แมทช์เมคกิ้งเอเจนท์

Matchmaking Agents

ปกรณ์พงศ์ โพธิพฤกษ์[1] ดร.ภัทรชัย ลลิตโรจน์วงศ์[2]

บทคัดย่อ

อินเทลลิเจนท์เอเจนท์คือตัวแทนที่สามารถทำงานตามที่ผู้ใช้กำหนดได้ด้วยตนเองอย่างไรก็ตามได้มีความพยายามที่จะนำเอเจนท์ดังกล่าวมาทำงานร่วมกันเพื่อขยายขีดความสามารถในการแก้ปัญหาให้มากขึ้น โดยนำหลักการทำงานเป็นกลุ่มเข้ามาใช้ งานวิจัยประเภทนี้เรียกว่า MAS ปัญหาสำคัญปัญหาหนึ่งที่เกิดขึ้นในระบบดังกล่าว ได้แก่ ทำอย่างไรเอเจนท์หนึ่งๆ จึงจะสามารถค้นหาเอเจนท์ที่ให้บริการที่ต้องการได้ แนวทางการแก้ไขปัญหาที่ได้รับความนิยมได้แก่ แนวคิด Matchmaking Agent ถึงกระนั้นงานวัจัยด้านนี้ก็ยังไม่สมบูรณ์นัก โดยปัญหาสำคัญที่เกิดขึ้นได้แก่การขาดความสามารถในการขยายขนาด ทำให้ไม่สามารถนำไปใช้งานได้จริงกับระบบเครือข่ายขนาดใหญ่

Abstract

Intelligent agent is a representative that can autonomously do user-specific jobs. However, there was an effort in introducing that agent to work together, in order to expand problem solving ability. Cooperation concept is applied in this idea. This category of research is called *MAS (Multi-agent System)*. One of many important problems in this system arises: “How can an agent find another agent that provides a set of required-services for him?” A popular solution to this problem is *Matchmaking Agent* concept. Nevertheless, research in this topic is still not yet complete. The important problem that happens here is the *lack in scalability* of this kind of agent. This causes impossibility of pragmatic implementation in any huge network.

1. บทนำ

เอเจนท์คือตัวแทนที่ทำหน้าที่อย่างใดอย่างหนึ่งโดยเฉพาะ สามารถทำงานให้สำเร็จตามวัตถุประสงค์ของผู้ใช้ได้ด้วยตนเอง เราสามารถแบ่งประเภทของเอเจนท์ได้หลายวิธีด้วยกัน เช่น Robotic Agent และ Software Agent เป็นต้น ในบทความฉบับนี้เน้นเฉพาะเอเจนท์ประเภทหลังเท่านั้น อย่างไรก็ตามเมื่อพูดถึงเอเจนท์มักจะหมายถึงอินเทลลิเจนท์เอเจนท์โดยนัยด้วย โดยอินเทลลิเจนท์เอเจนท์จะหมายถึงเอเจนท์ที่สามารถทำงานโดยสามารถปกครองและควบคุมตนเองได้ และรบกวนผู้ใช้น้อยที่สุด [4] แนวคิดนี้สามารถนำไปประยุกต์ใช้แก้ปัญหาต่างๆ ได้มากมาย รวมถึงสามารถช่วยอำนวยความสะดวกสบายให้แก่ผู้ใช้ได้มากขึ้นด้วย

อย่างไรก็ตามได้มีความพยายามนำเอเจนท์ดังกล่าวมาทำงานร่วมกัน เพื่อเพิ่มประสิทธิภาพในการทำงานให้สูงขึ้น รวมถึงเพื่อสามารถนำไปประยุกต์ใช้กับปัญหาที่มีขนาดใหญ่ขึ้นได้ โดยนำหลักการของการทำงานร่วมกันเข้ามาใช้ แนวคิด

---

[1] นักศึกษาปริญญาโทคณะเทคโนโลยีสารสนเทศ สจล.

[2] อาจารย์คณะเทคโนโลยีสารสนเทศ สจล.

ดังกล่าวเรียกว่าระบบ MAS (Multi-agent system) ระบบนี้ประกอบไปด้วยเอเจนท์จำนวนตั้งแต่สองตัวขึ้นไปมาทำงานร่วมกัน โดยเอเจนท์นั้นอาจจะมีความสามารถในการเคลื่อนที่หรือไม่ก็ได้ [5] ลักษณะการทำงานอาจจะทำงานร่วมกันเป็นกลุ่ม หรือเป็นการทำงานที่สนใจตนเองเป็นหลักก็ได้ กลไกของเอเจนท์ในระบบแต่ละแบบจะแตกต่างกันออกไป แต่ระบบดังกล่าวมีปัญหาหนึ่งที่สำคัญยิ่งที่ต้องแก้ไขให้ได้ก่อนไม่ว่าจะเป็นระบบแบบใดก็ตาม เพื่อให้สามารถนำระบบนั้นๆ ไปใช้งานได้จริง ได้แก่ เอเจนท์ต้องทราบก่อนว่ามีเอเจนท์ใดที่สามารถให้บริการที่ตนเองต้องการได้ และเอเจนท์ดังกล่าวอยู่ที่ใด

