

เนื้อหาการบรรยายครั้งที่ 2

- ระบบมัลติเอเจนต์ (Multi Agent Systems)
- เอเจนต์แบบเคลื่อนที่ (Mobile Agent)

ระบบมัลติเอเจนต์

MULTI AGENT SYSTEMS (MAS)

การประยุกต์ใช้

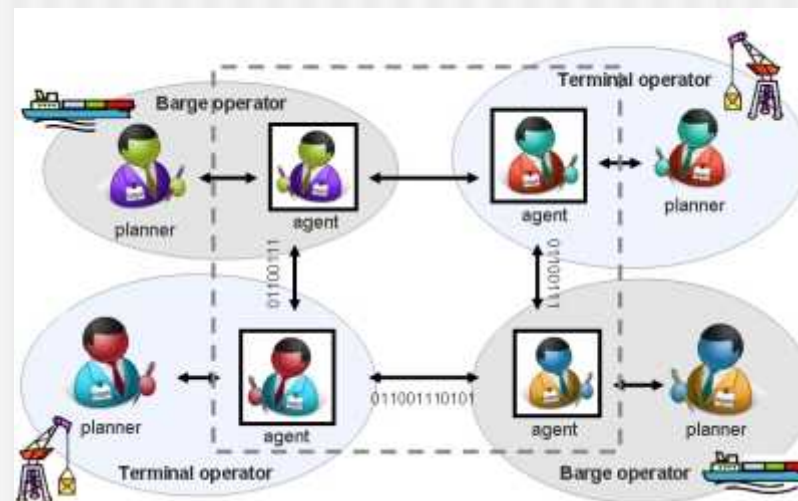
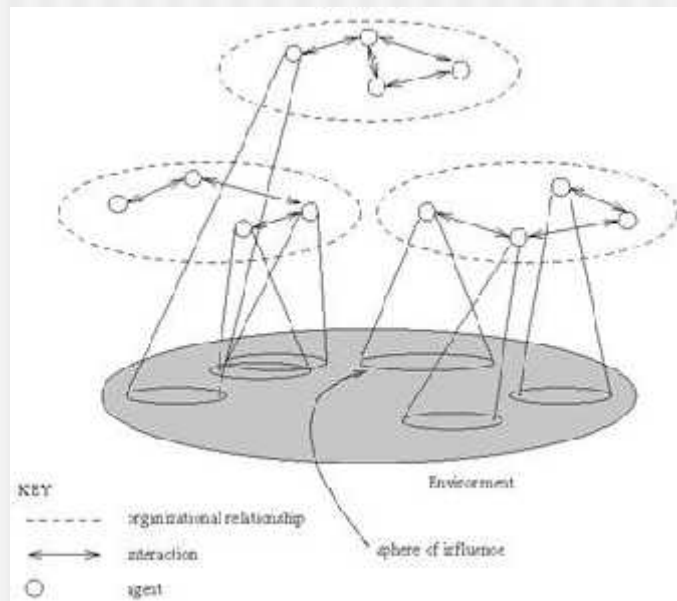
โครงสร้างของ MAS

รูปแบบการทำงาน

การสื่อสารระหว่าง AGENT

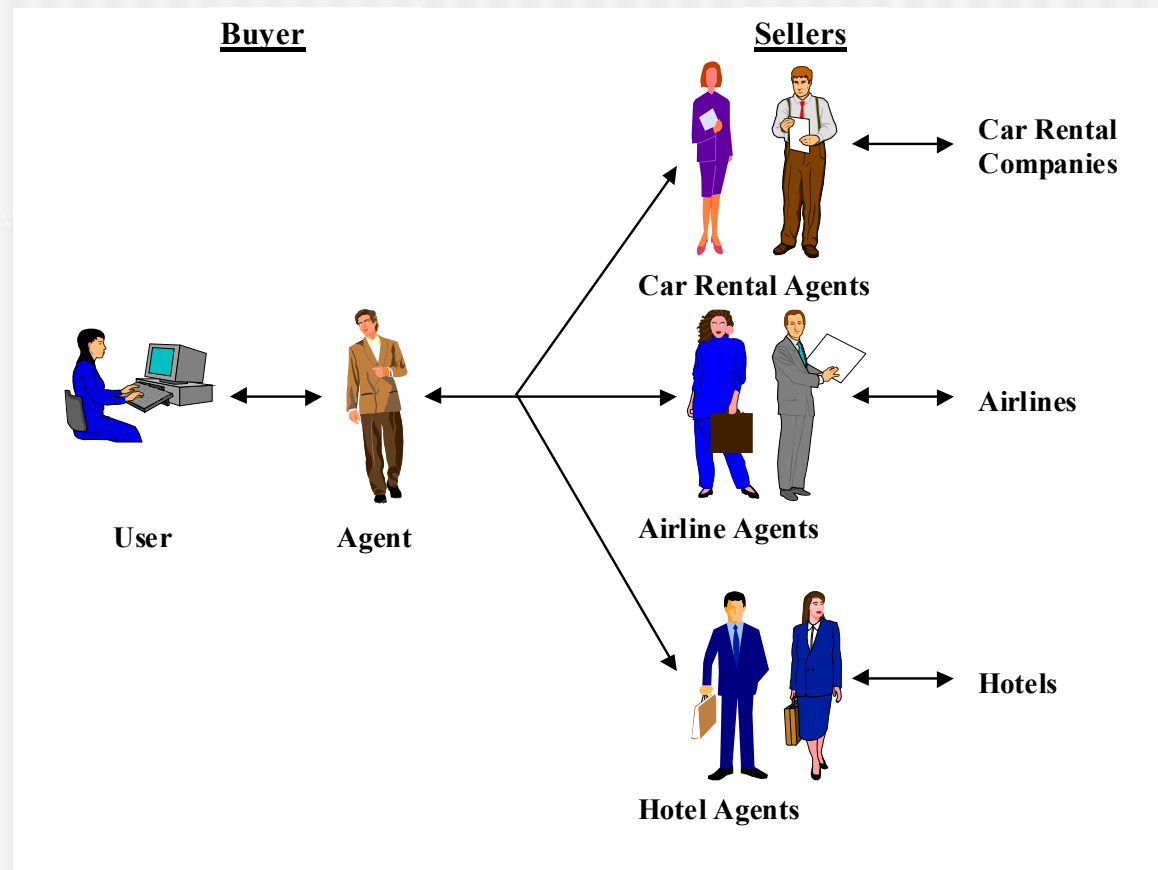
Multi Agent System (MAS)

