23

3.2 การออกแบบระบบ 23

3.3 การสร้างระบบ 23

3.4 การทดสอบระบบ 24

บทที่ 4 ปัญหาและแนวทางแก้ไข 25

4.1 การกำหนดปัญหา 25

4.2 แนวทางแก้ไข 30

บทที่ 5 การออกแบบและสร้างระบบ 35

5.1 การเลือกโครงสร้างแมทช์เมคกิ้งเอเจนท์ให้กับระบบ DMAA 35

5.2 การทำงานร่วมกันของแมทช์เมคกิ้งเอเจนท์ 38

5.3 การแสดงความต้องการและความสามารถของเอเจนท์ 41

5.4 กลไกการจับคู่ 43

5.5 การสร้างระบบ DMAA 45

5.6 สรุป 50

บทที่ 6 การทดสอบระบบ 51

6.1 การออกแบบการทดสอบ 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

6.2 ขั้นตอนการทดสอบ 58
6.3 ผลการทดสอบ 60
6.4 สรุปการทดสอบ 64
บทที่ 7 สรุปผลการวิจัยและข้อเสนอแนะ 66
บรรณานุกรม 71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ปัญหาและความเป็นมา

ในปัจจุบัน คอมพิวเตอร์เป็นเครื่องมือที่จำเป็นอย่างยิ่งสำหรับการทำงาน โดยมีการนำคอมพิวเตอร์เข้ามาช่วยงานในหลายด้าน โดยเฉพาะงานประเภทงานประจำ ซึ่งเป็นงานที่ทำให้ผู้ใช้เสียเวลามากในขณะที่ใช้ความรู้เพียงเล็กน้อยก็สามารถปฏิบัติงานดังกล่าวได้ ดังนั้น จึงได้เกิดแนวคิดในการใช้ตัวแทนเพื่อปฏิบัติหน้าที่แทนผู้ใช้ ตัวแทนดังกล่าวเรียกว่า เอเจนท์ (Agent) ซึ่งแบ่งเป็น 2 ประเภทหลัก ได้แก่ เอเจนท์ที่เป็นซอฟท์แวร์ (Software Agent) และ เอเจนท์ที่เป็นฮาร์ดแวร์ (Robotic Agent) อย่างไรก็ตาม ในงานวิจัยนี้ จะกล่าวถึงเฉพาะเอเจนท์ที่เป็นซอฟท์แวร์เท่านั้น ดังนั้นคำว่าเอเจนท์ที่จะกล่าวต่อไปจึงหมายถึงซอฟท์แวร์เอเจนท์นั่นเอง

การที่เอเจนท์ใดๆ จะทำงานได้สำเร็จโดยที่รบกวนผู้ใช้น้อยที่สุดนั้น จำเป็นอย่างยิ่งที่จะต้องมีข้อมูล ความรู้ และทรัพยากรที่เพียงพอ อย่างไรก็ตาม เหตุการณ์ดังกล่าวเป็นไปได้ยากหากเอเจนท์ทำงานเพียงลำพัง ดังนั้นเอเจนท์จึงจำเป็นต้องมีความสามารถในการติดต่อสื่อสารกับผู้อื่นด้วย ตัวอย่างเช่น การติดต่อกับเอเจนท์อื่นๆ และการติดต่อกับเว็บเพจ เป็นต้น การที่เอเจนท์หนึ่งๆ ต้องการติดต่อกับเอเจนท์อื่นนั้น มีสาเหตุมาจากการที่ตนเองไม่สามารถทำงานบางประการได้ เนื่องจากขาดข้อมูลหรือความสามารถบางอย่าง จึงร้องขอบริการหรือข้อมูลจากเอเจนท์อื่นนั่นเอง

อย่างไรก็ตาม ก่อนที่เอเจนท์จะสามารถร้องขอบริการจากเอเจนท์อื่นได้นั้น จำเป็นอย่างยิ่งที่เอเจนท์นั้นจะต้องทราบก่อนว่ามีเอเจนท์ที่สามารถให้บริการที่ต้องการอยู่แห่งใดบ้าง ดังนั้น จึงได้มีงานวิจัยพัฒนาเอเจนท์ชนิดหนึ่งขึ้นเพื่อทำหน้าที่ค้นหาเอเจนท์ผู้ให้บริการที่สามารถทำงานที่เอเจนท์อื่นร้องขอมาได้ เรียกว่าแมทช์เมคกิ้งเอเจนท์ (Matchmaking Agent - MMA) หรือ Matchmaker (Kuokka and Harada. 1998; Decker *et al.* 1996) เอเจนท์ประเภทนี้จะเก็บโฆษณาความสามารถของเอเจนท์ต่างๆ ภายในกลุ่ม/ระบบเอเจนท์ (Multi-agent System - MAS) ไว้จำนวนหนึ่ง เพื่อเป็นฐานข้อมูลในการเลือกเอเจนท์ที่เหมาะสมกับแต่ละคำร้องต่อไป

เทคโนโลยีแมทช์เมคกิ้งเอเจนท์ที่มีอยู่ในปัจจุบันมีลักษณะโดยทั่วไป ได้แก่

- เป็นระบบแบบรวมศูนย์ ซึ่งก่อให้เกิดข้อด้อยต่างๆ โดยเฉพาะอย่างยิ่งเมื่อนำไปใช้ในอินเทอร์เน็ต (Kurose and Ross. 2000) ได้แก่ มีปริมาณการจราจรของข้อมูลที่คับคั่ง การดูแลรักษาเป็นไปได้ยากเนื่องจากต้องจัดการกับข้อมูลเป็นจำนวนมาก และอยู่ห่างไกลจากผู้ใช้ซึ่งก็คือเอเจนท์ที่มาขอใช้บริการ
- มีความสามารถในการค้นหาที่เหนือกว่าสารบบโดยทั่วไป ไม่ใช่แค่ค้นหาจากคำทีละคำ หากแต่จะมีความสามารถที่จะค้นหาจากประโยคได้ด้วย (Sycara *et al.* 1999)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น แนวทางการนำแมทช์เมคกิ้งเอเจนท์มาใช้ โดยเฉพาะอย่างยิ่งบนระบบอินเทอร์เน็ต จึงควรประยุกต์ใช้งานแมทช์เมคกิ้งเอเจนท์หลายตัวพร้อมกัน โดยจัดให้มีอัลกอริทึมที่ใช้ในการควบคุมการให้บริการของเอเจนท์ดังกล่าวร่วมกันอยู่ ได้แก่ การแบ่งหน้าที่รับผิดชอบ และการค้นหาการโฆษณา โดยจะต้องเป็นกระบวนการที่ไม่ซับซ้อนมากและใช้แบนด์วิดท์น้อย ในงานวิจัยฉบับนี้จะเรียกแนวทางนี้ว่า สถาปัตยกรรมแมทช์เมคกิ้งเอเจนท์แบบกระจาย (Distributed Matchmaking Agent Architecture - DMAA)

อย่างไรก็ตาม ระบบ DMAA นั้นมีลักษณะคล้ายคลึงกับระบบค้นหาหมายเลขไอพีของเว็บไซต์ในเครือข่ายอินเทอร์เน็ตมาก ประกอบกับระบบดังกล่าวเป็นระบบที่ง่าย และสามารถนำไปใช้ได้จริงในปัจจุบัน ดังนั้นในงานวิจัยนี้จะได้นำเสนอระบบแมทช์เมคกิ้งเอเจนท์แบบกระจายโดยเลียนแบบการทำงานบางส่วนของโพรโทคอล DNS โครงแบบที่พัฒนาขึ้นจะจำลองมาจากการทำงานแบบลำดับชั้นของเนมเซิร์ฟเวอร์ที่ให้บริการค้นหาหมายเลขไอพี โดยกระจายการทำงานไปในเนมเซิร์ฟเวอร์แต่ละระดับชั้น ซึ่งในโครงแบบที่พัฒนาขึ้นนั้นจะประกอบไปด้วยแมทช์เมคกิ้งเอเจนท์ 3 ระดับ มาทำงานร่วมกันแบบลำดับชั้น และเพิ่มกลไกการแบ่งงานตามโดเมนของบริการที่เกี่ยวข้อง เพื่อให้สามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพ

1.2 วัตถุประสงค์ในการศึกษา

1) เพื่อประยุกต์เทคโนโลยีแมทช์เมคกิ้งเอเจนท์ให้สามารถใช้งานในระบบเอเจนท์ขนาดใหญ่ขึ้น และมีการทำงานที่มีประสิทธิภาพสูงขึ้น
2) แก้ไขปัญหาด้านแบนด์วิดท์ที่จะเกิดจากการใช้แมทช์เมคกิ้งเอเจนท์แบบรวมศูนย์

1.3 ขอบเขตการศึกษา

1) ศึกษาหลักการทำงานของแมทช์เมคกิ้งเอเจนท์และโพรโทคอล DNS
2) พัฒนาโครงแบบของระบบ DMAA ซึ่งนำหลักการของโพรโทคอล DNS เข้ามาประยุกต์ใช้ โดยเน้นการติดต่อระหว่างเอเจนท์ด้วยกันเองเท่านั้น
3) เปรียบเทียบประสิทธิภาพการทำงานของแมทช์เมคกิ้งเอเจนท์แบบรวมศูนย์และแนวทางใหม่ที่นำเสนอในงานวิจัยนี้ โดยสร้างระบบ DMAA และจำลองเอเจนท์ขึ้นมาเพื่อทำการทดสอบความสามารถของระบบที่นำเสนอ

1.4 ประโยชน์ที่คาดว่าจะได้รับ

งานวิจัยนี้จะช่วยให้รู้จักกับเทคโนโลยีทางด้านเอเจนท์มากขึ้นโดยเฉพาะอย่างยิ่งเทคโนโลยีแมทช์เมคกิ้งเอเจนท์ซึ่งเป็นหนึ่งในสาธารณูปโภคพื้นฐานของระบบเอเจนท์ และจะนำเสนอโครงแบบใหม่เพื่อช่วยให้สามารถนำแมทช์เมคกิ้งเอเจนท์ไปใช้กับระบบเอเจนท์ที่มีขนาดใหญ่ขึ้นได้ โดยเฉพาะอย่างยิ่งกับระบบเอเจนท์ที่กำลังเดินทางเข้าสู่โลกของอินเทอร์เน็ตในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
แนวคิดและทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะได้กล่าวถึงแนวคิดและทฤษฎีที่เกี่ยวข้องกับเทคโนโลยีแมทซ์เมคกิ้งเอเจนท์ ได้แก่ เทคโนโลยีเอเจนท์ โพรโทคอลแลกความรู้ เทคโนโลยีแมทซ์เมคกิ้งเอเจนท์ และเทคโนโลยีที่ใช้ในการปรับปรุงแมทซ์เมคกิ้งเอเจนท์ตามลำดับ นอกจากนี้ ยังได้อ้างอิงถึงงานวิจัยที่สำคัญในอดีตซึ่งเกี่ยวข้องกับเทคโนโลยีแมทซ์เมคกิ้งเอเจนท์ โดยมีวัตถุประสงค์เพื่อพิจารณาข้อดี-ข้อเสียของเทคโนโลยีที่มีอยู่เดิม และนำเสนอแนวคิดใหม่ในการปรับปรุงเทคโนโลยีดังกล่าวให้มีความสมบูรณ์มากยิ่งขึ้นต่อไป

2.1 เทคโนโลยีเอเจนท์

เอเจนท์เป็นแนวคิดในการใช้ตัวแทนในการปฏิบัติงานบางอย่างที่ผู้ใช้ต้องการ โดยที่รบกวนผู้ใช้เฉพาะในบางโอกาสที่จำเป็น Franklin and Graesser (1996) ได้นิยามเอเจนท์ไว้ว่าหมายถึงระบบคอมพิวเตอร์ที่สามารถรับรู้สภาพแวดล้อม และสามารถปฏิบัติงานเพื่อบรรลุวัตถุประสงค์ที่กำหนดได้ด้วยตนเอง โดยที่รบกวนผู้ใช้น้อยที่สุด ดังนั้นเอเจนท์จึงประกอบด้วยคุณสมบัติสำคัญ 2 ประการ ได้แก่ ความสามารถทางสังคม (Social Ability) และการมีภาวะอิสระ (Autonomy)

อย่างไรก็ตาม Petrie (1996) ได้แสดงให้เห็นว่า ในปัจจุบันยังขาดมาตรฐานในการแบ่งประเภทหรือระบุการเป็นเอเจนท์ และได้เสนอคุณสมบัติที่สำคัญที่ต้องมีในเอเจนท์ไว้ ได้แก่ ความสามารถทางสังคม กล่าวคือเอเจนท์ต้องสามารถสื่อสารแลกเปลี่ยนความรู้ระหว่างเอเจนท์ได้ นอกจากคุณสมบัติดังกล่าวแล้ว Tecuci (1998) ได้เสนอเพิ่มเติมไว้ว่าเอเจนท์ควรประกอบไปด้วยฐานความรู้ และเครื่องอนุมาน ทั้งนี้ เพื่อเพิ่มความฉลาดให้กับเอเจนท์ รวมถึงทำให้เอเจนท์สามารถมีภาวะอิสระในการทำงาน เนื่องจากสามารถนำความฉลาดที่ได้จากการมีฐานความรู้และเครื่องอนุมานมาใช้ในการควบคุมการกระทำของตนเองได้นั่นเอง

2.2 โพรโทคอลแลกความรู้

โพรโทคอลที่ใช้ในการติดต่อสื่อสารแลกเปลี่ยนความรู้ระหว่างเอเจนท์ เรียกว่า ACL (Agent Communication Language) (Labrou and Finin. 1999) โพรโทคอลที่นิยมใช้กันได้แก่ KQML (Knowledge Query and Manipulation Language) และ FIPA ACL โดยมีรายละเอียดดังนี้

2.2.1 KQML

KQML เป็นภาษาและโพรโทคอลระดับสูงเพื่อการแลกเปลี่ยนข้อมูลและความรู้ระหว่างเอเจนท์ โดยใช้ข้อความ ซึ่งไม่ขึ้นกับภาษาที่ใช้ในการแลกเปลี่ยนข้อมูลหรือความรู้ (เช่น Prolog หรือ KIF เป็นต้น) ไม่ขึ้นกับ Ontology (Gruber. 1993) และไม่ขึ้นกับโพรโทคอลที่ใช้ในการส่งข้อมูลที่อยู่ในชั้นต่ำกว่า (เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TCP/IP และ SMTP เป็นต้น) ข้อความแบบ KQML แบ่งเป็น 3 ส่วน ได้แก่ ส่วนเนื้อหา (Content) ส่วนการสื่อสาร (Communication) และส่วนข้อความ (Message) (Labrou and Finin. 1999)

- ส่วนเนื้อหา ได้แก่ ข้อมูลที่เอเจนท์ต้องการจะส่ง ซึ่งอยู่ในรูปแบบของภาษาและการแสดงความรู้ที่เอเจนท์นั้นๆ เลือกใช้ เช่น First-order Predicate Logic
- ส่วนการสื่อสาร ประกอบด้วยพารามิเตอร์ต่างๆ ที่จำเป็นสำหรับการสื่อสาร เช่น ชื่อเอเจนท์ผู้ส่ง ชื่อเอเจนท์ผู้รับ และรหัสประจำการสื่อสารแต่ละครั้ง เป็นต้น
- ส่วนข้อความ เป็นส่วนที่ระบุ Communication Primitive หรือที่ใน KQML เรียกว่า Performative (ประเภทของข้อความแบบ KQML เช่น เป็นคำสั่ง (Inform) คำถาม (Ask-one, Ask-all) หรือโฆษณา (Advertise) เป็นต้น) ที่เอเจนท์ผู้ส่งเลือกใช้ในการสื่อสารครั้งนั้นๆ

รูปที่ 2.1 แสดงตัวอย่างข้อความแบบ KQML โดยแบ่งเป็นข้อความเริ่มต้นการสื่อสาร และข้อความตอบกลับ ตามลำดับ เมื่อพิจารณาข้อความเริ่มต้นจะสามารถแบ่งเป็น 3 ส่วนได้ดังนี้

- ส่วนเนื้อหา ได้แก่ :content (PRICE IBM ?price)
- ส่วนการสื่อสาร ได้แก่ :sender joe :receiver stock-server และ :reply-with ibm-stock
- ส่วนข้อความ ได้แก่ :language LPROLOG และ :ontology NYSE_TICKS รวมถึง Performative ที่ชื่อ ask-one (บอกว่าข้อความนี้เป็นคำถาม) ด้วย

รูปที่ 2.1 ตัวอย่างข้อความ KQML (ก. ผู้ถาม และ ข. ผู้ตอบ)

ข้อความตอบกลับจะสามารถแบ่งเป็น 3 ส่วนได้เช่นกัน โดยในส่วนของ :content จะเป็นความรู้ที่เอเจนท์ stock-server ตอบกลับมายังเอเจนท์ joe ว่าขณะนี้หุ้น IBM มีราคา (PRICE) เท่ากับ 14 นั่นเอง

จากข้างต้นพบว่า นอกจาก KQML จะกำหนดรูปแบบของข้อความที่ใช้ในการสื่อสารระหว่างเอเจนท์แล้ว ยังเป็นการกำหนดขั้นตอนการสื่อสารระหว่างเอเจนท์โดยนัยด้วย โดยขั้นตอนดังกล่าวจะขึ้นอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับ Performative แต่ละประเภทของข้อความนั่นเอง ขั้นตอนการสื่อสารที่เกิดขึ้นจาก Performative หนึ่งๆ นั้นจะเรียกว่า การสนทนา (Conversation) ตัวอย่างเช่น หากมีข้อความ KQML ประเภท tell ถูกส่งเข้ามายังเอเจนท์ ข. โดยเอเจนท์ ก. จะแปลว่าสิ่งที่อยู่ในฟิลด์ :content นั้นอยู่ในฐานความรู้ของเอเจนท์ ก. และเอเจนท์ ก. มีความต้องการที่จะบอกให้กับเอเจนท์ ข. รับทราบถึงความรู้นั้น เอเจนท์ ข. อาจจะเชื่อหรือไม่เชื่อก็ได้ หากเชื่อเอเจนท์ ข. ก็จะจัดเก็บความรู้นั้นลงในฐานความรู้ของตนเองต่อไป เป็นต้น

2.2.2 FIPA ACL

FIPA ACL ถูกพัฒนาขึ้นโดยองค์กร FIPA (Foundation for Intelligent Physical Agent) ซึ่งเป็นองค์กรแรกๆ ที่พยายามร่างมาตรฐานเกี่ยวกับเอเจนท์ ทั้งที่เป็นซอฟท์แวร์และฮาร์ดแวร์ องค์กรนี้ก่อตั้งขึ้นในปี พ.ศ. 2539 FIPA ACL เป็นภาษาและโพรโทคอลที่มีลักษณะคล้ายคลึงกับ KQML โดยเฉพาะอย่างยิ่งในด้านของไวยากรณ์มีการใช้ Communication Primitive เช่นเดียวกันกับ KQML แต่ใน FIPA ACL นี้จะเรียกว่า Communicative Acts (CAs) แทน (Labrou and Finin. 1999; FIPA. 2001)

2.3 ระบบหลายเอเจนท์

การศึกษาเทคโนโลยีเอเจนท์มิได้จำกัดอยู่เฉพาะภายในเอเจนท์แต่ละตัวเท่านั้น แต่ยังครอบคลุมถึงงานวิจัยที่เกี่ยวข้องกับกลุ่มของเอเจนท์หลายๆ ตัวด้วย ในอดีต การศึกษาระบบที่ประกอบด้วยเอเจนท์ตั้งแต่สองตัวขึ้นไปนั้นจะอยู่ในเรื่องของ Distributed Artificial Intelligence (DAI) การศึกษาดังกล่าวสามารถแบ่งได้เป็น 2 แนวทาง ได้แก่ Distributed Problem Solving (DPS) และระบบหลายเอเจนท์ (Multi-Agent System - MAS) ซึ่ง MAS นั้นจะมีความหมายในเชิงทั่วไปมากกว่า DPS โดย MAS จะหมายถึงระบบทุกชนิดที่ประกอบด้วยเอเจนท์ที่มีภาวะอิสระในการทำงานตั้งแต่ 2 ตัวขึ้นไป ในขณะที่ DPS จะหมายถึงระบบที่ประกอบด้วยเอเจนท์ตั้งแต่ 2 ตัวขึ้นไปมาทำงานร่วมกันเพื่อแก้ไขปัญหาใดปัญหาหนึ่ง โดยมีการร่วมมือกันในการแบ่งปันและแลกเปลี่ยนความรู้เกี่ยวกับปัญหาและวิธีแก้ไขปัญหาที่ค่อยๆ ปรากฏขึ้น สำหรับระบบ DPS แนวทางในการติดต่อสื่อสารและกลไกการร่วมมือกันระหว่างแต่ละเอเจนท์จะถูกรวมเป็นส่วนหนึ่งของระบบไว้ล่วงหน้าแล้ว แต่ในระบบ MAS จะไม่มีการกำหนดแนวทางดังกล่าวล่วงหน้า เนื่องจากเอเจนท์ในระบบนี้จะมีอยู่ก่อนแล้ว มิได้เกิดขึ้นมาพร้อมกันเพื่อแก้ไขปัญหาใดปัญหาหนึ่งโดยเฉพาะ หากแต่จะมาทำงานร่วมกันก็ต่อเมื่อมีงานบางอย่างเกิดขอบเขตความสามารถของแต่ละเอเจนท์

ลักษณะโดยทั่วไปของระบบ MAS มีดังนี้ (Jennings *et al.* 1998)

- เอเจนท์แต่ละตัวจะมีข้อมูลหรือความสามารถเพียงบางอย่างเท่านั้น ซึ่งไม่เพียงพอต่อการแก้ปัญหาที่เกิดขึ้นในการปฏิบัติงานของแต่ละเอเจนท์
- ไม่มีระบบควบคุมส่วนกลาง
- ข้อมูลกระจัดกระจายไปอยู่ในเอเจนท์หลายๆ ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การติดต่อสื่อสารจะเป็นแบบอะซิงโครนัส

งานวิจัยด้าน MAS ที่ได้รับความสนใจในอดีตนั้นประกอบไปด้วย การเพิ่มประสิทธิภาพให้กับระบบ MAS การนำระบบ MAS ไปทำงานร่วมกับสารสนเทศแบบดั้งเดิมโดยเฉพาะระบบฐานข้อมูลและระบบผู้เชี่ยวชาญ และการเพิ่มขีดความสามารถในการแก้ปัญหาให้กับระบบ MAS ในกรณีที่ข้อมูล ความเชี่ยวชาญ และการควบคุมกระจายอยู่ตามแต่ละส่วนของระบบในรูปแบบต่างๆ สำหรับงานวิจัยทางด้าน DAI นั้นจะเน้นเฉพาะการค้นหากลไกการทำงานร่วมกันในระดับสูงของผู้แก้ปัญหาที่ชาญฉลาดเพื่อแก้ไขปัญหาประเภทใดประเภทหนึ่งอย่างมีประสิทธิภาพ แต่จะไม่รวมถึงกลไกการทำงานร่วมกันในระดับต่ำ เช่น การกระจายกระบวนการทำงานระหว่างเครื่องคอมพิวเตอร์ เป็นต้น อย่างไรก็ตาม DPS และ MAS นั้นเป็นส่วนหนึ่งของ DAI ดังนั้นงานวิจัยทางด้าน DPS และ MAS จึงเน้นไปที่การนำเสนอกลไกการทำงานร่วมกันระดับสูงเช่นเดียวกันกับงานวิจัยทางด้าน DAI โดยจะนำเสนอกลไกที่มีประสิทธิภาพ เพิ่มขีดความสามารถการแก้ไขปัญหาให้แก่ระบบ และสามารถนำไปประยุกต์ใช้กับระบบสารสนเทศแบบอื่นได้โดยง่าย

Jennings *et al.* (1998) ได้แบ่งระบบ MAS ตามประเภทการทำงานร่วมกันออกเป็น 2 ประเภท ได้แก่ Cooperative MAS และ Self-interested MAS

ประเภทแรก Cooperative MAS เป็นระบบที่เอเจนท์จะทำงานร่วมกันตามวัตถุประสงค์หลักของกลุ่ม ซึ่งได้แก่ระบบ DPS นั่นเอง MAS ประเภทนี้แบ่งได้เป็น 2 ชนิด คือ Common Goals-based MAS และ Explicit Teamwork Model-based MAS ชนิดแรก Common Goals-based MAS เป็น MAS ที่วัตถุประสงค์ของกลุ่มจัดเป็นวัตถุประสงค์หลักของแต่ละเอเจนท์ เอเจนท์ในระบบชนิดนี้จะใช้กลไกในการร่วมมือที่เน้นประสิทธิภาพของกลุ่มเป็นหลัก ตัวอย่างเช่น PGP (Partial Global Planning) (Durfee. 1988) เป็นต้น ส่วน Explicit Teamwork Model-based MAS จะเป็น Common Goals-based MAS ที่มีการกำหนดแบบจำลองในการทำงานเป็นกลุ่มของเอเจนท์ไว้ล่วงหน้า ตัวอย่างของระบบเอเจนท์ประเภทนี้ได้แก่ ระบบเอเจนท์สำหรับฟุตบอล RoboCup (Veloso *et al.* 1997) ซึ่งประกอบไปด้วยหุ่นยนต์จำนวน 11 ตัวมาทำงานร่วมกันเพื่อจุดประสงค์หลักในการนำลูกบอลไปไว้ในประตูฝั่งตรงข้ามให้ได้มากที่สุด เอเจนท์แต่ละตัวจะมีความสามารถในการหลบหลีกสิ่งกีดขวางและควบคุมลูกบอล รวมถึงสามารถรับรู้ตำแหน่งของตนเอง ตำแหน่งของเพื่อนร่วมทีม ตำแหน่งของเอเจนท์ฝ่ายตรงข้าม ตำแหน่งของประตูทั้งสองฝั่ง และตำแหน่งของลูกบอลในสนามด้วย เอเจนท์แต่ละตัวจะถูกกำหนดแนวทางปฏิบัติงานร่วมกับเอเจนท์อื่นไว้ล่วงหน้า ตัวอย่างเช่น เอเจนท์ที่ทำหน้าที่เป็นผู้รักษาประตูจะมีหน้าที่หลักคือป้องกันไม่ให้ลูกบอลเข้าประตูของฝั่งตนเอง โดยประสานงานกับเอเจนท์ที่ทำหน้าที่เป็นกองหลัง นอกจากนี้ตำแหน่งของเอเจนท์ทั้งสองประเภทบนสนามนั้นจะถูกออกแบบไว้ล่วงหน้าแล้ว เช่นหากผู้รักษาประตูกำลังวิ่งออกมาจากเขตประตู ในขณะที่มีเอเจนท์ฝ่ายตรงข้ามวิ่งเข้ามา จะต้องให้เอเจนท์กองหลังที่อยู่ใกล้ประตูที่สุดวิ่งตามเอเจนท์ฝ่ายตรงข้ามเข้าไปเพื่อป้องกันการทำประตู เป็นต้น ทั้งนี้การที่ต้องออกแบบการทำงานร่วมกันของกลุ่มเอเจนท์ประเภทนี้ไว้ล่วงหน้ามีสาเหตุมาจากการที่ระบบเอเจนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

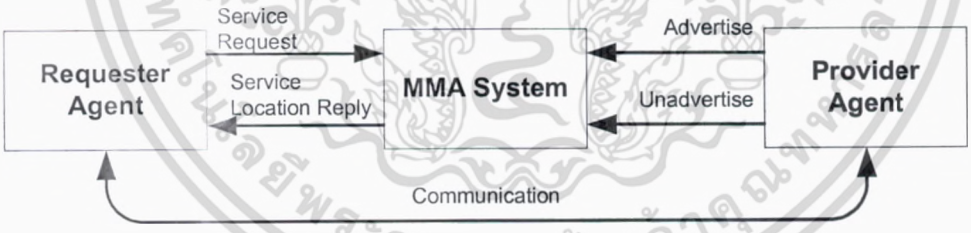
แบบนี้ต้องการความรวดเร็วในการคำนวณ หากมิได้กำหนดแนวทางการทำงานร่วมกันไว้ล่วงหน้าแล้วจะทำให้ระบบทำงานได้ช้ามาก (Jennings *et al.* 1998)

ประเภทที่สอง Self-interested MAS เป็นระบบเอเจนท์ที่แต่ละตัวจะทำงานของตนเป็นหลัก เอเจนท์จะมาร่วมมือกันก็ต่อเมื่อไม่สามารถทำงานใดงานหนึ่งได้ด้วยตนเองเท่านั้น ดังนั้นระบบเอเจนท์ประเภทนี้จึงเป็นประเภทหลักของระบบ MAS นั่นเอง ในระบบประเภทนี้จะต้องมีกลไกการเจรจาต่อรองระหว่างเอเจนท์เพื่อแก้ไขความขัดแย้งที่เกิดขึ้นในเจตนาของแต่ละเอเจนท์และส่งเสริมความร่วมมือของเอเจนท์ภายในระบบ ระบบเอเจนท์ประเภทนี้มีประเด็นปัญหาที่สำคัญที่ต้องแก้ไขเสียก่อนจึงจะสามารถทำงานได้จริง ปัญหาดังกล่าวได้แก่ ปัญหาการค้นหาเอเจนท์ที่สามารถให้บริการที่ต้องการได้ ดังนั้นเทคโนโลยีแมทช์เมคกิ้งเอเจนท์จึงได้ถือกำเนิดขึ้นมาเพื่อแก้ไขปัญหาดังกล่าว ทั้งนี้ระบบแมทช์เมคกิ้งเอเจนท์นั้นจะทำงานอยู่ภายใน Self-interested MAS ซึ่งเป็นลักษณะของ MAS โดยทั่วไป ในขณะที่ตัวเองนั้นจะมีลักษณะการทำงานเป็นแบบ DPS หรือ Cooperative MAS ชนิด Explicit Teamwork Model-based MAS เพื่อให้มีประสิทธิภาพสูงในการให้บริการจับคู่แก่เอเจนท์ภายในระบบที่ตนเองรับผิดชอบ

2.4 เทคโนโลยีแมทช์เมคกิ้งเอเจนท์

2.4.1 แมทช์เมคกิ้งเอเจนท์

แมทช์เมคกิ้งเอเจนท์เป็นเอเจนท์ที่ทำหน้าที่ช่วยค้นหาเอเจนท์ที่สามารถให้บริการที่เอเจนท์หนึ่งๆ ร้องขอมาได้ (Decker *et al.* 1996) มีลักษณะการทำงานโดยทั่วไปเป็นดังรูปที่ 2.2



รูปที่ 2.2 การทำงานทั่วไปของแมทช์เมคกิ้งเอเจนท์

เมื่อเอเจนท์ผู้ให้บริการ (Provider Agent) เข้าสู่ระบบ จะลงทะเบียนกับแมทช์เมคกิ้งเอเจนท์โดยทำการโฆษณาบริการที่ตนเองสามารถให้กับเอเจนท์อื่นๆ ได้ไปยังแมทช์เมคกิ้งเอเจนท์ซึ่งเรียกว่าโฆษณา (Advertisement) และข้อมูลนี้จะถูกจัดเก็บอยู่ที่แมทช์เมคกิ้งเอเจนท์ดังกล่าว เมื่อมีเอเจนท์ผู้ขอรับบริการหรือเอเจนท์ผู้ร้องขอ (Requester Agent) ร้องขอบริการเข้ามายังแมทช์เมคกิ้งเอเจนท์ แมทช์เมคกิ้งเอเจนท์จะทำการค้นหาฐานข้อมูลการโฆษณาเพื่อเปรียบเทียบว่ามีโฆษณาบริการของเอเจนท์ผู้ให้บริการรายใดที่สอดคล้องกับคำร้องขอบริการนั้น เมื่อสิ้นสุดการค้นหาแล้วแมทช์เมคกิ้งเอเจนท์จะตอบกลับไปยังเอเจนท์ผู้ขอรับบริการว่ามีเอเจนท์ผู้ให้บริการรายใดบ้างที่สามารถให้บริการดังกล่าวได้ พร้อมกับระบุที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่หรือวิธีการติดต่อกับเอเจนท์ดังกล่าว เพื่อให้เอเจนท์ผู้ขอรับบริการสามารถติดต่อกับเอเจนท์ดังกล่าวโดยตรงได้ต่อไป

Kuokka and Harada (1998) ได้นำเสนอไว้ว่าองค์ประกอบหลักที่สำคัญของแมทช์เมคกิ้งเอเจนท์แบ่งเป็น 2 ส่วน ได้แก่ กลไกการจับคู่ (Matching Engine) และโพรโทคอลแลกความรู้ (Knowledge-Sharing Protocol) ซึ่งจะขอกล่าวถึงรายละเอียดถัดไป

2.4.2 กลไกการจับคู่

องค์ประกอบแรกของแมทช์เมคกิ้งเอเจนท์ได้แก่ อัลกอริทึมที่แมทช์เมคกิ้งเอเจนท์ใช้ในการเปรียบเทียบคำร้องขอบริการที่เข้ามากับโฆษณาความสามารถที่เก็บอยู่ในตัวแมทช์เมคกิ้งเอเจนท์เอง เพื่อค้นหาโฆษณาที่มีความสอดคล้องกับคำร้องขอบริการ กล่าวคือ เป็นกระบวนการในการค้นหาเอเจนท์ผู้ให้บริการที่สามารถให้บริการที่เอเจนท์ผู้ขอรับบริการต้องการได้ เทคนิคที่นิยมใช้กันในปัจจุบันได้แก่ Word-to-Word Matching และ Shared-Concept Matching (Kuokka and Harada. 1998)

2.4.2.1 Word-to-Wword Matching ได้แก่ การจับคู่โดยอาศัยการเปรียบเทียบคำ โดยการนำข้อความในส่วนของ :content ของข้อความแบบ KQML มาทำการเปรียบเทียบรูปแบบและคำศัพท์แบบคำต่อคำ ตัวอย่างเช่น มีเอเจนท์ผู้ให้บริการรายหนึ่งโฆษณาเข้ามาดังรูปที่ 2.3 (ก) ซึ่งเป็นการแสดงว่าเอเจนท์นี้สามารถบอกได้ว่าเมืองหนึ่งๆ มีที่ตั้งอยู่ในประเทศอะไร เป็นต้น

```
(advertise
        : sender Provider01
        : content (tell-about (is-in ?City ?Country))
        : receiver MMA01
        : language CLIPS
        : ontology Geography)
```

(ก)

```
(request
        : sender Requester01
        : content (tell-about (is-in pusan ?Country))
        : receiver MMA01
        : reply-with Requester0100001
        : language CLIPS
        : ontology Geography)
```

(ข)

รูปที่ 2.3 ตัวอย่างการจับคู่แบบ Word-to-Word Matching

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีเอเจนท์ผู้ขอรับบริการติดต่อเข้ามายังแมทช์เมคกิ้งเอเจนท์ดังรูปที่ 2.3 (ข) เพื่อสอบถามว่ามีเอเจนท์ให้บริการใดสามารถบอกได้ว่าเมืองปูซานมีที่ตั้งอยู่ในประเทศอะไร แมทช์เมคกิ้งเอเจนท์ก็จะนำข้อความในส่วน :content ไปเปรียบเทียบคำศัพท์แบบคำต่อคำกับโฆษณาที่เก็บไว้ ถ้าตรงกันทั้งหมดก็จะค้นหาชื่อหรือวิธีติดต่อเอเจนท์ดังกล่าวขึ้นมาเก็บไว้ในรายการที่จะตอบกลับไปยังเอเจนท์ผู้ขอรับบริการ โดยรายการดังกล่าวจะประกอบไปด้วยชื่อและ/หรือวิธีติดต่อเอเจนท์ผู้ให้บริการทั้งหมดที่สามารถให้บริการที่ร้องขอมาได้

อย่างไรก็ตาม อัลกอริทึมประเภทนี้มีข้อเสียที่สำคัญได้แก่ หากข้อความในโฆษณาและคำร้องขอใช้คำศัพท์ต่างกัน แต่เป็นคำที่มีความหมายเหมือนกัน (Synonym) จะทำให้ไม่สามารถจับคู่ข้อความประเภทดังกล่าวได้ ซึ่งในปัจจุบันได้มีวิธีการแก้ไขข้อด้อยดังกล่าวแล้วโดยการนำหลักการของ Ontology เข้ามาใช้ ดังจะได้กล่าวต่อไป

2.4.2.2 Shared-Concept Matching เป็นอัลกอริทึมที่นำหลักการของ Ontology เข้ามาใช้ประกอบการจับคู่ ทั้งนี้เพื่อให้การจับคู่มีความแม่นยำสูงขึ้น โดยมีการเพิ่มฐานความรู้ที่กำหนด Shared Concept เข้ามาใช้ในกระบวนการจับคู่ ตัวอย่างเช่น หากข้อความในส่วน :content ในรูปที่ 2.3 (ข) เปลี่ยนเป็น

:content (tell-about (locate-in pusan ?Country))

ถ้าใช้ Word-to-word Matching จะทำให้เอเจนท์ผู้ให้บริการนั้นไม่ถูกเลือก แต่หากใช้ Shared-concept Matching แล้วจะทำให้โฆษณาดังกล่าวจับคู่กับคำร้องขอได้ เนื่องจาก is-in และ locate-in มีความหมายเหมือนกันสำหรับ Ontology "Geography" นั่นเอง

2.4.3 โพรโทคอลแลกความรู้

องค์ประกอบที่สำคัญอีกประการหนึ่งของแมทช์เมคกิ้งเอเจนท์ได้แก่ โพรโทคอลที่ใช้ในการติดต่อสื่อสารแลกเปลี่ยนความรู้ระหว่างเอเจนท์ โดยโพรโทคอลที่ได้รับความนิยมได้แก่ KQML และ FIPA ACL อย่างไรก็ตาม KQML นั้นได้รับความนิยมมากกว่า FIPA ACL เป็นอย่างมากและกลายมาเป็นมาตรฐานตามความนิยม (de facto standard) ของโพรโทคอลประเภทนี้แล้ว รายละเอียดอื่นๆ ของโพรโทคอลแลกความรู้ได้ถูกกล่าวไว้ในหัวข้อ 2.2 ข้างต้นแล้ว

2.4.4 ผลงานวิจัยที่เกี่ยวข้อง

ในปัจจุบันได้มีผู้พัฒนาระบบแมทช์เมคกิ้งเอเจนท์ขึ้นมาบ้างแล้ว แต่ละระบบมีจุดเด่นจุดด้อยต่างกัน ในหัวข้อนี้จะยกตัวอย่างที่สำคัญของระบบดังกล่าว ดังนี้

