一個行動代理人系統之塑模與驗證方法－以網路買方召募為例

The Modeling and Verification of Mobile Agent-based Systems—Illustrated with Internet Buyer Recruiting

林加賢 (Jia-Sian Lin)
南台科技大學 資訊管理學系
m9190106@webmail.stut.edu.tw

陳炳文 (Ping-Wen Chen)
南台科技大學 資訊管理學系
pwchen@mail.stut.edu.tw

摘要

聯合議價乃是凝聚多數人之購買力，形成強大之買方，藉其優勢與賣方進行議價，以創造優惠之價格。而以代理人為機制之聯合議價乃是利用軟體代理人(software agents)代表買賣雙方進行議價，其過程是先由買方的軟體代理人在網路上凝聚購買力，然後再與賣方的軟體代理人進行議價，以此方式為個別買方創造較低的購買價格。以此為例，我們提出了一個包含五個步驟、由上而下(top-down)的方法，有系統地利用物件導向塑模語言 UML 及其延伸版本 Agent-based UML (AUML)，從不同的面向描繪此設計的概念，而且我們提出以“狀態為基礎的模擬”方式來驗證此系統模型，此法對整個系統的開發時程、成本與品質相信會有相當大的影響。

關鍵字：UML、Agent、Agent-oriented Software Engineering

一、 緒論

目前在網路上，聯合議價[2][3][5][6]大多是由供應商透過仲介商在網路上召募買方，再由使用者主動上網來加入，然而，利用此一方式，即使加入電子郵件的推銷，仲介商也只能被動地等待買方的加入，直到買方聚合的截止時間為止。

基於以上的缺失，我們提出一加入行動代理人(mobile agents)的 client-server 架構來加以改善，此架構中包含一個伺服端的仲介網站，利用行動代理人可於網路上遷移(migration)的特性，主動至客戶端向買方推銷商品，其好處在於行動代理人可以採用更積極主動的方式，加入商業推銷的手法，將商品資訊有系統、有策略地介紹給各買方，藉此來吸引更多人的回應與加入，而且也可以藉由此互動的過程，幫助供應商分析買方對商品各種規格的偏好，以便進行顧客分群，並在議價之前預作價量分析。

在本文中，我們將特別針對凝聚買方的部分來著墨，利用目前軟體工程界最廣為接受的物件導向塑模語言 UML (Unified Modeling Language) [4][8] 及其延伸版本 AUML (Agent-based Unified Modeling Language) [21]來描繪之。並且根據設計此架構的需求，我們加入新的符號於 AUML 的 protocol diagrams，以擴增其功能。

因為系統分析與設計階段所產生的大部分為設計文件，所以一般的作法都是採取人工的方式去發現錯誤，如 review 和 inspection [4]，但是對於隱藏於複雜設計中的錯誤，還是難以發現，因此，本文提出採用以“狀態為基礎的模擬”(state-based simulation) [9][10]方式所建構的軟體工具 JMICE [10]來發現錯誤。

本文接下來將在第二節對代理人及 UML 做相關文獻的探討；在第三節中，將利用 UML 及 AUML 所提供的圖型來塑模(modeling)本文所提出來的架構；第四節探討系統模型的驗證方法；第五節是本文的結論。

二、 相關文獻

(一) 代理人

代理人是近幾年來相當流行的一種技

術，它們可以代表使用者執行一些工作。代理人具有自主性並且可以在異質環境下獨立自主的運作，根據 Object Management Group (OMG)的定義[19]，代理人具有以下幾點重要的特性：

1. 自主性(autonomous)：代理人不需外界的干涉，它們可以自我控制內部狀態並且根據經驗來採取行動。
2. 互動性(interactive)：代理人會與它身處的環境或其它的代理人進行溝通以完成使用者交付的任務。
3. 適應性(adaptive)：代理人可以對目前所處環境或其它代理人的要求產生回應，並且根據經驗修改其行為。