- agent หลายตัวทำงานร่วมกันเป็นระบบ
- แต่ละ agent ทำงานตามหน้าที่ของตนอย่างอิสระ
- agent แต่ละตัวต้องสามารถโต้ตอบสื่อสารระหว่างกันได้
- distributed control + distributed data



ตัวอย่าง MAS

- A multi agent system for traveling arrangements



การประยุกต์ใช้ MAS

- peer-to-peer program (bittorrent)
- web application
- social networks
- network application (router, system monitoring)
- just-in-time manufacturing and procurement
- ad auctions
- negotiations among self-interested parties
- sensor network

ทำไมต้อง MAS

- ช่วยแก้ปัญหาที่ใหญ่เกินไปสำหรับ single agent / centralized agent
- อนุญาตให้ระบบสามารถทำงานโดยการเชื่อมต่อและเข้ากันได้ (interconnection & interoperation)
- ช่วยให้การออกแบบมีความชัดเจนและเรียบง่าย (clarify & simplicity)
- เหมาะสำหรับการแก้ปัญหาที่ในระบบที่ทำงานอย่างกระจายตัว (distributed system)

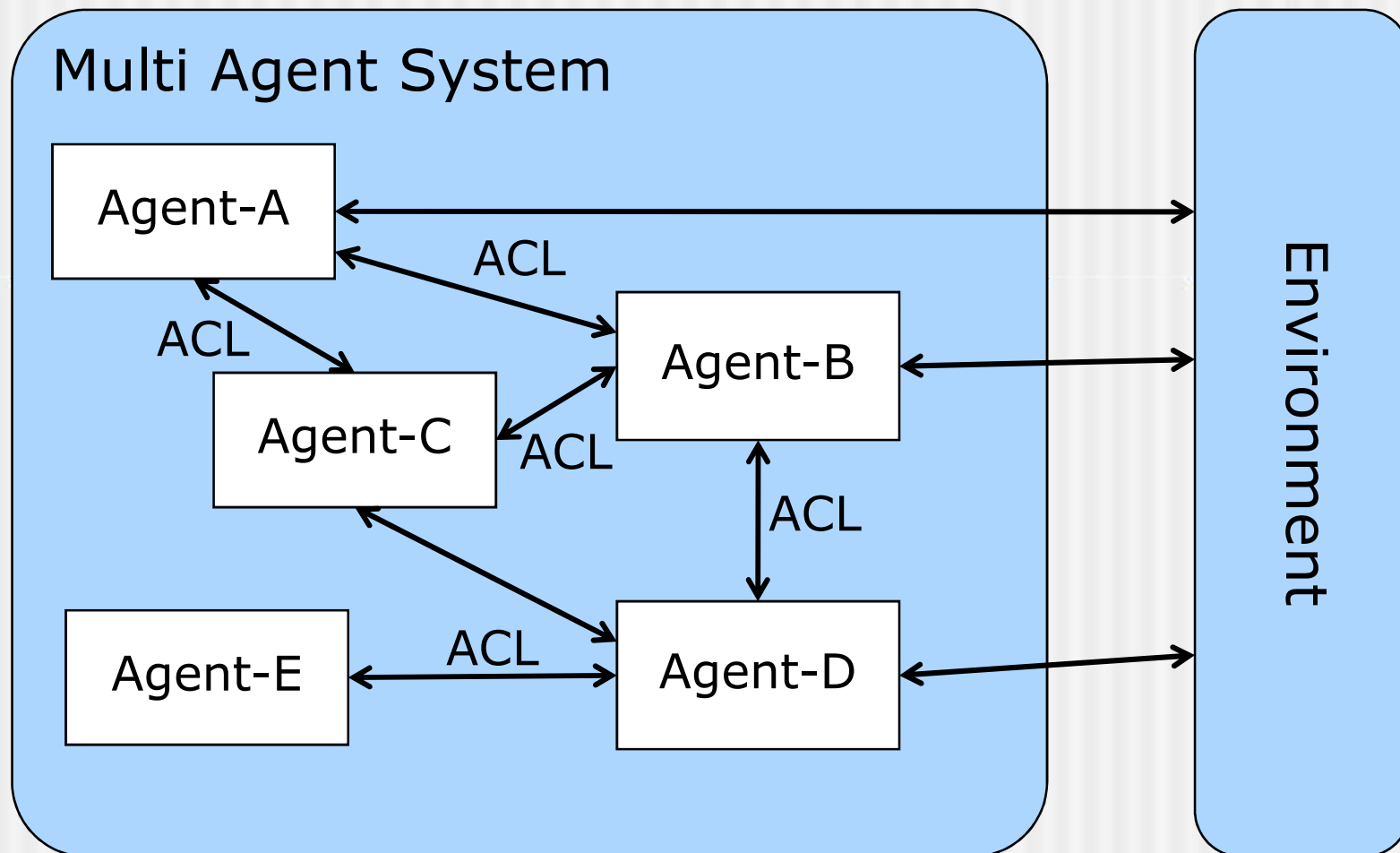
ข้อดีของ MAS

- แก้ไขปัญหาได้เร็วขึ้น
- มีความยืดหยุ่นมากขึ้น
- เพิ่ม reliability
- ลดการสื่อสารหรือติดต่อกันผ่านตัวแปลภาษา/ภาษาเดียวกัน
(สื่อสารโดยใช้ภาษา ACL แทน)

วิวัฒนาการของ MAS

- 1970s AI Boom
- 1980s AI Bust, Distributed AI (DAI– concerning with problem solving where agent solve sub-tasks)
