多樣需求與資源環境中之智慧型e化服務決策研究—以垃圾桶模式為基礎

苑守慈
政治大學資訊管理系

呂知穎
政治大學資訊管理系

摘要

為因應人類生理或心理上的需求，而產生了形形色色之服務。隨著高科技不斷地發展，人類的未來生活，將會是充滿e化服務的生活環境。在此環境中，並非所有人均能了解各應用服務，更不知該選擇何服務才能滿足自身之多重需求。本論文設計一決策機制，當人們有多重需求時，能考慮有形及無形資源之有效利用，並考量不同個體之使用偏好及興趣，提供適合個人的e化服務決策建議。本論文之應用環境，符合垃圾桶模式中的無政府狀態之三大特性，然而原垃圾桶決策方式卻不適用於個人。因此，本論文之主體為一智慧代理人，將以垃圾桶模式的決策原理做為基礎，並對其加以修改，分為二階段的決策過程。在第一階段，將使用一考量資源使用效率之task-chosen演算法，並搭配增強式學習中之AH-learning演算法；在第二階段，則是使用BDI代理人的架構。本研究所提出之提供e化服務建議的決策機制，預期將促使應用服務能不斷地創新及進步，並使資源獲得更有效之利用，使得人類擁有高品質的生活環境。

關鍵字：e化服務、垃圾桶模式、智慧型代理人、增強式學習

An Intelligent e-Service Strategy Given Manifold Needs and Resources: The Garbage Can Model Perspective

Soe-Tsyr Yuan
Department of Management Information Systems, National Chengchi University

Chih-Ying Lu
Department of Management Information Systems, National Chengchi University

Abstract

There are manifold services to fulfill people's physical and mental needs. Through the continual development of high technique, people will live in the environment surrounding e-services in the future. In this environment, it is hard for everyone to understand all e-services and choose a service to fulfill his/her multiple needs. Therefore, the paper presents a decision mechanism which provides a suitable e-service strategy for people when they have the multiple needs, considering the uses of resources (tangible and intangible) and different preferences and interests for different people. This paper's application environment satisfies the three general properties of an organized anarchy of "Garbage Can Model". This paper extends the model to accommodate its application to individuals in terms of an intelligent agent. The intelligent agent uses a two-phase decision process. The first phase is a task-chosen algorithm considering resource utility and AH-learning in the context of reinforcement learning. The second phase then exerts the BDI reasoning. This paper presents a decision strategy providing e-service tactics (that can use resources effectively) and enabling people to enjoy high quality life.

Key words: e-Services, garbage can model, intelligent agent, reinforcement learning

C.E.P.S.

壹、緒論

人類的慾望無窮，無論是生理或心理方面，均有所需求。在ERG理論(Alderfer 1969)之中指出，人有存在(existence)需求、關係(relatedness)需求以及成長(growth)需求，並且在同一時間內，會有一個或一個以上的需求存在，且需求之間並無特定之順序，甚至是因人而異。人類為了滿足需求，便會去尋找解決之道。少數人從中視得商機，於是提供了相對應的服務，來滿足人們各式各樣的需求。

傳統的服務過程，大多需要提供服務者與被服務者實際參與，但自從電腦和網路與起之 後，產生了極大的變化，也創造了更多的商機。以拍賣為例，傳統的拍賣方式，參與者必須到指定場所；但現在知名的eBay、Yahoo奇摩等，均提供了線上拍賣的服務。近幾年來，無線技術的快速進步以及行動裝置的不斷改良，使得行動商務成為當紅產業，此外，普及運算(pervasive computing)亦是當下的熱門話題。往後數年，當行動商務與普及運算愈趨於成熟，再搭配上為因應人類需求而生的服務應用，人們將無論在何時何地，均能享受到服務無所不在的高品質生活。在未來，人們會愈來愈習慣充滿e化服務的生活，但服務提供者所提供之服務，多半屬於被動 式。當人們要選擇使用何種服務來滿足自身的需求時，有些問題也隨之而生：

- 服務種類過多，難以下手：同一種類的服務，不會只有一個提供者，而各個提供者所提出的服務又不盡然相同，造成難以決策的情況。
- 無法在短時間內找到適合的服務：人的需求有時是臨時的，在緊急的情況之下，怎麼可能再花時間去投入找尋。
- 找到的服務不見得真正符合自身需求：雖然取得了服務，但有可能因為不願再花時間去嚐試其他相類似的服務，或是根本不知道還有其他的服務，結果反而沒使用到真正符合自己需求的服務。

