國立暨南國際大學資訊管理研究所

碩士論文

指導教授：俞旭昇

運用行動代理人建構一自動化議價之電子市集

Automated Negotiation in Electronic Marketplace by

Mobile Agent-Based Approach

研究生：鄭維元

中華民國九十年六月

論文摘要

隨著 Internet 的發展，許多企業也試圖將其傳統交易模式，如封閉式的平台，轉變成開放式 B2B、B2C 等模式，而加入電子市集的環境之中（Electronic Marketplace）已可說是必經之途了。

E-market 透過本身的群聚效應，集結了許多對該同類型商品的買賣方，如此大大的提高市場效率，進而加速促成供應鏈的形成。但是買賣雙方如何從眾多供需者中即時地找出符合自己的廠商，並減少比價議價所耗費的人力成本、代理成本與電子市集的建構時程與開發成本等等，依然是個癥結點。

因此我們提出以電子市集中立的第三仲介者為運作基礎，運用智慧型代理人與模糊之技術來建構本系統的議價協商機制，應用層級程序分析法（Analytical Hierarchy Process, AHP）做為評估各個商品間替代方案的屬性權重，且在 Mobile Agent 為基礎下，可減少 Agent 在網路上相互溝通的傳訊量，充分發揮系統彈性與延展性，重要是不局限於商品買賣種類，以提升 E-market 開發效率，並藉此達到買賣雙方交易前置過程（商品篩選、尋價、比價、議價協商）的自動化。

關鍵字：電子市集（E-Marketplace），代理人程式，協商議價

Abstract

With the rapid development of the Internet, many corporations are trying to do their business transactions through B2B and E-marketplace. Although these new technologies can reduce agency cost, there are some problems to be resolved. Some of them are how to explore uncertain information and reduce the cost of system development. This paper proposed an architecture adopting the mobile agent and fuzzy membership function technologies to building an E-marketplace, which is automatic negotiation machinery, reducing human cost and more flexible.

Keywords: E-market, Mobile Agent, Negotiation

目　錄

第壹章、緒論 1-1

1.1 研究背景與動機 1-1

1.2 研究目的 1-2

1.3 論文結構 1-2

第貳章、文獻探討 2-1

2.1、電子市集（Electronic Marketplace） 2-1

2.1.1、三種交易市集的模式 2-1

2.2 電子市集之流程 2-3

2.3 電子市集開發之網路架構 2-4

2.3.1 以主從式(Client-Server)與多層次(Multi-tier)架構 2-4

2.3.2 「Code-on-Demand」 2-6

2.3.3 以行動代理人(Mobile Agent)架構之電子市集 2-7

2.4 利用 Java 程式語言來開發本系統之優點 2-9

2.5 動態定價(Dynamic Pricing) 2-11

2.6 代理人程式(Agent) 2-12

2.7 層級程序分析法(Analytic Hierarchy Process , AHP) 2-13

第參章、系統設計 3-1

3.1 系統網路架構 3-1

3.1.1 系統功能之設計 3-2

3.1.2 系統資料庫配置與系統運作之管理流程設計 3-4

3.2 Fuzzy Membership Function 在本系統的運用 3-6

3.3 在本系統的協商議價屬性之階層分析運用 3-12

3.4 議價協商機制與決策屬性權重的產生 3-14

第肆章、系統實作與實驗結果 4-1

4.1 系統運作核心技術介紹 4-1

4.1.1 系統對資料庫的連結方式 4-3

4.1.2 代理人系統運作方式 4-4
4.2 系統介面介紹 4-5
4.2.1 系統登入 4-5
4.2.2 產品篩選(Input Constraint) 4-6
4.2.3 選擇要進一步協商之商品、授權代理人添購之附屬配備與其他交易條件(Selecting Suitable Goods) 4-7
4.2.4 代理人決策屬性設定 4-8
4.2.5 決策屬性－性能之下層子評準屬性之偏好評估 4-10
4.2.6 議價協商後之決策評估結果 4-11
4.2.7 賣方伺服器(Seller Server)啟動畫面 4-12
4.2.8 仲介服務中心(Intermediary Server) 4-12
4.3 實驗結果 4-13
4.3.1 偏好低價位、高性能、配備佳之採購決策評估 4-13
4.3.2 偏好中間價位、平均性能、配備中等之採購決策評估 4-14
第伍章、結論與未來研究方向 5-1
5.1 結論 5-1
5.2 未來研究方向 5-2
參考文獻 R-1
附錄一 A-1

圖　目　錄

【圖 2.3.1-1】Client-Server 架構圖.....2-5
【圖 2.3.1-2】三層式架構.....2-5
【圖 2.3.2-1】code-on-demand paradigm.....2-6
【圖 2.3.3-1】代理人運作模式.....2-7
【圖 2.5-1】動態定價的４種模式.....2-11
【圖 3.1-1】本電子市集之網路架構圖.....3-2
【圖 3.1.2-1】系統運作之維護管理.....3-5
【圖 3.1.2-2】ER-Model.....3-5
【圖 3.2-1】S-function.....3-6
【圖 3.2-2】S’-function.....3-7
【圖 3.2-3】V’s-function.....3-8
【圖 3.2-4】LH-function.....3-9
【圖 3.2-5】RH-function.....3-9
【圖 3.2-6】V-function.....3-9
【圖 3.2-7】T-function.....3-9
【圖 3.2-8】T’-function.....3-10
【圖 3.2-9】買方計算最大馬力之模糊數.....3-11
【圖 3.3-1】第一層與第二層之階層關係圖.....3-13
【圖 3.3-2】第二層價格與第三層之階層關係圖.....3-13
【圖 3.3-3】第二層整體性能與其下層之階層關係圖.....3-13
【圖 3.4-1】賣買雙方議價協商過程.....3-15
【圖 3.4-2】使用者個人之偏好函數.....3-16
【圖 3.4-3】AHP 階層屬性與替代方案關係圖.....3-17
【圖 3.4-4】價格屬性與替代方案階層關係圖.....3-18
【圖 3.4-5】整體性能與替代方案階層關係圖.....3-19
【圖 3.4-6】附屬配備與替代方案階層關係圖.....3-19
【圖 4.1-1】系統運作流程.....4-1
【圖 4.1-2】各個類別之間的運作關係圖.....4-2
【圖 4.1.1-2】JDBC-ODBC Connection Architecture.....4-3
【圖 4.1.2-1】Agent Transfer.....4-4
【圖 4.2.1-1】系統登入畫面.....4-5
【圖 4.2.2-1】產品篩選畫面.....4-6
【圖 4.2.3-1】Selecting Suitable Goods User interface.....4-7
【圖 4.2.4-1】代理人決策屬性設定.....4-8
【圖 4.2.4-2】代理人決策屬性設定 2.....4-9
【圖 4.2.5-1】Setting Criteria Evaluated Function Type.....4-10

【圖 4.2.6-1】Result of Negotiated 4-11
【圖 4.2.7-1】SellerServer 停止服務狀態..... 4-12
【圖 4.2.7-2】SellerServer 啟動服務狀態..... 4-12
【圖 4.2.8-1】啟動 Intermediary Server..... 4-13

表　目　錄

【表 3.2-1】產品性能評估表 3-10
【表 3.2-2】產品業界分類準則表 3-11
【表 3.4-1】Seller Negotiate Rule Base 3-15
【表 3.4-2】Values of the Random Index 3-17
【表 3.4-3】L1-L2 判斷矩陣 3-18
【表 3.4-4】L2 價格-L3 判斷矩陣 3-18
【表 3.4-5】L2 整體性能-L3 判斷矩陣 3-19
【表 3.4-6】L2 附屬配備-L3 判斷矩陣 3-19
【表 3.4-7】L2-L3 判斷矩陣 3-20
【表 4.3.1-1】車款效用評估表 1 4-13
【表 4.3.1-2】車款效用評估表 2 4-14
【表 4.3.1-2】車款效用評估表 3 4-14

第壹章、緒論

1.1 研究背景與動機

隨著 Internet 的發展，全球化趨勢、創新掛帥的世紀中，企業逐漸走向外包達到專業分工，保留自己核心能力，以期提高效率與降低成本，到最後可能只有站在頂端的企業才有超額營餘了。在如此競爭的環境中，電子市集(Electronic Marketplace)的已成趨勢；如美國的 GwoCities 以及臺灣的 CityFamily，也逐漸發展到企業間的網路社群，CommerceNet 就有２０個國家、五百多個企業成員加入，也依產業別在打造電子化交易市集。

在汽車產業供應鏈的主導者－裕隆，經我們在訪問其採購部門中得知，該企業在供應鏈採購決策上有著一個特點，就是在協力廠商之中有百分之七十左右的供應商是採行中長期合作關係，而大約有百分三十的貨源是處於尋找新夥伴的狀態，據了解這主要是為了降低風險發生，避免供應商不確定因素影響而中斷物料的供給，由此來看還是有許多的廠商一直在尋找忠實的合作夥伴，而不斷耗費在這費時費力的作業上；至於如何在如此繁多的買賣雙方的電子市集中，能快速有效率找出供應商或是需求者，而仲介者(Third Party Intermediary)又能扮演一個中立角色，並能提供在交易協商過程中所需的資訊，為採購決策著重之處。而仲介者、Service Provider 如何運用自己的專家知識在此環境中生存，就成了主要的成敗因素。

根據 Gartner Group 的資料顯示，目前 50%的交易集將會失敗，並且會逐漸會走向合併的命運。過去交易市集傾向於單一性，對於多面向的交易服務以及互動上則非常的缺乏，但交易市集的服務上過於侷限於單一的服務，無法滿足廣大的客戶需求，如果企業加強對於交易的服務提供以及即時互動上，則可以在交易市集上存活及獲利。自今年初開始，90%的交易

市集面臨嚴重的挑戰，除了在服務的提供上不具多面向性外，交易市集的互動性上也不高，無法滿足客戶的多樣化需求，容易被其他廠商所取代。

1.2 研究目的

為了解決交易市集互動性的不足，無法滿足客戶的多樣化需求，交易市集所提供的服務容易被模仿等，如何提高市場效率，進而加速促成供應鏈的形成，而在買賣雙方如何從眾多供需者中即時地找出符合自己的廠商，並減少比價議價所耗費的人力成本、代理成本與建構多樣化電子市集的開發成本等問題，仍是個癥結點。

因此我們提出以電子市集中立的第三仲介者為運作基礎，運用智慧型代理人與糢糊之技術來建構本系統的議價協商機制，應用層級程序分析法（Analytical Hierarchy Process, AHP）做為評估各個商品間替代方案的屬性權重，且在 Mobile Agent 為基礎下，可減少 Agent 在網路上相互溝通的傳輸訊量，充分發揮系統彈性與延展性，重要的一點是，不局限於商品買賣種類，以提升 E-market 開發效率，並藉此達到買賣雙方交易前置過程（商品篩選、尋價、比價、議價協商）的自動化，提高交易市集的互動性，配合專業領域知識核心，才能建構一不易被模仿的電子交易市集。

1.3 論文結構

本論文共分成五個章節，茲分述如下：

- 第一章為緒論，說明本研究背景、研究動機與目的與論文結構。
- 第二章為文獻探討，說明電子市集的定義、運作流程與相關網路系統架構的比較，還有為何我們採用 Mobile Agent 架構，且以協商議價方式來達到動態定價模式，加上運用 AHP 方式讓使用者在決策評估上更

有系統化的進行思考。

- 第三章為系統設計，主要說明本系統之架構，如何運用 Membership Function 來評估屬性質好壞、Agent 的協商機制與 AHP 在系統中的運用方式。
- 第四章為系統實作與實驗結果，主要對系統運作核心技術與系統介面的說明，之後再將車子相關資料與使用者個人偏好的輸入，並討論其結果是否真能符合消費者需求。
- 第五章為結論與未來研究方向，主要將本論文做最後的結論評估，並建議未來系統相關研究方向。

第貳章、文獻探討

2.1、電子市集（Electronic Marketplace）

Lindermann and Schmid[1] 指出，電子市場需包含顧客、供應商、服務提供者及操作者－顧客可從電子市場中快速地找到他所需要的產品訊息，有效減少因搜尋資訊而花費的成本；供應商可讓更多的顧客瞭解他們所提供的商品或服務；服務提供者可執行市場上相關的服務如市場情報供應、認證中心及商品配送的工作；操作者就是電子商店，是提供一個讓顧客進行消費的地方，亦可能是一個供應商或服務提供者，可販售自己的商品或扮演商品買賣仲介商的角色。他們提出一個電子市場的參考模型，主要涵蓋了組織和技術兩個層面，來完成交易過程中資訊蒐集階段、同意交易階段、及完成交易階段三個時期所需要的實施策略與技術援助。

在組織層面上，企業模式是整個電子市場的重心，必需要有明確的市場目標、策略並瞭解企業的商機，而企業社群的形成可促進企業與顧客之間良好的互動，企業程序及企業情境則關係到整個交易的完成，如認證與商品的配送都屬企業程序；在技術層面上，以交易的觀點來看，必需提供電子商品型錄、網路交易合約處理工具及電腦整合的商品後勤服務，而最底層的資訊與通訊網路基礎建設則是電子市集環境中最基本的條件。

2.1.1、三種交易市集的模式

目前交易市集的模式大致上可分為：買方主導、賣方主導或是由中立的第三者主導[24]。

賣方主導的市場，通常是由單一供應商面對多家買方的形態。目的是了在交易過程中，創造或維繫市場力量和價值。例如，思科的企業網站就

是能讓買方自行管理採購路由器的過程，確認零件準備時間、價格、下訂單以及確認送貨狀態，同時也能和技術專業人員直接商談。思科的網站一年可以產生 30 億美元的營業額，約佔公司總營業額 40%。

在買方控制的交易市集，通常是由一家或多家買方成立，目的是為了將市場上操控權和價值轉移到買方。例如，以買方服務中心仲介商，扮演代理者(Agent)或是匯聚者(Aggregator)的角色。FreeMarkets Online 公司，就是專門協助傳統產業找到提供半成品零件供應商的代理者，它為買方提供線上競標服務，這種為時三小時的競標，可以為買方節省平 10~25%的採購成本，幫助買方更有效率的找到貨源。奇異公司下的 TPN-Register，原本是為了整合公司各部門的採購而成立，後來衍生成可容納其他公司一起進行採購的匯聚者。該採購市集為企業節省了採購流程，也讓企業取得比原先低 10~15%的價格。

中立的交易市集，是由第三者仲介商所成立，用以撮合眾多買家與供應商。例如 FastPart，他們組成匿名的電子零件庫存交易市集。當接到賣方有庫存貨的通知，透過線上拍賣撮合交易，買賣方都能因此獲利，同時 FastPart 對品質也做了把關。FastParts 則賺取 8%交易仲介費，而最大的輸家則是傳統的仲介商。

那我們為何要選擇中立第三者來做為仲介商呢？主要因素如下：交易過程中的規模優勢。試想一個賣只 CPU 的市集，跟一個賣 computer 所有零組件的市集那一個容易達到一定的經濟規模，若提供更多的加值服務，如消費者資料分析、付款機制、後勤支援體，以便更容易吸引人潮，取得使用者規模的優勢，而在買方或賣方主導的交易市集，群聚的效果與範圍明顯地小了許多。不論買方或賣方所主導的市集，相對的非主導的一方即處於較不利的劣勢，而中立的第三者較能公正評估雙方資訊與產品屬性，並能為品質多做一次把關。企業通常會擔心敏感的資訊會透過網路外流到競爭者陣營，而中立網站亦可以提供讓參與者匿名被保護的功能。基於上