- 1990s The Web, Multi Agent Systems
- 2000s Ad auction, Social Networks, web application

โครงสร้างของ MAS



รูปแบบการทำงานของ MAS

- Cooperation - agent ที่ทำงานร่วมกัน เช่น ระบบขับรถอัตโนมัติ ระบบควบคุมไฟจราจร
- Competition - agent ที่ทำงานแบบแข่งขันกัน เช่น เกมหมากรุก
- Semi-competition - agent ที่ทำงานแบบกึ่งแข่งขันกัน เช่น การหาเส้นทางที่เหมาะสม
- Negotiation agents - agent ที่มีการเจรจาต่อรอง เช่น e-auction

การสื่อสารระหว่าง agent

- การสื่อสารระหว่าง agent ใช้ภาษาโดยเฉพาะเรียกว่า Agent Communication Language (ACL)
 - ภาษา KQML
 - ภาษา XML-ACL
 - ภาษา FIPA-ACL (www.fipa.org)



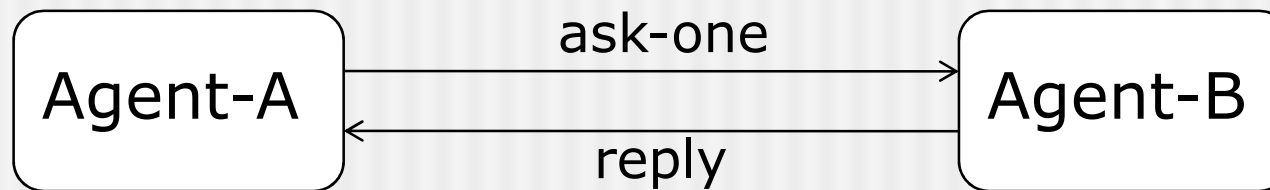
ภาษา KQML

- Knowledge Query and Manipulation Language
- บอกวัตถุประสงค์ให้กับ agent ที่กำลังติดต่อให้ทราบความต้องการ
- แลกเปลี่ยนความรู้/ข้อมูลระหว่าง agent
- รูปแบบของภาษา KQML

(*KQML Performative* [*tell* | *ask-one* | *reply* | *sorry*]

: sender	ชื่อ agent ที่ส่งคำร้องขอ
: receiver	ชื่อ agent ที่รับคำร้องขอ
: content	เนื้อหาที่ต้องการร้องขอ
: language	ภาษาที่ใช้สื่อสารใน content เช่น Java, prolog
: ontology	ข้อตกลงเพื่อแลกเปลี่ยนข้อมูล)

