

群體行動代理人商務環境之合作協商機制

Multi-Agent Cooperative Negotiation for Electronic Commerce based on Mobile Agents

林芳昌 郭建男
朝陽科技大學 資訊管理系
台中縣霧峰鄉吉峰東路 168 號
fclin@cyut.edu.tw

摘要

務(Electronic Commerce)、協商(Negotiation)

近年來，電子商務蓬勃地發展，各種研究與應用領域吸引了學者與專家投入研究。為了減少買賣雙方的交易成本，代理人(Agent)的技術被使用來發展自動化的商務環境。此外，行動代理人(Mobile Agent)的技術更賦予代理人程式在網路上自主移動的能力。本研究發展了以行動代理人為主之分散式交易的電子商場，在此環境下，買方指派代理人在網路上搜尋產品、並派遣代理人至供應商伺服器端與供應商的代理人交易，為了快速地完成交易，本研究使用多個代理人同時合作地完成交易。換句話說，買方根據供應商的數量同時指派多個代理人程式與供應商的代理人進行交易。我們已建構了以上的分散式交易的電子商場，其中行動代理人乃架構於 IBM 之 Aglet 工具所開發完成。

Over the past few years, electronic commerce has been developed prosperously. There have been a lot of research and applications on it. In order to reduce the trade cost between buyers and suppliers, agent technique has been utilized to develop the environment for automatic commerce. Besides, mobile agent technique provided the mobility for agents to travel the network. This paper developed an e-Marketplace system of distributed trade based on mobile agents. In this system, the buyers assigned agents to search the product information and dispatched agents to the supplier sides and traded with the supplier agents. In order to work efficiently, multiple buyer agents were utilized. The buyer assigned multiple agents to search and trade with the supplier agents simultaneously according to the number of supplier agents. We have implemented this e-Marketplace for distributed trade. The IBM Aglet toolkit were employed to implement the mobility of agents.

關鍵詞：行動代理人(Mobile Agent)、電子商

一、前言

網際網路的盛行改變了傳統的商業模式，買賣雙方可以透過 B2B 的交易模式來提高營運效率和節省成本。在商務系統發展的演進中，從 EDI 至網路商店的成立到目前 B2B 電子商場，皆是 B2B 的交易模式，也慢慢的縮小建置成本與交易成本。B2B 電子商場提供一個交易的場所與環境，包含了自動化的交易機制，使買賣雙方可以降低交易成本。

在 B2B 電子商務的發展中，如何整合企業間之供應鏈體系，使企業能減低庫存與交易自動化，一直是重要的研究課題。而代理人(Agent)提供了解決之道[4,5,8]。因此，以代理人(Agent)為基礎的電子商場的研究因應而生[6,7,10,11,14,16]，其目的在於減少使用者的介入而達到交易自動化。在此環境下，代理人代替買方在網路上搜尋產品、議價和完成交易；也幫助供應商行銷、議價。

在目前電子商場的架構中，討論最多的是集中式的交易市場，所有的商業活動均集中在同一伺服器中，包括尋找商品、提出需求、詢價、報價和協商等活動。但是網際網路是一個開放式的分散式環境，為了能在這樣的環境下有效的使用對方的服務，亦即能夠在供應商的環境進行交易，故分散式的交易市場也因應而生。在這樣的環境下，為了使代理人能夠根據設計的目的在網路中移動，行動代理人(Mobile Agent)的技術提供了更好的應用[12,15,17]。

針對上述的論點，本研究的目標在於利用 Mobile Agent 的技術來發展 B2B 電子商場系統，發展一個買方對應多個賣方架構下之非同步合作的協商機制。買方同時派遣多個代理人至賣方伺服器與賣方代理人進行協商，買方的代理人則彼此藉由交換詢價資訊來達到合

作的協商行為。

本研究的研究重點為代理人在 B2B 商務環境下之合作協商機制。本論文架構如下：第二節介紹 Mobile Agent 的概念與 IBM 所發展用來實作 Mobile Agent 的 Aglet；第三節則討論目前相關之以 Mobile Agent 為基礎之電子商場架構與協商機制的研究；第四節探討非同步之合作協商機制；第五節則描述本研究發展以 Mobile Agent 為基礎之非同步合作協商機制；第六節則是本系統的實作與模擬，並舉一個例子來介紹本系統的運作流程；最後一節則對本研究做一結論與建議。

二、行動代理人(Mobile Agent)

在 Client-Server 架構中，指的是在網路上需求端利用訊息傳遞向服務端要求服務，通常兩端點必需透過多次的交談來完成任務。而 Mobile Agent，簡單來說，是使代理人程式 (Agent) 具有移動能力的一種技術，在 D.B. Lange 等人(1998)的研究中，對行動代理人的定義如下：「行動代理人在執行時不會侷限在其電腦系統中，他能夠藉由網路把自己從一電腦系統中傳送到另一電腦系統。[12]」圖 1 顯示了 Client-Server 與 Mobile Agent 這兩種不同的概念。圖(a)為 Client-Server 的概念圖，兩端的應用程式透過網路傳遞訊息以完成任務；而在圖(b) Mobile Agent 的概念圖中，Host A 把任務程式 (Agent 1) 派遣至 Host B 與其應用程式交談，不須透過繁覆的訊息傳遞來完成任務。

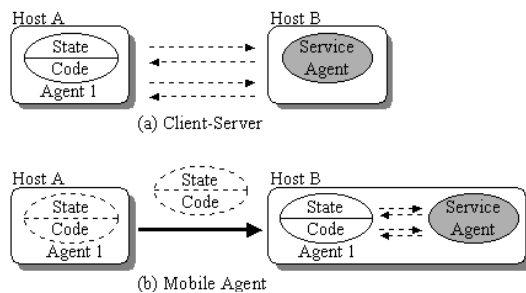


圖 1 Client-Server 與 Mobile Agent 運作模式

為了實作 Mobile Agent 的技術，選擇在網路上具有跨平台能力的程式語言是一件重要的事，JAVA 則提供了這樣的便利性，除了跨平台，安全機制使執行中的代理人程式受到保護；多執行緒 (multithread) 的能力也使多個代理人程式能平行處理；物件序列化 (object serialization) 則使代理人程式可以暫存目前的執行狀態，等到移動到目的地後繼續執行。

