

# 在行動代理者系統進行資訊共享及訊息傳遞之研究

## Information Sharing and Message Delivery in Mobile Agent Systems

盧天麒 (Tainchi Lu)\* 傅明淇 (MingChi Fu) 馮建彰 (Chienchang Feng)

國立嘉義大學資訊工程學系

\*E-mail: [tclu@mail.ncyu.edu.tw](mailto:tclu@mail.ncyu.edu.tw)

### 摘要

隨著資訊科技的發展，以及網際網路應用的普及化，網路和生活將更緊密的做一結合，此外也由於行動代理者的技術也日益精進，在兩者的配合之下，期望能夠為使用者提供更便利的生活。

本文主要藉由行動代理者的技術，開發出自主式的行動代理者在非集中式的模擬環境下，完成使用所希望達到的目標。並且將此一研究應用非集中式的模擬環境中，配合 IEEE 標準 1516 之高階模擬架構 (High Level Architecture) 和 IBM Aglets 行動代理者環境，提出以行動代理者為基礎的資料分發管理機制 (mobile agent-based data distribution management)。

首先本研究在伺服器端部署常駐代理者 (stationary agent) 來擔任伺服器 and 客戶端雙方的橋接仲介者，並使用環形網路技術建立一個邏輯性結構來進行常駐代理者的物件訊息交換和傳遞；當模擬執行時，模擬成員的物件索引表可透過鄰接路由表的記錄，提供客戶端來選擇合適的連線伺服器。

常駐代理者可根據客戶端有興趣訂閱的模擬物件於索引表格中進行關聯性物件比對，藉由代理者池 (agent pool) 發佈出的領航代理者 (navigating agent) 將客戶端導引配置於合適的模擬聯盟伺服器裡進行參演，而代理者技術的整合則能使模擬系統具有智能來提供輔助決策。

在實驗結果中，顯示提昇自主式行動代理者的資料蒐集及訊息傳遞能力，以更具有彈性的方式，來調整模擬環境之訊息交付量，以達到整體系

統效能的進步。

**關鍵字：**高階模擬架構、資料分發管理、自主式行動代理者、資訊蒐集、分散式系統

### Abstract

In this paper we incorporate IEEE standard 1516 High Level Architecture (HLA) with IBM Aglets mobile agent system to develop a decentralized simulation environment with autonomous mobile agents. First, we devise stationary agents to serve as a communication bridge between servers and clients. Furthermore, we employ a ring construction to build a logical structure to facilitate the object sharing and transactions among the stationary agents. The stationary agent compares the clients' most interested objects with each federate's local objects for creating an object index table. As a result, a navigating agent created from the agent pool can conduct the clients to transfer to an appropriate federate server by looking up in the object index table. Besides, once the clients change their mind to retrieve other specific objects, the navigating agent follows the itinerary that was given by the stationary agent to travel in order to provide the clients with a personalized information retrieval service. The experimental results show that the autonomous mobile agents can increase adaptability

and applicability to most large-scale HLA simulations.

**Keywords:** High Level Architecture, Data Distribution Management, Autonomous Mobile Agent, Distributed System, Information Retrieval

## 一、前言

本論文所架構的非集中式網路環境為一個結合分散式有線網路與可適性無線連結的網路環境，在異質連網中搭配採用 IEEE 標準 1516 的高階模擬架構 (High Level Architecture) 來建立模擬聯盟環境[8][9]。其主要目的為提供現行各種數位裝置，諸如桌上型電腦、筆記型電腦、平板電腦、行動電話及個人數位助理等，能夠透過分散式模擬聯盟環境，達到區域內網路多人互動、訊息共享傳遞、資料一致性維護以及個人化資料蒐集等功能。

為了使異質數位裝置連線至高階模擬架構所建立的模擬聯盟環境，本研究將原本的模擬聯盟 (federation) 內的模擬成員個體 (federate)，提升為同時可供客戶端連線的模擬聯盟伺服器角色 (federate server, FS)，同時並於模擬聯盟伺服器端設置常駐代理者 (stationary agent) 負責客戶端的連線品質掌控與狀況偵測[2]，來提供一個效能良好的底層平台架構。

當客戶端進行連線到模擬聯盟環境時，由於在模擬環境中，每一個參加模擬的成員，都會有自己本身的模擬物件，因此我們會根據客戶端有訂閱興趣的模擬物件，在模擬中進行比對，比對出適宜連線的模擬聯盟伺服器。此種管理策略以期能將客戶端之通訊負載儘可能地留在本地伺服器端 (intra-communication)，來降低外部的通訊負載 (inter-communication) [1] 同時提供可適性資料蒐集服務，符合客戶端要求。

並考量以無線網路進行連線的數位裝置，先天的限制並無法使用如有線網路般大傳輸量的頻寬，為客戶端在透過無線數位裝置連線到模擬聯盟

伺服器後，兼顧無線裝置可適性之優點與個人化的資訊蒐集。

## 二、相關研究

### 2.1 行動代理者

在代理者系統裡，代理者可以因為不同的需求，而發展出不同的機制。其中，若是代理者具有在網路環境中傳送自己的能力，稱之為行動代理者 (mobile agents)[5] [19]。

行動代理者可以憑藉著移動的能力，而將程式或者狀態帶到橫跨網路上的遠端電腦執行，執行完畢後，再將結果送回原發送端，如圖 1 所示。在這段執行的期間，電腦主機並不會因為代理者的執行而閒置，不論是發送端或者接收端的電腦，都可以繼續執行自己的工作，只有當代理者需要遷徙的時候，才需倚靠電腦執行動作。

而在目前已發表多種代理者系統中，本系統採用的是 Aglets[7] [16]。

在整個網際網路的環境中，無法計數的電腦和數位設備存在於整個網路中，而對於代理者而言，為了執行並完成所肩負的任務，必須能在不同的網路中穿梭，在每個所行經的節點，搜尋所需的資訊和服務[17]。因此根據所指派的工作性質不同，主要可將代理者分為兩類，分別是工作代理者和領航員代理者。