แนวทางหนึ่งในการแก้ไขปัญหาข้างต้น ที่มีความเป็นไปได้ และได้รับความสนใจจากนักวิจัยชั้นนำ คือแนวคิด Matchmaking Agent (MMA) ในบทความฉบับนี้จะได้นำเสนอผลงานวิจัยที่ผ่านมาเกี่ยวกับแนวคิดนี้ ซึ่งได้แก่ Expert Finder [12], Middle Agent [11], Scalable Agent Location Mechanism [10] และ Facilitator [3] อย่างไรก็ตามแนวคิดเหล่านี้ยังคงมีปัญหาหนึ่งที่ไม่สามารถแก้ไขได้ ปัญหาดังกล่าวก็คือการไม่สามารถนำระบบที่นำเสนอไปใช้ในระบบที่มีขนาดใหญ่กว่าได้ ดังนั้นการจะสามารถนำแนวคิด MMA ไปใช้งานได้จริงนั้นจึงจำเป็นอย่างยิ่งที่จะต้องแก้ไขปัญหาดังกล่าวเสียก่อน ซึ่งประเด็นนี้จะได้ถูกศึกษาต่อไปในอนาคตอันใกล้

2. แมทช์เมคกิ้งเอเจนท์

เป็นเอเจนท์ที่ทำหน้าที่ช่วยค้นหาเอเจนท์ที่สามารถให้บริการที่เอเจนท์หนึ่งๆ ร้องขอมาได้ [2] โดยมีลักษณะการทำงานโดยทั่วไปเป็นดังรูปที่ 1

เอเจนท์ผู้ให้บริการ (Provider Agent) จะทำการ register ตัวเองกับ MMA โดยทำการโฆษณาบริการที่ตนเองสามารถให้กับเอเจนท์อื่นๆ ได้ไปยัง MMA ซึ่งเรียกว่า Advertisement และข้อมูลนี้จะถูกจัดเก็บอยู่ที่ MMA ดังกล่าว เมื่อมีเอเจนท์ผู้ขอรับบริการ (Requester Agent) ร้องขอบริการเข้ามายัง MMA ตัว MMA นี้จะทำการค้นหาฐานข้อมูล Advertisement ของตนเองเพื่อเปรียบเทียบว่ามีบริการของเอเจนท์ใดที่โฆษณามาซึ่งสอดคล้องกับ Service Request จากนั้น MMA จะตอบกลับไปยังเอเจนท์ผู้ขอรับบริการว่ามีเอเจนท์ใดที่สามารถให้บริการดังกล่าวได้บ้าง พร้อมกับระบุที่อยู่หรือวิธีการติดต่อกับเอเจนท์ดังกล่าว เพื่อให้เอเจนท์ผู้ขอรับบริการสามารถติดต่อกับเอเจนท์ดังกล่าวได้โดยตรง

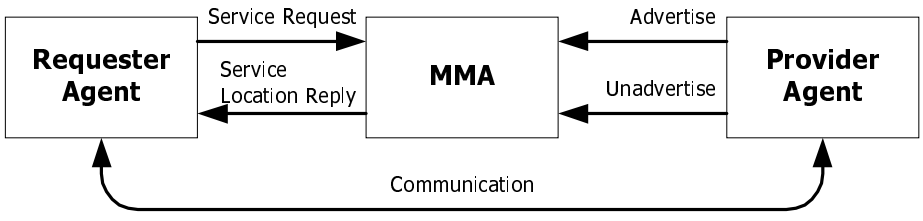
องค์ประกอบหลักที่สำคัญของ MMA แบ่งเป็น 2 ส่วน [8] ได้แก่

1. Specific Matchmaking Algorithms
2. Knowledge-sharing Protocols

ส่วนแรก ได้แก่ อัลกอริทึมที่ MMA ใช้ในการเปรียบเทียบ Request ที่เข้ามากับ Advertisement ที่เก็บอยู่ในตัว MMA เอง องค์ประกอบที่สำคัญอีกประการหนึ่งของ MMA คือโปรโตคอลที่ใช้ในการแบ่งปันKnowledge ระหว่างเอเจนท์ ที่นิยมใช้กันได้แก่ ภาษา KQML (Knowledge Query and Manipulation Language) นอกจากนี้ยังมีอีกภาษาที่ใช้กันได้แก่ FIPA ACL แต่ได้รับความนิยมน้อยกว่า KQML มาก

3. โปรโตคอลแลกความรู้

เอเจนท์แต่ละรายจะต้องมีการติดต่อสื่อสารระหว่างกัน ไม่



รูปที่ 1. การทำงานทั่วไปของแมทช์เมคกิ้งเอเจนท์

ว่าจะเป็นการติดต่อกันภายในกลุ่มของเอเจนท์ที่ทำงานร่วมกัน หรือการแลกเปลี่ยนความรู้ระหว่างเอเจนท์คนละกลุ่มกัน ประกอบกับในปัจจุบันมีสถาปัตยกรรมเอเจนท์ต่างๆ เกิดขึ้นมากมาย ดังนั้นการติดต่อดังกล่าวจึงต้องการภาษาที่เป็นมาตรฐานเพื่อให้เอเจนท์ต่างสถาปัตยกรรมกันสามารถติดต่อสื่อสารกันได้ ซึ่งก็คือ ACL (Agent Communication Language) หรือโปรโตคอลแลกความรู้ (Knowledge-sharing Protocol) นั่นเอง

