

第三章 多對多合作協商模式與多對多退讓策略

3.1 協商機制之建立

由於本研究的目的是在於提出中立的多對多協商機制，不偏好買方或賣方，所以買賣雙方使用同樣的協商策略來進行雙向提案溝通，協商結束後買方可以選擇賣方，賣方也可以選擇買方。參考到電子市集多對多商業模式—交易所模式和雙向競標模式，交易所模式有交易所來維繫買賣雙方關係，買賣雙方甚至不必有直接的往來關係便可達成交易，而雙向競標模式中有拍賣主持人依類似股票交易所撮合買賣雙方的方法，決定成交價格以及買賣雙方。由此看來，電子市集多對多商業模式中均有一個中介角色存在，用來負責撮合及聯繫買賣雙方，因此本研究的多對多協商機制中也有一個中介角色存在，稱為市場代理人(Market Agent)。而在多對多協商中買賣雙方對撮合的需求有二，第一是買賣雙方不知道跟誰協商，也就是不知道誰是適合的協商對象，第二是依據協商結果選擇協商對象，買方可以選擇賣方，賣方也可以選擇買方。所以一定要有中間人來幫雙方作兩次的撮合，第一次撮合是幫買賣雙方篩選適合的協商對象，第二次撮合則是由買賣雙方的選擇來決定成交對象。所以本研究的多對多協商模式可以下圖來表示整個多對多協商流程大致可以分成幾個步驟(見圖 12)：註冊、提出買賣需求、市場代理人進行第一次撮合、進行多對多協商、市場代理人進行第二次撮合、決定成交對象。

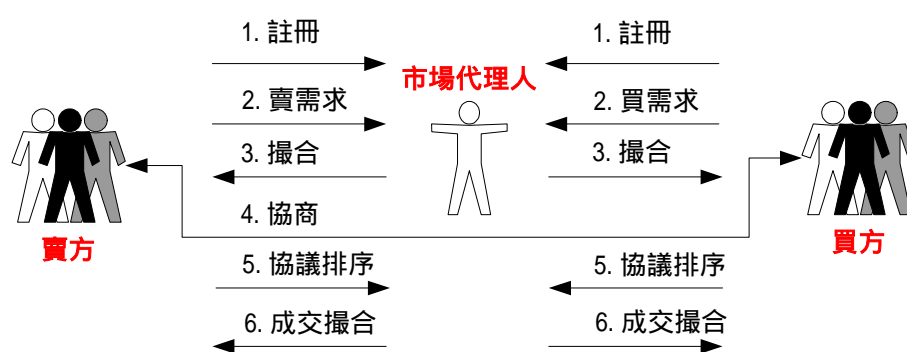


圖 12、多對多合作協商模式

而在多對多協商機制的設計，主要是參考並結合多個研究(朱宏揚，民 89；張婉華，民 90；Kowalczyk and Bui, 2000)的協商機制而來，見表 3。在文獻探討 2.3.2 中已經介紹了一些自動協商技術，其各有優缺，而本研究採用限制滿足問

題搭配集合式提案(Grouping Proposal)策略來產生每次協商回合的一組候選提案。本研究認為，更佳協議策略無法跟集合式提案並用，因為集合式提案的協議一定落在效率前緣上，也就是無法在找到另外的點可以同時滿足雙方，因此雙方找到滿意的解之後不需要再繼續協商。而因為集合式提案使協商結果能落在效率前緣，然而更佳協議策略不一定能保證使協商結果落在效率前緣上，所以本研究選擇集合式提案而棄更佳協議策略(Better deal strategy)。

並使用決策分析工具--多屬性效用理論，讓代理人能夠評估多個提案之間的效用。而在退讓策略方面，本研究則提出「多對多退讓策略」以適切多對多的協商環境，讓多頭協商之間是互相影響的。

表 3、協商機制之設計

使用者喜好設定	交易對象條件與需求、急切程度 協商議題的可接受範圍、偏好值、權重
提案衡量	多屬性效用理論(MAUT)
可接受度標準	Kowalczyk and Bui 2000
提案的產生	限制滿足問題(CSP)
協商策略	多對多讓步策略、集合式提案