以上三點是代理人較為重要的幾個特性，而根據 OMG 的定義，代理人可分為軟體代理人(software agents)、自主性代理人(autonomous agents)、行動代理人(mobile agents)、智慧代理人(intelligent agents)等型態，然而此分類並不互斥，也就是一代理人可能同時屬於幾種不同的型態。

利用行動代理人的好處包括[1][15]：可以將計算工作移到各個客戶端的機器，以減輕伺服端的負荷，同時也可以減少互動過程中網路的負擔，並去除通過網路的時間；還有可以提昇對錯誤的容忍度，如在低頻寬或較不穩定的網路環境中(例如在無線電腦網路、無線通訊網路裡)，可能因收訊的問題而斷線，在此情況下，行動代理人可繼續執行而不致中斷。

(二) 統一塑模語言

統一塑模語言 UML(Unified Modeling Language)是軟體工程界當前最新的物件導向系統塑模語言，其提供了九種圖形(diagrams)，讓使用者從不同的面向(views)描述一個系統，以做為各方溝通與系統驗證之用，而其目的便在於以視覺化的方式來訂定、建構以及記錄一個物件導向系統。

UML 提供了一些延伸機制：模板型別(stereotypes)、標記值(tagged values)及限制(constraints)，以擴大 UML 所能支援的領域，在本文中，網路部署圖便利用了 UML 的部署圖及模板型別所構成的。

由於我們所提出的是一個多代理人系統(multi-agent systems) [18]架構，因此描述代理人之間的溝通協定(protocols)是一項很重要的議題。在[16]中所做的研究認為，利用現有的 UML 塑模語言及其所提供的延伸機制即可用來描繪代理人之間的溝通協定，但是，有許多可用來描繪代理人溝通協定的塑模語言[7][12][13][14][17][20][21]被提出，然而，我們認為 AUML 以 UML 為基礎，因此，其本身就具有了 UML 的延伸機制，可提供分析人員針對其領域的需求進行塑模，且對 UML 熟悉的人員可輕易的學會使用 AUML;而其它塑模語言，如 MSCs (Message sequence charts) [17]雖然是由 ITU (International Telecommunication Union) 所通過的一種標準視覺化語言，但是都沒有提供延伸機制，與我們的需求不符合，因此，我們選擇以 AUML 來塑模此一溝通協定，並針對 protocol diagrams 之不足，新增了幾個符號，這些符號可應用於同步或循序的溝通上，例如:一對多或多對一的拍賣、競標等。

三、 系統塑模

在本篇文章中，我們將利用以下的步驟來進行系統分析與設計的塑模：

(1) 由於此代理人系統為一個多代理人系統(multi-agent system)，所以我們將先區分出此系統中的各個代理人角色及其功能以利於後續的分析。

(2) 因為本系統利用行動代理人，所以先將各個角色在網路中所在的節點(nodes)位置，及可能的遷移路徑描述清楚。

(3) 針對召募買方此一任務，從高階的角度，分配各個角色所負責的活動，並決定活動間執行的順序（循序或平行）。

(4) 為完成召募買方此一任務，從高階的角度，定義角色之間的互動及其流程。

(5) 對於召募買方此一任務，從低階細微的角度，描述各個角色單獨的行為模式及其中行為狀態的轉換。

(一) 社會結構

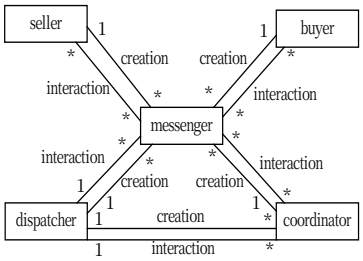

圖 1. 以類別圖描述角色間之社會結構

依功能上的需要，我們將此架構中的代理人分為 dispatcher、coordinator、messenger、buyer 與 seller 五種角色，各角色間因互動而有 creation 及 interaction 兩種關係，我們以類別圖(圖 1)來表示此一社會結構。各角色主要功能介紹如下：