ตัวอย่างของภาษา KQML



```
( ask-one  
  : sender   Agent-A  
  : receiver  Agent-B  
  : content  (price = book.Price("AI Book"))  
  : language  Java  
  : ontology  Bath-Price )
```

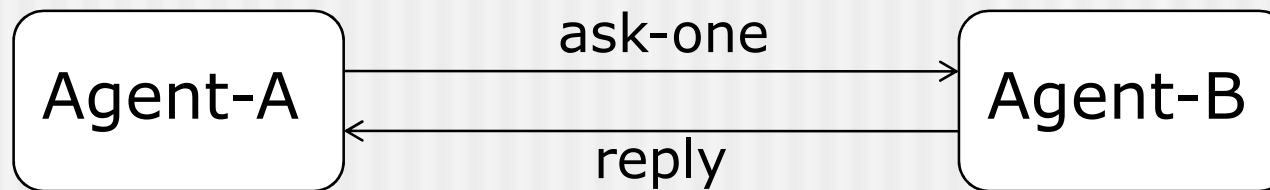
```
( reply  
  : sender   Agent-B  
  : receiver  Agent-A  
  : content  (price = 420)  
  : language  Java  
  : ontology  Bath-Price )
```

ภาษา XML-ACL

- eXtensible Markup Language-Agent Communication Language
- คล้ายกับภาษา KQML แต่ใช้ tag ในการระบุรายละเอียด
- รูปแบบของภาษา XML-ACL

```
<msg>  
  <performative>...</performative>  
  <sender>... </sender>  
  <receiver>...</receiver>  
  <content>... </content>  
  <language>... </language>  
  <ontology>... </ontology>  
</msg>
```

ตัวอย่างของภาษา XML-ACL



```
<msg>
  <performative>ask-one</performative>
  <sender>Agent-A</sender>
  <receiver>Agent-B</receiver>
  <content>price = book.Price("AI Book")</content>
  <language>Java</language>
  <ontology>Bath-Price</ontology>
</msg>
```

```
<msg>
  <performative>reply</performative>
  <sender>Agent-B</sender>
  <receiver>Agent-A</receiver>
  <content>price = 420</content>
  <language>Java</language>
  <ontology>Bath-Price</ontology>
</msg>
```