上述的決定服務情境，極類似於組織決策過程中的無政府狀態(organized anarchy)。針對無政府狀態的模糊決策情境，學者(Cohen et al. 1972)提出了垃圾桶模式(garbage can model)來解決此問題。然而垃圾桶模式雖可解決無政府狀態的決策過程，但至今多是應用在教育及政治領域上的決策環境，而本論文之決策情境是應用於社會環境，是希望能找出滿足人們多重需求的應用服務，因此本論文提出一個改良式之垃圾桶模式以進行服務決策之建議。

簡言之，人類未來將會生活在充滿e化服務的環境，在任何時間地點均能享受多種服務帶來的便利性，但人們在同一時間可能有多重需求，造成難以選擇適當應用服務的混亂情況。此種混亂情況恰符合無政府狀態的三特性。針對此一現象，本論文設計出一套服務決策機制，面對現今的各式各樣服務，當人們有多重需求產生時，即能運用此一機制，提供適合每位使用者的服務決策建議，並且考慮到其有限之資源達到相當程度的主動性建議。本論文將以人類的社會環境做為應用環境，提出一套能提供符合需求的服務建議之決策機制，而本論文的主要目的為：

- 對於一個應用領域中現有之服務加以分析，做功能上的分類，並使用語意網路(semantic web) 技術中的本體論(ontology)對各類服務做描述，了解各類服務的特性和內容，有助於提供適當之服務類別建議。
- 藉由智慧代理人(intelligent agent)的概念，來使每位使用者皆擁有自己的代理人，幫助其做服務上的決策，並且可以負責和其他使用者做互動，做為和外界的連繫者。
- 取得使用者的需求之後，以預應式(proactive)的方法，依垃圾桶決策模式(garbage can model)，來處理使用者的多個需求，主動給予適當的服務決策建議。
- 在決策的過程中，將使用者以往的使用經驗和結果做為考量，並記錄之，從中獲取不同個體 的偏好及興趣，達到因人而異的目標。
- 因資源是有限且會隨時間而變動，在決策時必需能夠使用最少的資源，而達到最大的效用，以免資源浪費。

本論文內容主要有四部分：第一部分為本研究之相關研究，第二部分為研究方法，第三部分研究方法之實驗設計與研究方法評估，第四部分為論文結論。

貳、相關文獻

一、 垃圾桶模式

傳統組織行為理論認為組織決策是理性的，然而實際上，受到許多情境因素影響，使得組織的決策過程及選擇行為具有某程度的模糊性，而使得組織無法作出理性的決策。這種組織決策過程所面臨的模糊情境，可稱之為組織的無政府狀況(Cohen 1972)，並歸納成三特性：

（1）問題之偏好(problematic preferences)，或可稱之為目標模糊，由於組織對各問題之偏好不清楚，更無法知道問題間的優先順序為何。

（2）不明的科技(unclear technology)，雖然組織總是在解決問題，但實際上卻沒有一套正規的解決問題方式。

（3）流動的參與者(fluid participation)，組織中的人是流動的，其所能提供的注意力和努力無法固定。

垃圾桶模式可用來解釋無政府狀態的決策過程，其中包含了四個獨立之要素：(1) 問題 (problem)：組織內外人所擔心之事 (2) 解決方案(solution)：人的某些產出，是早存在之事實，待時間到了才知是解決方案 (3)參與者(participant)：組織中的所有人 (4)選擇機會(choice)：某些特殊事件或場合，使得解決方案有可能受到注意。

垃圾桶模式決策理論即將選擇機會(choice opportunity)視為一個垃圾桶，垃圾桶當中容納了許多已存在之問題與解決方案，而這些問題和解決方案有可能是產生之初就被參與者給丟棄的，故稱為垃圾桶模式。參與者藉由參與選擇機會來解決環境所產生之各種問題，而決策則是上述四要素幾近隨機碰撞而產生的結果。

另外，每一個參與者對解決組織問題所花費的時間與努力可能不相同。決策之能量分配，代表其參與決策對於選擇機會和問題的注意力大小，並會隨著時間而改變。所有

參與者皆會對選擇機會提供能量，但會隨時間不同，使得對每個選擇機會提供之能量就不同。一個問題在一個時間內只能使用一個選擇機會，而每個選擇機會有一必須之能量總量，當某選擇機會之總量達到此必須總量時，即完成了一個決策。