A. dispatcher：其將代表仲介商接受外來之要求，如召募買方之要求，對於每一個要求，分別指派不同的代理人 coordinator 來負責。

B. coordinator：其將代表仲介商負責 dispatcher 指派之工作，如召募買方。

C. messenger：其將在網路上移動，負責傳達訊息，是唯一的行動代理人。

D. buyer ：其將代表顧客(買方)處理事務，如過濾不感興趣的商品，並安排向顧客進行商品推銷。

E. seller：其將代表供應商(賣方)處理事務，如對 dispatcher 提出召募買方要求。

以下介紹角色間的關係：

A. creation 關係：表示一角色的代理人產生另一個角色的代理人。dispatcher、coordinator、buyer、seller 為了與其他代理人溝通，會產生 messenger 並指派其傳送訊息，因可能需要傳送多次訊息，所以與 messenger 之間有一對多的關係；而 dispatcher 對於每一個外來的要求，也將產生一個 coordinator 來負責，因可能會有多個外來的要求，所以與 coordinator 之間有一對多的關係。

B. interaction 關係：表示兩個角色的代理人之間有互動。coordinator、buyer、seller 與 messenger 之間有多對多的關係；因 dispatcher 只有一個，所以與 messenger 之間有一對多的關係；同樣，dispatcher 與 coordinator 之間有一對多的關係。

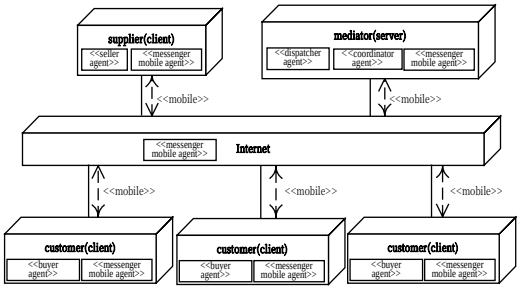

圖 2. 以部署圖描述角色位置及遷移路徑

(二) 網路結構

在本節中，我們將介紹本系統的部署情形並以 UML 的部署圖來描繪，如圖 2。圖中共有四種節點（nodes）：

1. mediator：代表一個使用者註冊及買、賣方交易仲介的主機，其中有 dispatcher 跟 coordinator 長駐，在此 client-server 架構中屬於 server 端。
2. customer：代表買方顧客的主機，其中有 buyer 長駐，屬於 client 端。
3. supplier：代表賣方供應商的主機，其中有 seller 長駐，屬於 client 端。
4. internet：代表有線及無線的網際網路，messenger 經由 internet 在以上三種節點之間遷移。

另外，我們利用 UML 的 stereotype 延伸機制，使用以下的符號來描述多代理人系統：

1.  ：以類別圖中代表類別的符號“方框”來表示一個代理人，並於方框內註以<<代理人角色名稱>>，方框位置

也代表了代理人部署的節點。

2. <-> ：以一個虛線雙箭頭符號來表示行動代理人遷移路線與方向，並於虛線旁加註<<mobile>>來說明；而圖中的實線則是部署圖的標準符號，它是用來表示兩個節點之間的連線。

(三) 活動流程

接下來，將特別針對召募買方此一任務，從高階的角度，分配各個角色所負責的活動，並決定活動間執行的順序，我們以圖 3 的活動圖來描繪其間的執行流程。圖中每一欄(swimlane)代表一個代理人所負責的活動，而 messenger1 及 messenger2 分別代表 seller 及 coordinator 所產生、派出的代理人 messenger。