3.1.1 使用者喜好設定

(1)交易對象條件與需求

買賣雙方對交易對象可能會有一些限制條件，而中介的市場代理人便根據這些條件來幫買賣雙方找到合適的協商對象。

(2) 完成交易的急切程度

使用者必須設定完成交易的急切度 *desire*，以利讓代理人計算出提案的最低可接受效用。

(3) 協商議題的可接受範圍、偏好值、權重

協商者必須在進行協商之前設定好自己對每個協商議題的可接受範圍以及

所偏好的值(preferred value)和對各協商議題的權重。

3.1.2 提案衡量

在本研究中，應用多屬性效用理論以衡量協商提案的效用，可依據效用值以評估、比較和排序候選方案。 x_i 是第 i 個協商議題，則 v_i 是 x_i 的價值函數(utility function or value function)， w_i 是 x_i 的權重。一個包含協商議題 $x = \{x_1, x_2, \dots, x_n\}$ 的提案效用 v 可視為這些協商議題的函式：

$$v(x_1, \dots, x_n) = \sum_{i=1, \dots, n} w_i \times v_i(x_i), \quad \sum_{i=1, \dots, n} w_i = 1$$

每個協商議題的價值函數為衡量每個議題的獲利狀況，可以表示為：

$$v_i(x_i) = \frac{L_i - x_i}{L_i - H_i}$$

L_i ：第 i 個議題的最低獲利出價

H_i ：第 i 個議題的最高獲利出價

由於每個協商議題對於使用者的效用函數可能是各種圖形，邊際效用遞增(Convex)、邊際效用遞減(Concave)或不規則的函數圖形。本研究為簡化系統，假設使用者對各議題的效用函數圖形為線性，即效用為線性增加或線性減少。

3.1.3 可接受度標準(Acceptability Criteria)

對使用者而言決定接受對方提案與否完全根據本身對提案感覺，然而協商代理人無法依據此方法來判斷是否成交，因此使用者必須將本身的感覺化成效用函數，代理人藉由效用函數來評估使用者對提案的滿意程度。當滿意度超過一定的

數值，且滿足使用者所設定的各屬性可接受範圍，代理人才可以決定接受對方提案而完成協商。

假如一個提案(或 solution)同時滿足雙方的可接受度標準則可為雙方所接受。此可接受標準牽涉到一個可接受的效用值，和對各屬性的可接受範圍和相關限制，然而可接受的效用值要怎麼計算呢？張婉華(民 90)認為當對方提案給予我方的效用 > 0 ，且提案中各議題的出價滿足我方的可接受範圍，則可以接受，也就是最低可接受效用 > 0 ；而朱宏揚(民 89)則讓使用者自行設定一個可接受效用值。而本研究採用 Kowalczyk and Bui (2000)的做法，考量每個協商議題所偏好的值 (preferred value)以及使用者完成交易的急切度 *desire*，來計算提案最低可接受效用值。一個可接受度標準的最低可接受效用可表示為：

$$v^* = d \times v(x^*), d = 1 - 0.3 \times \text{desire}, \text{desire} \in [0,1]$$

$x^* = \{x_1^*, \dots, x_n^*\}$ ：為各議題使用者所偏好的值

v^* ：為一個可接受的提案效用值

d ：為急切度的係數(coefficient expressing desirability)

desire：為使用者完成交易的急切度

一般來說達成交易的急切度越高，則可接受度標準越低。有研究指出真實世界中，一個可被接受的協議往往具有 70%或以上的偏好值(preferred value)，所以急切度係數為急切度的函式 $d=1-0.3 \text{desire}$ 。而且急切度也可能影響了協商的策略和協商行為，例如，急切度越高，讓步的程度也越高，但是本研究僅將急切度用來計算最低可接受效用，當一個提案的效用高於或等於可接受度標準 $v(x) \geq v^*$ ，則該協商者可以接受此提案。同時可接受度標準也是自己的一個限制，以確保本身所提出的提案可為自己所接受。

3.1.4 提案的產生(Offer generation)

協商提案的產生為主要的決策制定過程，影響了協商的過程和結果。協商代理人提出的初始提案為對協商者能產生最大效用的提案(此做法同張婉華，民 90；Kowalczyk and Bui 2001)，而且由賣方先提提案，並採用限制滿足問題配合集合式提案來產生協商提案。協商提案的產生為搜尋本身的可行解空間，朝向可以達成協議的方向提出提案，而協商策略則影響著如何搜尋。提案的產生牽涉了每個協商者所握有的資訊，包括自己設定的偏好資訊包括：可接受度標準、限制、偏好，以及上一次己方提出的提案和對方的因應提案(counter-offer)。

一個提案的產生包括保持限制一致性(Constraint Consistency Maintenance)和搜尋協商議題的數值。

一、保持限制一致性(Constraint Consistency Maintenance)