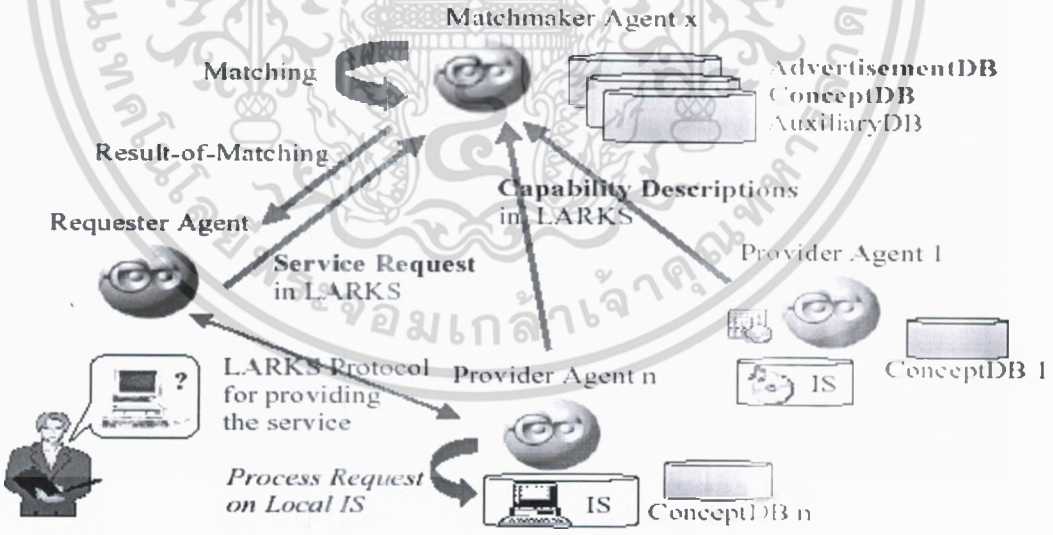
2.4.4.1 Expert Finder (Vivacqua. 1999) เป็นระบบแมทช์เมคกิ้งเอเจนท์ระดับพื้นฐานที่สุด ทำหน้าที่ช่วยในการค้นหาผู้เชี่ยวชาญเฉพาะด้านที่สามารถให้คำปรึกษาที่เหมาะสมกับระดับความรู้ของผู้ใช้แต่ละคน หลักการที่สำคัญคือผู้ใช้ (ผู้ขอรับบริการ) และผู้เชี่ยวชาญ (ผู้ให้บริการ) ทุกคนในระบบนี้จะต้องมีเอเจนท์ Expert Finder ซึ่งเอเจนท์แต่ละตัวนั้นจะต้องมีรายชื่อและที่อยู่ของเอเจนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อื่นๆ ทั้งหมดที่อยู่ในระบบนี้ จากคุณสมบัตินี้ ส่งผลให้ระบบแมทช์เมคกิ้งเอเจนท์ประเภทนี้ไม่สามารถนำไปใช้กับระบบที่มีเอเจนท์จำนวนมากได้ เนื่องจากจะทำให้เอเจนท์ทุกตัวมีฐานความรู้เกี่ยวกับข้อมูลของเอเจนท์อื่นๆ ที่ใหญ่มาก อีกทั้งหากเอเจนท์สามารถเข้าออกระบบได้อย่างอิสระแล้ว การปรับปรุงข้อมูลความสามารถของเอเจนท์อื่นๆ จะทำให้ระบบเต็มไปด้วยข้อความที่วิ่งไปมาในระบบเพื่อปรับปรุงข้อมูลดังกล่าวให้มีความทันสมัยเพียงพออยู่เสมอ

2.4.4.2 Middle Agent (Sycara *et al.* 1999) เป็นโครงแบบแมทช์เมคกิ้งเอเจนท์ที่เป็นที่รู้จักและเข้าใจกันทั่วไปมากที่สุด มีกลไกการทำงานดังแสดงในรูปที่ 2.4

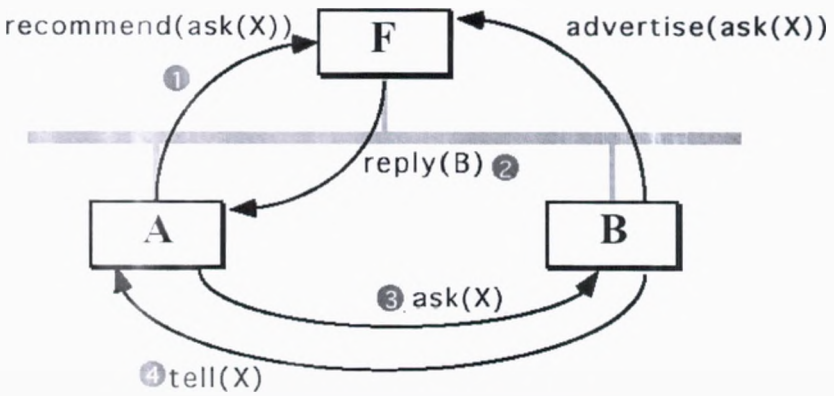
หลักการทำงานของระบบแมทช์เมคกิ้งเอเจนท์แบบนี้ได้แก่ เอเจนท์ผู้ให้บริการจะทำการโฆษณาตัวเองกับ Middle Agent (แมทช์เมคกิ้งเอเจนท์ประเภทหนึ่ง) โดยส่งความสามารถบริการที่ตนเองสามารถให้กับเอเจนท์อื่นๆ ได้ (Capability Descriptions) ไปยัง Middle Agent และข้อมูลนี้จะถูกจัดเก็บอยู่ที่ฐานข้อมูลโฆษณาของ Middle Agent เมื่อมีเอเจนท์ผู้ขอรับบริการร้องขอบริการเข้ามายัง Middle Agent ตัว Middle Agent นี้จะทำการค้นหาฐานข้อมูลของตนเอง เพื่อเปรียบเทียบว่ามีบริการของเอเจนท์ใดที่โฆษณาเข้ามาซึ่งสอดคล้องกัน จากนั้น Middle Agent จะตอบกลับไปยังเอเจนท์ผู้ขอรับบริการว่ามีเอเจนท์ใดที่สามารถให้บริการดังกล่าวได้บ้าง เพื่อให้ติดต่อไปยังเอเจนท์ดังกล่าวและขอรับบริการโดยตรงจากเอเจนท์นั้น



รูปที่ 2.4 การทำงานของ Middle Agent

การทำงานของระบบนี้เป็นกระบวนการที่ได้รับความนิยมมากที่สุด และมีลักษณะเหมือนกับการทำงานในรูปที่ 2.2 หากเปลี่ยนจากการใช้ LARKS (ACL ประเภทหนึ่งที่ถูกสร้างขึ้นมาใช้กับ Middle Agent โดยเฉพาะ) มาเป็นการใช้ KQML แทนแล้ว จะมีกลไกการสนทนาดังรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 กลไกการสนทนาของ Middle Agent เมื่อเปลี่ยนมาใช้ KQML

แมทช์เมคกิ้งเอเจนท์ที่มีรูปแบบการทำงานเหมือนรูปที่ 2.5 จะเรียกว่า Recommender หลักการสำคัญของแมทช์เมคกิ้งเอเจนท์แบบนี้คือจะตอบเฉพาะชื่อของเอเจนท์ผู้ให้บริการกลับมา หลังจากนั้นเอเจนท์ผู้ร้องขอจะต้องไปติดต่อขอรับบริการจากเอเจนท์ดังกล่าวเองต่อไป นอกจากนี้ ยังมีแมทช์เมคกิ้งเอเจนท์รูปแบบอื่นๆ อีก (Finin *et al.* 1994) เช่น Broker และ Subscriber เป็นต้น ระบบแมทช์เมคกิ้งเอเจนท์ประเภทนี้สามารถแก้ไขปัญหาที่เกิดขึ้นในระบบก่อนอย่างเช่น Expert Finder ได้ โดยอาศัยการรวมหน้าที่ในการจับคู่เข้ามาไว้ในเอเจนท์เพียงตัวเดียวซึ่งก็คือ Middle Agent เมื่อมีเอเจนท์ใดๆ เข้า-ออกระบบแล้วจะมีการปรับปรุงโฆษณาความสามารถของเอเจนท์ดังกล่าวเกิดขึ้นที่เอเจนท์เพียงตัวเดียวซึ่งก็คือ Middle Agent นั่นเอง

อย่างไรก็ตาม ระบบนี้ได้ก่อให้เกิดปัญหาใหม่ขึ้นมาได้แก่ การให้บริการจับคู่นั้นถูกกระทำโดยเอเจนท์เพียงตัวเดียว ทำให้การให้บริการจับคู่นั้นช้ากว่าระบบก่อน

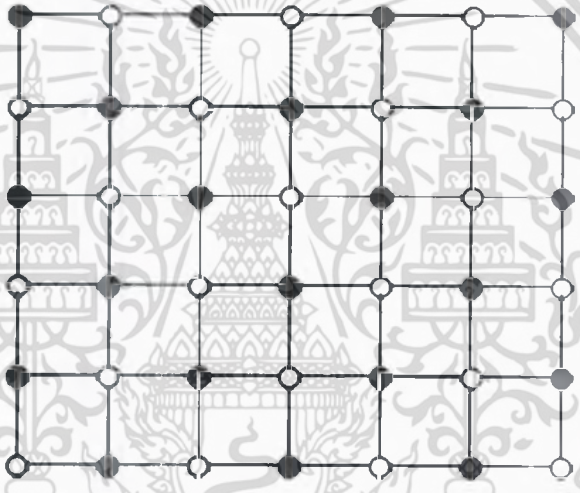
2.4.4.3 Lattice Agents (Shehory. 2000) งานวิจัยนี้นำเสนอกลไกในการค้นหาเอเจนท์โดยที่ไม่จำเป็นต้องมีแมทช์เมคกิ้งเอเจนท์ แต่ใช้หลักการของการกระจายข้อมูลส่วนกลางไปยังเอเจนท์แต่ละตัว โดยได้ชี้ให้เห็นว่าการใช้แมทช์เมคกิ้งเอเจนท์มีข้อเสียที่สำคัญดังนี้

- แมทช์เมคกิ้งเอเจนท์เป็นสาธารณูปโภคพิเศษที่ต้องจัดให้มีเพิ่มขึ้น
- มีต้นทุนที่เพิ่มขึ้นจากการที่ต้องติดต่อแมทช์เมคกิ้งเอเจนท์ก่อนเสมอ
- เอเจนท์แต่ละตัวต้องมีโพรโทคอลที่ใช้ติดต่อกับแมทช์เมคกิ้งเอเจนท์ที่ตนเองสังกัดด้วย

หลักการของ Lattice Agents คือ นำฟังก์ชันในการค้นหาเอเจนท์ไปไว้ที่ตัวเอเจนท์ทุกๆ ตัวแทนการใช้แมทช์เมคกิ้งเอเจนท์ ซึ่งจะเห็นได้ว่ามีความคล้ายคลึงกับระบบ Expert Finder มาก อย่างไรก็ตามระบบนี้มีความแตกต่างที่สำคัญกับระบบ Expert Finder ตรงที่เอเจนท์แต่ละตัวไม่จำเป็นที่จะต้องเก็บข้อมูลเกี่ยวกับเอเจนท์ทั้งหมดที่มีอยู่ในระบบ แต่เอเจนท์แต่ละตัวจะรู้จักเอเจนท์อื่นเป็นจำนวนหนึ่งเท่านั้น เมื่อเอเจนท์ต้องการค้นหาเอเจนท์ใดๆ ก็จะทำการแลกเปลี่ยนข้อมูลซึ่งกันและกันภายในระบบนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

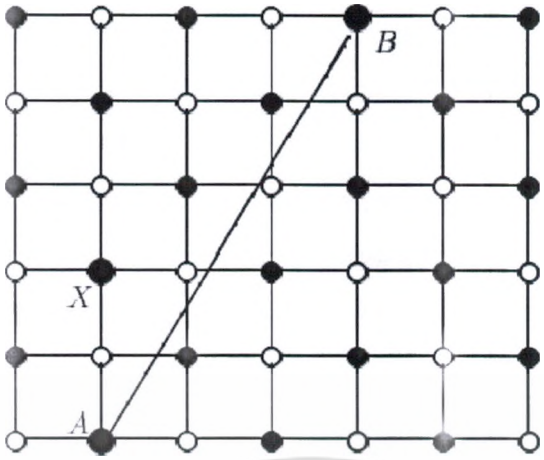
ในงานนี้มีสมมติฐานที่สำคัญคือกลุ่มเอเจนท์จะมีโครงสร้างแบบ Lattice ดังรูปที่ 2.6 กล่าวคือ เอเจนท์หนึ่งๆ จะติดต่ออยู่กับเอเจนท์ข้างเคียง 4 ตัวเท่ากัน และจะมีข้อมูลทั้งหมดเกี่ยวกับเอเจนท์ข้างเคียงดังกล่าว หรืออีกนัยหนึ่งก็คือ เอเจนท์หนึ่งๆ จะรู้จักเอเจนท์ข้างเคียงจำนวน 4 ตัวเป็นอย่างดี และรู้จักความสามารถของเอเจนท์ดังกล่าวทั้งหมด การทำเช่นนี้ก็เพื่อที่จะลดขนาดของฐานความรู้ความสามารถที่จัดเก็บอยู่ที่เอเจนท์แต่ละตัวลง และส่งผลให้การค้นหาความสามารถที่เข้าคู่กับคำร้อง ณ เอเจนท์หนึ่งๆ นั้นมีความรวดเร็วมากขึ้น เนื่องจากมีขนาดของฐานความรู้ที่ต้องสืบค้นเล็กลง ทั้งนี้หากไม่สามารถค้นหาบริการที่ต้องการพบ คำร้องดังกล่าวจะถูกส่งต่อไปยังเอเจนท์ข้างเคียงเพื่อใช้บริการจับคู่ของเอเจนท์นั้นต่อไป และจะมีกระบวนการเช่นนี้ไปเรื่อยๆ จนกว่าจะมีเอเจนท์ใดเอเจนท์หนึ่งสามารถค้นหาบริการที่เข้าคู่กับคำร้องนั้นได้ หรือจนกว่าจะไม่มีเอเจนท์ให้ส่งต่อแล้ว



รูปที่ 2.6 กลุ่มเอเจนท์แบบ Lattice

ระบบเอเจนท์แบบนี้มีข้อด้อยที่สำคัญคือ ในระบบเอเจนท์ขนาดใหญ่ เราไม่สามารถทราบได้เลยว่าการส่งต่อคำร้องนั้นจะสิ้นสุดเมื่อใด ดังนั้นการค้นหาบริการที่เหมาะสมกับคำร้องหนึ่งจึงอาจใช้เวลานานมาก งานวิจัยนี้พยายามแก้ปัญหาดังกล่าวโดยอาศัยการเพิ่มจำนวนเอจเข้าไปในระบบ กล่าวคือเพิ่มจำนวนเอเจนท์ที่เอเจนท์แต่ละตัวรู้จัก แล้วดูว่าระบบจะมีประสิทธิภาพสูงขึ้นมากน้อยเพียงใด ตัวอย่างเช่น ในรูปที่ 2.7 หากเพิ่มเอจระหว่างโหนด A และ B แล้วจะทำให้การค้นหาบริการที่เข้าคู่กับคำร้องที่เริ่มต้นจากเอเจนท์อื่น เช่น เอเจนท์ X มีความรวดเร็วสูงขึ้นมากน้อยเพียงใด เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

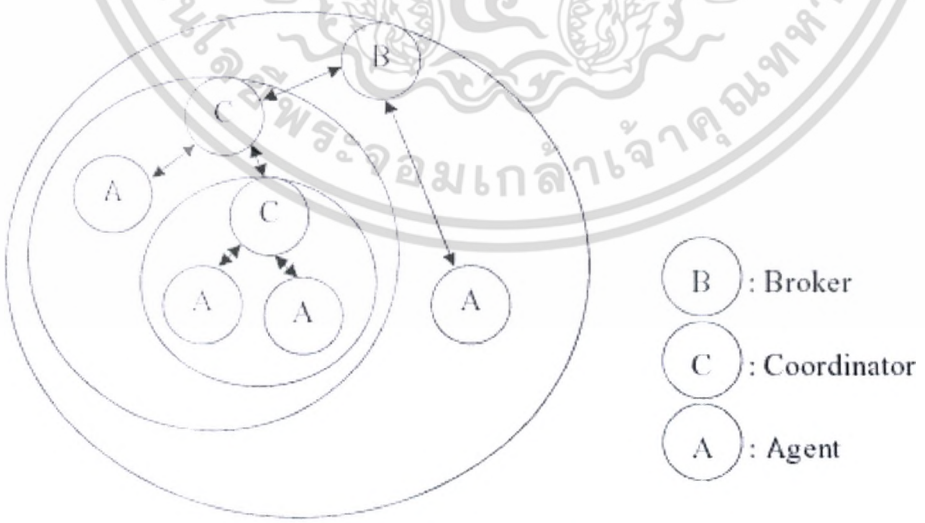


รูปที่ 2.7 การปรับปรุงประสิทธิภาพของ Lattice Agents

ผู้วิจัยได้อ้างไว้ว่าวิธีนี้สามารถลดความซับซ้อนของอัลกอริทึมลงไปได้มาก จนเหลือความซับซ้อนของจำนวนการติดต่อระหว่างเอเจนท์สำหรับการค้นหาบริการของเอเจนท์หนึ่งๆ แค่ $O(\sqrt{n})$ เท่านั้น

ข้อเสียเพิ่มเติมของงานวิจัยนี้คือยังไม่ใกล้เคียงกับการนำไปประยุกต์ใช้จริง เนื่องจากไม่มีระบบเอเจนท์ใดที่จะมีโครงสร้างเป็นแบบ Lattice ดังรูปที่ 2.6 อย่างไรก็ตาม งานวิจัยนี้จัดเป็นงานวิจัยแรกที่กระจายงานจับคู่ไปยังเอเจนท์หลายๆ ตัว ซึ่งจัดเป็นประเด็นที่น่าสนใจอย่างยิ่ง

2.4.4.4 A4 System (Cao *et al.* 2000) ระบบ A4 เป็นระบบเอเจนท์แบบกระจายที่มีแนวคิดหลักคือการแบ่งกลุ่มของเอเจนท์ออกเป็นชั้นๆ ดังรูปที่ 2.8



รูปที่ 2.8 โครงสร้างของเอเจนท์แบบ A4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอเจนท์ในระบบ A4 เมื่อแรกเริ่มเข้าสู่ระบบจะต้องค้นหาเอเจนท์อื่นที่ตนเองจะลงทะเบียนด้วย โดยเอเจนท์ที่มีคนมาลงทะเบียนด้วยแล้วจะเรียกว่า Coordinator เมื่อลงทะเบียนแล้วเอเจนท์ดังกล่าวก็จะเกาะติดอยู่กับ Coordinator นั้นจนกระทั่งเอเจนท์ดังกล่าวออกจากระบบไป ทั้งนี้เอเจนท์แต่ละตัวจะมี Coordinator ได้อย่างมากที่สุดเพียงแค่ 1 ตัวเท่านั้น ส่วนเอเจนท์ที่ไม่ต้องลงทะเบียนกับเอเจนท์อื่นใดเลย จะได้แก่เอเจนท์ที่อยู่บนสุดของลำดับชั้น ซึ่งจะเรียกเอเจนท์ดังกล่าวว่า Broker นั่นเอง

ความสามารถของเอเจนท์หนึ่งๆ ในระบบ A4 จะเรียกว่า ACT (Agent Capability Table) ดังนั้น การค้นหาบริการหนึ่งๆ ก็คือการค้นหาและ/หรือจัดการกับ ACT นั่นเอง โดยที่การค้นหาดังกล่าวจะเป็นไปตาม Discovery Model แบบต่างๆ ได้แก่ การใช้แคช (C_ACT) การเก็บข้อมูลเกี่ยวกับเอเจนท์ที่มาลงทะเบียนด้วย (L_ACT) และการเก็บ L_ACT ของเอเจนท์ที่อยู่ในระดับที่เหนือกว่า (G_ACT)

ในงานนี้ได้จัดให้มีการจำลองระบบเอเจนท์ตามรูปแบบที่เสนอมา เพื่อทดสอบหาว่าการใช้ Discovery Model แบบใดจะให้ประสิทธิภาพการค้นหาสูงสุด การทดสอบดังกล่าวมีโครงสร้างของเอเจนท์ดังตารางที่ 2.1 ซึ่งเป็นโครงสร้างแบบลำดับชั้นที่มีความลึกเท่ากับ 2 และมีจำนวนเอเจนท์ทั้งหมด 251 ตัว การทดสอบอาศัยการนับจำนวนการติดต่อระหว่างเอเจนท์มาใช้เป็นต้นทุนในการวัดประสิทธิภาพของระบบ ต้นทุนของระบบแมทช์เมคกิ้งเอเจนท์มีสมการคือ

$$c = d + a \tag{2.1}$$

เมื่อ d คือจำนวนการติดต่อระหว่างแมทช์เมคกิ้งเอเจนท์ที่ใช้ในการค้นหาโฆษณาที่เข้าคู่กับคำร้องและ a คือจำนวนการติดต่อที่ใช้ในการอัพเดตโฆษณาระหว่างแมทช์เมคกิ้งเอเจนท์ หากกำหนดให้ r เป็นจำนวนคำร้องขอทั้งหมดที่เข้ามาในระบบแมทช์เมคกิ้งเอเจนท์ภายในช่วงเวลาที่กำหนดแล้ว จะได้สมการแสดงประสิทธิภาพของระบบแมทช์เมคกิ้งเอเจนท์แบบ A4 System ดังสมการที่ 2.2

ตารางที่ 2.1 โครงสร้างของระบบเอเจนท์แบบ A4 ที่ใช้ทดสอบ (Cao *et al.* 2000)

Agents	Upper Agent
gem	-
sprite~0.....sprite~49	gem
tup~0.....tup~49	sprite~9
cola~0.....cola~49	sprite~19
tango~0.....tango~49	sprite~29
pepsi~0.....pepsi~49	sprite~39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$e = \frac{r}{c} \qquad (2.2)$$

เมื่อ e คือประสิทธิภาพของระบบแมทช์เมคกิ้งเอเจนท์

ผลลัพธ์จากการทดลองดังกล่าวได้แสดงไว้ดังตารางที่ 2.2

ตารางที่ 2.2 ผลลัพธ์การทดสอบระบบ A4 แบบต่างๆ (Cao *et al.* 2000)

Experiment No:		1	2	3	4	5	6
Strategies	Using C_ACT	-	✓	✓	✓	✓	✓
	Using L_ACT	-	-	✓	✓	✓	✓
	Using G_ACT	-	-	-	✓	✓	✓
	Updating L_ACT	-	-	-	-	✓	✓
	Updating G_ACT	-	-	-	✓	✓	✓
	Advertising L_ACT	-	-	✓	✓	✓	✓
	Multicasting L_ACT	-	-	-	-	-	✓
Result	r	12296	12355	12576	12560	12645	11715
	a	0	0	5604	8051	10172	285148
	d	65595	51113	7435	6901	6910	7056
	e	0.19	0.24	0.96	0.84	0.74	0.04

จากตารางที่ 2.2 สามารถสรุปได้ว่าการใช้ C_ACT ร่วมกับ L_ACT รวมถึงจัดให้มีการโฆษณา L_ACT จะให้ประสิทธิภาพการค้นหาบริการสูงสุด กล่าวคือระบบ A4 จะมีประสิทธิภาพการจับคู่สูงสุดก็ต่อเมื่อเอเจนท์แต่ละตัวที่ให้บริการจับคู่นั้นใช้แคช (ใช้ C_ACT) และรู้จักความสามารถของเอเจนท์ที่อยู่ในระดับที่ต่ำกว่า (ใช้ L_ACT และโฆษณา L_ACT) มิใช่รู้จักเฉพาะเอเจนท์ที่ลงทะเบียนอยู่กับตนเองเท่านั้น หากแต่รู้จักเอเจนท์ที่อยู่ในระดับที่ต่ำกว่าทั้งหมด การรู้จักดังกล่าวเกิดจากการที่เอเจนท์ที่อยู่ในระดับที่ต่ำกว่าโฆษณาความสามารถขึ้นไปยังเอเจนท์ที่อยู่ในระดับที่สูงกว่านั่นเอง

หากพิจารณาระบบเอเจนท์แบบ Lattice แล้ว สำหรับเอเจนท์จำนวน 251 ตัว จำนวนการติดต่อระหว่างเอเจนท์ต่อคำร้องที่ต้องใช้มีค่าเท่ากับ $\frac{3\sqrt{2}}{4\sqrt{\pi}}\sqrt{251}$ หรือเท่ากับ 9.5 ครั้ง ดังนั้นสำหรับ 12,576 คำร้อง จะต้องใช้จำนวนการติดต่อระหว่างเอเจนท์ทั้งหมดประมาณ 120,000 ครั้ง ซึ่งมากกว่ากรณีที่แย่ที่สุดของระบบ A4 เสียอีก (65,595 ครั้ง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4.5 สรุปงานวิจัยที่ผ่านมาเกี่ยวกับแมทช์เมคกิ้งเอเจนท์

ระบบ Expert Finder และ Middle Agent นั้นมีปัญหาที่สำคัญ ได้แก่ ปัญหาการขาดความสามารถในการขยายขนาด กล่าวคือ ไม่สามารถนำไปประยุกต์ใช้กับระบบเอเจนท์ขนาดใหญ่ได้อย่างมีประสิทธิภาพ หากแต่เหมาะสมที่จะนำไปใช้กับระบบขนาดเล็กมากกว่า สำหรับระบบ Expert Finder นั้นหากนำมาใช้กับระบบขนาดใหญ่จะทำให้มีปัญหาที่สำคัญเกิดขึ้น 2 ประการได้แก่

- ฐานข้อมูล/ฐานความรู้ที่ใช้เก็บรายชื่อ และ/หรือวิธีติดต่อเอเจนท์ของเอเจนท์ Expert Finder แต่ละตัวนั้นจะมีขนาดใหญ่มาก ส่งผลให้การค้นหาเพื่อการจับคู่ระหว่างโฆษณาและคำร้องขอนั้นช้ามาก
- เนื่องจากระบบเอเจนท์นั้นมีการเปลี่ยนแปลงเกิดขึ้นตลอดเวลา (Dynamic) กล่าวคือ มีเอเจนท์เข้าและ/หรือออกจากระบบอยู่ตลอด ทำให้ต้องมีการปรับปรุงข้อมูล/ความรู้เกี่ยวกับชื่อและ/หรือที่อยู่ของเอเจนท์อยู่ตลอดเวลา และการปรับปรุงนี้เกิดขึ้นที่เอเจนท์ Expert Finder ทุกตัว ทำให้สิ้นเปลืองทรัพยากรมาก ทั้งที่ตัว Expert Finder แต่ละตัวเอง และแบนด์วิดท์ของระบบเครือข่ายที่ต้องสูญเสียไปเพื่อการนี้

สำหรับระบบ Middle Agent นั้น หากนำมาใช้กับระบบขนาดใหญ่จะมีปัญหาต่างๆ เกิดขึ้น อันเป็นผลมาจากการเป็นระบบการจับคู่แบบรวมศูนย์ ซึ่งก่อให้เกิดข้อด้อยต่างๆ โดยเฉพาะอย่างยิ่งเมื่อนำไปใช้ในอินเทอร์เน็ต (Kurose and Ross. 2000) ดังนี้

- มีปริมาณการจราจรของข้อมูลที่มาติดต่อกับ Middle Agent คับคั่ง
- การดูแลรักษาเป็นไปได้ยาก เนื่องจากต้องจัดการกับข้อมูลมหาศาล
- Middle Agent อยู่ห่างไกลจากผู้ใช้

นอกจากนี้ยังมีระบบแมทช์เมคกิ้งเอเจนท์ที่มีลักษณะคล้ายคลึงกับแบบ Middle Agent ได้แก่ SHADE Matchmaker และ COIN Matchmaker (Kuokka and Harada. 1998) ซึ่งก็พบปัญหาข้างต้นเช่นเดียวกัน

ดังนั้น ระบบแมทช์เมคกิ้งเอเจนท์ที่มีอยู่ในปัจจุบันจึงขาดความเหมาะสมที่จะนำไปใช้กับระบบเอเจนท์ขนาดใหญ่ อย่างไรก็ตาม ได้มีความพยายามที่จะประยุกต์ใช้เอเจนท์บนระบบเครือข่ายอินเทอร์เน็ตกันอย่างมาก ซึ่งระบบดังกล่าวจัดเป็นระบบเอเจนท์ที่มีขนาดใหญ่ที่สุดเท่าที่เคยมีมา ดังนั้น ระบบแมทช์เมคกิ้งเอเจนท์ที่จะนำมาใช้กับระบบเครือข่ายอินเทอร์เน็ตจะต้องเป็นระบบที่มีความสามารถในการขยายขนาดที่เหมาะสม โดยกลไกในการค้นหาเอเจนท์ที่นำเสนอของระบบ Lattice Agents และ A4 System นั้นเป็นแนวความคิดหนึ่งที่น่าสนใจ อย่างไรก็ตาม การที่จะประยุกต์กลไกดังกล่าวให้สำเร็จได้นั้นจำเป็นอย่างยิ่งที่จะต้องบังคับให้เอเจนท์ทุกตัวที่อยู่ในระบบมีกลไกดังกล่าวเป็นฟังก์ชันหนึ่งของตัวเอเจนท์นั้น ซึ่งเป็นไปได้ยากยิ่ง ดังนั้นจึงควรมีการพัฒนาโครงแบบที่สามารถช่วยให้นำแมทช์เมคกิ้งเอเจนท์ไปใช้กับระบบขนาดใหญ่ เป็นเอเจนท์ที่อุทิศตนเองให้กับงานจับคู่โดยเฉพาะ และสามารถนำมาประยุกต์ใช้ได้จริงต่อไป

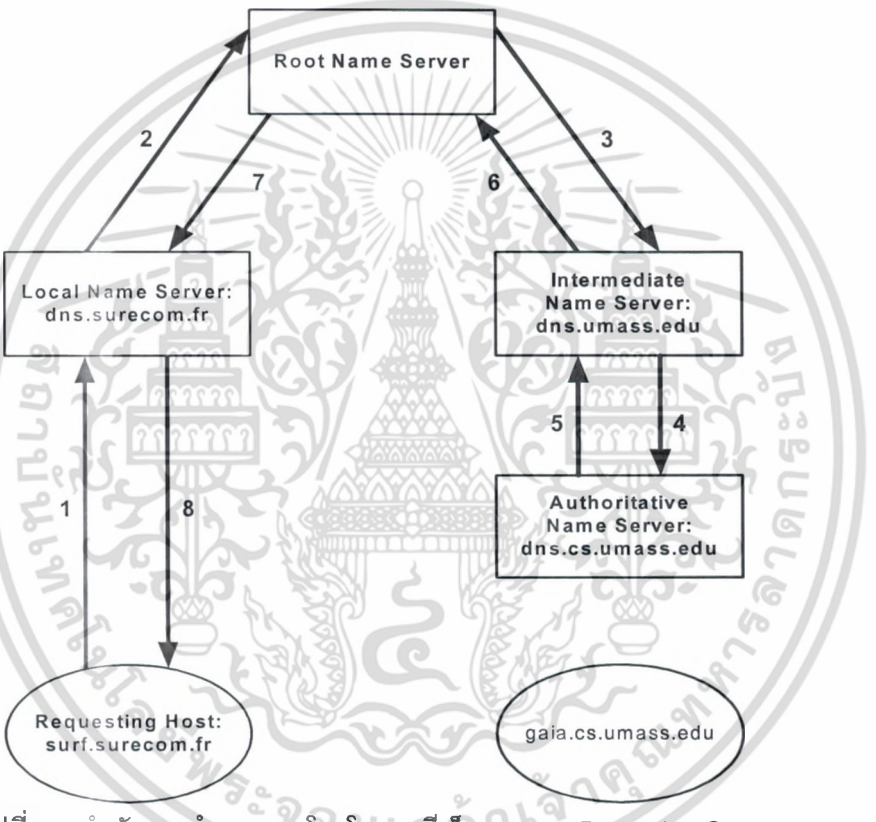
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

2.5 เทคโนโลยีที่ใช้ในการปรับปรุงแมทช์เมคกิ้งเอเจนท์

2.5.1 โพรโทคอลดีเอ็นเอส

โพรโทคอลดีเอ็นเอส (Domain Name System - DNS) (Mockapetris. 1987) เป็นโพรโทคอลที่มีหน้าที่หลักในการค้นหาหมายเลขไอพีของเว็บไซต์ในอินเทอร์เน็ต ประกอบไปด้วย Name Server (NS) จำนวนมากมาทำหน้าที่ร่วมกันในการเก็บ ดูแลรักษา และค้นหา Resource Record (RR) ซึ่งได้แก่ข้อมูลที่บอกหมายเลขไอพีของเว็บไซต์ใดๆ นั่นเอง การทำงานโดยทั่วไปของดีเอ็นเอสเป็นดังรูปที่ 2.9 (Kurose and Ross. 2000)



รูปที่ 2.9 ลำดับการทำงานของโพรโทคอลดีเอ็นเอสแบบ Recursive Query

การทำงานของ NS จะเป็นแบบลำดับชั้น ซึ่งสามารถแบ่งเป็น 2 ประเภทได้แก่ Recursive Query และ Iterative Query ตัวอย่างเช่น การทำงานในรูปที่ 2.9 ซึ่งเป็นแบบ Recursive Query นั้น NS จะทำการตรวจสอบ RR ของตนเองก่อนถ้าไม่พบก็จะส่งต่อคำถามไปเรื่อยๆ เป็นทอดๆ ตามลำดับชั้นของ NS โดยที่ NS แต่ละตัวจะได้รับคำตอบจาก NS ตัวที่ตนเองส่งคำถามไป เมื่อ NS ผู้สร้างคำถามตัวจริงได้รับคำตอบ ก็จะส่งคำตอบ (หมายเลขไอพี) ไปยังเครื่องของผู้ใช้ที่ร้องขอบริการดีเอ็นเอส

จากที่ได้อธิบายข้างต้นพบว่า โดยปกติแล้ว NS แต่ละตัวจะทำหน้าที่เป็นผู้ให้บริการ แต่ก็พร้อมจะเปลี่ยนฐานะไปเป็นผู้ร้องขอบริการหากตนเองไม่สามารถให้บริการดังกล่าวได้ คุณสมบัตินี้ช่วยให้โพร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โทคอลดีเอ็นเอสสามารถนำไปใช้ในระบบเครือข่ายขนาดใหญ่อย่างอินเทอร์เน็ตได้ เนื่องจากการทำงานลักษณะนี้มีรูปแบบที่ง่ายและสามารถนำไปปฏิบัติได้จริง

นอกจากนี้ โพรโทคอลนี้ยังอนุญาตให้มีการ Caching ข้อมูล (RR) ที่ได้รับจากคำตอบไว้เป็นช่วงระยะเวลาหนึ่งด้วย ซึ่งช่วงระยะเวลาดังกล่าวสามารถกำหนดได้จากค่า TTL (Time-to-leave) ใน RR ทั้งนี้ เพื่อช่วยให้ประหยัดแบนด์วิดท์ที่ใช้ และลดเวลาและปริมาณงานในการค้นหาคำตอบลงได้

ในการจำลองการทำงานของดีเอ็นเอสเพื่อประยุกต์ใช้กับแมทช์เมคกิ้งเอเจนท์นั้น จะคำนึงถึงคุณสมบัติที่สำคัญของโพรโทคอลดีเอ็นเอส 2 ส่วนได้แก่ การทำงานของ NS เป็นลำดับขั้นในการค้นหา RR และการแบ่งหน้าที่รับผิดชอบของ NS แต่ละตัว ในระบบดีเอ็นเอสจะมี NS ที่เป็น Authoritative NS อยู่ NS เหล่านี้จะได้แก่ NS ใดก็ได้ที่มีความสามารถให้บริการแก่คำถามนั้นๆ หาก Authoritative NS ตอบกลับมาว่าไม่สามารถตอบคำถามนั้นๆ ได้ก็เป็นอันสิ้นสุด ไม่ต้องส่งคำถามต่อไปยัง NS อื่น อย่างไรก็ตามระบบดีเอ็นเอสนี้อาศัย Sub-domain ในการแบ่งว่า NS ใดเป็น Authoritative NS ของคำถามใดบ้าง ตัวอย่างเช่น ในรูปที่ 2.14 NS: dns.cs.umass.edu จะเป็น Authoritative NS ของคำถามที่ต้องการค้นหาหมายเลขไอพีของเครื่อง gaia.cs.umass.edu เป็นต้น แต่หากนำจุดนี้ไปใช้กับแมทช์เมคกิ้งเอเจนท์จะต้องหาให้ได้ก่อนว่าจะใช้สิ่งใดหรืออะไรมาจำกัดความรับผิดชอบของแต่ละ Authoritative MMA เพื่อให้การส่งต่อคำร้องมายัง Authoritative MMA นั้นกระทำเพียงครั้งเดียว มีจุดสิ้นสุดของกระบวนการค้นหาที่แน่นอน ดังจะได้กล่าวในบทที่ 5 ต่อไป

2.5.2 WordNet

WordNet เป็นฐานข้อมูลเกี่ยวกับกลุ่มคำศัพท์และความสัมพันธ์ระหว่างกลุ่มคำศัพท์ในรูปของภาษาอังกฤษที่มีชื่อเสียงที่สุดอันหนึ่ง ในเวอร์ชั่น 1.6 ที่ใช้ในงานวิจัยนี้ (CSL. 2002) จะประกอบไปด้วยกลุ่มคำศัพท์มากกว่า 50,000 กลุ่มและความสัมพันธ์ระหว่างกลุ่มคำศัพท์มากกว่า 200,000 หน่วย

คำศัพท์ภายในฐานข้อมูล WordNet นี้จะถูกจัดอยู่ในกลุ่มคำศัพท์ที่เรียกว่า Synset ซึ่งได้แก่ กลุ่มของคำที่มีความหมายเหมือนกันหรือ Synonym นั่นเอง กลุ่มของคำที่อยู่ใน Synset เดียวกันจะใช้แนวคิด (Concept) ร่วมกัน ตัวอย่างเช่น Synset หมายเลข n02383458 จะแสดงถึงแนวคิดของยานพาหนะ 4 ล้อที่ติดตั้งเครื่องยนต์ประเภทสันดาปภายใน ซึ่งประกอบไปด้วยสมาชิกได้แก่ {car, auto, automobile, machine, motorcar} เป็นต้น