เอเจนต์เคลื่อนที่

MOBILE AGENT

รูปแบบการทำงาน

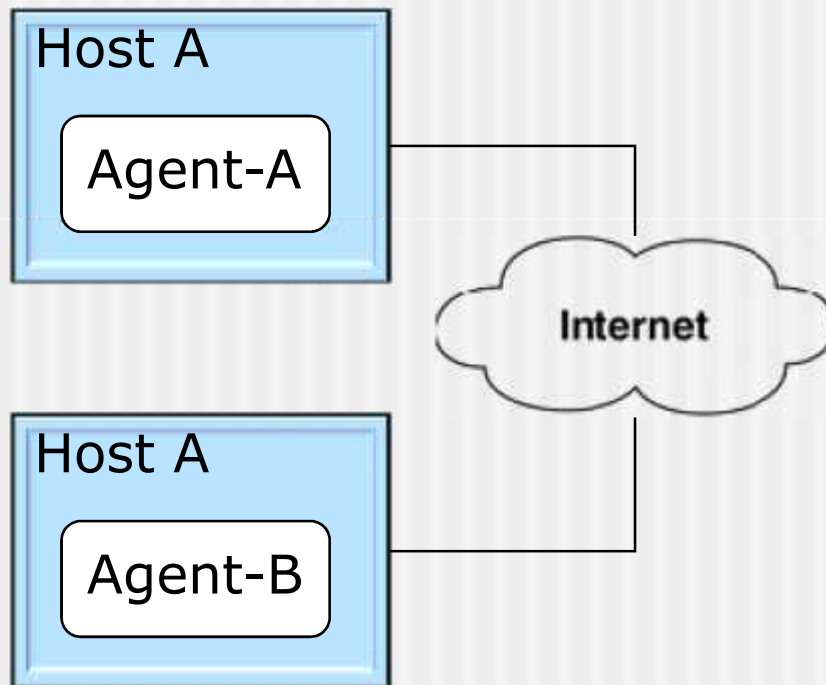
ตัวอย่าง

Mobile Agent

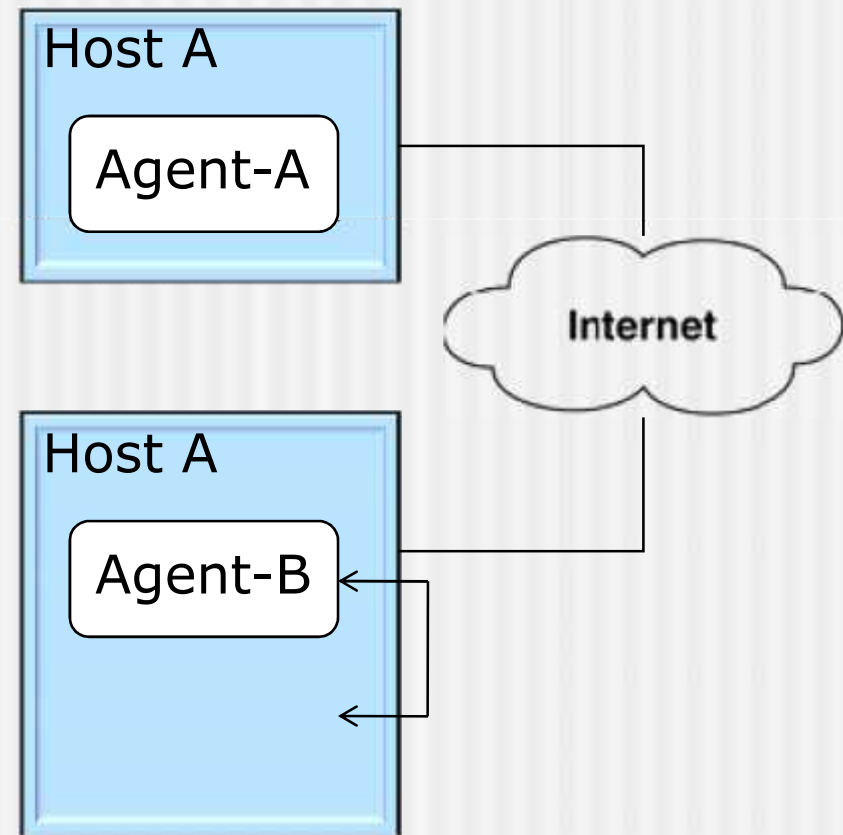
- เทคโนโลยีที่รวมเอาความสามารถของ agent กับความสามารถในการเคลื่อนที่เข้าด้วยกัน
- agent ที่สามารถจำลองตัว และเคลื่อนย้ายจากเครื่องหนึ่งได้
- ไม่ขึ้นกับระบบสถาปัตยกรรมใด
- เดินทางไปตามเครือข่ายเพื่อทำภารกิจที่ได้รับมอบหมายในเครื่องต่างๆ แล้วส่งผลลัพธ์กลับมา
- IBM aglet = mobile agent + applet

Stationary agent vs. mobile agent

Stationary agent



Mobile agent



ทำไมต้อง mobile agent

ข้อดี

- ลดการใช้ทรัพยากรที่สิ้นเปลือง ทำให้ host ใช้เวลาไปทำงานอื่นได้
- ลดการติดต่อข้ามเครือข่าย ทำให้ลดการจราจรของเครือข่ายลง
- เพิ่มความเสถียรให้กับโปรแกรมเมื่อใช้งานบนเครือข่ายที่ไม่น่าเชื่อถือ
หากเกิดปัญหาในการเชื่อมต่อเครือข่าย agent ก็ยังทำงานได้

ข้อเสีย

- host แต่ละตัวต้องสร้าง runtime environment ให้ agent รันโค้ดได้
- ในแง่ความปลอดภัย ต้องระมัดระวังโค้ดที่ไม่ประสงค์ดี

รูปแบบของ mobile agent

- **Traveling** - กำหนดเส้นทางให้ agent เดิน หรือบอก Host ทั้งหมดให้ agent แล้วให้ agent ตัดสินใจเลือกทางเอง
- **Task** - แยกงานใหญ่ให้เป็นงานเล็กๆแล้วให้ agent ตัวเดียวจัดการโดยย้ายไปเรื่อยๆ หรืออาจให้ Master agent สร้าง Slave agent หลายตัวช่วยกันทำก็ได้
- **Interaction** - จัดการให้ agent ตั้งแต่ 2 ตัวไปพบกันที่ Host หนึ่งแล้วปฏิสัมพันธ์กัน หรือจัดหาพื้นที่เก็บข้อมูลบน Host แล้วให้ agent เอาข้อมูลมาเก็บไว้เหมือนเป็น Locker หรืออาจให้ agent นำพาข้อความไปยังที่ต่างๆ หรือให้ agent ตัวหนึ่งเดินทางไปตาม server ต่างๆ เพื่อหา agent ที่ต้องการเพื่อจะปฏิสัมพันธ์ หรืออาจจะเป็นการสร้างกลุ่มของ agent ที่ทุกตัวรู้จักกันแล้วให้เดินทางไปทั้งหมด

