

# 捷運內階層式網路拍賣系統之設計

## Design of a Hierarchical Internet Auction System for MRT

黃一軒\*、蔡景翔<sup>+</sup>、許宏國<sup>+</sup>、楊正仁\*

元智大學資訊工程系

{ihuang,czyang}@syslab.cse.yzu.edu.tw\*

{s912338,s912326}@mail.yzu.edu.tw<sup>+</sup>

廖名寬

元智大學電機工程系

kenliao@linux.kenliao.idv.tw

**摘要**—捷運系統是都市中重要的公共運輸基礎設施。由於搭乘捷運系統的人數眾多，今日的捷運系統也逐漸發展出商業機能。以台北捷運系統為例，已經有數個購物商城在捷運系統內營業。根據我們的觀察，捷運系統內除了可以發展傳統面對面的商業交易模式外，也可以利用新架設的液晶螢幕來架設網路拍賣系統。因此，我們在本論文中提出一個以台北捷運系統為場景的階層式網路拍賣系統，稱之為 BidTogether。BidTogether 網路拍賣系統是以 web service 的方式設計和階層式架構部署伺服器，除此之外，BidTogether 網路拍賣系統還考慮負載平衡、網路錯誤處理、使用者漫遊和傳輸安全等議題。我們並以微軟公司所提出的 C# 程式語言和 .Net Framework 實作 BidTogether 系統雛形。

**關鍵字：**網路拍賣，階層式架構

### I. 導論

如同世界上其他都市的捷運系統，台北大眾捷運系統在公共交通運輸上扮演十分重要的角色。根據台北捷運公司的統計，經歷十年的發展，目前台北捷運系統每日的載運量已經達到近百萬人次 [1]。不僅如此，除了交通運輸的功能外，台北捷運系統還提供休閒購物的場地，不僅僅百貨公司業者試圖將人潮從捷運系統直接引入百貨公司內，更有業者直接在捷運系統內開設購物商城，利用大量的通勤人潮刺激買氣。

這些商家和業者，仍舊是套用舊有的商業模型，也就是透過實體店面與顧客互動。根據我們的觀察，我們可以更進一步將電子化引入捷運商家中。捷運系統目前架設了許多的大型液晶顯示螢幕，藉由這些螢幕，除了捷運公司可以透過螢幕傳遞列車資訊外，捷運商家也可以透過這些螢幕進行商品的拍賣。

為了讓更多的消費者可以參與拍賣競標活動，利用無線網路技術，消費者可以利用手持式裝置透過網路參與拍賣競標活動。此外，人群藉由群聚在捷運站內的螢幕前參與拍賣競標活動，傳統拍賣競標具有的實際參與競標和搶標的感覺仍然可以保留下來。

因此在本論文中，我們以台北捷運系統為場景開發了一個階層式網路拍賣系統，稱之為 BidTogether。BidTogether 系統包含用戶索引服務模組、內容服務提供模

組和競標後勤服務模組。模組之間利用 web service 技術互相溝通。用戶索引服務模組負責處理使用者登入、商品展示以及拍賣和比價等功能。內容服務提供模組負責 BidTogether 系統與商家之間的商品張貼、商品過濾等功能。最後，競標後勤服務模組負責處理成交後商家與消費者之間的資訊交換和商品交貨等功能。此外，由於使用者可能分佈在不同的捷運站內，BidTogether 系統為每一個捷運站都放置個別的索引伺服器以記錄和比較使用者出價的資訊。這些索引伺服器採用階層式架構，使得 BidTogether 系統可以容易的擴大系統規模和加入新的索引伺服器。

在設計捷運系統的網路拍賣系統時有四項必須考慮的設計議題，首先，系統規模必須足夠容納大量且密集的使用者。由於每日搭載捷運系統的使用者眾多，而且使用者分佈的密度疏密差異很大，BidTogether 系統必須能夠同時服務大量且密集的使用者。在 BidTogether 系統中，我們一方面以 web service 設計系統，因此可以快速部署大量伺服器，另一方面，我們在使用者較多且密度較高的地區放置多台索引伺服器，並根據索引伺服器的負載量將使用者分派給不同的伺服器處理以達成負載平衡的功能。其次，由於使用者在連線的過程中使用無線網路來連接系統，使用者隨時有可能因為網路的錯誤而離線。如果發生網路錯誤導致使用者離線，整個出價的過程都必須還原到使用者出價前的狀態。因此，在 BidTogether 系統中，使用者出價是以 Transaction 的方式設計。接著，由於使用者可能會在不同的捷運站和索引伺服器之間漫遊，使用者的競標行為必須維持正常運作。最後，由於拍賣交易的過程牽涉到敏感的金錢資訊和個人身份資料，通訊過程的保密機制也必須列入考慮。在通訊保密上，BidTogether 系統在整個交易的傳輸過程中採用加密演算法保障使用者的資料不會遭到竊取和竄改。