ความสัมพันธ์ที่มีใน WordNet นั้นมิใช่ความสัมพันธ์ระหว่างคำใดคำหนึ่ง แต่จะเป็นความสัมพันธ์ระหว่าง Synset ความสัมพันธ์ที่สำคัญๆ ได้แก่ hypernym, hyponym, holonym และ meronym

ความสัมพันธ์ hypernym ของ Synset หนึ่งๆ จะหมายถึง Synset ที่มีลักษณะทั่วไปมากกว่า Synset นั้นๆ Synset X จะเป็น hypernym ของ Synset Y ก็ต่อเมื่อ Y เป็นชนิดหนึ่งของ X ตัวอย่างของความสัมพันธ์แบบนี้ได้แสดงไว้ในตารางที่ 2.3 เช่น Synset n09396070 ซึ่งแทนแนวคิดของต้นไม้นั้นมีความทั่วไปมากกว่า Synset n08572627 ซึ่งแทนแนวคิดของต้นโอ๊ค ดังนั้น Synset n09396070 จึงเป็น hypernym ของ Synset n08572627 เป็นต้น

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 ตัวอย่างความสัมพันธ์แบบ Hypernym

Synset	Hypernym	Explanation
n08572627 (oak, oak tree)	n09396070 (tree)	an oak is a kind of tree
n09396070 (tree)	n09395329 (woody plant)	a tree is a kind of woody plant
n02837386 (house)	n02618194 (dwelling, home, habitation, ...)	a house is a kind of dwelling
n05605549 (disappointment, letdown)	n05604874 (dissatisfaction)	disappointment is a kind of dissatisfaction
v01308960 (amble, mosey)	v01299603 (walk)	To amble is to walk in some manner
v00664682 (lisp)	v00662608 (say, articulate, pronounce, ...)	To lisp is to say in some manner
v00846778 (slam, bang)	v00960484 (hit)	To slam is to hit in some manner

สำหรับความสัมพันธ์แบบที่ 2 ความสัมพันธ์ hyponym ของ Synset หนึ่งๆ จะหมายถึง Synset ที่มีความเฉพาะเจาะจงมากกว่า Synset นั้น Synset X จะเป็น hyponym ของ Y ก็ต่อเมื่อ X เป็นชนิดหนึ่งของ Y หรืออาจกล่าวได้ว่า hyponym คือความสัมพันธ์ที่กลับกันของ hypernym นั่นเอง ตัวอย่างของความสัมพันธ์ประเภทนี้ได้จากนำตารางที่ 2.3 มาเปลี่ยนสดมภ์ Synset เป็น Hyponym และเปลี่ยนสดมภ์ Hypernym เป็น Synset แทนนั่นเอง

ความสัมพันธ์ต่อมาได้แก่ความสัมพันธ์ holonym ความสัมพันธ์ holonym ของ Synset ใดๆ จะหมายถึง Synset ที่ได้เป็นองค์ประกอบหลักของ Synset นั้น หรืออาจกล่าวได้ว่า Synset X จะเป็น holonym ของ Y ก็ต่อเมื่อ Y เป็นส่วนหนึ่งของ X นั่นเอง ตัวอย่างของความสัมพันธ์แบบนี้ได้แสดงไว้ในตารางที่ 2.4 เช่น Synset n01660115 ซึ่งแทนแนวคิดของปีกนั้นเป็นส่วนหนึ่งของ Synset n01026695 ซึ่งแทนแนวคิดของนก ดังนั้น Synset n01026695 จึงเป็น holonym ของ Synset n01660115 เป็นต้น

ส่วนความสัมพันธ์แบบสุดท้ายได้แก่ความสัมพันธ์ meronym ความสัมพันธ์แบบนี้ก็คือความสัมพันธ์ที่ตรงกันข้ามกับความสัมพันธ์แบบ holonym นั่นเอง เช่น Synset n01026695 ซึ่งแทนแนวคิดของนกนั้นจะมี meronym คือ Synset n01660115 ซึ่งแทนแนวคิดของปีก เป็นต้น

ทั้งนี้ การที่จะสามารถใช้การจับคู่แบบ Shared-concept Matching ได้นั้น จะต้องอาศัยฐานข้อมูลเกี่ยวกับคำศัพท์มาประยุกต์ใช้งานด้วย ซึ่งในงานวิจัยนี้ได้เลือกใช้ WordNet 1.6 มาใช้เพื่อนำความสัมพันธ์เชิงความหมาย (Semantic Relation) ระหว่างแต่ Synset ที่มีอยู่ในฐานข้อมูลซึ่งได้แก่ Hypernym, Hyponym, Holonym และ Meronym มาใช้ในการจับคู่แบบนี้ด้วย ความสัมพันธ์ทั้งสี่รวมถึงความสัมพันธ์แบบ Synset จะถูกแยกออกมาจาก WordNet เพื่อนำมาใส่ในฐานความรู้ของแมทช์เมคกิ้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอเจนท์อีกทีหนึ่ง ฐานความรู้ประเภทนี้จะเรียกว่า Taxonomic Knowledge Base (TKB) ฐานความรู้ประเภทนี้จัดเป็นส่วนย่อยของฐานความรู้ประเภท Ontological Knowledge Base (OKB) อีกทีหนึ่ง จากรูปที่ 2.10 เราจะแยกเฉพาะเรคคอร์ดที่มีฟิลด์ PTR เป็น @ (hypernym), ~ (hyponym), # (holonym) และ % (meronym) ออกมา แล้วนำไปใส่ในฐานความรู้ของแมทช์เมคกิ้งเอเจนท์ที่สร้างขึ้นจาก Berkeley DB (Sleepycat Software. 2001) อีกทีหนึ่ง

ตารางที่ 2.4 ตัวอย่างความสัมพันธ์แบบ Holonym

Synset	Holonym	Explanation
n01660115 (wing)	n01026695 (bird)	object-component
n04115147 (mouth)	n04340595 (face, human face)	object-component
n09396070 (tree)	n06241974 (forest, wood)	collection-member
n10495237 (oxygen, O, ...)	n10646492 (air)	object-substance
n10871270 (adolescence)	n10871133 (youth)	process-phase

Pointers : ตาราง

PTR	SYNSET_ID	POS	S_WORD_NR	TARGET_ID	T_POS	T_WORD_NR
=	n04047716	n	0	a00001740	a	0
=	n04349777	n	0	a00001740	a	0
!	a00002062	a	1	a00001740	a	1
=	n04047716	n	0	a00002062	a	0
!	a00001740	a	1	a00002062	a	1
!	a00002493	a	1	a00002287	a	1
!	a00002287	a	1	a00002493	a	1
!	a00002856	a	1	a00002687	a	1
!	a00002687	a	1	a00002856	a	1
&	a00003314	a	0	a00003057	a	0
&	a00003469	a	0	a00003057	a	0
&	a00003588	a	0	a00003057	a	0
!	a00003694	a	1	a00003057	a	1
&	a00003057	a	0	a00003314	a	0
&	a00003057	a	0	a00003469	a	0

ระเบียน: 1 จาก 238495

รูปที่ 2.10 ฐานข้อมูลดั้งเดิมของ WordNet 1.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 Sentence Similarity

การนำ TKB มาใช้ จะทำให้ทราบว่าคำต่างๆ ระหว่างแต่ละประโยค (ระหว่างคำร้องขอและโฆษณา) มีความสัมพันธ์กัน แต่มิได้บอกว่ามีความสัมพันธ์กันมากน้อยเพียงใด ดังนั้นจึงต้องมีวิธีการอื่นช่วยในการบอกถึงระดับของความสัมพันธ์ดังกล่าว

Resnik (1995) ได้เสนอวิธีการคำนวณ Semantic Similarity ระหว่างแนวคิดโดยอาศัย WordNet เอาไว้ดังสมการที่ 2.3

$$sim(c1, c2) = \max_{c \in S(c1,c2)} [- \log p(c)] \quad (2.3)$$

กล่าวคือ ความเหมือนระหว่างคำ 2 คำจะได้มาจากการหาคำที่เป็นแนวคิดร่วมกันของคำทั้งสองและเป็นคำที่มี Informativeness หรือ Information Content สูงสุด ความเหมือนระหว่างคำทั้งสองจะมีค่าเท่ากับ Informativeness ของคำร่วมกันดังกล่าวนั่นเอง โดยที่ Informativeness ของแนวคิดหนึ่งๆ นั้นจะหาได้จาก Negative log likelihood ของความน่าจะเป็นที่จะพบ Instance ของแนวคิดนั้นๆ ทั้งนี้ คำที่มีแนวคิดเดียวกันจะมีค่า Informativeness เท่ากันอีกด้วย

อย่างไรก็ตาม วิธีนี้มีข้อเสียที่สำคัญคือ ไม่ว่าคำทั้งสองจะอยู่ห่างจากแนวคิดร่วมกันมากน้อยเพียงใดก็จะยังคงมี Similarity เท่าเดิม ดังนั้นจึงได้มีผู้คิดวิธีแก้ไขขึ้น Lin (1998) ได้นำเสนอการคำนวณหา Semantic Similarity ระหว่างแนวคิดแบบใหม่ดังแสดงในสมการที่ 2.4

$$sim(c1, c2) = \frac{2 \times \max_{c \in S(c1,c2)} [\log p(c)]}{\log p(c1) + \log p(c2)} \quad (2.4)$$

Lin ได้นำ Informativeness ของคำทั้งสองที่ต้องการเปรียบเทียบเข้ามาใช้ในการคำนวณหา Similarity ด้วย ยิ่งคำทั้งสองมี Informativeness สูงมากเท่าใด ก็ยิ่งทำให้คำทั้งสองมีความหมายเหมือนกันน้อยลงเท่านั้น

Girardi and Ibrahim (1994) ได้นำเสนอไว้ว่าการวัด Similarity ระหว่างประโยค 2 ประโยคจะสามารถคำนวณได้จากสมการดังต่อไปนี้

$$sim(s1, s2) = \sum_{\forall j \in SC} w_j \times sim(c1_j, c2_j) \quad (2.5)$$

เมื่อ SC คือเซ็ตของคำทั้งหมดภายในแต่ละประโยค

ความเหมือนระหว่างประโยค 2 ประโยคจะคำนวณได้โดยการถ่วงน้ำหนักให้กับความเหมือนระหว่างแต่ละคำในประโยคนั้น สำหรับความเหมือนของประโยคประเภท PCS นั้น จะคำนวณได้จากการถ่วงน้ำหนักความเหมือนระหว่างแนวคิดในแต่ละระดับ ด้วยน้ำหนักที่มีค่าเท่ากับส่วนกลับของระดับนั้นๆ นั่นเอง ดังนั้นสมการที่ 2.5 จึงสามารถแปลงได้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$sim(pcs1, pcs2) = \sum_{\forall k \in LV_{PCS}} \frac{1}{k} \times \overline{sim}\,(c1_k , c2_k) \qquad (2.6)$$

เมื่อ k คือระดับของคำภายใน PCS ตัวอย่างเช่นประโยค (friend john mary) จะมีคำระดับที่ 1 ได้แก่ friend และคำระดับที่ 2 ได้แก่ john และ mary เป็นต้น

เมื่อเราสามารถคำนวณความเหมือนของแต่ละประโยคได้แล้ว เราก็สามารถคำนวณความเหมือนระหว่างคำร้องขอรับบริการและโฆษณาความสามารถได้ด้วยวิธีดังกล่าว

2.6 สรุป

บทนี้ได้นำเสนอแนวคิดและทฤษฎีที่เกี่ยวข้องกับเทคโนโลยีแมทช์เมคกิ้งเอเจนท์ โดยเริ่มจากการแนะนำเทคโนโลยีเอเจนท์ต่างๆ ก่อน เช่น PCS และ KQML เป็นต้น หลังจากนั้นจึงได้นำงานวิจัยที่สำคัญในอดีตมาวิเคราะห์เพื่อค้นหาแนวทางในการปรับปรุงเทคโนโลยีดังกล่าว แล้วจึงนำเสนอแนวคิดเบื้องต้นที่ใช้ปรับปรุงระบบ ได้แก่ การเลียนแบบการทำงานร่วมกันของระบบดีเอ็นเอส และการใช้ WordNet เพื่อเพิ่มความสามารถในการจับคู่ให้แก่ระบบแมทช์เมคกิ้งเอเจนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ขั้นตอนการศึกษา

การนำเสนอโครงสร้างแมทช์เมคกิ้งเอเจนท์ในงานวิจัยนี้จะแบ่งขั้นตอนการศึกษาออกเป็น 4 ขั้นตอนหลัก ได้แก่ การกำหนดขอบเขตของปัญหาและแนวทางการแก้ไข การออกแบบโครงแบบระบบแมทช์เมคกิ้งเอเจนท์ การสร้างระบบโดยการเขียนโปรแกรม และการทดสอบระบบ แต่ละขั้นตอนจะมีรายละเอียดเบื้องต้นดังหัวข้อที่ 3.1 – 3.4 ตามลำดับ ทั้งนี้ในบทที่ 4-6 จะนำเสนอแต่ละขั้นตอนโดยละเอียดอีกทีหนึ่ง

3.1 การกำหนดปัญหาและแนวทางการแก้ไข

ขั้นแรกของการศึกษาคือ การกำหนดปัญหาที่เกิดขึ้นกับระบบแมทช์เมคกิ้งเอเจนท์ที่มีอยู่แล้วในปัจจุบัน ดังนั้นการดำเนินการของขั้นตอนนี้โดยส่วนใหญ่จึงเป็นการนำผลงานวิจัยเกี่ยวกับแมทช์เมคกิ้งเอเจนท์ในอดีตมาวิเคราะห์เพื่อค้นหาปัญหาหลักๆ ของแมทช์เมคกิ้งเอเจนท์ที่มีอยู่ และกำหนดขอบเขตลงไปว่างานวิจัยนี้จะมุ่งเน้นแก้ไขปัญหาด้านใดของเทคโนโลยีแมทช์เมคกิ้งเอเจนท์

จากนั้น จึงค้นหาเทคนิคต่างๆ ที่มีอยู่ในปัจจุบันและมีความเป็นไปได้ที่จะนำมาประยุกต์ใช้กับเทคโนโลยีแมทช์เมคกิ้งเอเจนท์ได้ รวมถึงศึกษาผลงานวิจัยในอดีตเพื่อค้นหาจุดอ่อนที่สำคัญของแมทช์เมคกิ้งเอเจนท์ที่มีอยู่ รวมถึงค้นหาแนวคิดหลักๆ ในการแก้ไขปัญหาเกี่ยวกับแมทช์เมคกิ้งเอเจนท์ที่ได้มีผู้นำเสนอมาก่อนหน้าแล้ว

3.2 การออกแบบระบบ

หลังจากกำหนดแนวทางแก้ไขปัญหาแล้ว ก็พยายามออกแบบระบบแมทช์เมคกิ้งเอเจนท์ตามแนวทางที่วางไว้ สิ่งที่ต้องออกแบบจะได้แก่ โครงสร้างของระบบแมทช์เมคกิ้งเอเจนท์ การทำงานร่วมกันของแมทช์เมคกิ้งเอเจนท์ภายในระบบ การแสดงความต้องการและความสามารถของเอเจนท์ และกลไกการจับคู่ ตามลำดับ

การออกแบบระบบนี้มีวัตถุประสงค์เพื่อให้ได้ระบบที่สามารถแก้ไขปัญหาได้ รวมถึงสามารถนำงานออกแบบไปพัฒนาเป็นระบบแมทช์เมคกิ้งเอเจนท์ขึ้นมาได้ เพื่อใช้ทดสอบว่าโครงแบบที่นำเสนอนั้นมีประสิทธิภาพมากน้อยเพียงใดต่อไป

3.3 การสร้างระบบ

เพื่อทดสอบว่าสถาปัตยกรรม DMAA ที่นำเสนอในงานวิจัยนี้มีประสิทธิภาพดีพอที่จะสามารถนำไปประยุกต์ใช้กับระบบเอเจนท์ขนาดใหญ่ได้ ระบบแมทช์เมคกิ้งเอเจนท์จึงได้ถูกสร้างขึ้นมาตามแนวคิดของสถาปัตยกรรม DMAA ข้างต้น โดยภาษาที่ใช้ในการพัฒนาได้แก่ ภาษาจาวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การทดสอบระบบ

ขั้นตอนนี้เป็นการทดลองเพื่อหาว่าระบบ DMAA นั้นสามารถนำไปใช้กับระบบเอเจนท์ขนาดใหญ่ได้หรือไม่ กล่าวคือเป็นการทดสอบคุณสมบัติ Scalability นั่นเอง การทดลองจะทำการเปรียบเทียบระหว่างระบบแมทช์เมคกิ้งเอเจนท์แบบดั้งเดิม (ได้แก่ Middle Agent) ซึ่งมีการทำงานเป็นแบบรวมศูนย์และการทำงานของระบบ DMAA ซึ่งเป็นแบบกระจาย รวมถึงเปรียบเทียบกับระบบแมทช์เมคกิ้งเอเจนท์แบบ A4 System ซึ่งเป็นระบบล่าสุดที่พยายามแก้ไขปัญหาประสิทธิภาพของเทคโนโลยีแมทช์เมคกิ้งเอเจนท์อีกด้วย โดยจะเน้นเปรียบเทียบให้เห็นถึงประสิทธิภาพของระบบแต่ละประเภท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

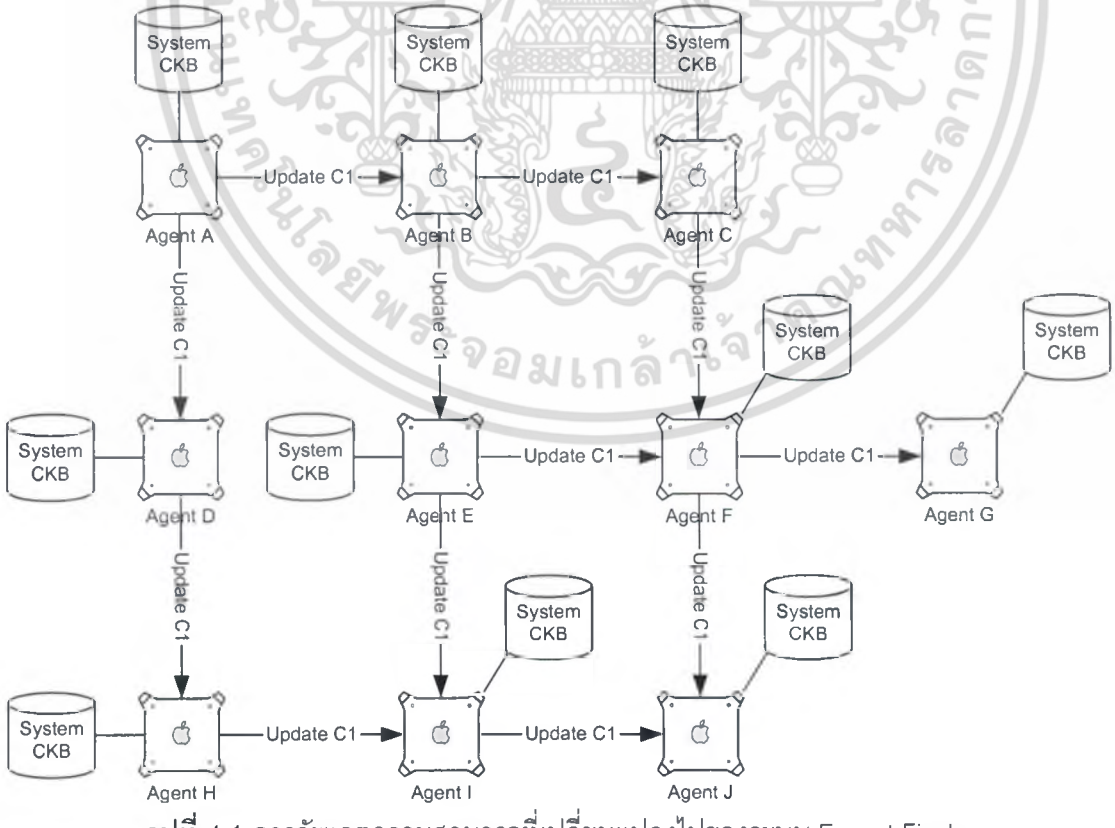
บทที่ 4
ปัญหาและแนวทางแก้ไข

ขั้นตอนแรกของการศึกษาได้แก่การกำหนดปัญหาของเทคโนโลยีแมทช์เมคกิ้งเอเจนท์ที่มีอยู่ในปัจจุบัน และนำเสนอแนวทางการแก้ไขปัญหาดังกล่าว ดังนี้

4.1 การกำหนดปัญหา

ในปัจจุบันได้มีผู้พัฒนาระบบแมทช์เมคกิ้งเอเจนท์ขึ้นมาบ้างแล้ว แต่ละระบบมีจุดเด่นจุดด้อยต่างกัน ตัวอย่างของระบบดังกล่าวได้แก่ Expert Finder (Vivacqua. 1999), Middle Agent (Sycara *et al.* 1999), A4 System (Cao *et al.* 2000) และ Lattice Agents (Shehory. 2000)

Expert Finder เป็นระบบแมทช์เมคกิ้งเอเจนท์ระดับพื้นฐานที่สุด ในระบบนี้เอเจนท์แต่ละตัวจะต้องมีรายชื่อและที่อยู่ของเอเจนท์อื่นๆ ทั้งหมดของระบบ คุณสมบัตินี้ส่งผลให้เทคโนโลยีแมทช์เมคกิ้งเอเจนท์ประเภทนี้ไม่สามารถนำไปใช้กับระบบที่มีเอเจนท์จำนวนมากได้เนื่องจากมีปัญหาในการอัพเดตข้อมูลความสามารถที่เปลี่ยนแปลงให้แก่ฐานความรู้ความสามารถ (Capability Knowledge Base - CKB) ของเอเจนท์แต่ละตัวดังรูปที่ 4.1

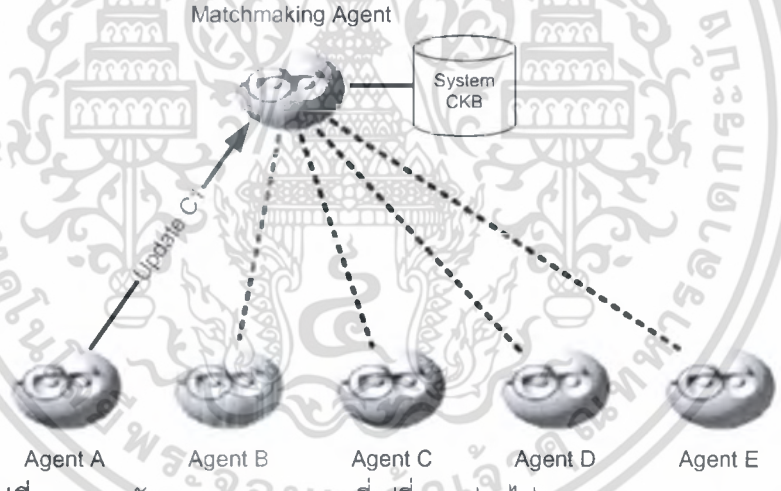


รูปที่ 4.1 การอัพเดตความสามารถที่เปลี่ยนแปลงไปของระบบ Expert Finder

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.1 เอเจนท์ A มีความสามารถ C1 ที่เปลี่ยนแปลงไปจากเดิม ตัวอย่างเช่น แต่เดิมเอเจนท์ดังกล่าวจะมีความสามารถ C1 คือสามารถจัดตารางการท่องเที่ยวประเทศออสเตรเลียได้ ต่อมาเอเจนท์ดังกล่าวได้รับข้อมูลการท่องเที่ยวในประเทศนิวซีแลนด์มามากพอจนมีความสามารถในการจัดตารางการท่องเที่ยวประเทศนิวซีแลนด์ได้ด้วย จึงได้ประกาศการเปลี่ยนแปลงความสามารถ C1 จากที่สามารถจัดตารางท่องเที่ยวได้เฉพาะประเทศออสเตรเลียไปเป็นสามารถจัดตารางท่องเที่ยวได้ทั้งออสเตรเลียและนิวซีแลนด์ เป็นต้น การประกาศการเปลี่ยนแปลงดังกล่าวจะต้องถูกส่งไปยังเอเจนท์ทั้งหมดภายในระบบ ดังนั้นจึงมีความซับซ้อนของจำนวนการติดต่อระหว่างเอเจนท์ในการอัพเดตอย่างน้อยที่สุดเท่ากับ O(n) ต่อการเปลี่ยนแปลงความสามารถหนึ่งอย่าง ดังนั้นในระบบเอเจนท์ขนาดใหญ่จึงเป็นไปไม่ได้เลยที่ระบบเอเจนท์ดังกล่าวจะทำงานได้อย่างมีประสิทธิภาพ เนื่องจากจะมีแต่ข้อมูลการเปลี่ยนแปลงความสามารถของเอเจนท์วิ่งไปมาภายในระบบอยู่เต็มไปหมด

ดังนั้นต่อมาจึงได้มีผู้พัฒนาระบบแมทช์เมคกิ้งเอเจนท์แบบ Middle Agent ขึ้น ในระบบนี้จะมีเอเจนท์เพียงตัวเดียวเพื่อทำหน้าที่เป็นแมทช์เมคกิ้งเอเจนท์โดยเฉพาะซึ่งถูกเรียกว่า Middle Agent นั่นเอง คุณสมบัติดังกล่าวทำให้การอัพเดตความสามารถที่เปลี่ยนแปลงของเอเจนท์หนึ่งๆ มีลักษณะดังรูปที่ 4.2



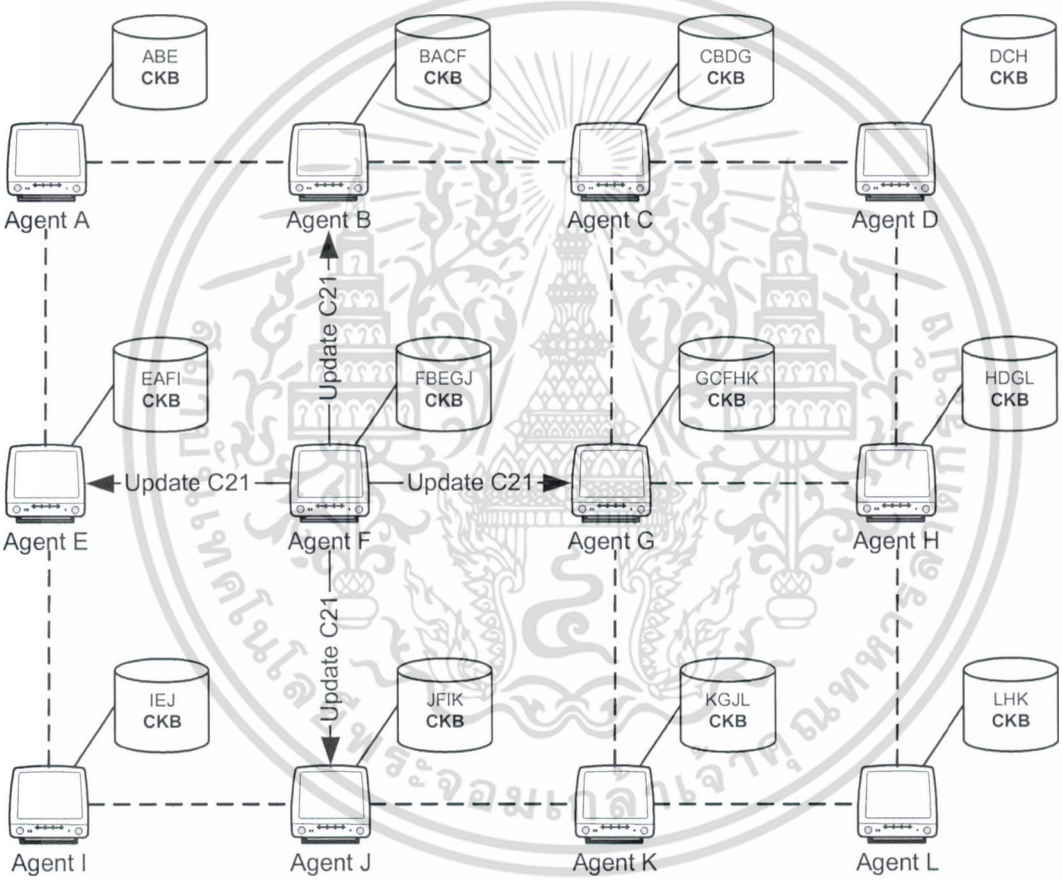
รูปที่ 4.2 การอัพเดตความสามารถที่เปลี่ยนแปลงไปของระบบ Middle Agent

จากรูปที่ 4.2 เอเจนท์ A มีความสามารถ C1 ที่เปลี่ยนแปลงไปจากเดิม ดังนั้นจึงประกาศการเปลี่ยนแปลงดังกล่าวไปยังเอเจนท์ที่มีฐานความรู้ความสามารถของระบบเอเจนท์ทั้งหมดซึ่งก็คือแมทช์เมคกิ้งเอเจนท์หรือ Middle Agent นั่นเอง ระบบแมทช์เมคกิ้งเอเจนท์แบบนี้จะช่วยลดจำนวนการติดต่อระหว่างเอเจนท์เพื่ออัพเดตความสามารถที่เปลี่ยนแปลงลงให้เหลือเพียงหนึ่งครั้งต่อหนึ่งการเปลี่ยนแปลงเท่านั้น

อย่างไรก็ตาม ระบบนี้ได้ก่อให้เกิดปัญหาใหม่ขึ้นมาได้แก่ การให้บริการจับคู่นั้นถูกกระทำโดยเอเจนท์เพียงตัวเดียวจึงทำให้การให้บริการจับคู่นั้นช้ากว่าระบบแมทช์เมคกิ้งเอเจนท์แบบก่อนซึ่งใช้เอเจนท์หลายตัวในการให้บริการดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อมาได้มีความพยายามที่จะแก้ไขปัญหาข้างต้นเกิดขึ้นโดยเริ่มจาก Shehory (2000) ได้นำเสนอกลไกในการค้นหาเอเจนท์โดยที่ไม่จำเป็นต้องมีแมทช์เมคกิ้งเอเจนท์ หลักการของวิธีนี้คือนำหน้าที่ในการค้นหาเอเจนท์ไปไว้ที่ตัวเอเจนท์ทุกๆ ตัวแทนการใช้แมทช์เมคกิ้งเอเจนท์ หลักการนี้มีความคล้ายคลึงกับระบบ Expert Finder อย่างไรก็ตาม ระบบนี้มีความแตกต่างออกไปตรงที่เอเจนท์แต่ละตัวไม่จำเป็นที่จะต้องเก็บข้อมูลเกี่ยวกับเอเจนท์ทั้งหมดที่มีอยู่ในระบบ แต่เอเจนท์แต่ละตัวจะรู้จักเอเจนท์อื่นเป็นจำนวนหนึ่งเท่านั้น เมื่อเอเจนท์ต้องการค้นหาเอเจนท์ใดๆ ก็จะทำการแลกเปลี่ยนข้อมูลซึ่งกันและกันภายในระบบนั้นแทน ดังนั้นการอัพเดตข้อมูลเกี่ยวกับความสามารถที่เปลี่ยนแปลงไปจึงถูกจำกัดอยู่ภายในวงจำกัด ดังรูปที่ 4.3



รูปที่ 4.3 การอัพเดตความสามารถที่เปลี่ยนแปลงไปของระบบ Lattice Agents

จากรูปที่ 4.3 เอเจนท์ F มีความสามารถ C21 ที่เปลี่ยนแปลงไปจากเดิม ดังนั้นจึงประกาศการเปลี่ยนแปลงดังกล่าวไปยังเอเจนท์ที่มีฐานความรู้ความสามารถของตนเองอยู่ซึ่งได้แก่เอเจนท์ B, E, G และ J นั่นเอง ระบบแมทช์เมคกิ้งเอเจนท์แบบนี้จะช่วยลดจำนวนการติดต่อระหว่างเอเจนท์เพื่ออัพเดตความสามารถที่เปลี่ยนแปลงลงให้เหลือเพียงไม่เกิน 4 ครั้งต่อหนึ่งการเปลี่ยนแปลง และยังคงมีความรวดเร็วในการให้บริการจับคู่อยู่เนื่องจากใช้เอเจนท์หลายตัวในการให้บริการดังกล่าวนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

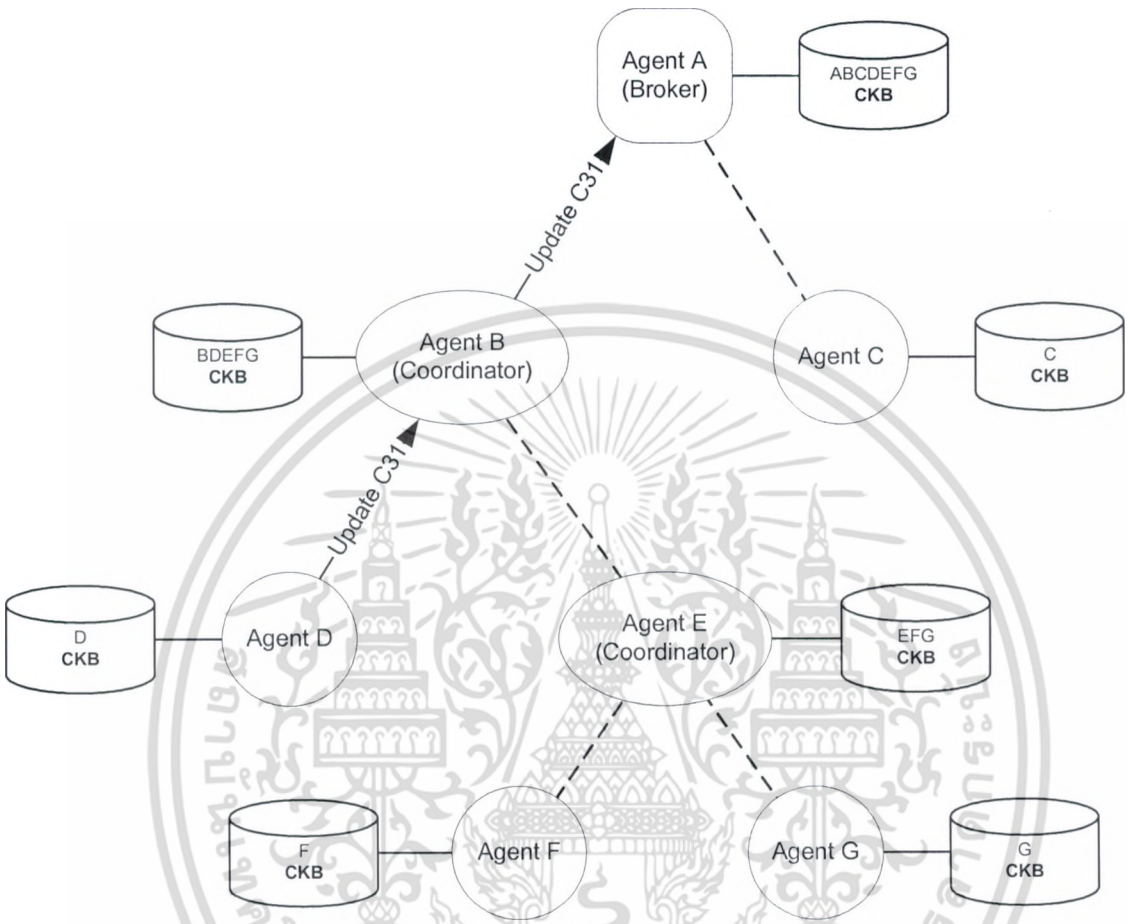
อย่างไรก็ตาม ระบบเอเจนท์แบบนี้ได้สร้างปัญหาใหม่ให้เกิดขึ้น การที่ใช้เอเจนท์หลายตัวให้บริการจับคู่จะช่วยให้ปริมาณงานที่เอเจนท์แต่ละตัวต้องรองรับนั้นมีปริมาณที่น้อยลงจากการที่ใช้แมทช์เมคกิ้งเอเจนท์เพียงตัวเดียว ส่งผลให้การให้บริการจับคู่ที่เอเจนท์หนึ่งๆ ต้องให้บริการนั้นเป็นไปด้วยความรวดเร็ว แต่ในกรณีของระบบ Lattice Agents นี้มีความแตกต่างออกไปจากระบบก่อนๆ สำหรับระบบ Expert Finder นั้นการค้นหาบริการจะกระทำเพียงครั้งเดียวและสำหรับระบบ Middle Agent ก็เช่นกัน แต่สำหรับระบบ Lattice Agents นั้นการค้นหาบริการหนึ่งบริการจะต้องถูกค้นหาหลายครั้งในเอเจนท์หลายตัวตามเส้นทางที่คำร้องขอรับบริการถูกส่งออกไป ตัวอย่างเช่น เมื่อเอเจนท์ A ต้องการรับบริการหนึ่งๆ ที่เอเจนท์ L สามารถให้บริการได้ (แต่ยังไม่ทราบว่าเอเจนท์ L สามารถให้บริการดังกล่าวได้) เมื่อเอเจนท์ A ใช้บริการจับคู่ของตนเองแล้วไม่พบ ก็จะส่งต่อคำร้องเพื่อขอรับบริการดังกล่าวไปยังเอเจนท์ข้างเคียงที่ตนเองรู้จัก ได้แก่ เอเจนท์ B และ E เอเจนท์ทั้งสองก็จะไปค้นหาบริการในฐานความรู้ความสามารถของตนเองซึ่งก็ไม่พบจึงส่งต่อคำร้องดังกล่าวไปยังเอเจนท์ข้างเคียงต่อไปได้แก่ เอเจนท์ C, F และ I กระบวนการนี้จะกระทำซ้ำไปเรื่อยๆ จนกว่าคำร้องจะไปถึงเอเจนท์ H หรือเอเจนท์ K เอเจนท์ดังกล่าวจะสามารถค้นหาบริการที่เอเจนท์ A ต้องการได้พบเนื่องจากมีข้อมูลความสามารถของเอเจนท์ L เก็บไว้ในฐานความรู้ความสามารถของตนเอง และสามารถตอบกลับไปว่าให้เอเจนท์ A ไปติดต่อขอรับบริการกับเอเจนท์ L ต่อไปได้ เป็นต้น จากตัวอย่างจะพบว่าการส่งต่อคำร้องนั้นมีการส่งซ้ำซ้อนกัน เช่น เอเจนท์ B และ E จะส่งต่อไปยังเอเจนท์ F เหมือนกัน ทำให้สิ้นเปลืองทรัพยากรในการติดต่อสื่อสารโดยไม่จำเป็น