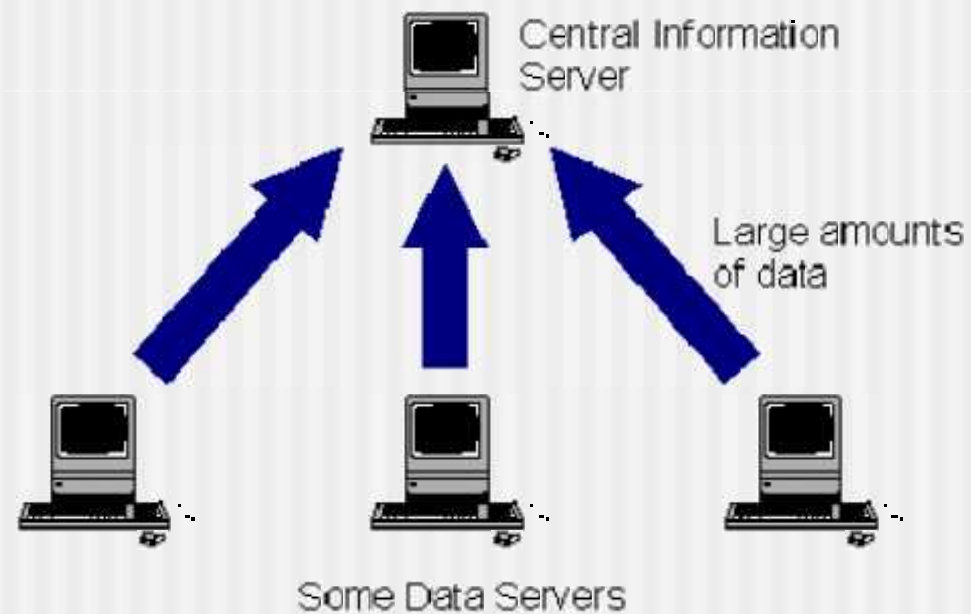
ตัวอย่าง mobile agents

งานรวบรวมข้อมูล

- Search engine ที่รวบรวมข้อมูลจาก web server ทั่วโลก
- Central information server ของบริษัทที่รวบรวมข้อมูลจากแต่ละแผนก
- Data mining กับฐานข้อมูลแบบกระจายที่มีขนาดใหญ่