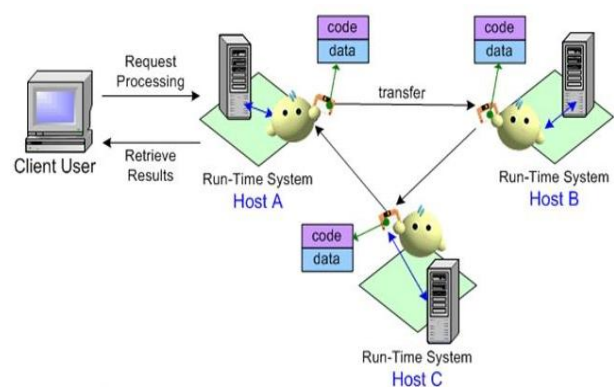


圖 1 行動代理者示意圖

在這種兩代理者中，工作代理者的角色可視為一個觀光客，在數個不同的地方旅行。而領航員代理者則相當為為觀光客的嚮導。指引觀光客往來於各地。

## 2.2 點對點系統

在點對點架構中 [14][20]，它被設計用來分享電腦資源，諸如訊息傳遞、遠端儲存體合併以及中央處理器週期分享等，此種非集中式結構經過直接的資源交換，來達成目的。

點對點的網路模型是一種分散式的網路架構 [15]，和傳統的集中式 client/server 的模型並不相同，也擁有自身的運作方式。

在點對點的系統中，其最大的特點，就是每個節點既是 client 也是 server，節點和節點之間的關係，採取的方式是直接將資源共享[4]，每一個單一節點，既可要求別人的服務，也可將自己的服務提供給其他節點。

節點和節點的聯繫，可以直接做資源的交換，或直接將服務提供給別的節點，而不必經由中央伺服器，如圖 2 所示。

目前在點對點的網路架構下，已經有了各種不同網路模型[18]，各種模型也存在優點，但仍然有缺陷需要改進，表一為各種架構的說明。

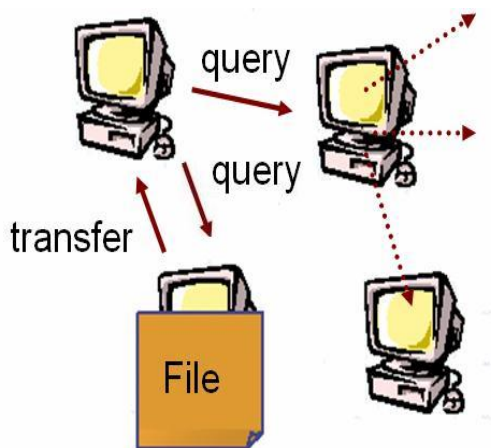


圖 2 點對點架構示意圖

表 1 點對點系統分類

Peer-to-Peer Systems	
System	Brief Description
Napster	Distributed file sharing—hybrid decentralize.
Kazaa	Distributed file sharing—partially centralized.
Gnutella	Distributed file sharing—purely decentralized.

## 2.3 高階架構

高階架構 (HLA) 為複雜系統建置與模擬提供了公共的技術，它由模擬聯盟／模擬成員、物件模型樣板、接口規範、執行時基礎 RTI (run time infrastructure) 等組成。

高階架構 (HLA) 具有部分模擬系統所缺乏的靈活性以及可擴充性問題，如此可以減少網路多餘的訊息傳輸，並且可將多種時間管理機制的即時模擬整合到一個綜合的模擬環境中。

高階架構(HLA)採用了一種“過濾”機制，換言之，即物件的生產者可以向 RTI 宣告自己所能生產的物件，物件的消費者可以向 RTI 訂購自己所需要的物件[10] [11]，最後由 RTI 負責生產者和消費者物件的物件比對。這種匹配比對可以在對象類或互動類層上進行，也可以在屬性實例或互動實例層上進行。

宣告管理機制 (DM, Declaration Management) 為模擬成員提供了類層次上的宣告與訂購機制，如圖 3 所示。當使用宣告管理時，模擬成員透過宣告管理服務向 RTI 表明自己的意願 (生產的意願或消費的意願)，RTI 負責模擬成員之間的物件匹配，並將結果傳給正確的模擬成員。

## 2.4 黑板架構

在黑板架構中[12][21]，可將各個代理者所取得的資訊，記錄在黑板上，提供一個資訊交換的機制和資訊交流的平台，而各代理者程式也可透過黑板上所記錄的訊息，找到所需的資訊。

在未使用黑板架構之前，每一個代理者對於所負擔的任務，或需要找的資訊，必須由代理者，到各個處去尋找 [6]，而如果搜尋到相關的訊息之後，便將資料送回。如果沒有搜尋到，代理者便遷徙到另一台主機或伺服器上，重覆上述的方式。

除此之外，可能發生的情況是，不同使用者卻產生相同的查詢時，結果並不能直接做訊息的交換，在樣這的運作方式之下，將會造成系統資源的浪費，並且造成網路流量的負荷。

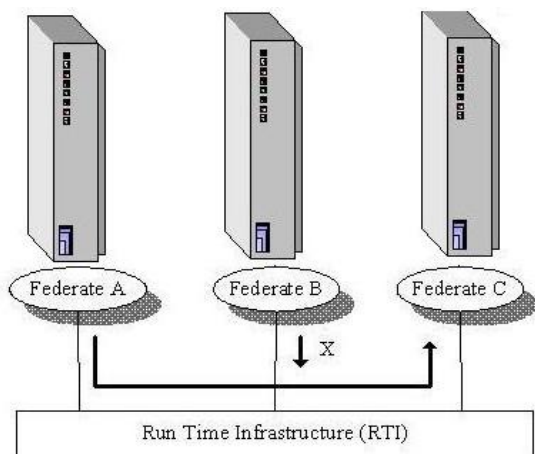


圖 3 宣告管理機制示意圖

針對前面提到的問題，黑板架構則具有一個特點[22]，就是可以共同合作，將問題做處理，並尋求解決方法，當應用於行動代理者系統後 [3][13]，讓系統中的代理者，可以彼此互相取得對方的資訊，增加效率。

黑板架構的主要起源，主要來自於專家系統。在這樣的系統中，目的是為了解決對於知識的表達、交換和分享。

在黑板架構中，對於加入系統中的成員，可

以將其視為一個對知識甚為了解的一位學者專家，而當成員在一起時，則可視為一群學者專家共同交換意見，彼此做意見的交流。

針對黑板架構的特點，共同合作將問題做處理，並尋求解決方法，將此方法與行動代理者系統做結合，將行動代理者視為一群學者專家，使其把所搜集的資料和訊息，藉由黑板架構的方式，讓行動代理者做訊息的溝通，使行動代理者用更有效率的方式，做資訊的交換和共享，完成所肩負的任務。