本論文的第二節將回顧拍賣制度以及過往的網路拍賣系統。第三節將介紹 BidTogether 系統的架構設計和設計議題，第四節介紹我們實作的系統雛形。最後總結全文並探討未來研究方向。

## II. 拍賣制度及文獻回顧

在這一節，我們首先回顧拍賣交易的歷史。接著我們回顧過往的網路拍賣系統，包括集中式拍賣伺服器架構、階層式拍賣伺服器架構和分散式拍賣伺服器架構。

### A. 拍賣的制度和種類

拍賣行為已經擁有超過至少 2000 年的歷史。最早發源於古希臘時代，拍賣是當時交易奴隸的方式之一。如果我們企圖解釋拍賣行為起源的動機的話，回溯前人的研究成果，我們可以參考 McAfee 和 McMillan [2] 引用 Cassady [3] 的解釋：「One answer is, perhaps, that some products have no standard value. For example, the price of any catch of fish (at least of fish destined for the fresh fish market) depends on the demand and supply conditions at a specific moment of time, influenced possibly by prospective market developments.」，換句話說，創造拍賣制度的重要動機之一源自於「不確定的商品價值」。由於受到市場供需情形的影響，商品的真實價格也不斷地隨之變動。

根據拍賣的制度的不同，我們可以區分出四種不同的拍賣形式，分別是英式拍賣（English auction）、荷式拍賣（Dutch auction）、最高價拍賣（first-price sealed-bid auction）以及維式拍賣（Vickrey auction）[2][4][5]。英式拍賣是目前使用最為廣泛的拍賣制度，當拍賣商品時，由拍賣官（auctioneer）負責宣布目前的最高出價記錄。消費者可以喊價，但是所出的價格必須高於目前的最高出價記錄。隨著越多消費者喊價，最高出價記錄也不斷提高。在 BidTogether 系統中，我們也是採用英式拍賣。這種方式不但簡單，也容易被消費者接受。

荷式拍賣與英式拍賣則與英式拍賣相反，在英式拍賣中，拍賣物的價格是不斷地提高的，但是在荷式拍賣中，價格是由高漸漸降低的。首先，拍賣官會喊出最高價，如果有人願意以最高價購買，則成交。如果沒有人願意以最高價購買，拍賣官會逐漸將商品價格降低直到有人願意購買為止。相較於英式拍賣，荷式拍賣的最高價是受到限制的，因此在 BidTogether 中，我們採用的是英式拍賣制度。

最高價拍賣則與前兩種拍賣制度大不相同。當拍賣商品時，所有欲出價的消費者以秘密形式將自己的出價金額交與除拍賣貨主和消費者外的第三者，當一段時間經過之後，第三者會公布出價情形並由最高出價者購買拍賣物。這一種拍賣的方式普遍使用在政府採購上，由於應用在網路拍賣上會讓整個拍賣過程顯得冷清，而且出價次數受限，最高價拍賣制度不合適使用在一般的網路拍賣系統上。

最後一種拍賣制度是維式拍賣制度。其制度與最高價拍賣制度大抵相同，唯一的差別在於雖然由最高價的消費者購買，但是成交金額則是以第二高價的出價記錄為準。這種拍賣制度雖然理論上可行，但是實際應用的場合卻很少。