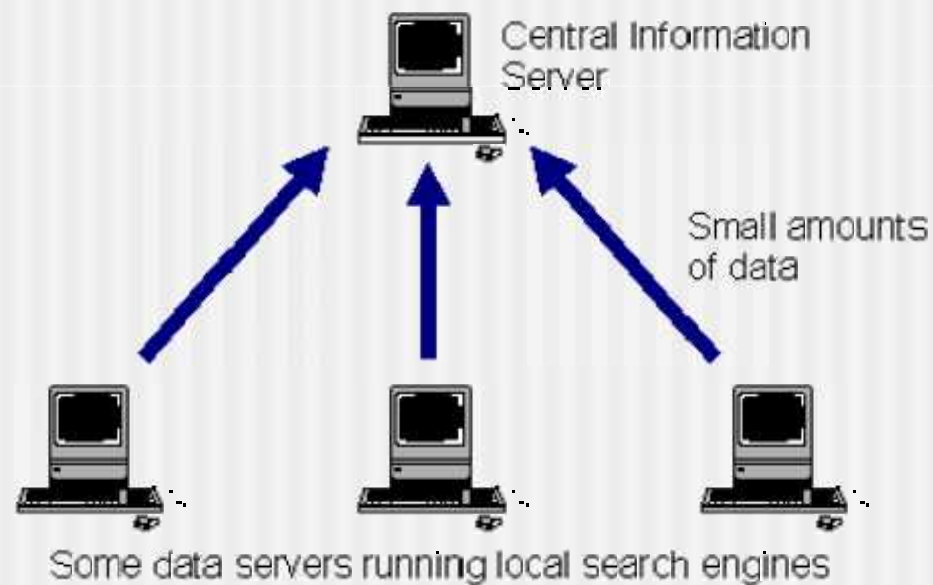
วิธีการแบบดั้งเดิม

Conventional Data Collection



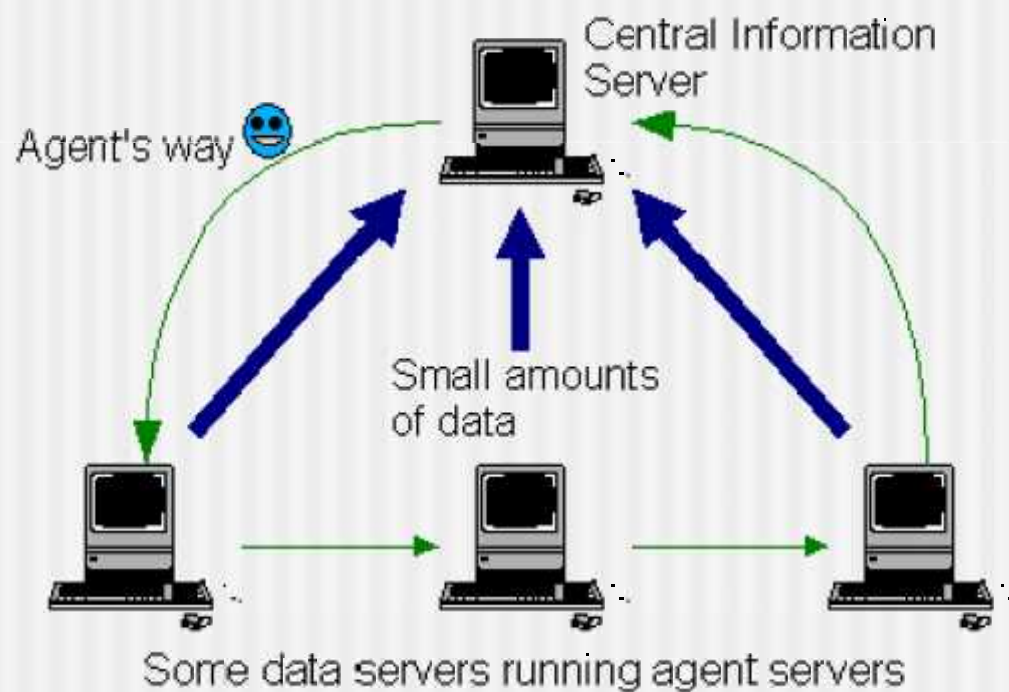
วิธีการแบบไม่รวมศูนย์

Decentralized Data Collection



วิธีการแบบ mobile agents

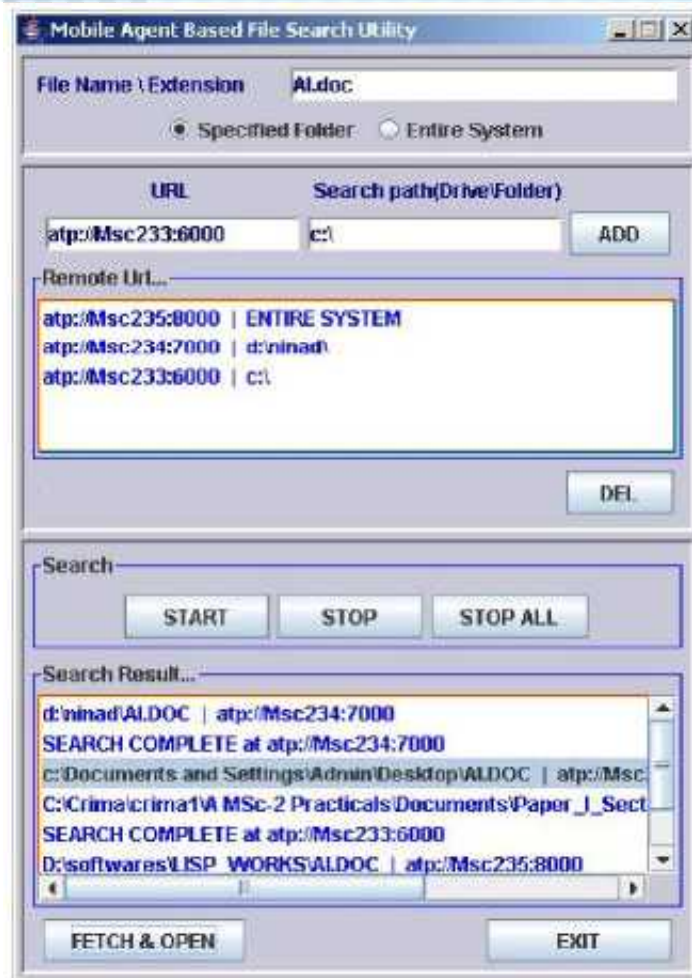
Data Collection with Mobile Agents



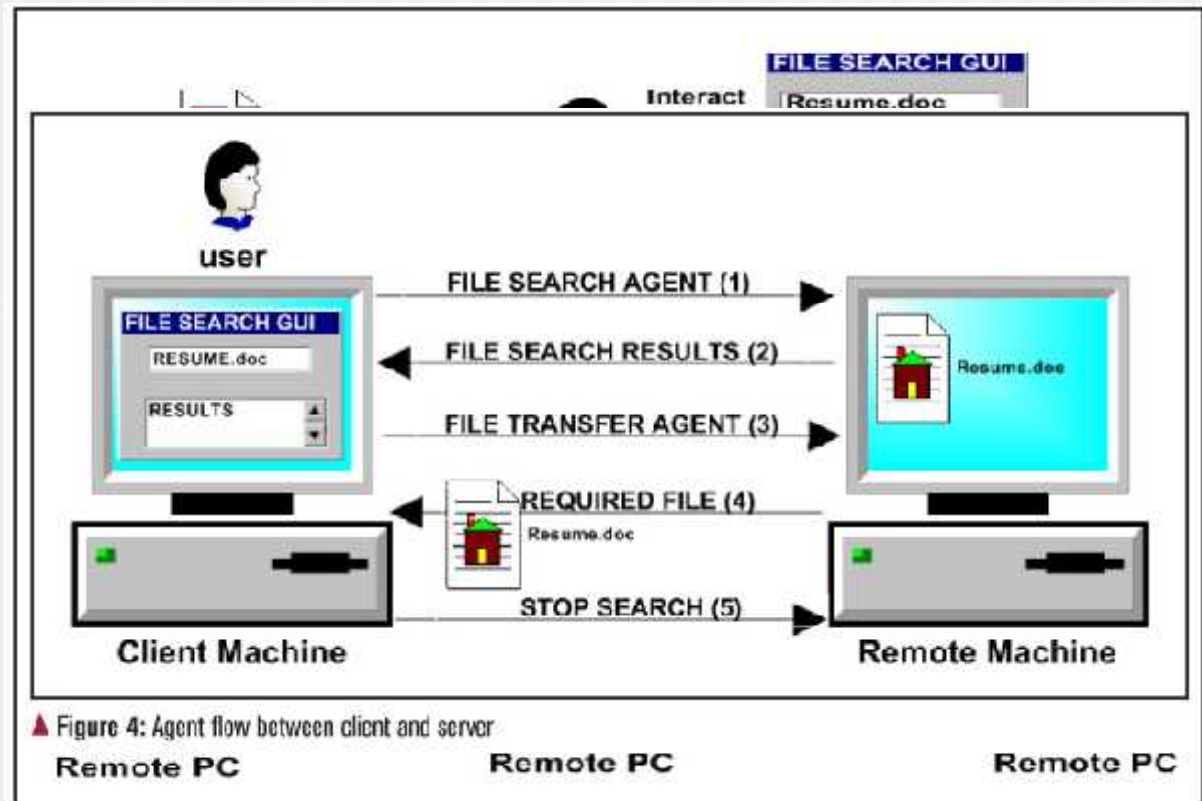
ตัวอย่าง mobile agents

- Mobile Agent-Based File Search Utility
- สร้างโปรแกรมที่สามารถค้นหาเอกสารที่ต้องการได้จาก remote PC หลายๆ เครื่องพร้อมกันได้
- ประกอบด้วย agent ทั้งหมด 3 ตัว
- Master – main agent จัดการ UI และสร้าง Slave 2 ตัว เป็น stationary agent
- Slave - file search agent เมื่อเริ่มค้นหา mobile agent นี้จะถูกสร้างขึ้นเพื่อทำการค้นหาสำหรับแต่ละ remote machine และส่ง path ที่เจอกลับไปให้ main
- Slave - file transfer agent เมื่อผู้ใช้ต้องการเปิดไฟล์ mobile agent นี้จะถูกสร้างขึ้นเพื่ออ่าน content ของไฟล์แล้วส่งกลับให้ main

Mobile Agent-Based File Search Utility



▲ Figure 3: User interface displaying the search results



▲ Figure 4: Agent flow between client and server

▲ Figure 1: Application structure

การประยุกต์ใช้ตัวแทนปัญญา

- Internet-based software
- E-commerce
- Operating system agent
- Information management
- User interface
- Spreadsheet agent
- IA in game
- ChatterBot

Internet-based software agents

- E-mail agents (mailbots)
- Web browsing assistant agents
- FAQ agents
- Intelligent search agents