所謂的限制推論(Constraint-based reasoning or Constraint-based propagation)便是藉由剔除那些不可行的，而不斷的縮小提案的可搜尋空間。保持限制一致性就是不斷把所得到的資訊形成限制式，並確保提案可行解空間均符合限制。而協商中所會產生的資訊為對方所提的提案，和我方已提的提案。假設對方上次的提案為 $\{x_1'', x_2'', x_3''\}$ ，其給予我方的效用為 $v(x_1'', x_2'', x_3'')$ ，則我方新提案的效用不會低於 $v(x_1'', x_2'', x_3'')$ ；假設上次我方提出的提案為 $\{x_1', x_2', x_3'\}$ ，效用為 $v(x_1', x_2', x_3')$ ，但不為對方所接受，則我方新提案的效用不會高於 $v(x_1', x_2', x_3')$ ，我方必須提出同效用的其他替代方案或退讓，以確保提案不會遠離協議的方向。下表為協商進行中所會產生的限制：

表 4、協商時會產生的限制

起始資訊	限制
協商議題 $x = \{x_1, x_2, x_3\}$	$C(x) : x_1^{\min} \leq x_i \leq x_1^{\max}$
提案的效用	$C(x) : x_{123} = v(x) = \sum_{i=1, \dots, 3} w_i v_i(x_i), \sum_{i=1, \dots, 3} w_i = 1$
可接受度標準	$C(x) : x_{123} \geq v^*$
交換資訊	限制

上次產生的提案 $\{x_1', x_2', x_3'\}$	$C(x) : x_{123} \quad v(x_1', x_2', x_3')$
上次收到的對方提案 $\{x_1'', x_2'', x_3''\}$	$C(x) : x_{123} \quad v(x_1'', x_2'', x_3'')$

二、搜尋新提案

協商策略描述了協商進行規則以達成協議，也說明協商退讓的程度 Δ ，本研究所提出的多對多讓步策略就是在計算每個協商周期的讓步程度 Δ 。而本研究採用集合式提案應用在限制滿足問題上，集合式提案在搜尋提案做法如下：上一次己方所提出的提案效用值為 x_{123}^{t-1} ，扣掉退讓的程度 Δ 後為 $x_{123}^t = x_{123}^{t-1} - \Delta$ ， x_{123}^t 到 x_{123}^{t-1} 的效用範圍即為這次集合提案的搜尋範圍，然後以此新限制

$x_{123}^t \leq \sum_{i=1, \dots, n} w_i v_i(x_i) < x_{123}^{t-1}$ 配合己方其他限制(見表 5)，找出一組協商議題的數值 $\{x_1, x_2, x_3\}$ ，然後將此集合提案提給協商對方，讓對方回覆哪個提案對對方而言效用最高。

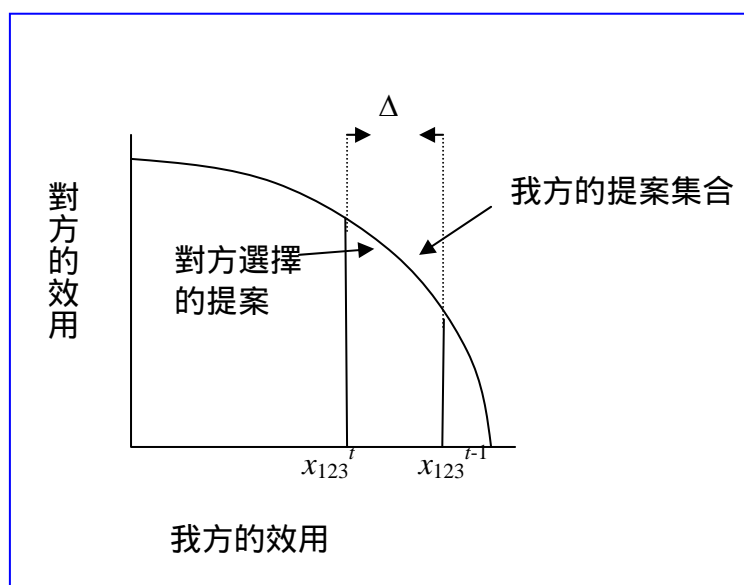


圖 13、退讓和集合式提案

3.1.5 協商的結束

協商的結果有兩種，成功或失敗。成功的協商是雙方找到了彼此都可以接受的協議，也就是當甲方提出的提案可以滿足乙方的可接受度標準，甲乙兩方達成協議而協商結束。失敗的協商則是當甲方已無提案可提，而且甲方仍無法接受乙方這次協商周期的提案時，協商便告失敗而結束，因為協商乃是一來一往的動作，當某方已無提案可提且無法接受對方目前所提出的提案時，協商便失敗。