雖然 BidTogether 系統實際上採用的是英式拍賣制度，但是也可以支援其他三種拍賣制度。所以 BidTogether 系

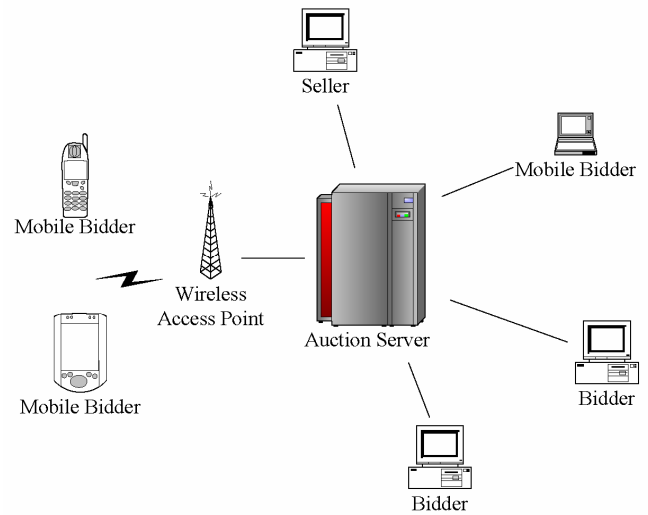


圖 1：集中式伺服器架構。

統本身並不會限制拍賣制度，系統管理者仍舊擁有選擇不同拍賣制度的能力。

### B. 過往的網路拍賣系統

網路拍賣系統經歷多年的發展，根據伺服器架構的不同，過往的網路拍賣系統可以區分為三類，分別是集中式伺服器架構、階層式伺服器架構和分散式伺服器架構。

集中式伺服器架構是三種架構中最容易實作和容易管理的架構。如圖 1，在使用集中式伺服器架構的網路拍賣系統中，系統架構的中心是由拍賣伺服器組成，而參與拍賣的使用者透過網路直接登入拍賣伺服器參與競標。雖然集中式伺服器架構容易實作和管理，這種架構的系統規模卻沒有辦法負荷大量的使用者。

在階層式伺服器架構中，拍賣伺服器則是以樹狀結構被組織起來。Ezhilchelvan 等人在 2001 年時就曾經提出以階層式伺服器架構設計的網路拍賣系統 [6][7]。在 BidTogether 系統中，我們也是採用階層式伺服器架構。階層式伺服器架構具有容易擴充系統規模的特性，適合用在具有大量使用者的網路拍賣系統中。

分散式伺服器架構則與集中式或階層式伺服器架構大不相同。分散式伺服器架構最早可以追溯到 Banâtre 等人提出的 Enchère 系統 [8]。在 Enchère 系統中，商家和使用者的之間可以直接通訊，因此 Enchère 系統也是一個 serverless 的網路拍賣系統。雖然像 Enchère 系統這樣的分散式伺服器架構具有極佳的系統規模擴充性，但是分散式伺服器架構卻有不易管理的缺點。對於一個公共空間和商業系統來說，是否容易管理是一個必須考量的議題。

## III. BIDTOGETHER 網路拍賣系統

在這一節，我們首先介紹 BidTogether 的系統架構。BidTogether 系統採用的是階層式伺服器架構。接著，我們介紹 BidTogether 系統在負載平衡、網路錯誤處理、使用者漫遊和傳輸安全等四個議題上的設計。

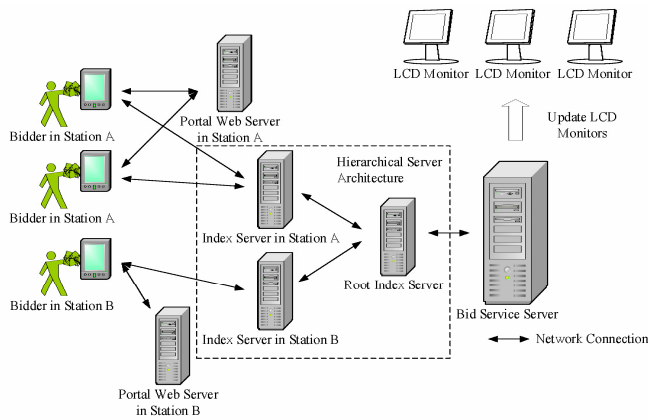


圖 2：用戶索引服務模組。

#### A. 系統架構

BidTogether 系統按照功能可以區分為三個模組，分別是用戶索引服務模組 (Client-ISCM module)、內容服務提供模組 (CSPM module) 以及競標後勤服務模組 (ARSM module)。同時，為了使系統規模可以容易的擴大，用戶索引服務模組中的索引伺服器採用階層式伺服器架構。此外，為了使 BidTogether 系統可以很容易的加入新伺服器元件和功能，我們以 web service 的概念來設計 BidTogether 系統。