ACL ได้แก่ภาษาและโปรโตคอลที่ใช้สำหรับสื่อสารกันระหว่างเอเจนท์ เพื่อแลกเปลี่ยนข้อมูล และความรู้ของเอเจนท์ โดยความรู้ที่เอเจนท์จะแลกเปลี่ยนกันได้แก่ สภาวะของทัศนคติต่างๆ (โดยมากมักเป็น BDI States [7]) ซึ่งควรจะอยู่ในรูปของภาษาธรรมชาติเพื่อให้สามารถติดต่อระหว่างเอเจนท์ต่างประเภทกันได้

ในปัจจุบัน ACL ที่มีชื่อนั้นแบ่งเป็น 2 ประเภท ได้แก่ KQML และ FIPA ACL [9]

3.1 KQML

KQML (Knowledge Query and Manipulation Language) คือภาษาและโปรโตคอลระดับสูงเพื่อการแลกเปลี่ยนข้อมูลและความรู้โดยใช้ข้อความ โดยไม่ขึ้นกับ Syntax ของเนื้อหาที่สื่อสารกัน ไม่ขึ้นกับ Ontology [6] ของแต่ละเอเจนท์ ไม่ขึ้นกับโปรโตคอลที่ใช้ในการส่งข้อมูลที่อยู่ในชั้นต่ำกว่า (เช่น TCP/IP, SMTP เป็นต้น) และไม่ขึ้นกับภาษาที่ใช้เขียนเอเจนท์ (เช่น Prolog, KIF เป็นต้น) ข้อความแบบ KQML แบ่งเป็น 3 ส่วน ได้แก่ ส่วนเนื้อหา (Content) ส่วนการสื่อสาร (Communication) และส่วนข้อความ (Message)

ส่วนเนื้อหา ได้แก่ข้อมูลที่เอเจนท์ต้องการจะส่ง ซึ่งอยู่ในรูปแบบของภาษาและการแสดงความรู้ที่เอเจนท์นั้นๆ เลือกใช้ (มักไม่รวมส่วนสารบัญของข้อมูล เว้นแต่กรณีที่ต้องการหาจุดสิ้นสุดของข้อมูลดังกล่าว)

ส่วนการสื่อสาร ประกอบด้วยพารามิเตอร์ต่างๆ ที่จำเป็นสำหรับการสื่อสาร เช่น Identifier ของผู้ส่ง ของผู้รับ และของการสื่อสารแต่ละครั้ง เป็นต้น

```
(ask-one
        : sender joe
        : content (PRICE IBM ?price)
        : receiver stock-server
        : reply-with ibm-stock
        : language LPROLOG
        : ontology NYSE-TICKS)

                (ก)

(tell
        : sender stock-server
        : content (PRICE IBM     )
        : receiver joe
        : in-reply-to ibm-stock
        : language LPROLOG
        : ontology NYSE-TICKS)

                (ข)
```

รูปที่ 2. ตัวอย่างข้อความ KQML (ก. ผู้ถาม และ ข. ผู้ตอบ)

ส่วนข้อความ ระบุ Performative (ประเภทของข้อความ เช่น เป็นคำสั่ง คำถาม หรือคำยืนยัน เป็นต้น) ที่ผู้ส่งใช้ในการสื่อสารครั้งนั้นๆ

รูปที่ 2 เป็นตัวอย่างของข้อความแบบ KQML แบบหนึ่ง โดยแบ่งเป็นข้อความของผู้ส่ง และข้อความตอบกลับตามลำดับ สำหรับข้อความของผู้ส่งแบ่งเป็นส่วนๆ ได้ดังนี้

- 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 และรวมถึง Performative ที่ชื่อ ask-one (บอกว่าข้อความนี้เป็นข้อความประเภทสอบถาม) ด้วย

KQML จัดเป็น ACL ที่ได้รับความนิยมอย่างสูงในปัจจุบัน มีการนำ KQML มาประยุกต์ใช้ในระบบเอเจนท์ต่างๆ จำนวนมาก อย่างไรก็ตามการประยุกต์ดังกล่าวยังขาดความเป็นมาตรฐานอยู่อีกมาก เนื่องจากแต่ละระบบพยายามที่จะปรับปรุง KQML ให้เหมาะสมกับระบบของตนเองให้มากที่สุด ดังนั้นจึงมีรูปแบบย่อยๆ ของ KQML เกิดขึ้นเช่น TKQML [1] เป็นต้น

3.2 FIPA ACL

พัฒนาโดย Foundation for Intelligent Physical Agent ซึ่งเป็นองค์กรแรกๆ ที่พยายามตั้งมาตรฐานเกี่ยวกับเอเจนท์ทั้ง Software Agent และ Robotic Agent โดยก่อตั้งขึ้นในปีพ.ศ. 2539 เป็นภาษาที่มีลักษณะคล้ายคลึงกับ KQML โดยเฉพาะอย่างยิ่งในด้านของ Syntax รายละเอียดเพิ่มเติมสามารถหาได้จากเว็บไซต์ http://www.fipa.org