述理由，我們希望採用了第三中立仲介者為主的形態，來建構一個富有彈性與延展性電子交易市集，進而減少其開發成本，縮短建置時程。

2.2 電子市集之流程

隨著電子商務逐漸取代傳統商務模式，交易行為已大大不同，但仍屬於商業行為，因此在研究電子市集的同時，也必須考慮到消費者購買行為；根據 Guttman 等人[2][3]的整理與歸納，現實世界裡消費者行買行為模式(Consumer Buying Behavior Model , CBB MODEL)可以分為下列六個階段：

1. 需求確認階段（Need Identification）：消費者依據個興趣，並藉由資訊的刺激，逐漸認清自己的需求。
2. 產品仲介階段（Product Brokering）：消費者主動擷取資訊以便決定要購買什麼產品。其資訊可能包括一些評估準則、以及一些消費者認為可能符合需求之產品選項。
3. 商家仲介階段（Merchant Brokering）：決定要購買的商家及指定的商家選取標準，或對不同的商家加以評估，以決定要向什麼商家購買。消費者開始以自訂的評估準則來評估所有可能的選擇方案。
4. 協商階段（Negotiation）：本階段消費者與商家開始議定雙方均滿意的交易條件，協商的主題可能是價格、交易地點或交易日期等等。
5. 購買及交貨階段（Purchase and Delivery）：買賣雙方對交條件已經達成協議後，金錢與貨品的轉移是本階段的主要行為。
6. 售後服務與評估（Service and Evaluation）：消費者對商家售後服務、事後評對交易之滿意度、產品品質等行為，均在本階段發生。

從消費者的立場而言，理想的代理基礎電子市集環境，必須提供完整

的流程，讓買方代理人能夠比照消費者之行為，代理使用者完成整個交易行為；僅從消費者行為的觀點來決定 E-Marketplace 環境該有的流程，對賣方而言是不足夠的。一個完整的流程以外，也應相對提供賣方完整流程，使代理人也能代理廠商完成整個銷售過程, 一個合理的 E-Marketplace 流程應以下列的順序安排：

1. 描述代理人的身分（包括供給者與需求者之身分)、標的物（賣方的供給或買方的需求)、標的物價值的認定、與對方協商的策略等等。
2. 代理人被派遣至市場中，向市場詢問對手所在，所謂對手包括身分之對立、標的物相同等等。
3. 各自以自訂的協商策略、尋求本身利益最大化的前題下，為尋求雙方都能滿意的交易條件而協商。
4. 協商成功，買賣雙方履行契約。
5. 買賣雙方均為此次交易的結果下評論。

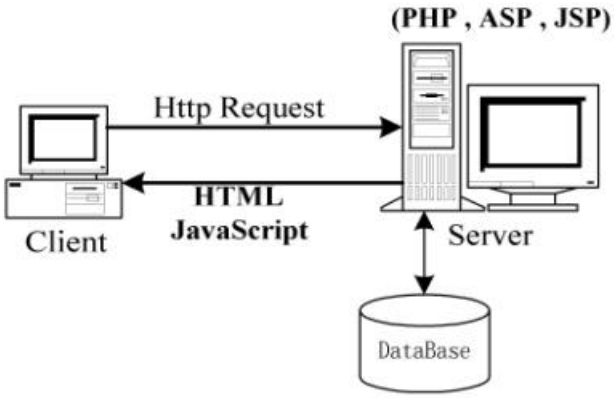
本研究主要重點是放在前三個步驟，至於在實體物流與交易後雙方的評筆並未涉及。

2.3 電子市集開發之網路架構

電子市集之網路架構，可分為下列三種：以行動代理人為主之分散式交易市集、以主從式或多層次架構之電子市集與 code on demand 方式。

2.3.1 以主從式(Client-Server)與多層次(Multi-tier)架構

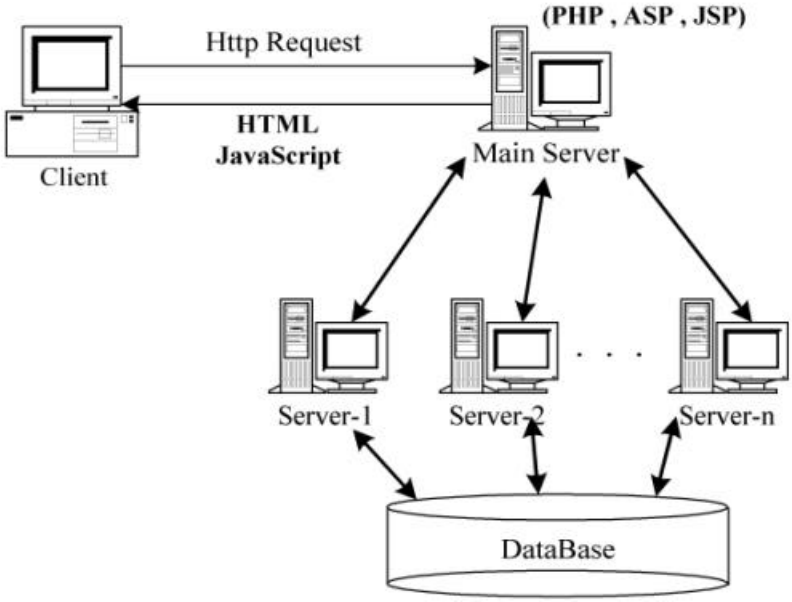
早期絕大多數的購物網站或是商場，最常見的就是 Client-Server 架構【圖 2.3.1-1】，將 Web Server 建構在單一伺服器上，所有的資料文件及網頁，都是由單一伺服器來管理。



【圖 2.3.1-1】Client-Server 架構圖

當 client 提出需求要擷取資料時，則透過內建 CGI 或是 Server Site Programs(ASP、PHP、JSP-Servlets)與資料庫連結；而此一架構可說是最經濟的方式，但是伺服器的負擔卻是非常的重，幾乎沒有利用到 client 端的資源，只有一些 Script 程式是在使用者端執行。

多層次架構【圖 2.3.1-2】的方式，解決了單一 Server 負荷太過於繁重的問題，它的方法主要是採用多台伺服器來分擔 client 端的需求。



【圖 2.3.1-2】三層式架構

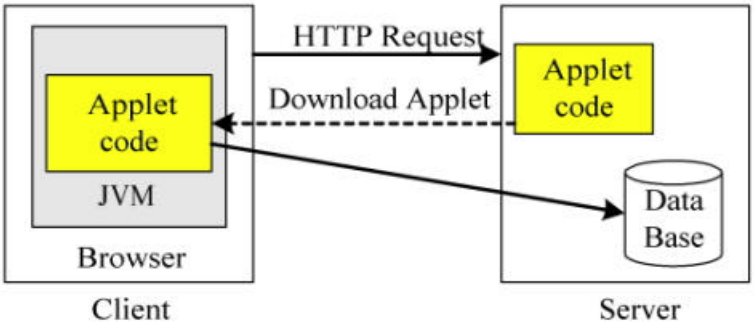
其中會有一個主要伺服器（如 AP-Server），所有 client 端的需求會先到此一伺服中，而主伺服器再將 Job 平均分攤到數台伺服中，這方式是增加硬體設備來提高效能的方式，但是跟 client-server 架構一樣的缺點，無

法充分利用 client 端的資源。

A.K, Jain 等人所發展的 SoCom 系統[20]就是屬於 Client-Server 型，它提供給買賣方一個交易環境，買賣方都必須先在此註冊，當買方代理人向 SoCom 系統提出需求後，SoCom 便告知有能力提供產品的供應商，若供應商要進行交易便與買方代理人程式進行協商動作，而所有的商業活動都在 SoCom 系統中執行。

2.3.2 「Code-on-Demand」

Java Applet 提供了一個很好運作模式，決解了無法充分利用 client 端資源的問題，它也是屬 client-server 形態，但是可以從 Server 端 download 程式碼到 client 端行執，當然 client 端的 Browser 中必須包含 JVM(Java Virtual Machine)才得以執行，如【圖 2.3.2-1】。



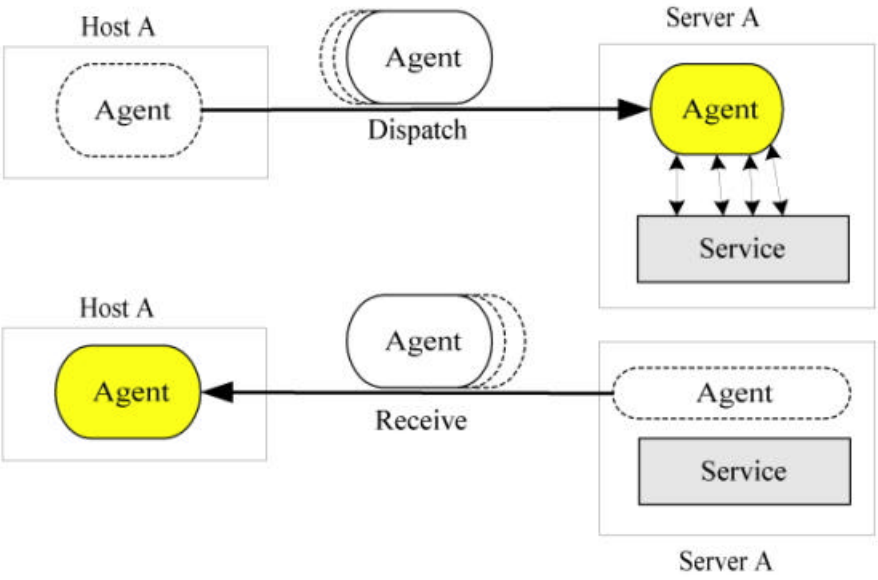
【圖 2.3.2-1】code-on-demand paradigm

在王善德所提出的供應鏈整合虛擬市集之架構中[21](1999)，使用下載的 Java Applet 來設定談判代理人行為模式，虛擬市集則提供了各方代理人程式行交易的環境。

但是 client 是否也可以將 code 派送至 server 端執呢？使系統架構更加有彈性呢？下面所述 Mobile Agent 提供了這個機制。

2.3.3 以行動代理人(Mobile Agent)架構之電子市集

Mobile Agent 顧名思意就是一個具備移動能力的代理程式，行動式代理人在執行時並不侷限在固定電腦中，它可以透過網路移動至另一台電腦中執行[6]。從【圖 2.3.3-1】可以了解 Mobile Agent 概念。



【圖 2.3.3-1】代理人運作模式

Agent 從 Host A Dispatch 到 Server A，如此 Agent 再與 Server A 中的 Application 做 communicate。Mobile Agent 有一個特點就是，Agent 中會儲存在 host A 時所設定的一些參數與資訊，隨著 code 帶到另一台電腦中處理，並且再把處理後的資訊與況態存在 mobile Agent 中，然後繼續 dispatch 到其他 node，直到完成 Job 回到自己的電腦中。我們從中可整理出行動代理人的幾個特性與優點：

- 減低網路上的負載

  當 Agent 與 server 間的溝通傳訊次數頻繁或兩端的距離是非常遙遠，而且傳輸品質不佳，如果把 Agent 派送至 Server 端，就不用透過網路來相互傳遞，可以大幅減低網路上的負載。

- 克服網路延遲問題

例如在一個即時環境中，Robots 在製造流程上運作時，它是必須要立刻做出反應來應對環境的變動，如果是透過工廠的內部網路來進行控制，勢必會因延遲問題而無法處理。而 Mobile Agent 供了一個良好的解決方案，Mobile Agent 可以直接派送至 Robots 中處理並進行控管行為，去除了網路上的延遲

- 分享計算資源

  Mobile Agent 可透過網路派送至另一台電腦中執行，分享計算資源。類似 Remote Procedure Call。

- 自動化非同步式運作

  通常在使用行動式電話時，其連線費用是十分昂貴的，假使 Agent 派送出去後必須很久才能回傳結果，如果一直持續不斷處於 on-line，非常不符合成本與時間上的效益。因而 Mobile Agent 亦可以達到自動化非同步式的運作方式，以節省成本。

- 動態的服務環境

  Mobile Agent 的架構提供了彈性的網路環境，服務端只需提供執行 Mobile Agent 的環境，需求端需要什麼樣的服務，即可以將 Mobile code 傳送到 server 端執行，而且需求端所需的服務亦可動態的改變，如此就可大幅增加應用程式的使用上彈性。

「行動代理人」的相關研究如下：

L.Esmahi 等人提出了一個以「行動代理人」為主的電子商場[22]，買賣雙方的代理人是派遣到這集中式電子商場進行交易行為。在商場中有一管理者代理人來負責控制與協調，在控制與協調機制中是依循一些 Rules Base、Case base 及 Common Services 來運作。

在 P.Dasgupta(1999)的研究中[23]，將買方代理人派送至賣方伺服器中進行協商行為，而不在一集中式環境中進行。當買方代理人與供應商之間協商若覺得滿意時，可先進行預購動作，否則就放棄再往下一供應商進行

交易，等到所有旅程完成後就會回到自己的伺服器中，將已預購之商品予使用者做最後的決策。

2.4 利用 Java 程式語言來開發本系統之優點

我們為何要採用 Java 來開發此一分散式交易之前置過程的決策支援系統呢?因為它有以下幾個優點，所以我們採用其功能來建構本系統：

- 簡單的（Simple）

  Java 使用簡單的原因主要是很類似 C Language，而且 Java 去除了 C 會造成的臭虫及記憶體漏洞的部份，取而代之的是更有效率且自動完成的工作，此外 Java 也提供了強大的類別庫，減少系統開發時間，提高了程式碼再利用性。

- 物件導向（Object-Oriented）

  Java 是物件導向語言，它使用叫做類別(class)的軟體物件，這些軟體物可以再使用(Reusable)而且也可以再延伸(Extendable)，程式設計師不用再注意實作的細節，而且也不用為了新增或改變 class 功能而重改原始碼，只需實作樣板中的 code，或繼承改寫父類別方式即可，如此開發程式時程可大幅提高。

- 強而有力（Robust）

  C Language 的指標可以說許多程式設計師在維護系統時的惡夢，但是在 Java 在分散代的環境下被使用，若還是能直接觸及記憶體更動，是非常危險且易造成當機，基於安全考量 Java 把指標拿掉。Java 自己會利用垃圾收集器來管理 Object 在記憶體上的回收，讓程式設計師不用再花心力去管理記憶體的使用方法。

- 中立的架構（Architecture Neutral）

  .java 的程式必須編譯成所謂的 byte code（.class）型態，而 byte

code 只適合 JVM 來讀取執行，也就是說不同的作業平台只要 JVM 就可以執行.class 程式，這也是真正讓 Java 成為跨平台程式語言的主要原因，有了所謂的中立的架構，發展軟體就不用針對不同的作業平台而重新開發。