黑板架構的運作模式，就如同一群學者專家集合在一起，共同運用黑板架構，一起作意見的交換，心得的共享。在面對需要解決的問題時，每一個專家也都會有自己的想法和見地，經過這樣的一個環境和共同運作方式，系統中的學者專家們，就可以根據本身的經驗和知識，提出對問題的思考想法，然後提出自己對問題的解答，進一步再經由和不同的專家學者，彼此共享意見互相討論問題，尋求合理的解釋之後，藉由黑板這樣的共同的一個方式，以幫助其他專家學者，在遇到相同問題時，可以很方便的去閱讀黑板上的解答，在最有效的方法下，了解到不同的專家學者，對於問題的推理和答案。

黑板架構主要是由三個部分所組成，其中包括了黑板 (blackboard)、控制 (control)和知識源 (knowledge source)。這三個組成元素共同組成了黑板結構，如圖 4 所示。在黑板架構中，彼此都具有關聯性。以下將針對各部分做說明。

知識源 (knowledge source) 在針對某些特定問題，會把在解決問題的過程中，所必須牽涉的知識，包含進來，並根據不同的知識領域分門別類，成為每一個獨立的部分，而將這些部分稱為知識源。知識源可以視為一種規則，當發生符合規則的事時，便提供解決問題的方法。

黑板 (blackboard): 將要解決問題的資料放黑板上，也就是放在資料庫中，而對於黑板上的資料，則必須持續不斷的和由知識源所輸出的資料做一個處理。在這樣的方式運作之下，其目的可以和具有關聯的部分做連結。



控制 (Control)：主要的目的在於監控變化，對於黑板區的情況，做一個完整的監控。對於知識來源產生異動，則可以新資料送到黑板上。

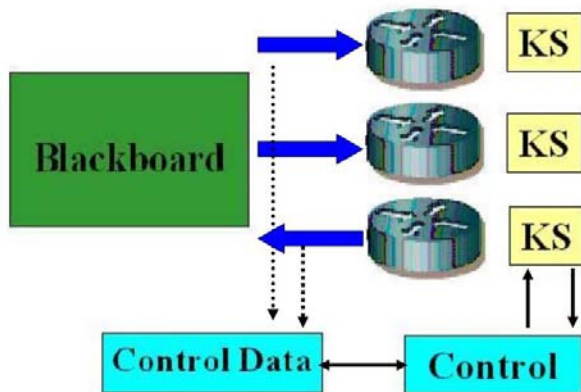


圖 4 黑板架構圖

### 三、行動代理者系統

#### 3.1 混合型系統架構

在研究方法中的模擬架構中，我們採用的是非集中式的模擬系統，其中平台的部分是由 IEEE 標準 1516 高階模擬架構，當作整個系統的平台。

在模擬環境中的其中一個組成要件，是由模擬中的聯盟伺服器 (Federate Server, FSs)，構成其中的組成要素。其主要的工作包括了將模擬環境內的所有訊息，做彼此的交換和傳送，除此之外，在模擬環境中的每一個客戶端，當要進行和聯盟伺服器做連結動作時，其中的連接過程處理步驟和必需的所有程序，都經由聯盟伺服器來負責。

在模擬環境中，每一個參與模擬的物件，在事前都必須詳加定義，因此在能夠預期的情況下，在聯盟中的物件訊息，都可由參加模擬的行動代理者，取得並把資料傳達給客戶端，經由這樣的方式，得以完成個人化資訊提供和服務

首先初始狀態可由系統或使用者自訂，供行動代理者搜尋標的物 (target)，在執行期間可使用輪詢法 (round robin) 決定下一個標的物以利代理者閒置時搜尋，回傳時包含主要搜尋結果和次要目

標的搜尋結果，供使用者選定。亦即是行動代理者如同 SNG (Satellite News Gathering) 採訪車一樣會預先到已設定的相關範圍作訊息採集，一旦使用者感興趣則自動選定為主要目標物，並由輪詢的結果決定下一個次目標物，達成主動式資訊收集之目的。代理者運行過程中，一切的訊息傳輸使用 push, pop, 和 predict；使用者將命令參數由 push 交付給代理者，由行動代理者作訊息蒐集比對並將所得結果由 pop 回傳給使用者，在蒐集期間 predict 下一個使用者感興趣的標的物資料。

在圖 5 中，在整個模擬環境中，採用的是非集中式的一個模擬型態，全部模擬的進行，是建立在聯盟伺服器上，將聯盟伺服器當作金融模擬的主要平台，以利整個金融交易的運作。

在模擬環境中，資訊的收集和監控，是由聯盟伺服器中所設立的常駐型代理者負責，而客戶端則由網路的連結傳輸，與模擬環境中的伺服器主機進行連線，連線的方式，可使用有線網路的傳輸方式，除了有線網路之外，無線傳輸的方式，也讓客戶端自由選擇。

經由以上的連接方式，完成和模擬環境的溝通程序，在模擬運作時，便可經由上述步驟，取得模擬進行時的相關資訊，以做為訊息的交換。

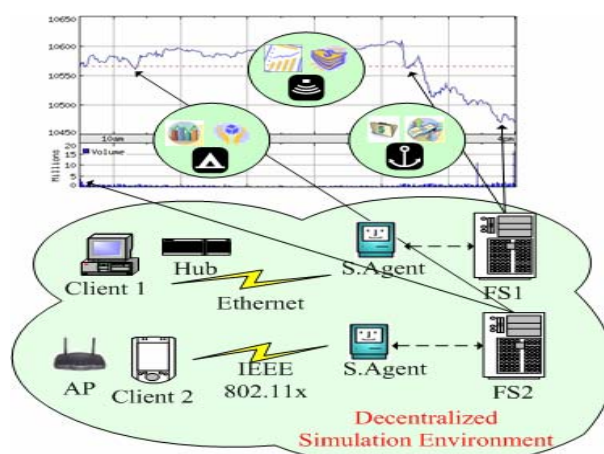


圖 5 非集中式混合型模擬系統架構

在系統中，由常駐代理者 (Stationary Agent, S.Agent) 擔任管理網路區段中盟伺服器的工作，而代理者彼此之間的連繫，則採用點對點的非集中式

架構，來進行連接，在建立了點對點的連線之後，便可將服務提供給客戶端。而在客戶端的環境中，則可利用有線網路的方式或無線網路的方式，與模擬環境連線。

### 3.2 行動代理者

自主式行動代理者 (Autonomous Mobile Agent) 為一個任務取向代理者，它必須能經由預設狀態 (系統自訂或使用者自訂) 主動地行使它擁有的動作，並依據使用者喜好或給定之規則提供個人化的資料蒐集。行動代理者能收集取回遍佈於各節點的資料，因此決定資料流向的代理者計劃 (mobile agent plan) 成為完成蒐集工作的重要項目。使用自主式行動代理者應當有如下保證，分別有：有效的資源分享率和妥善的網路負載管理，給予明智的瀏覽路徑，以及提供適當層級的安全可信賴管理。