→ 代表活動之間循序地進行，┼ 則代表數個活動可同時進行。以下我們將簡略說明活動的流程：

A. 供應商的代理人 seller 產生並指派一個 messenger 向仲介商傳達召募買方的請求，messenger 於接獲命令後，將從 supplier 節點遷移到 mediator 端並且向 dispatcher 提出請求。

B. dispatcher 接受請求後，將產生並指派一位 coordinator 來負責此任務。

C. 在買方召募期限之內，coordinator 將會發送 messenger 到每個上線的 customer 節點，與顧客的代理人 buyer 進行溝通，buyer 將安排 messenger 向顧客推銷商品，然後再將顧客意願等結果帶回 mediator 節點，回報給 coordinator。

D. 期限過後，coordinator 將會統計結果，並經由 messenger 將結果回報給供應商及每個加入的顧客知道。

(四) 互動流程

為完成召募買方任務，從高階的角度，定義代理人角色之間的互動及其流程，利用 AUML 中被我們延伸的 protocol diagram 來制定出一個可以重複使用的樣板，如圖 4。樣板(templates) [8]是一種描述具有重複使用(reuse)概念的設計樣式(design patterns) [11]的規格，它不能夠直接使用，必須先針對不同的需求進行客製化(customization)，並將樣板中的參數實例化(instantiation)後才能夠使用。例如此樣板可以被利用在其它聯合議價模式[6]中的買方召募。

首先，說明 protocol diagrams 中 AUML

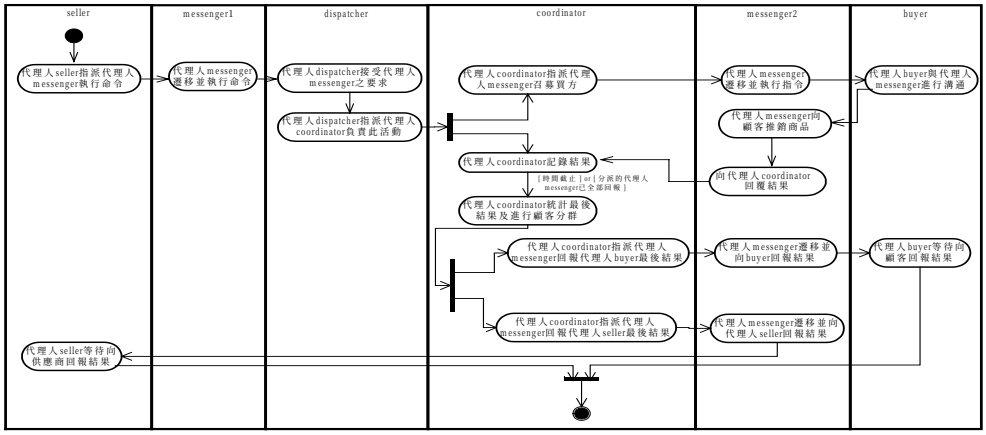

圖 3. 各代理人角色執行買方召募之活動圖

的標準符號，以及根據我們的需求，所加入的新符號：

1. ■：這是我們所新增的符號，代表一個行動代理人正在進行遷移，並加註<<mobile>>來表示之。

2. □ ：代表一個代理人，其名稱寫在框中，格式為“agent-name/role：class”，agent-name 代表一個代理人實體(instance)名稱；role 代表代理人實體所扮演的角

色；class 代表角色之類別。在進行分析與設計時，可只列出重要部分，

3. ⟶：代表兩個代理人間之“互動行為”，在連線的起始位置的代理人，是此互動行為的發起者，雖然只有一條連線，但此連線是可擴充，其可能隱含兩者間有數個訊息的來回傳遞，或只是一個簡單的訊息傳遞。