ในเวลาต่อมา Cao *et al.* (2000) ได้นำระบบ Lattice Agents มาปรับปรุงและกำจัดความซ้ำซ้อนในการส่งต่อคำร้องออกไปจากระบบโดยอาศัยการจัดโครงสร้างแบบลำดับชั้นให้แก่ระบบเอเจนท์และทำการทดลองค้นหา Discovery Model ที่มีความเหมาะสมกับโครงสร้างดังกล่าว (ดูหัวข้อ 2.5.4.4) จากตารางที่ 2.2 จะสามารถสรุปได้ว่าระบบ A4 System จะมีประสิทธิภาพการจับคู่สูงสุดก็ต่อเมื่อเอเจนท์แต่ละตัวที่ให้บริการจับคู่นั้นใช้แคชและรู้จักความสามารถของเอเจนท์ที่อยู่ในระดับที่ต่ำกว่าทั้งหมด ดังรูปที่ 4.4 การค้นหาโฆษณาในระบบนี้จะกระทำโดยเอเจนท์หนึ่งๆ จะค้นหาจากฐานความรู้ความสามารถของตนเองก่อน ถ้าไม่เจอก็จะส่งคำร้องไปยังเอเจนท์ในระดับที่สูงกว่าอยู่หนึ่งระดับหรือ Coordinator ของเอเจนท์นั้นๆ นั่นเอง หากเอเจนท์ดังกล่าวมีโฆษณาความสามารถที่เข้าคู่กับคำร้องแล้วก็จะตอบกลับไป แต่ถ้าไม่สามารถจับคู่ได้ก็จะส่งต่อคำร้องไปยังเอเจนท์ในระดับที่สูงกว่าตนเองอยู่อีกหนึ่งระดับ และจะเป็นเช่นนี้ไปเรื่อยๆ จนหากคำร้องถูกส่งต่อไปยังเอเจนท์ที่อยู่ในระดับที่สูงสุดแล้วยังไม่สามารถหาความสามารถที่เข้าคู่กับคำร้องดังกล่าวได้ก็แสดงว่าในระบบเอเจนท์นั้นไม่มีเอเจนท์ใดสามารถให้บริการที่ร้องขอมาได้

ระบบนี้สามารถจำกัดจำนวนการติดต่อระหว่างแมทช์เมคกิ้งเอเจนท์เพื่อให้บริการจับคู่หนึ่งๆ ให้อยู่ในวงจำกัดได้ แต่จะมีข้อเสียคือจำนวนการอัพเดตความสามารถที่เปลี่ยนแปลงไปจะสูงขึ้น ซึ่งโดยเฉลี่ยแล้วจะต้องใช้จำนวนการติดต่อระหว่างเอเจนท์มากกว่าทั้งระบบ Lattice Agents และ Middle

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Agent ทั้งนี้เป็นผลมาจากการที่ความสามารถที่เปลี่ยนแปลงไปสำหรับเอเจนท์หนึ่งๆ จะต้องถูกโฆษณาขึ้นไปตามลำดับชั้นเรื่อยๆ จนถึงเอเจนท์ตัวบนสุดนั่นเองดังรูปที่ 4.4



รูปที่ 4.4 การอัพเดตความสามารถที่เปลี่ยนแปลงไปของระบบ A4 System

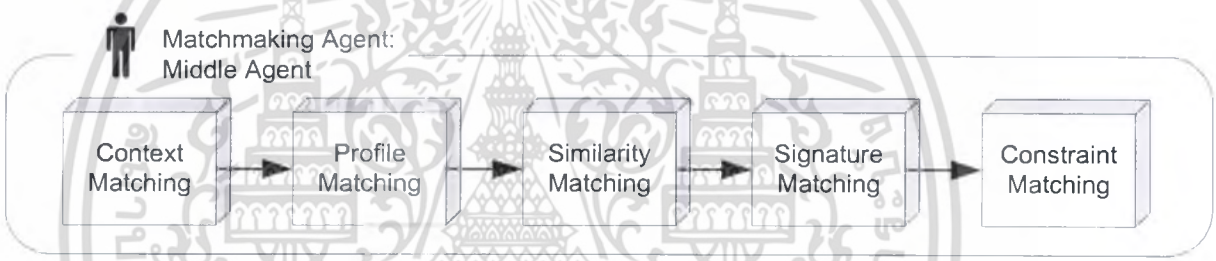
จากการวิเคราะห์งานวิจัยที่กล่าวมาข้างต้นจะพบว่าพัฒนาการของเทคโนโลยีแมทช์เมคกิ้งเอเจนท์นั้นเป็นไปเพื่อในทิศทางเพื่อที่จะเพิ่มประสิทธิภาพให้แก่ระบบแมทช์เมคกิ้งเอเจนท์ขึ้นเรื่อยๆ ตั้งแต่ระบบ Expert Finder (Vivacqua. 1999) ระบบ Middle Agent (Sycara *et al.* 1999) ระบบ A4 System (Cao *et al.* 2000) ไปจนถึงระบบ Lattice Agents (Shehory. 2000) ดังนั้นปัญหาที่แท้จริงของเทคโนโลยีดังกล่าวในปัจจุบันก็คือการให้บริการของแมทช์เมคกิ้งเอเจนท์ยังคงมีประสิทธิภาพที่ต่ำอยู่ ทำให้ไม่สามารถนำไปใช้ในระบบเอเจนท์ได้จริง ดังนั้นงานวิจัยนี้จึงมุ่งเน้นที่จะนำเสนอเทคนิคใหม่ๆ ที่ช่วยเพิ่มประสิทธิภาพให้แก่เทคโนโลยีแมทช์เมคกิ้งเอเจนท์ เพื่อสามารถที่จะนำไปใช้กับระบบเอเจนท์ที่มีขนาดใหญ่ได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 แนวทางแก้ไข

แนวทางการแก้ไขปัญหาข้างต้นมีอยู่หลายทางเลือกด้วยกัน เช่น การเพิ่มประสิทธิภาพของฮาร์ดแวร์ที่ใช้ การลดความสามารถของกลไกการจับคู่ลงเพื่อให้การจับคู่หนึ่งๆ ใช้เวลาน้อยลง การใช้วิธีแสดงความรู้ที่มีความสอดคล้องกับกลไกการจับคู่เพื่อกำจัดขั้นตอนที่ไม่จำเป็น เป็นต้น โดยสรุปแล้วแนวทางการแก้ไขสามารถจำแนกได้เป็น 2 ประเภทหลัก ได้แก่ การแก้ไขทางฮาร์ดแวร์ และการแก้ไขทางซอฟต์แวร์

สำหรับการแก้ไขปัญหาข้างต้นโดยเน้นแก้เฉพาะที่ซอฟต์แวร์นั้นจะหมายถึงการปรับปรุงอัลกอริทึมที่ใช้ในกลไกการจับคู่ให้มีประสิทธิภาพสูงขึ้นเป็นหลัก ซึ่งแมทช์เมคกิ้งเอเจนท์ที่มีอยู่ในปัจจุบันนั้นมักจะมีความสามารถในการจับคู่หลายๆ แบบ ตัวอย่างเช่น Middle Agent ของ Sycara *et al.* (1999) ใช้กลไกการจับคู่แบบ Context Matching, Profile Matching และ Similarity Matching ดังแสดงในรูปที่ 4.5 เป็นต้น

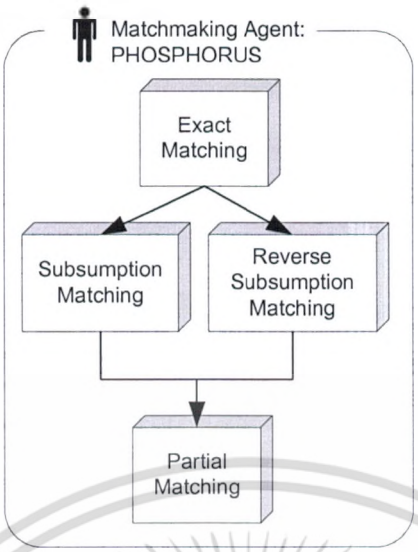


รูปที่ 4.5 กระบวนการจับคู่ของ Middle Agent (Sycara *et al.* 1999)

สำหรับ Phosphorus Matchmaking Agent ของ Gil and Ramachandran (2001) นั้นจะประกอบไปด้วยขั้นตอนการจับคู่เรียงตามลำดับได้แก่ Exact Matching, Subsumption Matching และ Partial Matching ดังรูปที่ 4.6

การที่แมทช์เมคกิ้งเอเจนท์หนึ่งๆ นำวิธีการจับคู่หลากหลายวิธีมาใช้ จะทำให้โดยธรรมชาติแล้วแมทช์เมคกิ้งเอเจนท์จะใช้เวลาค่อนข้างนานสำหรับการจับคู่โฆษณาความสามารถที่เหมาะสมให้กับคำร้องขอรับบริการหนึ่งๆ กล่าวคือ เวลาตอบสนองเฉลี่ยต่อการให้บริการหนึ่งคำร้องจะมีค่ามาก ตัวอย่างเช่น เฉพาะการค้นหาความสามารถของเอเจนท์โดยใช้วิธี Pattern Matching และอัลกอริทึม RETE ไปค้นหาจากฐานข้อมูลโฆษณาขนาด 10,000 เรคคอร์ด ในการค้นหา จะใช้เวลาในการค้นหาประมาณ 2 วินาทีต่อหนึ่งคำร้อง (ทดสอบโดยใช้ JESS เป็นฐานข้อมูลโฆษณา และใช้เครื่อง AMD Athlon XP 1500+ ในการทดสอบ) (Friedman-Hill. 2001) ซึ่งเมื่อผนวกกับวิธีการจับคู่ประเภทอื่นด้วยแล้วทำให้ต้องใช้เวลาค่อนข้างนานในการให้บริการคำร้องหนึ่งๆ ซึ่งอาจส่งผลให้เอเจนท์ผู้ร้องขอล้มเลิกความตั้งใจในการร้องขอบริการไปเสียก่อน ดังนั้นแนวทางการแก้ไขโดยอาศัยการปรับปรุงอัลกอริทึมจึงเป็นไปได้ยากเนื่องจากธรรมชาติของกลไกที่ใช้ในการจับคู่นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 กระบวนการจับคู่ของ Phosphorus (Gil and Ramachandran. 2001)

ส่วนแนวทางการนำแมทช์เมคกิ้งเอเจนท์หลายตัวมาทำงานร่วมกันนั้นจัดเป็นอีกแนวทางหนึ่งที่น่าสนใจ และมีงานวิจัยที่เกี่ยวข้องเกิดขึ้นมาบ้างแล้ว ได้แก่ สถาปัตยกรรมแบบ Lattice (Shehory. 2000) และสถาปัตยกรรม A4 (Cao *et al.* 2001) งานเหล่านี้ได้นำเสนอแนวคิดที่สำคัญคือ การพัฒนาโครงสร้างของกลุ่มเอเจนท์ที่ให้บริการจับคู่จะต้องคำนึงถึงความสอดคล้องกับลักษณะเฉพาะของเอเจนท์ให้มากที่สุด และพยายามที่จะลดหรือจำกัดจำนวนการติดต่อระหว่างเอเจนท์ในการให้บริการคำร้องขอรับบริการหนึ่งๆ โดยมีจุดประสงค์หลักคือเพื่อลดความล่าช้าที่เกิดจากระบบเครือข่ายนั่นเอง แนวคิดทั้งสองนี้จัดเป็นแนวทางเริ่มต้นสำหรับการเพิ่มศักยภาพให้กับระบบแมทช์เมคกิ้งเอเจนท์ที่อาศัยการนำแมทช์เมคกิ้งเอเจนท์หลายตัวมาทำงานร่วมกัน

งานวิจัยนี้มุ่งเน้นที่จะแก้ไขปัญหาประสิทธิภาพของระบบแมทช์เมคกิ้งเอเจนท์โดยอาศัยการพัฒนาโครงสร้างของกลุ่มแมทช์เมคกิ้งเอเจนท์ที่สนับสนุนการทำงานร่วมกันของแมทช์เมคกิ้งเอเจนท์ มีความสอดคล้องกับความเป็นเอเจนท์ ใช้ทรัพยากรเครือข่ายน้อยลง และ ส่งผลให้ประสิทธิภาพโดยรวมของระบบแมทช์เมคกิ้งเอเจนท์สูงขึ้น

การออกแบบโครงสร้างของแมทช์เมคกิ้งเอเจนท์ที่ดีนั้นจะต้องคำนึงถึงลักษณะโดยทั่วไปของเอเจนท์ด้วยได้แก่ ความสามารถทางสังคม และการมีภาวะอิสระ โดยเฉพาะลักษณะอย่างหลังนั้น จะทำให้เอเจนท์สามารถเลือกสามารถตัดสินใจปัญหาต่างๆ ได้ด้วยตนเอง เช่น การเลือกที่จะเข้าหรือออกจากระบบเอเจนท์ด้วยตนเอง และการล้มเลิกความตั้งใจในการทำงานหนึ่งๆ เนื่องจากมีข้อมูลและความสามารถไม่พอเพียง หรือขอความช่วยเหลือไปยังเอเจนท์อื่นแล้ว แต่ต้องรอเป็นเวลานาน เป็นต้น

ความสามารถทางสังคมของเอเจนท์ หมายถึง ความสามารถที่จะค้นหา ติดต่อ และสื่อสารกับเอเจนท์อื่นได้ การสื่อสารระหว่างเอเจนท์นั้นจัดเป็นหัวข้อที่มีความยากทีเดียว เนื่องจากเอเจนท์นั้นมีความหลากหลายมาก โดยเฉพาะอย่างยิ่งในเรื่องของภาษาที่ใช้แสดงความรู้ที่แตกต่างกันสำหรับเอเจนท์แต่ละตัว ข้อเท็จจริงนี้ได้นำไปสู่ปัญหาหลักที่เกิดขึ้นกับการสื่อสารระหว่างเอเจนท์ได้แก่ ปัญหาในการทำความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้าใจข้อความที่เอเจนท์อื่นส่งมา ปัญหานี้มีสาเหตุมาจากการใช้ภาษาแสดงความรู้ที่มีความแตกต่างทางไวยากรณ์และทางความหมาย ดังนั้น จึงได้มีความพยายามที่จะกำหนดภาษาแสดงความรู้มาตรฐานของเอเจนท์เพื่อแก้ไขปัญหาแรกเกิดขึ้นมาในอดีต ได้แก่ ภาษา KIF ภาษานี้เป็นภาษาที่สามารถแปลงไปเป็นภาษาแสดงความรู้แบบอื่นได้ และในทางกลับกันก็สามารถแปลงจากภาษาอื่นมาเป็นภาษานี้ได้เช่นกัน ดังนั้น การที่เอเจนท์ใดๆ จะติดต่อกับเอเจนท์อื่นได้ จะต้องมี KIF Translator ที่ช่วยแปลงไปมาระหว่างภาษาแสดงความรู้ของตนเองและภาษา KIF ด้วย ส่วนการแก้ไขปัญหาการใช้คำนั้นได้มีตัวอย่างเกิดขึ้นมาบ้างแล้ว ได้แก่ WordNet (CSL. 2002) และ CYC (Cycorp. 2002) ซึ่งเป็นฐานความรู้ประเภท Taxonomy และ Ontology ตามลำดับ ฐานความรู้เหล่านี้จะช่วยให้เอเจนท์สามารถเข้าใจความหมายของคำที่เอเจนท์อื่นใช้ได้

นอกจากการสื่อสารแล้ว การค้นหาและการติดต่อก็เป็นอีกคุณสมบัติที่รวมอยู่ในความสามารถทางสังคมของเอเจนท์ แต่เนื่องจากเอเจนท์หนึ่งๆ นั้นไม่สามารถมีความรู้เกี่ยวกับภาพรวมของระบบเอเจนท์ทั้งระบบได้ (Ogston and Vassiliadis. 2001) ดังนั้นการค้นหาและติดต่อเอเจนท์อื่นจึงต้องอาศัยกระบวนการเรียกซ้ำ (Recursive) ตัวอย่างเช่น เอเจนท์ ก. ต้องการติดต่อเอเจนท์ ข. แต่เอเจนท์ ก. ไม่รู้วิธีการติดต่อหรือที่อยู่ของเอเจนท์ ข. ดังนั้นเอเจนท์ ก. จึงส่งข้อความที่บ่งบอกถึงความต้องการในการติดต่อกับเอเจนท์ ข. ไปยังเอเจนท์ข้างเคียงซึ่งได้แก่เอเจนท์ ค. หากเอเจนท์ ค. รู้วิธีการติดต่อกับเอเจนท์ ข. เอเจนท์ ค. ก็จะส่งวิธีการติดต่อกับเอเจนท์ ข. กลับไปยังเอเจนท์ ก. แต่หากไม่ทราบก็จะส่งต่อไปยังเอเจนท์อื่นต่อ และจะเป็นเช่นนี้ไปเรื่อยๆ จนกว่าจะพบ หรือถูกยกเลิกความต้องการนั้นไปเสียก่อน เป็นต้น อย่างไรก็ตาม วิธีการประเภทนี้จัดเป็นวิธีที่สิ้นเปลืองทรัพยากรเครือข่ายมาก (CDR. 1999) ดังนั้นเพื่อบรรเทาปัญหาดังกล่าว เทคโนโลยี Agent Name Server หรือ ANS (ISAG. 2002) จึงได้ถูกพัฒนาขึ้น ในระบบเอเจนท์หนึ่งจะประกอบไปด้วย ANS อย่างน้อย 1 ตัว เพื่อให้บริการด้านการติดต่อสื่อสารระหว่างเอเจนท์ ดังนั้น ANS จึงสามารถจัดได้ว่าเป็นสาธารณูปโภคขั้นพื้นฐานของระบบเอเจนท์ที่ให้บริการความสามารถทางสังคมแก่เอเจนท์ต่างๆ เช่นเดียวกันกับแมทช์เมคกิ้งเอเจนท์ โดยจะให้บริการด้าน Name resolution แก่เอเจนท์ภายในกลุ่มของตนเอง นอกจากนี้ ในระบบเอเจนท์ยังประกอบไปด้วยสาธารณูปโภคอื่นๆ อีกหลายประเภท เช่น MAS Policy Controller, MAS Security Manager และ Agent Terminator เป็นต้น ดังนั้นสาธารณูปโภคของระบบเอเจนท์หนึ่งๆ จึงควรที่จะมีกระบวนการทำงานซึ่งอาศัยทรัพยากรทางเครือข่ายต่ำ เพื่อเหลือไว้ให้เอเจนท์และสาธารณูปโภคอื่นๆ ในระบบสามารถใช้งานได้อย่างเต็มที่ด้วย

ระบบแมทช์เมคกิ้งเอเจนท์นั้นจัดเป็นสาธารณูปโภคพื้นฐานของระบบเอเจนท์เช่นกัน ดังนั้นโครงสร้างของแมทช์เมคกิ้งเอเจนท์ที่ดีจึงต้องเป็นโครงสร้างที่ช่วยลดการติดต่อสื่อสารระหว่างแมทช์เมคกิ้งเอเจนท์ให้เหลือน้อยที่สุด ซึ่งมีข้อดีเพิ่มเติมจากที่ได้กล่าวมาแล้วคือ ทำให้ประสิทธิภาพการทำงานของระบบแมทช์เมคกิ้งเอเจนท์สูงขึ้น เนื่องจากใช้เวลาไปกับการส่งผ่านข้อมูลระหว่างแมทช์เมคกิ้งเอเจนท์น้อยเมื่อเทียบกับเวลาทำงานของกลไกการจับคู่ภายในแต่ละแมทช์เมคกิ้งเอเจนท์ แต่หากโครงสร้างแมทช์เมคกิ้งเอเจนท์ที่ออกแบบมานั้นทำให้จำนวนการติดต่อสื่อสารระหว่างแมทช์เมคกิ้งเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เจนท์เพิ่มขึ้นแล้ว ประสิทธิภาพการทำงานของระบบก็จะลดลงทันที ทั้งของระบบเอเจนท์และของระบบแมทช์เมคกิ้งเอเจนท์เอง Shehory (2000) ได้นำเสนอปัญหาที่จะเกิดขึ้นกับระบบเอเจนท์ขนาดใหญ่ไว้ว่า จำนวนการติดต่อสื่อสารที่เกิดขึ้นภายในระบบจะมีค่าเป็นฟังก์ชันโพลิโนเมียลของจำนวนเอเจนท์เลยทีเดียว ดังนั้นหากระบบแมทช์เมคกิ้งเอเจนท์ยังใช้จำนวนการติดต่อสื่อสารมาก ก็จะยิ่งเพิ่มปัญหาให้กับระบบเอเจนท์นั้นๆ ขึ้นไปอีก

นอกจากการออกแบบโครงสร้างแมทช์เมคกิ้งเอเจนท์ที่ช่วยลดจำนวนการสื่อสารแล้ว Shehory (2000) ยังได้สรุปแนวทางการลดจำนวนการติดต่อสื่อสารระหว่างเอเจนท์ด้วยวิธีอื่นอีก ได้แก่การร่วมมือกันทำงานโดยลดจำนวนการติดต่อระหว่างกันและกันลงให้มีความซับซ้อนของจำนวนการติดต่อคงที่ และการกำหนดขอบเขตความร่วมมือระหว่างสมาชิกภายในกลุ่ม ดังจะได้กล่าวในบทต่อไป

นอกจากข้อพิจารณาข้างต้นแล้ว การออกแบบยังต้องพิจารณาถึงความจริงที่ว่า เอเจนท์สามารถเข้าหรือออกจากระบบเมื่อใดก็ได้ ซึ่งจะทำให้มิได้มีแต่ข้อความประเภทโฆษณาและคำร้องหากแต่มีข้อความที่ใช้ถอนโฆษณาหรือถอนคำร้องที่เข้ามาเกี่ยวข้องในระบบแมทช์เมคกิ้งเอเจนท์ด้วย ข้อเท็จจริงนี้ทำให้ระบบแมทช์เมคกิ้งเอเจนท์มีโครงสร้างแตกต่างจากระบบสารบบทั่วไปที่มักมีสมมติฐานว่า โฆษณาที่นำมาฝากไว้ที่สารบบจะไม่ค่อยเปลี่ยนแปลง ตัวอย่างเช่น Microsoft Active Directory Service (Microsoft, 2000) เป็นต้น การถอนหรือเปลี่ยนโฆษณาความสามารถจากสารบบที่มีการทำงานแบบ Push Model หรือ Semi-Push Model นั้นจะทำให้จำนวนการติดต่อสื่อสารระหว่างสารบบในระบบสูงขึ้นอย่างมาก อันเป็นผลมากจากการอัพเดตการถอนหรือเปลี่ยนโฆษณาไปยังสารบบอื่นๆ นั่นเอง ดังนั้นโครงสร้างของระบบแมทช์เมคกิ้งเอเจนท์จึงต้องเป็นโครงสร้างที่ได้รับผลกระทบจากการอัพเดตหรือเปลี่ยนแปลงโฆษณาให้น้อยที่สุด ทั้งนี้โครงสร้างที่จะได้รับผลกระทบจากปัญหาดังกล่าวมาก ได้แก่ โครงสร้างที่มีการเชื่อมต่อจำนวนมากระหว่างแมทช์เมคกิ้งเอเจนท์ ทำให้แมทช์เมคกิ้งเอเจนท์จำนวนมากต้องมีการอัพเดตโฆษณาในกรณีที่มีการทำงานแบบ Push model ตัวอย่างเช่น โครงสร้างแบบ Lattice และโครงสร้างแบบลำดับชั้นที่มีความสูงมากได้แก่ A4 โครงสร้างทั้งสองมีความซับซ้อนของจำนวนการติดต่อระหว่างแมทช์เมคกิ้งเอเจนท์อยู่ที่ $O(\sqrt{n})$ และ O (d) ตามลำดับ (ความซับซ้อนของโครงสร้างหลังอาจแปลงได้เป็น $O(\log_x n)$ เมื่อ x คือจำนวน child per parent ของลำดับชั้น และ n คือจำนวนแมทช์เมคกิ้งเอเจนท์) แนวทางแก้ไขปัญหาประเภทนี้จะต้องกระทำโดยพยายามลดการติดต่อสื่อสารเพื่อการถอนหรือปรับเปลี่ยนโฆษณาระหว่างแมทช์เมคกิ้งเอเจนท์ลง กล่าวคือ เปลี่ยนกระบวนการดังกล่าวให้มีลักษณะเป็นแบบ Pull model มากขึ้นนั่นเอง หรืออาจจะใช้การหน่วงการอัพเดตโฆษณาแทนก็ได้ ตัวอย่างของสารบบที่ใช้แนวทางการหน่วงโฆษณาได้แก่ Microsoft ADS เป็นต้น อย่างไรก็ตาม แนวทางนี้ไม่เหมาะสมกับระบบเอเจนท์เป็นอย่างยิ่ง เนื่องจากผู้ใช้ของระบบแมทช์เมคกิ้งเอเจนท์ได้แก่เอเจนท์ โดยภาพรวมแล้วเอเจนท์จะทำงานอยู่ตลอด ซึ่งแตกต่างจากมนุษย์ที่ต้องมีช่วงเวลาพักซึ่งจะส่งผลให้ภาระงานของระบบในช่วงเวลานั้นน้อยลง และทำให้สามารถอาศัยช่วงเวลานั้นทำการอัพเดตโฆษณาที่ถูกหน่วงได้ แต่สำหรับเอเจนท์แล้วแทบจะไม่มีช่วงเวลาดังกล่าวอยู่เลย ดังนั้นการแก้ปัญหาข้างต้นจึงต้องกระทำโดยอาศัยการลดการติดต่อสื่อสารระหว่างแมทช์เมคกิ้งเอเจนท์ลงเท่านั้น ในงานวิจัยนี้จะนำเสนอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างแบบลำดับชั้นเช่นเดียวกันกับโครงสร้างแบบ A4 แต่จะเป็นลำดับชั้นที่มีความสูงไม่มากและมีความสูงคงที่เพื่อลดจำนวนการติดต่อระหว่างแมทช์เมคกิ้งเอเจนท์ รวมถึงจะมีการทำงานภายในระบบแมทช์เมคกิ้งเอเจนท์เป็นแบบ Pull Model ด้วย

โดยสรุปแล้ว งานวิจัยนี้จะพยายามแก้ไขปัญหาประสิทธิภาพของเทคโนโลยีแมทช์เมคกิ้งเอเจนท์โดยเสนอว่า ควรจัดเรียงแมทช์เมคกิ้งเอเจนท์เป็นแบบลำดับชั้นที่มีความสูงน้อยและคงที่ ทั้งนี้เพื่อลดปริมาณการติดต่อระหว่างแมทช์เมคกิ้งเอเจนท์ในการให้บริการคำร้องขอหนึ่งๆ และพัฒนาให้มีความสอดคล้องกับลักษณะเฉพาะต่างๆ ของเอเจนท์ ดังที่ได้กล่าวมาแล้วนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การออกแบบและสร้างระบบ

เพื่อทดสอบว่าระบบที่นำเสนอมีความสามารถเพียงใดและพิสูจน์ให้เห็นว่าสถาปัตยกรรมที่นำเสนอมีประสิทธิภาพสูงกว่าสถาปัตยกรรมแบบอื่น ดังนั้นระบบแมทช์เมคกิ้งเอเจนท์จึงได้ถูกพัฒนาขึ้นมาให้เห็นจริงตามแนวทางที่นำเสนอไว้ โดยมีขั้นตอนหลักๆ ได้แก่ การออกแบบโครงสร้างของแมทช์เมค-กิ้งเอเจนท์ การออกแบบการทำงานร่วมกัน การเลือกใช้การแสดงความรู้ การออกแบบกลไกการจับคู่ และการพัฒนาระบบ ตามลำดับ

5.1 การเลือกโครงสร้างแมทช์เมคกิ้งเอเจนท์ให้กับระบบ DMAA

ขั้นตอนแรกได้แก่การออกแบบโครงสร้างของแมทช์เมคกิ้งเอเจนท์ที่มีความเหมาะสมและมีประสิทธิภาพในการจับคู่ จากบทที่ 4 จะสามารถสรุปได้ว่าโครงสร้างของแมทช์เมคกิ้งเอเจนท์ที่ดีจะต้องคำนึงถึงปัจจัยต่างๆ ได้แก่

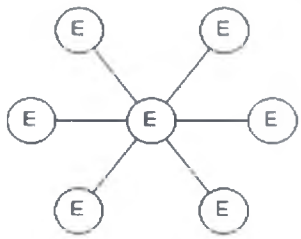
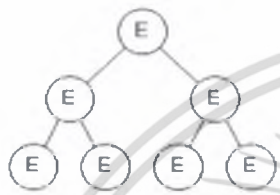
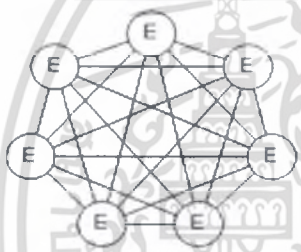
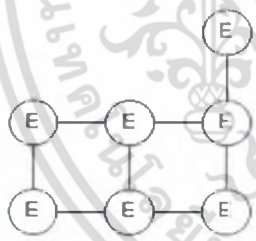
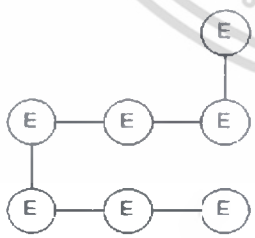
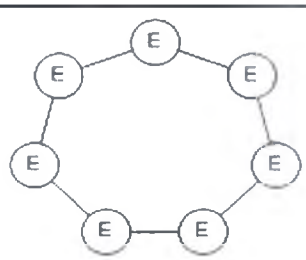
- เป็นโครงสร้างที่จัดเรียงแมทช์เมคกิ้งเอเจนท์โดยคำนึงถึงคุณสมบัติโดยทั่วไปของเอเจนท์ได้แก่ความสามารถทางสังคมและการมีภาวะอิสระ
- กลไกการร่วมมือกันทำงานของแมทช์เมคกิ้งเอเจนท์อาศัยการติดต่อสื่อสารระหว่างแมทช์เมคกิ้งเอเจนท์น้อย เพื่อจะได้ไม่รบกวนการทำงานของเอเจนท์และสาธารณูปโภคอื่นของระบบเอเจนท์นั้นๆ
- มีการกำหนดขอบเขตความร่วมมือระหว่างแมทช์เมคกิ้งเอเจนท์ให้อยู่ในวงจำกัด
- เป็นโครงสร้างที่ได้รับผลกระทบจากการอัพเดตหรือเปลี่ยนแปลงโฆษณาน้อยที่สุด

จากคุณสมบัติข้างต้น โครงสร้างที่มีความเหมาะสมที่สุดจะได้แก่โครงสร้างแบบลำดับชั้นที่มีความสูงไม่มากและมีความสูงคงที่เพื่อลดจำนวนการติดต่อระหว่างแมทช์เมคกิ้งเอเจนท์ มีการทำงานภายในระบบเป็นแบบ Pull Model

โครงสร้างมาตรฐานที่เป็นไปได้สำหรับเอนทิตีใดๆ ในระบบเครือข่ายจะได้แก่ Star, Tree (Hierarchy), Network, Bus และ Ring โดยโครงสร้างแบบ Network แบ่งได้เป็นแบบ Full-Meshed และแบบ Partial-Meshed (ตัวอย่างเช่น โครงสร้างแบบ Lattice) โครงสร้างแต่ละประเภทมีความซับซ้อนของจำนวนการติดต่อเฉลี่ยระหว่างเอนทิตี สำหรับการติดต่อหนึ่งครั้ง ตัวอย่างเช่น ดังตารางที่ 5.1 หากพิจารณาระบบแมทช์เมคกิ้งเอเจนท์ที่ประกอบไปด้วยแมทช์เมคกิ้งเอเจนท์ 7 ตัว จะมีจำนวนการติดต่อสื่อสารเฉลี่ยของแมทช์เมคกิ้งเอเจนท์ภายในโครงสร้างแมทช์เมคกิ้งเอเจนท์แต่ละประเภทดังตารางที่ 5.2 เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.1 ความซับซ้อนของจำนวนการติดต่อเฉลี่ยระหว่างแต่ละเอนทิตีภายในระบบเครือข่าย

ตัวอย่างรูปของโครงสร้าง	ชื่อของโครงสร้าง	ความซับซ้อน	หมายเหตุ
	Star	O(1)	DMAA
	Tree	$O(\log_x n)$	A4 (Cao, Kerbyson and Nudd. 2001), DMAA
	Network: Full-meshed	O(1)	Ideal
	Lattice (Network: Partial-meshed)	$O(\sqrt{n})$	Lattice (Shehory. 2000)
	Bus	O(n)	
	Ring	O(n)	

หมายเหตุ: n คือจำนวนเอนทิตีและ x คือความลึกของลำดับชั้น

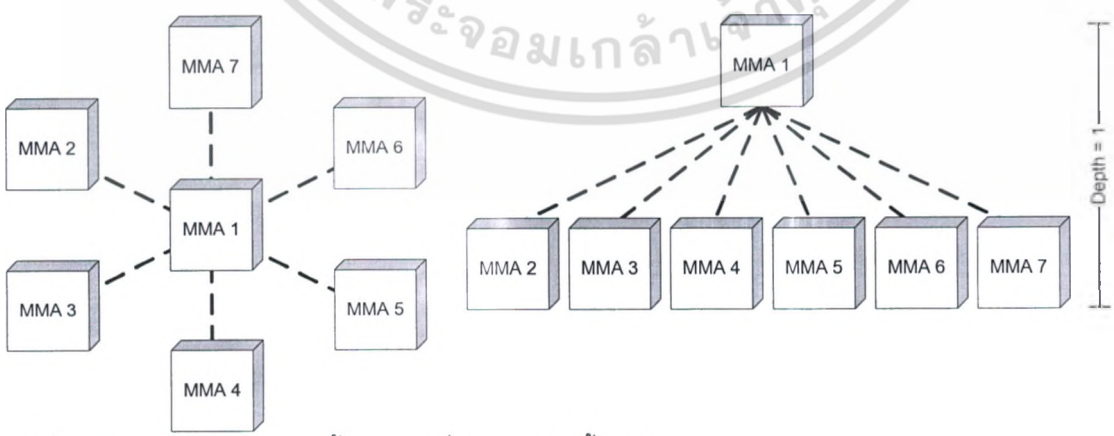
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.2 ตัวอย่างจำนวนการติดต่อเฉลี่ยระหว่างแต่ละแมทซ์เมคกิ้งเอเจนท์ภายในระบบแมทซ์เมคกิ้งเอเจนท์แบบต่างๆ

ชื่อของโครงสร้าง	ความซับซ้อน	จำนวน MMA	จำนวน Logical Agent Link (LINK)	จำนวนการติดต่อเฉลี่ย (DIST)	ต้นทุนการติดต่อ (LINK x DIST)
Star	O(1)	7	6	1.71	10.26
Tree (Binary)	$O(\log_x n)$		6	2.29	13.74
Network: Full-meshed	O(1)		21	1.00	21.00
Lattice (Network: Partial-meshed)	$O(\sqrt{n})$		8	1.90	15.20
Bus	O(n)		6	2.67	16.02
Ring	O(n)		7	2.00	14.00

จากตารางที่ 5.1 และ 5.2 จะพบว่าโครงสร้างแบบ Ring และ Bus เป็นโครงสร้างที่มีความซับซ้อนของจำนวนการติดต่อระหว่างแมทซ์เมคกิ้งเอเจนท์สูงที่สุด ในขณะที่โครงสร้างประเภท Network แบบ Full-meshed เป็นโครงสร้างที่มีความซับซ้อนต่ำที่สุด แต่โครงสร้างนี้กลับกลายเป็นโครงสร้างที่มีต้นทุนสูงสุด เพราะแมทซ์เมคกิ้งเอเจนท์ทุกตัวในระบบต้องรู้เองว่าจะส่งคำร้องไปที่แมทซ์เมคกิ้งเอเจนท์ใดจึงจะทำให้ได้รับคำตอบกลับมาและต้องสร้างการติดต่อระหว่างแต่ละแมทซ์เมคกิ้งเอเจนท์โดยตรงด้วย

ดังนั้นในงานวิจัยนี้จึงเลือกใช้โครงสร้างที่อาศัยจำนวนการติดต่อสูงกว่า Full-Meshed เล็กน้อย แต่มีความซับซ้อนเหมือนกัน และมีต้นทุนที่ต่ำกว่าแทน ได้แก่ โครงสร้างแบบ Star ซึ่งก็คือโครงสร้างแบบลำดับชั้นที่มีความสูงเท่ากับ 1 เสมอนั่นเอง ดังรูปที่ 5.1



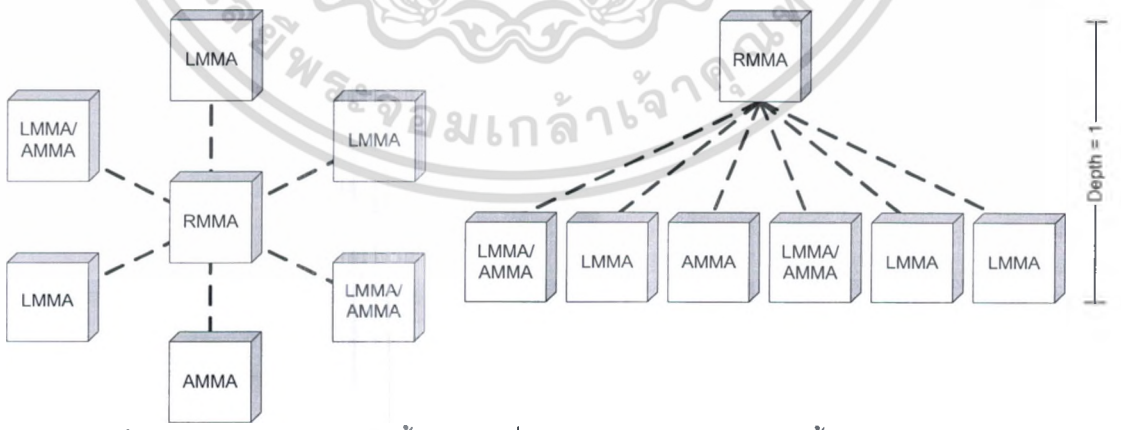
รูปที่ 5.1 โครงสร้างแมทซ์เมคกิ้งเอเจนท์ที่นำเสนอในเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งต่อไปที่ต้องคำนึงถึงคือการกำหนดขอบเขตความร่วมมือระหว่างแมทซ์เมคกิ้งเอเจนท์ให้อยู่ในวงจำกัด กล่าวคือกำหนดหน้าที่รับผิดชอบของแมทซ์เมคกิ้งเอเจนท์แต่ละตัวให้มีความชัดเจนนั่นเอง