本研究所使用的 Mobile Agent 開發工具 - The Aglets Software Development Kit

(ASDK)，是由 IBM 公司所發展的，使用 JAVA 語言所設計來支援撰寫 Mobile Agent 程式的 JAVA 類別。它包含了 Aglet API、說明文件、Aglet 程式範例與 Aglet Server (稱作 Tahiti)，其專屬網站 [3] 提供了 Aglet 免費下載。

Aglet API 是由一些 JAVA class 和 interface 所組成，其功能包含產生 Mobile Agent (Aglet) 並操作它。其核心為 Aglet、AgletProxy 和 AgletContext 等類別；Aglet class 定義了 Aglet 的動作，包含建立 (create)、派遣 (dispatch)、複製 (clone)、關閉 (dispose)、取回 (retract)、暫停 (deactivate) 和恢復暫停 (activate)。程式設計師可以針對其需求來設計其行為動作。

AgletProxy class 定義了 Aglet 間溝通的方式，Aglet 間可透過此界面同步或非同步地傳遞訊息，就像行動電話一樣，程式設計師可以用來聯繫執行中的 Aglet，提供區域透明性 (location transparency)，也就是說程式設計師不用知道，也無法知道執行中的 Aglet 遷移動作的細節，只要透過其 Aglet Proxy 即可聯繫執行中的 Aglet。另外也提供了保護的功能，可以防止其它 Aglet 直接的存取，所有 Aglet 間的溝通必須透過此界面。

而 AgletContext class 則提供了 Aglet 工作的環境，當 Aglet Server 啟動後，會建立 Context，而後所有 Aglet 的建立與運作必須由 Context 來管理，包含分配識別碼 (identifier) 並初始化。

除了上述所列出的基本功能，Aglet API 還提供了 Message 類別讓我們可以管理及自定訊息的格式，並可以同步 (synchronous) 與非同步 (asynchronous) 地傳送訊息，透過 Aglet Proxy 即可將訊息傳送給對方。

圖 2 說明了 Aglet 的架構與功能，Aglet 1 建立了 Aglet 2 並派遣 Aglet 2 到遠端的主機上，彼此間透過 Proxy 來傳遞訊息，在遠端主機上的 Aglet 2 可以暫停、返回或結束自己。

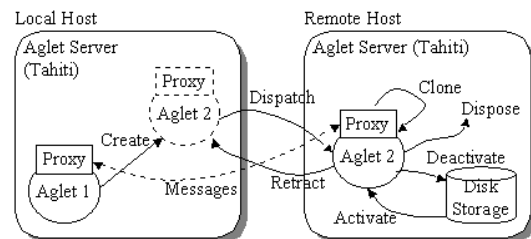


圖 2 Aglet 的架構與功能

本研究基於 Aglet 容易取得並完全免費，且比較容易入門，還有其說明文件比較詳細清楚；另外，其 Aglet Server 提供了完整的

使用者介面，並可在同一台電腦建立多個 Aglet Server，方便使用者能在同一台電腦設計與偵錯，故本研究採用 Aglet 來作為發展「行動代理人」的工具。

三、以 Mobile Agent 為主的電子商場架構

表 1 比較了代理人技術在 Client-Server 與 Mobile Agent 架構下的特性。目前成熟電子商務系統皆為 Client-Server 的架構，如 e-Bay、Kasbah，買賣雙方皆透過 Web 介面來設定其代理人程式。而 Mobile Agent 則提供了使用者更有彈性的系統設計。以下將針對 Mobile Agent 的架構加以介紹。

表 1 Client-Server 與 Mobile Agent 特性

網路架構	特點
以 Client-Server 為主的電子商場	(1) 代理人程式分散在買賣雙方系統，或 (2) 由電子商場系統替買賣雙方產生代理人程式(使用 Web 來設定)。 (3) 若訊息傳遞多，則佔網路頻寬。 (4) 成熟的網路架構，穩定性高。
以 Mobile Agent 為主的電子商場	(1) 應用程式隨著任務需求而改變工作地點，動態地改變程式執行的環境，程式設計彈性高。 (2) 買賣雙方可在同一伺服器中交談，減低雙方在網路上的訊息傳遞。 (3) 支援離線操作，減少使用者連線成本。 (4) 增加監控代理人程式的成本。 (5) 目前沒有統一的技术標準。

在 Mobile Agent 的技術相關研究中，L. Esmahi 等人(1999)提出了一個以 Mobile Agent 為主的電子商場[7]，買方與供應商均在這電子商場進行交易。作者提出的機制中，主要是透過一個集中式的系統來充當買賣雙方代理人程式的協調者，促使其達成交易。在其系統中的電子商場架構大致上將顧客的代理人程式分成 Buyer agents 及 Seller agents 二個角色，而系統中主要由一個 Manager Agent 來負責控制與協調，而 Manager Agent 控制與協調的活動行為則是由系統中建置的一些 Rules base、Cases base、及 Common services 來決定。

另外，在 T. Sandholm 等人(2000)所提出 Nomad 系統中[16]，作者所提出 Nomad 系統是一個能讓買賣雙方進行拍賣(Auction)動作的集中式交易架構，這系統的設計是先由供應

商登入商品資訊，然後由買方廠商的代理人程式與系統對商品進行拍賣的動作。這系統除了能讓買方廠商派遣代理人程式來進行交易，也允許買方廠商利用 Applet 來設定代理人程式，由系統幫買方廠商產生代理人程式來進行交易。

上述研究主要是建構一個集中式交易的商務系統，買賣雙方均可派遣代理人至此進行交易。集中式交易的優點在於管理方便，但是網際網路是一個開放的分散式環境，為了有效的使用對方的服務，亦即能夠在供應商的環境進行交易，故分散式的交易市場也因應而生。