此外在指派代理者移動前，決定有效的遷移路徑將左右整個蒐集的效能比。遷徙時機的選擇如下：

考慮一：外部訊息傳輸總量大於內部訊息傳輸總量時

考慮二：以及欲至新節點提供服務。

### 3.3 代理者分類

我們將代理者依據功能性的不同加以區分，用以提供客戶端個人化的資訊蒐集服務，其中客戶端可為行動設備或是固定裝置。除了代理者集散池之外，其它的代理者依循預先定義的工作負載給予不同的能力。如圖 6 所示，代理者架構包含四個元件，分別為擁有所有功能函數的代理者集散池，於模擬環境中擷取物件格式的常駐代理者，可參照旅行指南進行遷移的領航代理者，以及監控代理者情況的狀態擷取等。

Agent Pool：

代理者集散池，包含代理者所需的基本功能函數，並允許管理者建立 (create) 新的代理者；使用 dialog 指令選定代理者；使用 dispose 來中止執行中的代理者；或使用 clone 建立一個執行中的代理者副本指派給系統。代理者集散池為一個固定容器，藉由 Java 的 serialization 機制可存放閒置狀態的領航代理者，讓它等待下一個工作交付。當使用者參與模擬聯盟時，會由代理者集散池指派相符合的行動代理者，依據可動態修改的 retrieval rule 進行資料蒐集，提供使用者個人化的服務。

Stationary Agent：

常駐代理者，功能近似為一個 proxy，在模擬參演之前，預先常駐於各 Federation Server，統合各類物件的發佈訊息供領航代理者做資訊蒐集。並處理遷徙問題，當代理者遷徙到另一個節點之前，常駐代理者藉由 dispatch 指令指派欲遷移之代理者；反之，使用 retract 來撤銷對代理者的指派。另外，代理者的所在區域必須提供給常駐代理者記錄之，如此一來才可以發送訊息將之召回。對於代理者的執行狀態，可藉由 deactivate 指令來強制停止被選定的代理者，或藉由 active 指令啟動閒置睡眠狀態的代理者。除此之外，當外部通訊總量大於內部通訊總量時，常駐代理者將提供旅行指南給領航代理者進行遷徙。

Navigating Agent：

領航代理者，做資訊蒐集以處理客戶端交付之工作。會自動連接至模擬聯盟中最接近的常駐代理者，做訊息比對，並將結果回傳給客戶端。當遷徙條件成立，則依據常駐代理者提供的旅行指南，進行遷移，並適時的修改 retrieval rule。為了避免無上限進行遷移增加系統成本，根據使用者喜愛所影響的回報結果命中率 (hit rate) 來決定之，一旦低於額定臨界值，則由下一任代理者接替之。

State Capture：

代理者狀態擷取，用來監控領航代理者的狀態並記錄之，以決定領航代理者之生命週期。換句

話說，代理者的環境變數皆記錄於此。

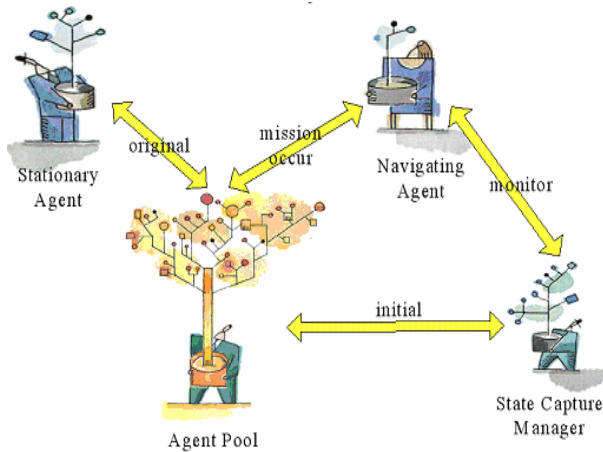


圖 6 基本代理者關係圖

### 3.4 模擬運作管理

在整個模擬系統中，常駐代理者為負責管理監控聯盟伺服器端資料傳遞以及客戶端即時連線狀態的重要角色，如圖 7 所示，其運作過程簡述如下：

Step 1. 在模擬系統環境初始化前，整個模擬環境中的聯盟伺服器向其常駐代理者進行註冊登錄的動作（包含聯盟伺服器的 IP 位址、連接埠端口等），此後常駐代理者將獲得聯盟伺服器提供的物件模型。

Step 2. 常駐代理者定期收集其負責的聯盟伺服器之資料分布情形，並與附近的代理者做訊息交換，將所得記錄於鄰接路由表 (neighbor routing table)，以作為分配聯盟伺服器給客戶端時的參考依據。

Step 3. 當客戶端欲加入模擬環境參與演練時，可依據所在地距離因子 (location-based distance factor) 先挑選最接近的常駐代理者進行連線，以取得客戶端識別碼，再由常駐代理者考量客戶端需求，與鄰近路由表比對，最後指派客戶端至合適的聯盟伺服器。

Step 4. 常駐代理者進行物件比對後，透過行動代理者將蒐集結果回傳給客戶端。依據客戶

端的訂閱率，來決定訊息價值性。

Step 5. 常駐代理者會根據 Neighbor routing table 中的資料分布參考值來為客戶端挑選出一台最適當的聯盟伺服器供客戶端進行連線，並初始化領航代理者 (Navigating Agent) 處理預先分配好的資料分送與短暫斷線 (handoff)。

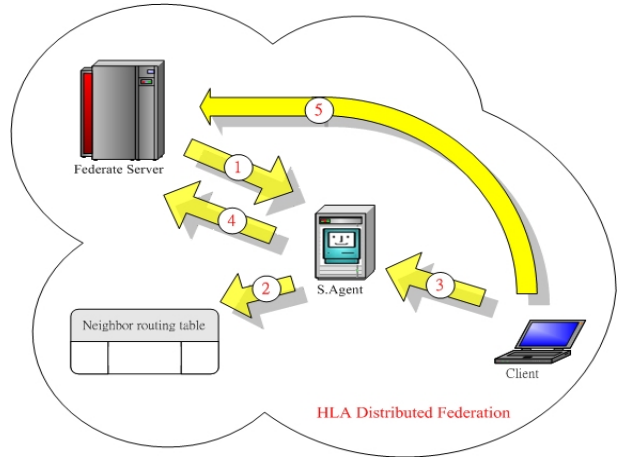


圖 7 常駐代理者運作流程

### 3.5 建立環狀連線架構

在模擬環境裡採用點對點連網，交換彼此伺服器中有的物件資料，爾後根據客戶端連線要求，分配適宜的聯盟伺服器提供連線。而在非集中的系統中，我們使用環形點對點技術來定位各個代理者與其中的聯盟伺服器。