3.3 สรุป ACL

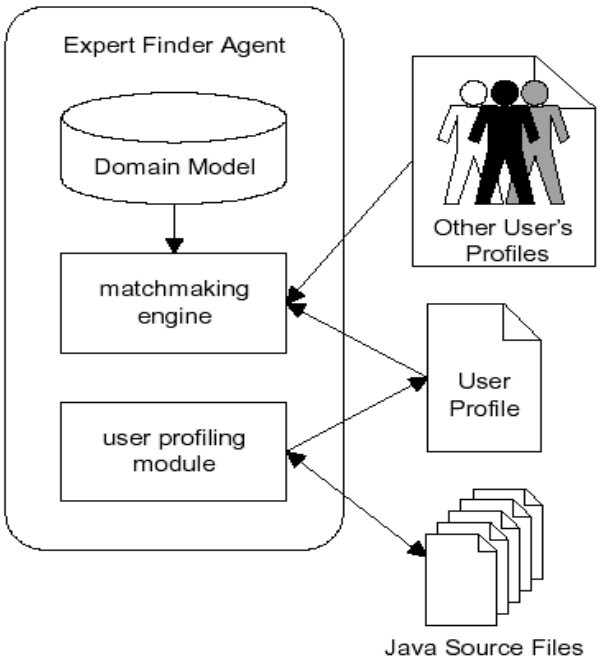
ในการพัฒนาเอเจนท์โดยเฉพาะอย่างยิ่งประเภท MAS นั้นต้องการ ACL อย่างน้อยแบบใดแบบหนึ่งเพื่อการติดต่อสื่อสารระหว่างเอเจนท์ด้วยกัน ทั้งนี้เราอาจมองระบบ MMA เป็น MAS ประเภทหนึ่งได้ โดยระบบนี้ต้องมีการประสานงานกันระหว่างเอเจนท์ผู้ขอรับบริการ เอเจนท์ผู้ให้บริการ และ MMA ซึ่งต้องการ ACL เข้ามารองรับ เพื่อช่วยให้สามารถติดต่อสื่อสารกันได้ โดยภาษาที่นิยมใช้กันในงานวิจัยทางด้านนี้ได้แก่ ภาษา KQML [3] นั่นเอง

4. งานวิจัยด้าน MMA

ในปัจจุบันได้มีผู้พัฒนาระบบ MMA ขึ้นมาหลายหลาย โดยแต่ละแบบจะมีข้อดีและข้อเสียแตกต่างกันไป ในบทความฉบับนี้จะได้นำเสนอตัวอย่างแนวคิดในระบบ MMA 4 รูปแบบด้วยกัน ได้แก่ Expert Finder [12], Middle Agent [11], Scalable Agent Location Mechanism [10] และ Facilitator [3]

4.1 Expert Finder

Expert Finder [12] เป็น MMA ระดับพื้นฐานที่สุด ทำหน้าที่ช่วยในการหาผู้เชี่ยวชาญเฉพาะด้านที่สามารถให้คำปรึกษาที่เหมาะสมกับระดับความรู้ของผู้ใช้แต่ละคนได้



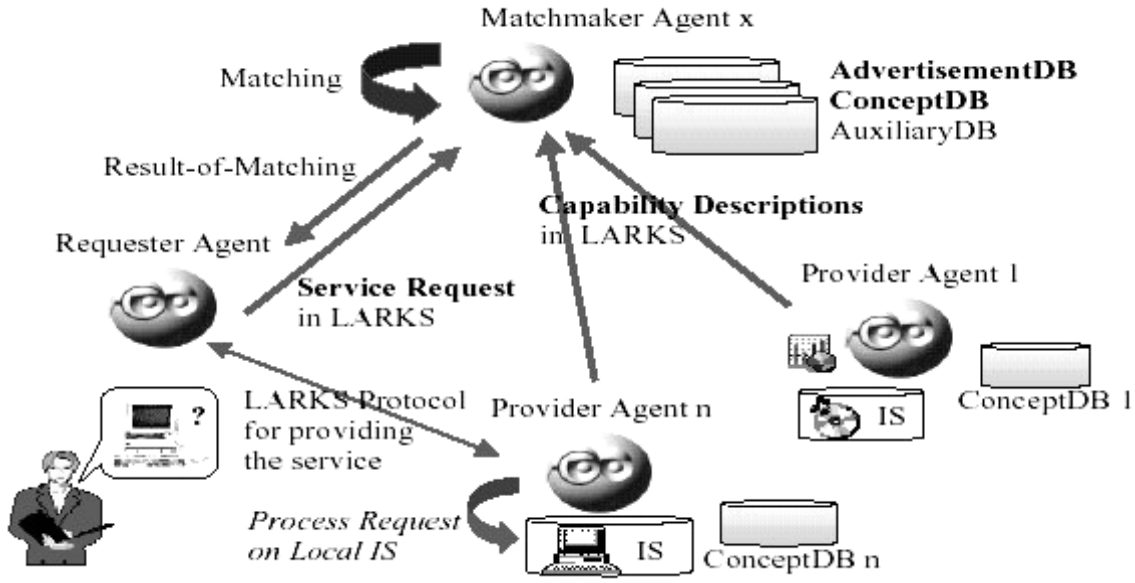
รูปที่ 3. การทำงานภายใน Expert Finder [12]