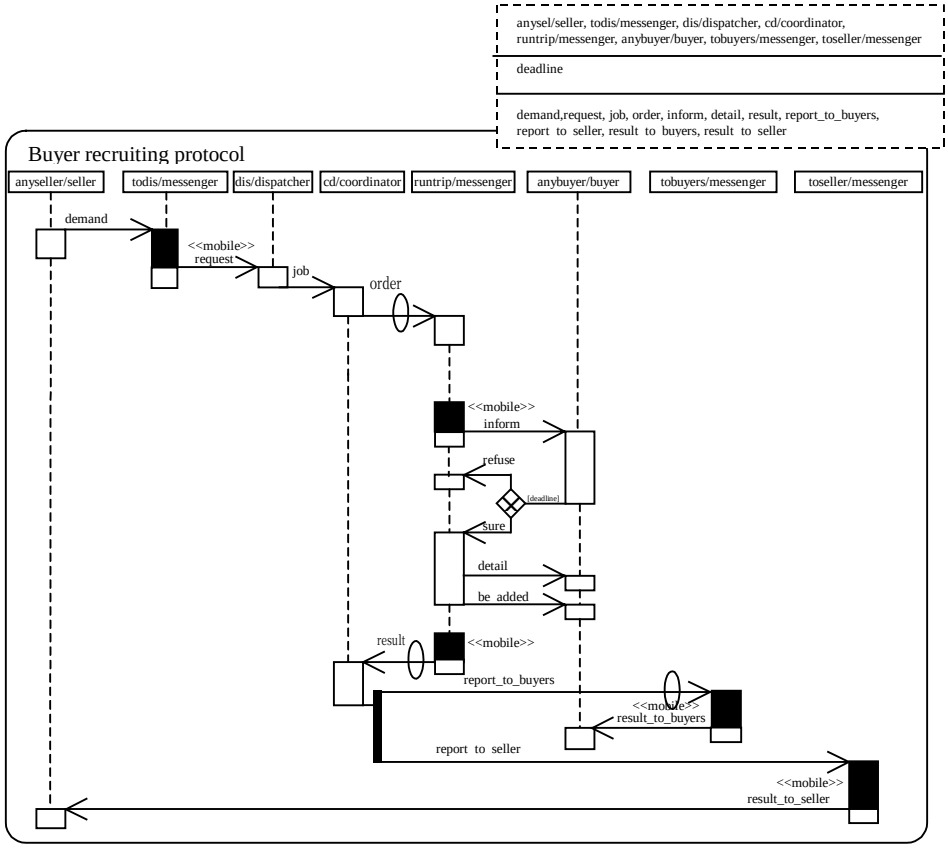

圖 4．召募買方的代理人互動樣板

4. ：代表一個在起始位置的代理人，與數個不同在結尾位置的代理人，同時會有互動行為。

5. ：代表從數個可能的互動行為中，選擇其中之一來進行。

6. ：這是我們新增的符號，代表一個在起始位置的代理人，與數個相同在結尾位置的代理人，同時會有互動行為。

7. ：這是我們新增的符號，代表數個在起始位置的代理人，與一個結尾位置的代理人，同時有互動行為。

8. ：這是我們新增的符號，代表一個在起始位置的代理人，會循序與數個相同在結尾位置的代理人有互動行為。

9. ：這是我們新增的符號，代表一個在結尾位置的代理人，會循序與數個相同在起始位置的代理人有互動行為。

10. ：一個虛線方框包含數種需要實例化的參數類型，而此樣板有三種：(1)代理人名稱：如 anyseller/seller；(2)期限：deadline；(3)互動行為：如 demand。

圖 4 中，代理人 todis/mesenger 相當於圖 3 中的代理人 messenger1，是由 anyseller/seller 所發出的；而代理人 runtrip/messenger、tobuyer/messenger、toseller/messenger 則相當於圖 3 中的代理人 messenger2，是由 cd/coordinator 所發出的。接下來，依據圖 4 中的互動行為，簡單的描述其間之互動：

代理人 anyseller/seller 向代理人 todis/messenger 發出命令後，todis/messenger 遷移到 mediator 端向 dis/dispatcher 提出需求。而後 dis/dispatcher 指派一位 cd/coordinator