3.2 多對多讓步策略

讓步策略說明在協商時如何決定讓步幅度，猶如協商者常常必須決定何時該退讓多少才算適宜，退讓太多會讓自己太快喪失談判籌碼，退讓太少又會造成談判破裂。本研究的多對多讓步策略的啟發想法是：多頭協商之間實際上必須能夠互相影響，也就是觀察其他同時在進行協商的夥伴，藉由夥伴達成協議的情況來調整己方的讓步幅度。而買賣雙方均使用多對多讓步策略，跟買賣雙方使用其他非多對多讓步策略來比較，會有什麼不同？在文獻探討中的四種讓步策略(固定、遞減、互利、期望效用讓步策略)，若直接放在多對多協商的環境下使用，則有一個缺點，多頭協商之間無法互相影響，也就是我方和甲的協商成果並不影響我方跟乙方的協商，而這在現實世界中是不可能的。試想，你才剛去甲商店買了一件商品花了 500 元，後來到乙商店中類似的商品卻要賣你 600 元，那麼你一定會跟乙方殺價，因為你有了之前協商成果的的經驗。

多對多讓步策略為了使多頭協商之間互相影響，因此派出去協商的子代理人必須在每個協商週期回報協商進度，由回報的協商結果來影響下一協商周期的協商，簡述如下：當我方派出去協商的子代理人均尚未達成協議之前，使用的是互利的讓步策略，然而當有一個子代理人達成效用 u_1 的協議，那麼其他尚未完成協商的子代理人的提案效用下限在接近(效用差 10)效用 u_1 以下開始減少退讓，也就是將對方提案給予我方的效用乘上 $(1/p)$ 就是我方的讓步幅度，接著當又有一個子代理人完成效用 u_2 的協議，則將 u_1 和 u_2 從大排到小，假設 $u_1 > u_2$ ，則在快接近效用 u_1 時開始減少 $(1/p)$ 的退讓，在快接近效用 u_2 時，退讓縮減幅度(c_shrink)

累加成 2，開始減少 $(1/p)^2$ 的退讓，如此以下類推。

```
c_shrink=0; //c_shrink 代表退讓縮減的幅度
for(int i=0; i<deal_uti.size(); i++){
    if (utiDownBound -10 <= ((Float)deal_uti.get(i)).floatValue()){
        c_shrink++;
    }
}
```

所以每個協商週期必須重新計算 c_shrink ，再用 c_shrink 來計算此次提案的退讓幅度(concession)，計算公式為：

退讓(concession)=此協商週期對方給予我方的效用增加 $\times (1/p)^{c_shrink}$

因為本研究將使用集合式提案，在某個效用範圍內的所有提案一次提給對方選擇，所以每個協商週期都要以退讓幅度來計算提案效用上限和效用下限，而當計算出來的提案效用上限(utiUpBound)和提案效用下限(utiDownBound)之間有包含效用 u_1 (其他夥伴達成協議的效用)時，便把該次 utiDownBound 設定為效用 u_1 ，程式如下。用意為當你知道你夥伴達成效用 u_1 的協議時，而此次提案效用範圍中間有效用 u_1 的值，故把提案的效用下限設為效用 u_1 ，希望能達到比效用 u_1 更好協議。

```
for(int i=0; i<deal_uti.size(); i++){
    float deal_utility = ((Float)deal_uti.get(i)).floatValue();
    if (utiDownBound<deal_utility & utiUpBound>deal_utility){
        utiDownBound=deal_utility;
        concession=utiUpBound-utiDownBound;
    }
}
```