- 可攜性（Portable）

  Java 的資訊形態是固定的，如 float 運算的資料形態永遠代表了 IEEI 位元浮點運算值，這種標準值對於不同的中央處理器是很重要的，藉由嚴格限定的資料型態的長度和數學運算，來保證在不同的平台應用程式都一樣，這是為何 Java 易攜性的重要環節。

- 安全性（Secure）

  Java 程式在你的機器上真正執行前，會分別被 Java 環境中每一個不同層級的安全系統檢查，因為 Java 環境可以被分成許多層，所以深入的安全檢查是重要的。但雖然有如此的安全措施，但也不能阻止程式設計師設計一個安全的程式，因此 Java 環境中其他層就負起對來自外部編譯過的 Byte code 做最後的檢查，以確保執行程式是沒有危險的。如果 Java 不是一那麼安全的程式語言，那它也就沒有資格遨遊於網路世界之中了。

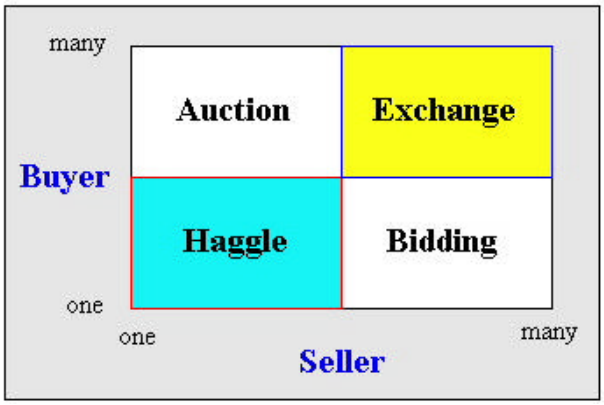
- 分散式（Distributed）

  支援網路應用是 Java 另一個重要功能，因應網路的快速發展，Java 為了普及的考量，發展了一個專門連結網路的 package，透過此一 package，Java 就可以與網路各層做溝通，使 Java 對網路存取更加有彈性。Java 更提供了 RMI、Serverlet 讓 client 端能多一此管道使用 server 端的資源，所以 Java 真正可以做到一個分散式的環境。

至於我們如何利用 Java 來實作 Mobile Agent 的環境，在第四章的系統實作裡會詳細說明之。

2.5 動態定價(Dynamic Pricing)

動態定價的概念代表改變單一價格的定價策略[25]，它展現了彈性調整貨品或服務的價值，以因應市場的短期變化。根據市場的特性，如買賣雙方的參與人數，動態定價可區分為 4 種模式。如圖【圖 2.5-1】



【圖 2.5-1】動態定價的 4 種模式

動態定價包括有競標(Auction)、議價(Haggle)、交換(Exchange)以及比價(Bidding)交易方式，這 4 種模式皆可適用於所有線上交易模式，包含 C2C(Customer to Customer)、B2C(Business to Customer)、B2B(Business to Business)等。

- 競標(Auction)：拍賣通常是由單一賣方提供拍賣品，而由多位買方出價進行拍賣品之競價。
- 議價(Haggle)：議價在買賣行為模式中是最普遍且常見之模式，由單一買方與單一賣方進行價格之協商或討價還價以達到雙方皆同意之價格。
- 比價(Bidding)：比價交易模式是一般傳統商業所使用之模式，由多位賣方針對單一買方所欲購買之貨品進行報價，最後由買方比價後決定欲下單之賣方。
- 交換(Exchange)：買賣雙方皆為多數，在線上進行價協商以滿足雙方之需求。

動態定價是以不同之模式協助線上商業活動，它消除了傳統交易上普遍存在資訊不足情形，使採購流程更為快速流暢。如此一來，賣方將能夠減少存貨成本、降低風險、減少其中不必要的中間商與增加收入等；而買方也可以較低的價格購得所需之貨品與減少支出。在動態價模式中，由買賣兩方依照某規則協議貨品售價，最後再形成該貨品之價格的線上拍賣，可以說是發揮動態定價力量的代表模式。

本系統希望能透過 E-Market 提供市場價格機制，達成尋價、比價、議價、或進一步協商等過程，達到動態定價模式。同時提供買賣雙方一公平及公開的環境來進行交易。

2.6 代理人程式(Agent)

智慧型代理人本是源於人工智慧的領域，藉由類神經網路或模糊(fuzzy) 等理論，讓系統能以人的思考方式來運作。Selker[16]認為代理人程式具備模擬人際關係的能力，並能夠協助人們完成工作項。Aparico[15]認為Agent是一個能代表使用者，在獨立或些許引導的情況下，執行特定工作的電腦程式。Smith[17]定義代理人程式為達成特定目的而持續地執行的程式，它們知道如何完成任務或工作項目。由於有許多Agent的理論尚未統一，因此仍沒有一精確形式化定義，不過在眾多Agent理論中，可以歸納出Agent應具有自主性、適應性、主動性、及社會性等能力[4]，且由於Agent特別適合應用於資訊豐富且作業程序複雜的電子市場，因此我們藉以提供採購流程自動化之資訊服務。

而行動代理人為其中一類。行動代理人除了替使用者處理反覆、繁雜或長時間監控的工作外，並能依據排定的行程，自主地在網路各站台中遷移、觀察、取得資訊，以期完成委託人的工作。因此，在本論文中使用行動式代理人技術於電子市集上，對於促進市場效率與自動化確實有莫大的

助益。

2.7 層級程序分析法(Analytic Hierarchy Process，AHP)

層級程序分析法由美國賓州大學Wharton 管理學院Saaty教授所發展出來一個多規範(Multi-Criteria)、多目標(Multi-Objective)的層級程序分析法，在近幾年來已經被應用於不計其數的高層次決策分析問題上。Saaty[10]在其研究中指出，AHP的應用領域包括：決定優先順序、資源分配、規劃、預測未來風險評估、選擇方案之產生、最佳方案之決策、確定需求、績效評估、系統設計、確保系統穩定性、最佳化及解決衝突等十二項。它可以解決同時需要考慮數量性(Quantitative)和非數量性(Qualitative)條件的複雜多規範問題。

AHP 的應用特性在於允許決策者依據其目標、知識與經驗，藉由一種與其直覺思考程序一致的方式，將複雜的問題由上而下分解成層級結構，透過群體內外在資訊的判斷並加以整合，同時減少個人主觀決策錯誤之風險性，以求取最底層可行方案的優劣程度，並據以進行優先次序的排列與最佳方案的選擇。

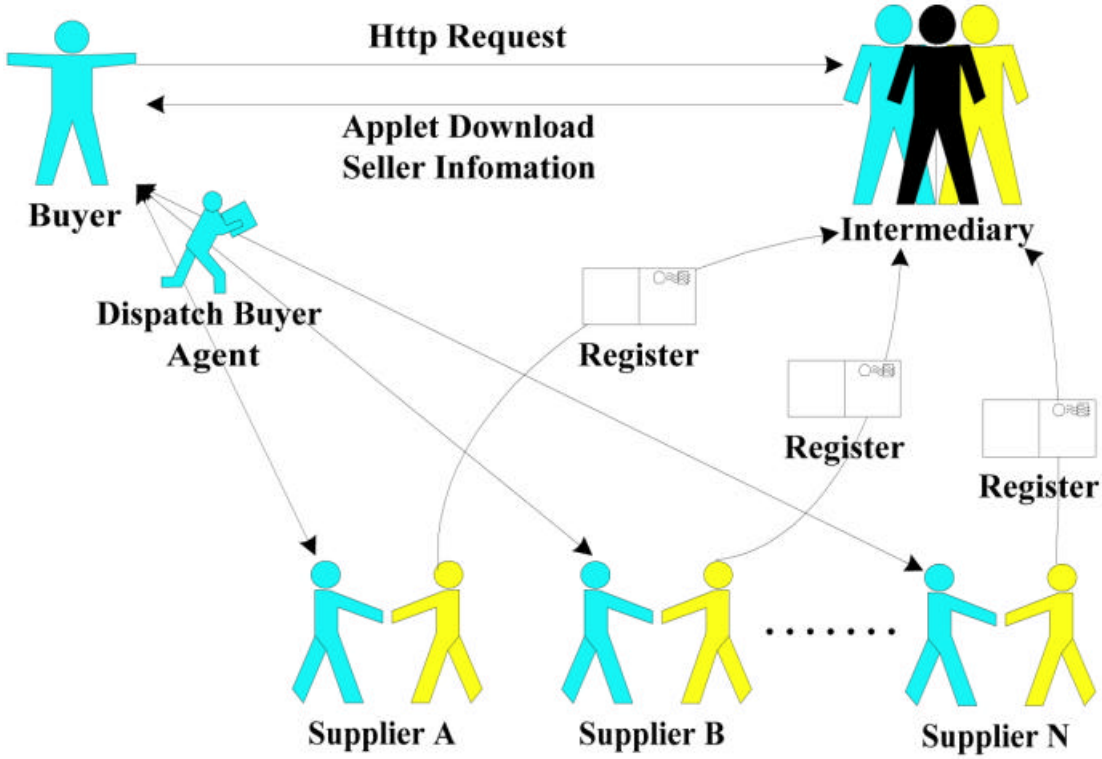
國內外有許多之論文、書籍介紹與應用此一方法[11][13][14]，但是亦有其限制與缺點[12]，由分支度過多時，會造成計算的困難與評估上的困擾，而且有可能會造成邏輯上不一致的評筆結果，因此在本系統中主要是讓使用者評估第二層與第一層間的決策權重，而下層與替代方案之詳細評估，則由專家來制定其規則庫，如此可避免煩雜的比較過程，也便方使用者，而且在評估過程運用CI/RI值檢測權重是否到一致，去除決策邏輯上的不一致問題。最後會再依照使用者個人偏好算出決策評估值，予買方下定最後的購買決策。

第參章、系統設計

本系統是運用行動代理人(Mobile Agent)為基礎，建構在中立第三者所主導的電子市集，透過公正第三者所評估賣買方之屬性質來做為商家搜尋依據與交易評準。首先介紹本研究之系統架構，我們使用 Java 跨平台物件導向語言來建構本系統，在各個產業中所用之作業平台不盡相同，因此 Java 為我們的第一考量，由於 Java 有著高度模組化的優點，如此要建構一個新的電子市集也會變得較為簡單化。而代理人之間的議價協商機制，主要是利用 Fuzzy Membership Function 來評估各個屬性質，使其正規化得以加總成一議價能力，買賣雙方依此來做為調整價格的幅度。層級程序分析法(AHP)，評估協商後的產品屬性，買方來裁決其最佳方案。

3.1 系統網路架構

本研究採取以『行動代理人』為主之分散式交易的 E-Marketplace。並同時派遣數個代理人至賣方做協商之行為。如【圖 3.1-1】為本系統之電子市集架構圖，主要包括了，使用者代理人(Buyer Applet)、買方代理人(Buyer Agent)、仲介服務中心(Intermediary Data Center)、賣方服務中心(Seller Service Center)以及賣方代理人(Seller Agent)等所組成。



【圖 3.1-1】本電子市集之網路架構圖

3.1.1 系統功能之設計

構成本系統之主要的代理人如下：

(1) **使用者代理人**(Buyer Applet)

使用者代理人是買方從仲介服務中心所下載的 Applet**，買方可以從中設定所需產品的規格屬性，再向仲介服務中心取得符合之產品名目與商家資訊**(IP address、Post Number…)**，買方可從中選定要進行議價協商的產品，並透過** AHP **設定** Criteria **權重分配與個別的偏好** Function**，之後依據要協商的資訊產生買方代理人，而買方代理人的派送、接收與最後產品評估決策值都是由使用者代理人控管。**

(2) **買方代理人**(Buyer Agent)

買方代理人帶著產品資訊、使用者所設定的參數到賣方的

Server 中，並向仲介服務中心取得賣方產品評估值的規則（Rule Base），依據這些資訊透過 Fuzzy Membership Function 來計算各個屬性的歸屬程度值，如此是為了使每個屬性值正規化，得以加總成一議價能力值，做為與賣方價格協商的依據。當議得雙方可以接受的狀態後，買方代理就會被送回給使用者代理人，以便與其代買方代理人做最後相對的決策評估動作。

(3) 仲介服務中心(Intermediary Data Center)

仲介服務中心主要記錄賣方 Server 註冊資訊(IP address、Port number、company Information .etc.)與該商家所販售的產品資訊(Ontology)，而整體同類型產品評估值亦是由中立的仲介者來維護與管理評估資訊，主要是透過專家經驗所評定產品良劣，這些都是公正的評估值，賣買雙都必須到此來擷取要議價之標的物的評估資訊。

(4) 賣方服務中心(Seller Service Center)

在賣方資訊中心裡主要提供賣方代理人一些授權資訊，而此中心可能會與公司的客戶關係管理(Custom Relationship Management ,CRM)做一整合，以了解那些客戶是要給與那種價位，客戶偏好是什麼等等，如此賣方代理人才可以提出有效方案予買方代理人。另一個功能是接收買方代理人要求議價協商，賣方服務中心就會開起一個 Thread 為賣方代理人與買方代理人做進一步的協商。

(5) 賣方代理人(Seller Agent)

賣方代理人由賣方服務中心產生後就負責與買方代理人做溝通, 它必須取得賣方服務中心的授權規則，並跟仲介服務中心取得該產品的評估屬性值，跟買方一樣以仲介服務中心公定的 Fuzzy Membership Function 計算出議價能力。完成與買方 Agent 的協議

後就自動結束程序。

3.1.2 系統資料庫配置與系統運作之管理流程設計

我們站在系統管理角度來看，主要成員如下：

(1)仲介商

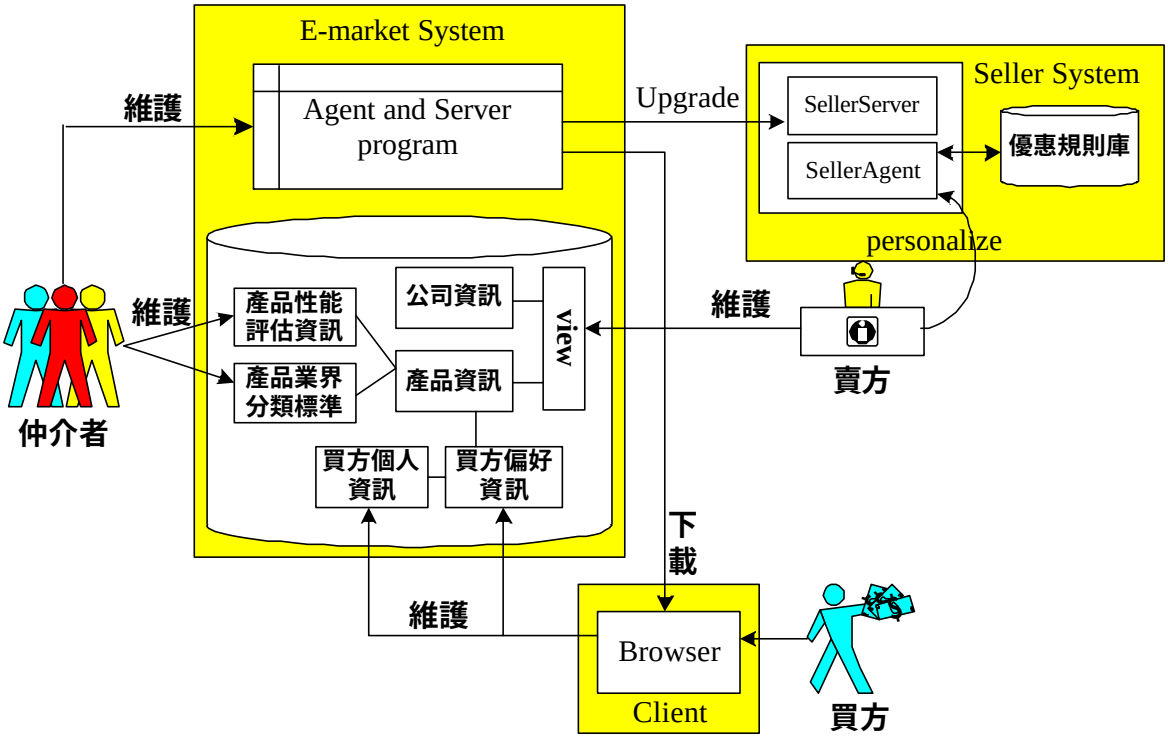
仲介者必須提供本身在產業中的專長來評估商品之中各個評準屬性、合理客觀協商議價的機制，並給予產品適當的分類，以利於使用者尋找的過程，亦可使議價時較有根據與意義。