於是，在 P. Dasgupta 等人(1999)的研究中[6]，提出了買方派遣代理人到賣方的伺服器進行交易，而不在統一的電子商場進行商務活動，在其所提出的架構中，買方的代理人根據供應商的資訊移動到每個供應商的伺服器，與供應商的代理人程式對其產品進行協商，若覺得條件滿意的話，可以進行預購的動作，否則就放棄交易，與這個供應商結束交易後，再到下一個供應商進行交易，等到所有的供應商均完成交易後，回到原始伺服器，將已預購的供應商交由使用者做最後的決策。

根據 P. Dasgupta(1999)的研究，他們提出一個分散式的交易的電子商場，即在供應商的伺服器端進行交易，這樣的利基點在於供應商能夠快速地與後端資料庫連繫，增加查詢商品資訊、庫存資訊與後勤系統的生產排程等工作。本研究的系統架構建立在這樣的基礎上，為了改善交易的效率，本系統發展買方同時指派多個買方代理人的與供應商交易，建立多個代理人合作的協商機制。

四、非同步之合作協商機制

在 B. Limthanmaphon 等人(2000)的研究中，描述了進行買賣雙方進行協商所需要的程序[13]：

- (1) 定義所要協商的產品屬性，如價格、品質等，各屬性的值與範圍設定。
- (2) 針對所發展的系統架構建立其協商方式，例如，在分散式的網路架構中，其協商方式要支援其架構。
- (3) 建立所有協商參與者的協商流程，使能夠符合買賣雙方的需求。

根據上述的論點，可以分成兩個重點，協商的網路架構與協商的決策行為模式，本研究的重點即為協商的網路架構，建構一個以 Mobile Agent 為基礎的協商機制。

在人類的實際世界中，若雙方的認知或需求不同，便會產生協商行為來達成共識。若在商務的活動中，協商則是藉由買賣雙方的交換意見，來達到彼此的同意點，以完成交易。在協商架構的研究中，Robert Guttman 等人(1998)介紹幾種協商架構，如一對多(單一買方對應多個賣方)、多對多(多數個買方與賣方)、股票市場的撮合方式(具有協調者)、拍賣方式(單一賣方對應多個買方)等[9]。通常，為了能得到更好的利益，買方會與多個賣方進行比價、協商來完成交易，在目前的協商模式中，多採用一對一的協商，在各個賣方經由循序的協商來完成交易。若要同時與多個賣方進行交易，則須要整合各個賣方的協商條件，才能進行一對多的協商模式。圖 3 顯示目前常使用的一對多協商模式。

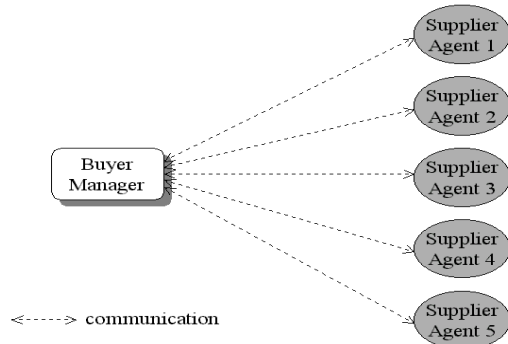


圖 3 單一買方對多個賣方的協商模式

在本研究的架構中，買方同時派遣多個代理人至每一個供應商中進行交易，為了能使每一個買方代理人能夠知道其它供應商的報價資訊，故每個買方代理人必須透過交談才能獲得其它供應商的報價資訊，這樣的架構下，所有的買方代理人必須合作交換資訊才能進行協商，故本研究稱此種協商方式稱為合作的協商機制(Cooperative Negotiation)。圖 4 顯示了本研究所採用的合作協商機制。

在協商進行時，可分為同步與非同步兩種協商模式。同步的協商模式意即每個被派出去的買方代理人皆收到其它買方代理人的一次詢價資訊即做一次的評量，例如(見圖 4)，當 Buyer Agent 1 向 Supplier Agent 1 詢價後，也都收到了其它 Buyer Agent 的詢價資訊就進行一輪的決策評量；而非同步協商模式就是當 Buyer Agent 1 詢價後，等待一段時間後，不管其它的 Buyer Agent 是否傳送來其詢價資訊，即自行做決策評量。採用非同步合作協商模式可使代理人具有自主性，能迅速的進行決策，不受網路頻寬或是其它 Buyer Agent 在它方發生故障的影響，具有容錯能力。但相對的，非同步合作協商模式中每個買方代理人的決策

結果不盡相同，協商結束後必須有一決策的機制來決定最後的提案選擇。

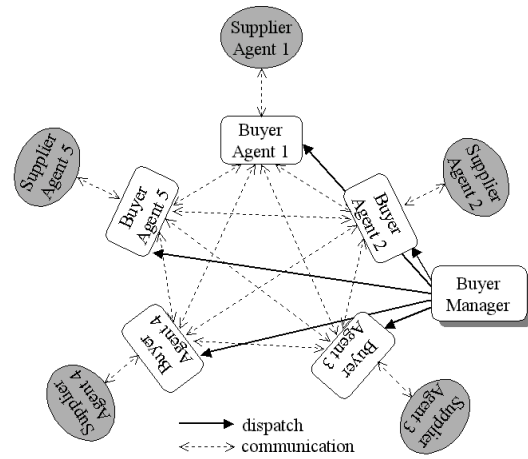


圖 4 本研究所採用的合作協商機制

五、系統架構

根據第二節到第四節對 Mobile Agent 與其在商務環境下的非同步合作協商機制的探討，本研究採取以 Mobile Agent 為主之分散式交易的電子商場架構，並採取同時指派多個代理人來完成交易的模式，圖 5 為本研究所設計之電子商場架構概圖，主要由買方、商務資訊中心及供應商所組成，共有三種主要的代理人元件：Buyer Manager、Buyer Agent 與 Supplier Agent。