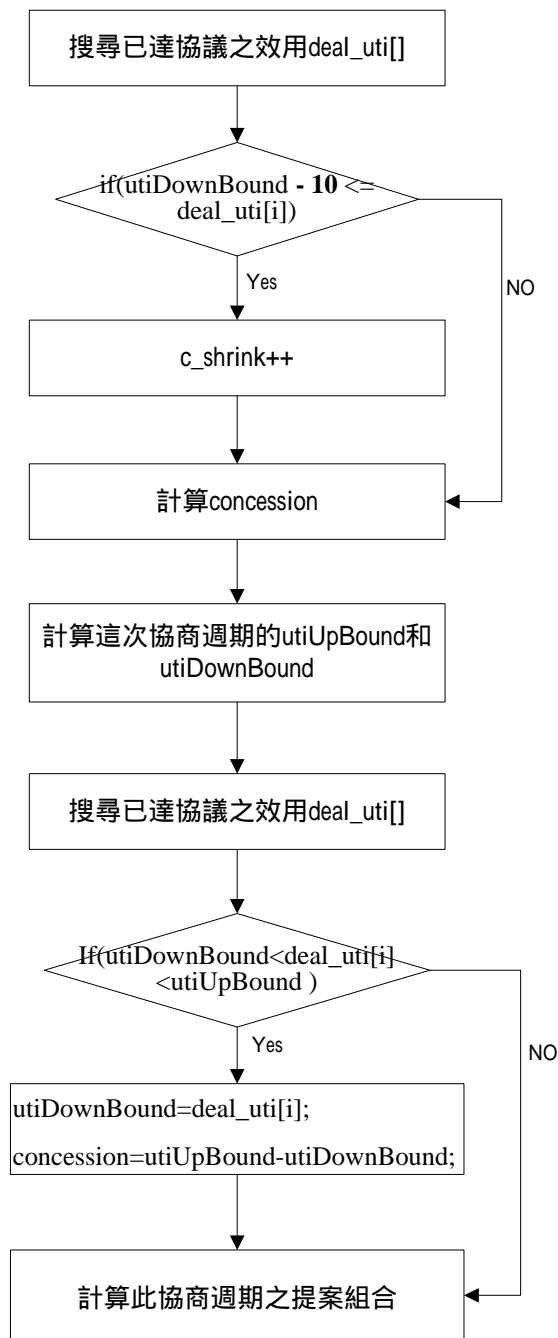



圖 14、多對多讓步策略

在文獻探討的四個讓步策略中，何者比較適合用在多對多協商環境下來跟多對多讓步策略來比較呢？張婉華(民 90)研究一對一協商時，將期望效用的讓步策略與其他三種讓步策略作比較，買賣雙方其中一方使用期望效用讓步策略，另一方則使用固定、遞減、互利的讓步策略，實驗發現期望效用策略可以獲得較高的效用。但是本研究是要提出一個中立的協商環境，買賣雙方使用相同的協商策略，以確保公平性，倘若協商雙方都採用期望讓步策略的話，就不一定會替買賣

雙方同時提高效用，並且雙方在設定期望效用的不同會影響雙方最後協議的效用，如此造成協商雙方一定會誇大其期望效用，使得雙方所設定的期望效用之間沒有交集而造成協商失敗。況且買賣雙方設定的期望效用不同，這樣的協商是否具有公平性，令人質疑。而在張婉華(民 90)的實驗中，可以看出互利的讓步策略是優於固定讓步策略和遞減的讓步策略，因此本研究的多對多讓步策略便是延伸自互利讓步策略而產生，並和互利讓步策略作模擬實驗比較。

3.3 系統架構

整個多對多協商流程大致可以分成幾個步驟(見圖 15)：註冊、提出買賣需求、市場代理人進行第一次撮合、進行多對多協商、市場代理人進行第二次撮合、決定成交對象。詳述如下：

- (1).買賣雙方在進行協商前，必須先透過第三方所提供之註冊機制先行註冊、審核通過，並自註冊網站上下載所需的程式。
- (2).買賣雙方的主機均各有一個 AgletManager，負責管理協調多個與交易對方協商的 aglet，當使用者有交易的需求時，可以透過 AgletManager 對市場代理人發出交易請求。
- (3).市場代理人將發出請求的代理人紀錄在 AUL(Active User List)，等到累積一定的買方賣方後，才進行第一次撮合，所謂的第一次撮合是指市場代理人會依據買賣雙方的需求(地區、數量等)來撮合彼此，使買賣雙方所協商的對象均能符合自己的需求條件。
- (4).第一次撮合後，MarketAglet 產生買賣雙方所適合的協商對象清單(Negotiation Partner list, NPL)，並傳送給買賣雙方 AgletManager。
- (5).買方的 AgletManager 根據 NPL 決定複製多少個 solverAglet 並派遣到賣方的主機上進行協商；而賣方的 AgletManager 根據買方所派遣來的 solverAglet 數目，決定複製多少個 s_solverAglet 以進行協商，每個 aglet 與對方的 aglet 進行一對一的協商，也就是買賣雙方同時進行多頭協商(見圖 15)，並且多頭協商之間因為多對多讓步策略而互相影響。
- (6).協商完畢後，達成的協商協議不一定是實際交易的協議，必須看買賣

雙方是否都選擇對方作為交易對象，所以買賣雙方的 AgletManager 對各協議的效用予以排序產生交易對象排序清單 (Transaction Partner Ordering List , TPOL) ，此時使用者可以介入刪除不理想的協商結果。