(2)賣方

賣方主要是負責維護與管理自己產品型錄給仲介商，告知仲介者自己伺服狀態與相關資訊，加上授權予代理人的優惠方案等等資訊，必須詳實記錄在資料庫中。在流程上的轉變則必須跟仲介商取得更新的程式，而賣方亦可以繼承方式，改寫存取資料庫的部分程式碼，以便配合各個不同的 database，或進一步整合公司內部系統，如生管部門(MRP)、客服部門(CRM)的資訊系統等，以達到即時反應使用者需求。

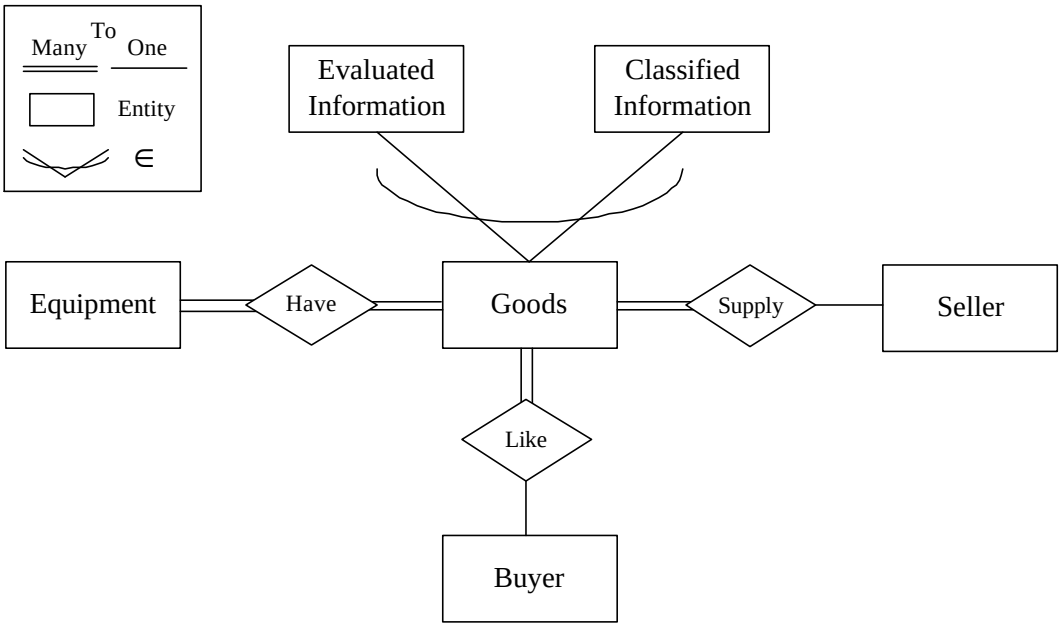
(3)買方

買方對產品屬性的偏好，算是提供於代理人做決策的重要資訊，而這一部分可以是個人偏好或是採購部門主管的群體決策偏好，該資訊當然是由買方自己維護。



【圖 3.1.2-1】系統運作之維護管理

本系統的 ER-Model 如下圖：



【圖 3.1.2-2】ER-Model

細部的 Table 內容，我們列在附錄一中。

3.2 Fuzzy Membership Function 在本系統的運用

Fuzzy Membership Function 最主是是將各屬性質做一正規化的動作，以便加總統合加權運算。透過第三仲介者對產品的評選標準區間與該產品屬性在此一區間相對位址的比值，依其公定之函數，輸入不同的 Fuzzy Membership function 算出該產品屬性之 Fuzzy 值，其參數如下：

- 非線性函數

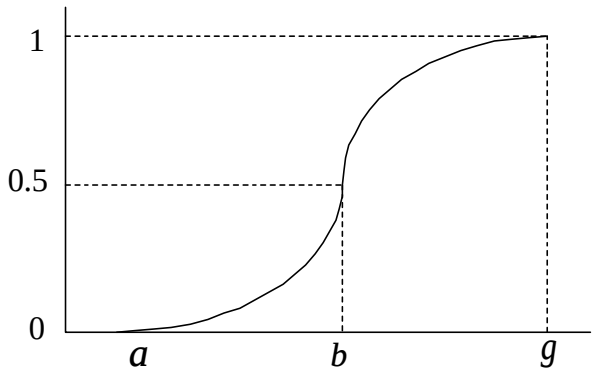
X_a：第 a 項屬性值　　a = 1,2,….,n

f_a：第 a 項屬性值 Fuzzy 值　　a = 1,2,….,n

若第 a 項屬性值越大，表示其議價能力越強，則將 X_a 代入 S-function【圖 3.2-1】運算。

$$S(x_a, a, b, g) = \begin{cases} 0 & for \quad x_a \le a \\ 2\left(\frac{x-a}{g-a}\right)^2 & for \quad a < x_a < b \\ 0.5 & for \quad x_a = b \\ 1-2\left(\frac{x-a}{g-a}\right)^2 & for \quad b < x_a < g \\ 1 & for \quad x_a \ge g \end{cases}$$

其中，$\alpha = min[X_a]$；$\beta = avg[X_a]$；$\gamma = max[X_a]$　得 $f_a = S(X_a)$，a=1,2,….,n；



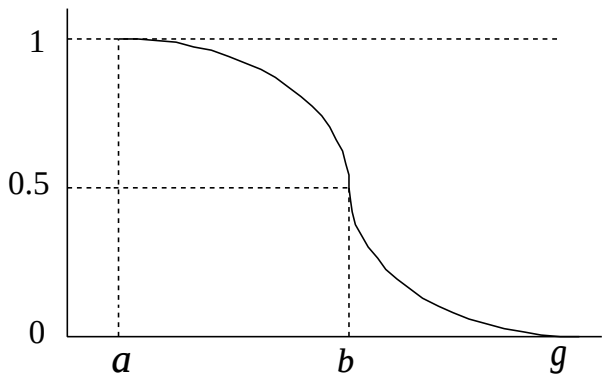
【圖 3.2-1】S-function

若第 a 項屬性值越小，表示其議價能力越強，則將 X_a 代入 S-function

(圖)運算。S’-function = 1-(S-function)。

$$S'(x_a, a, b, g) = \begin{cases} 1 & for \quad x_a \le a \\ 1-2\left(\frac{x-a}{g-a}\right)^2 & for \quad a < x_a < b \\ 0.5 & for \quad x_a = b \\ 2\left(\frac{x-a}{g-a}\right)^2 & for \quad b < x_a < g \\ 0 & for \quad x_a \ge g \end{cases}$$

其中，α $=min[X_a]$；β $=avg[X_a]$；γ $=max[X_a]$ 得 $f_a = S[X_a]$ a = 1, 2, …, n ;

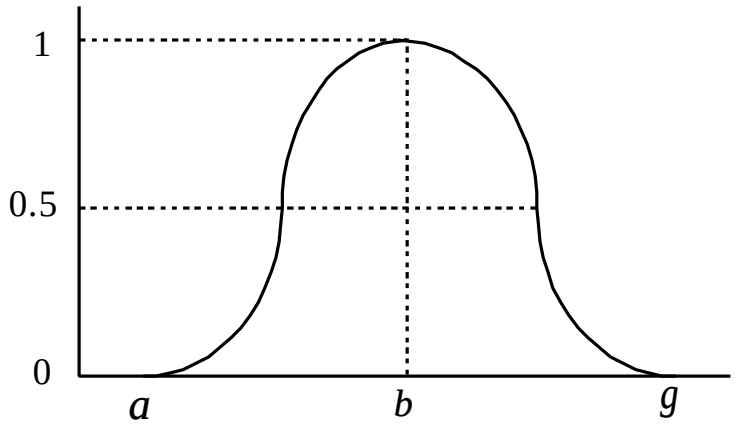


【圖 3.2-2】S’-function

上述所用之S-function 與 S’-function 主要是為了能快速區格出中間平均區域，例如某一屬性質在其同類型之產品中，大家都是處於伯仲之間，如果比平均值好或差一些，但所轉換出來的數值會呈現指數增加或減少的方式，以便區格之。

在不同的屬性與不同的個人偏好，我們的 Membership function 也包含了下列數種模式：

V’s-function 主要結合上述之 S-function 與 S’-function，而其最佳點是落在β，也就是在平均值所產出的 fuzzy 值會是最好的，反之離平均值越遠則越低。



【圖 3.2-3】V’s-function

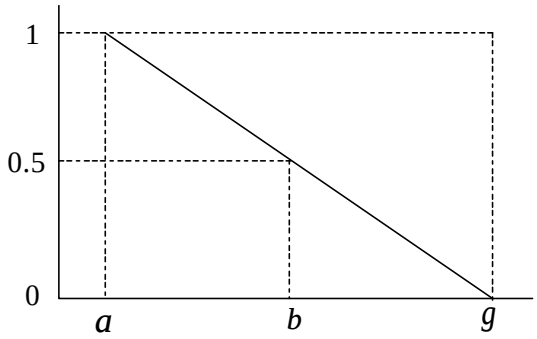
- **線性函數**

Linear function RH、LH **與** S-function、S’-function **同樣的在** Y **軸** X **軸有同方向與反方向變動的關係。如下：**

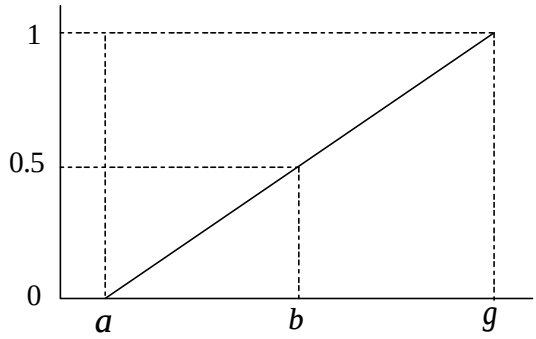
$$m_{LH}(x_a, a, b, g) = \begin{cases} 0 & for \quad x_a \le a \\ 0.5 \times \dfrac{x-a}{b-a} & for \quad a < x_a < b \\ 0.5 & for \quad x_a = b \\ 0.5 \times \left(\dfrac{x-b}{g-b} + 1 \right) & for \quad b < x_a < g \\ 1 & for \quad x_a \ge g \end{cases}$$

$$m_{RH}(x_a, a, b, g) = \begin{cases} 1 & for \quad x_a \le a \\ 0.5 \times \left(\dfrac{b-x}{b-a} + 1 \right) & for \quad a < x_a < b \\ 0.5 & for \quad x_a = b \\ 0.5 \times \dfrac{g-x}{g-b} & for \quad b < x_a < g \\ 0 & for \quad x_a \ge g \end{cases}$$

其中，α $= min[X_a]$; **β** $= avg[X_a]$; **γ** $= max[X_a]$ **得** $f_a = S(X_a)$**，** a=1,2,….,n ;

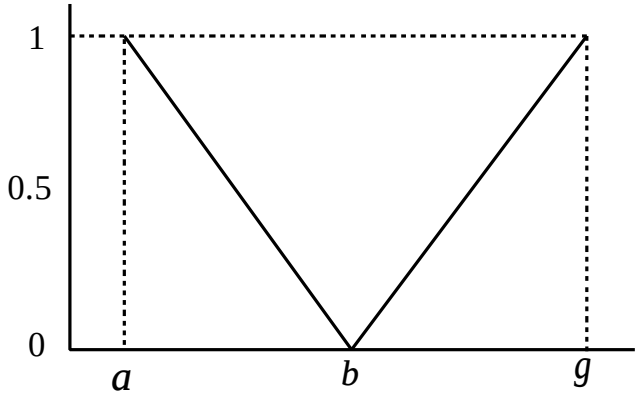


【圖 3.2-4】LH-function



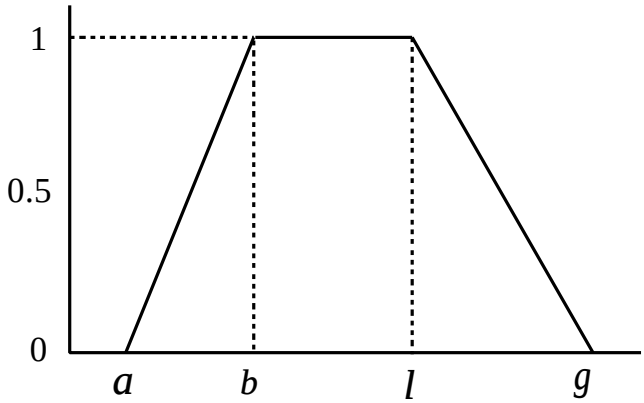
【圖 3.2-5】RH-function

V-function **在平均值中所對應出的** fuzzy **值是最小的，反之離平均值越遠則越高。**



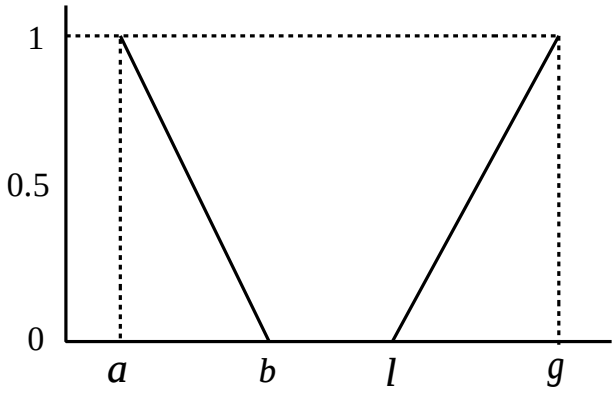
【圖 3.2-6】V-function

T-function **在定義的某一區間內所對應出的** fuzzy **值是最大的，反之離此一區間範圍後則呈現遞減的狀況。**



【圖 3.2-7】T-function

T’-function 在定義的某一區間內所對應出的 fuzzy 值是最小的，反之離此一區間範圍後則呈現遞增的狀況。



【圖 3.2-8】T’-function

利用上述之 Membership function 我們可以將【表 3.2-1】產品性能評估表，再透過專家（第三中立仲介者）所評估分類規則庫【表 3.2-2】，將屬性值轉換成一模糊數做一加總成一議價能力。