5.2 การทำงานร่วมกันของแมทซ์เมคกิ้งเอเจนท์

นอกจากการออกแบบโครงสร้างของกลุ่มแมทซ์เมคกิ้งเอเจนท์แล้ว ยังต้องคำนึงถึงการทำงานร่วมกันของแมทซ์เมคกิ้งเอเจนท์ในแต่ละส่วนของโครงสร้างด้วย ตัวอย่างเช่น การกำหนดว่าแมทซ์เมคกิ้งเอเจนท์ใดมีหน้าที่อะไร และต้องติดต่อกับแมทซ์เมคกิ้งเอเจนท์ใดบ้าง เป็นต้น หลักการสำคัญในการออกแบบการทำงานร่วมกันของแมทซ์เมคกิ้งเอเจนท์ที่ได้จากการวิเคราะห์ในบทก่อนได้แก่ การกำหนดขอบเขตความร่วมมือระหว่างแมทซ์เมคกิ้งเอเจนท์ให้อยู่ในวงจำกัดนั่นเอง อย่างไรก็ตาม ก่อนที่จะออกแบบการทำงานร่วมกันของแมทซ์เมคกิ้งเอเจนท์นั้นจะต้องเข้าใจในลักษณะทั่วไปของกลุ่มเอเจนท์ก่อน กลุ่มเอเจนท์หรือระบบเอเจนท์ได้แก่กลุ่มของเอเจนท์จำนวนหนึ่งที่มาทำงานร่วมกัน โดยการทำงานร่วมกันของเอเจนท์ภายในกลุ่มสามารถแบ่งได้เป็น 2 ประเภทได้แก่ Coordination และ Self-Interested Operation ภายในกลุ่มเอเจนท์หนึ่งจะประกอบไปด้วยกลุ่มเอเจนท์ย่อยหลายกลุ่มมาทำงานร่วมกันอีกทีหนึ่ง กลุ่มเอเจนท์ย่อยเหล่านี้จะใช้สาธารณูปโภคพื้นฐานของกลุ่มเอเจนท์หลักร่วมกันซึ่งได้แก่ Place, ANS และ แมทซ์เมคกิ้งเอเจนท์ เป็นต้น ดังนั้น กลุ่มแมทซ์เมคกิ้งเอเจนท์ที่ให้บริการกลุ่มเอเจนท์ใดๆ จะแบ่งได้เป็น 2 ประเภทได้แก่แมทซ์เมคกิ้งเอเจนท์ที่ให้บริการเฉพาะกลุ่มเอเจนท์ย่อย และแมทซ์เมคกิ้งเอเจนท์หลักที่ให้บริการกับทั้งระบบเอเจนท์นั้น ซึ่งแมทซ์เมคกิ้งเอเจนท์เหล่านี้จะทำงานร่วมกันตามแนวปฏิบัติที่ได้กำหนดเอาไว้ล่วงหน้าแล้ว นอกจากนี้แมทซ์เมคกิ้งเอเจนท์แต่ละตัวจะถูกกำหนดหน้าที่รับผิดชอบที่แตกต่างกันเอาไว้ล่วงหน้าด้วยเช่นกัน



รูปที่ 5.2 โครงสร้างแมทซ์เมคกิ้งเอเจนท์ซึ่งประกอบไปด้วยแมทซ์เมคกิ้งเอเจนท์ 3 ประเภท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในงานวิจัยนี้แบ่งแมทซ์เมคกิ้งเอเจนท์ตามหน้าที่รับผิดชอบออกเป็น 3 ประเภท ได้แก่ Local MMA (LMMA), Authoritative MMA (AMMA) และ Root MMA (RMMA) เมื่อจัดเข้ากับโครงสร้างที่นำเสนอจะมีลักษณะดังรูปที่ 5.2

แมทซ์เมคกิ้งเอเจนท์แต่ละประเภทจะมีหน้าที่ดังต่อไปนี้

- LMMA คือแมทซ์เมคกิ้งเอเจนท์ที่มีหน้าที่รับผิดชอบการให้บริการจับคู่แก่เอเจนท์ภายในกลุ่มเอเจนท์ย่อยที่ตนเองสังกัดอยู่ ในงานวิจัยนี้ได้กำหนดให้ LMMA หนึ่งตัวรับผิดชอบกลุ่มเอเจนท์ย่อยเพียง 1 กลุ่มเท่านั้น
- AMMA คือแมทซ์เมคกิ้งเอเจนท์ซึ่งสังกัดกลุ่มเอเจนท์หนึ่งและมีโฆษณาความสามารถที่สอดคล้องกับบริการที่ร้องขอมา กล่าวคือ สามารถจับคู่เอเจนท์ผู้ใช้บริการกับเอเจนท์ผู้ขอรับบริการได้ แมทซ์เมคกิ้งเอเจนท์ที่มีศักดิ์เป็น AMMA นั้น อาจเป็นแมทซ์เมคกิ้งเอเจนท์อิสระที่ไม่สังกัดกลุ่มเอเจนท์ย่อยใดๆ หรืออาจเป็นตัว LMMA เองก็ได้
- RMMA คือแมทซ์เมคกิ้งเอเจนท์ซึ่งสังกัดกลุ่มเอเจนท์หนึ่งและมีความสามารถในการควบคุมการแจกจ่ายคำร้องที่ยังไม่สามารถจับคู่โฆษณาที่เหมาะสมได้ไปยังแมทซ์เมค-กิ้งเอเจนท์ภายในกลุ่มเอเจนท์ที่มีความสามารถให้บริการจับคู่กับคำร้องนั้นได้

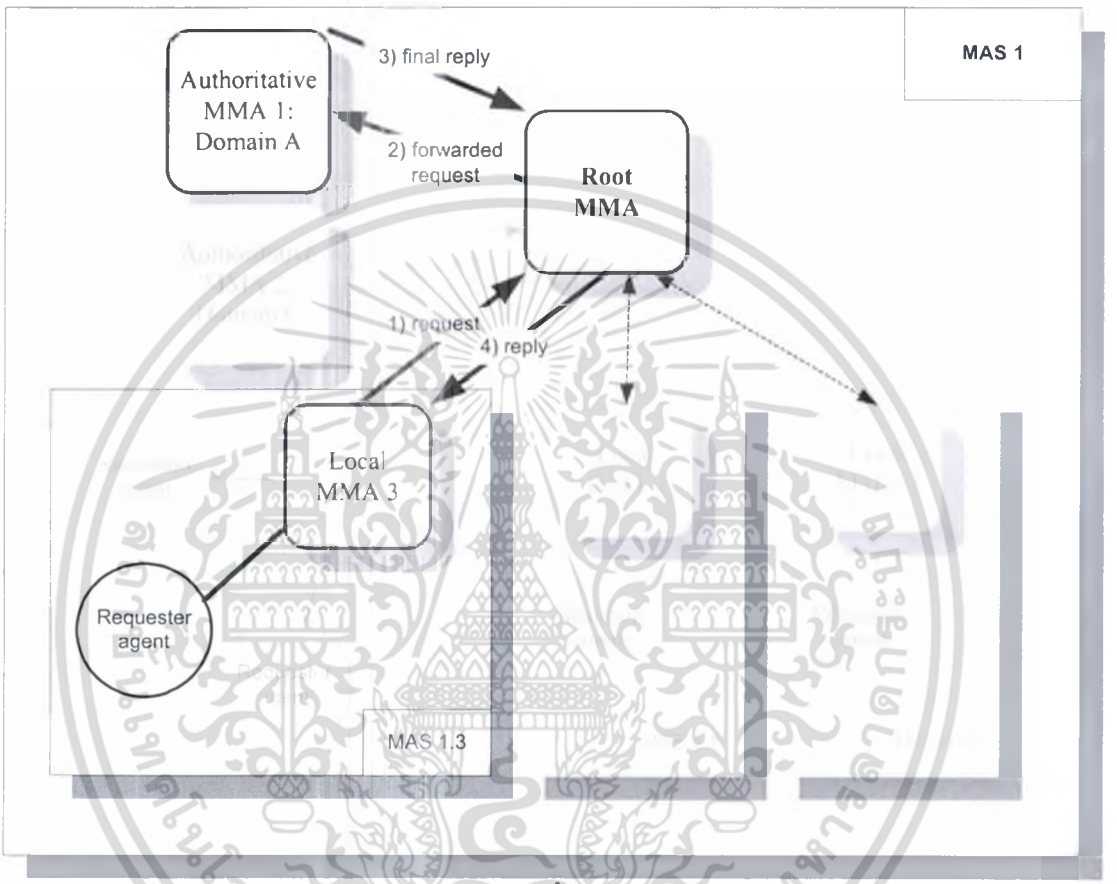
ความสัมพันธ์หลักของแมทซ์เมคกิ้งเอเจนท์ทั้งสามประเภทนี้จะเป็นไปดังรูปที่ 5.3 จากรูป LMMA จะส่งคำร้องขอที่ได้รับมาจากเอเจนท์ผู้ร้องขอไปยัง RMMA เนื่องจาก LMMA ดังกล่าวไม่สามารถค้นหาโฆษณาที่เข้าคู่กับคำร้องขอนี้ได้ เมื่อ RMMA ได้รับคำร้องขอนี้แล้วก็จะใช้วิธีการบางอย่างมาตรวจสอบว่าควรจะส่งต่อคำร้องขอดังกล่าวไปยังแมทซ์เมคกิ้งเอเจนท์ใด วิธีการดังกล่าวนั้นมีหลากหลายมาก ตัวอย่างเช่น 1) จัดส่งไปยังแมทซ์เมคกิ้งเอเจนท์อื่นที่อยู่ใกล้ที่สุด (Shehory. 2000) หรือ 2) จัดส่งไปยังแมทซ์เมคกิ้งเอเจนท์ที่อยู่ในระดับที่เหนือกว่า เนื่องจากเป็นแมทซ์เมคกิ้งเอเจนท์ที่มีโฆษณาทั้งหมดของแมทซ์เมคกิ้งเอเจนท์ที่อยู่ในระดับที่ต่ำกว่า (Cao *et al.* 2001)

วิธีการแรกนั้นเป็นวิธีที่ใช้กับระบบแมทซ์เมคกิ้งเอเจนท์แบบ Lattice ซึ่งมีการทำงานเป็นแบบ Pull model เป็นวิธีที่มีข้อเสียที่สำคัญคือ ไม่สามารถระบุได้ว่าการส่งต่อคำร้องหนึ่งๆ จะสิ้นสุดเมื่อใด ในขณะที่วิธีหลังนั้นใช้กับระบบแมทซ์เมคกิ้งเอเจนท์แบบ A4 ซึ่งมีการทำงานเป็นแบบ Push model ที่มีข้อเสียที่สำคัญคือต้องสิ้นเปลืองทรัพยากรเครือข่ายจำนวนมากให้กับระบบแมทซ์เมคกิ้งเอเจนท์ประเภทนี้ แต่วิธีนี้เป็นวิธีที่มีข้อดีคือสามารถจำกัดจำนวนการส่งผ่านคำร้องให้อยู่ในวงจำกัดได้

ระบบแมทซ์เมคกิ้งเอเจนท์ในงานวิจัยนี้มีการทำงานเป็นแบบ Pull Model ที่ไม่ใช้วิธีจัดส่งไปยังแมทซ์เมคกิ้งเอเจนท์ข้างเคียงเหมือนสถาปัตยกรรมแบบ Lattice หากแต่จะเลือกใช้วิธีที่สามารถจำกัดการส่งผ่านคำร้องให้อยู่ในวงจำกัดได้ เพื่อนอกจากจะได้รับผลดีจากการทำงานแบบ Pull Model แล้ว ยังสามารถขจัดข้อเสียของการทำงานแบบนี้ได้อีกด้วย ดังนั้น จึงนำหลักการของการจำกัดขอบเขตความรับผิดชอบและการจำกัดความร่วมมือระหว่างแมทซ์เมคกิ้งเอเจนท์เข้ามาใช้ โดยแมทซ์เมคกิ้งเอเจนท์แต่ละตัวที่ RMMA จะส่งผ่านคำร้องไปให้จะต้องเป็น AMMA เสมอ กล่าวคือ คำตอบที่แมทซ์เมคกิ้งเอเจนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอบกลับมาจะถือเป็นที่สุด หากแมทซ์เมคกิ้งเอเจนท์ดังกล่าวไม่สามารถค้นหาเอเจนท์ผู้ให้บริการได้แล้ว ก็แปลว่าในระบบเอเจนท์นั้นไม่มีแมทซ์เมคกิ้งเอเจนท์อื่นที่สามารถให้บริการจับคู่แก่คำร้องขอรับบริการนี้ได้ และ RMMA ก็ไม่จำเป็นจำต้องส่งผ่านคำร้องดังกล่าวไปให้แมทซ์เมคกิ้งเอเจนท์อื่นภายในระบบเอเจนท์นี้อีก



รูปที่ 5.3 ความสัมพันธ์ระดับแนวคิดของแมทซ์เมคกิ้งเอเจนท์ 3 ประเภท: LMMA, AMMA และ RMMA

การจำกัดขอบเขตความรับผิดชอบที่งานวิจัยนี้เลือกใช้กับ AMMA จะได้แก่ การแบ่ง AMMA ตามโดเมนของโฆษณาความสามารถ AMMA แต่ละตัวจะทำหน้าที่ในการจับคู่ให้กับเฉพาะคำร้องขอรับบริการที่อยู่ในขอบเขตของโดเมนซึ่ง AMMA นั้นรับผิดชอบอยู่เท่านั้น ดังนั้นโฆษณาความสามารถที่นำมาฝากไว้กับ AMMA หนึ่งๆ จะต้องเป็นโฆษณาที่มีเนื้อหาสาระอยู่ในขอบเขตของโดเมนที่ AMMA นั้นๆ รับผิดชอบด้วย การประยุกต์ใช้การจำกัดความรับผิดชอบตามโดเมนของงานนี้ทำให้มีข้อดีเพิ่มขึ้นได้แก่การที่ AMMA แต่ละตัวสามารถเลือกใช้อัลกอริทึมจับคู่ที่เหมาะสมกับประเภทของงานที่ตนเองรับผิดชอบได้ซึ่งจะทำให้กระบวนการจับคู่มีความแม่นยำสูงขึ้นรวมถึงอาจทำให้มีประสิทธิภาพการทำงานสูงขึ้นด้วย อย่างไรก็ตาม ประเด็นดังกล่าวนั้นไม่ใช่ประเด็นหลักของงานวิจัยนี้ ดังนั้น จึงเลือกใช้อัลกอริทึมในการจับคู่ของแต่ละแมทซ์เมคกิ้งเอเจนท์ที่จะใช้ทดสอบเหมือนกันทั้งหมดแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่กล่าวมาข้างต้น สามารถสรุปได้ว่า RMMA ในงานวิจัยนี้มีหน้าที่ในการควบคุมการแจกจ่ายคำร้องที่ยังไม่ได้รับบริการจาก LMMA ต่างๆ ไปยัง AMMA แต่ละตัว โดยอาศัยกระบวนการจัดประเภทให้กับคำร้องขอนั้นว่าอยู่ในเรื่องอะไร เพื่อจะได้ส่งต่อไปยัง AMMA ที่รับผิดชอบโดเมนซึ่งเกี่ยวข้องกับเรื่องนั้น

ในงานวิจัยนี้ โดเมนจะหมายถึงกลุ่มของแนวคิดที่มีความสัมพันธ์ซึ่งกันและกัน และมีแนวคิดหลักร่วมกันอยู่ โดยที่ชื่อของโดเมนจะนิยมตั้งตามแนวคิดหลักดังกล่าว ตัวอย่างเช่น หากเราสนใจเรื่องต่างๆ เกี่ยวกับดนตรี โดยมีเพลง การขับร้อง โน้ตเพลง และการแสดงคอนเสิร์ต เป็นตัวอย่างของแนวคิดที่มีความสัมพันธ์กัน และมี "ดนตรี" เป็นแนวคิดหลักร่วมกันแล้ว "ดนตรี" ก็จะจัดเป็นโดเมนหนึ่ง เป็นต้น ตัวอย่างการใช้งานเช่น เมื่อมีคำร้องขอรับบริการค้นหาเอเจนท์ที่สามารถให้โน้ตเพลงของเพลงชื่อ Canon ได้ถูกส่งเข้ามา RMMA ก็จะสามารถตรวจสอบได้ว่าคำร้องขอดังกล่าวอยู่ในขอบเขตของแนวคิด "โน้ตเพลง" ซึ่งมีแนวคิดหลักคือ "ดนตรี" อีกทีหนึ่ง จากนั้น RMMA ก็จะส่งต่อคำร้องขอดังกล่าวไปยัง AMMA ที่รับผิดชอบโดเมน "ดนตรี" ได้ เป็นต้น อย่างไรก็ตามการกำหนดว่าแนวคิดใดคือโดเมนนั้นขึ้นอยู่กับว่าระบบเอเจนท์ที่ระบบแมทช์เมคกิ้งเอเจนท์นี้สังกัดประกอบไปด้วยเอเจนท์ที่ทำงานประเภทใดบ้าง เช่น หากนำระบบแมทช์เมคกิ้งเอเจนท์ในงานวิจัยนี้ไปใช้กับระบบเอเจนท์ขนส่งระดับประเทศ โดเมนที่สนใจก็อาจจะประกอบไปด้วย การขนส่งคนด้วยรถส่วนบุคคล การขนส่งคนด้วยรถมวลชน การขนส่งสินค้าด้วยรถบรรทุก เป็นต้น ดังนั้นจุดอ่อนที่สำคัญของงานวิจัยนี้คือหากแบ่งโดเมนที่แต่ละ AMMA รับผิดชอบได้ไม่ดีพอแล้ว จะทำให้ความแม่นยำและประสิทธิภาพของระบบแมทช์เมคกิ้งเอเจนท์ลดลงนั่นเอง

5.3 การแสดงความต้องการและความสามารถของเอเจนท์

หลังจากที่ได้ออกแบบลักษณะและการทำงานภายนอกของระบบแมทช์เมคกิ้งเอเจนท์ไปแล้ว สิ่งต่อไปที่ต้องคำนึงถึงก็คือ การออกแบบระบบการทำงานภายในแต่ละแมทช์เมคกิ้งเอเจนท์นั่นเอง สิ่งที่ต้องคำนึงถึงสำหรับการทำงานภายในแต่ละแมทช์เมคกิ้งเอเจนท์แบ่งเป็น 2 ประเด็น ได้แก่ 1) การแสดงความต้องการและความสามารถของเอเจนท์ และ 2) กลไกการจับคู่ที่ใช้

การแสดงความต้องการและความสามารถของเอเจนท์นั้นมีหลากหลายรูปแบบเช่นเดียวกับการแสดงความรู้ โดยการแสดงความรู้ที่เป็นมาตรฐานนิยมในปัจจุบันของเอเจนท์นั้นได้แก่ First Order Logic หรือ Predicate Calculus Sentence การแสดงความรู้ด้วย PCS นั้นถูกนำไปใช้ในเทคโนโลยีเอเจนท์หลายๆ อย่าง ได้แก่ KIF, CYC Ontology, JESS, PowerLOOM และ LOOM เป็นต้น ดังนั้น ในงานวิจัยนี้จึงเลือกใช้ PCS ในการแสดงความต้องการและความสามารถของเอเจนท์

ประโยคของ PCS จะประกอบขึ้นจากสัญลักษณ์ 4 ประเภท ได้แก่ ตัวแปร ค่าคงที่ สัญลักษณ์จริง-เท็จ และฟังก์ชันหรือเพรดิเคต ประโยคโดยทั่วไปใน PCS จะประกอบไปด้วยสัญลักษณ์เพรดิเคตและอาร์กิวเมนต์ตั้งแต่หนึ่งตัวขึ้นไป ซึ่งรวมเรียกว่า Atomic Sentence หรือ Atom ตัวอย่างเช่น map(australia), father(joseph,?Person) เป็นต้น เราสามารถใช้สัญลักษณ์ประเภทใดก็ได้เป็นอาร์กิว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมนต์ นอกจากนี้ ยังอาจใช้สัญลักษณ์เพรดิเคตซ้อนกันหลายๆ ชั้นเพื่อสามารถแสดงความต้องการหรือความสามารถของเอเจนท์ได้ละเอียดยิ่งขึ้น ทั้งนี้ ในงานวิจัยนี้จะรองรับเฉพาะประโยคประเภทอะตอมเท่านั้น ดังนั้น ระบบ DMAA จึงยังไม่สามารถรองรับกรณีที่เอเจนท์ส่งความต้องการหรือความสามารถหลายๆ ประโยคมาในการติดต่อครั้งเดียวได้

ในกรณีที่มีการใช้สัญลักษณ์เพรดิเคตซ้อนกัน สัญลักษณ์เพรดิเคตตัวนอกสุดของประโยคจะแสดงถึง Performative ของเอเจนท์ที่ส่งข้อความนั้นมา ถ้าหากเป็นการโฆษณาแล้ว เพรดิเคตนี้จะแสดงถึงการกระทำที่เอเจนท์ผู้ให้บริการสามารถปฏิบัติให้กับเอเจนท์อื่นได้ แต่หากเป็นการร้องขอ เพรดิเคตดังกล่าว จะระบุถึงการกระทำที่เอเจนท์นี้คาดหวังจากเอเจนท์รายอื่น ตัวอย่างเช่น โฆษณา give(map(orlando)) ซึ่งมีความหมายว่าเอเจนท์นี้สามารถให้แผนที่ของเมืองออร์ลันโด้กับเอเจนท์อื่นๆ ได้นั้นมี Performative คือ give โดยจะเป็นการแสดงว่าเอเจนท์อื่นๆ สามารถมาดาวน์โหลดข้อมูลที่ระบุ (แผนที่เมืองออร์ลันโด้) ไปจากเอเจนท์นี้ได้ เป็นต้น

ตัวอย่างของโฆษณาความสามารถที่ใช้การแสดงความรู้ตามวิธีที่ได้กล่าวมาแล้ว ได้แก่

- ความสามารถของเอเจนท์ ก : "(ฉัน) สามารถค้นหาบทวิจารณ์ที่เกี่ยวกับหนังสือที่แต่งโดยใครก็ได้" จะเขียนในรูปของประโยค PCS ได้เป็น "find(review (book(Title,Author)))"
- ความสามารถของเอเจนท์ ข : "(ฉัน) สามารถบอกเวลาที่เที่ยวบินใดๆ ของสายการบินแคนตัสเริ่มออกเดินทางได้" จะเขียนได้เป็น "tell-about(departure_time (flight(Flight_number,airline(qantas))))"
- ความสามารถของเอเจนท์ ค : "(ฉัน) สามารถให้แผนที่ของเมืองอะไรก็ได้ในประเทศมาเลเซีย" จะเขียนได้เป็น "give(map(City,malaysia))"
- ความสามารถของเอเจนท์ ง : "(ฉัน) สามารถค้นหาแผนที่ของเมืองอะไรก็ได้" จะเขียนได้เป็น "find(map(City))"

ในงานวิจัยนี้ ทั้งโฆษณาและคำร้องขอจะถูกแปลงจากประโยค PCS ไปเป็น List เพื่อให้สามารถเลือกใช้เครื่องอนุมานชั้นนำซึ่งนิยมใช้กับเอเจนท์ (ตัวอย่างเช่น JESS หรือ CLIPS) มาเป็น Matching Engine ได้ เมื่อนำตัวอย่างโฆษณาความสามารถข้างต้นมาแปลงเป็น List แล้วจะมีลักษณะดังนี้

- ความสามารถของเอเจนท์ ก : "find(review(book(Title,Author)))" จะแปลงได้เป็น "(find (review (book Title Author)))"
- ความสามารถของเอเจนท์ ข : "tell-about(departure_time(flight(Flight_number ,airline(qantas))))" จะแปลงได้เป็น "(tell-about (departure_time (flight Flight_number (airline qantas))))"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความสามารถของเอเจนท์ ค : "give(map(City,malaysia))" จะสามารถแปลงได้เป็น "(give (map City malaysia))"
- ความสามารถของเอเจนท์ ง : "find(map(City))" เมื่อนำมาแปลงเป็น List แล้วจะได้ว่า "(find (map City))"

5.4 กลไกการจับคู่

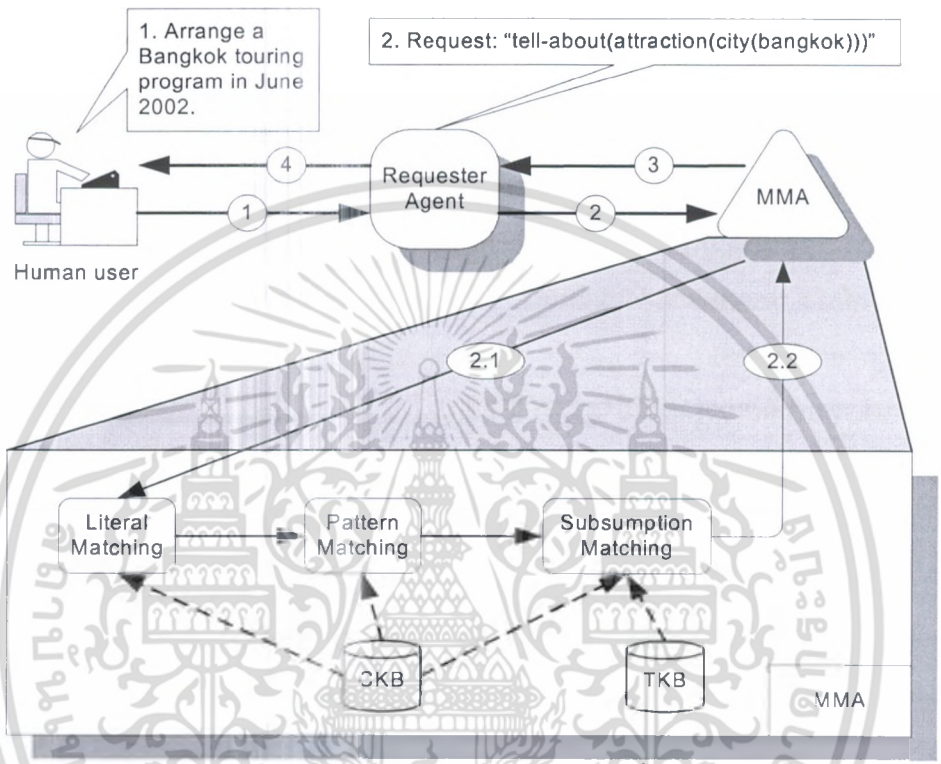
ประเด็นต่อไปที่ต้องคำนึงถึงในการออกแบบการทำงานภายในของแต่ละแมทช์เมคกิ้งเอเจนท์ ได้แก่กลไกการจับคู่ การจับคู่ หมายถึง กระบวนการนำโฆษณาความสามารถของเอเจนท์ผู้ให้บริการซึ่งฝากไว้กับแมทช์เมคกิ้งเอเจนท์มาเปรียบเทียบกับคำร้องขอรับบริการโดยมีวัตถุประสงค์เพื่อค้นหาเอเจนท์ผู้ให้บริการที่สามารถให้บริการเอเจนท์ใดๆ ที่ร้องขอบริการเข้ามาได้ ทั้งนี้ ผลลัพธ์จากการจับคู่นั้นสามารถแบ่งได้เป็น 3 ประเภทตามความแม่นยำ ได้แก่ Exact Match, Plug-In Match และ Relaxed Match (Sycara *et al.* 1999)

การเข้าคู่กันที่มีความแม่นยำสูงสุด ได้แก่ การเข้าคู่แบบ Exact Match ซึ่งจะได้มาจากกระบวนการจับคู่แบบ Literal Matching หรือ Word-to-Word Matching (ค้นหาโฆษณาที่เหมือนคำร้องทุกตัวอักษร) และ Pattern Matching (โดยใช้กระบวนการ Unification) การเข้าคู่กันที่มีความแม่นยำรองลงมา ได้แก่ การเข้าคู่แบบ Plug-In Match การเข้าคู่แบบนี้จะได้แก่กรณีที่โฆษณา subsume คำร้อง (Subsumption-based Match) และกรณีที่กลับกัน (Reverse Subsumption-based Match) สำหรับการเข้าคู่ที่มีความแม่นยำน้อยที่สุด (ค้นหาได้โฆษณาที่มีความเกี่ยวข้องเพียงเล็กน้อยกับคำร้องขอรับบริการ) จะได้แก่การเข้าคู่แบบ Relaxed Match ซึ่งวิธีการจับคู่ที่ให้ผลลัพธ์ประเภทนี้ได้แก่ Context (Domain) Matching (ดูว่าอยู่ในโดเมนเดียวกันหรือไม่) และ Profile Matching (เปรียบเทียบโดยใช้เทคนิค Term Frequency-Inverse Document Frequency หรือ TF-IDF (Salton and Wong. 1975))

กลไกการจับคู่ของแมทช์เมคกิ้งเอเจนท์ในงานวิจัยนี้จะประกอบไปด้วยขั้นตอนดังรูปที่ 5.4 จากรูป เมื่อผู้ใช้ซึ่งเป็นมนุษย์มีความต้องการที่จะไปเที่ยวพักผ่อน โดยเลือกสถานที่และเวลาเป็นกรุงเทพฯ และมิถุนายน พ.ศ. 2545 ตามลำดับ ผู้ใช้ดังกล่าวก็จะสั่งเอเจนท์ผู้ช่วยประจำตัว (Personal Assistant Agent) ลงไปว่าให้ช่วยจัดโปรแกรมการท่องเที่ยวกรุงเทพฯ ในเดือนมิถุนายน พ.ศ. 2545 มาให้เลือกที (ขั้นตอนที่ 1) จากนั้นเอเจนท์ดังกล่าวก็จะทำงานตามที่ผู้ใช้สั่งมา โดยหากไม่มีข้อมูลใดๆ ก็จะทำการติดต่อกับเอเจนท์อื่นเพื่อขอข้อมูล ในรูปที่ 5.4 เอเจนท์ของผู้ใช้กำลังหาข้อมูลเกี่ยวกับสิ่งที่น่าสนใจในกรุงเทพฯ อยู่ และพบว่าตัวเองไม่มีข้อมูลดังกล่าว รวมถึงไม่รู้จักเอเจนท์ที่มีข้อมูลดังกล่าวด้วย ดังนั้น จึงได้ส่งคำร้องขอรับบริการให้ข้อมูลเกี่ยวกับสิ่งที่น่าสนใจในกรุงเทพฯ ไปยังแมทช์เมคกิ้งเอเจนท์ (ขั้นตอนที่ 2) จากนั้นแมทช์เมคกิ้งเอเจนท์จะนำคำร้องดังกล่าวไปเข้าแถวคอยเพื่อรอรับบริการจับคู่ โดยการจับคู่จะเริ่มจากวิธี Literal Matching ก่อน หากยังไม่พบเอเจนท์ผู้ให้บริการที่เข้าคู่กันก็จะนำไปจับคู่ด้วยวิธี Pattern Matching ต่อไป หากยังไม่พบอีกก็จะนำไปเข้าสู่วิธีการจับคู่แบบสุดท้ายได้แก่ Subsumption Matching (หรือ Shared-Concept Matching) จากนั้นเมื่อเสร็จสิ้นกระบวนการจับคู่ภายในแมทช์เมคกิ้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอเจนท์แล้ว แมทช์เมคกิ้งเอเจนท์ก็จะตอบกลับไปยังเอเจนท์ผู้ร้องขอบริการด้วยชื่อของเอเจนท์ที่สามารถบอกข้อมูลเกี่ยวกับสิ่งที่น่าสนใจในกรุงเทพฯ ได้ (ขั้นตอนที่ 3) เมื่อถึงขั้นนี้เอเจนท์ของผู้ใช้ก็สามารถที่จะไปติดต่อกับเอเจนท์ข้างต้นเพื่อขอข้อมูลที่ตนเองขาดหายไปได้ และสามารถทำงานที่ผู้ใช้สั่งได้สำเร็จในที่สุด (ขั้นตอนที่ 4)



รูปที่ 5.4 กลไกการจับคู่ภายในแมทช์เมคกิ้งเอเจนท์

ในบรรดาวิธีการจับคู่ทั้งสามวิธีที่เลือกใช้ในงานวิจัยนี้ วิธีการจับคู่แบบ Subsumption Matching ดูจะเป็นวิธีการจับคู่ที่ต้องอาศัยเวลาในการปฏิบัติงานมากที่สุด การทำงานของวิธีนี้จะอาศัย Taxonomic Knowledge Base (TKB) มาใช้ในการค้นหาการเข้าคู่แบบ Subsumption-based Match และ Reverse Subsumption-based Match ซึ่งจะให้ผลลัพธ์เป็นโพษณาที่มีความเป็นทั่วไปมากกว่าหรือเจาะจงกว่าคำร้องตามลำดับ ตัวอย่างเช่น เมื่อใช้การจับคู่แบบ Subsumption Matching กับคำร้องขอรับบริการ "(find (publication Title Author_name))" จะได้ผลลัพธ์ที่เจาะจงกว่าเป็น "(find (journal Title Author))" และได้ผลลัพธ์ที่มีความเป็นทั่วไปมากกว่าเป็น "(find (printing Title Author))" ทั้งนี้เนื่องจากเมื่อพิจารณาถึงความสัมพันธ์ของแนวคิดแบบ ISA (ISA relation) ใน TKB แล้วพบว่าแนวคิด publication นั้น subsume แนวคิด journal (publication π journal) และถูก subsume โดยแนวคิด printing ตามลำดับ (printing π publication) TKB ที่เลือกใช้ในงานวิจัยนี้ได้มาจาก Lexical Database for English: WordNet 1.6 (CSL. 2002) ซึ่งเป็นฐานข้อมูลเกี่ยวกับความสัมพันธ์ระหว่างกลุ่มของคำหรือแนวคิดในภาษาอังกฤษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

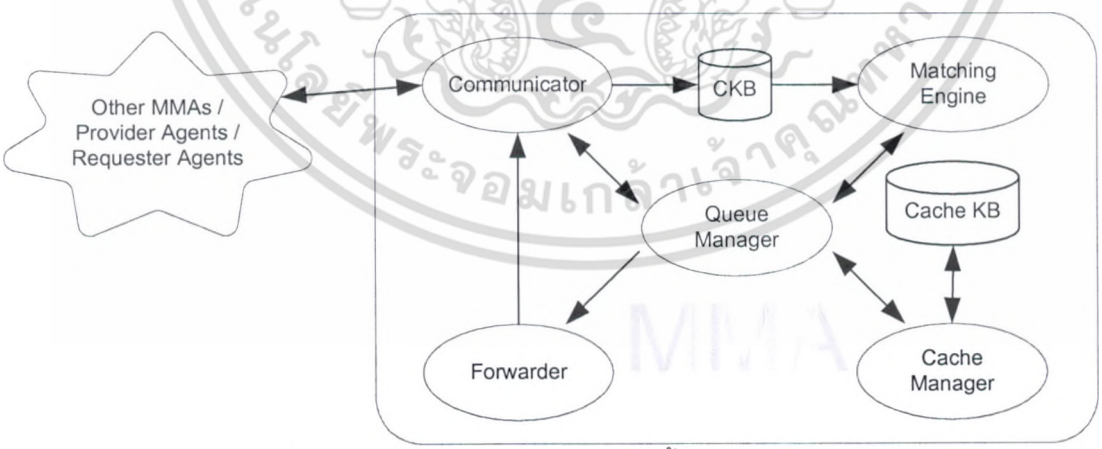
วิธีการจับคู่ข้างต้นเป็นวิธีการที่ให้ผลลัพธ์การจับคู่แบบ Plug-In Match ส่วนการเข้าคู่แบบ Exact Match ในระบบแมทช์เมคกิ้งเอเจนท์ที่นำเสนอนั้นจะได้มาจากวิธีการจับคู่แบบ Literal Matching และ Pattern Matching ตามลำดับ สำหรับการเข้าคู่แบบสุดท้ายซึ่งได้แก่ Relaxed Match เนื่องจากเทคนิค TF-IDF นั้นไม่เหมาะสมที่จะนำมาใช้กับข้อความขนาดเล็กอย่างโฆษณาและคำร้องขอรับบริการของเอเจนท์ ดังนั้น จึงไม่ใช้การจับคู่แบบ Profile Matching ในงานวิจัยนี้ แต่จะใช้เพียงการจับคู่แบบ Domain Matching ซึ่งสถาปัตยกรรมที่นำเสนอก็ได้ประยุกต์ใช้โดยนัยอยู่แล้ว ณ แมทช์เมคกิ้งประเภท RMMA นั่นเอง

5.5 การสร้างระบบ DMAA

เพื่อทดสอบว่าสถาปัตยกรรม DMAA ที่นำเสนอในงานวิจัยนี้มีประสิทธิภาพดีพอที่จะสามารถนำไปประยุกต์ใช้กับระบบเอเจนท์ขนาดใหญ่ได้ ระบบแมทช์เมคกิ้งเอเจนท์จึงได้ถูกสร้างขึ้นมาตามแนวคิดของสถาปัตยกรรม DMAA ข้างต้น

ภาษาที่ใช้พัฒนาระบบนี้ได้แก่ภาษาจาวา เหตุผลสำคัญในการเลือกใช้ภาษานี้ในการพัฒนาก็คือความสามารถในการทำงานได้บนแพลตฟอร์มที่หลากหลาย รวมถึงการเป็นภาษาที่ได้รับความนิยมสูงสุดในการนำมาใช้กับเทคโนโลยีเอเจนท์ (Bigus and Bigus. 2001) ตัวอย่างของเอเจนท์ที่ถูกพัฒนาขึ้นด้วยภาษาจาวา ได้แก่ AgentBuilder (Reticular Systems), Aglet (IBM), JADE (CSELT), JATLite (Stanford), JESS (Sandia National Lab) และ Voyager (Object Space) เป็นต้น

สำหรับระบบ DMAA, A4 และ Middle Agent ที่จะใช้ทดสอบ ภายในแมทช์เมคกิ้งเอเจนท์แต่ละตัวที่สร้างขึ้นจะประกอบไปด้วยมอดูลต่างๆ ดังรูปที่ 5.5



รูปที่ 5.5 มอดูลต่างๆ ภายในแมทช์เมคกิ้งเอเจนท์แต่ละตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.1 **มอดูล** Communicator

มอดูลนี้จะรับผิดชอบการติดต่อสื่อสารกับเอเจนท์อื่นๆ ทั้งหมด โดยตั้งอยู่บนพื้นฐานของโพรโทคอลแลกความรู้ KQML มอดูลนี้สามารถแบ่งได้เป็น 3 ส่วน ได้แก่ 1) Router 2) Router Client และ 3) KQML Parser