Buyer Manager 是買方的主系統，其流程為：(1)向商務資訊中心詢問供應商的資訊後，(2)建立 Buyer Agent，並(3)派遣至供應商伺服器與(4)Supplier Agent 進行交易與協商，Buyer Agent 在協商的過程中彼此藉著(5)交換供應商的報價資訊以獲得完整的資訊以決策，(6)在協商進行時，也定時地將目前的狀態傳回給 Buyer Manager，也包括最後的協商結果。當協商結束後即完成交易。以下將對這三個元件詳細的介紹。

(一) Buyer Manager

Buyer Manager 可以說是買方的採購主系統，使用者在此設定採購需求與決策參數，然後系統根據需求去商務資訊中心(Commerce Information Center)找到符合需求的供應商資訊，根據供應商的數量派遣相同數量的 Buyer Agent 至供應商伺服器端與供應商的代理人(Supplier Agent)進行交易。在交易進行時，Buyer Manager 會收到 Buyer Agent 每次詢價、決策的結果，讓使用者可以監看交易的進行。圖 6 描述了 Buyer Manager 元件的虛擬碼。

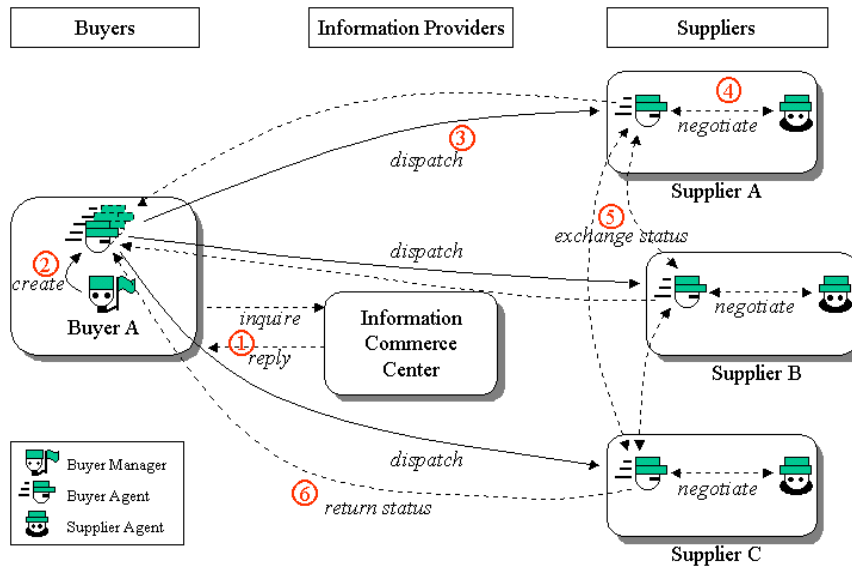


圖 5 本研究所設計之電子商場架構概圖

```
//Class for the buyer manager
import com.ibm.aglet.*;
public class BuyerManager extends Aglet {
    public void onCreation(Object init) {
        create the operative window;
    }
    public void queryDatabase() {
        query the supplier's information from the
        Commerce Information Center;
    }
    public void createBuyerAgent() {
        create buyer agents according to the amount of
        the suppliers.
        deliver the information of all buyer agents,
        products, decision model, and supplier's
        information to each buyer agent;
    }
    public void dispatchBuyerAgent() {
        dispatch buyer agents to their respective supplier;
        send the proxy of all buyer agents to each
        dispatched buyer agent;
    }
    public void sendCommand(String command) {
        send the command message to each buyer agent;
    }
    public boolean handleMessage(Message msg) {
        if buyer agents send the current status
        record and show the status;
        if buyer agents send supplier's quotation
        record and show the quotation;
        if buyer agents send the decided result
        record and show the decided best supplier;
    }
}
```

圖 6 Buyer Manager 元件的虛擬碼

(二) Buyer Agent

Buyer Manager 根據採購需求建立 Buyer Agent 並派遣到其對應的供應商伺服器端與其 Supplier Agent 進行交易，進行交易協商時，Buyer Agent 彼此交換訊息以獲得所有供應商的報價資訊，經由評估各供應商的報價資訊後，以評量效用最高的報價提案再與 Supplier Agent 進行下一回合的協商，直至協商結束。

圖 7 描述了 Buyer Agent 元件的虛擬碼。

```
//Class for the buyer agent
import com.ibm.aglet.*;
public class BuyerAgent extends Aglet {
    public void onCreation(Object init) {
        extract the information of other buyer agents,
        supplier, products, and decision model;
        initialize oneself using the above information;
    }
    public boolean onArrival() {
        obtain the proxy of supplier agent;
        modify the current location;
    }
    public void run() {
        if current location is at the assigned supplier side
        {
            waiting for the proxy of other buyer agents
            which was sent by the buyer manager;
            request contact;
            if the supplier permit contact {
                request quote until the negotiation is finish {
                    send the current status to buyer manager;
                    get the quotation;
                    send the quotation to other buyer agents
                    and buyer manager;
                    decide the best supplier proposal;
                    send the decided result to other buyer
                    agents and buyer manager;
                }
            }
            send the current status to buyer manager;
            dispose oneself;
        }
    }
    public boolean handleMessage(Message msg) {
        if buyer manager send the proxy of other buyers
        record the proxy and send the current status to
        other buyer agents;
        if other buyer agents send the current status
        record the current status;
        if other buyer agents send the quotation
        record the quotation;
        if buyer manager send the "Suspend" command
        suspend oneself;
        if buyer manager send the "Resume" command
        resume itself;
        if buyer manager send the "Stop" command
        dispose itself;
    }
}
```