- Internet softbot for finding information
- Network management and monitoring

E-commerce agents

- Need identification
- Product brokering
- Negotiation
- Purchase and delivery
- Product/service evaluation

Operating Systems Agents

Wizards ใน Microsoft Windows NT Operating Systems

- Add user accounts
- Group management
- Managing file and folder access
- Add printer
- Add/remove programs
- Network client administrator
- Install new modems

Information management

- Data mining – collecting data, query-and-reporting tools, analysis
- Web mining – resource discovery, information extraction
- Monitoring and alerting - personalized News Alert
- Workflow and administrative management – automate user needs/business process

ข้อควรคำนึงถึงในการสร้าง agent

- Cost justification
- Security
- Privacy
- Ethics
- Agent accuracy
- Agent learning
- System Technology

ข้อควรคำนึงถึงในการสร้าง agent

- พัฒนากฎสำหรับ agent ให้มีพฤติกรรมที่เหมาะสม
- พิจารณาความถูกต้องของสารสนเทศที่รวบรวมมา
- เคารพข้อกำหนดของผู้ให้บริการอื่นๆ
- ทำเท่าที่ได้รับสิทธิ/อนุญาต

สรุป

- agent ทำงานได้เป็นอิสระ ไม่ต้องการควบคุม
- agent สามารถประยุกต์ใช้ได้หลากหลาย ช่วยให้งานคอมพิวเตอร์ง่ายขึ้น
- agent มีทั้งแบบฉลาดและไม่ฉลาด สามารถทำให้ฉลาดได้โดยการเรียนรู้
- รู้จักสภาพแวดล้อมและคุณสมบัติของสภาพแวดล้อมของ agent
- agent สามารถทำงานร่วมกันได้ (MAS) และสามารถเคลื่อนย้ายได้ (Mobile agent)