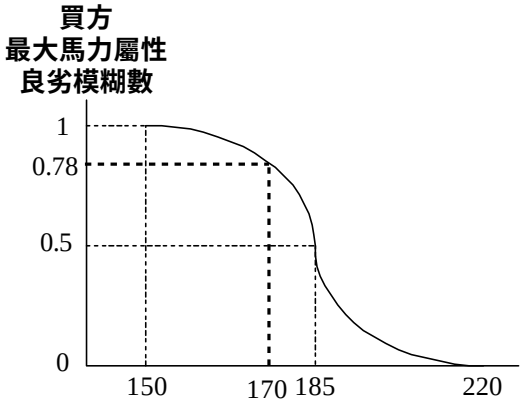
【表 3.2-1】產品性能評估表

產品 ID	評準 ID	Criteria ID	Value
E240	C1	*輪胎尺寸(寬/?/R?)	205/65/R15
E240	T1	2 前輪/4 輪-驅動	2
E240	D1	*長/寬/高 (mm)	4818/1799/1441
E240	D2	軸距(mm)	2833
E240	E1	C.C.	2597
E240	E2	最大馬力(hp)	170
E240	E3	最大扭力(kgm)	24.5
E240	P1	0-60mph(秒)	10.99
E240	P2	0-400m(秒)	18.39
E240	P3	100km/h-0(m) (秒)	39.3
E240	T2	*A 自排/N 手排	1

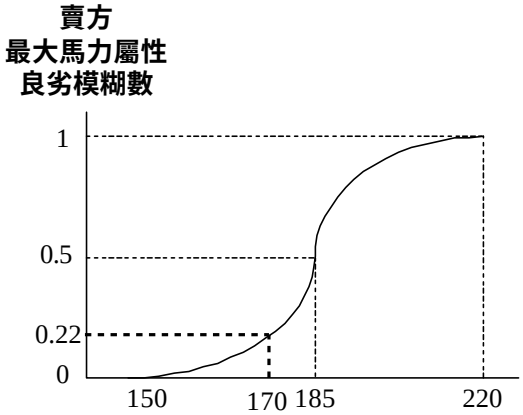
【表 3.2-2】產品業界分類準則表

產品類型	α	β	γ	Criteria Name	Buyer Function Type
A	2750	2850	2980	軸距(mm)	S’-function
A	2250	2650	3000	C.C.	S’-function
A	150	190	220	最大馬力(hp)	S-function
A	20	25	30	最大扭力(kgm)	S’-function
A	9.5	10.6	12.5	0-60mph(秒)	S-function
A	16.7	18	19.6	0-400m(秒)	S-function
A	32	37	42	100km/h-0(m)(秒)	S-function
A	175	200	233	*B_Price Range	S’-function

透過上述兩個表格，可以得知各個評準屬性分別必須使用何種函數，來轉換成一個可以加總的模糊數，以便做為更進一步的議價與協商依據。下面我們以汽車產業為例子，說明了最大馬力是如何利用歸屬函做轉換的動作：



【圖 3.2-9】買方計算最大馬力之模糊數



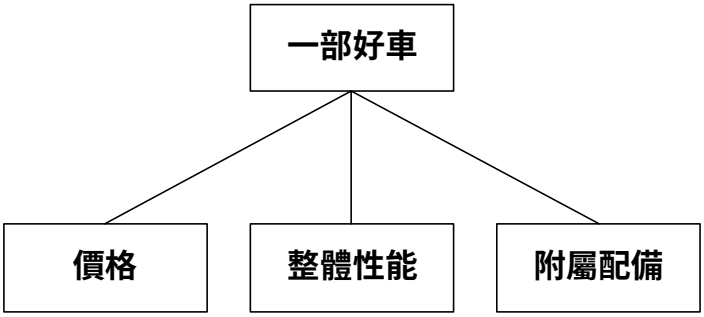
【圖 3.2-10】賣方計算最大馬力之模糊數

$$\sum_{i=1}^{n} m_{x_i}(x_i) \times w_{x_i}$$

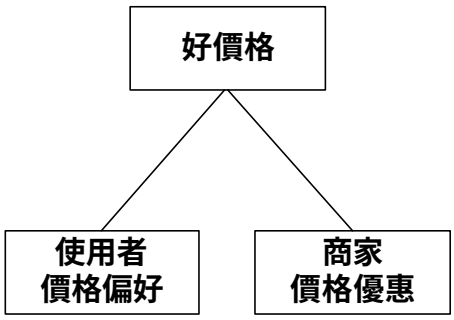
上述之公式，將各個屬性評估值(x_i)經過各自的歸屬函數(μ_{xi})，乘上權重(w_{xi})後就我們所要的<u>議價能力值</u>，再透過【圖 3.4-1】的函數，即可產生價格。再這一部分的權重涉及到在議價時價格好壞，如果雙方都將權重放在對己方有利的評準屬性值上，如此在協商議價上時，買方就會堅持在價格最低限，而賣方就堅持在上限，因而可能導致交易的效率降低或失敗，而且在雙方在交涉過程因權重不同而產生不同的價格是不合理的，所以在計算議價能力時，權重的與歸函屬的訂定，全部是由中立第三仲介者來裁定。

3.3 在本系統的協商議價屬性之階層分析運用

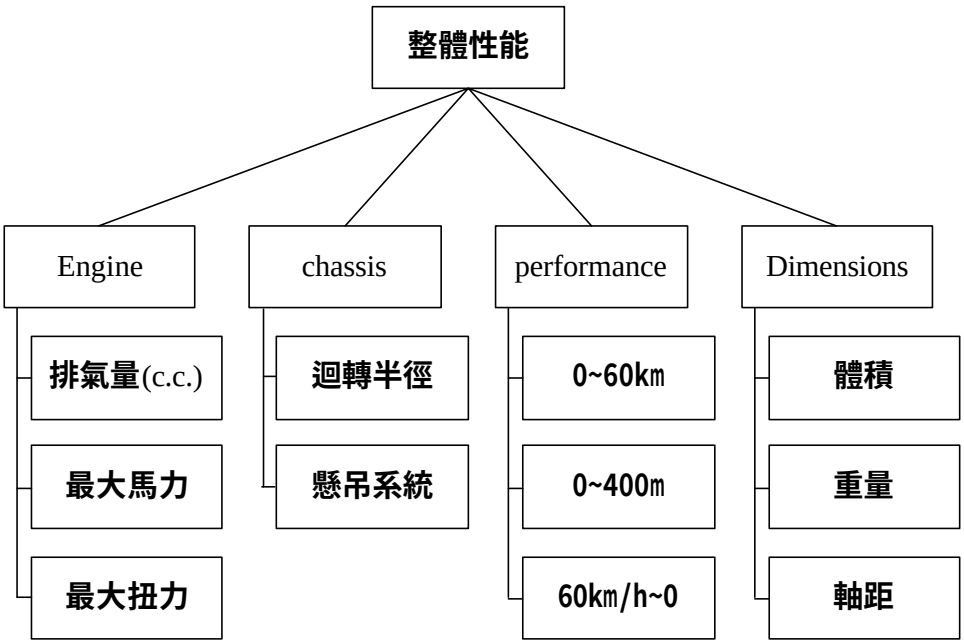
一般消費者在評估一件商品的良劣大致上可以分為三個主要大項目，第一要素不外乎是在價格上的考量，第二個 Criteria 可能就是產品本身的性能好壞，第三個可能是其他附屬配備或是其他交貨條等等細項，如【圖 3.3-1】是第一層與第二層的階層關係圖。我們以汽車為例子，考量其中有那些第二層以下之評準屬性。



【圖 3.3-1】第一層與第二層之階層關係圖



【圖 3.3-2】第二層價格與第三層之階層關係圖



【圖 3.3-3】第二層整體性能與其下層之階層關係圖

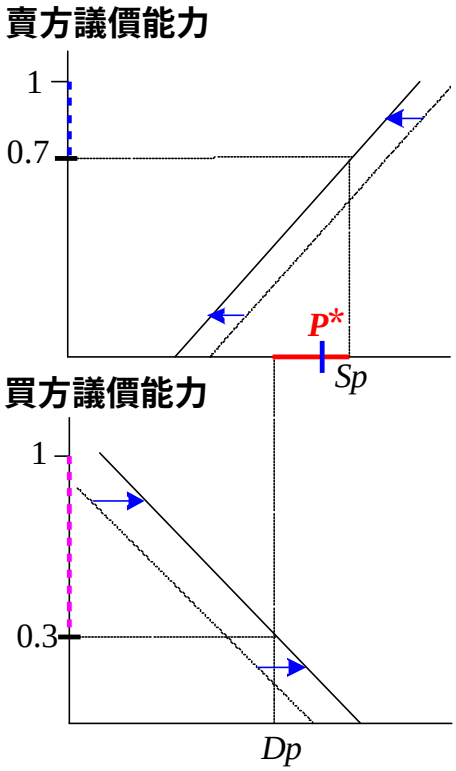
由上述之屬性評準階層關係可以看出，只有整體性能無法產生一個可評估的價格數據，因此我們將這一部分的屬性值，透過 membership function

轉換成一議價能力，之後再利用該議價能力產生買賣雙方的初始價格，而雙方在價格上或是附屬配備上還是有歧見時，亦是利用它做為價格退讓的依據。

在如此繁多的 criteria，若要使用者一一對其做兩兩比較的方式決定權重是非常麻煩困擾的，因此我們只給使用者輸入第一層與第二層的比較權重，而第二層以下的比較則交由 Agent 來代理，亦能防止決策權重邏輯不一致的問題發生。

3.4 議價協商機制與決策屬性權重的產生

主要是利用 Fuzzy Membership Function 來評估各個屬性質，使其正規化得以加總成一議價能力，買賣雙方依此來做為調整價格的幅度。在賣方價格區間主要是由各別的 Rule Base 所產生，可能是透過公司內部 CRM，對每一客戶不同偏好，給予不同的價格或是附屬配備等變動，以迎合消費者需求。如【圖 3.4-1】賣方給予買方優惠或是過時促銷車款，如此賣方價格區間就會向左移；而在雙方還是有所差距時，賣方可能提出贈送附屬商品或是低價出售方式，希望買方能提高購買價格，若接受買方價格區間就會向右移。【表 3.5-1】清楚列出賣方代理人的 Rule Base，而買方代理人的 Rule Base 產生，在下一章節的【圖 4.2.3】Selecting Suitable Goods User interface 中，使用者可輸入代理人決策參數。



【圖 3.4-1】賣買雙方議價協商過程

【表 3.4-1】Seller Negotiate Rule Base

車款新舊優惠 2001/2/21				
原始價格	最大折扣	開始折扣月	最終折扣月	採用函數
213	0.1	4	12	1
附屬配備				
產品 ID	配備	基本配備	price	Discount
E240	ABS	0	1.3	0.2
E240	TCS	1	1.55	
E240	AIRBAGS	0	5.3	0.3
E240	真皮坐椅	0	1.2	0.3
E240	恆溫空調	1	0.8	
E240	CD 音響	0	0.8	0.1
E240	動電天窗	1	3.1	

D_p：由買方議價能力所產生的價格

S_p：由賣方議價能力所產生的價格

D_{np}：依買方各評準參數而產生之議價能力

S_{np}：依賣方各評準參數而產生之議價能力

P^*：由下列公式算出之協議價格

如果買賣雙方價格依然有差距，我們就依照雙方議價能力做為相互退讓的依據，議價能力越高者退讓越少，越低者價格退讓越多，因此我們導出下面公式來計算出合理退讓值。

If ($D_p < S_p$)

$$P^* = \left(\frac{D_p + S_p - D_{np} S_p - S_{np} D_p}{2 + D_{np} - S_{np}} \right)$$

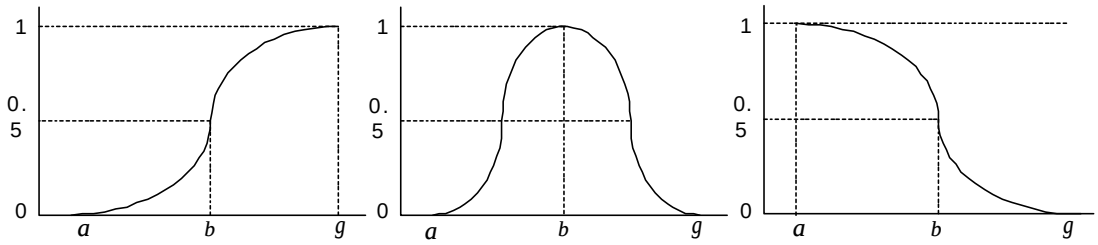
Else

$$P^* = \left(\frac{D_{np} S_p + S_{np} D_p}{D_{np} + S_{np}} \right)$$

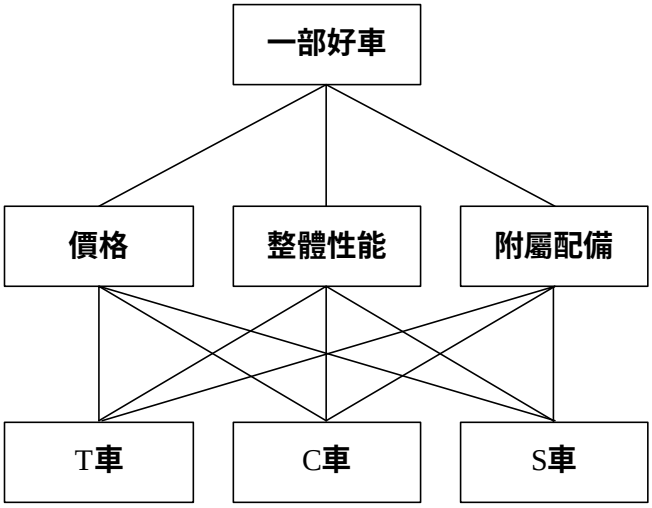
End if

經過上述之協商機制產生買方可接受之價格與附屬配備，之後利用 AHP 來做評選優劣。而這一部分是透過第三中立者對同類型產品的統合評估值，經過 Membership function 正規化，對其屬性做兩兩比較所產生，同時也遵循使用者所下達的偏好函數，提供買方在收個方案中的決策指標。其計算流程如下例：

【圖 3.4-3】為階層屬性與替代方案關係，使用者針對第二層輸入相對權重值而替代方案間的比較，如下面的 compare matrix。【圖 3.4-2】為使用者可選擇之偏好函數，將 criteria 做一正規化，以便做比較運算。



【圖 3.4-2】使用者個人之偏好函數



【圖 3.4-3】AHP 階層屬性與替代方案關係圖

L1-L2 **判斷矩陣使用下列公式計算出特徵向量，並且檢查邏輯權重是否一致。這一部份是由買方依個人需求所下的權重值。**

$$W_i = \sum_{j=1}^{n} a_{ij} \div \sum CompareMatrix$$

$$\Delta w^T = \sum CompareMatrix \times W^T$$

$$ConsistencyIndex(CI) = \frac{\frac{1}{n}\sum_{i=1}^{n}\frac{ithEntryIn \quad \Delta w^T}{ithEntryIn \quad W^T} - n}{n-1}$$

【表 3.4-2】Values of the Random Index(RI)

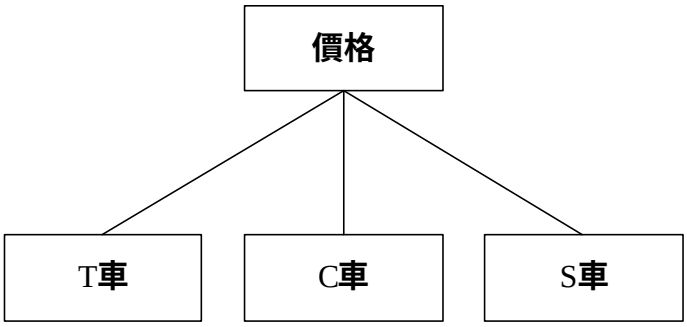
n	2	3	4	5	6	7	8	9	10
RI	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.51

if CI / RI < 0.10 ,the degree of consistency is satisfactory.