ในปัจจุบัน โมเดลการติดต่อสื่อสารระหว่างโปรแกรมคอมพิวเตอร์โดยส่วนใหญ่แล้วจะเป็นแบบรับ-ให้บริการ (ไคลเอนท์เซิร์ฟเวอร์) ซึ่งเป็นการติดต่อสื่อสารแบบประสานเวลา (Synchronous) ในขณะที่การติดต่อสื่อสารระหว่างเอเจนท์นั้นจะเป็นแบบไม่ประสานเวลา (Asynchronous) เป็นหลัก (CDR. 1999) ดังนั้น Router ซึ่งเป็นส่วนแรกของมอดูลนี้จึงมีคุณสมบัติที่ช่วยให้เอเจนท์หนึ่งสามารถติดต่อกับเอเจนท์อื่นได้ โดยที่ไม่จำเป็นต้องรอให้เอเจนท์นั้นเริ่มทำงานเสียก่อน มอดูลย่อย Router จะให้บริการ Agent Message Router หรือ AMR ซึ่งมีลักษณะคล้ายคลึงกับระบบไปรษณีย์อิเล็กทรอนิกส์ โดย AMR นี้จะทำหน้าที่จัดเก็บและส่งต่อข้อความ ตัวอย่างเช่น เมื่อเอเจนท์ ก. ต้องการจะส่งข้อความไปยังเอเจนท์ ข. เอเจนท์ ก. ก็จะส่งข้อความมายัง Router จากนั้นเมื่อ Router พบว่าเอเจนท์ ข. กำลังทำงานอยู่ Router ก็จะส่งต่อข้อความไปให้เอเจนท์ ข. ทันที หรือหากพบว่ายังไม่สามารถติดต่อเอเจนท์ ข. ได้ Router ก็จะเก็บข้อความนี้ไว้และรอจนกว่าเอเจนท์ ข. จะกลับมาทำงานอีกครั้ง จึงจะส่งข้อความไปให้ ในการติดต่อสื่อสารหนึ่งๆ เอเจนท์ ก. ไม่จำเป็นต้องรู้ที่อยู่ (หมายเลขไอพีและพอร์ท) ของเอเจนท์ ข. เพียงแต่รู้จักชื่อของเอเจนท์ ข. ก็สามารถติดต่อสื่อสารกันได้แล้ว ทั้งนี้เป็นเพราะว่า AMR มีบริการ ANS ให้ด้วย ทั้งนี้การทำงานของ Router จะตั้งอยู่บนสมมติฐานที่ว่าเอเจนท์ทุกตัวจะต้องรู้จักชื่อ ที่อยู่ และวิธีติดต่อ Router เสมอ

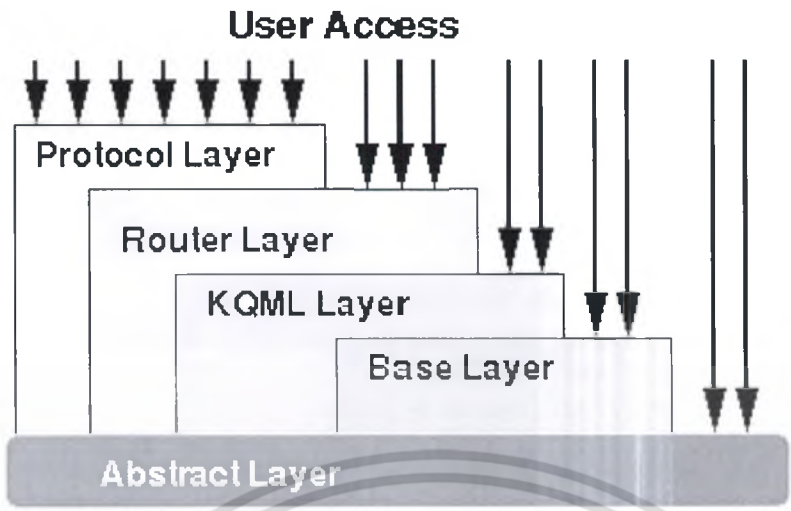
สำหรับส่วนที่สองของมอดูลนี้ ได้แก่ Router client ซึ่งทำหน้าที่ในการติดต่อสื่อสารกับเอเจนท์อื่นในรูปแบบของข้อความประเภท KQML ผ่านทาง Router นอกจากนี้มอดูลย่อยนี้ยังให้บริการโพรโทคอล ในการติดต่อสื่อสารโดยผ่านทาง Router ด้วย ได้แก่ การลงทะเบียนและการติดต่อกับ Router รวมถึงการยกเลิกการติดต่อก่อนที่จะหยุดการทำงานของตัวเองลง เป็นต้น

ส่วนสุดท้ายของมอดูล Communicator ได้แก่มอดูลย่อย KQML Parser เป็นส่วนที่ให้บริการในการสร้างและถอดข้อความตามข้อกำหนดของโพรโทคอล KQML ทั้งนี้ส่วนย่อยทั้งสามของมอดูล Communicator นั้นถูกสร้างขึ้นมาจาก JATLite (CDR. 1999) อีกทีหนึ่ง

สถาปัตยกรรม JATLite นั้นแบ่งออกเป็นชั้นๆ ดังรูปที่ 5.6 ได้แก่ Abstract Layer, Base Layer, KQML Layer, Router Layer และ Protocol Layer ทุกชั้นถูกเขียนขึ้นด้วยภาษาจาวาทั้งหมด

ชั้นแรกได้แก่ Abstract Layer จะประกอบไปด้วยกลุ่มของ Abstract Class ที่จำเป็นสำหรับการนำ JATLite มาใช้งานจริง ชั้นที่ 2 ได้แก่ Base Layer เป็นชั้นที่ให้บริการการติดต่อสื่อสารแบบ TCP/IP กล่าวคือเป็นชั้นที่ใช้สร้างซ็อกเก็ต ในการติดต่อระหว่างโปรแกรมคอมพิวเตอร์นั่นเอง ชั้นที่ 3 ได้แก่ KQML Layer จะให้บริการสร้างหรือถอดความจากข้อความประเภท KQML ชั้นต่อมาได้แก่ Router Layer เป็นชั้นที่ให้บริการ AMR และบริการอื่นๆ ที่เกี่ยวข้อง และสำหรับชั้นสุดท้ายได้แก่ Protocol Layer เป็นชั้นที่ให้บริการโพรโทคอลการสื่อสารต่างๆ ได้แก่ FTP และ SMTP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 สถาปัตยกรรมของ JATLite

ส่วน Router และ Router Client ของมอดูลนี้จะใช้บริการของชั้น Router Layer ของ JATLite เป็นหลัก สำหรับส่วน KQML Parser นั้นจะใช้บริการของ KQML Layer เป็นหลักนั่นเอง

นอกจากคุณสมบัติพื้นฐานข้างต้นแล้ว มอดูล Communicator ยังมีหน้าที่อื่นอีกได้แก่ การรับและจัดเก็บโฆษณาความสามารถลงในฐานความรู้ความสามารถ ในขั้นต้นได้เลือกใช้ JESS มาทำเป็นฐานความรู้ความสามารถโดยการใช้ JESS API โปรแกรมติดต่อกับระบบแมทช์เมคกิ้งเอเจนท์ที่พัฒนาขึ้น ดังรูปที่ 5.7

อย่างไรก็ตาม จากการทดสอบประสิทธิภาพเบื้องต้นของ JESS นั้นพบว่าเหมาะสมที่จะใช้จัดเก็บโฆษณาไม่เกิน 10,000 โฆษณาเท่านั้น ซึ่งหากเกินจากนี้จะทำให้การทำงานของการจับคู่ช้ามาก นอกจากนี้ JESS ยังไม่รองรับการจับคู่แบบ Subsumption Matching ด้วย ดังนั้น จึงเปลี่ยนไปใช้ Berkeley DB (Sleepycat Software. 2001) ซึ่งเป็นฐานข้อมูลแบบ B$^+$Tree มาประยุกต์ทำเป็นฐานความรู้ความสามารถแทน

5.5.2 มอดูล Matching Engine

มอดูลนี้มีหน้าที่หลักคือการให้บริการจับคู่แบบต่างๆ แก่คำร้องขอรับบริการใดๆ มอดูลนี้ประกอบไปด้วยส่วนต่างๆ ได้แก่ 1) ฐานความรู้ความสามารถ (ได้แก่ฐานความรู้ที่เก็บความสามารถของเอเจนท์ที่ถูกโฆษณามายังแมทช์เมคกิ้งเอเจนท์นี้) 2) TKB (เป็นฐานข้อมูล Ontology ที่ไม่มีการกำหนดชนิดของอาร์กิวเมนต์ที่เป็นไปได้สำหรับความสัมพันธ์หนึ่งๆ) และ 3) ตัว Matching Engine เอง ซึ่งจะให้บริการกลไกจับคู่ 3 แบบ (ดูหัวข้อ 5.4) สำหรับกลไกแบบ Subsumption Matching นั้นจะใช้เทคนิค Similarity Rating ของ Girardi and Ibrahim (1994) และ Lin (1998) ซึ่งเป็นเทคนิคที่ใช้ TKB ช่วยในการให้คะแนนความเหมือนสำหรับแต่ละคู่ของโฆษณาและคำร้องขอรับบริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
import jess.*;
...
public class REngine{
    Rete rete;
    ...

    public REngine() {
            rete = new Rete(); //create the Jess engine
            FileReader fr = new FileReader("./CKB/CKBHeader.clp"); //open CKB header for rule
            Jesp j = new Jesp(fr, rete); // file parser I/O
            j.parse(false); //parse CKB header into Jess engine
            fr.close();
            ...
    }//constructor1

    public String Inference(String ReqCmd){//Inference
            ...
            sw = new StringWriter();
            rete.addOutputRouter("t", sw); //redirect router "t" to stream "sw"
            ...
            rete.executeCommand("(assert " + ReqCmd + ")"); //put request into matching engine
            rete.executeCommand("(run)"); //run engine
            StringTokenizer st = new StringTokenizer(sw.toString(), "\n");//get the result
            ...
            rete.executeCommand("(reset)");
            ...
            return Result;
    }//Inference

    public void AddADV(String RuleNo, String Sndr, String Cont){
            ...
            rete.executeCommand("(defrule r" + RuleNo + " " + Cont.trim() + " => (printout t \"" +
               Sndr.trim() + "\" crlf))"); //add a new advertisement into CKB
            ...
    }//AddADV