圖 7 Buyer Agent 元件的虛擬碼

(三) Supplier Agent

每個供應商伺服器端將存在一個 Supplier Agent，可以跟買方所派遣的 Buyer Agent 進行交易，開始交易時，Supplier Agent 根據產品資訊產生不同的報價提案，由效用最高的提案開始與 Buyer Agent 協商，隨著 Buyer Agent 的回應，然後根據使用者的設定來改變其它的提案，直至協商結束。圖 8 描述了 Supplier Agent 元件的虛擬碼。

```
//Class for the supplier agent
import com.ibm.aglet.*;
public class SupplierAgent extends Aglet {
    public void onCreate(Object init) {
        query the products information from database;
        create the operative window;
    }
    public boolean handleMessage(Message msg) {
        if buyer agent request contact {
            check and identify the buyer;
            if the buyer is legal
                reply contact permit;
            else
                reply contact forbid;
        }
        if buyer agent request quote
            if the product is valid
                if buyer agent request quote first
                    reply the quotation according the type and
                    quantity of the product.
                else
                    reply the next quotation;
            else
                reply the product is invalid;
        }
    }
}
```

圖 8 Supplier Agent 元件的虛擬碼

在本研究的合作協商機制中，採用單一買方對應多個供應商的協商模式，當買方代理人派遣至供應商端後，協商流程如下：

- (1) Buyer Agent 針對所需之產品與數量向供應商代理人詢價。
- (2) Supplier Agent 根據使用者的設定決定起始提案、中間提案與保留提案，然後提出對自己效用最大的報價提案。
- (3) Buyer Agent 收到 Supplier Agent 的報價提案後，彼此交換資訊以便獲得所有供應商的報價提案，然後評量各供應商的提案，選擇效用最高的做為最佳選擇，並把選擇結果傳回給 Buyer Manager。
- (4) 若本身所對應的 Supplier Agent 的提案不是最佳的選擇，向 Supplier Agent 繼續詢價。
- (5) Supplier Agent 提出下一個提案。
- (6) 回到(3)。直到使用者結束協商或各供應商的提案不再變化。

另外，商務資訊中心 (Commerce Information Center) 存放買賣雙方的相關資訊，例如產品資訊、供應商(即 Supplier Agent)

的狀態與位置等，讓買方(即 Buyer Manager) 根據需求來查詢供應商資訊並與之交易。

為了要使代理人具有自主性，在本研究的架構中，每個 Buyer Agent 皆有獨立自主的決策能力，換句話說，當 Buyer Agent 收到 Supplier Agent 的報價資訊後，是與其它 Buyer Agent 交換資訊來進行決策，而不是傳回給 Buyer Manager 由 Buyer Manager 來決策。另外，非同步協商也讓各 Buyer Agent 立即決策並回應。這樣的利基點在於當其中若發生訊息傳遞遺失或是某一網路節點或代理人發生問題，其它的代理人還是可以繼續工作，使這樣的系統具有容錯能力。

而在各供應商的協商決策上，本研究採用多屬性產品的決策模式來評量各項報價提案。本研究採用了模糊推論(Fuzzy Inference)法與簡單加權(Simple Weight)法做為協商的決策模式[1,2,18]，本研究利用這些決策模式來模擬買方對多屬性的產品與多提案的評量方式。

六、系統實作與模擬

首先，我們必須先介紹 Aglet Server，由於行動代理人的活動必須在某一標準的環境中，所有的行動代理人才能在這環境中運作，並與其它的行動代理人交談，這樣的環境我們稱作為「Agent Server」。因為本研究是利用 IBM 所發展的 Aglet 來建構行動代理人，它也提供了這樣的環境來讓行動代理人運作，故這樣的環境稱為 Aglet Server，又稱 Tahiti。在本系統的運作方面，本研究建立了三個 Aglet Server 來模擬 B2B 的電子商場，一個代表買方的 Agent Server，IP Address 為 163.17.9.155；另外兩個則代表供應商的 Agent Server，IP Address 分別為 163.17.9.89 和 163.17.9.77。圖 9 顯示了本次模擬的架構圖。

在本次的模擬，本研究以旅遊業的商務環境來作為運用領域，更詳細的說，本研究採用旅行社與上遊旅館產業的訂房交易活動來建構系統。本次模擬共建構了 7 個 Supplier Agent，5 個位於台北區域的旅館(旅館代號：SH0001~SH0005)，2 個位於台中(旅館代號：SH0006 與 SH0007)。旅館代號：SH0001、SH0003、SH0005、SH0007 置於 IP Address 為 163.17.9.89 的 Aglet Server；SH0002、SH0004、SH0006 則置於 IP Address 為 163.17.9.77 的 Aglet Server。

圖 10 顯示了 Buyer Manager 的主畫面，使用者可以在 Buyer Manager 上設定採購需求，包括住宿區域、住宿日期、房間的種類與數量、價格與品質的需求、決策模式的選擇

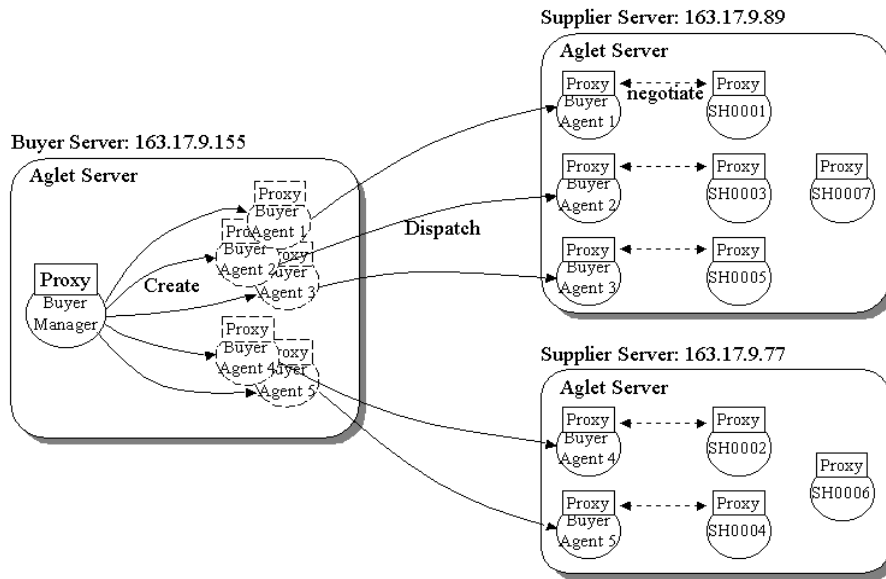


圖 9 本次模擬架構圖

等。設定完成後再按『查詢』鈕，系統根據需求向商務資訊中心(Commerce Information Center)尋找合適的供應商。在本次模擬中，採購需求設定後向商務資訊中心(Commerce Information Center)查詢後，得知可提供服務的供應商有 5 間。