【表 3.4-3】L1-L2 判斷矩陣

L1-L2 判斷矩陣				
一部好車	價格	整體性能	附屬設備	特徵向量
價格	1	1/3	2	0.26
整體性能	3	1	4	0.61
附屬設備	1/2	1/4	1	0.13

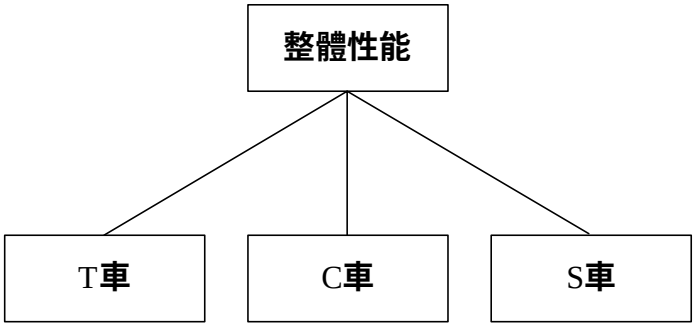
在下例第二層與第三層間的權重屬性分配，是由產品本身的屬性值（【表 3.3-1】Goods Ontology），透過專家所評估屬性良劣範圍（【表 3.3-2】分類評準範圍規則表）與買方個人設定之偏好 function，而計算出一 fuzzy 值，經兩兩比較產生一 Compare Matrix 後，得到該屬性權重值。如下：



【圖 3.4-4】價格屬性與替代方案階層關係圖

【表 3.4-4】L2 價格-L3 判斷矩陣

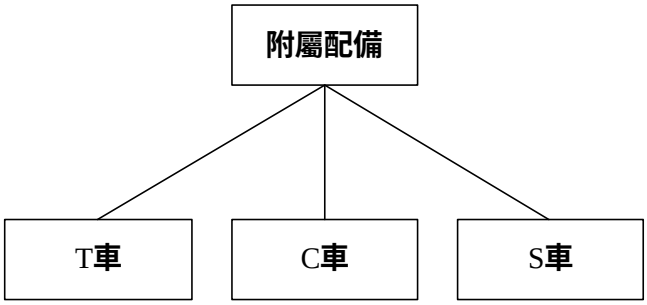
L2 價格-L3 判斷矩陣				
好價格	T 車	C 車	S 車	特徵向量
T 車	1	1/3	2	0.26
C 車	3	1	4	0.61
S 車	1/2	1/4	1	0.13



【圖 3.4-5】整體性能與替代方案階層關係圖

【表 3.4-5】L2 整體性能-L3 判斷矩陣

L2 整體性能-L3 判斷矩陣				
整體性能	T 車	C 車	S 車	特徵向量
T 車	1	1/4	2	.14
C 車	4	1	2	.57
S 車	2	1/2	1	.29



【圖 3.4-6】附屬配備與替代方案階層關係圖

【表 3.4-6】L2 附屬配備-L3 判斷矩陣

L2 附屬配備-L3 判斷矩陣				
附屬配備	T 車	C 車	S 車	特徵向量
T 車	1	2	1/2	.31
C 車	1/2	1	1/3	.16
S 車	2	3	1	.53

經由第二層-第三層比較矩陣結果與第一層-第二層使用者所訂定之相互重要程度後，利用下表即可計算出的最後結果，提供給使用者做決策的指標。如下：

【表 3.4-7**】**L2-L3 **判斷矩陣**

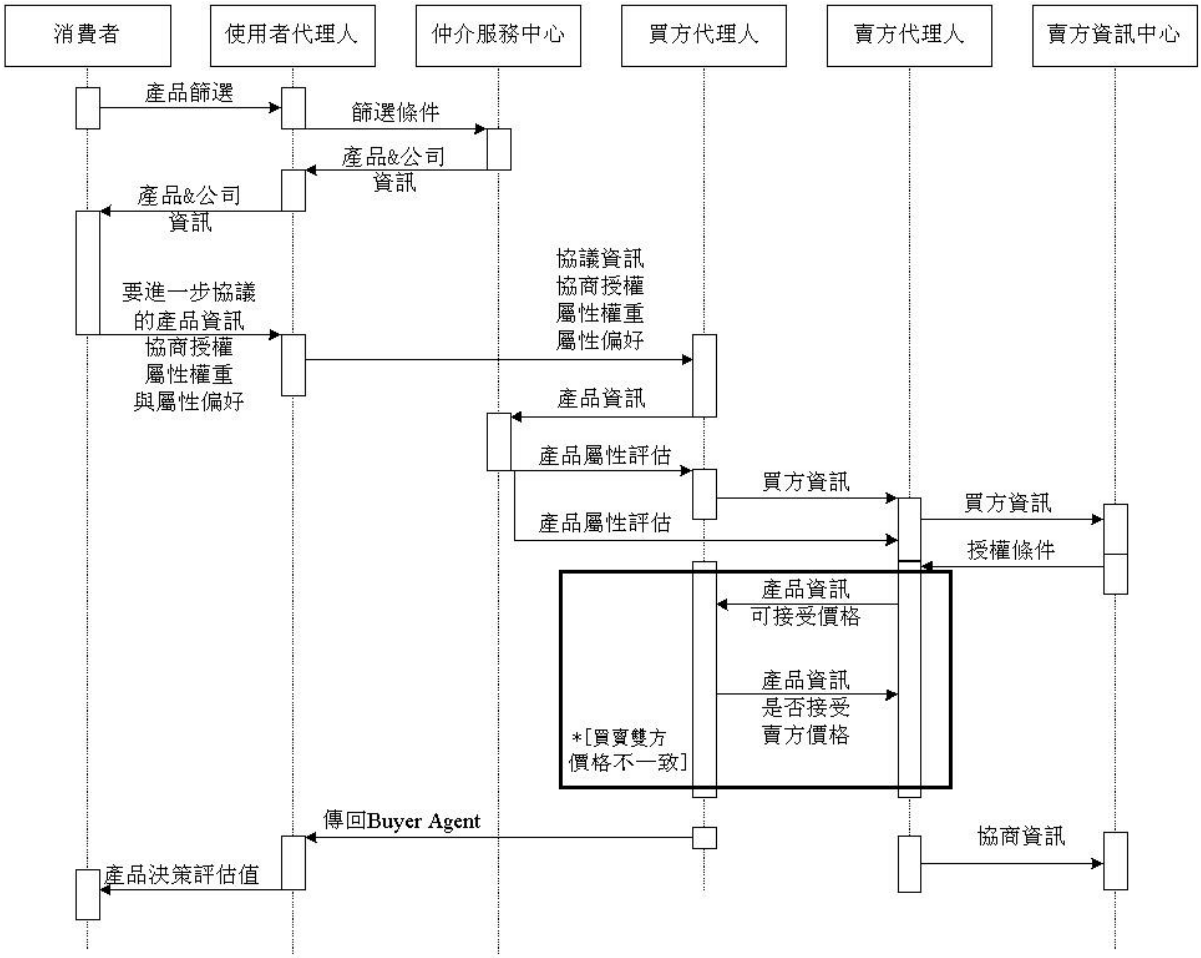
L2-L3 判斷矩陣				
L2 / L3	**價格**	**整體性能**	**附屬設備**	Result
	0.26	0.61	0.13	
T **車**	.55	.14	.31	.2687
C **車**	.22	.57	.16	.4257
S **車**	.23	.29	.53	.3056

第肆章、系統實作與實驗結果

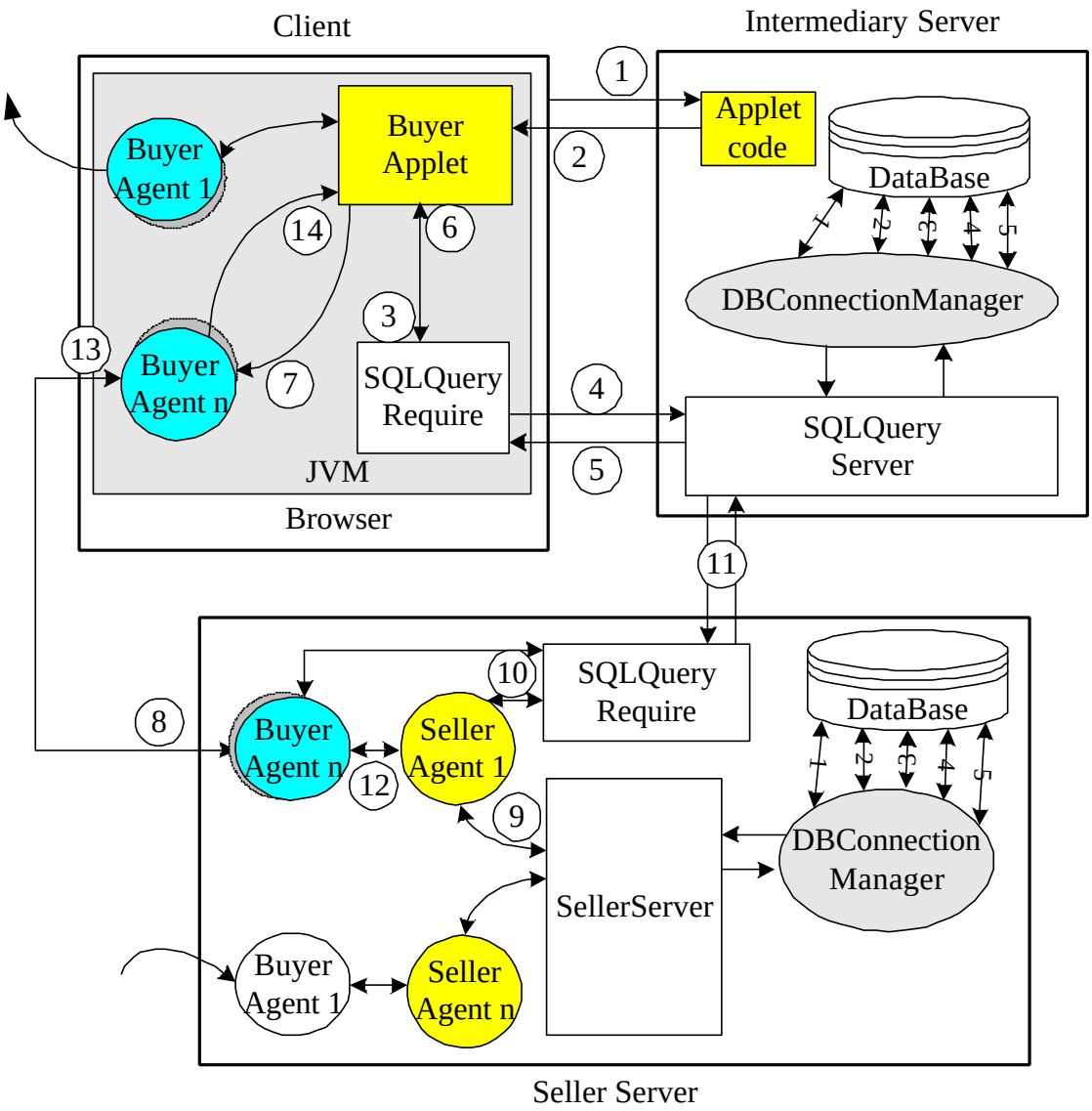
本章節會詳細說明本系統運作核心架構與整體流程，如何建構行動式代理人平台與內部各個類別間是如何運作等等，都會在下面做進一步的解釋。接下來是本系統的呈現畫面，亦會說明其操作方式。最後對使用者所輸入之參數與產出結果是否真能反應其偏好做一探討。

4.1 系統運作核心技術介紹

接下來我們會依據【圖 4.1-2**】相關類別之運作流程，配合【圖** 4.1-1**】系統流程紹介此一分散式交易市集之實作方法。**



【圖 4.1-1**】系統運作流程**



【圖 4.1-2】各個類別之間的運作關係圖

下列為類別(class)**與類別間運作流程中所傳遞訊息資料：**

(1) Http Request

(2) Download Applet (Buyer Applet)

(3) SQL-Query Request

(4) Creating a Socket Connection for Return Query Result Set

(5) Return SQL-Query Result Set, then Disconnection

(6) Get the Result Set then Generating a Setting Form

(7) Generating Buyer Agent and Setting The Criteria Values and Weight

(8) Dispatching Buyer Agent to Seller Server

(9) Accepting Buyer Agent and Generating Seller Agent

(10) The same as Step (3), (6)

(11) The same as Step (4), (5)

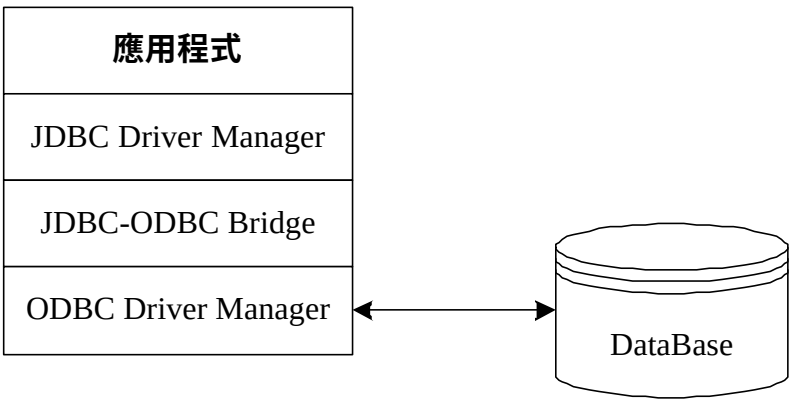
(12) Negotiating

(13) Dispatching Buyer Agent to Client (buyer)

(14) Providing the Information of Proposals Evaluated to Buyer Agent

4.1.1 系統對資料庫的連結方式

本系統必須要有一個的資料庫來儲存產品的 Ontology、廠商資訊、買方資訊、同類型產品屬性評估資訊等等，因此我們建構一個基本 Client-Server 架構，如此買賣雙方以便到仲介服務中心(Intermediary Data Center)擷取所需的資訊。而 Java 與 Ms-SQL 的連結只有透過 JDBC-ODBC Bridge，利用這個 Bridge 我們可以將 JDBC 的命令轉換成 ODBC，如此只要是支援 ODBC 的資料庫，即可以存取資料庫中的資訊。以下是 JDBC 與 ODBC 建立 connection 的架構圖。