來負責此活動；而 cd/coordinator 會向所有上網的 anybuyer/buyer 循序發送 runtrip/messenger 以推銷商品。當 runtrip/messenger 遷移到 customer 端後，其將通知 anybuyer/buyer 活動訊息，並詢問其意願；而 anybuyer/buyer 須於時間[deadline]前表示其意願，refuse 表示拒絕而 sure 表示答應；而若答應，runtrip/messenger 則提供較詳細之訊息並告其已被加入。時間截止後，cd/coordinator 循序與所有遷移回來的 runtrip/messenger 互動，以獲得結果，而後循序地發送 tobuyers/messenger 及 toseller/messenger 向加入的 anybuyer/buyer 及 anyseller/seller 回報結果。

(五) 各代理人的行為模式

接下來，對於召募買方此一任務，開始從低階的角度，使用 UML 的狀態圖分別描述每個角色的行為模式，及其中行為的狀態轉換。因篇幅的關係只列出其中幾個：

一、 代理人 messenger (代理人 seller 發出)

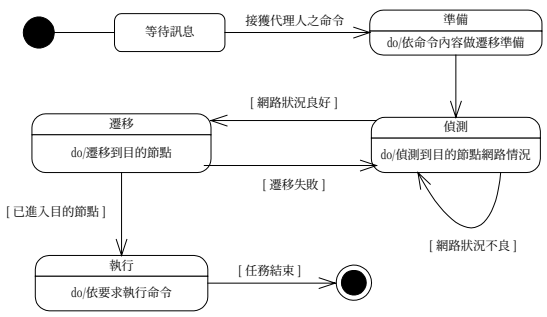

圖 5. 代理人 seller 發出之代理人 messenger 的行為模式狀態圖

A. 開始狀態：代理人 messenger 已被產生，並轉換至等待訊息狀態。
B. 等待訊息狀態：如發生“接獲代理人之命令”之事件，則轉換至準備狀態。而此命令在這裡指的是向 dispatcher 要求執行買方召募及提供產品資訊。
C. 準備狀態：代理人 messenger 記錄代理人 seller 所給予之命令並準備進行遷移，完成之後將轉換至偵測狀態。
D. 偵測狀態：代理人 messenger 將偵測目的節點(customer 節點)之網路情形。
   1. 若是網路環境不穩定或收訊不良的話，則將再自身轉換到偵測目的地節點網路情況狀態。
   2. 若是網路狀況良好，代理人 messenger 將會轉換至遷移狀態。
E. 遷移狀態：代理人 messenger 進行遷移，此時，若因其它原因導致代理人 messenger 遷移失敗的話，則將再轉換至偵測狀態；而若代理人成功的進入到目的地節點，則轉換至執行狀態。
F. 執行狀態：進入到目的地節點後，代理人 messenger 便執行所負予之命令，而結束後，便轉換至結束狀態。
G. 結束狀態：代理人 messenger 結束其生命。

二、 代理人 coordinator

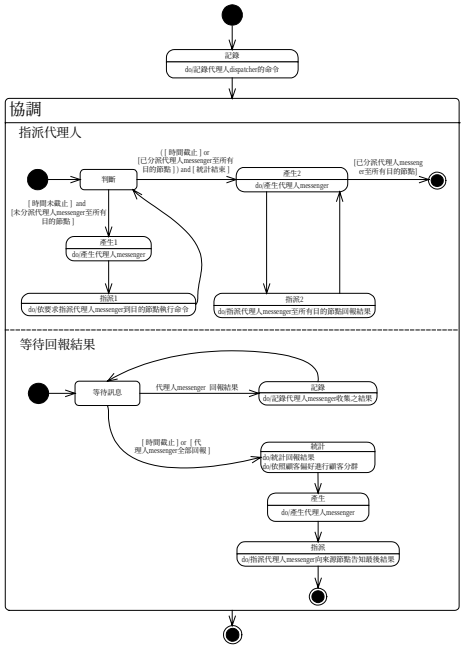

圖 6. 代理人 coordinator 行為模式狀態圖

A. 開始狀態：代理人 coordinator 開始執行並且轉換至記錄狀態。
B. 記錄狀態：coordinator 記錄代理人 dispatcher 的命令並轉換至協調狀態。
C. 協調狀態：此狀態為一同時組合狀態 (concurrent composite state)，它包含兩