當 Buyer Agent 被派遣出去後，會停留在供應商的 Aglet Server 等待 Buyer Manager 傳來其它 Buyer Agent 的 Proxy(利用其 Proxy 交談)，收到後就開始與其所對應的 Supplier Agent 進行交易，其交易活動包含識別身份、登入認證、詢價、協商與決策，並且與其它的 Buyer Agent 交換訊息，以便得知其它供應商的報價資訊來做決策，在本次模擬中，Buyer Manager 共派出了 5 個 Buyer Agent，其中圖 11 即為其中編號 1 的 Buyer Agent(Buyer Agent 1)的主畫面，上方顯示了其它 Buyer Agent 的報價資訊，由於本身不用顯示在上面，故用灰色來表示，左下方為提案的決策過程，右下方則是與供應商的對話視窗。在小括號()裡的數字代表決策的第幾次數；中括號[]裡的數字則代表詢價的第幾次數。

圖 12 也顯示了 Supplier Agent(SH0001)與 Buyer Agent 1 交易時的畫面。畫面左邊是與買方的對話視窗，在其右方中，顯示了本身產品的起始價格、保留價格與每次可以折扣的價格。若 Buyer Agent 對其提案不滿意的話，Supplier Agent 下次就可降低價格，直至保留價格為止。

為了使代理人具有自主性，能迅速的進行決策，不受網路頻寬或是其它 Buyer Agent 在它方發生故障的影響，在第一次決策後，系

統開始進行非同步的協商，換句話說，Buyer Agent 向 Supplier Agent 詢價後，等待一個時間點，不管其它 Buyer Agent 是否有傳送最新的詢價資訊，即自行做決策。圖 13 顯示了當進行第二回合的協商時，每個 Buyer Agent 所收到其它 Buyer Agent 的詢價資訊不會一致。

另外，在本系統中，也賦予 Buyer Manager 監控 Buyer Agent 的功能，使用者可透過此界面來得知所有 Buyer Agent 的狀態，包括協商中每回合的詢價資訊、決策結果及協商結束後的統計資訊。另外，使用者可以在交易進行時暫停、中止所有的 Buyer Agent。圖 14 即為 Buyer Agent 的監控畫面。

七、結論

本研究提出了一個以行動代理人為主的商務環境中，一個多重代理人的合作協商機制，討論在一個買方對應多個供應商的情況下，買方同時指派多個代理人至供應方進行交易。買方為了能取得最大利益，利用反覆競標的模式來找到最佳的提案。本研究有下列主要的成果：

- (1) 建構一個以 Mobile Agent 為主的商務環境，主要有買方、資訊商務中心與供應商三者所組成，由於 Mobile Agent 的優點，買方可根據需求來派遣代理人至供應方與供應商的代理人進行交易，提供一個更有彈性與效率之電子商場的網路架構。
- (2) 發展一個買方對應多個供應商的情況下，買方同時指派多個代理人至供應方進

行交易的合作與非同步協商機制。合作的協商機制提供買方同時可與多個供應商進行交易的架構與流程；非同步的協商機制可使買方代理人不受其它買方代理人或網路節點故障的影響，使個別買方代理人具有自主性與容錯力。

- (3) 使用旅行社與上遊旅館業者間訂房的商務活動作為本研究的運用領域，期望能建立一個適用於旅遊產業的電子商場，提供一個自動化的交易環境，改善現有旅遊產業的效率與成本，並推廣至各個產業。

未來的發展與建議

由於本研究只建立了初步的系統模型，故還有許多的問題還需要改進：

- (1) 在代理人的協商機制方面，本研究並沒有考慮買賣雙方動態地調整協商策略，例如，賣方除了降價外，也可以維持原來價格甚至漲價，調整價格的幅度也能動態地改變。這在目前是一項重要且難理想化的問題。
- (2) 在交易進行時，不論是在協商或決策時，買賣雙方必需透過訊息傳遞來交換資訊，尤其在特定的工作模式下，更要遵守其工作流程來交談，原則上，在代理人的溝通交談方面，應是透過類似人類用語言交談的方式來交換資訊，而非透過傳統的函式呼叫來相互溝通。在本研究中，則採用自行定義的訊息格式，所以功能簡單。在目前代理人溝通語言 (Agent Communication Language, ACL) 的研究中，較著名的有 FIPA 的 ACL 與 ARPA 的 KQML 兩種，若能加入標準的溝通語言，將使本系統與其它的外部系統容易整合。
- (3) 本系統的重點放在一個買方對應多個賣方的交易機制，但在現實的世界中，一個完整的電子商場是屬於多個買方對應多個賣方，買方可以藉由尋找相同的目的來結合成社群，以達到最佳的購買力、協商力；而賣方可以結合策略聯盟，以達到最佳的協商力與競爭力。

