

運用自主式行動代理者於非集中式模擬環境之研究

Using Autonomous Mobile Agents in Decentralized Simulation Environment

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

摘要

本研究採用 IEEE 標準 1516 之高階模擬架構 (High Level Architecture) 和 IBM Aglets 行動代理者環境，開發了自主式行動代理者技術且應用於非集中式模擬環境中，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)，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間的物件訊息交換和傳遞。並根據客戶端有興趣訂閱的模擬物件進行關聯性物件比對，將客戶端導引配置於合適的模擬聯盟伺服器裡進行參演。本研究顯示出自主式行動代理者的資料蒐集及訊息傳遞能力，有彈性的調整模擬環境之訊息交付量，提昇大規模高階架構模式模擬的系統效能。

關鍵詞：高階架構、自主式行動代理者、資訊蒐集、點對點架構、環狀連線架構、旅行指南

1. 前言

近年來由於電腦科技的進步及網際網路的發展，大規模的模式模擬與電腦兵棋亦隨之應用發展，成為各國建軍備戰、部隊訓練與軍事作戰上重要的輔助工具。在歐美國防先進國家中，已有非常多的運用模擬與電腦兵棋成功案例，促使國防事務的重大革新。為了解決各類型模擬系統整合的問題，美國國防部已於 1995 年起推動下一代的模擬架構稱為高階模擬架構 (High Level Architecture) [13]，期能藉此共通架構將各類模擬應用，進行有效的整合應用。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 的高階模擬架構來建立模擬聯盟環境，以行動代理

者 [1,2,4,7] 為基礎，運用 IBM Aglets [5,6,10]，開發了自主式行動代理者技術且應用於非集中式模擬環境中。

當客戶端進行連線到模擬聯盟環境時，由於模擬成員個體有各自負責產生運行模擬的物件，因此我們根據客戶端有訂閱興趣的模擬物件進行伺服器端的物件相關性比對，常駐代理者會依照客戶端個人化資料需求以及模擬聯盟伺服器端之物件分配，比對出適宜連線的模擬聯盟伺服器，藉由發送領航代理者 (navigator agent) 進行客戶端配置，此種管理策略以期望能提供合適性的資料蒐集服務，以符合客戶端要求。提供個人化的資訊服務。

2. 相關研究

2.1 行動代理者

行動代理者可以依據來源電腦欲完成的特定工作，從而處理被分派到之程序，如同廣域網路分布下的異質資料庫存取。一旦被指派工作，行動代理者會獨立且自動自發處理接獲的程序。當行動代理者與伺服器聯繫後，即開始執行程序，依照伺服器主機給予行動代理者的安全性做認證授權。而為了要完成給定的程序，行動代理者能傳送至另一個伺服器主機，建立一個新的代理者，且能在彼此間，做訊息溝通，直到工作完成將結果回傳給使用者或傳給另一個伺服器主機。

如圖 1 所示，讓代理者主動在不同的子網路遊走，能在每一個所經過的節點，執行指派的工作，主要可分為兩類代理者：工作代理者與領航員代理者，其中工作代理者相當於一個觀光客，它搭乘旅遊巴士參訪不知名小鎮；而領航員代理者相當於一個旅遊巴士嚮導，它引領觀光客穿梭於小鎮中。

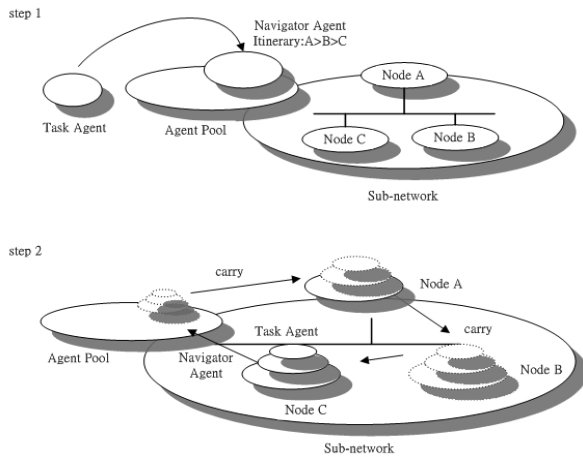


圖 1 領航員代理者與工作代理者示意圖

2.2 點對點系統

點對點分散式架構 [3,8,9,11,12]，可藉由直接存取各個電腦節點所分享的資源，而不需要透過中央式伺服器。而中央式伺服器能在某些時間被使用來做特定的工作，例如增加一個新節點至網路上，獲得全域的關鍵值以及系統間的同步等。