個循序子狀態(sequential substates)：(1)指派代理人；(2)等待回報結果，而這兩個子狀態將會同時執行，直到都轉換至結束狀態，協調狀態才轉換至下一個狀態。這兩個循序子狀態在此不詳細說明。

D. 結束狀態：代理人 coordinator 完成任務並結束其生命。

以上為塑模行動代理人系統的五個步驟，接下來是驗證此模型的正確性。

四、 系統模型驗證的輔助方法

系統分析與設計是軟體開發過程中，非常具有關鍵性的階段，在此階段發生的錯誤經常會延續至其他階段，並衍生出更多的錯誤，因此，如何在此階段有效地發現錯誤並修正，將對整個開發的時程、成本與系統品質有相當大的影響。

因為此階段所產生的大部分為設計文件，所以一般的作法都是採取人工的方式去發現錯誤，如 review 和 inspection [4]，但是對於隱藏於複雜設計中的錯誤，還是難以發現，因此，本文提出採用以“狀態為基礎的模擬”(state-based simulation)[9][10]方式所建構的軟體工具 JMICE [10]，來輔助發現錯誤。

採取此模擬的方法分為三個步驟：

1. 首先，利用我們已經建構好用來描述每一個代理人的行為模式的狀態圖，在 JMICE 提供的 GUI 介面(見圖 7)中，將這些狀態圖建立成可以在JMICE 中執行的一組狀態機(state machines)。
2. 然後，進一步修改這些狀態機，並透過界面去驅動一狀態機，使狀態機之間產生一連鎖反應，以分別模擬之前在第三之三節建立的活動流程，及第三之四節建立的互動流程。
3. 最後，依照狀態機在模擬過程中所產生的結果，分別與這兩個流程的內容作對照，如此便可以輕易地檢驗出模型是否有錯誤和前後矛盾的地方。

五、 結論

在電子商務中，聯合議價是買賣雙方完成交易的一種方式，而聯合議價的精神便是在於凝聚強大的買方力量來取得較低的商品價格。因此，在聯合議價中，買方如何凝聚便成為一個值得討探的問題，而本篇文章以此為例，提出一個塑模方法來塑造一個以行動代理人為基礎的買方召募架構，而此架構將可主動的幫助賣方進行買方召募之工作。

在本篇文章中，我們主要的貢獻是：

1. 提出了一個包含五個步驟、由上而下(top-down)的方法，有系統地利用了 UML 的類別圖、部署圖、活動圖、狀態圖及 AUML 的 protocol diagrams 設計出此一“行動代理人為基礎的買方召募”系統模型。
2. 根據我們設計此架構的需求，加入新的符號於 AUML 的 protocol diagrams，擴增了 protocol diagrams 功能。
3. 提出以“狀態為基礎的模擬”方式所建構的軟體工具 JMICE，來輔助驗證系統模型，此方法對整個系統的開發時程、成本與品質將有相當大的影響。