【圖 4.1.1-2】JDBC-ODBC Connection Architecture

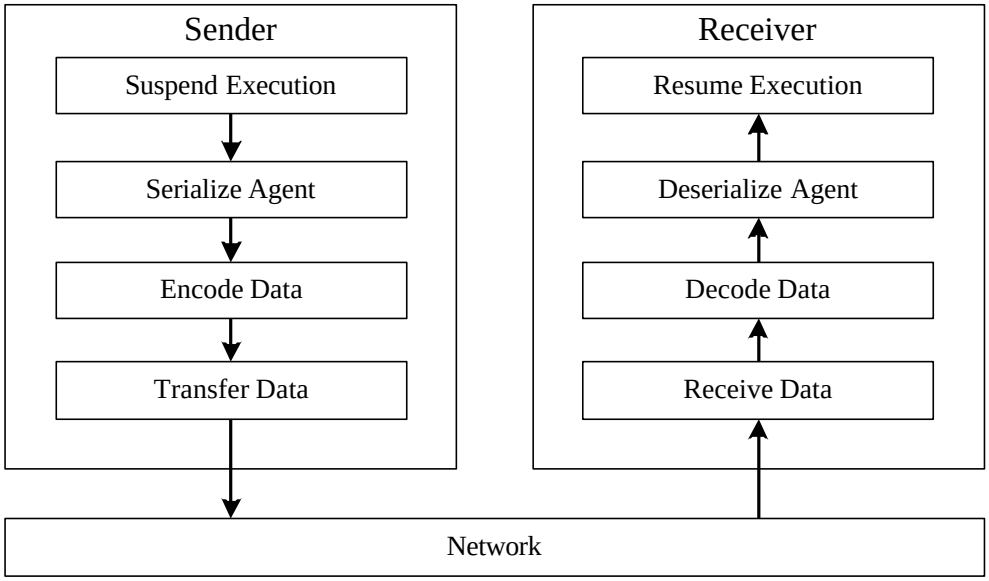
在 Client-Server 上最大的效率問題點是在與資料庫連結的地方，而我們使用了 connection pool 的方法來提高系統效能，避免過多的 connection

佔去太多的資源。如【圖 4.1-1】中的 Intermediary Server 與 Seller Server 都是採用了 DBConnectionManager.class，它先開起了五個 connection，而後要下達 Query 時就先 getConnection()，使用完了之後就 Free 掉，如果都在使用中的話，其他 user 就會 queue 在後面，等待 connection 的釋放。

BuyerApplet、Buyer Agent、Seller Server 與 Seller Agent 等 class，在擷取 Intermediary Server 評準資料庫的資訊時，主要是利用 SQLQueryRequire 程式，SQLQueryRequire 利用 Java Socket API 與 SQLQueryServer 建立起連結，之後把 client 端所要下的 query 代碼、keyword 或 SQL-code 傳送給 SQLQueryServer，而 SQLQueryServer 再把執行結果存放在 Vector 中，Vector serializable 後，即可把資料回傳給 client 端。

4.1.2 代理人系統運作方式

為了可以將 Agent 派送到賣方伺服器上執行，所以在賣方伺服器中會有個 Seller Server 等待 client 派送一個 Agent，當 Seller Server 接收到此一訊息，即會 new 一個 SellerAgent Object 與 BuyerAgent 做議價協商，而 BuyerAgent 在派送與接收過程如【圖 4.1.2-1】。



【圖 4.1.2-1】Agent Transfer

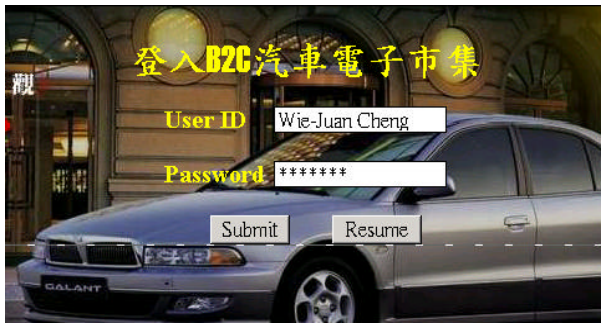
買賣雙方的代理人都在追求對自己較好的效用值與利益，所以雙方是處於競爭的狀態，兩者在協商議價過程主要是將價格與附屬配備給對方，而賣方代理人則是儘量迎合買方需求，以期達成交易。

4.2 系統介面介紹

本系統主要著重在消費者使用操作畫面，主要包含以下幾個主要的系統單元介面：系統登入、產品篩選(Input Constraint)、選擇要進一步協商之商品(Selecting Suitable Goods)、授權代理人添購之附屬配備與其他交易條件、代理人決策屬性設定、議價協商後之決策評估結果等。

4.2.1 系統登入

為了使賣方能夠分享消費者資訊，買方必須登入檢驗名稱及密碼的程序，賣方或許透過公司本身的 CRM，給更準確的行銷策略。



【圖 4.2.1-1】系統登入畫面

4.2.2 產品篩選(Input Constraint)

此一畫面會從 Intermediary Server 的產品資料庫中擷取出相關資訊，進而產生篩選條件介面。

【圖 4.2.2-1】產品篩選畫面

在條件值的輸入時，必須先勾選該篩選條件，若未輸入上下限值，程式會自動乎略，條件上下限值亦可只輸入其中一個。按下篩選後，Buyer Applet 就會向 Intermediary Server 取得符合之商家。

4.2.3 選擇要進一步協商之商品、授權代理人添購之附屬配備與其他交易條件(Selecting Suitable Goods)

在 Selecting Suitable Goods 中，使用者可以選擇要進一步協商的產品與廠商資訊，還可以輸入要授權代理人選購之附屬配備或其他交易條件值等。

【圖 4.2.3-1】Selecting Suitable Goods User interface

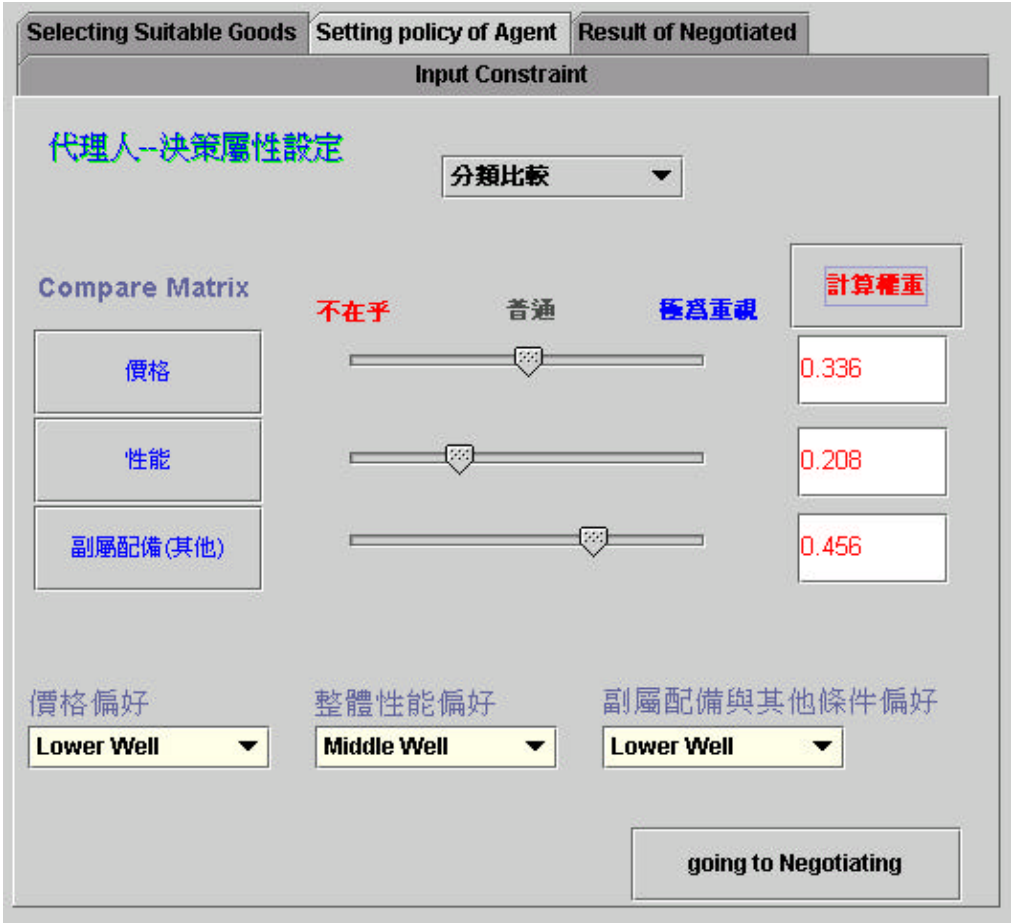
4.2.4 代理人決策屬性設定

代理人決策屬性的設定畫面中，使用者可以對其第二層決策屬性下其權重，例如【圖 4.2.4-1】，性能：價格＝4　代表性能比價格好的程度是 4 個單位，而其範圍是 9　1/9 之間（a：b＝9 極端重要　1/9 極端不重要，1 代表一樣重要)，在 Compare Matrix 中會判斷使用者所輸入的權重比是否一致，當性能比價格重要，而價格又比附屬配備重要時，則附屬配備不能比性能重要，這就是所謂權重比值的一致性。

【圖 4.2.4-1】代理人決策屬性設定

在底下有價格偏好、整體性能偏好、附屬配備與其他條件偏好等三個下拉式選單，使用者可對各別評準屬性設定其偏好 function，在決策評估結果的產出就會依其偏好設定不同而很大的差異。

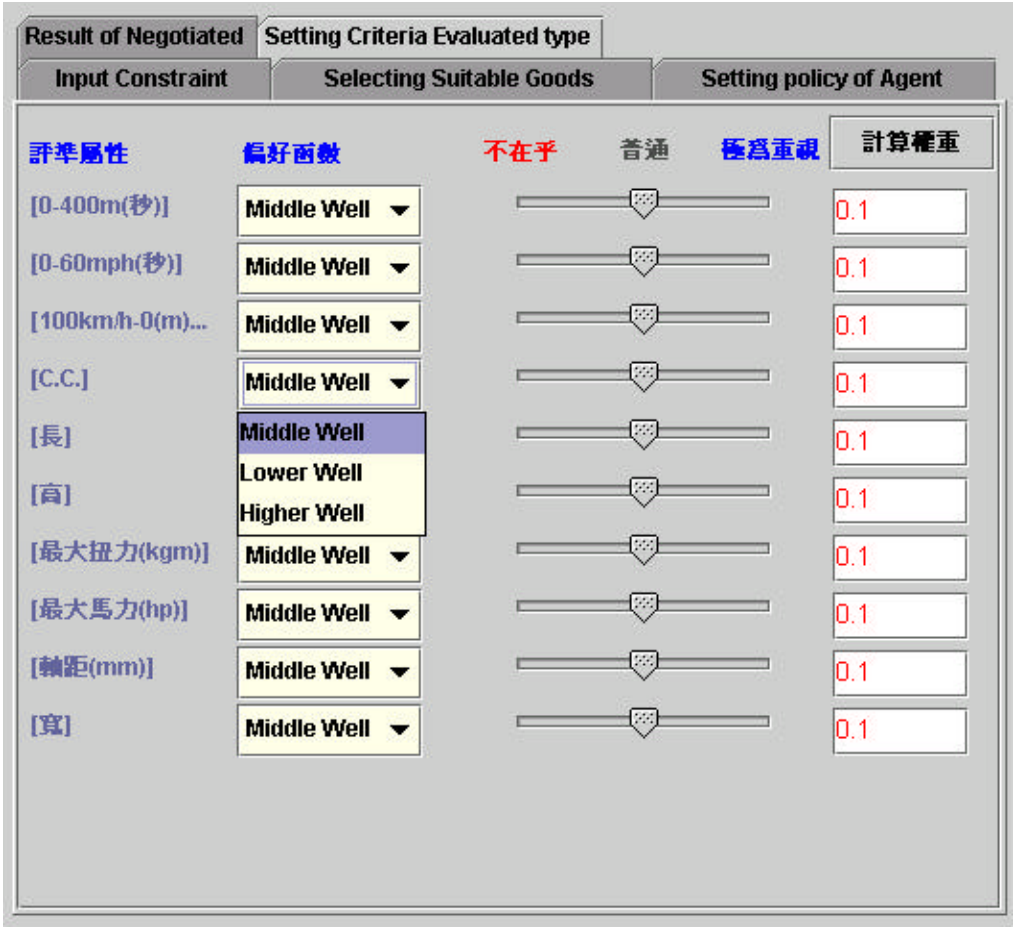
但是考量於使用者在進行兩兩比較中，其複雜性是很高的，當評準屬性一多時，在輸入上是一個問題，並且還必須判斷其一致性，如此是非常累人的。因此我們改變用較簡易的方式來做輸入如下【圖 4.2.4-2】：



【圖 4.2.4-2】代理人決策屬性設定 2

4.2.5 決策屬性－性能之下層子評準屬性之偏好評估

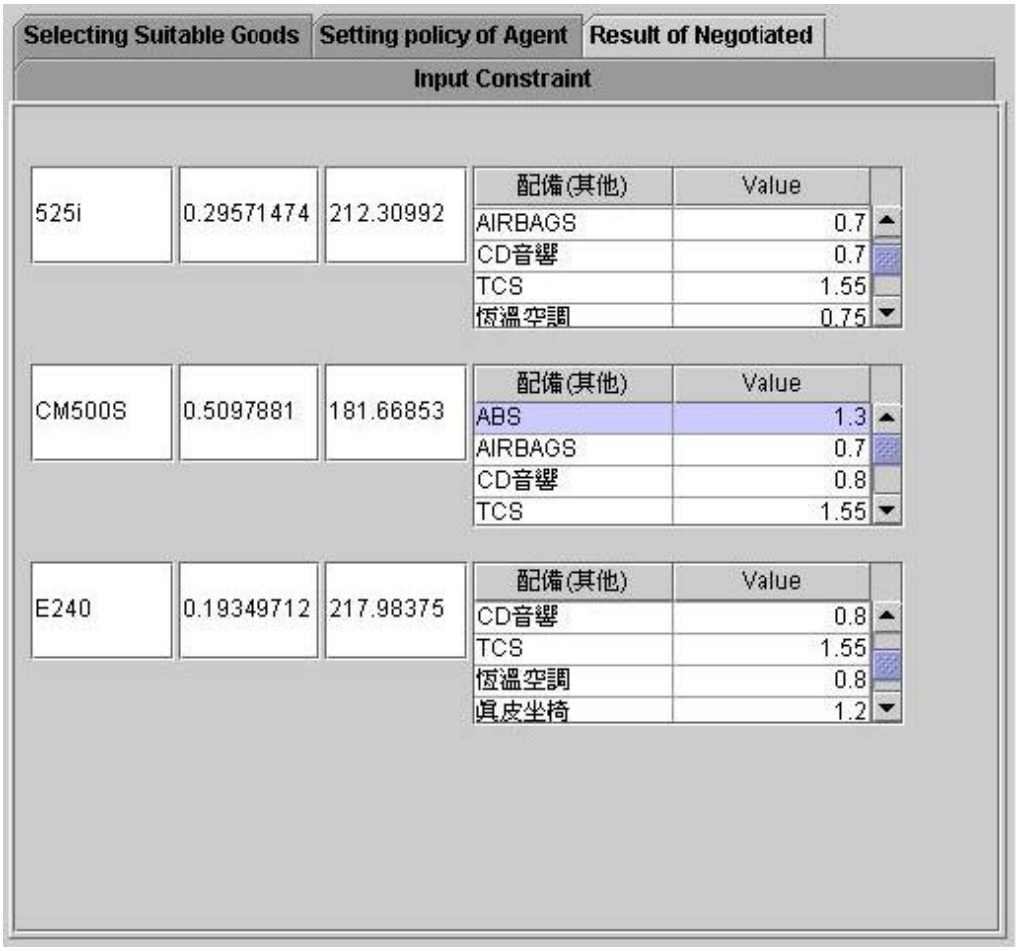
使用者亦可對細部的評準屬性設定個人之偏好，使最後 Agent 所提供的決策評準值能更加精準地符合買方需求。