งานวิจัยนี้ได้นำเสนอ Matchmaking Algorithm ที่ใช้ในการจับคู่หาผู้เชี่ยญชาญที่สามารถให้คำปรึกษากับผู้ใช้หนึ่งๆ ได้ โดยอาศัย User Profile (รูปที่ 3) ในการประเมินระดับความรู้ด้านใดด้านหนึ่งของผู้ใช้ดังกล่าวโดยอาศัยการเปรียบเทียบกับผู้ใช้คนอื่นโดยเทียบกับ Other User's Profiles นอกจากนี้ยังได้นำเสนอกลไกในการตรวจสอบผู้เชี่ยวชาญปลอมด้วย โดยอาศัยการตรวจสอบความสามารถในการตอบคำถามให้กับผู้ใช้ หลักการที่สำคัญคือผู้ใช้ และผู้เชี่ยวชาญทุกคนในระบบนี้จะต้องมีเอเจนท์ Expert Finder ซึ่งแต่ละตัวจะต้องมีรายชื่อและที่อยู่ของเอเจนท์อื่นๆ (Other Expert Finder) ทั้งหมดที่อยู่ในระบบนี้

4.2 Middle Agent

Middle Agent [11] เป็นโครงแบบ MMA ที่เป็นที่รู้จักและเข้าใจกันทั่วไปมากที่สุด ซึ่งมีการทำงานเป็นไปตามในรูปที่ 4

หลักการทำงานของ MMA แบบนี้คือเอเจนท์ผู้ให้บริการ (Provider Agent) จะทำการโฆษณาตัวเองกับ Middle Agent โดยส่งความสามารถบริการที่ตนเองสามารถให้กับเอเจนท์อื่นๆ ได้ (Capability Descriptions) ไปยัง Middle Agent และ



รูปที่ 4. การทำงานของ Middle Agent [11]

ข้อมูลนี้จะถูกจัดเก็บอยู่ที่ฐานข้อมูล Advertisement ของ Middle Agent เมื่อมีเอเจนท์ผู้ขอรับบริการ (Requester Agent) ร้องขอบริการเข้ามายัง Middle Agent ตัว Middle Agent นี้จะทำการค้นหาฐานข้อมูลของตนเองเพื่อเปรียบเทียบว่ามีบริการของเอเจนท์ใดที่โฆษณาเข้ามาซึ่งสอดคล้องกับ Service Request ที่เข้ามา จากนั้น Middle Agent จะตอบกลับไปยังเอเจนท์ผู้ขอรับบริการว่ามีเอเจนท์ใดที่สามารถให้บริการดังกล่าวได้บ้าง เพื่อให้ติดต่อไปยังเอเจนท์ดังกล่าวและขอรับบริการโดยตรงจากเอเจนท์นั้น

ในงานวิจัยนี้ได้เน้นส่วนของ Knowledge-sharing Protocol โดยได้พัฒนา LARKS (Language for Advertisement and Request for Knowledge Sharing) เพื่อใช้ในการติดต่อร้องขอบริการ และโฆษณาบริการกับ Middle Agent เอง โดยจะทำหน้าที่เป็นทั้งโปรโตคอลในการติดต่อและใช้ในการแสดงความสามารถของเอเจนท์ในการให้บริการกับเอเจนท์อื่นๆ ด้วย

4.3 Scalable Agent Location Mechanism

บทความนี้ [10] นำเสนอกลไกในการค้นหาเอเจนท์โดยที่ไม่จำเป็นต้องมี MMA แต่ใช้หลักการของการกระจายข้อมูลส่วนกลางไปยังเอเจนท์แต่ละตัว โดยได้ชี้ให้เห็นว่าการใช้ MMA มีข้อเสียที่สำคัญดังนี้

1. ต้นทุนที่เพิ่มขึ้นจากการที่ต้องติดต่อ MMA ก่อนเสมอ และต้นทุนการสื่อสารภายในกลุ่มของ MMA เอง
2. MMA เป็น Infrastructure พิเศษที่ต้องมีเพิ่มขึ้น และเอเจนท์ต้องมีโปรโตคอลที่ใช้ติดต่อกับ Local MMA ด้วย

หลักการของวิธีนี้คือ นำฟังก์ชั่นในการค้นหาเอเจนท์ไปไว้ที่ตัวเอเจนท์ทุกๆ ตัวแทนการใช้ MMA ซึ่งเป็นเสมือนฐานข้อมูลของเอเจนท์ในระบบ ซึ่งเป็นวิธีการที่คล้ายคลึงกับ Expert Finder แต่มีความแตกต่างที่สำคัญตรงที่เอเจนท์แต่ละตัวไม่จำเป็นที่จะต้องเก็บข้อมูลเกี่ยวกับเอเจนท์ทั้งหมดที่มีอยู่ในระบบ แต่เอเจนท์แต่ละตัวจะรู้จักเอเจนท์อื่นเป็นจำนวนหนึ่งเท่านั้น เมื่อเอเจนท์ต้องการค้นหาเอเจนท์ใดๆ ก็จะทำการแลกเปลี่ยนข้อมูลซึ่งกันและกันภายในระบบนั้น โดยผู้วิจัยได้อ้างไว้ว่าวิธีนี้สามารถลดความซับซ้อนของอัลกอริทึมลงไปได้มาก