換言之，每個選擇機會需要的有效能量(effective energy)，為進入此選擇機會的所有問題之必要能量(required energy)總合。而每個選擇機會可得之有效能量，為所有參與此選擇機會的參與者，其所貢獻之能量總合，如圖1所示。當選擇機會可得之有效能量，大於或等於總必要能量時，一個決策就產生了，在圖2-1-9中，只有第三個選擇機會的可得有效能量(=5)，是大於總必要能量(=3)。

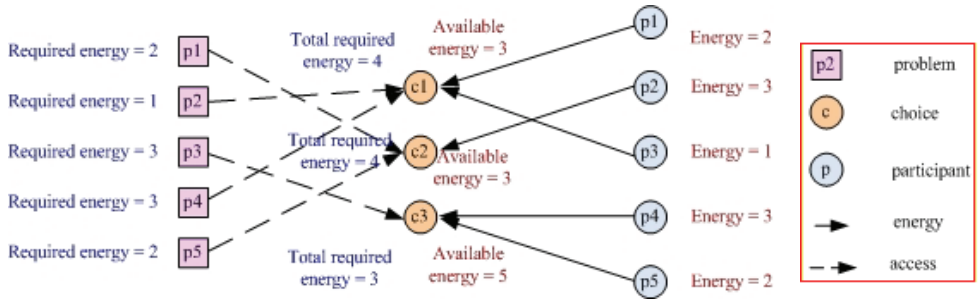

圖1：能量總和示意圖

在垃圾桶決策模式提出之後，陸續有學者投入後續之研究，垃圾桶模式已成功應用在教育(Clark & David 1980)、公眾事務(Sproull et al. 1978；Lavitt & Nass 1989)、政府政策(Kingdon 1984；Kingdon 1995)及軍事(Bromiley 1985)等領域，另外也有對原先之垃圾桶模提出簡化模式(Takahashi 1993；Takahashi 1997)，稱為簡單垃圾桶模式(Single Garbage Can Model)。

垃圾桶模式雖可解決無政府狀態的決策過程，本研究之決策情境是應用於社會環境，是希望能找出滿足人們多重需求的應用服務，因此垃圾桶模式產生了下列的問題和挑戰：

- 不適用於個人：在過去的研究中，垃圾桶模式是被應用於政治和教育領域中，並以整個組織的決策過程來分析，屬巨觀(macro view)角度。由於是巨觀去看組織，因此組織中的個人特質易被忽略。而在本研究的社會環境中，為了要找出符合使用者需求之服務，故每位使用者之特質是重要的影響要素，不可被忽略。使用者的特質，包括了個人的興趣偏好，以及過去的使用經驗等。由於每個人的特質不同，而其特質對決策有極大影響力，故不可以原先垃圾桶模式之巨觀角度來看待，必須改以微觀(micro view)的角度放到每一個體身上。
- 對能量(energy)的定義不完整：在垃圾桶模式決策過程中的一個重要因子，就是能量(energy)，會隨著時間的流逝而變動。然而在做決策的過程中，個體對問題所付出的不只是注意力，且能付出的又不同。在本研究之問題環境中，個體間所能付出的差異性更大，對使用者本身而言，可付出的有時間、金錢，或者是使用服務所需之相關軟、硬體設備等；對同樣有參與決策過程的其他人而言，例如使用者之親友，付

出的則是其過去使用經驗或對使用者之認知和了解等。因此，能量之定義，不足以用注意力大小來概括之。此外，某些能量的變動期間長，以決策所需花費的時間區段來看，可視為無變動。

為了克服並解決人類在社會環境中享受應用服務所遭遇到的問題，本研究擬設計出一具學習性的智慧代理人(intelligent agent)，並改良垃圾桶決策模式(garbage can model)的方法，建構出一套能運用在社會環境中，依不同個體的個別需求及喜好，替其決定最適合的服務決策建議。

二、智慧型代理人

隨著人類社會不斷地進步，新興科技之應用環境也愈趨於複雜，在管理大範圍之嵌入式軟體系統(embedded software system)也愈加有挑戰性。一般而言，軟體系統之設計，希望能設計出一可靠(reliable)、可維護(maintainable)以及可擴充的(extensible)系統(Kinny et al. 1996)，為因應此種需求，現今在設計應用系統上提出了代理人導向(agent-oriented)的觀念。

代理人(agent)為人工智慧領域中之專有名詞，代理人所處之環境為一會變化且為不確定(uncertain)的世界，代理人可意識環境的改變並做出適當動作，其具有可反應的(reactive)、自主的(autonomous)、內部自我激發(internally-motivated)之特性存在。