    public void DelADV(long RNo){//delete an advertisement from CKB (RNo is the ADV no.)
            ...
            rete.executeCommand("(undefrule r" + SS.StringFillChaF(Long.toString(RNo),18,"0") + ")");
            ...
    }//DelADV
}// class REngine
```

รูปที่ 5.7 การใช้งาน JESS API

ใน DMAA แมทช์เมคกิ้งเอเจนท์ที่มีกลไกการจับคู่เป็นแบบข้างต้นจะได้แก่ LMMA และ AMMA ส่วนแมทช์เมคกิ้งเอเจนท์ประเภท RMMA นั้นจะไม่ใช้กลไกข้างต้น เนื่องจากในระบบ DMAA นั้น RMMA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่ได้ให้บริการจับคู่แก่เอเจนท์ทั่วไป หากแต่จะให้บริการจับคู่แก่แมทช์เมคกิ้งเอเจนท์ด้วยกันเองเท่านั้น RMMA จะทำหน้าที่จำแนกคำร้องที่ถูกส่งต่อมาจาก LMMA ว่าอยู่ในโดเมนใด เพื่อที่จะได้ส่งต่อคำร้องดังกล่าวไปยัง AMMA ที่รับผิดชอบได้ (ทั้งนี้ RMMA จะรู้ว่า AMMA หนึ่งรับผิดชอบให้บริการจับคู่แก่คำร้องและโฆษณาที่อยู่ในโดเมนใดได้นั้น AMMA นั้นๆ จะต้องโฆษณาความสามารถของตนเองมายัง RMMA ไว้ล่วงหน้าแล้ว) กลไกที่ RMMA ใช้ในการตรวจสอบว่าคำร้องขอรับบริการหนึ่งจัดอยู่ในโดเมนใดนั้นจะอาศัยฟิลด์ ":ontology" ของข้อความนั้น ค่าที่อยู่ในฟิลด์ดังกล่าวจะบ่งบอกว่าคำร้องนั้นอยู่ในเรื่องอะไร และเมื่อนำ TKB เข้ามาใช้งานร่วมด้วยจะทำให้สามารถสรุปได้ว่าคำร้องดังกล่าวจัดอยู่ในโดเมนที่ตรงกับความรับผิดชอบของ AMMA ใดบ้าง ตัวอย่างเช่น คำร้องขอรับบริการ "(tell-about (fare (ticket (bus Bus_name (line_no Line_number)))))" ซึ่งมีฟิลด์ :ontology คือ bus_fare และ เป็นคำร้องที่ LMMA ประจำกลุ่มไม่สามารถให้บริการได้ จึงถูกส่งต่อมายัง RMMA จากนั้น RMMA จะค้นหาแนวคิดที่เกี่ยวข้องกับแนวคิด bus_fare ว่าอยู่ในความรับผิดชอบของ AMMA ใดในระบบโดยใช้ลำดับชั้นแสดงความสัมพันธ์แบบ ISA ระหว่างแนวคิด (TKB) ผลลัพธ์ที่ได้ในตัวอย่างนี้คือคำร้องนี้ถูกจัดอยู่ในโดเมน public_transportation และถูกส่งต่อไปยัง AMMA ที่มีหน้าที่รับผิดชอบโดเมนดังกล่าวอยู่ เป็นต้น

5.5.3 มอดูล Cache Manager

เพื่อให้การทำงานของแมทช์เมคกิ้งเอเจนท์เร็วขึ้น และสามารถเปรียบเทียบกับระบบแมทช์เมคกิ้งเอเจนท์แบบอื่นได้ DMAA จึงมีมอดูลหนึ่งซึ่งทำหน้าที่ให้บริการจัดเก็บผลลัพธ์จากกระบวนการจับคู่ครั้งก่อนๆ และสามารถนำข้อมูลดังกล่าวมาใช้งานในภายหลัง มอดูลนี้ได้แก่ Cache Manager นั่นเอง มอดูลนี้จะให้บริการจัดเก็บผลลัพธ์จากกระบวนการจับคู่ที่แมทช์เมคกิ้งเอเจนท์อื่นตอบกลับมายังแมทช์เมคกิ้งเอเจนท์นี้ เมื่อมีคำร้องขอรับบริการที่แมทช์เมคกิ้งเอเจนท์นี้ไม่สามารถให้บริการได้ถูกส่งเข้ามา แมทช์เมคกิ้งเอเจนท์นี้จะใช้มอดูลนี้ค้นหาโฆษณาจากฐานข้อมูลแคช (Cache KB) ก่อน ถ้าพบ ก็จะส่งผลลัพธ์ที่พบในแคชกลับไป แต่หากไม่พบก็จะส่งคำร้องขอดังกล่าวให้กับมอดูล Forwarder ต่อไป

5.5.4 มอดูล Forwarder

ทำหน้าที่ส่งต่อคำร้องขอรับบริการที่แมทช์เมคกิ้งเอเจนท์นี้ไม่สามารถให้บริการได้ ไปยังแมทช์-เมคกิ้งเอเจนท์อื่นที่สามารถให้บริการจับคู่แก่คำร้องนั้นได้ ไม่โดยทางตรงก็ทางอ้อม ตัวอย่างเช่น ในระบบ DMAA หาก LMMA หนึ่งไม่สามารถค้นหาโฆษณาที่เข้าคู่กับคำร้องขอได้ ก็จะส่งคำร้องนี้ไปยังมอดูล Forwarder ของตนเอง จากนั้น Forwarder จะส่งคำร้องดังกล่าวไปยัง RMMA โดยเปลี่ยนฟิลด์ ":sender" จากชื่อของเอเจนท์ผู้ร้องขอไปเป็นชื่อของ LMMA นั้น เปลี่ยนฟิลด์ ":receiver" จากชื่อของตัว LMMA เองไปเป็นชื่อของ RMMA และเปลี่ยนฟิลด์ ":reply-with" ให้เป็นรหัสที่ LMMA นั้นกำหนดขึ้นมาเองแทนรหัสที่เอเจนท์ผู้ร้องขอกำหนด หลังจากที่ RMMA จัดประเภทได้แล้วว่าคำร้องนั้นอยู่ในโดเมนอะไรก็จะส่งคำร้องนั้นต่อไปยังมอดูล Forwarder ของตนเอง โดยจะเปลี่ยนเฉพาะฟิลด์ ":receiver" ให้เป็นชื่อของ AMMA ที่รับผิดชอบ สาเหตุที่เปลี่ยนเฉพาะฟิลด์ดังกล่าวนั้นเพื่อให้ AMMA สามารถตอบกลับไปยัง LMMA ได้โดยตรงโดยไม่ต้องผ่าน RMMA นั่นเอง อย่างไรก็ตาม AMMA จะไม่ตอบตรงกลับไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยังเอเจนท์ผู้ร้องขอเลย แต่จะส่งผ่าน LMMA แทน เนื่องจากเหตุผลทางด้านความปลอดภัยเป็นหลัก นอกจากสาเหตุหลักข้างต้นแล้ว กระบวนการดังกล่าวยังช่วยให้ LMMA สามารถจัดเก็บแคชผลลัพธ์จากแมทช์เมคกิ้งเอเจนท์อื่นเพื่อเพิ่มประสิทธิภาพการทำงานของระบบ DMMA ในระยะยาวได้อีกด้วย

5.5.5 **มอดูล** Queue Manager

Queue Manager เป็นมอดูลที่ทำหน้าที่ประติดประต่อกระบวนการทำงานส่วนต่างๆ ของแมทช์-เมคกิ้งเอเจนท์เข้าด้วยกัน และควบคุมลำดับการให้บริการให้มีลักษณะเป็นแบบ First Come First Serve เมื่อมีคำร้องขอรับบริการถูกส่งเข้ามาโดยมอดูล Communicator มอดูลนี้จะไปตรวจสอบว่ามีแถวคอยรับบริการอยู่หรือไม่ ถ้าไม่มีก็จะส่งคำร้องดังกล่าวเข้าสู่กระบวนการจับคู่ทันที แต่หากมีแถวคอยก็จะนำคำร้องดังกล่าวไปเข้าแถวเพื่อคอยรับบริการต่อไป เมื่อถึงลำดับที่คำร้องนั้นจะได้รับบริการแล้ว มอดูล Queue Manager นี้จะส่งต่อคำร้องนั้นไปยังมอดูล Matching Eengine เพื่อค้นหาโฆษณาที่เข้าคู่กันจากฐานความรู้ความสามารถของตนเอง ถ้าพบก็ตอบกลับไปยังเอเจนท์ผู้ร้องขอทันที แต่หากไม่พบก็จะส่งต่อคำร้องนั้นไปค้นหาในมอดูล Cache Manager ต่อไป ถ้ายังไม่พบอีก ก็จะส่งต่อคำร้องนั้นไปยังมอดูล Forwarder เพื่อส่งต่อไปให้แมทช์เมคกิ้งเอเจนท์อื่นช่วยค้นหาแทน

5.6 สรุป

เพื่อทดสอบว่าสถาปัตยกรรมของระบบแมทช์เมคกิ้งเอเจนท์ DMAA ที่นำเสนอในงานวิจัยนี้สามารถทำงานได้อย่างมีประสิทธิภาพ ระบบแมทช์เมคกิ้งเอเจนท์จึงได้ถูกพัฒนาขึ้นมาตามแนวคิดที่นำเสนอ แมทช์เมคกิ้งเอเจนท์แต่ละตัวจะประกอบไปด้วยมอดูล 5 มอดูล ได้แก่ มอดูล Communicator, มอดูล Matching Engine, มอดูล Cache Manager, มอดูล Forwarder และมอดูล Queue Manager

มอดูล Communicator และมอดูล Matching Engine มีหน้าที่ให้บริการติดต่อสื่อสารและการจับคู่โฆษณาตามลำดับ ในขณะที่มอดูล Cache Manager นั้นจะให้บริการด้านแคชแก่แมทช์เมคกิ้งเอเจนท์ ส่วนมอดูล Forwarder จะให้บริการส่งต่อคำร้องไปยังแมทช์เมคกิ้งเอเจนท์อื่น ซึ่งจะเป็นแมทช์เมคกิ้งเอเจนท์ใดนั้นขึ้นอยู่กับสถาปัตยกรรมที่เลือกใช้นั่นเอง มอดูลสุดท้ายได้แก่มอดูล Queue Manager เป็นมอดูลที่ทำหน้าที่จัดลำดับการให้บริการแก่คำร้องต่างๆ

นอกจากระบบที่ได้พัฒนาขึ้นตามแนวคิดของสถาปัตยกรรม DMAA แล้ว ผู้วิจัยยังได้ดัดแปลงบางมอดูลของระบบที่พัฒนาขึ้นเพื่อให้มีคุณสมบัติสอดคล้องกับสถาปัตยกรรมอื่น โดยมีจุดประสงค์คือสามารถทดสอบเปรียบเทียบสถาปัตยกรรมของระบบที่นำเสนอในอดีตกับสถาปัตยกรรมที่นำเสนอในงานวิจัยนี้ได้ ดังจะได้กล่าวในบทต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การทดสอบระบบ

เพื่อตรวจสอบว่าระบบ DMAA ที่นำเสนอในงานวิจัยนี้สามารถแก้ไขปัญหาประสิทธิภาพของเทคโนโลยีแมทช์เมคกิ้งเอเจนท์ที่มีอยู่ในปัจจุบันได้ ดังนั้นผู้วิจัยจึงได้พัฒนาระบบแมทช์เมคกิ้งเอเจนท์ขึ้นมาตามแนวคิดที่นำเสนอเพื่อทดสอบหาความสามารถของระบบดังกล่าว นอกจากนี้ยังได้พัฒนาระบบแมทช์เมคกิ้งเอเจนท์ขึ้นมาตามแนวคิดของสถาปัตยกรรมแบบอื่นที่ได้มีผู้นำเสนอไว้เพื่อเปรียบเทียบกับระบบที่นำเสนอในงานวิจัยนี้ด้วย

6.1 การออกแบบการทดสอบ

การทดสอบจะอาศัยการจำลองเอเจนท์ผู้ให้บริการและเอเจนท์ผู้ร้องขอ เอเจนท์ดังกล่าวจะทำงานอยู่ในเครื่องคอมพิวเตอร์เครื่องหนึ่งโดยเอเจนท์ผู้ร้องขอจะส่งคำร้องขอรับบริการไปให้ระบบแมทช์เมคกิ้งเอเจนท์ซึ่งทำงานอยู่ในเครื่องคอมพิวเตอร์อีกเครื่องหนึ่ง โดยเป็นการพัฒนาจริงขึ้นมาสำหรับการทดสอบ จากนั้นระบบแมทช์เมคกิ้งเอเจนท์จะตอบกลับมายังเอเจนท์ผู้ร้องขอว่าให้ไปติดต่อเอเจนท์ผู้ให้บริการตัวใดหรืออาจจะตอบมาว่าไม่มีเอเจนท์ที่สามารถให้บริการนี้ได้ก็ได้ การทดสอบจะเปรียบเทียบกันทั้งหมด 3 สถาปัตยกรรมคือ A4, DMAA และ Middle Agent ทั้งนี้แมทช์เมคกิ้งเอเจนท์แต่ละตัวจะมีลักษณะดังหัวข้อที่ 5.5

นอกจากการออกแบบโครงแบบการทำงานแล้ว ยังจะต้องออกแบบในเรื่องของสมมติฐานการทดลอง ข้อมูลที่ใช้ในการทดสอบ กรณีทดสอบต่างๆ และตัววัดประสิทธิภาพที่เลือกใช้ด้วย ดังนี้

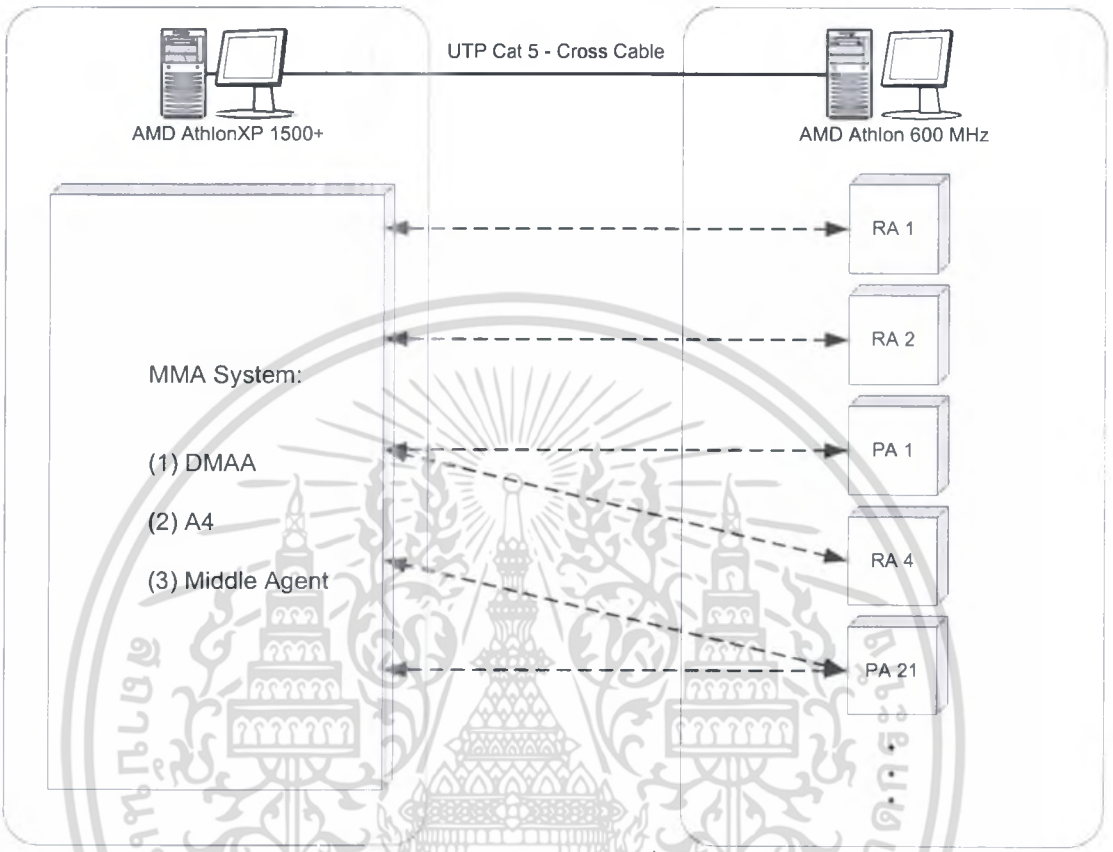
6.1.1 ตัววัด

การทดลองในงานวิจัยนี้มีวัตถุประสงค์หลักเพื่อตรวจสอบประสิทธิภาพของสถาปัตยกรรมที่นำเสนอ โดยตัววัดที่สามารถสะท้อนให้เห็นถึงประสิทธิภาพของระบบนั้นได้แก่ อัตราการให้บริการของระบบแมทช์เมคกิ้งเอเจนท์ (Service Rate) และเวลาตอบสนอง (Response Time) ในที่นี้ อัตราการให้บริการหมายถึงจำนวนคำร้องที่ระบบแมทช์เมคกิ้งเอเจนท์สามารถให้บริการจับคู่ได้ต่อหนึ่งหน่วยเวลา ซึ่งหาได้จากการนำจำนวนคำร้องขอที่ได้รับคำตอบมาหารด้วยระยะเวลาที่ระบบแมทช์เมคกิ้งเอเจนท์ใช้นับตั้งแต่เมื่อคำร้องแรกถูกส่งจนถึงเมื่อระบบสามารถตอบผลการจับคู่ของคำร้องสุดท้ายกลับไปยังเอเจนท์ผู้ร้องขอได้ และเวลาตอบสนองคือเวลาที่ระบบแมทช์เมคกิ้งเอเจนท์ใช้ในการให้บริการจับคู่นับตั้งแต่เมื่อคำร้องถูกส่งจนถึงเมื่อระบบตอบผลการจับคู่กลับไปยังเอเจนท์ผู้ร้องขอเฉลี่ยต่อหนึ่งคำร้อง ทั้งนี้การคำนวณหาเวลาตอบสนองของระบบแมทช์เมคกิ้งเอเจนท์หนึ่งๆ นั้นสามารถหาได้โดยการนำอัตราการให้บริการไปเข้าสูตรคำนวณของทฤษฎีแถวคอยนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.2 โครงแบบ

โครงแบบของระบบที่ใช้ในการทดสอบสำหรับทุกกรณีทดสอบมีลักษณะดังรูปที่ 6.1



รูปที่ 6.1 สภาพแวดล้อมของระบบที่ใช้ในการทดสอบ

จากรูปที่ 6.1 ระบบที่ใช้ในการทดสอบจะประกอบไปด้วยอุปกรณ์ดังต่อไปนี้

- เครื่องคอมพิวเตอร์รุ่น AMD AthlonXP 1500+ ความเร็ว 1.33 GHz ใช้หน่วยความจำ DDR-SDRAM 266 MHz ขนาด 256 MB และใช้ฮาร์ดดิสก์ความเร็วในการหมุน 7,200 รอบที่มีอินเตอร์เฟสเป็นแบบ UDMA-66 ใช้ในการรันระบบแมทช์เมคกิ้งเอเจนท์แบบต่างๆ ทั้งสามแบบ
- เครื่องคอมพิวเตอร์รุ่น AMD Athlon ความเร็ว 600 Ghz ใช้หน่วยความจำ SDRAM ขนาด 448 MB และใช้ฮาร์ดดิสก์ความเร็วในการหมุน 7,200 รอบ ใช้รันเอเจนท์ผู้ร้องขอ (RA) และเอเจนท์ผู้ให้บริการ (PA) ที่จำลองขึ้น
- การ์ดเครือข่ายแบบ Fast Ethernet บนเครื่องทั้งสอง พร้อมสายต่อสลับ UTP Cat 5 ซึ่งต่อตรงระหว่างเครื่องทั้งสอง
- ทดสอบบนแพลตฟอร์ม Windows Millennium ของไมโครซอฟท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.3 กรณีทดสอบ

การเลือกกรณีทดสอบ (Test Case) นั้นจะต้องพิจารณาถึงปัจจัยหลายอย่างที่มีผลกระทบต่อประสิทธิภาพของระบบ โดยปัจจัยที่สำคัญๆ ได้แก่ จำนวนโฆษณา สถาปัตยกรรมที่เลือกใช้ และจำนวนแมทช์เมคกิ้งเอเจนท์

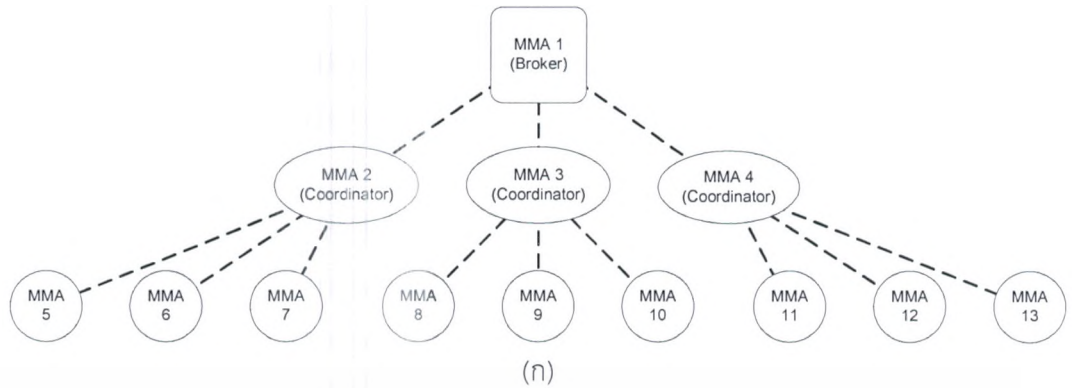
ปัจจัยแรกที่ต้องคำนึงถึงได้แก่ จำนวนโฆษณาที่เอเจนท์ผู้ให้บริการนำมาฝากไว้กับระบบแมทช์เมคกิ้งเอเจนท์ งานวิจัยนี้ได้เลือกใช้จำนวนโฆษณา 6 ขนาดเพื่อใช้ในการเปรียบเทียบ ได้แก่ 1,000 5,000 10,000 20,000 50,000 และ 100,000 โฆษณา โดยจะทดสอบสถาปัตยกรรม 3 ชนิด ได้แก่ Middle Agent, A4 และ DMAA ในระบบ Middle Agent จะมีแมทช์เมคกิ้งเอเจนท์ 1 ตัวให้บริการจับคู่ ส่วนระบบ A4 และ DMAA นั้นใช้จำนวนแมทช์เมคกิ้งเอเจนท์หลายๆ ตัวช่วยกันทำงาน โดยมีการกำหนด 4 แบบคือ 13, 21, 31 และ 43 ตัว สาเหตุที่ทำให้ระบบแมทช์เมคกิ้งเอเจนท์มีจำนวนแมทช์เมคกิ้งเอเจนท์ดังกล่าว เกิดจากการกำหนดให้ระบบแมทช์เมคกิ้งเอเจนท์ที่ใช้สถาปัตยกรรมแบบ A4 มีความลึกคงที่เท่ากับ 2 โดยที่ node ใน A4 มีจำนวน child node เท่ากับ 3 ทำให้จำนวน node ทั้งหมดเป็น 13 ดังรูปที่ 6.2 (ก) แต่ถ้าแต่ละ node ใน A4 มี child node 4 node ก็จะทำให้จำนวน node ทั้งหมดเป็น 21 ดังรูปที่ 6.3 (ก) และถ้า node ใน A4 มี child node เท่ากับ 5 ก็จะทำให้จำนวน node ทั้งหมดเป็น 31 ดังรูปที่ 6.4 (ก) ทั้งนี้ความลึกของระบบ A4 นั้นไม่สามารถน้อยกว่า 2 ได้เนื่องจากจะทำให้สถาปัตยกรรมไม่ครบถ้วน กล่าวคือไม่มี Coordinator จะมีแต่ Broker และแมทช์เมคกิ้งเอเจนท์ปกติเท่านั้น ดังนั้น ความลึกที่เลือกใช้ทดสอบจึงเป็นกรณีที่ดีที่สุดของระบบ A4 (ถ้าความลึกยิ่งมากจะยิ่งทำให้ห่วงโซ่ของการส่งต่อคำร้องยาวขึ้นเรื่อยๆ เหมือนกับระบบ Lattice Agent) ทั้งนี้ ในระบบแมทช์เมคกิ้งเอเจนท์ทั้งสองแบบมีการแบ่งประเภทการทำงานของแมทช์เมคกิ้งเอเจนท์ดังตารางที่ 6.1 และรูปที่ 6.2 ถึง 6.5

ทั้งสามสถาปัตยกรรมใช้วิธีการจับคู่แบบเดียวกันคือ Subsumption Matching ดังที่ได้อธิบายในหัวข้อ 5.4

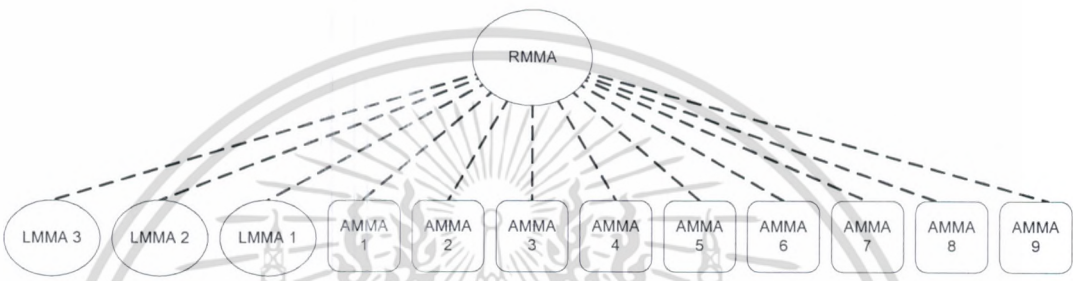
ตารางที่ 6.1 สรุปจำนวนแมทช์เมคกิ้งเอเจนท์แต่ละประเภทในแต่ละกรณีทดสอบ

	จำนวน MMA ทั้งหมดในระบบ	จำนวน Broker	จำนวน Coordinator	จำนวน MMA
A4	13	1	3	9
	21	1	4	16
	31	1	5	25
	43	1	6	36
	จำนวน MMA ทั้งหมดในระบบ	จำนวน Broker	จำนวน Coordinator	จำนวน MMA
DMAA	13	1	9	3
	21	1	9	11
	31	1	9	21
	43	1	9	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

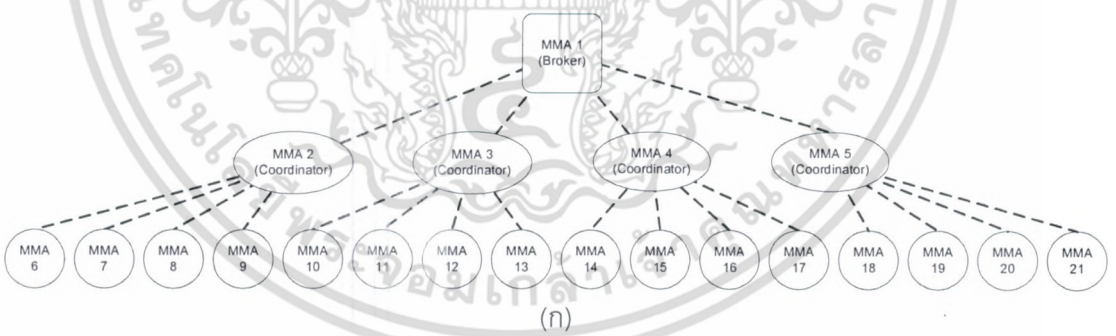


(ก)

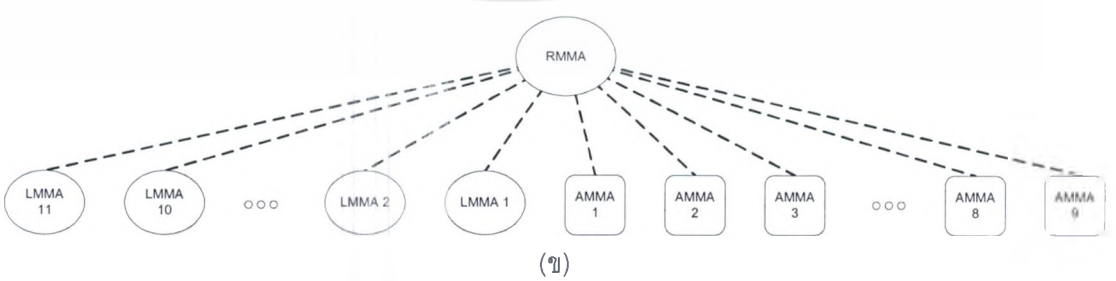


(ข)

รูปที่ 6.2 โครงสร้างของระบบแมทช์เมคกิ้งเอเจนท์ที่ใช้ในกรณีทดสอบ (ก) ใน A4-13 และ (ข) ใน DMAA-13



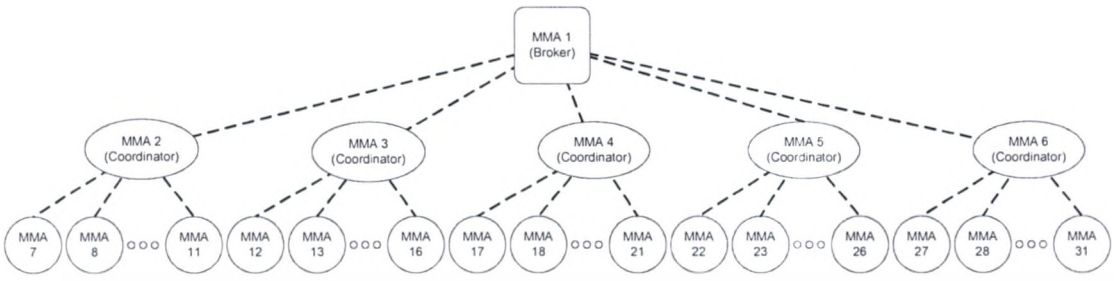
(ก)



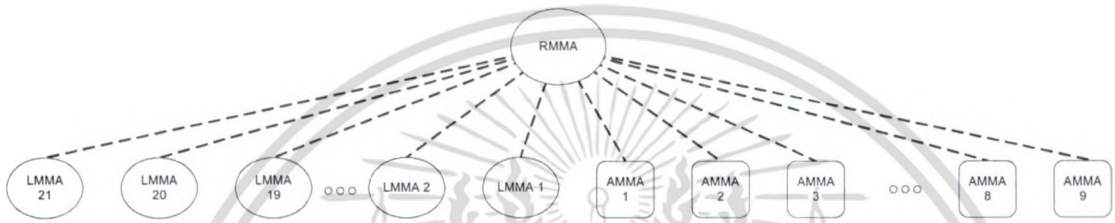
(ข)

รูปที่ 6.3 โครงสร้างของระบบแมทช์เมคกิ้งเอเจนท์ที่ใช้ในกรณีทดสอบ (ก) ใน A4-21 และ (ข) ใน DMAA-21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



(ข)

รูปที่ 6.4 โครงสร้างของระบบแมทช์เมคกิ้งเอเจนท์ที่ใช้ในกรณีทดสอบ (ก) ใน A4-31 และ (ข) ใน DMAA-31

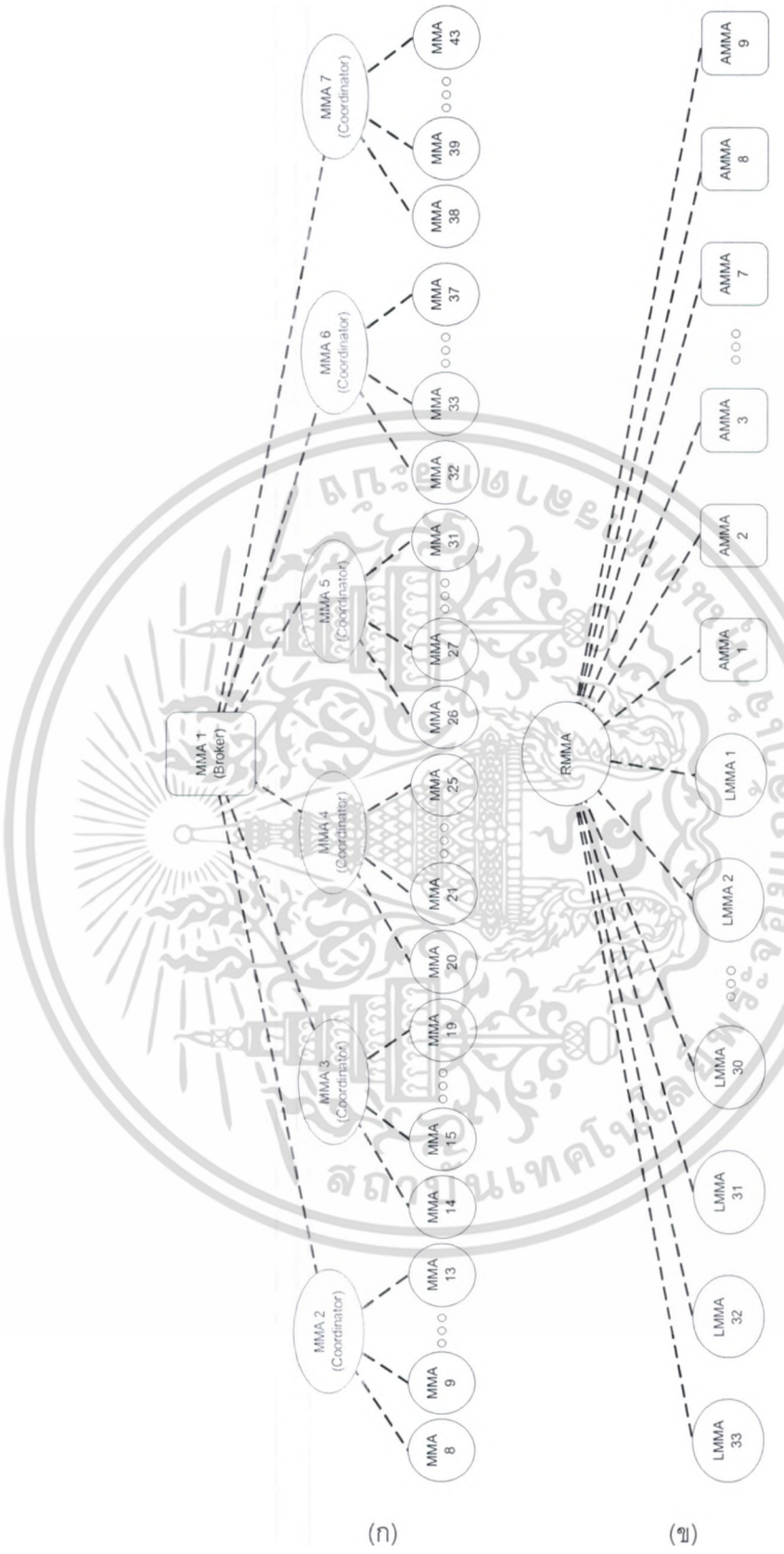
6.1.4 ข้อมูลที่ใช้ทดสอบ

ข้อมูลที่ใช้ทดสอบระบบแบ่งเป็น 3 ส่วนได้แก่

- โฆษณาความสามารถ (จากเอเจนท์ผู้ให้บริการ)
- คำร้องขอรับบริการที่ต้องการ (จากเอเจนท์ผู้ร้องขอ)
- ฐานข้อมูลแสดงความสัมพันธ์ระหว่างกลุ่มของคำศัพท์ (ในแมทช์เมคกิ้งเอเจนท์)

โฆษณาความสามารถของเอเจนท์ผู้ให้บริการที่ไม่ซ้ำกันจำนวน 100,000 เรคคอร์ด ได้ถูกสร้างขึ้นมาเพื่อใช้ในการทดสอบโดยใช้รูปแบบแสดงความรู้ตามหัวข้อที่ 5.3 เช่นเดียวกับคำร้องขอรับบริการ โดยตัวอย่างบางส่วนได้แสดงไว้ในภาคผนวก ข. สำหรับส่วนสุดท้ายนั้นจะได้เลือกใช้ข้อมูลจาก WordNet 1.6 ซึ่งเป็นฐานข้อมูลคำศัพท์ภาษาอังกฤษระดับแสนคำและได้รับความนิยมนำไปประยุกต์ใช้กับงานด้านปัญญาประดิษฐ์จำนวนมาก (CSL. 2002)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.5 โครงสร้างของระบบแมทช์เมคกิ้งเอเจนท์ที่ใช้ในกรณีทดสอบ (ก) ใน A4-43 และ (ข) ใน DMAA-43

ในแต่ละกรณีทดสอบที่มีจำนวนโฆษณาความสามารถเท่ากันนั้น (เช่น A4-13, DMAA-13 ที่มีขนาดของฐานความรู้ความสามารถเท่ากันคือ 10,000 โฆษณา เป็นต้น) ระบบแมทช์เมคกิ้งเอเจนท์จะจัดเก็บโฆษณาความสามารถชุดเดียวกันไว้ นอกจากนี้ เอเจนท์ผู้ร้องขอยังจะส่งคำร้องขอรับบริการชุดเดียวกันด้วย สาเหตุที่เลือกใช้ข้อมูลทดสอบแบบนี้ก็เพื่อให้สามารถเปรียบเทียบผลการทดสอบระบบของแต่ละกรณีทดสอบที่มีจำนวนโฆษณาความสามารถอยู่ภายในระบบเท่ากันแต่มีคอนฟิกเกอเรชั่นต่างกันได้ เนื่องจากแมทช์เมคกิ้งเอเจนท์ที่ใช้ในการทดสอบนั้นใช้กลไกการจับคู่เหมือนๆ กันไม่ว่าจะเป็นระบบแมทช์เมคกิ้งเอเจนท์ประเภทไหนก็ตาม (ดูบทที่ 5) นั่นเอง อย่างไรก็ตามโครงสร้างของแมทช์เมคกิ้งเอเจนท์สำหรับระบบแบบต่างๆ นั้นอาจจะทำให้การตอบกลับผลลัพธ์แตกต่างกันได้บ้าง ดังนั้นจึงได้ทำการทดสอบเปรียบเทียบความแม่นยำในการให้บริการจับคู่ของระบบแมทช์เมคกิ้งเอเจนท์แบบต่างๆ ขึ้นมา โดยเลือกทดสอบที่กรณีทดสอบ A4-13 และ DMAA-13 สำหรับขนาดโฆษณาเท่ากับ 10,000 โฆษณา การทดสอบจะกระทำโดยการใช้ข้อมูลชุดเดียวกันไปทดสอบด้วย Middle Agent เสียก่อนเพื่อจัดเก็บว่าคำร้องขอรับบริการหนึ่งๆ ควรจะเข้าคู่กับโฆษณาความสามารถใด ผลลัพธ์ที่ได้นั้นเป็นผลที่ไม่มีปัจจัยทางด้านการจัดเรียงโครงสร้างของแมทช์เมคกิ้งเอเจนท์เข้ามาเกี่ยวข้อง เนื่องจากเป็นการนำแมทช์เมคกิ้งเอเจนท์ตัวเดียวมาให้บริการ จากนั้นจึงนำข้อมูลดังกล่าวไปทดสอบกับระบบ A4-13 และ DMAA-13 และนำผลลัพธ์ที่ได้มาเปรียบเทียบกับผลลัพธ์จากการทดสอบด้วย Middle Agent อีกทีหนึ่ง

จากการทดสอบพบว่า ระบบ A4-13 นั้นสามารถให้บริการจับคู่ที่เหมือนกับระบบ Middle Agent ได้ 99,343 โฆษณา ในขณะที่ระบบ DMAA-13 นั้นสามารถให้บริการจับคู่ที่เหมือนกับระบบ Middle Agent ได้ 99,312 โฆษณา กล่าวคือระบบทั้งสองสามารถให้บริการจับคู่ที่เหมือนกับระบบ Middle Agent ได้มากกว่าร้อยละ 99 นั่นเอง ดังนั้นเราจึงสามารถสรุปได้ว่าระบบแมทช์เมคกิ้งเอเจนท์ทั้งสองประเภทนี้ทำให้ความแม่นยำในการจับคู่ลดลงเพียงเล็กน้อยเท่านั้น โดยให้ความแม่นยำในการจับคู่ที่ใกล้เคียงกันและใกล้เคียงกับการใช้แมทช์เมคกิ้งเอเจนท์เพียงตัวเดียว

6.1.5 ปริมาณงาน

ปริมาณงาน (Workload) ในการทดสอบนี้ได้แก่ จำนวนคำร้องขอรับบริการที่ส่งเข้ามายังระบบนั่นเอง การทดสอบจะกระทำโดยส่งคำร้องขอรับบริการมายังระบบแมทช์เมคกิ้งเอเจนท์เรื่อยๆ จนถึงจำนวนที่กำหนด (100,000 คำร้อง) ก็จะหยุดการส่ง โดยระยะเวลาระหว่างการส่งแต่ละครั้งนั้นจะมีการกระจายแบบเอ็กซ์โปเน็นเชียล

ทั้งนี้ คำร้องขอรับบริการที่จะถูกส่งเข้ามายังระบบแมทช์เมคกิ้งเอเจนท์นั้น จะเป็นคำร้องที่เหมือนกันทั้งหมดในทุกๆ กรณีทดสอบ ส่วนโฆษณาความสามารถนั้นจะเหมือนกันสำหรับกรณีทดสอบที่มีขนาดของโฆษณาเท่ากันไม่ว่าจะเป็นระบบประเภทใดก็ตาม ทั้งนี้เพื่อให้สามารถเปรียบเทียบกันระหว่างกรณีทดสอบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.6 สมมติฐานของการทดสอบ

การทดสอบระบบในงานวิจัยนี้มีสมมติฐานที่สำคัญ ได้แก่

- การทดสอบในงานวิจัยนี้ได้เลือกใช้เครื่องคอมพิวเตอร์เพียงเครื่องเดียวในการรันระบบแมทช์เมคกิ้งเอเจนท์ ทั้งนี้เพื่อให้ระบบแมทช์เมคกิ้งเอเจนท์ทั้งสามแบบมีทรัพยากรการคำนวณให้ใช้งานได้เท่าๆ กัน
- ในงานวิจัยนี้ได้ตัดส่วนของการวัดทรัพยากรเครือข่ายออกไป และสมมติให้แมทช์ เมคกิ้งเอเจนท์แต่ละตัวรู้จักวิธีติดต่อกับแมทช์เมคกิ้งเอเจนท์อื่นที่เกี่ยวข้องกับการทำงานของตนเองเสมอ ไม่ว่าจะเป็นระบบ DMAA หรือ A4 ก็ตาม
- สมมติให้ระบบสามารถดำเนินการไปได้ตลอดโดยไม่ติดขัด และไม่มีการสูญหายของข้อมูลที่ส่งผ่านระหว่างเอเจนท์
- ในความเป็นจริงแล้ว เอเจนท์อาจจะเปลี่ยนที่อยู่ได้ โดยเฉพาะเอเจนท์ประเภทเอเจนท์เคลื่อนที่ แต่ในการทดสอบจะไม่คำนึงถึงปัจจัยดังกล่าว โดยสมมติให้เอเจนท์ทั้งหมดที่มีในระบบมีที่อยู่คงที่

ข้อควรระวังของการนำระบบ DMAA ที่นำเสนอไปประยุกต์ใช้งาน คือประสิทธิภาพของระบบจะถูกจำกัดโดยความสามารถของ RMMA อย่างไรก็ตาม สถาปัตยกรรม DMAA นั้นได้ระบุให้ RMMA มีหน้าที่ในการจำแนกคำร้องเพื่อส่งต่อไปยัง AMMA ที่เหมาะสมเท่านั้น ซึ่งเป็นกระบวนการทำงานที่ใช้เวลาน้อยกว่ากระบวนการจับคู่ ดังนั้นความเร็วในการทำงานของ RMMA จะสูงกว่าแมทช์เมคกิ้งเอเจนท์ประเภทอื่น และทำให้ระบบมีประสิทธิภาพมากขึ้น

ข้อควรระวังประการที่สองของระบบนี้คือ การแบ่งขอบเขตความรับผิดชอบของแต่ละ AMMA หากการแบ่งขอบเขตไม่ชัดเจนแล้วจะทำให้ RMMA อาจส่งต่อคำร้องขอรับบริการบางส่วนไปยัง AMMA ที่ไม่เหมาะสมได้

6.2 ขั้นตอนการทดสอบ

การทดสอบระบบแบ่งเป็น 3 ขั้นตอนได้แก่ เตรียมข้อมูลและระบบที่จะทดสอบ ดำเนินการทดสอบระบบ และรวบรวมผลลัพธ์และสรุปผล

6.2.1 ขั้นเตรียมข้อมูลและระบบที่จะทดสอบ

หลังจากที่ได้สร้างระบบแมทช์เมคกิ้งเอเจนท์แบบต่างๆ ขึ้นมาแล้ว ขั้นต่อมาคือการเตรียมข้อมูลที่จะใช้ในการทดสอบระบบนั่นเอง โดยข้อมูลแรกที่จะถูกจัดเตรียมได้แก่โพษณาและคำร้องขอรับบริการของเอเจนท์ผู้ให้บริการและเอเจนท์ผู้ร้องขอตามลำดับ

ในขั้นแรกโพษณาความสามารถที่ไม่ซ้ำกันจำนวน 100,000 โพษณาได้ถูกสร้างขึ้น จากนั้นจึงนำข้อมูลดังกล่าวไปจัดให้อยู่ในรูปแบบทีเอเจนท์ผู้ให้บริการซึ่งจำลองขึ้น สามารถอ่านไปใช้ในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดสอบได้ ในขั้นต่อมา จะนำโฆษณาที่ผ่านการจัดรูปแบบมาแล้วมาเรียงลำดับใหม่แบบสุ่ม และขั้นตอนสุดท้ายคือการแบ่งโฆษณาให้กับเอเจนท์ผู้ให้บริการแต่ละตัวจัดเก็บไว้ ส่วนการเตรียมคำร้องขอรับบริการนั้นมีขั้นตอนเหมือนกับการเตรียมโฆษณาความสามารถทุกประการ

หลังจากเตรียมข้อมูลโฆษณาและคำร้องเสร็จสิ้นแล้ว ขั้นตอนต่อไปคือการเตรียมไฟล์โครงแบบ (Configuration file) ของระบบแมทช์เมคกิ้งเอเจนท์ที่จะทดสอบอันได้แก่ ไฟล์ A4-13, A4-21, A4-31, DMAA-13, DMAA-21, DMAA-31 และ Middle Agent รวมถึงยังต้องกำหนดโหมดการจับคู่ให้แก่แต่ละแมทช์เมคกิ้งเอเจนท์ภายในระบบด้วย คอนฟิกเกอเรชั่นไฟล์ที่สำคัญที่ต้องจัดเตรียมก่อนการทดสอบระบบทุกครั้งได้แก่ไฟล์ Type.nfo, ไฟล์ Domain.nfo, ไฟล์ PAList.nfo, ไฟล์ RAList.nfo และไฟล์ MMAList.nfo ไฟล์ Type.nfo จะระบุประเภทของระบบแมทช์เมคกิ้งเอเจนท์ที่ต้องการทดสอบ ในขณะที่ไฟล์ Domain.nfo จะระบุโดเมนที่ AMMA หนึ่งๆ รับผิดชอบ (ใช้เฉพาะกับระบบ DMAA เท่านั้น) ส่วนไฟล์ PAList.nfo จะระบุกลุ่มของเอเจนท์ผู้ให้บริการที่ต้องการรันในการทดสอบ ในขณะที่ไฟล์ RAList.nfo จะระบุกลุ่มของเอเจนท์ผู้ขอรับบริการที่ต้องการรันในการทดสอบรวมถึงระบุกลุ่มเอเจนท์ที่เอเจนท์ดังกล่าวสังกัดอีกด้วย ไฟล์สุดท้ายได้แก่ MMAList.nfo จะระบุกลุ่มของแมทช์เมคกิ้งเอเจนท์ที่ต้องการรันรวมถึงกำหนดโครงสร้างของแมทช์เมคกิ้งเอเจนท์ในระบบดังกล่าวด้วย

6.2.2 ขั้นดำเนินการทดสอบระบบ

หลังจากจัดเตรียมข้อมูลและระบบเรียบร้อยแล้ว ขั้นต่อมาก็จะได้ทำการทดสอบระบบที่ได้จัดเตรียมไว้ โดยก่อนการทดสอบทุกครั้งจะต้องรีบูทเครื่องใหม่เพื่อให้ทรัพยากรทางการคำนวณที่รองรับมีความใกล้เคียงกันในทุกๆ ครั้งของการทดสอบ และทำการล้างฐานความรู้ความสามารถที่มีอยู่เดิมในแต่ละแมทช์เมคกิ้งเอเจนท์ด้วย

การดำเนินการทดสอบนั้นแบ่งเป็น 2 ช่วง ได้แก่ ช่วงการโฆษณาเบื้องต้น และช่วงการทำงานปกติ

ในช่วงการโฆษณาเบื้องต้น เอเจนท์ผู้ให้บริการจะโฆษณาความสามารถของตนเองไปยังระบบแมทช์เมคกิ้งเอเจนท์ล่วงหน้า ทั้งนี้เพื่อให้ระบบแมทช์เมคกิ้งเอเจนท์ที่มีอยู่เสมือนกับว่าทำงานมาระยะหนึ่งแล้ว และมีจำนวนโฆษณามากพอที่จะสามารถให้บริการแก่เอเจนท์ใดๆ ได้

ช่วงต่อมาได้แก่การปล่อยให้ระบบเอเจนท์ทำงานปกติ ในช่วงนี้เอเจนท์ผู้ร้องขอจะส่งคำร้องขอรับบริการเข้ามายังระบบแมทช์เมคกิ้งเอเจนท์ตามปกติ เมื่อแมทช์เมคกิ้งเอเจนท์ใดๆ ได้รับคำร้องขอแล้วก็จะดำเนินการตามกระบวนการจับคู่ที่ได้ถูกโปรแกรมเอาไว้ต่อไป

6.2.3 ขั้นรวบรวมผลลัพธ์และสรุปผล

หลังจากที่คำร้องขอรับบริการได้ถูกส่งเข้ามายังระบบแมทช์เมคกิ้งเอเจนท์ได้รับการจับคู่กับโฆษณาที่มีอยู่ และตอบกลับไปยังเอเจนท์ผู้ร้องขอทั้งหมดแล้ว เอเจนท์ทั้งหมดก็จะหยุดการทำงานของตัวเองลง โดยที่เอเจนท์ผู้ร้องขอจะเป็นผู้เขียนผลลัพธ์ที่ได้จากการทดลองลงในไฟล์ Performance.nfo

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นจึงใช้โปรแกรมสรุปผลเก็บรวบรวมข้อมูลจากไฟล์ดังกล่าวเพื่อนำมาสรุปขั้นสุดท้ายสำหรับแต่ละการทดสอบ

ทั้งนี้ เมื่อทำการทดสอบและรวบรวมผลลัพธ์เสร็จสิ้นสำหรับแต่ละกรณีทดสอบแล้ว ฐานความรู้ความสามารถของแมทช์เมคกิ้งเอเจนท์แต่ละตัวจะถูกล้างเพื่อเริ่มต้นใหม่อีกครั้งหนึ่ง

6.3 ผลการทดสอบ

จากการทดสอบจะสามารถสรุปผลการทดสอบได้ดังตารางที่ 6.2 ซึ่งสามารถแสดงด้วยกราฟได้ดังรูปที่ 6.6 หากกำหนดให้อัตราการส่งคำร้องเข้ามายังระบบแมทช์เมคกิ้งเอเจนท์แต่ละประเภทเท่ากับ 1 คำร้องต่อนาทีแล้ว เวลาที่ใช้ในการตอบสนองเฉลี่ยต่อคำร้องสำหรับแต่ละกรณีทดสอบจะมีค่าดังตารางที่ 6.3

ตารางที่ 6.2 อัตราการให้บริการของระบบแมทช์เมคกิ้งเอเจนท์ (Service Rate)

CKB Size	A4-13	A4-21	A4-31	A4-43	DMAA-13	DMAA-21	DMAA-31	DMAA-43	Middle Agent
1,000	17.3	24.8	17.2	15.9	36.3	45.7	41.1	33.2	40.0
5,000	4.8	6.9	5.2	3.5	14.1	21.1	17.2	16.4	7.5
10,000	2.3	2.4	2.3	2.0	4.6	7.5	7.4	7.0	3.0
20,000	1.2	1.4	1.3	1.1	3.7	6.2	7.4	5.9	1.6
50,000	0.2	0.2	0.2	0.1	0.9	2.6	2.6	2.4	0.3
100,000	n.a.	n.a.	n.a.	n.a.	0.6	1.1	1.2	1.1	0.02

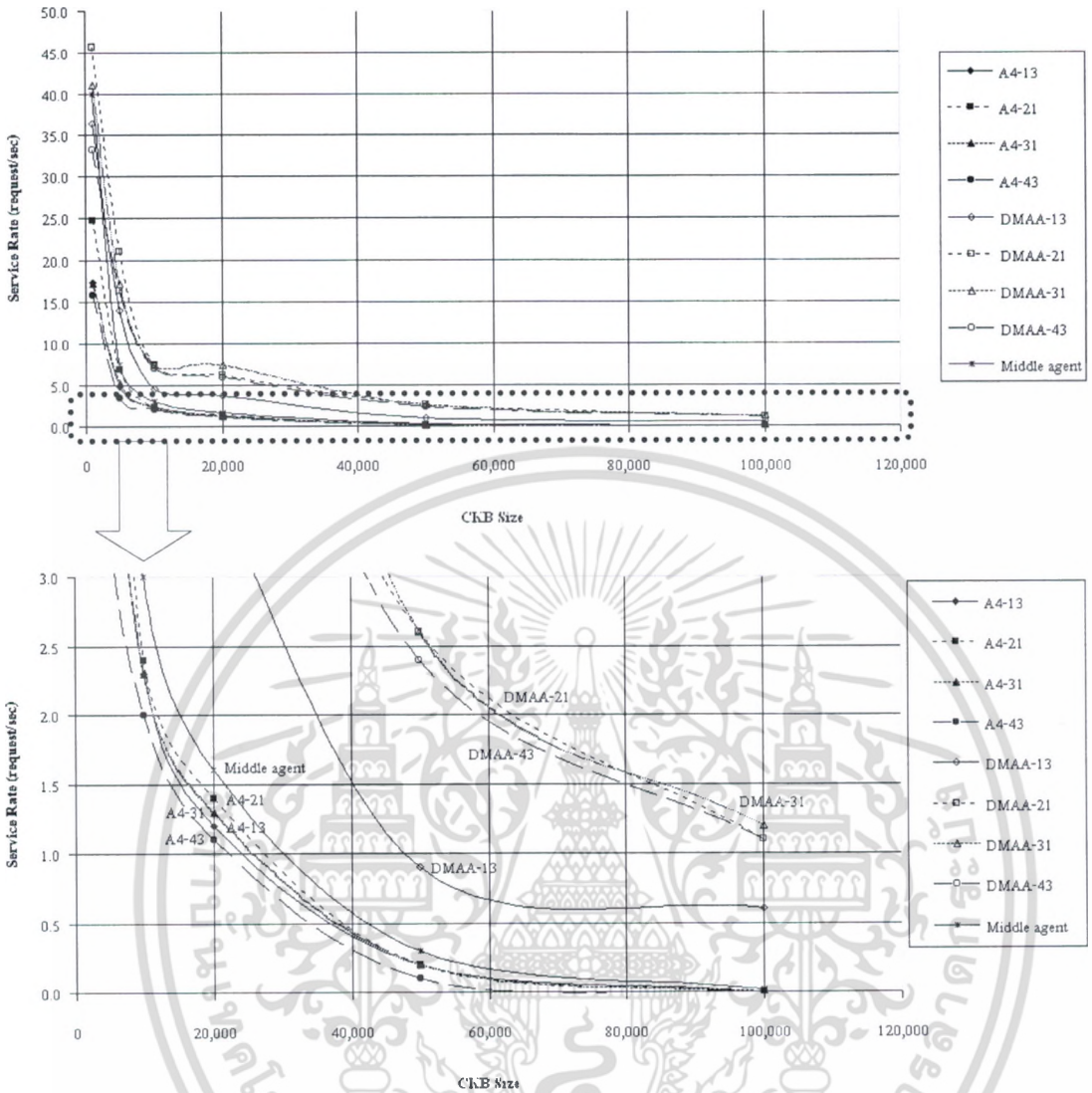
หน่วย: คำร้อง/วินาที

ตารางที่ 6.3 เวลาตอบสนองเฉลี่ยต่อคำร้อง (Response Time)

CKB Size	A4-13	A4-21	A4-31	A4-43	DMAA-13	DMAA-21	DMAA-31	DMAA-43	Middle Agent
1,000	0.058	0.040	0.058	0.063	0.028	0.022	0.024	0.030	0.025
5,000	0.209	0.145	0.193	0.287	0.071	0.047	0.058	0.061	0.134
10,000	0.438	0.420	0.438	0.504	0.218	0.134	0.135	0.143	0.335
20,000	0.845	0.723	0.779	0.923	0.271	0.162	0.135	0.170	0.632
50,000	5.455	5.455	5.455	12.000	1.132	0.387	0.387	0.420	3.529
100,000	n.a.	n.a.	n.a.	n.a.	1.714	0.923	0.845	0.923	300.000

หน่วย: วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.6 ความสัมพันธ์ระหว่างอัตราการให้บริการและจำนวนโฆษณาความสามารถภายในระบบ

จากผลลัพธ์โดยรวมพบว่าสถาปัตยกรรม DMAA ที่นำเสนอในงานวิจัยนี้มีประสิทธิภาพสูงกว่าสถาปัตยกรรมแบบ A4 โดยสามารถให้บริการคำร้องได้มากกว่าในช่วงเวลาที่เท่ากันสำหรับในทุกๆ กรณีทดสอบ (ทุกขนาดของโฆษณาความสามารถ และทุกๆ จำนวนของแมทช์เมคกิ้งเอเจนท์ที่รันในการทดสอบ)

สาเหตุสำคัญที่ทำให้สถาปัตยกรรม DMAA มีประสิทธิภาพสูงกว่าสถาปัตยกรรม A4 คือการออกแบบโครงสร้างและการทำงานร่วมกันของแมทช์เมคกิ้งเอเจนท์ที่แตกต่างกันนั่นเอง โครงสร้างของแมทช์เมคกิ้งเอเจนท์ภายในระบบ DMAA นั้นจะประกอบด้วยลำดับชั้นของแมทช์ เมคกิ้งเอเจนท์ที่มีความลึกคงที่เท่ากับ 1 เสมอ หรือกล่าวคือประกอบด้วยแมทช์เมคกิ้งเอเจนท์ 2 ชั้นเสมอนั่นเอง ส่วนโครงสร้างของแมทช์เมคกิ้งเอเจนท์ภายในระบบ A4 นั้นจะประกอบด้วยลำดับชั้นของแมทช์เมคกิ้งเอเจนท์ตั้งแต่ 2 ชั้นขึ้นไป มิได้มีการจำกัดจำนวนชั้นเอาไว้ เมื่อพิจารณาถึงกระบวนการทำงานของการจับคู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของแต่ละระบบประกอบด้วยแล้วจะพบว่า เนื่องจากระบบ DMAA นั้นได้กำหนดให้ AMMA เป็นผู้ที่มีหน้าที่จับคู่ขั้นสุดท้าย (หาก AMMA ไม่สามารถจับคู่ให้กับคำร้องใดได้ แสดงว่าไม่มีเอเจนท์ผู้ให้บริการรายใดที่สามารถให้บริการซึ่งระบุอยู่ภายในคำร้องนั้นได้) ดังนั้น การส่งต่อคำร้องภายในแมทช์เมคกิ้งเอเจนท์ของระบบ DMAA นั้นจะส่งอย่างมากที่สุดเพียง 2 ครั้งเท่านั้น (ส่งจาก LMMA ไปยัง RMMA และส่งจาก RMMA ไปยัง AMMA) ในขณะที่สำหรับระบบ A4 นั้น คำร้องจะถูกส่งต่อขึ้นไปตามลำดับชั้นของแมทช์เมคกิ้งเอเจนท์เรื่อยๆ จนกว่าจะพบโฆษณาที่เข้าคู่กับคำร้องดังกล่าว โดยอาจจะต้องส่งต่อคำร้องเป็นจำนวนครั้งเท่ากับความลึกของลำดับชั้นของแมทช์เมคกิ้งเอเจนท์ทีเดียว (โครงสร้างของระบบ A4 ได้บังคับให้เอเจนท์ที่อยู่ในระดับสูงสุด หรือ Broker เป็นผู้ให้บริการจับคู่ขั้นสุดท้ายของระบบประเภทนี้) ดังนั้นในแง่ของจำนวนครั้งของการติดต่อซึ่งกันและกันระหว่างแมทช์เมคกิ้งเอเจนท์แล้ว ระบบแบบ A4 ซึ่งใช้จำนวนการติดต่อที่มากกว่าระบบแบบ DMAA นั้นจึงสูญเสียประสิทธิภาพการทำงานของระบบไปกับการส่งต่อข้อมูลระหว่างแมทช์เมคกิ้งเอเจนท์ภายในระบบนั่นเอง

ในแง่ของการโฆษณาความสามารถ ระบบ A4 นั้นได้กำหนดให้แมทช์เมคกิ้งเอเจนท์ต้องโฆษณาความสามารถของเอเจนท์ผู้ให้บริการที่ตนเองทราบไปยังแมทช์เมคกิ้งเอเจนท์ที่อยู่ในระดับที่สูงกว่า ดังนั้น แมทช์เมคกิ้งเอเจนท์ที่อยู่ในระดับที่สูงๆ (Coordinator) จึงมีฐานความรู้ความสามารถที่มีขนาดใหญ่ขึ้นเรื่อยๆ เนื่องจากมิได้ทราบเฉพาะความสามารถของเอเจนท์ผู้ให้บริการที่โฆษณาไว้กับตนเองเท่านั้น หากแต่ทราบความสามารถของเอเจนท์ผู้ให้บริการที่โฆษณาไว้กับแมทช์เมคกิ้งเอเจนท์อื่นที่อยู่ในระดับที่ต่ำกว่าตนเองด้วย โดยเฉพาะแมทช์เมคกิ้งเอเจนท์ที่อยู่ในระดับสูงสุด (Broker) จะประกอบไปด้วยฐานความรู้ความสามารถที่จัดเก็บความสามารถของเอเจนท์ผู้ให้บริการที่โฆษณาเข้ามายังแมทช์เมคกิ้งเอเจนท์ทั้งหมดภายในระบบทีเดียวดังตารางที่ 6.4 และ 6.5 ในขณะที่สำหรับระบบ DMAA นั้น แมทช์เมคกิ้งเอเจนท์แต่ละตัวจะทราบเฉพาะความสามารถของกลุ่มเอเจนท์ผู้ให้บริการที่ตนเองรับผิดชอบเท่านั้น โดยที่ LMMA จะทราบเฉพาะความสามารถของเอเจนท์ผู้ให้บริการที่อยู่ภายในระบบ MAS ที่ตนเองสังกัด และ AMMA จะทราบเฉพาะความสามารถที่เกี่ยวข้องกับโดเมนที่ตนเองรับผิดชอบให้บริการจับคู่เท่านั้น ดังนั้นกระบวนการจับคู่ในแต่ละแมทช์เมคกิ้งเอเจนท์ของระบบ DMAA นั้นจึงกระทำกับฐานความรู้ความสามารถที่มีขนาดเท่ากันหรือเล็กกว่าฐานความรู้ความสามารถเฉลี่ยของแต่ละแมทช์เมคกิ้งเอเจนท์ภายในระบบ A4 (ดูตารางที่ 6.4) ในการทดสอบระบบในงานวิจัยนี้ได้เลือกใช้อัลกอริทึมจับคู่เดียวกันสำหรับแต่ละแมทช์เมคกิ้งเอเจนท์ ทั้งสำหรับระบบ DMAA และ A4 การใช้อัลกอริทึมเดียวกันบนฐานความรู้ความสามารถที่มีขนาดที่ต่างกันย่อมจะทำให้ความเร็วในการจับคู่แตกต่างกัน โดยหากกระทำการจับคู่บนฐานความรู้ความสามารถที่มีขนาดใหญ่กว่า ก็จะทำให้ความเร็วในการจับคู่น้อยกว่าความเร็วในการจับคู่บนฐานความรู้ความสามารถที่มีขนาดเล็กกว่า ดังนั้นปัจจัยดังกล่าวจึงเป็นอีกปัจจัยหนึ่งที่ส่งผลให้ประสิทธิภาพการทำงานของระบบ A4 ลดลง