(7).買賣雙方的 AgletManager 把確定的 TPOL 傳送給 MarketAglet ， MarketAglet 藉由進行第二次撮合後才決定買賣雙方的成交對象(Deal List) ，然後把 deal list 回傳給買賣雙方。買方只會有一位賣方作為成交對象，而賣方可能會有多位買方，依據賣方的產品欲銷售數量而定。

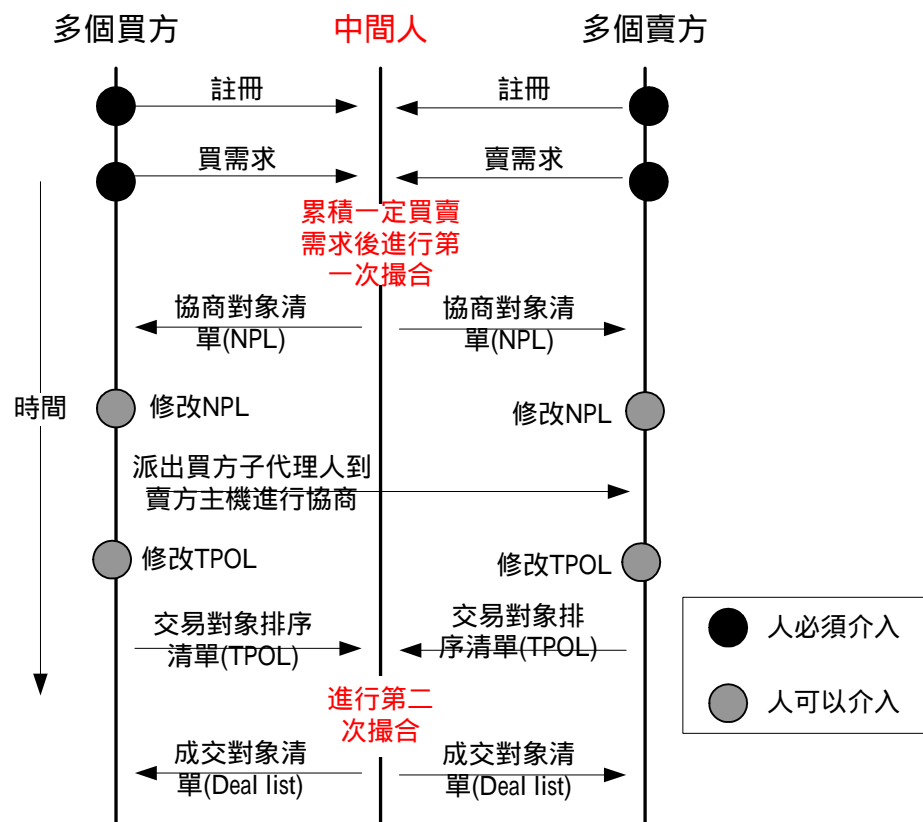


圖 15、多對多協商流程圖

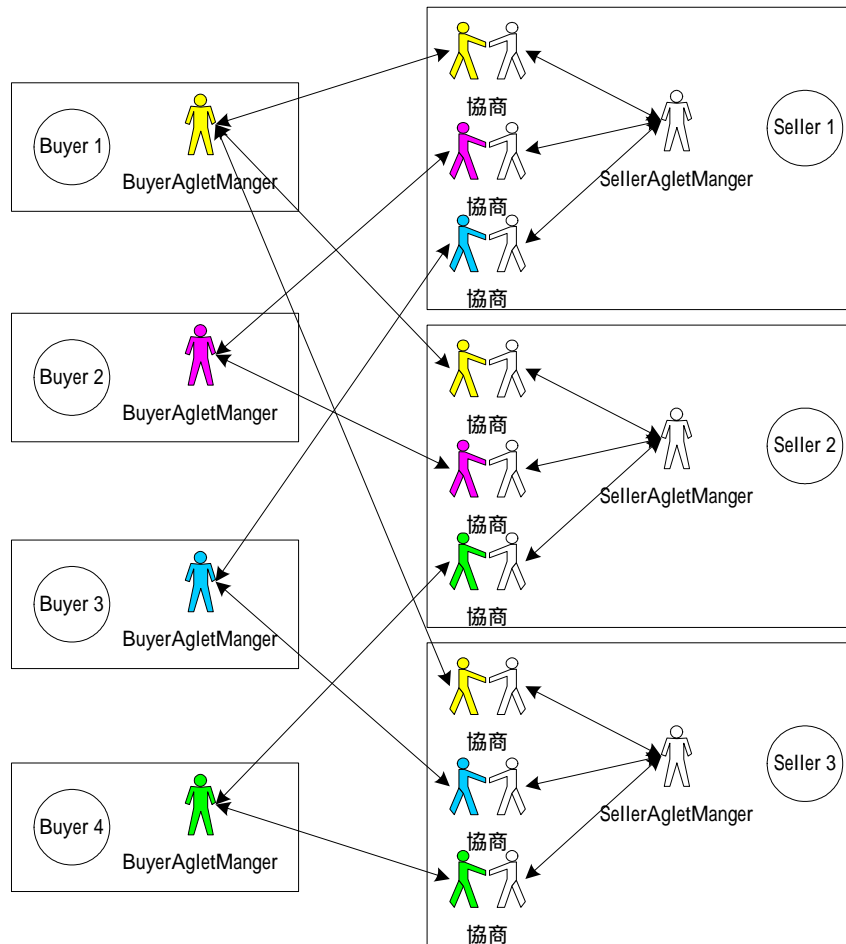


圖 16、行動代理人之多對多協商圖

3.3.1 買方管理代理人(BuyerAgletManager)

BuyerAgletManager 可以說是買方的採購主系統，使用者在此設定採購需求與決策參數。當使用者想要購買商品時，透過 BuyerAgletManager 發出 request，通知 MarketAglet 想購買多少數量的商品，以及對賣方的篩選條件。

當使用者設定好協商變數即協商相關設定後，BuyerAgletManager 便產生 solverAglet 來攜帶這些資料，並產生符合使用者需求的限制滿足問題，當收到 MarketAglet 傳回的 NPL 後，使用者可再刪除不想與之進行協商的對象，然後派出 solverAglet 到賣方的主機上進行協商，在協商進行時，BuyerAgletManager 會收到 solverAglet 在每個協商週期(提出提案和收到對方提案)的成果，讓使用者可以監看交易的進行。

協商結束後，BuyerAgletManager 將各協議依其效用排序，此時使用者可參

考賣方的詳細資料，例如賣方提供的設備和服務，決定刪除哪些不滿意的協議，得到一份 TPOL(Trading Partner ordering list)，BuyerAgletManager 再將 TPOL 傳給 MarketAglet 進行第二次撮合。

3.3.2 賣方管理代理人 (SellerAgletManager)