環形點對點技術為識別空間 (identifier space) 與行動節點 (mobile node) 提供一個單一的映射，亦即是行動節點為一個有 IP 位址及埠號的主機，每個節點參照一個識別符 (identifier)。對應識別符  $i$ ，而有著比識別符  $i$  大，而又小於其它識別符者，此節點稱之為識別符  $i$  的後繼者 (successor)。當整個識別符與各節點的後繼者建立好後，行動代理者即可從中作訊息溝通。

其中，所有的節點對應到  $m$  個位元 (m-bit) 的環形識別空間，每個節點持有各自的路由表稱之

為鄰接路由表，鄰接路由表記錄環形架構中其它節點的訊息，包含有節點識別符以及專屬的網路位址。在鄰接路由表中，符合大於  $i+2k-1$  計算式，則節點  $i$  的第  $k$  個進入點 (entry) 的最小節點為節點  $s$ 。節點  $s$  也是節點  $i$  第  $k$  個可被候選的後繼者。換句話說，鄰接路由表記錄著由  $m$  個識別符區間所挑選出的  $m$  個進入點，節點  $i$  的第  $k$  個區間座落於：

$$\left[ (i+2^{k-1}) \bmod 2^m, (i+2^k \bmod 2^m) \right] \quad (1)$$

圖 8 所示，當  $m=3$  且存在三個節點分別為節點 1, 3 以及 7。那麼節點 1 緊接著的後繼者為  $(1+2^0) \bmod 2^3 = 2$  或者是  $(1+2^1) \bmod 2^3 = 3$  等兩個節點。

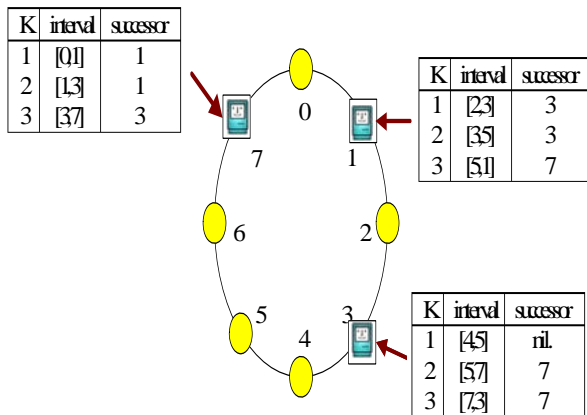


圖 8 常駐代理者建立連線流程

### 3.6 物件比對

在進行相互連結建立環型結構後，常駐於每個聯盟伺服器的常駐代理者，會記錄該模擬聯盟伺服器裡模擬成員的物件模型，並且建立物件索引表來供環型結構裡的常駐代理者進行兩兩交叉尋找，最後每個常駐代理者將擁有整個模擬聯盟進行中所有的模擬成員物件及所屬之模擬伺服器，物件比對演算法如圖 9 所示。

### 3.7 黑板共用區

在黑板架構下，黑板將提供資訊共享的功能，以利代理者做資訊交換。對於領航代理者移動到伺服器時，會把該伺服器中所能提供的資訊和服務做記錄，即使目前並無需要，也可把資訊記錄在黑板上，當其他代理者來到此節點時，可以檢視並參考其內容，是否和目前所需尋找的資訊相同或具高度關聯性，可做為搜尋的參考和指引。

```

//procedure object sharing of each node
for i=1 to (n-1) do begin
  f=0;
  for j=1 to (n-i) do
    if (List[j].successor).index <=
List[j].index) then
      begin
        add (List[j].successor).index,
List[j].index);
        f=1;
      end
    if (f=0) then exit
end

```

圖 9 物件比對演算法

在研究中當常駐代理者來到派聯盟伺服器時，便會收集各類物件的發佈訊息供領航代理者做資訊蒐集。

當領航代理者遷徙到另一個節點之前，便是依據常駐代理者提供的旅行指南，移動到遷移的目標，而此時領航代理者，已經對於所經過的節點，有所了解，對該聯盟伺服器的資訊做了蒐集，也因此，當領航代理者來到一個新節點時，便可以將部分的資訊傳遞給新節點的常駐代理者。

對於常駐代理者而言，除了本身存在的聯盟伺服器之外，每當新的領航代理者來到該節中時，便可藉由黑板，將新的訊息記錄在黑板上，如此便可以了解其他伺服器的資訊，以比較聯盟伺服器所能提供的資訊是否改變，而如果改變，便可以適時做更新的動作，而更新的動作，也是記錄在黑板上。

在黑板的架構下，除了可以將模擬聯盟中，



其他模擬伺的狀態，做到一個資訊更新的程度外，當有另外的領航代理者來到該伺服器，便可以再次做一交換資訊的動作，一方面可把領航代理者所希望取得的資訊，提供給他，並將更新過後的聯盟伺服器狀態資訊，藉由黑板的記錄，重新讓領航代理者找尋到新的目標。而領航代理者本身所具有的資訊，也會交付給常駐代理者，記錄在黑板上，而常駐代理者，又再度將資訊傳回客戶端。

在這樣的黑板架構下，可以解決兩個問題，第一個問題在於，數個客戶端欲查詢的資訊重覆時，如果先前查詢的結果，不能讓其他使用者知道，勢必數個使用都會逐一查詢，在這樣的情況下，當然會造成系統的負擔而藉由黑板架構，讓代理者間彼此做溝通，便可以把資訊共享，而避免重覆的資訊不斷流通造問題。

第二個問題在於如果沒有黑板架構，做不斷做資訊交換，當其中一個代理者發現最新資訊時，並不能讓其代理者知道，而造成資訊的落差，使用者也不能收到新的資訊，藉由黑板的方式，也對此問題有了解決。

### 3.8 黑板架構時機

在模擬架構中，物件彼此藉由互動關係，產生新的值，如當利率發生變化時，匯率走勢也會發生影響，但這些互動關係，都只會影響的物件屬性，除了物件互動之外，由於金融資訊會隨著時間而改變，故列為影響物件的要素之一。