八、參考文獻

註：下面 URL 後面日期代表最後一次確認日期。

- [1] 郭建男，”以行動代理人為基礎商務環境下之群體代理人合作協商機制 - 設計與實作”，朝陽科技大學資訊管理系，碩士論文(2001)。

- [2] 蘇木春、張孝德，機器學習：類神經網路、模糊系統以及基因演算法則，全華科技(1999)。
- [3] Aglets, URL: <http://www.trl.ibm.co.jp/aglets/> (2001/7/12).
- [4] J.P. Bigus and J. Bigus, *Constructing Intelligent Agents with Java*, Wiley Computer Publishing (1998).
- [5] Caglayan and C. Harrison, *Agent Sourcebook*, Wiley Computer Publishing (1997).
- [6] P. Dasgupta, N. Narasimhan, Louise E. Moser and P.M. Melliar-Smith, “MAGNET: Mobile Agents for Networked Electronic Trading”, *IEEE Transactions on knowledge and data engineering*, Vol. 11, No. 4 (1999).
- [7] L. Esmahi, P. Dini and J.C. Bernard, “Toward an Open Virtual Market Place for Mobile Agents”, *IEEE 8th International Workshops on WET ICE '99*, pp. 279-286 (1999).
- [8] J. Ferber, *Multi-Agent Systems-An Introduction to distributed artificial intelligence*, Addison Wesley (1999).
- [9] R. Guttman and P. Maes, “Cooperative vs. Competitive Multi-Agent Negotiations in Retail Electronic Commerce”, *Proceedings of the Second International Workshop on Cooperative Information Agents (CIA'98)* (1998).
- [10] A.K. Jain, M. Aparicio IV and M.P. Singh, “Agents for Process Coherence in Virtual Enterprises”, *Communication of the ACM*, Vol. 42, No. 3 (1999).
- [11] J.J. Jung and G.S. Jo, “Brokerage between buyer and seller agents using Constraint Satisfaction Problem models”, *Decision Support Systems*, Vol. 28, Issue: 4, pp. 293-304 (2000).
- [12] D.B. Lange and M. Oshima, *Programming and Deploying Java Mobile Agents with Aglets*, Addison Wesley (1998).
- [13] B. Limthanmaphon, Y. Zhang, and Z. Zhang, “An agent-based negotiation model supporting transactions in electronic commerce”, *IEEE 11th International Workshop on Database and Expert Systems Applications*, pp. 440-444 (2000).
- [14] R. Lin and S.T. Chou, “Forming Dynamic Supply Chains in the Electronic Marketplace: A Partnership Negotiation Mechanism”, *National Computer Symposium*, (1999).

- [15] M.J. Mendes and F.M.A. Silva, "Mobile Agents in Autonomous Decentralized Systems", *The Fourth International Symposium on Integration of Heterogeneous Systems*, pp. 324-327 (1999).
- [16] T. Sandholm and Q. Huai, "Nomad: Mobile Agent System for an Internet-Based Auction House", *IEEE Internet Computing*, Vol. 42, pp. 80-86 (2000).
- [17] D. Wong, N. Paciorek, and D. Moore, "Java-based Mobile Agents", *Communication of the ACM*, Vol. 42, No. 3 (1999).
- [18] J. Yen and R. Langari, *Fuzzy Logic: Intelligence, control, and Information*, Prentice Hall (1999).