每個供應商伺服器端將存在一個 SellerAgletManager，當使用者想要出售商品時，可在此設定好使用者的偏好和效用函數等資料，然後透過 SellerAgletManager 通知 MarketAglet 有多少數量的商品想要進行交易。

當 BuyerAgletManager 派遣 solverAglet 過來時，SellerAgletManager 便複製產生一個 s_solverAglet 與之協商，並協調管理多個 s_solverAglet。在協商進行時，SellerAgletManager 會收到 s_solverAglet 每協商週期的協商結果，讓使用者可以監看交易的進行。

協商結束後，SellerAgletManager 將各協議依其效用排序，此時使用者可決定刪除哪些不滿意的協議，得到一份 TPOL，SellerAgletManager 再將 TPOL 傳給 MarketAglet 進行第二次撮合。

3.3.3 市場代理人(MarketAglet)

當買賣雙方的 AgletManager 通知市場代理人要進行交易時，市場代理人先確認此買賣雙方之代理人事先已經註冊過，並驗證其帳號密碼正確與否，然後將之排序和紀錄在 AUL，等到累積一定的買賣人數之後，便進行第一次的撮合。第一次撮合買賣雙方，是依據賣方所提出對交易對象的偏好，和產品需求，例如甲旅行社需要的旅館是一般觀光旅館等級以上，位於北投附近。市場代理人將撮合後的 NPL 傳給買賣雙方的 AgletManager，讓買賣雙方知道哪些是適合進行協商的交易夥伴。

當買賣雙方協商結束，並回傳自己的 TPOL 時，市場代理人便進行第二次撮合，賣方可有多個買方來成交，以賣方所給定的商品數量為限；而買方只有一個賣方來成交。然後再傳回買賣雙方所成交的對象名單(deal list)。

市場代理人(Market Agent)的元件 (見下圖)：

1. Market Cycle Manager (MCM)：管理並執行市場週期。一個市場週期的頻率是一個事先定義的系統變數，累積 M 個買方及 N 個賣方後才啟動 Match Maker I 執行撮合，開始一個市場週期，在本研究中設定 $M=4$ 、 $N=4$ 。