在純粹的非集中式架構中 (purely decentralized architecture)，網路上所有的節點皆確切的執行相同的工作，有著如同伺服器端與客戶端的動作行為，並且期間的活動沒有中央的協調者進行協調。如圖 2 所示。

實心線表示網路節點的連結情形，由一個請求搜尋的節點發送要求到它鄰近的節點，依此循環。當接收到重複訊息時，會藉由回覆一個錯誤訊息來避免訊息迴圈減低網路傳輸量。當發現欲找尋的檔案則回傳成功訊息。

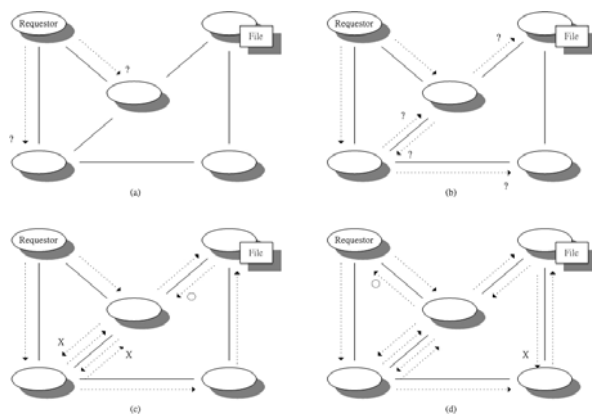


圖 2 非集中式搜尋機制示意圖

2.3 高階架構

高階架構 (HLA) 為複雜系統建置與模擬提供的一種技術，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。HLA 的顯著特點為支持各類的系統模擬，支援物件為主的模擬應用開發模式，提供通用的訊息交換通信協議以及通用的、開放的、可根據不同應用領域制定的訊息語句語意互動協議、將模擬功能與通用的支撐系統分離的結構等。因此，HLA 在解決分散式、協同合作的模擬系統與物件的可操作及可重用性方面取得了重大的進展。

3. 系統架構

3.1 混合型系統架構

在此研究中的非集中式模擬系統架構內，我們依然以 IEEE 標準 1516 高階模擬架構為系統平台，由數台聯盟伺服器 (federate server, FS) 組合成分散式網路伺服器叢集，負責模擬環境內的訊息交換以及處理各個客戶端連線至聯盟伺服器時所需要的連網動作。模擬環境中所有的物件都必須事先定義設置完成，因此在可預期下，各個物件訊息傳送可以藉由行動代理者將資料攜回，來提供個人化的行動服務。

如圖 3 所示，在此非集中式架構中，聯盟伺服器端為金融交換模擬運作平台，並於其中設置常駐代理者來監控蒐集訊息，客戶端藉由有線或無線網路傳輸環境與伺服主機連線，並取得模擬進行時之交換訊息。

在系統架構中，以圖 3 為例，藉由指派常駐代理者 (stationary agent, S. Agent) 負責管理網路區段裡的聯盟伺服器，代理者與代理者互相之間，依據點對點非集中式架構來建立連線並提供服務給客戶端。客戶端的連線可選擇使用有線網路或者無線網路環境。

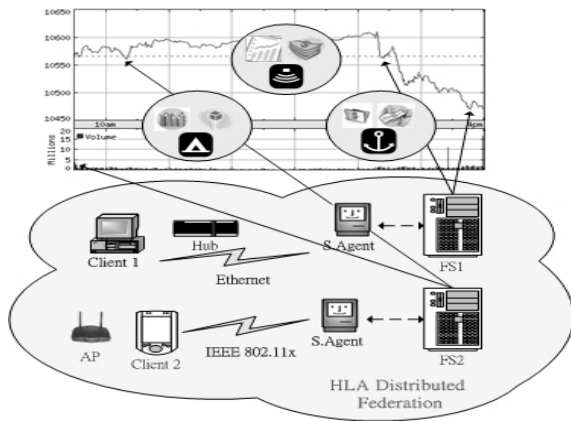


圖 3 非集中式混合型模擬系統架構

3.2 行動代理者

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。

- 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。

- 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。

- 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。

- 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。

3.3 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 4 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)。