代理人導向系統目前已有許多的成功應用例子，其中一個常見的代理人架構為BDI，使用BDI架構之代理人即稱為BDI代理人(BDI agent)。在BDI架構中，將代理人視為具有某些人類心智上的態度-Beliefs、Desires、Intentions，可以完整地表達其可感應之事件、可執行之動作、其所具備之知識、其所採用之目標以及來源自intention之計畫。BDI代理人有三個model，分述如下：

- Belief model: 描述此代理人所處環境的所有資訊，其所持有之內部狀態以及可執行之動作。所有可能之belief或properties，皆被一belief set來描述之。而一個或多個的belief states-instances of the belief set，會被定義且用來描述成此代理人之初始狀態。
- Goal model(Desire model): 描述代理人可能採用之目標(goal)，以及可做反應的事件(event)。此model中包括了一goal set，來詳細指明所有的goal及event，以及一個或多個的goal states，其為用來描述代理人之初始狀態。
- Plan model: 描述代理人可能用來執行以達到目標之計畫(plan)。其中包括一plan set，來指明各別計畫之properties以及控制結構(control structure)。

要建造一個BDI架構之代理人，可分為二個步驟：

- Analyze the means of achieving the goals: 針對每個目標(goal)，分析要如何各別達到，並將其分解，分成不同之子目標(subgoal)及動作。接著分析子目標之順序，及在何種狀況之下，要執行何種子目標，還有什麼樣的錯誤必須被處理，並對產生能達到各個子目標之計畫。
- Build the beliefs of the system: 分析所有控制活動或動作之狀況，並將之分解。再針對子目標之計畫，分析其必要之input及output資料，並確定所有資料皆存在於belief

中，或是存在於較早執行的計畫中。

在尋找餐廳之應用實例(Lin et al. 2003)中，是將BDI架構應用於行動裝置上，當使用者想找餐廳時，可使用行動裝置做為輔助，來找尋在某範圍內適合的餐廳。在BDI架構中，belief為餐廳之資訊，包括位置、價格、型態等，以及使用者之profile，記錄使用的每次選擇結果；desire則為找到符合使用興趣之餐廳；intension則是依使用者的所有興趣，分成多個子計畫。

在BDI代理人中的成功應用，還有由Agent Oriented Software Pty. Ltd.(AOS)所開發出之JACK(Busetta et al. 1999)，其為一智慧代理人，是利用Java技術所發展出之multi-agent系統，並採用BDI架構，可提供一用於商業、工業及研究上的應用平台。

參、研究方法

人類未來將會生活在充滿e化服務的環境，在任何時間地點均能享受多種服務帶來的便利性，但人們在同一時間可能有多重需求，造成難以選擇適當應用服務的混亂情況。此種混亂情況恰符合無政府狀態的三特性：(1) 模糊的偏好：同一時間的多重需求，皆是當下極需被滿足之需要，並無特定的順序，在短時間內並無法知道該用何種偏好來解決 (2)不明的決策技術：多重需求環境下，有時在某一時間點就恰好有一需求可被解決，但究竟是用了何種決策技術並無法明確說出(3)流動的參與者：人在解決自身需求時，有時會求助他人之幫助，但每個人不見得隨時都有空。

然而垃圾桶模式本是以巨觀(macro view)之角度來看待整體環境，易忽略掉個人特性；而在決定服務的過程中，參與者所能貢獻的能量，不只是對每個服務的注意力，更有其他有形和無形的資源。因此本論文將以垃圾桶決策之理論為基礎並將此模式加以擴充，對能量給予完整定義，並使用增強式學習機制(reinforcement learning)以及BDI (believe、desire、intention)代理人技術，從微觀(micro view)的角度來看待問題本身，將個人特質融入到決策過程當中，以決定出適合不同個體之服務建議。而本論文之研究方式乃是一種Design Science 的方式。

Design Science是一種資訊系統的研究方式。它源自於engineering和science of artificial (Simon 1996)。它根本上是一種問題解決（problem solving）的方法。它的尋求創新乃是由想法定義, 實踐, 技術能力,及人工製品(artifact)中實現；對資訊系統的分析、設計、實施、管理, 和用途及如何能有效地和高效率地完成的創新亦從此人工製品中探討(Denning 1997; Tsichritzis 1997) 。這樣人工製品不是豁免於自然科學或行為社會科學相關的理論。相反地，它們的創作依靠是應用、測試、修改、和延伸整個現有研究的理論(Markus 2002; Walls 1992)。換言之，設計是包括過程與人工製品，此乃有別於其它倚靠統計性分析之其他研究方式（如empirical research等）。人工製品亦可以根據適當的分析度規 (evaluation metric) 之相關的量化指標來評估其功能性與準確性。