2. Match Maker I (MM I)：由 USM 取得 AUL，並進行買賣雙方交易條件需求的第一次撮合(例如：地區、數量、等級)。

3. Match Maker II (MM II)：利用 TPOL 進行第二次撮合，產生確定成交清單 (deal list)。

4. Active User List (AUL)：紀錄等待被撮合的 active 買方和賣方。

5. Negotiation Partner List (NPL)：經 MM I 撮合後產生的協商對象清單。

6. Deal list：成交清單

7. User Status Manager (USM)：為與買賣代理人進行溝通的窗口。當每個新使用者提出交易需求時，以及一個市場週期結束時，USM 負責更新 AUL，當 MM I 被啟動時，MM I 便向 USM 索取 AUL 以進行第二次撮合；買賣雙方協商完後，負責蒐集買賣代理人傳回的 TPOL，並傳給 MM II 以進行第二次撮合。

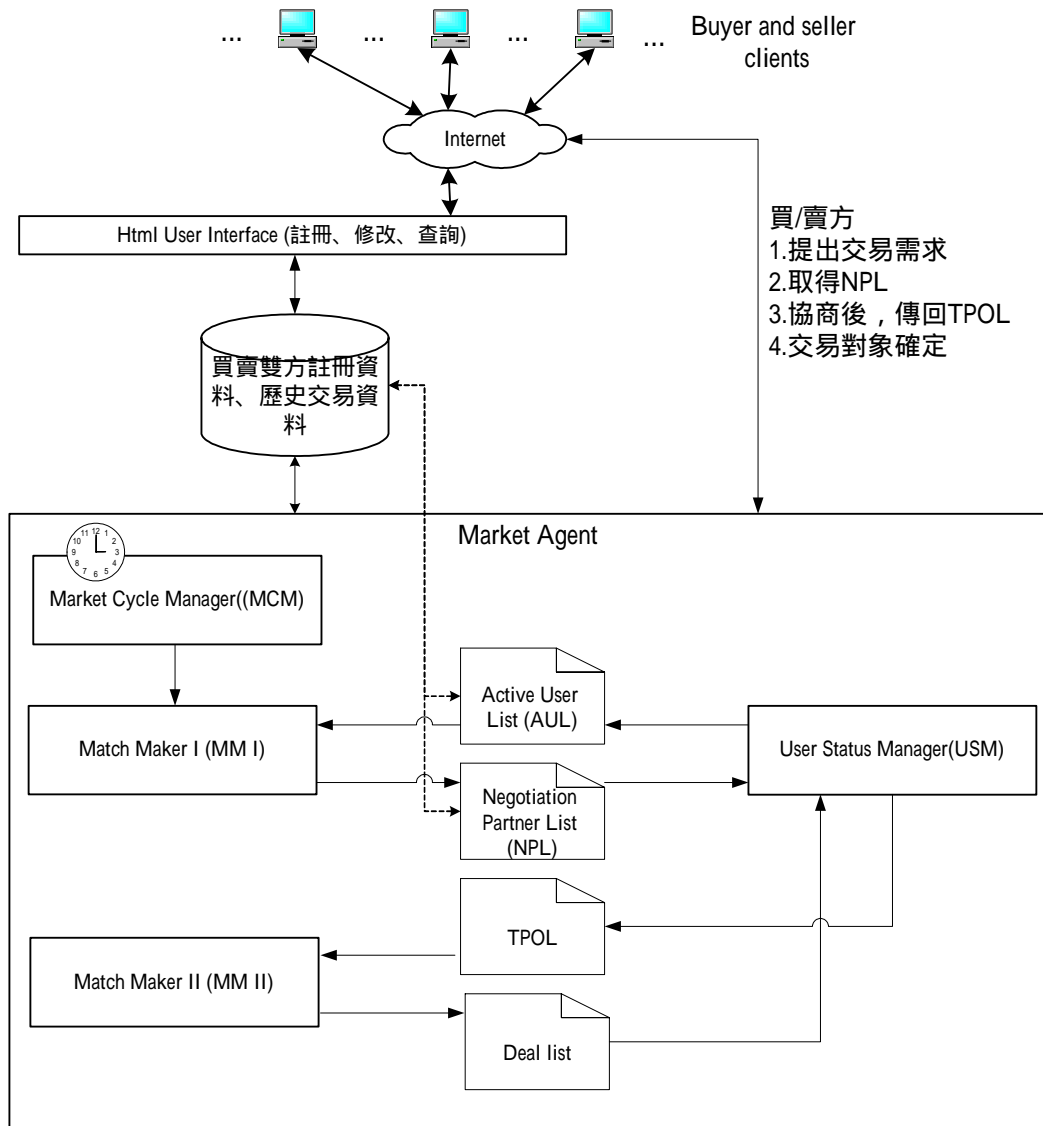


圖 17、系統架構圖