วิธีนี้มีกลไกการเก็บข้อมูลเกี่ยวกับที่อยู่ของเอเจนท์ที่เปรียบได้กับฐานข้อมูลแบบกระจาย และมีวิธีการค้นหาเอเจนท์ที่เหมือนกับหลักการของ Recursive DNS Protocol

4.4 Facilitator [3]

เป็นบทความที่สรุปการใช้งาน MMA ในรูปแบบต่างๆ ซึ่งมีทั้งหมด 4 รูปแบบด้วยกัน ได้แก่ Subscriber, Broker, Recruiter และ Recommender แต่ละรูปแบบจะมีวิธีการทำงานและใช้ KQML Performatives ที่แตกต่างกัน ออกไปดังรูปที่ 5

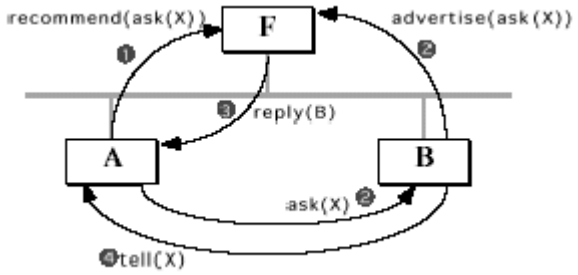
ในบทความนี้ได้เรียก MMA ว่า Facilitator (เอเจนท์ F) โดย Facilitator ทั้งสี่แบบนั้นมีการทำงานที่แตกต่างกันเล็กน้อย แบบที่เป็นนิยมใช้ทั่วไป ได้แก่ แบบ Recommender ซึ่งมีกระบวนการทำงานเหมือนในรูปที่ 5.ก

สำหรับการทำงานของ Facilitator ประเภท Broker นั้น (รูปที่ 5.ข) จะแตกต่างจากแบบแรกตรงที่ Facilitator จะไม่ส่ง Reply Performative กลับไปยังเอเจนท์ผู้ขอรับบริการ แต่จะทำหน้าที่เป็น Forwarder ให้ในตัวด้วย กล่าวคือทำหน้าที่เป็นตัวกลางในการติดต่อระหว่างเอเจนท์ผู้ขอรับบริการและเอเจนท์ผู้ให้บริการ โดยอาศัย Ask และ Tell Performative

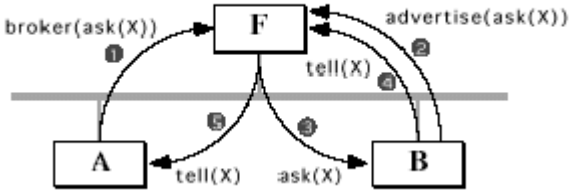
ส่วน Facilitator ประเภท Recruiter นั้น (รูปที่ 5.ค) จะแตกต่างจากแบบ Broker ตรงที่ Facilitator จะส่งรายการบริการที่ตัวเองต้องการไปยังเอเจนท์ผู้ให้บริการเท่านั้น จากนั้นให้เอเจนท์ผู้ให้บริการตอบคำถามตรงไปยังเอเจนท์ผู้ขอรับบริการโดยตรง

สำหรับ Facilitator ประเภทสุดท้าย ได้แก่แบบ Subscriber (รูปที่ 5.ง) เป็นกรณีที่มอง Facilitator เหมือนเป็นผู้ให้บริการใดบริการหนึ่ง โดยเอเจนท์ผู้ขอรับบริการจะทำการ subscribe ขอรับบริการจากตัว Facilitator โดยตรง และเมื่อถึงเวลาที่กำหนด เอเจนท์ให้บริการจะส่งบริการผ่านมายัง Facilitator เพื่อให้ Facilitator ทำการส่งผ่านบริการดังกล่าวต่อไปยังเอเจนท์ผู้ขอรับบริการ

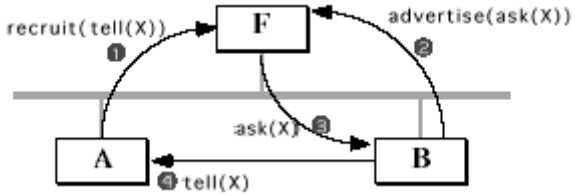
จากข้างต้นจะพบว่า Facilitator รูปแบบที่ 2 และ 3 จะเป็นลักษณะที่นำ Facilitator ไปใช้กับงานทางด้าน Information Retrieval มากกว่า ในขณะที่รูปแบบสุดท้าย จะใช้กับการ Subscribe ขอรับบริการข้อมูลผ่านทางไปรษณีย์อิเล็กทรอนิกส์ ดังนั้นจึงไม่เป็นรูปแบบทั่วไปที่จะใช้กับเอเจนท์ได้หลายหลายประเภทเหมือนกับรูปแบบแรก