用戶索引服務模組負責拍賣系統與消費者之間的互動，包括入口網站、接收使用者出價紀錄和計算最高出價紀錄等等。如圖 2 所示，用戶索引服務模組可以細分為入口網站 (portal web server)、索引伺服器 (index server) 和拍賣服務伺服器 (bid service server)。當使用者以無線手持裝置如 PDA 或是手機連線 BidTogether 系統後，最初連接到的就是入口網站。入口網站首先要求使用者輸入認證資訊，在通過身份驗證後，入口網站會按照目前使用者在捷運站的位置將使用者重新導向到適當的索引伺服器。索引伺服器的功能是接收使用者出價的資訊，並且從眾多消費者出價的紀錄中找出最高者。由於使用者可能位於不同的捷運站，BidTogether 系統必須在不同的捷運站間都放置索引伺服器，為了使 BidTogether 系統規模可以容易的擴充，BidTogether 系統中的索引伺服器採用階層式的架構部署。當拍賣商品時，索引伺服器會記錄自己伺服器上的最高出價紀錄。當使用者出價時，索引伺服器會將使用者提出的價格與自己伺服器上儲存的最大價格比較，如果使用者出價紀錄並未高於自己伺服器上的最高出價紀錄，索引伺服器會拒絕使用者的出價，並且以警告訊息通知使用者競價失敗。如果使用者出價高於自己伺服器上的最高價格，索引伺服器會先更新自己伺服器上的最高出價紀錄，並且進一步的將出價紀錄傳送給根索引伺服器 (root index server)。根索引伺服器上儲存了整個拍賣系統的最高價紀錄。如果由底層索引伺服器傳送來的使用者出價金額高於根索引伺服器上的最高出價紀錄，根索引伺服器會更新自己的最高出價紀錄、LCD 液晶螢幕上的最高出價紀錄，並且通知使用者競價成功。有一點要特別注意的是，因為在 BidTogether 系統中使用者出價是以 Transaction 的方式

設計，在使用者尚未真正完成出價過程前，所有的變更都尚未生效。這麼一來，如果出價的過程沒有完成，系統才能回復到未變更前的狀態。

如果由底層傳送來的使用者出價紀錄低於根索引伺服器上的最高出價紀錄，根索引伺服器會拒絕使用者的競價要求。此時，底層的索引伺服器會先接收到來自根索引伺服器的拒絕訊息，而底層的索引伺服器會用拒絕訊息內附帶的根索引伺服器最高出價紀錄代替自己的最高出價紀錄。透過階層式的伺服器架構，使用者如果出價過低，並且低於底層索引伺服器內記錄的最高出價紀錄，底層索引伺服器會直接拒絕使用者的出價，減輕根索引伺服器判斷使用者出價記錄的工作負載。

在商品競標結束後，根索引伺服器會將得標的使用者資訊、得標金額傳給拍賣服務伺服器，並由拍賣服務伺服器負責將得標資訊傳送給競標後勤服務模組，開始進行結帳和交貨的程序。另一方面，拍賣伺服器上儲存了所有待拍賣的商品清單，捷運站的螢幕必須依序從拍賣服務伺服器上獲取拍賣商品的資訊，索引伺服器也得向拍賣服務伺服器獲取商品資訊。

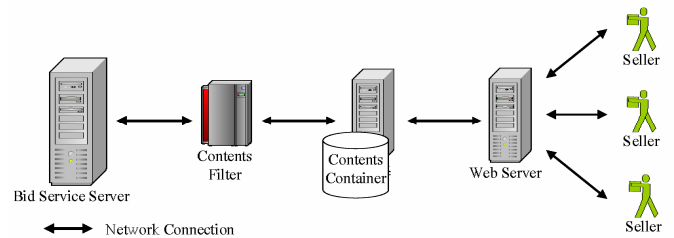


圖 3：內容服務提供模組。

內容服務提供模組提供維護競標商品清單的功能。如圖 3 所示，內容服務提供模組包含拍賣服務伺服器 (bid service server)、內容篩選器 (content filter)、內容容器 (contents container) 和提供商家和系統管理人員操作的網頁伺服器介面。當商家想要拍賣一項新的商品時，首先得登入控制內容服務提供模組的網頁伺服器。透過網頁伺服器，商家可以將商品資訊輸入到內容容器中，而拍賣服務伺服器會按照商家輸入的順序依次拍賣內容容器中的商品。為了防止不適當或不合法的商品透過 BidTogether 系統被拍賣，BidTogether 系統管理者可以透過內容篩選器將不適當的商品刪除。