此外由於本模擬環是建立在點對點的環境中，而在此一環境的特色，則為任一節點，可能任意加入或離開網路，因此在點對點的網路中，可能在前一時間點可提供資訊服務的節點，在下一時刻，就離開網路，因此藉由黑板架構，讓代理者可以交換資訊，彼此溝通，完成代理者合作機制。

上述的情況都是在常駐代理者所取得物件所在模擬伺服器資訊方面，仍可有效通知領航代理者，讓使用者聯接到適當的模擬伺服器。而且並沒有影響到在 OMT 中所定義的物件結構。因此都是

為使用黑板架構的時機。

採用黑板架構讓代理者做物件資訊交換得的時機：

- 1 物件交互作用所產生的內含值改變。
- 2 網路環境變動，節點加入或離開。
- 3 時間變化，對物件發生內含值改變。

### 3.9 運作原理

圖 10 圖 11 為在黑板架構下，代理者做溝通的流程圖，其運作過程簡述如下：

Step 1. 在模擬環境中，各常駐代理者已派駐在聯盟伺服器中，並記錄該聯盟伺服器中的資訊（包含聯盟伺服器的物件資訊，所能提供的服務，盟伺服器的 IP 位址、連接埠端口等）。

Step 2. 常駐代理者為領航代理者提供一個聯盟伺服器做連線。

Step 3. 當領航代理者抵達進行連線的聯盟伺服器時，常駐代理者便將該聯盟伺服器所擁有的資訊，交負予領航代理者，提供給客戶端。

Step 4. 此時常駐代理者，將查詢黑板上是否還有其他聯盟伺服器，具有符合客戶端所需的資料，如果有則提供新的聯盟伺服器進行遷徙。

Step 5. 而領航代理者在將遷徙到該聯盟伺服器之前，途中行經其他聯盟伺服器時的資訊，記錄在黑板上，提供給其他理者。

Step 6. 完成訊息交換的動作之後，常駐代理者便藉由黑板上的記錄，更新資訊，以準備下一次的服務。

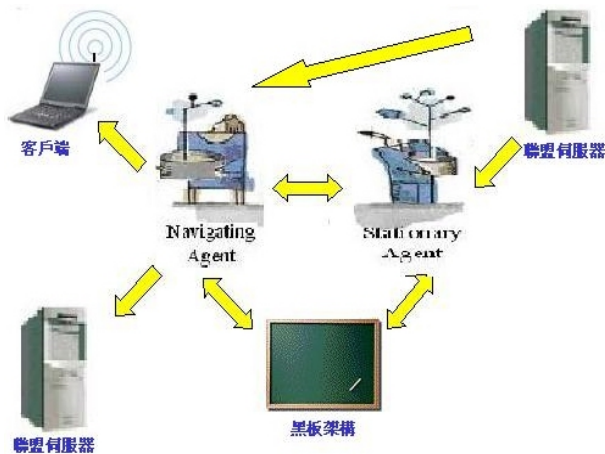


圖 10 黑板架構代理者運作圖

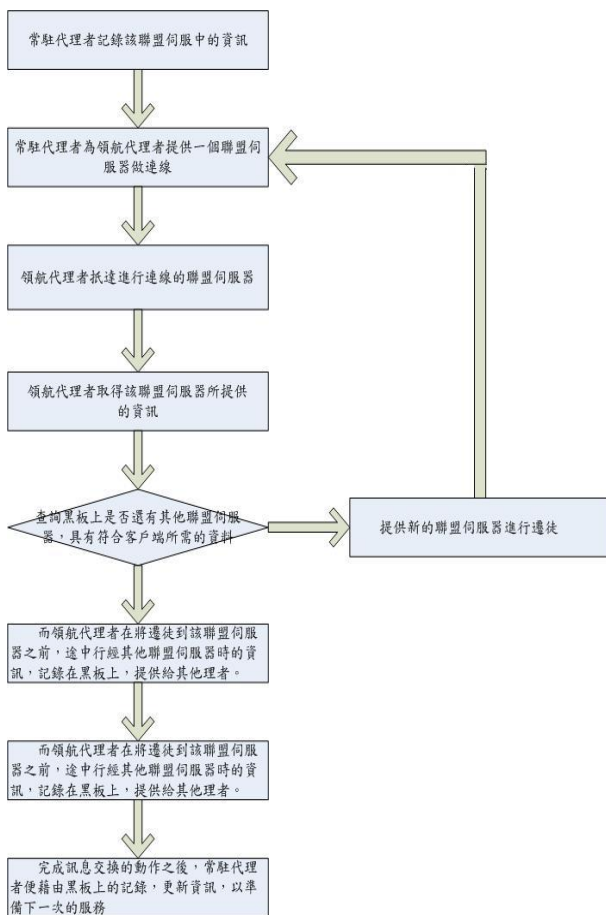


圖 11 黑板架構代理者溝通流程圖

## 四、實驗結果

### 4.1 宣告管理與資料分發管理實驗

在本實驗中，模擬聯盟伺服器採用 P4 2.4

GHz、512 MB RAM 以及 Microsoft Windows 2000 架設之環境，底層高階模擬架構為美國國防部 DMSO 版本的 HLA RTI-NG 1.3 v4，模擬聯盟伺服器所產生二十個模擬成員個體分別發佈十種不同的物件種類，由模擬成員個體隨機訂閱其它模擬成員個體之運行物件。

首先經由執行時基礎進行的發布與訂閱動作，各別利用宣告管理機制與資料分發管理機制來達到初階的訊息管理動作，以觀察模擬聯盟個體在模擬運行中執行時間的影響。金融投資匯率之物件模型識別表，如表 2 所示，其中遠期匯率互動關係如表 3 所示。

表 2 物件模型識別列表

Federate	Taiwan dollar deposit rate per annum	U.S. dollar deposit rate per annum
Bank 1	1.5%+0.5% (Bank 4+0.5%)	2.8% × 0.8 (Bank 5 × 0.8)
Bank 2	1.5% × 1.5 (Bank 4 × 1.5%)	2.8%+0.2% (Bank 5+0.2%)
Bank 3	1.5% × 0.95 (Bank4 × 0.95)	2.8% × 1.25 (Bank 5 × 1.25)
Bank 4	1.5%	Null
Bank 5	Null	2.8%