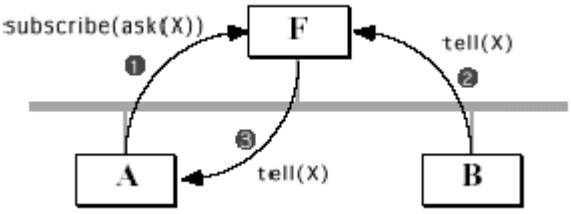
(ก)



(ข)



(ค)



(ง)

รูปที่ 5. Facilitator รูปแบบต่างๆ : (ก) Recommender (ข) Broker (ค) Recruiter และ (ง) Subscriber [3]

4.5 สรุป

ในบทความแรกนั้นเป็นการนำเสนอฟังก์ชั่น Matchmaking ที่เป็นพื้นฐานอย่างง่ายๆ อย่างไรก็ตามข้อเสียที่สำคัญของวิธีนี้คือ เอเจนท์ทุกตัวต้องรู้จักกับเอเจนท์อื่นๆ ทั้งหมดในระบบ ซึ่งเป็นไปไม่ได้ในระบบขนาดใหญ่ ที่ๆ เอเจนท์ของหลายๆ องค์กร จากหลากหลายผู้ใช้ มาทำงานร่วม

กัน ในขณะที่ Middle Agent นั้นพยายามรวบรวมฟังก์ชั่นในการค้นหาบริการจากเอเจนท์มาไว้ที่เอเจนท์เพียงตัวเดียวซึ่งทำให้สามารถทำงานได้ในระบบดังกล่าว แต่มีข้อด้อยที่สำคัญคือขาดความสามารถในการขยายขนาด (Scalable) ดังนั้น MMA ประเภทนี้จึงเหมาะกับระบบขนาดเล็กมากกว่า สำหรับบทความที่ 3 นั้นได้พยายามกระจายงาน Matchmaking กลับไปยังเอเจนท์ทุกๆ ตัวเหมือนบทความแรก โดยเน้นข้อดีของการที่เอเจนท์แต่ละตัวไม่จำเป็นต้องรู้จักกับเอเจนท์ทุกตัวในระบบ เพียงแค่รู้จักบางตัวก็สามารถทำงานได้แล้ว และสามารถทำงานได้อย่างมีประสิทธิภาพภายใต้ระบบเครือข่ายขนาดใหญ่ อย่างไรก็ตามการทดลองซึ่งทดสอบหลักการของบทความนี้นั้นเป็นการพิสูจน์ทางคณิตศาสตร์ล้วนๆ และมีสมมติฐานมากมาย โดยที่สำคัญได้แก่ เอเจนท์แต่ละตัวจะรู้จักเอเจนท์ข้างเคียงจำนวน 4 ตัวเท่ากันหมดทั้งระบบ หากจำนวนเอเจนท์ที่เอเจนท์แต่ละตัวรู้จักเปลี่ยนแปลงมากขึ้น หรือมีจำนวนไม่เท่ากันในแต่ละเอเจนท์ก็จะทำให้ความซับซ้อนของอัลกอริทึมที่นำเสนอในบทความนี้มากขึ้น อย่างไรก็ตามบทความนี้ได้แสดงแนวคิดที่น่าสนใจประการหนึ่งคือพยายามจำลองการทำงานของระบบ DNS Protocol ซึ่งเป็นโปรโตคอลที่มีความคล้ายคลึงกับ MMA เข้ามาใช้กับฟังก์ชั่น Matchmaking ของเอเจนท์แต่ละตัว ซึ่งมีแนวโน้มสามารถทำให้การทำงานมีความรวดเร็วขึ้นได้ ซึ่งแนวคิดนี้จะถูกนำไปศึกษาต่อไปเพื่อพัฒนาระบบ MMA ที่มีความสามารถทำงานได้ในระบบขนาดใหญ่

5. ปัญหาของ MMA ในปัจจุบัน

จากการศึกษาพบว่าระบบ MMA ที่ได้นำเสนอมาในอดีตยังไม่สามารถที่จะนำไปใช้งานได้จริงในระบบเครือข่ายขนาดใหญ่ โดยเฉพาะอย่างยิ่งอินเตอร์เน็ท โดยมากแล้วจะเหมาะกับการนำไปใช้กับระบบ MAS ขนาดเล็ก ที่จำกัดอยู่เฉพาะภายในองค์กรใดองค์กรหนึ่ง ซึ่งเมื่อเป็นเช่นนั้นแล้วปัญหาในการค้นหาเอเจนท์ก็ไม่น่าจะเป็นปัญหาที่สำคัญอีกต่อไป ดังนั้นการที่จะนำแนวคิด MMA มาใช้ได้จริงนั้นจึงควรมีความสามารถในการนำมาใช้กับระบบเครือข่ายขนาดใหญ่ได้ โดยมีประสิทธิภาพดีเพียงพอ และกินทรัพยากรของระบบไม่มาก