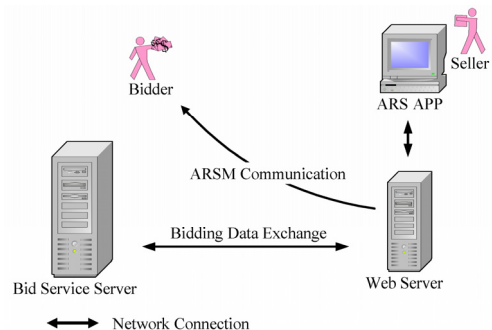


圖 4：競標後勤服務模組。



競標後勤服務模組負責拍賣成交後的後續結帳和交貨等功能。如圖 4 所示，當拍賣成交後，在用戶索引服務模組的拍賣服務伺服器會將成交資訊送到商家用以登入 BidTogether 系統的網頁伺服器上。網頁伺服器會進一步將得標消費者資訊和得標通知透過競標後勤服務訊息 (ARSM connection, 實作上可以是簡訊或電子郵件) 通知消費者和商家。另一方面，商家也可以利用競標後勤服務應用程式 (ARS APP) 主動查詢目前拍賣情形。

## B. 設計議題

在設計捷運系統的網路拍賣系統時有四項必須考慮的設計議題，分別是負載平衡、網路錯誤處理、使用者漫遊和傳輸安全。

首先，由於單一捷運站內的使用者可能數量很多，超過一個索引伺服器所能處理的量，因此在 BidTogether 系統中，每一個捷運站可以部署一個以上的索引伺服器。當使用者登入入口網站後，入口網站會先根據使用者的地點分派索引伺服器。如果按照地區分派的索引伺服器負載量超過該捷運站所有索引伺服器負載的平均值達到 threshold 值，入口網站會改指派另一個負載較輕的索引伺服器給使用者。為了讓入口網站擁有各索引伺服器的負載資訊，所有的索引伺服器都必須向入口網站回報自己目前的負載量。如果不幸地整個捷運站內的索引伺服器都已經達到滿載，入口網站會將使用者分派給鄰近的捷運站內的索引伺服器。

其次，由於使用者透過無線網路連結 BidTogether 系統，我們必須考慮網路發生錯誤的情形。如果使用者出價到一半，在還沒接受到根索引伺服器的訊息前就斷線，BidTogether 系統必須將最高出價紀錄還原到這個使用者尚未出價前的值。為了達成這個目標，在 BidTogether 系統中，使用者出價的過程是以 Transaction 設計[11]。因此，在 BidTogether 系統中，只要使用者在出價的過程中發生錯誤導致未完成整個過程，根索引伺服器和底層伺服器就會略過因為這一位使用者所造成出價紀錄的改變。此外，使用者的出價紀錄不會互相干擾，一旦出價的過程完成，最高交易紀錄也會更新儲存在索引伺服器的資料庫中。

接著，由於使用者可能會在不同的捷運站之間漫遊，因此，使用者很有可能會在捷運站 A 出第一次價，然後移動到捷運站 B。如果使用者打算在捷運站 B 重新出價，使用者首先得登入捷運站 B 的入口網站，然後遵循正常的出價方式參與競標。如果使用者不打算重新出價，如果使用者在捷運站 A 的出價得標，拍賣服務伺服器會詢問所有的入口網站，找出使用者目前所在的位置。如果使用者沒有登入任何一台入口網站，拍賣服務伺服器就將得標金額和使用者資料交給後勤服務模組做進一步的處理。

最後，由於網路拍賣系統中傳遞的都是敏感資料，例如使用者資料，出價金額等等，在網路拍賣的過程中，所有的訊息都應該受到保護，免於遭到他人的竊取和竄改。因此，在 BidTogether 系統中，使用者和 BidTogether 系



(a) 使用者登入。



(b) 使用者出價。

圖 5：BidTogether 系統使用者介面。

統、BidTogether 系統之間的傳輸都必須加密，並且傳輸的資料都必須附上 hash 值，以確保資料的 integrity。

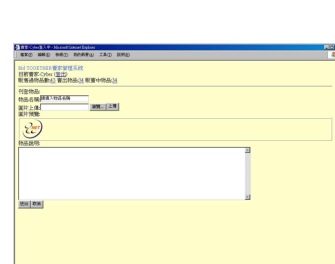
## IV. 系統雛形實作

BidTogether 系統雛形是開發在以 Microsoft Windows 2003 Server 為作業系統的個人電腦之上，同時以 C# 語言 [9] 來開發我們的應用程式。我們的系統雛形遵循微軟公司提出之 .NET 架構 [10] 設計。