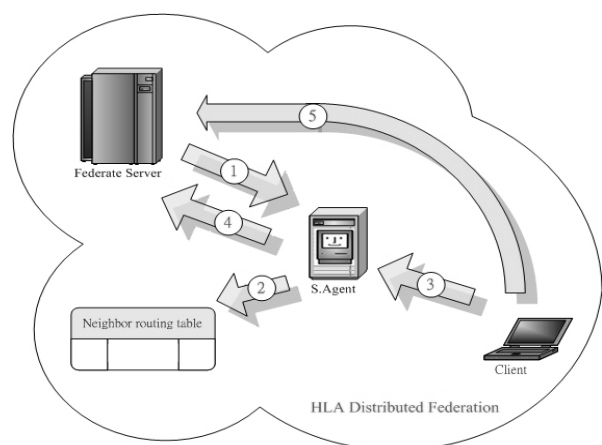


圖 4 常駐代理者運作流程

3.4 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 a，而有著比識別符 a 大，而又小於其它識別符者，此節點稱之為識別符 a 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 $r+2^k-1$ 計算式，則節點 r 的第 k 個進入點 (entry) 的最小節點為節點 s。節點 s 也被稱為節點 r 的第 k 個後繼者。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 r 的第 k 個區間座落於：

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

圖 5 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。

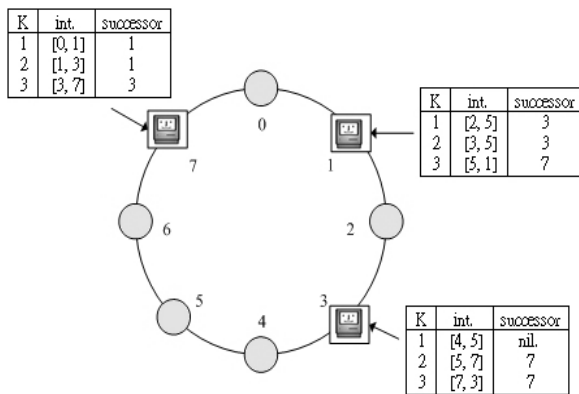


圖 5 常駐代理者建立連線流程

4. 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。

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 1 所示，在示三台模擬聯盟伺服器之間，模擬成員個體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 6 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；然而，在資料分發管理機制上由於僅使用簡單的資料過濾策略所提供的接收方資料管理，對於發送方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。

表 1 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	$\frac{90-S}{S} \times \frac{12}{3} = (I_{Taiwan} - I_{USA}) \times \frac{90}{360}$

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。

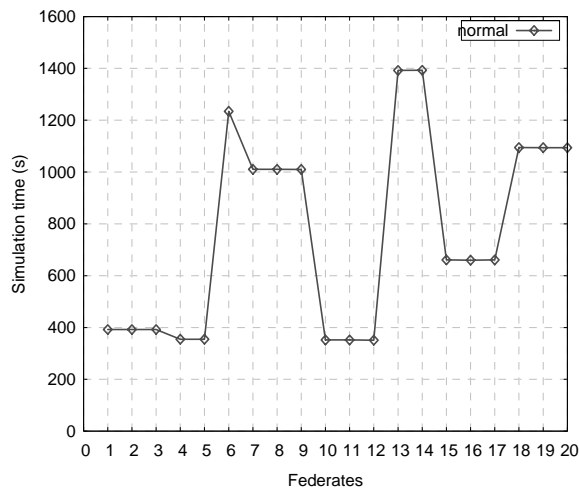


圖 6 宣告管理與資料分發管理實驗數據圖

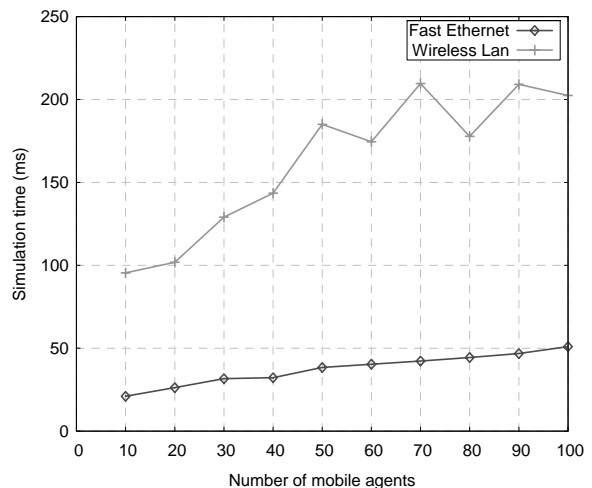


圖 7 環形結構建置實驗數據圖

4.2 環形結構建置實驗

在本實驗中，利用模擬聯盟伺服器建置好的高階架構為系統底層，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。