เมื่อเปรียบเทียบระบบ A4 และ DMAA กับ Middle Agent พบว่าระบบ A4 นั้นมีความรวดเร็วในการให้บริการจับคู่ต่ำกว่า Middle Agent ในทุกๆ กรณีทดสอบ เมื่อพิจารณาระบบ DMAA พบว่าในบางกรณีทดสอบได้แก่ กรณีทดสอบ DMAA-13 และ DMAA-43 ณ ขนาดของฐานความสามารถเท่ากับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1,000 โฆษณา ระบบ DMAA นั้นมีความเร็วในการให้บริการจับคู่ที่ต่ำกว่า Middle Agent อย่างไรก็ตาม สำหรับกรณีทดสอบอื่นที่เหลือทั้งหมด ระบบ DMAA สามารถให้บริการจับคู่ที่รวดเร็วกว่า Middle Agent มาก สาเหตุสำคัญของความแตกต่างนี้เกิดจากการที่ระบบ DMAA นั้นช่วยลดขนาดของฐานความสามารถที่ใช้ในการจับคู่แต่ละครั้งลง (ดูตารางที่ 6.5 ประกอบ) โดยที่ยังคงรักษาการจำกัดการส่งต่อคำร้องให้อยู่ภายในขอบเขตได้นั่นเอง

ตารางที่ 6.4 ขนาดฐานความรู้ความสามารถเฉลี่ยต่อแมทช์เมคกิ้งเอเจนท์ 1 ตัว

หน่วย: เมกกะไบต์

CKB Size	MMA Type	A4-13	A4-21	A4-31	A4-43	DMA A-13	DMA A-21	DMA A-31	DMA A-43	Middle Agent
1,000	Broker/AMMA	9.1	9.1	9.1	9.1	1.9	1.9	1.9	1.9	9.1
	Coordinator	3.5	3.1	2.7	2.3					
	MMA/LMMA	1.9	1.5	1.2	0.9	3.3	1.4	1.0	0.8	
5,000	Broker/AMMA	39.5	39.5	39.5	39.5	8.1	8.1	8.1	8.1	39.5
	Coordinator	15.1	11.9	9.8	8.4					
	MMA/LMMA	5.8	4.3	2.8	2.0	13.9	4.4	2.5	1.8	
10,000	Broker/AMMA	79.5	79.5	79.5	79.5	16.2	16.2	16.2	16.2	79.5
	Coordinator	26.8	20.9	17.3	14.7					
	MMA/LMMA	10.4	6.5	4.4	3.1	28.0	8.3	4.6	3.5	
20,000	Broker/AMMA	157.4	157.4	157.4	157.4	32.2	32.2	32.2	32.2	157.4
	Coordinator	56.3	43.5	35.2	29.4					
	MMA/LMMA	20.3	11.7	7.8	5.4	55.1	15.9	8.7	6.5	
50,000	Broker/AMMA	394.5	394.5	394.5	394.5	84.1	84.1	84.1	84.1	394.5
	Coordinator	132.2	101.1	81.4	68.1					
	MMA/LMMA	46.0	26.9	18.0	12.8	133.4	37.5	20.7	14.1	
100,000	Broker/AMMA	771.3	771.3	771.3	771.3	169.1	169.1	169.1	169.1	771.3
	Coordinator	260.2	200.1	162.1	135.4					
	MMA/LMMA	90.5	52.5	35.1	24.7	265.7	76.5	41.6	27.5	

หมายเหตุ: Broker/AMMA หมายถึง แมทช์เมคกิ้งเอเจนท์ที่เป็น Broker หรือ AMMA สำหรับระบบ A4 และ DMAA ตามลำดับ

Coordinator หมายถึง แมทช์เมคกิ้งเอเจนท์ที่เป็น Coordinator สำหรับระบบ A4

MMA/LMMA หมายถึง แมทช์เมคกิ้งเอเจนท์ปกติ หรือ LMMA สำหรับระบบ A4 และ DMAA ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.5 จำนวนความสามารถภายในฐานความรู้ความสามารถเฉลี่ยต่อแมทช์เมคกิ้งเอเจนท์ 1 ตัว

CKB Size	MMA Type	A4-13	A4-21	A4-31	A4-43	DMAA-13	DMAA-21	DMAA-31	DMAA-43	Middle Agent
1,000	Br/Am	1,000	1,000	1,000	1,000	111	111	111	111	1,000
	Co	308	238	194	163					
	Ag/Lm	77	48	32	23	333	91	48	30	
5,000	Br/Am	5,000	5,000	5,000	5,000	556	556	556	556	5,000
	Co	1,538	1,190	968	814					
	Ag/Lm	385	238	161	116	1,667	455	238	152	
10,000	Br/Am	10,000	10,000	10,000	10,000	1,111	1,111	1,111	1,111	10,000
	Co	3,077	2,381	1,935	1,628					
	Ag/Lm	769	476	323	233	3,333	909	476	303	
20,000	Br/Am	20,000	20,000	20,000	20,000	2,222	2,222	2,222	2,222	20,000
	Co	6,154	4,762	3,871	3,256					
	Ag/Lm	1,538	952	645	465	6,667	1,818	952	606	
50,000	Br/Am	50,000	50,000	50,000	50,000	5,556	5,556	5,556	5,556	50,000
	Co	15,385	11,905	9,677	8,140					
	Ag/Lm	3,846	2,381	1,613	1,163	16,667	4,545	2,381	1,515	
100,000	Br/Am	100,000	100,000	100,000	100,000	11,111	11,111	11,111	11,111	100,000
	Co	30,769	23,810	19,355	16,279					
	Ag/Lm	7,692	4,762	3,226	2,326	33,333	9,091	4,762	3,030	

หมายเหตุ: Br/Am หมายถึง แมทช์เมคกิ้งเอเจนท์ที่เป็น Broker หรือ AMMA สำหรับระบบ A4 และ DMAA ตามลำดับ

Co หมายถึง แมทช์เมคกิ้งเอเจนท์ที่เป็น Coordinator สำหรับระบบ A4

Ag/Lm หมายถึง แมทช์เมคกิ้งเอเจนท์ปกติ หรือ LMMA สำหรับระบบ A4 และ DMAA ตามลำดับ

6.4 สรุปการทดสอบ

จากการทดสอบระบบพบว่า สถาปัตยกรรม DMAA ที่นำเสนอนั้น สามารถให้ประสิทธิภาพการทำงานที่เหนือกว่าระบบแมทช์เมคกิ้งเอเจนท์แบบอื่นทั้งหมด ไม่ว่าจะมีจำนวนโฆษณาความสามารถอยู่ภายในระบบเป็นจำนวนเท่าใด หรือมีจำนวนแมทช์เมคกิ้งเอเจนท์ที่ให้บริการเท่าใดก็ตาม ดังนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาปัตยกรรมที่นำเสนอจึงจัดเป็นสถาปัตยกรรมที่มีประสิทธิภาพสูง และเหมาะสมกับการนำไปใช้งานกับระบบเอเจนท์ขนาดใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
สรุปผลการวิจัยและข้อเสนอแนะ

ในปัจจุบันได้มีแนวคิดในการใช้ตัวแทนเพื่อปฏิบัติหน้าที่แทนผู้ใช้เกิดขึ้น เรียกว่าเอเจนท์ การที่เอเจนท์หนึ่งๆ จะสามารถทำงานที่ได้รับมอบหมายมาโดยที่รบกวนผู้ใช้น้อยที่สุดนั้น เอเจนท์ดังกล่าวจะต้องมีความรู้และความสามารถเพียงพอ ในกรณีที่เอเจนท์มีความรู้หรือความสามารถไม่เพียงพอต่อการทำงานหนึ่งๆ ก่อนที่เอเจนท์ดังกล่าวจะขอความช่วยเหลือจากผู้ใช้ซึ่งเป็นมนุษย์หรือล้มเลิกความตั้งใจในการปฏิบัติการดังกล่าว เอเจนท์นั้นจะลองติดต่อขอความช่วยเหลือจากเอเจนท์ด้วยกันเองก่อน ดังนั้น เอเจนท์จึงจำเป็นต้องมีความสามารถในการติดต่อสื่อสารกับเอเจนท์อื่นๆ ด้วย อย่างไรก็ตาม ก่อนที่เอเจนท์หนึ่งๆ จะสามารถร้องขอความช่วยเหลือจากเอเจนท์อื่นได้ จำเป็นอย่างยิ่งที่จะต้องทราบก่อนว่ามีเอเจนท์ใดที่สามารถให้บริการที่ต้องการได้บ้าง ดังนั้น จึงได้มีงานวิจัยพัฒนาเอเจนท์ชนิดหนึ่งขึ้นเพื่อทำหน้าที่ค้นหาเอเจนท์ผู้ให้บริการที่สามารถทำงานที่เอเจนท์อื่นร้องขอมาได้ ซึ่งได้แก่แมทช์เมคกิ้งเอเจนท์นั่นเอง

เอเจนท์ที่มาติดต่อกับระบบแมทช์เมคกิ้งเอเจนท์จะแบ่งเป็น 2 ประเภท ได้แก่ เอเจนท์ผู้ขอรับบริการ และเอเจนท์ผู้ให้บริการ เริ่มแรกเมื่อเอเจนท์ผู้ให้บริการเข้าสู่ระบบก็จะลงทะเบียนกับแมทช์เมคกิ้งเอเจนท์และโฆษณาบริการที่ตนเองสามารถให้เอเจนท์อื่นได้ไปยังแมทช์เมคกิ้งเอเจนท์ซึ่งเรียกว่าโฆษณาความสามารถ ข้อมูลเหล่านี้จะถูกจัดเก็บอยู่ในฐานความรู้ความสามารถของแมทช์เมคกิ้งเอเจนท์ ดังกล่าว เมื่อมีเอเจนท์ผู้ขอรับบริการติดต่อเข้ามายังแมทช์เมคกิ้งเอเจนท์ แมทช์เมคกิ้งเอเจนท์จะทำการค้นหาฐานความรู้ความสามารถของตนเองเพื่อเปรียบเทียบว่ามีโฆษณาบริการของเอเจนท์ผู้ให้บริการรายใดที่สอดคล้องกับคำร้องขอบริการนั้น เมื่อสิ้นสุดการค้นหาแล้ว แมทช์เมคกิ้งเอเจนท์จะตอบกลับไปยังเอเจนท์ผู้ขอรับบริการว่ามีเอเจนท์ผู้ให้บริการรายใดบ้างที่สามารถให้บริการดังกล่าวได้ เพื่อให้เอเจนท์ผู้ขอรับบริการสามารถติดต่อขอรับบริการกับเอเจนท์ดังกล่าวได้โดยตรงต่อไป ดังนั้น แมทช์เมคกิ้งเอเจนท์จึงหมายถึงเอเจนท์ที่ทำหน้าที่ช่วยค้นหาเอเจนท์อื่นที่สามารถให้บริการแก่เอเจนท์ที่ร้องขอ โดยทั่วไปแล้วแมทช์เมคกิ้งเอเจนท์หนึ่งๆ จะสังกัดระบบเอเจนท์ระบบใดระบบหนึ่ง และจะให้บริการเฉพาะกับเอเจนท์ภายในระบบนั้น

องค์ประกอบหลักภายในแมทช์เมคกิ้งเอเจนท์หนึ่งๆ นั้นแบ่งเป็น 2 ส่วน ได้แก่ กลไกการจับคู่และโพรโทคอลแลกความรู้

กลไกการจับคู่เป็นอัลกอริทึมที่แมทช์เมคกิ้งเอเจนท์ใช้ในการเปรียบเทียบคำร้องขอบริการที่เข้ามากับโฆษณาความสามารถที่เก็บอยู่ในฐานความรู้ความสามารถ เพื่อค้นหาโฆษณาที่สอดคล้องกับคำร้องและสามารถตอบกลับไปได้ว่าเอเจนท์ผู้ร้องขอควรจะไปติดต่อขอรับบริการกับเอเจนท์ผู้ให้บริการรายใด สำหรับโพรโทคอลแลกความรู้เป็นโพรโทคอลที่ใช้ในการติดต่อสื่อสารแลกเปลี่ยนความรู้ระหว่างเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เจนท์ โดยโพรโทคอลที่ได้รับความนิยมและกลายมาเป็นมาตรฐานหลักของเทคโนโลยีเอเจนท์ได้แก่โพรโทคอล KQML (Labrou and Finin. 1999)

ในปัจจุบันได้มีผู้พัฒนาระบบแมทช์เมคกิ้งเอเจนท์ขึ้นมาบ้างแล้วได้แก่ระบบ Expert Finder (Vivacqua. 1999) ระบบ Middle Agent (Sycara *et al.* 1999) ระบบ A4 (Cao *et al.* 2000) และระบบ Lattice Agents (Shehory. 2000)

ระบบ Expert Finder ประกอบไปด้วยแมทช์เมคกิ้งเอเจนท์หลายๆ ตัวมาทำงานรวมกัน โดยที่แมทช์เมคกิ้งเอเจนท์แต่ละตัวจะมีฐานความรู้ความสามารถเหมือนกันหมดทั้งระบบ กล่าวคือ แมทช์เมคกิ้งเอเจนท์หนึ่งๆ จะมีโฆษณาความสามารถของเอเจนท์ผู้ให้บริการทุกตัวในระบบนั้นนั่นเอง คุณสมบัตินี้ทำให้ระบบ Expert Finder มีข้อเสียที่สำคัญคือเกิดปัญหาในเรื่องของปริมาณการอัพเดตข้อมูลความสามารถที่เปลี่ยนแปลงไปยังแมทช์เมคกิ้งเอเจนท์แต่ละตัว ดังนั้นต่อมาจึงได้มีผู้นำเสนอระบบแมทช์เมคกิ้งเอเจนท์แบบ Middle Agent ขึ้นเพื่อแก้ไขปัญหาดังกล่าว โดยรวมฐานความรู้ความสามารถที่กระจายให้มาอยู่ที่เอเจนท์เพียงตัวเดียวเพื่อทำหน้าที่เป็นแมทช์เมคกิ้งเอเจนท์โดยเฉพาะ เอเจนท์ดังกล่าวเรียกว่า Middle Agent ระบบนี้จะช่วยลดจำนวนการติดต่อระหว่างเอเจนท์เพื่ออัพเดตความสามารถที่เปลี่ยนแปลงลง อย่างไรก็ตาม วิธีนี้ได้ก่อให้เกิดข้อเสียคือผู้ให้บริการ (จับคู่) มีจำนวนน้อยกว่าผู้ขอรับบริการมาก จึงทำให้เกิดปัญหา Scalability ขึ้น ดังนั้นจึงได้มีแนวคิดที่จะกระจายการให้บริการจับคู่ออกไปให้กับเอเจนท์หลายๆ ตัวเหมือนเดิม แต่จะต้องแก้ไขปัญหาการอัพเดตความสามารถที่เปลี่ยนแปลงไปด้วย แนวคิดดังกล่าวได้แก่ระบบ Lattice Agents นั่นเอง หลักการของระบบนี้คือนำหน้าที่ในการจับคู่ไปไว้ที่ตัวแมทช์เมคกิ้งเอเจนท์หลาย ๆ ตัวโดยที่แต่ละตัวไม่จำเป็นที่จะต้องเก็บข้อมูลเกี่ยวกับเอเจนท์ทั้งหมดที่มีอยู่ในระบบ แต่จะมีเพียงฐานความรู้ความสามารถของเอเจนท์เพียงบางส่วนของระบบเท่านั้น เมื่อต้องการค้นหาความสามารถที่ตนเองไม่สามารถให้บริการได้ ก็จะไปติดต่อกับแมทช์เมคกิ้งเอเจนท์ข้างเคียงเพื่อขอความช่วยเหลือต่อไป อย่างไรก็ตาม ระบบนี้ได้สร้างปัญหาใหม่กล่าวคือการค้นหาบริการหนึ่งๆ จะต้องถูกค้นหาหลายครั้งในแมทช์เมคกิ้งเอเจนท์หลายตัวตามเส้นทางที่คำร้องขอรับบริการถูกส่งไป ทำให้สิ้นเปลืองทรัพยากรในการติดต่อสื่อสารโดยไม่จำเป็น ใช้เวลามากกว่าระบบก่อนๆ ในการให้บริการจับคู่ รวมถึงไม่สามารถกำหนดได้ว่าการส่งต่อคำร้องไปยังแมทช์เมคกิ้งเอเจนท์อื่นๆ นั้นจะสิ้นสุดเมื่อใด ดังนั้นต่อมาจึงได้มีผู้นำระบบดังกล่าวมาปรับปรุงโดยอาศัยการจัดโครงสร้างแบบลำดับชั้นให้แก่ระบบแมทช์เมคกิ้งเอเจนท์และเกิดเป็นระบบ A4 ขึ้น ระบบนี้อาศัยการจำกัดความรับผิดชอบให้แก่แมทช์เมคกิ้งเอเจนท์แต่ละตัว แมทช์เมคกิ้งเอเจนท์ที่อยู่ในระดับที่สูงกว่าจะมีความรับผิดชอบต่อการให้บริการจับคู่ที่สูงกว่าแมทช์เมคกิ้งเอเจนท์ที่อยู่ในระดับที่ต่ำกว่า กล่าวคือ หากแมทช์เมคกิ้งเอเจนท์ที่อยู่ในระดับที่ต่ำกว่าไม่สามารถให้บริการจับคู่แก่คำร้องขอรับบริการจับคู่ใดได้ ก็จะส่งต่อคำร้องดังกล่าวไปยังแมทช์เมคกิ้งเอเจนท์ที่อยู่ในระดับที่สูงกว่าตนเองหนึ่งระดับ และคำร้องดังกล่าวจะถูกส่งต่อขึ้นไปเรื่อยๆ จนกว่าจะพบบริการที่เหมาะสม หรือจนถึงแมทช์เมคกิ้งเอเจนท์ที่อยู่ในระดับสูงสุด ระบบนี้สามารถจำกัดจำนวนการติดต่อระหว่างแมทช์เมคกิ้งเอเจนท์เพื่อให้บริการจับคู่หนึ่งๆ ให้อยู่ในวงจำกัดได้ แต่จะมีข้อเสียคือจำนวนการอัพเดตความสามารถที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนแปลงไปจะสูงขึ้น ทั้งนี้เป็นผลมาจากการที่ความสามารถที่เปลี่ยนแปลงไปจะต้องถูกโฆษณาขึ้นไปตามลำดับชั้นเรื่อยๆ จนถึงแมทช์เมคกิ้งเอเจนท์ตัวบนสุดนั่นเอง

จากที่ได้กล่าวมาข้างต้นจะพบว่า เทคโนโลยีแมทช์เมคกิ้งเอเจนท์นั้นจะมีพัฒนาการไปในทิศทางเพื่อเพิ่มประสิทธิภาพให้แก่ระบบแมทช์เมคกิ้งเอเจนท์ขึ้นเรื่อยๆ ดังนั้นปัญหาหลักของเทคโนโลยีดังกล่าวในปัจจุบันก็คือประสิทธิภาพการให้บริการที่ต่ำนั่นเอง งานวิจัยนี้มุ่งเน้นที่จะนำเสนอระบบแมทช์เมคกิ้งเอเจนท์ที่มีประสิทธิภาพการให้บริการสูงขึ้นจากระบบที่ได้นำเสนอในอดีต ทั้งนี้ เพื่อให้เทคโนโลยีนี้มีความใกล้เคียงมากขึ้นกับการนำไปใช้ได้จริงในระบบเอเจนท์ขนาดใหญ่

แนวทางการแก้ไขปัญหาประสิทธิภาพของแมทช์เมคกิ้งเอเจนท์มีอยู่หลายทางเลือก แต่เราสามารถแบ่งได้เป็น 2 กลุ่มได้แก่ การปรับปรุงภายใน และการปรับปรุงภายนอก การปรับปรุงภายในที่สำคัญคือการปรับปรุงองค์ประกอบภายในหลักของแมทช์เมคกิ้งเอเจนท์ ซึ่งได้แก่การปรับปรุงอัลกอริทึมที่ใช้ในกลไกการจับคู่ให้มีประสิทธิภาพสูงขึ้นนั่นเอง อย่างไรก็ตาม แมทช์เมคกิ้งเอเจนท์ที่มีอยู่ในปัจจุบันนั้น มักจะมีความสามารถในการจับคู่หลายๆ แบบ การที่แมทช์เมคกิ้งเอเจนท์หนึ่งๆ นำวิธีการจับคู่หลากหลายวิธีมาใช้ จะทำให้โดยธรรมชาติแล้วแมทช์เมคกิ้งเอเจนท์นั้นใช้เวลาค่อนข้างนานสำหรับการจับคู่โฆษณาความสามารถที่เหมาะสมให้กับคำร้องขอรับบริการหนึ่งๆ ซึ่งทำให้เป็นการยากที่จะฝืนธรรมชาติ

ส่วนการปรับปรุงภายนอกที่สำคัญจะได้แก่การนำแมทช์เมคกิ้งเอเจนท์หลายตัวมาทำงานร่วมกัน แนวทางนี้เป็นแนวทางที่น่าสนใจและมีงานวิจัยที่เกี่ยวข้องเกิดขึ้นมาแล้ว ซึ่งได้แก่ระบบแมทช์-เมคกิ้งเอเจนท์ทั้ง 4 ประเภทที่ได้กล่าวมาข้างต้น เมื่อพิจารณางานเหล่านั้นแล้วพบว่า ล้วนมุ่งที่จะนำเสนอแนวคิดที่สำคัญคือพยายามพัฒนาโครงสร้างของกลุ่มแมทช์เมคกิ้งเอเจนท์ที่คำนึงถึงความสอดคล้องกับลักษณะเฉพาะของเอเจนท์ให้มากที่สุด และเป็นโครงสร้างที่ช่วยลดหรือจำกัดจำนวนการติดต่อระหว่างแมทช์เมคกิ้งเอเจนท์ในการให้บริการคำร้องขอรับบริการหนึ่งๆ

ดังนั้น งานวิจัยนี้จึงพยายามที่จะแก้ไขปัญหาประสิทธิภาพของระบบแมทช์เมคกิ้งเอเจนท์โดยอาศัยการพัฒนาโครงสร้างของกลุ่มแมทช์เมคกิ้งเอเจนท์ที่ช่วยสนับสนุนการทำงานร่วมกันของแมทช์เมคกิ้งเอเจนท์ มีความสอดคล้องกับความเป็นเอเจนท์ และใช้ทรัพยากรเครือข่ายน้อยลง ทั้งนี้ โครงสร้างของแมทช์เมคกิ้งเอเจนท์ที่ดีจะต้องคำนึงถึงปัจจัยต่างๆ ได้แก่

- เป็นโครงสร้างที่จัดเรียงแมทช์เมคกิ้งเอเจนท์โดยคำนึงถึงคุณสมบัติโดยทั่วไปของเอเจนท์ได้แก่ความสามารถทางสังคมและการมีภาวะอิสระ
- กลไกการร่วมมือกันทำงานของแมทช์เมคกิ้งเอเจนท์อาศัยการติดต่อสื่อสารระหว่างแมทช์เมคกิ้งเอเจนท์น้อย เพื่อจะได้ไม่รบกวนการทำงานของเอเจนท์และสาธารณูปโภคอื่นของระบบเอเจนท์นั้นๆ
- มีการกำหนดขอบเขตความร่วมมือระหว่างแมทช์เมคกิ้งเอเจนท์ให้อยู่ในวงจำกัด
- เป็นโครงสร้างที่ได้รับผลกระทบจากการอัพเดตหรือเปลี่ยนแปลงโฆษณาน้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากคุณสมบัติข้างต้น โครงสร้างที่มีความเหมาะสมที่สุดจะได้แก่โครงสร้างแบบลำดับชั้นที่มีความสูงไม่มากและมีความสูงคงที่เพื่อสามารถลดจำนวนการติดต่อระหว่างแมทช์เมคกิ้งเอเจนท์ โครงสร้างดังกล่าวจะได้แก่โครงสร้างแบบ Star หรือโครงสร้างแบบลำดับชั้นที่มีความสูงเท่ากับ 1 เสมอนั่นเอง

นอกจากการเลือกโครงสร้างที่เหมาะสมแล้ว ในงานวิจัยที่นำเสนอนี้ยังมีการแบ่งหน้าที่รับผิดชอบของแมทช์เมคกิ้งเอเจนท์เพื่อจำกัดขอบเขตความร่วมมือด้วย งานวิจัยนี้จะได้แบ่งแมทช์เมคกิ้งเอเจนท์ออกเป็น 3 ประเภท ได้แก่ Local MMA (LMMA), Authoritative MMA (AMMA) และ Root MMA (RMMA) ประเภทแรก LMMA มีหน้าที่ให้บริการจับคู่แก่เอเจนท์ภายในกลุ่มเอเจนท์ย่อยที่ตนเองสังกัดอยู่ ประเภทที่สองได้แก่ AMMA เป็นแมทช์เมคกิ้งเอเจนท์ซึ่งสังกัดกลุ่มเอเจนท์หนึ่งซึ่งอาจสังกัดหรือไม่สังกัดกลุ่มเอเจนท์ย่อยใดๆ ก็ได้และมีโฆษณาความสามารถที่สอดคล้องกับบริการที่ร้องขอมา ประเภทสุดท้ายได้แก่ RMMA เป็นแมทช์เมคกิ้งเอเจนท์ซึ่งสังกัดกลุ่มเอเจนท์หนึ่งและทำหน้าที่ควบคุมการส่งต่อคำร้องที่ยังไม่สามารถค้นหาโฆษณาที่เหมาะสมไปยังแมทช์เมคกิ้งเอเจนท์ที่สามารถให้บริการจับคู่คำร้องนั้นได้ (AMMA) อย่างไรก็ตาม เราไม่สามารถทราบได้ว่า AMMA ใดมีความสามารถดังกล่าวบ้างหากมิได้มีการแบ่งขอบเขตความรับผิดชอบไว้ล่วงหน้า ดังนั้นในงานวิจัยนี้จึงนำเสนอว่าควรจะกำหนดให้แต่ละ AMMA รับผิดชอบการจับคู่สำหรับบางโดเมนเท่านั้น ตัวอย่างเช่น AMMA1 รับผิดชอบโดเมนการขนส่งทางบก AMMA2 รับผิดชอบโดเมนการขนส่งทางอากาศ เป็นต้น

เพื่อทดสอบว่าระบบที่นำเสนอนั้นมีประสิทธิภาพเพียงใด ในงานวิจัยนี้จึงได้พัฒนาระบบแมทช์เมคกิ้งเอเจนท์ขึ้นตามแนวคิดที่ได้นำเสนอไปข้างต้นเพื่อใช้ในการทดสอบ เอเจนท์ทั้งหมดในระบบที่พัฒนาขึ้นจะใช้การแสดงความต้องการและความสามารถแบบ Predicate Calculus Sentence และใช้ KQML เป็นโพรโทคอลแลกความรู้ สำหรับแมทช์เมคกิ้งเอเจนท์ที่พัฒนาขึ้นนั้นจะแบ่งเป็น 5 มอดูลได้แก่ มอดูล Communicator, Matching Engine, Cache Manager, Forwarder และ Queue Manager

มอดูล Communicator จะรับผิดชอบการติดต่อสื่อสารกับเอเจนท์อื่นๆ ทั้งหมด โดยตั้งอยู่บนพื้นฐานของโพรโทคอลแลกความรู้ KQML มอดูลต่อมาได้แก่ Matching Engine มีหน้าที่ให้บริการจับคู่เป็นหลัก มอดูลนี้ประกอบไปด้วยส่วนต่างๆ ได้แก่ฐานความรู้ความสามารถ Taxonomic Knowledge Base (TKB) และตัว Matching Engine เอง ซึ่งจะให้บริการกลไกจับคู่ 3 แบบ ได้แก่ Exact Matching, Pattern Matching และ Subsumption Matching สำหรับกลไกสุดท้ายนั้นจะใช้เทคนิค Similarity Rating ของ Girardi and Ibrahim (1994) และ Lin (1998) ซึ่งเป็นเทคนิคที่ใช้ TKB ช่วยในการให้คะแนนความเหมือนสำหรับแต่ละคู่ของโฆษณาและคำร้องขอรับบริการ มอดูลที่สามได้แก่ Cache Manager ทำหน้าที่ช่วยจัดเก็บผลลัพธ์จากกระบวนการจับคู่ครั้งก่อนและสามารถนำข้อมูลดังกล่าวมาใช้งานในภายหลัง มอดูลต่อมาได้แก่ Forwarder ทำหน้าที่ส่งต่อคำร้องขอรับบริการที่แมทช์เมคกิ้งเอเจนท์หนึ่งๆ ไม่สามารถให้บริการได้ไปยังแมทช์เมคกิ้งเอเจนท์อื่นที่สามารถให้บริการจับคู่แก่คำร้องนั้นได้ ไม่โดยทางตรงก็ทางอ้อม มอดูลสุดท้าย Queue Manager ทำหน้าที่ประติดประต่อการทำงานแต่ละส่วนของแมทช์เมคกิ้งเอเจนท์เข้าด้วยกันและควบคุมลำดับการให้บริการให้มีลักษณะเป็นแบบ FCFS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดสอบพบว่าระบบแมทช์เมคกิ้งเอเจนท์ที่นำเสนอในงานวิจัยนี้มีประสิทธิภาพสูงกว่าสถาปัตยกรรมแบบอื่นซึ่งได้แก่แบบ A4 ในทุกๆ กรณีทดสอบ อย่างไรก็ตาม สถาปัตยกรรมที่นำเสนอยังคงมีข้อจำกัดที่สำคัญอยู่ คือ ประสิทธิภาพการทำงานของระบบจะถูกจำกัดโดยประสิทธิภาพของ RMMA ถึงแม้ว่าในการออกแบบที่ผ่านมาจะได้พยายามออกแบบให้ RMMA รับผิดชอบหน้าที่น้อยที่สุดแล้ว (กล่าวคือให้ RMMA จะทำหน้าที่ในการพิจารณาว่าควรจะส่งต่อคำร้องไปยัง AMMA ใดเท่านั้น มิได้ทำการจับคู่ให้กับคำร้องดังกล่าวแต่อย่างใด)

ในแง่ของการโฆษณาความสามารถ ระบบแมทช์เมคกิ้งเอเจนท์แบบ A4 ได้กำหนดให้แมทช์เมคกิ้งเอเจนท์ต้องโฆษณาความสามารถของเอเจนท์ผู้ให้บริการที่ตนเองทราบไปยังแมทช์เมคกิ้งเอเจนท์ที่อยู่ในระดับที่สูงกว่า ดังนั้น แมทช์เมคกิ้งเอเจนท์ที่อยู่ในระดับที่สูงกว่าจึงมีฐานความรู้ความสามารถที่มีขนาดใหญ่กว่าระดับที่ต่ำกว่า ส่วนระบบ DMAA นั้น แมทช์เมคกิ้งเอเจนท์แต่ละตัวจะทราบเฉพาะความสามารถของกลุ่มเอเจนท์ผู้ให้บริการที่ตนเองรับผิดชอบเท่านั้น ดังนั้น กระบวนการจับคู่ในแต่ละแมทช์เมคกิ้งเอเจนท์ของระบบ DMAA นั้น จึงกระทำกับฐานความรู้ความสามารถที่มีขนาดเท่ากันหรือเล็กกว่าขนาดฐานความรู้ความสามารถเฉลี่ยของแต่ละแมทช์เมคกิ้งเอเจนท์ภายในระบบ A4 ซึ่งส่งผลให้การจับคู่ของระบบ DMAA มีความรวดเร็วในการทำงานที่สูงกว่าการจับคู่ของระบบ A4 นั่นเอง

เทคโนโลยีแมทช์เมคกิ้งเอเจนท์ยังต้องการการวิจัยและพัฒนาต่อไปอีก เพื่อให้สามารถนำไปใช้ได้จริงและประสบความสำเร็จได้รับการยอมรับโดยทั่วไป งานวิจัยที่จะทำต่อไปนอกจากการการปรับปรุงกลไกการจับคู่แล้ว ยังจะต้องพัฒนาให้ระบบแมทช์เมคกิ้งเอเจนท์ที่นำเสนอนี้สามารถติดต่อโดยตรงกับมนุษย์ได้ รวมถึงเพิ่มความสามารถต่างๆ โดยเฉพาะในหัวข้อการรักษาความปลอดภัยให้แก่เทคโนโลยีแมทช์เมคกิ้งเอเจนท์อีกด้วย และที่จะขาดมิได้คือการกำจัดข้อจำกัดของระบบที่นำเสนอ โดยเสนอให้นำ RMMA หลายๆ ตัวมาทำงานร่วมกันภายในระบบ โดยให้ RMMA แบ่งงานกันทำเองโดยใช้การต่อรองซึ่งกันและกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ปกรณ์พงศ์ โพธิพฤกษ์. 2544. "อินเทลลิเจนท์เอเจนท์." **สารเนคเทค**. ฉบับที่ 43 (พ.ย.-ธ.ค.) หน้า 46-55.

ปกรณ์พงศ์ โพธิพฤกษ์ และ ภัทรชัย ลลิตโรจน์วงศ์. 2545. "แมทช์เมคกิ้งเอเจนท์." **สารเนคเทค**. ฉบับที่ 44(ม.ค.-ก.พ.) หน้า 39-45.

มีนา รัตนสาครกุล และ ภัทรชัย ลลิตโรจน์วงศ์. 2544. "เอเจนท์เคลื่อนที่." **สารเนคเทค**. ฉบับที่ 42 (ก.ย.-ต.ค.) หน้า 40-50.

Bayardo, R. J. et al. 1998. "InfoSleuth: Agent-based Semantic Integration of Information in Open and Dynamic Environments." 205-216. in Huhns, M. N. and Singh, M. P. **Readings in Agents**. San Francisco: Morgan Kaufmann.

Bigus, J.P. and Bigus, J. 2001. **Constructing Intelligent Agents Using Java**. 2nd ed. New York : John Wiley & Sons.

Cao, J. et al. 2000. "A4 Simulator." Annual Report, Department of Computer Science, University of Warwick, Coventry.

Cao, J. et al. 2001. "High Performance Service Discovery in Large-Scale Multi-Agent and Mobile-Agent Systems." in **Internatoinal Journal of Software Engineering and Knowledge Engineering: Special Issue on Multi-Agent Systems and Mobile Agents**. 11(5) : 621-641.

Carnegie Mellon University. 2002. **Intelligent Software Agents: Agent Name Server**. [Online]. Available : http://www.cs.cmu.edu/~softagents/ans.html.

Cycorp. 2002. **OpenCyc.org: Formalized Common Knowledge**. [Online]. Available : http://www.opencyc.org/.

Davis, R. and Smith, R. G. 1983. "Negotiation as a Metaphor for Distributed Problem Solving." **Artificial Intelligence**. 20 : 63-100.

Decker, K. et al. 1996. "Matchmaking and Brokering." in **Proceedings of the Second International Conference in Multi-Agent Systems**. Kyoto.

Durfee, E. H. 1988. **Coordination of Distributed Problem Solvers**. Boston : Kluwer Academic Publishers.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Finin, T. et al. 1994. "KQML as an Agent Communication Language." in Proceedings of CIKM-94. ACM Press.

Finin, T. et al. 1998. "The DARPA Knowledge Sharing Effort: Progress Report." 243-254. in Huhns, M. N. and Singh, M. P. Readings in Agents. San Francisco : Morgan Kaufmann.

FIPA. 2002. Foundation for Intelligent Physical Agent. [Online]. Available : http://www.fipa.org.

Foner, L. 1997. "Yenta: A Multi-agent Referral-based Matchmaking System." in Proceedings of the First International Conference on Autonomous Agents. Marina Del Rey.

Franklin, S. and Graesser, A. 1996. "Is It an Agent, or Just a Program? A Taxonomy for Autonomous Agents." in Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages. Berlin : Springer-Verlag.

Friedman-Hill, E. 2002. JESS, The Expert System Shell for the Java Platform. [Online]. Available : http://herzberg.ca.sandia.gov/jess/.

Gil, Y. and Ramachandran, S. 2001. "PHOSPHORUS: A Task-based Agent Matchmaker," in Proceeding of AGENTS'01. Montreal.

Girardi, M. R. and Ibrahim, B. 1994. "A Similarity Measure for Retrieving Software Artifacts." 478-485. in Proceedings of the Sixth International Conference on Software Engineering and Knowledge Engineering. Latvia.

Gruber, T. R. 1993. "A Translation Approach to Portable Ontology Specification." Knowledge Acquisition. 2 : 199-220.

Jain, R. 1991. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. New York : John Wiley & Sons.

Jennings, N. R. et al. 1998. "A Roadmap of Agent Research and Development." Autonomous Agents and Multi-Agent Systems. 1 : 275-306.

KeyLabs. 2000. Comparison Report: Microsoft Active Directory VS Netware NDS. [Online]. Available : http://www.keylabs.com/results/microsoft/ ad/comparisonreport.pdf.

Kuokka, D. and Harada, L. 1998. "Matchmaking for Information Agents." 91-97. in Huhns, M. N. and Singh, M. P. Readings in Agents. San Francisco : Morgan Kaufmann.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Kurose, J. F. and Ross, K. W. 2000. **Computer Networking: A Top-down Approach Featuring the Internet**. USA : Addison Wesley-Longman.

Labrou, Y. and Finin, T. 1999. "Agent Communication Languages: The Current Lanscape." **IEEE Intelligent Agents**. March/April : 45-52.

Lin, D. 1998. "An Information-theoretic Definition of Similarity." in **Proceedings of International Conference on Machine Learning**. Madison, Wisconsin.

Luger, G. F. and Stubblefield, W. A. 1998. **Artificial Intelligence: Structures and Strategies for Complex Problem Solving**. 3rd ed. Massachusetts : Addison Wesley Longman.

Microsoft. 2000. **Active Directory Architecture: White Paper**. [Online]. Available : http://www.microsoft.com/windows2000/techinfo/howitworks/activedirectory/adarch.asp.

Mockapetris, P. V. 1987a. **RFC 1034: Domain Names-Concepts and Facilities**. [Online]. Available : http://info.internet.isi.edu/in-notes/rfc/files/rfc1034.txt.

Mockapetris, P. V. 1987b. **RFC 1035: Domain Names-Implementation and Specification**. [Online]. Available : http://info.internet.isi.edu/in-notes/rfc/files/rfc1035.txt.

NASA. 1999. **CLIPS: A Tool for Building Expert Systems**. [Online]. Available : http://www.ghgcorp.com/clips/CLIPS.html.

Novell. 2000. **Competitive Analysis: A Comparative Overview of NDS and Active Directory**. [Online]. Available : http://www.novell.com/products/edirectory/competitive/nds-ad-compare.pdf.

Ogston E. and Vassiliadis, S. 2001. "Local Distributed Agent Matchmaking." in **Proceedings of the 9th International Conference on Cooperative Information Systems**. 67-79.

Petrie, C. J. 1996. "Agent-based Engineering, the Web, and Intelligence." **IEEE Expert**. 11(6) : 24-29.

Pothipruk, P. and Lalitrojwong, P. 2002. "An Ontology-based Multi-agent System for Matchmaking." in **Proceedings of the ICITA2002**. Bathurst, Australia.

Princeton University. 2002. **WordNet: A Lexical Database for English Language**. [Online]. Available : http://www.cogsci.princeton.edu/~wn/.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Resnik, P. 1995. "Using Information Content to Evaluate Semantic Similarity in a Taxonomy." 448-453. in **Proceedings of the 14th International Joint Conference on Artificial Intelligence**. Montreal.

Salton, G. and Wong, A. 1975. "A Vector Space Model for Automatic Indexing." **Communications of the ACM**. 18 : 613-620.

Shehory, O. 2000. "A Scalable Agent Location Mechanism." 162-172. in **Intelligent Agents VI - Proceedings of the Sixth International Workshop on Agent Theories, Architectures, and Languages (ATAL'99)**. Vol. 1757 of Lecture Notes in Artificial Intelligence. Heidelberg : Springer-Verlag.

Sleepycat Software. 2001. **Berkeley DB**. [Online]. Available : http://www.sleepycat.com/products.html.

Stanford University. 1999. **JATLite**. [Online]. Available : http://java.stanford.edu/.

Sycara, K. et al. 1997. "Middle-Agents for the Internet." 578-583. in **Proceedings of the 15th IJCAI**. Nagoya.

Sycara, K. et al. 1999. "Matchmaking among Heterogeneous Agents on the Internet." in **Proceedings AAAI Spring Symposium on Intelligent Agents in Cyberspace**. Stanford : AAAI.

Tecuci, G. 1998. **Building Intelligent Agent: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies**. San Diego : Academic Press.

Veloso, M. et al. 1997. "A Team of Robotic Soccer Agents Collaborating in an Adversarial Environment." in **Proceedings of the First International Workshop on RoboCup**. Nagoya.

Vivacqua, A. S. 1999. "Agents for Expertise Location." in **Proceedings of the AAAI Spring Symposium on Intelligent Agents in Cyberspace**. Stanford : AAAI.

Wooldridge, M. J. 1998. "Agent-Based Computing." **Interoperable Communication Networks**. 1(1) : 71-97.

Wooldridge, M. J. and Jennings, N. R. 1995. "Intelligent Agents: Theory and Practice." **Knowledge Engineering Review**. 10(2) : 115-152.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้