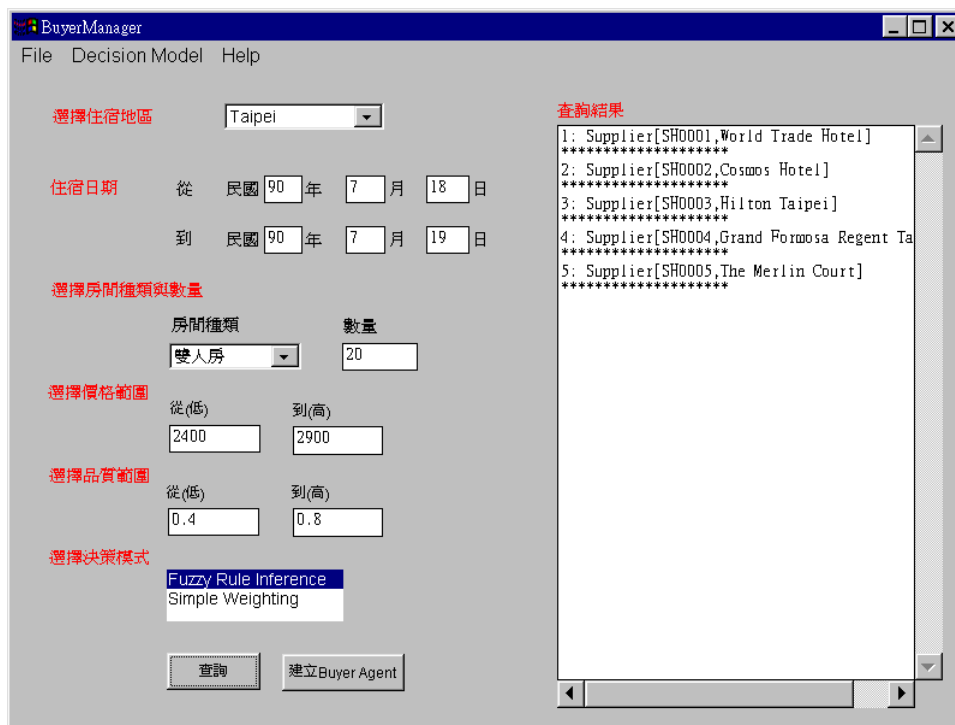


圖 10 Buyer Manager 的主畫面

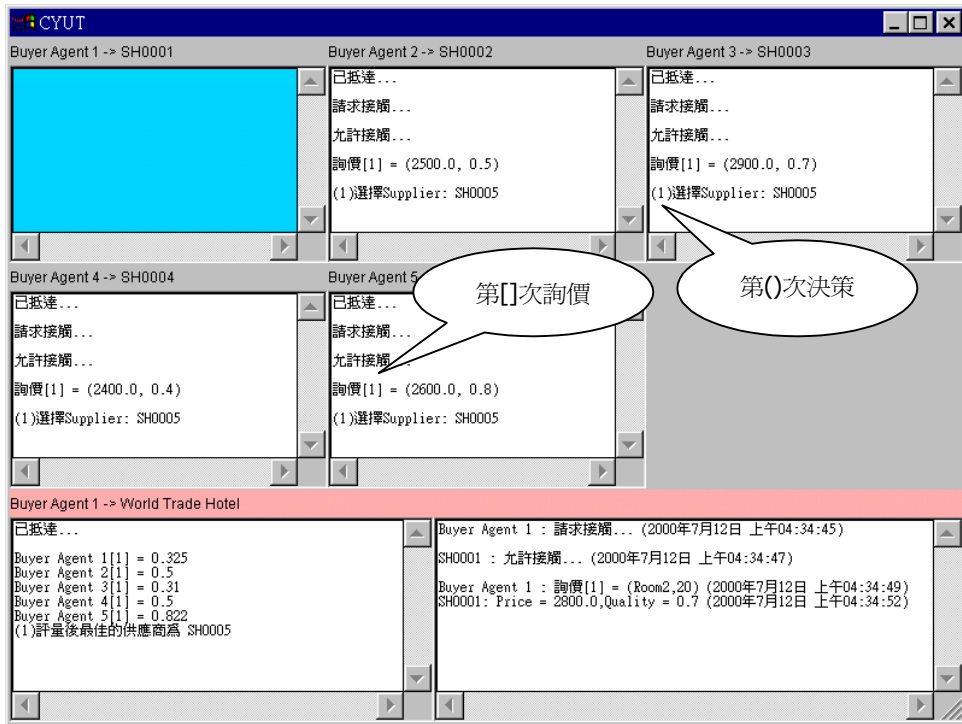


圖 11 Buyer Agent 1 的執行畫面



圖 12 Supplier Agent(SH0001)的主畫面

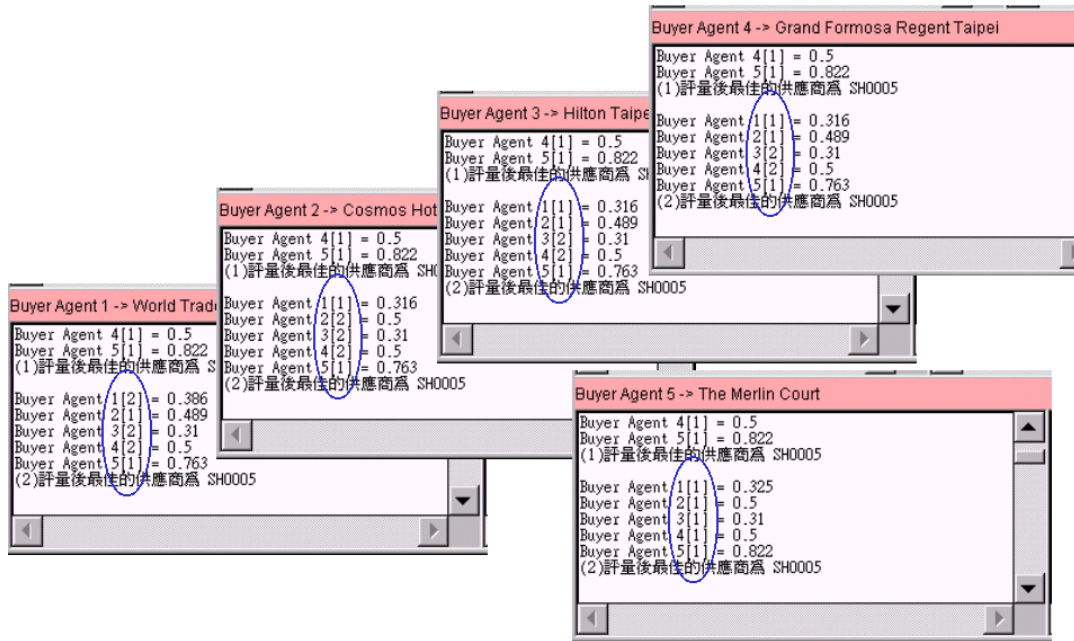


圖 13 非同步合作的協商模式

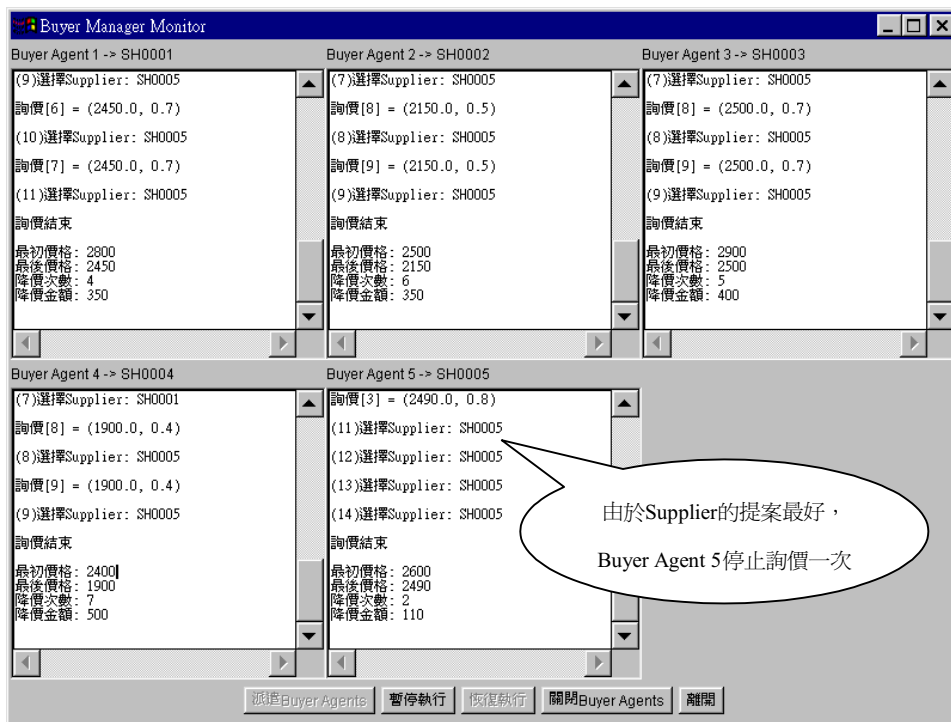


圖 14 Buyer Agent 的監控畫面