圖 5 是我們將所開發之應用程式開發在 Microsoft Visual Studio 內建 PDA 模擬器上的執行畫面，圖 5a 中顯示的是使用者登入提示，圖 5b 顯示的是競標過程的出價操作畫面。這一部份即相當於用戶索引提供模組中的入口網站。



(a) 商家登入。



(b) 上傳拍賣商品資料。

圖 6：商家端使用者介面。

圖 6 是 BidTogether 系統提供給商家的網頁伺服器畫面。包括登入和刊登商品功能展示。這部分屬於內容服務提供模組中的網頁伺服器。

圖 7 是拍賣後勤系統的執行畫面，我們可以看到成交商品資訊，並且查詢目前交易情形。這部分屬於競標後勤服務模組中的競標後勤服務應用程式。

## V. 結論與未來工作

如同世界上其他都市的捷運系統，台北捷運系統負責提供都會區內大眾運輸的功能。不僅如此，由於捷運系統可以帶來數量眾多的搭乘者，捷運系統也發展出商業機



(a) 後勤系統登入畫面。 (b) 商品資訊和狀態維護。

圖 7：後勤系統使用者介面。

能。根據我們的觀察，我們可以將網路拍賣系統引入捷運系統內，應用捷運系統內的液晶螢幕做為商品展示的媒介，並讓使用者可以透過無線網路登入網路拍賣系統。

在本論文中，我們提出一個以台北捷運系統為場景的階層式網路拍賣系統，稱之為 BidTogether。BidTogether 系統以 web service 開發，同時，負責紀錄使用者出價和找出最高價的索引伺服器採用階層式架構，使得 BidTogether 系統具有良好的系統規模擴充能力。另一方面，BidTogether 中也加入了負載平衡、網路錯誤處理、使用者漫遊和傳輸安全的設計。

不過，目前 BidTogether 系統僅僅考慮到網路錯誤的情形，一旦伺服器發生錯誤，我們的拍賣系統只能將使用者重新導向到其他正常的伺服器上。在未來，我們期望能夠更深入探討當拍賣伺服器發生錯誤後，應該如何將拍賣系統從錯誤中恢復的機制。

### 參考文獻

- [1] Taipei Rapid Transit Corporation. <http://www.trtc.com.tw/>.
- [2] R. P. McAfee and J. McMillan, "Auctions and Bidding," *Journal of Economic Literature*, vol. 25, no. 2, June 1987, pp.699-738.
- [3] R. Cassidy Jr., *Auctions and Auctioneering*, Berkeley: U. of Calif. Press, 1967.
- [4] M. P. Wellman and P. R. Wurman, "Real Time Issues for Internet Auctions," *Proc. 1st IEEE Workshop on Dependable and Real-time E-commerce Systems (DARE'98)*, June 1998..
- [5] M.-H. Chen, *The Study of Online Auction—A Case Study of Taiwan E-Bay Auction Website and Taiwan Yahoo-Kimo Auction Website*, master thesis, Natl' Chiao-Tung Univ., Dec. 2003.
- [6] P. Ezhilchelvan and G. Morgan, "A Dependable Distributed Auction System: Architecture and an Implementation Framework," *Proc. 5th International Symposium on Autonomous Decentralized Systems*, Mar. 2001.
- [7] P. Ezhilchelvan, S. K. Shrivastava, and M. C. Little, "A Model and Architecture for Conducting Hierarchically Structured Auctions," *Proc. 4th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'01)*, May 2001.
- [8] J.-P. Banâtre, M. Banâtre, G. Lapalme, and F. Ployette, "The Design and Building of Enchère, a Distributed Electronic Marketing System," *Communications of the ACM*, vol. 29, no. 1, Jan. 1986, pp.19-29.
- [9] A. Jones, *C# Programmers's Cookbook*, Microsoft Press, 2003.
- [10] J. Richter, *Applied Microsoft .NET Framework Programming*, Microsoft Press, 2002.

[11] G. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems: Concepts and Design*, 3rd edition, Addison-Wesley, 2001.