在實驗中，假設識別空間 $m=7$ 最大共提供 128 個行動節點，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 7 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 8 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。

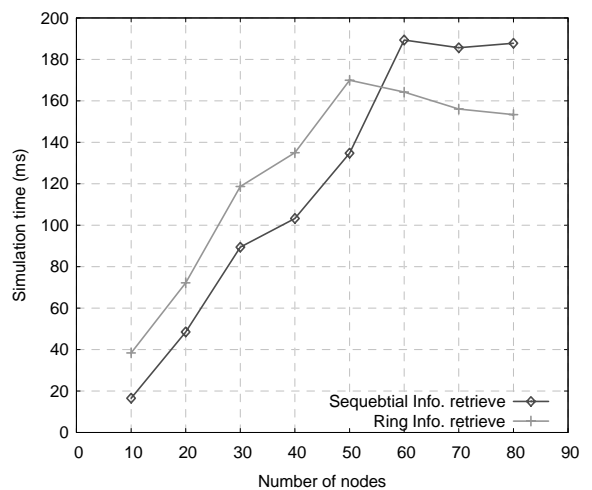


圖 8 資訊蒐集實驗數據圖

4.3 資訊蒐集實驗

在本實驗中，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。

如圖 8 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。

5. 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，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，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。

參考文獻

- [1] A. Bieszczad, B. Pagurek, and T. White, "Mobile agents for network management," *IEEE Communication Survey*, vol. 1, no. 1, 1998.
- [2] A. Puliafito and O. ToMarchio, "Advanced network management functionalities through the use of mobile software agents," in *Proc. of Workshop on Intelligent Agents for Telecommunication Applications*, vol. 1699, Springer, Aug. 1999, pp. 33–45.
- [3] 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, 2002, pp. 5–15.
- [4] C. Bohoris, G. Pavlou, and H. Cruickshank, "Using mobile agents for network performance management," in *Proc. of IEEE/IFIP Network Operations Management Symposium*, April 2000, pp. 637–652.
- [5] Danny B. Lange and Mitsuru Oshima, "Mobile Agents with Java: The Aglet API," *World Wide Web*, Vol. 1, Issue 3, pp. 111-121, 1998.
- [6] Danny B. Lange and Mitsuru Oshima. *Programming and Deploying Java Mobile Agents with Aglets*. Addison Wesley, Reading, Massachusetts, 1998.
- [7] D. Gavalas, D. Greenwood, M. Ghanbari, and M. O'Mahony, "An infrastructure for distributed and dynamic network management based on mobile agent technology," in *Proc. of Conf. on Communications*, 1999, pp. 1362–1366.
- [8] D. R. Karger, E. Lehman, F. Leighton, M. Levine, D. Lewin, and R. Panigrahy, "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web," in *Proc. of the 29th Annual ACM Symposium Theory of Computing*, pp. 654–663, 1997
- [9] D.S. Milojevic et al., *Peer-to-Peer computing, tech. report HPL 2002-57*, Hewlett-Packard Laboratories, Palo Alto, Calif., 2002.
- [10] IBM Japan Research Group "Aglets Workbench," Available through <http://aglets.trl.ibm.co.jp>.
- [11] Q. Lv et al., "Search and Replication in Unstructured Peer-to-Peer Networks," in *Proc. of the 16th International Conf. Supercomputing (ICS'02)*, ACM Press, 2002, pp. 84–95.
- [12] T. Suel et al., "ODISSEA: A Peer-to-Peer Architecture for Scalable Web Search and Information Retrieval," in *Proc. of International Workshop Web and Databases (WebDB'03)*, ACM Press, 2003, pp. 67–73.
- [13] U.S. Department of Defense Modeling and Simulation Office (DMSO), *High Level Architecture Rules. Version 1.3*. Available through <http://www.dmsomil>, 1998.