【圖 4.2.5-1】Setting Criteria Evaluated Function Type

4.2.6 議價協商後之決策評估結果

派出去 Agent 經過反復協商完成之後被派遣回到 Buyer Applet 中，透過 AHP 方式，將各個替代方案的評準屬性兩兩比較產生【圖 4.2.5-1】的結果。

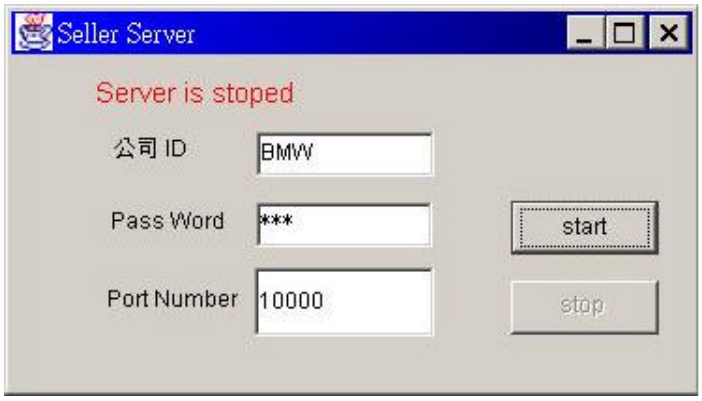


【圖 4.2.6-1】Result of Negotiated

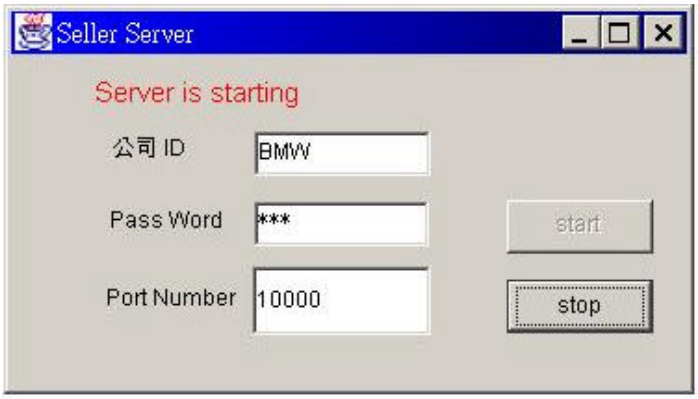
第一欄是產品名稱。第二欄是產品間之良劣評估的相對比值，也就是說買方可依其數據做為購買決策方針。第三欄是產品價格，而後面的 Table 則是存放該車款附屬配備。

4.2.7 賣方伺服器(Seller Server)啟動畫面

賣方管理者必須輸入公司 ID 與 Password，才能跟 Intermediary Server 註冊，並告知該伺服器已經開始運作，可以讓買方代理人進行協商議價之服務了。



【圖 4.2.7-1】SellerServer 停止服務狀態



【圖 4.2.7-2】SellerServer 啟動服務狀態

4.2.8 仲介服務中心(Intermediary Server)

Intermediary Server 主要是提供買賣雙方的各別資訊與產品屬性評估資訊，並不對其協商過程做監控之動作。

【圖 4.2.8-1】啟動 Intermediary Server

4.3 實驗結果

接下來我們以把實際汽車產品的數據輸入到資料庫中，並假設了數種使用者所輸入的重權狀態與偏好函數，探討其結果是否能確實符合使用者需求偏好。

4.3.1 偏好低價位、高性能、配備佳之採購決策評估

我們首先選定價格 60~80 萬之間的車種，在權重分配上以性能(0.518)最為重要，其次是價格(0.301)最後是配備(0.181)，在個人偏好上是價格越低越好、配備越多越好、高性能導向，各個車種評估的結果如下【表 4.3.1-1】：

【表 4.3.1-1】車款效用評估表 1

車款	價格(萬)	價格效用值	整體性能效用值	配備(萬)	配備效用值	總效用相對比值(100~0分)
CrollaAltis	67.73	0.836	0.485	14.45	0.396	23.77
Virage	67.98	0.821	0.385	15.65	0.515	21.92
Tierra	65.45	0.940	0.301	14.45	0.395	19.16
Sentra180	67.91	0.825	0.343	13.75	0.332	18.95
CivicFerio	67.70	0.838	0.265	12.95	0.268	16.18

在權重的重新分配下，若配備的權重程度高於價格時，代理人會自動購入其他附屬配備，以求得較高的效用值，如下【表 4.3.1-2】：

【表 4.3.1-2】車款效用評估表 2

車款	價格(萬)	價格效用值	整體性能效用值	配備(萬)	配備效用值	總效用相對比值(100~0 分)
CrollaAltis	73.73	0.342	0.485	20.45	0.885	23.28
Virage	72.77	0.425	0.385	20.45	0.885	21.29
Tierra	71.45	0.556	0.301	20.45	0.885	20.19
Sentra180	73.91	0.326	0.343	19.75	0.847	18.84
CivicFerio	73.70	0.345	0.265	18.95	0.797	16.48

4.3.2 偏好中間價位、平均性能、配備中等之採購決策評估

在此一實驗中我們不輸入價格範圍，且所有的屬性權重一致，來對所有車款進行不分類的混合比較，如下表：

【表 4.3.1-2】車款效用評估表 3

車款	價格(萬)	價格效用值	整體性能效用值	配備(萬)	配備效用值	總效用相對比值(100~0 分)
Zafira	88.73	0.473	0.804	9.95	0.438	15.42
Joice	84.21	0.398	0.583	11.9	0.655	13.49
Premacy	72.80	0.194	0.562	12.9	0.766	10.27
CM500S	194.05	0.337	0.209	11.95	0.661	9.89
CrollaAltis	67.76	0.103	0.409	14.45	0.939	8.34
Sentra180	67.91	0.106	0.306	13.75	0.846	7.39
E240	232.9	0.0086	0.678	11.95	0.661	7.34
Virage	67.98	0.107	0.247	15.65	0.9349	7.29
525i	226.69	0.054	0.536	11.95	0.661	7.28
CivicFerio	67.70	0.102	0.281	12.95	0.772	6.79
Tierra	65.45	0.0619	0.218	14.45	0.938	6.34

上表的例子是在不分類的狀況下，而代理可以依據使用者個人的偏好規則尋找並計算出最適合的車款，雖然在不同車款中做比較是困難的，且並無多大意義存在，但我們還是計算出整體車型中各個屬性歸屬函數區間值，期能在使用者在最少的資訊輸入下，亦可評選出最符合使用者的車型種類。

第伍章、結論與未來研究方向

5.1 結論

經由我們所建構的自動化協商議價之電子市集，獲得了以下成果：

1. **結合專家經驗**

   在電子市集競爭環境中，如何不被取代，進而併購他人的能力，重點就是在於 Intermediary **是否能提供不易被模仿的加值服務。而本系統主要結合了專家知識在產品各個屬性的評價與良劣值範圍的訂定，如此才能讓買賣雙方在協商過程得到一良好評估依據。在於中立第三仲介者評估是否公正客觀，則是大家所必須考量的重點。**

2. **代理成本的降低**

   在過去企業中，採購重要的原料時，有的可能外包廠商代理進行採購之行為，要不然就是由公司中的採購部門進行採購評估。在本系統環境之下，對產品做了良好的評估，並可依買方個人偏好，進行協商議價自動化過程，進而提供產品採購決策方針，大幅減少使用者在商品上評選的時間，增加企業運作效率。

3. **分享計算資源、減少網路傳訊負載**

   利用本系統架構相較 N-tier **集中式架構，有效運用** client **端與賣方伺服器之計算資源。而且在** Agent **間的溝通行為都是在當地伺服器端進行，相較於** Client-Server **架構，確實去除在網路間溝通訊息的傳遞，且在於網路品質不佳的環境中，亦能展示其優點。**

4. **縮短系統開發時程**

   在本系統運作架構之下，如要增加或減少其評準屬性，並不需更動到程式本身，而且新增產品類別資訊只要依照規則輸入產品 Ontology **與評準屬性值範圍等，馬上就可為電子市集增加一新種類產**

品，可說是增加了程式再使用率與彈性，如此不必再為新產品開發新環境，大幅提高系統開發效率。

5. 整體執行效率

本系統同時將數個代理人派送至所要協商議價之廠商伺服器中進行協商，完成協議後再傳回原始的 Applet 中，再進行替代產品間的決策權重分析，給予使用者做採購決策依據；相較於集中式或單一代理人等模式的執行效能都要來的好，而在如此的 Agent 傳播方式下，網路的負載相對的也提高，但是我們取決於效率能掛帥為由，而採取此一方法來提高整體運作效能。

5.2 未來研究方向

在結論中有討論到，第三中立仲介者是否能公正客觀評估各項產品屬性質的確是個重要的問題，還有產品的等級與分類亦是由 Intermediary 所制定，希望未來可以利用統計分群、模糊類神經網路等方式，冀能去除人的因素，以達到真正的公正客觀。而在於協商過程中的訊息、消費者權重設定類型等等，亦可儲存於資料庫中做為未來 Agent 協商時的 Case Base，以提高在協商過程的效率，並能確切符合使用者偏好，賣方代理人伺服器若加上公司內部 CRM 系統，即可對顧客採取更加有效的行銷策略，提高與買代理人在決策行為上的準確度。這些都可以做為未來加強本系統效率與可用性，如此必能更加健全系統整體運作機制。

附錄一

Table Name	UserInf	
Column Name	**Data Type**	**Column Size**
UserID	Char	10
Name	VerChar	50
Sex	Boolean	
Birthday	Datatime	
Income	Double	
Position	Char	20
Family	int	

Table Name	Company	
Column Name	**Data Type**	**Column Size**
CompanyID	Char	10
Name	VerChar	50
ServerIP	Char	39
PortNum	int	

Table Name	GoodsInf	
Column Name	**Data Type**	**Column Size**
GoodsID	Char	20
Name	VerChar	50
CompanyID	Char	10
Price	Float	
IssuingDate	datatime	

Table Name	Equipment	
Column Name	**Data Type**	**Column Size**
EID	Char	20
Name	VerChar	50
Price	Float	

Table Name	GoodsHaveE	
Column Name	**Data Type**	**Column Size**
GoodsID	Char	20
EID	Char	20

Table Name	CriteriaInf	
Column Name	**Data Type**	**Column Size**
CriteriaID	Char	20
Name	VerChar	50

Table Name	GoodsCriteria	
Column Name	**Data Type**	**Column Size**
GoodsID	Char	20
CriteriaID	Char	20
ClassType	Char	10
EValue	VerChar	20

Table Name	Class_Type	
Column Name	**Data Type**	**Column Size**
ClassType	Char	10
Name	VerChar	50
CriteriaID	Char	20
MaxValue	Float	
AvgValue	Float	
MinValue	Float	
ByerMfunction	int	
SellerMfunction	int	

參考文獻

Reference:

[1] Lindermann, M. A. and Schmid, B. F., "***Framework for Specifying, Building, and Operating Electronic Markets***", International Journal of Electronic Commerce, (3:2)winter 1998-1999, pp:7-21.

[2] Guttman, R.H., Moukas, A.G. and Maes, P. "***Agents as Mediators in Electronic Commerce***", Electronic Markets, Vol. 8, No. 1,1998, pp. 22-27.

[3] Guttman, R., A. Moukas, and P.Maes. "***Agent-mediated Electronic Commerce : A Survey***" to appear,Knowledge Engineering Review,June1998.

[4] Nwana, H.S. "***Software Agents: An Overview***", Knowledge Engineering Review, Vol. 11, No. 3, 1996, pp. 205-244.

[5] Castelfranchi. C., ***Guarantees for autonomy in cognitive agent architecture.*** In M. Wooldridge, and N.R. Jennings, editors, Intelligent Agents : Theories, Architectures, and Languages (LNAI Volume 890), 1995,pp.5-70. Spr inger-Verlag : Heidelberg , Germany.

[6] D.B. Lange and M. Oshima, "***Programming and Deploying Java Mobile Agents with Aglets"***, Addison Wesley, 1998.

[7] 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*, 2000, pp. 440-444.

[8] M. Barbuceanu and W.K. Lo, "***A multi-attribute utility theoretic negotiation architecture for electronic commerce***", *Proceedings of the 4th international conference on Autonomous agents*, 2000, pp. 239-246.

[9] J.Y. Kang and E.S. Lee, "***A negotiation model in electronic commerce to reflect multiple transaction factors and learning***", *Proceedings of 12th International Conference on Information Networking*, 1998, pp. 275-278.

[10] Saaty, T.L. (1996),"***Decision for Leaders : The Analytic Hierarchical Process***",2nd ed. University of Pittsburgh Press.

[11] Lu, M.H., Madu, C.N.,Kuei, C.H.,and Winokur, D., "***Integrating QFD, AHP, and Benchmarking in Strategic Marketing***", Journal of Business and Industrial Marketing, Vol.9, No1, 1944. P41-50.

[12] ***由決策習慣探討 AHP 之評估方法***，馮正民、李穗玲，中華管理學報，第一卷

[13] 資訊管理，吳琮璠、謝清佳著，智勝出版，2000/5.

[14] 方定國，"***連鎖加盟因素選擇評估－AHP 決策模式***"，國立空中大學管理與資訊學系，管理與資訊學報，民 89，第五期，P 1-26.

[15] Aparico, G., "***The role of intelligent agent in the information infrastructure***",IBM's White paper,1995

[16] Selker, T. "***Coach:a teaching agent that learns***", communications of the ACM, Vol.37, NO.7,1994

[17] Smith, D. C., et al, "***KidSim : programming agent without a Programming language***", communications of the *ACM,NO.* 37, 7 (Jul. 1994), Pages 54 – 67

[18] 李富民、陳瑞斌，1999，"以智慧型代理程式建構網路拍賣系統"，"第五屆資訊管理研究暨實務研討會"，p53-63

[19] 林毓忻、苑守慈，2000，"***Negotiation-Credit Based Group Buy/Group Sell in E-Markets***"，"第六屆資訊管理研究暨實務研討會"

[20] 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,March 1999.

[21] 王善德、王俊程，"***以合作式談判代理人輔助 B2B 網站中虛擬供應鏈之形成***"**，第** 11 **屆全國資訊管理學術研討會，**2000。

[22] 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*, 1999, pp. 279-286.

[23] 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*, July/August 1999.

[24]數位時代雜誌，第 10 **號，**p127-132**，**2000/4/1

[25]資訊與電腦，p102-106**，**2001/5