假設在高階模擬架構上進行的模擬演訓中，模擬聯盟伺服器之間所產生的模擬成員個體與其模擬物件如表 2 所示，在三台模擬聯盟伺服器之間，模擬成員會於執行時基礎進行發布之動作，而訂閱情形則依據物件互動關係設定為訂閱與否，藉此表現出宣告管理機制對資料的處理情形，其間訊息的傳送時間差異，經由實驗可得到如圖 12 之結果。由實驗數據可以觀察出，當利用宣告管理機制控制模擬成員個體所需傳送與接收的資料時，依據模擬成員個體訂閱設定之不同，減少了此模擬成員個體所需處理的資料傳輸量。

表 3 遠期匯率關係互動列表

Days	Long-term interest rate equation
30	$\frac{30-S}{S} \times \frac{12}{1} = (I_{Taiwan} - I_{USA}) \times \frac{30}{360}$
60	$\frac{60-S}{S} \times \frac{12}{2} = (I_{Taiwan} - I_{USA}) \times \frac{60}{360}$
90	$\frac{90-S}{S} \times \frac{12}{3} = (I_{Taiwan} - I_{USA}) \times \frac{90}{360}$

在資料分發管理機制上由於僅使用 Simple DDM 策略所提供的接收方資料管理，對於發送端並未有進一步的設定，因此模擬成員間的資料管理情形相近於宣告管理機制所處理資料分發的能力。綜合以上兩種結果可發現，資料傳輸量較少的模擬成員個體能夠提早完成模擬演訓，兩種資料管理皆可精簡訊息資料量，以節省執行時間。

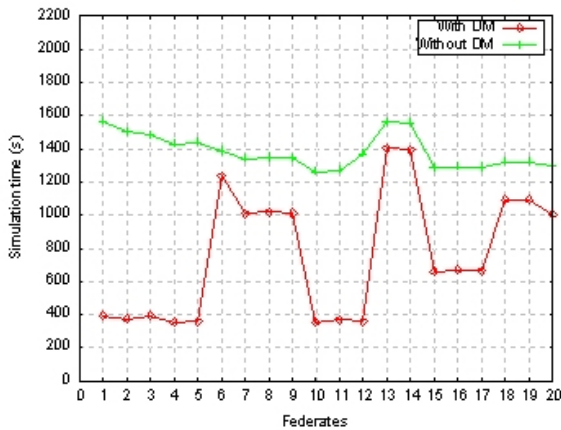


圖 12 資料分發與宣告管理實驗數據圖

#### 4.2 環形結構建製實驗

在實驗中首先利用模擬聯盟伺服器建製好的高階架構為系統底層，由模擬聯盟伺服器之常駐代理者分別紀錄所屬模擬聯盟伺服器之主機資訊，然後藉由後繼者節點的查找產生鄰接路由表來建立一個邏輯性環形結構，以完成非集中式架構下各模擬聯盟伺服器之常駐代理者初始化的訊息溝通。我們分別利用不同的網路裝置在有線環境與無線環境下，經由後繼者查找服務來比較點對點環形結構對模擬系統之影響時間。

在本實驗中，假設識別空間  $m=7$  最大共提供 128 個行動節點，我們分別比較在有線與無線裝置下，依序增加常駐代理者的數量，藉此觀察不同延展性下，邏輯性環形結構建製的時間差異。由圖 13 所示可知常駐代理者使用的行動節點，它所參照的識別符號能有效率地選定後繼者的座落區間，建立整體的環形結構。

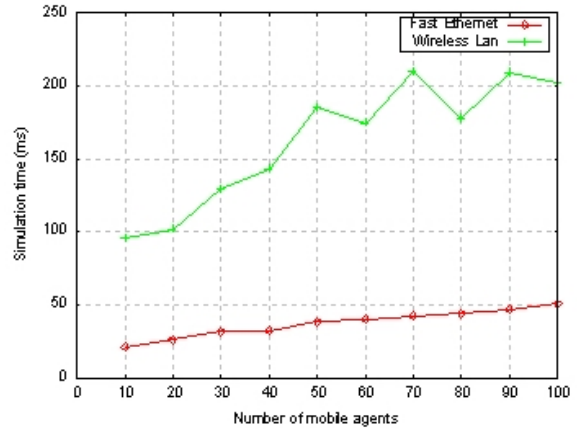


圖 13 環形結構建製實驗數據圖

當客戶端使用連線的網路環境為 100Mbps 的高速乙太網路時，我們觀察此時利用常駐代理者建立環形結構的效率高低情形。如圖 13 所示，在有線網路環境下建立環形結構，可避免因設置中央伺服器造成的負載分擔不一的過重情形，利用非集中式架構來設置的常駐代理者，其所花費在邏輯環狀結構的建立時間，會依據代理者的成長而逐步增加其執行時間，但是倚賴著有線網路環境具有的高速穩定傳輸的特性，從圖中可以看出其邏輯結構建立的時間上升的並不明顯。

當客戶端利用無線網路 (IEEE 802.11g) 進行連線時，我們觀察此時常駐代理者建立邏輯環形結構之效率高低情形。如圖 13 所示，我們加以對照使用有線網路的效率高低可以發現，針對以無線網路進行連線的常駐代理者來說，隨著行動節點數量的逐漸增加，對於執行時間有較大的影響，其原因為無線網路環境頻寬較有線網路為低，因此在硬體限制的先天差異上，對於行動節點的增長來說，常駐代理者環形結構的建置時間會受網路品質因素之影響。

### 4.3 資訊蒐集實驗

在此實驗中，分別利用傳統派遣代理者在每個節點裡循序搜尋資訊之方法，與我們提出的查找物件索引表後，藉由旅行指南派遣領航代理者做資訊蒐集之方法，討論針對客戶端所要求的資訊，如何可以提供有效的蒐集結果，並觀察資料量的多寡影響與兩種方法在模擬進行時的差異情形。

在實驗中，利用模擬聯盟伺服器進行互動時所產生之輸出資料，由模擬聯盟伺服器端之常駐代理者在初始化後建立的物件索引表格，經過與客戶端興趣比對後，視情況需要由代理者集散池建立的領航代理者，參照常駐代理者所提供之旅遊指南，將資訊蒐集的結果傳輸給客戶端。我們利用不同的蒐集方式，來分別比較傳統循序找尋方法與非集中式資訊蒐集方法，所造成的傳輸量影響與時間差異。

在進行實驗時，假設模擬環境內進行的物件互動中，依照客戶端需求來做的資訊蒐集，我們分別比較使用循序方式以及使用物件索引表後，領航代理者參照旅遊指南之訊息蒐集動作完成時間的差異。