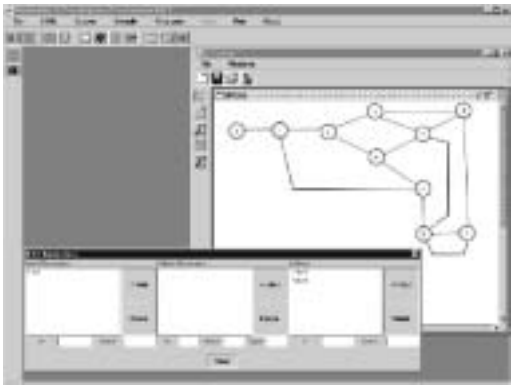

圖 7. 利用軟體工具 JMICE 建構狀態機

六、誌謝

感謝國科會計畫 NSC 91-2213-E-218-004 及 NSC 92-2213-E-218-021 補助。

七、參考文獻

[1] 王宗一，葉丁源，“行動代理者之位置追蹤機制”，*成大工程科學系碩士論文*（2001）。

[2] 林加賢，陳炳文，“利用 UML 塑模以代理人為基礎之聯合議價買方聚合機制”，*第四屆電子化企業經營管理理論暨實務研討會* (2003)。

[3] 李清發，許耿豪，“網路購物議價模式分析與系統實作”，*文化大學資管研究所碩士論文*（2001）。

[4] 吳仁和，林信惠，“系統分析與設計 理論與實務應用”，*智勝文化*（2002）。

[5] 黃建嵐，陳炳文，“群體採購模式應用於電子商務之探討與實作”，*第四屆電子化企業經營管理理論暨實務研討會*(2003)。

[6] 賴香菊，莊隆泰，“群體採購中間商系統之研究”，*中山大學資管研究所碩士論文*（2000）。

[7] Bernhard Bauer, Jörg P. Müller and James Odell, “An extension of UML by protocols for multiagent interaction”, *In Proceedings of Int’l Conf. on MultiAgent Systems (ICMAS'00), Boston*, July, 207-214, 2000.

[8] Grady Booch, James Rumbaugh and Ivar Jacobson, “The Unified Modeling Language User Guide”, *ADDISON-WESLEY,* 2000.

[9] Ping-Wen Chen, Yasuro Kawata, Hossam I. Gharib, and S. K. Chang, “An Approach for the Design and Simulation of Information Retrieval Protocols”, In *Proceedings of ICDCS Int’l Workshop on Distributed System Validation and Verification (DSVV 2K),* pp. E126--E133, Taipei, Taiwan, 2000.

[10] Ping-Wen Chen, “A Coordination Model for the Systematic Construction of Distributed Applications”, *Ph.D. Dissertation, Dept. of CS, Univ. of Pittsburgh, PA, USA*, 2001.

[11] Gamma, Erich, et al, “Design patterns”, *Addison-Wesley,* 1995.

[12] FIPA's Technical Committee C (TC C), “Extending UML for the Specification of Agent Interaction Protocols”, *FIPA response to the OMG Analysis and Design Task Force UML RTF 2.0 Request for Information*, 1999.

[13] Marc-Philippe Huget, “A Language for Exchanging Agent UML Protocol Diagrams”, *Technical Report ULCS-02-009, Dept. of Computer Science, Univ. of Liverpool*, 2002.

[14] Jean_Luc Koning and et al, “Extended Modeling Languages for Interaction Protocol Design”, *AOSE 2001*, LNCS 2222, pp. 68-83, 2002.

[15] Danny B. Lange and Mitsuru Oshima. “Seven Good Reasons for Mobile Agents.” *Comm. of the ACM*, 42(3):88--89, 1999.

[16] Jürgen Lind, “Specifying Agent Interaction Protocols with Standard UML”, *AOSE 2001*, LNCS2222, 136-142, 2002.

[17] Mauw, Reiners, and Willemse. “Message Sequence Charts in the Software Engineering Process”, In *Handbook of Software Engineering and Knowledge Engineering*, S. K. Chang editor, World Scientic, 2001.

[18] Wooldridge, Michael and Paolo Ciancarini, “Agent-Oriented Software Engineering”, *Handbook of Software Engineering and Knowledge Engineering*, Vol. 1, 2001.

[19] Object Management Group, “Agent Technology Green Paper”, *Agent Working Group OMG Document*, Aug. 1, 2000.

[20] James Odell and et al, “Extending UML for Agents”, In *Proc. of the Agent-Oriented Information System Workshop at the 17 National Conf. on AI* , Austin, USA , 2000.

[21] The FIPA Agent UML Web Site, http://www.auml.org/.