6. แนวทางแก้ไข

เนื่องจาก Shehory [10] ได้นำเสนอการพัฒนาอัลกอริทึมสำหรับใช้ในงาน Matchmaking ระหว่างเอเจนท์ โดยการจำลองระบบการทำงานบางส่วนของ DNS มาใช้ในรูปแบบการสื่อสารเพื่อแลกเปลี่ยนข้อมูลที่ใช้ในการทำ Matchmaking ภายในระบบ MAS โดยได้แสดงให้เห็นว่าสามารถนำไปใช้ในระบบขนาดใหญ่ได้อย่างมีประสิทธิภาพ อย่างไรก็ตามวิธีการพิสูจน์ของงานวิจัยนี้ยังอ่อนมาก เนื่องจากไม่มีการจำลองหรือทดสอบแนวคิดดังกล่าวกับเอเจนท์เลย ประกอบกับเขาได้เสนอให้หน้าที่ Matchmaking นั้นมีในเอเจนท์ทุกตัว ซึ่งในความเป็นจริงแล้วเป็นไปได้ยากยิ่งเนื่องจากไม่มีแรงจูงใจมากพอที่จะมีผู้ใดยอมให้ใส่ฟังก์ชั่นดังกล่าวลงในเอเจนท์ของตน ดังนั้นจึงควรใช้ MMA ทำหน้าที่ดังกล่าวเหมือนเดิม โดยทำงานร่วมกันเป็นกลุ่ม และจัดให้มีลักษณะการสื่อสารแลกเปลี่ยนข้อมูลเกี่ยวกับบริการและที่อยู่ของเอเจนท์ที่คล้ายคลึงกับ DNS คือแบ่งเป็นลำดับชั้นของเอเจนท์ดังกล่าว (Hierarchy of MMA) แทน และจะเรียกแนวนี้ว่า DMAA (Distributed Matchmaking Agent Architecture)

7. สรุป

ปัจจุบันมีงานวัจัยเพื่อแก้ปัญหาการค้นหาเอเจนท์ที่สามารถให้บริการที่ต้องการได้แล้ว ซึ่งที่นิยมกันได้แก่ แนวคิด MMA นั่นเอง อย่างไรก็ตาม ส่วนใหญ่แล้วแนวคิดที่นำเสนอมาในอดีตนั้นขาดความสามารถในการขยายขนาด ทำให้ไม่สามารถนำไปใช้ได้จริงในระบบขนาดใหญ่ ดังนั้นจึงควรมีการพัฒนาโครงแบบ MMA แบบใหม่เพื่อแก้ไขปัญหาดังกล่าวต่อไป

แนวทางหนึ่งที่มีความเป็นไปได้คือการพัฒนาระบบกระจายของ MMA ขึ้น โดยแนวคิดที่จะถูกพัฒนาขึ้นนี้เรียกว่า DMAA

เอกสารอ้างอิง

[1] Cost, R. S., et al. "TKQML: a scripting tool for building agents." in Wooldridge, M., Singh, M., and Rao, A. (eds.); Proceedings of the Fourth International Workshop on Agent Theories, Architectures and Languages, pp. 336-340, Berlin, Springer-Verlag, 1997.

[2] Decker, K. et. al. "Matchmaking and brokering." in Proceedings of the Second International Conference in Multi-Agent Systems, Kyoto, 1996.

[3] Finin, T. and Fritzson, R. "KQML as an agent communication language." in Proceedings of the Third International Conference on Information and Knowledge Management, 1994.

[4] Frankin, S. and Graesser, A. "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, 1996.

[5] Fuggetta, A., Picco, G. P., and Vigna, G. "Understanding code mobility." in IEEE Transactions on Software Engineering, Vol. 24, 1998.

[6] Gruber, T. R. "A translation approach to portable Ontology specification." in Knowledge Acquisition, Vol. 2, pp. 199-220, 1993.

[7] Jennings, N. R., Sycara K., and Wooldridge M. J. "A roadmap of agent research and development." in Autonomous Agents and Multi-Agent Systems, Vol.1, pp. 275-306, 1998.

[8] Kuokka, D. and Harada, L. "Matchmaking for information agents." 91-97. in Huhns, M. N. and Singh, M. P. in Readings in Agents, San Francisco, Morgan Kaufmann, 1998.

[9] Labrou, Y., Finin, T., and Peng, Y. "Agent communication language." in IEEE Intelligent Agents, March/April, pp. 45-52, 1999.

[10] Shehory, O. "A scalable agent location mechanism." in Intelligent Agents VI - Proceedings of 6th International Workshop on Agent Theories, Architectures, and Languages (ATAL'99), Vol. 1757 of Lecture Notes in Artificial Intelligence, pp. 162-172, Heidelberg, Springer-Verlag, 2000.

[11] Sycara, K. et. al. "Matchmaking among heterogeneous agents on the internet." in Proceedings AAAI Spring Symposium on Intelligent Agents in Cyberspace, Stanford, 1999.

[12] Vivacqua, A. S. "Agents for expertise location." in Proceedings of the AAAI Spring Symposium on Intelligent Agents in Cyberspace, Stanford, 1999.