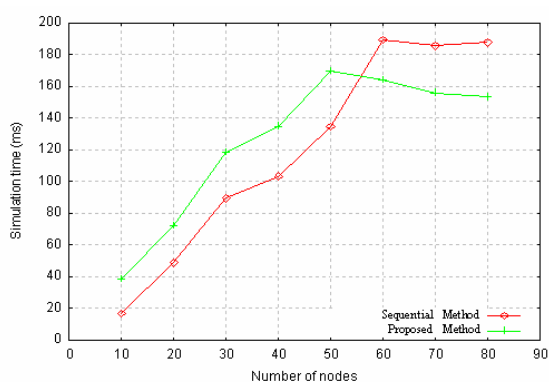


圖 14 資訊蒐集實驗數據圖

如圖 14 所示。當使用循序搜尋時，可以觀察出節點數較少的時候，代理者資訊搜尋的回報時間遠比我們提出的環型結構資訊搜尋時間少，因此可

以有效率的達成客戶端要求；但是隨著節點數的增加，也提高了循序搜尋所花費的時間，此時我們提出的環型結構資訊搜尋並不會因節點複雜的增加而大幅地花費更多的搜尋時間，反而因為預先的環型初始化後所建立之物件索引表，依據客戶端的需求而能精準的判斷正確的節點位置，並派遣代理者將資訊攜回。因此此種非集中式邏輯環型結構，適合在大型的模擬環境中，隨著節點數的增加，其資訊蒐集的效益反而優良於傳統循序搜尋的方法。

## 五、結論

本論文提出非集中式網路環境，在 IEEE 1516 高階模擬架構 (HLA) 下結合有線與無線網路，讓使用者能夠隨時隨地透過各種通訊裝置進行連網，其中並以高階模擬架構為基礎進行模擬演訓，以求達到人機互動與資訊互享之功能。本論文並發展模擬聯盟伺服器 (federate server) 供客戶端連線使用，並於模擬環境內設置常駐代理者 (stationary agent)，利用掌控客戶端訊息傳輸量情況以及其蒐集喜好策略的變化，動態地為客戶端分配適宜的模擬聯盟伺服器來進行連線。

最後，經由系統模擬實驗結果得知，以行動代理者為基礎的資料分發管理機制，透過自主式行動代理者可達到客戶端主動式資訊擷取服務。並將高階模擬架構延伸至混合型異質行動網路，並建置自主式行動代理者，來提供使用者較多元與廣泛的服務，使得整個系統更具有擴充性與實用性。

## 六、誌謝

本論文由國科會計畫編號 NSC93-2213-E-415-006 補助辦理執行。



## 七、参考文献

- [1] A. Acharya and B.R. Badrinath, "Checkpointing Distributed Applications on Mobile Computers", In Proc. of Conf. on Parallel and Distributed Information Systems, pp. 73-80, 1994.
- [2] A. Bieszczad, B. Pagurek, and T. White, "Mobile agents for network management", IEEE Commun. Surv., Vol. 1, No. 1, 1998.
- [3] Brzykcy G., Martinek, J., Meissner A., Skrzypczynski P., "Multi-agent blackboard architecture for a mobile robot", In Proc. of the 2001 IEEE/RSJ International Conf. on Intelligent Robots and Systems Vol. 4, pp.2369–2374, Oct. 29 – Nov. 3, 2001.
- [4] B. Yang and H. Garcia-Molina, "Improving Search in Peer-to-Peer Networks", In Proc. of the 22nd International Conf. on Distributed Computing Systems (ICDCS'02), IEEE CS Press, pp 5–15, 2002.
- [5] C.P. Martin and K. Ramamritham, "Recovery Guarantees in Mobile Systems", In Proc. of Conference on Workshop Data Eng. for Wireless and Mobile Access, pp.1-7, Aug. 1999.
- [6] C. T. Kiang, M.Y and E. Turban, "Distributed intelligent agents in decentralized organizations", In Proc. of the 26th Hawaii International Conference on System Sciences, Vol. 4, pp.292 – 300, Jan. 1993
- [7] D. B. Lange and M. Oshima, "Programming and Deploying Java Mobile Agents with Aglets", MA: Addison-Wesley, 1998.
- [8] Defense Modeling and Simulation Office, "HLA Rules, V. 1.3," U.S. Department of Defense, Washington D.C., Feb. 1998.
- [9] Defense Modeling and Simulation Office, "HLA Interface Specification, V. 1.3," U.S. Department of Defense, Washington D.C. Apr. 1998.
- [10] Defense Modeling and Simulation Office, "HLA Object Model Template Specification, V. 1.3," U.S. Department of Defense, Washington D.C., Feb. 1998.
- [11] Defense Modeling and Simulation Office, "RTI 1.3-Next Generation Programmer's Guide V. 6," U.S. Department of Defense, Washington D.C., Feb. 2002.
- [12] D. Deugo, " Mobile agent messaging models", In Proc. of the 5th International Symposium on Autonomous Decentralized Systems, pp.278 – 286, March 2001.
- [13] D. Jing et al., "Event-Based Blackboard Architecture for Multi-Agent Systems", In Proc. of International Conf. on Information Technology: Coding and Computing ITCC 2005, Vol.2, pp.379–384, 2005
- [14] D. R. Karger, E. Lehman, F. Leighton, M. Levine, D. Lewin, and R. Panigrahy, "Distributed caching protocols for relieving hot spots on the World Wide Web", In Proc. of the 29th Annual on ACM Symposium Theory of Computing, pp.654–663, 1997.
- [15] D.S. Milojevic et al., "Peer-to-Peer computing", tech. report HPL 2002-57, Hewlett-Packard, 2002.
- [16] IBM Japan Research Group "Aglets Workbench," web site: <http://aglets.trl.ibm.co.jp>.
- [17] P. Stavros, S. George and P. Evaggelia, "Mobile Agents for World Wide Web Distributed Database Access", IEEE Transactions on Knowledge and Data Engineering, Vol. 12, Issue 5, pp.802-820, Sep. 2000
- [18] Q. Lv et al., "Search and Replication in

Unstructured Peer-to-Peer Networks”, In Proc. of the 16th International Conf. on Supercomputing (ICS’02), ACM Press. , pp.84–95, 2002.

- [19] S. Ichiro, “Building Reusable Mobile Agents for Network Management”, IEEE Transactions on Systems, Vol. 33, No.3, Aug. 2003.
- [20] S. Ion, “Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications”, IEEE/ACM Transactions on Networking, Vol. 12, No.11 pp.17-32, Feb. 2003
- [21] S. Ram, V. Ramesh,” A blackboard-based cooperative system for schema integration”, Expert, IEEE Vol. 10, Issue 3, pp. 56-62, June 1995.