一、研究假設與系統架構

本論文是以垃圾桶決策模式做為基礎，再加上增強式學習(Sutton & Barto 1998)以及BDI代理人(Kinny et al. 1996)來做改良，發展一在有多重需求及資源的環境下，提供使用者服務建議的服務機制，如圖2所示。

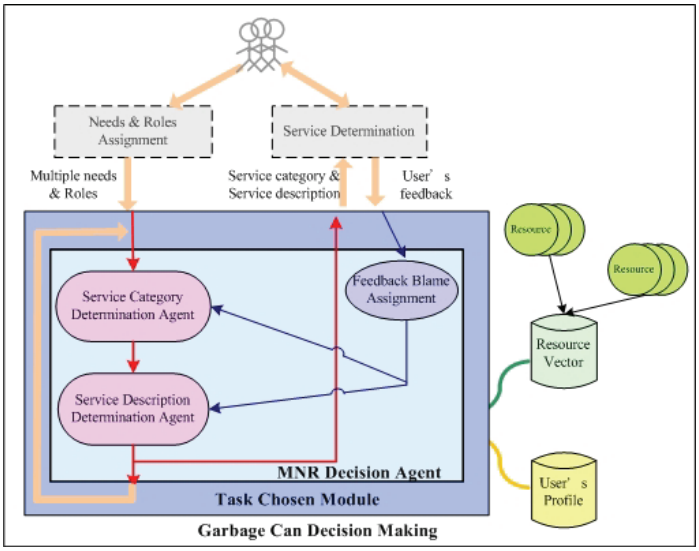

圖2：系統環境與架構

在本研究之環境架構中，為使系統運作順暢，必須符合幾項前提假設，如下列所述：(1) 使用者之需求可被偵測並分析(2)在前端有一模組，負責接收使用者需求並決定可能參與決策過程之各種角色(role)，在適當之時機指派需求及角色給決策代理人(3) 所有的e化應用服務均能加以分類(service category)，並利用本體論(ontology)來進行描述(service description) (4)在後端有一模組，接收到由決策代理人（MNR agent）產生之服務分類和描述後，決定一服務給使用者，並負責偵測和記錄個體的使用反應。

由於本研究採用垃圾桶決策模式做為基礎，故包含了原模式中的四個要素participant、problem、choice及solution，和一決策因子energy。但因本研究之應用領域為e化服務環境，因此做了些許調整，如表1所示。

表1：名詞對照

垃圾桶模式	本研究之決策機制
Participant	Role
Problem	Need
Choice	Service category
Solution	Service description
Energy	Resource

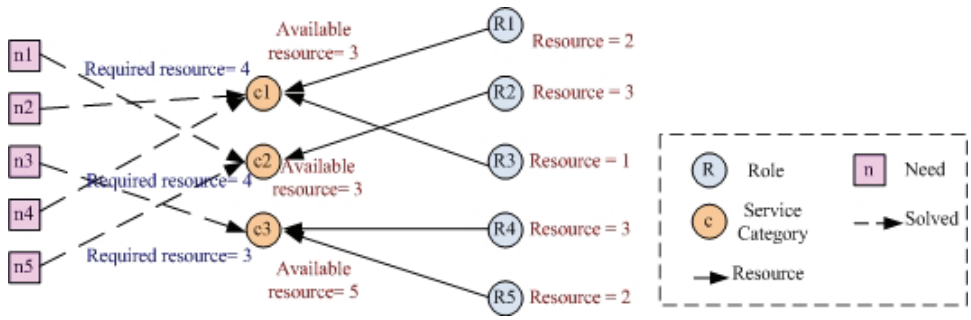

圖3：經調適後垃圾桶模式

經由表1對應，本研究調適後之垃圾桶決策模式可由圖3表示之，與原圖(圖1)相較，原圖右方圓形代表的是participant，在本圖則為role；原圖中間圓形的choice在本圖為service category；原圖左邊方形的problem在本圖則為need；原圖虛線箭頭代表單位時間下一problem可進入一choice，在本圖中，則代表在單位時間下，一個需求(need)只能被一服務分類(service category)解決；原圖實線箭頭代表單位時間下一participant只能進入一choice，在本圖則為一角色(role)只能參與一服務分類；原圖的energy在本研究為resource，故一服務分類可得資源量為所有參與其中的角色資源總和，由於在本研究中將必要資源量提升至服務分類，故當服務分類的可得資源量大於或等於必要資源量時，一決策則產生，即此服務分類可滿足某需求(例如在本圖中，c3的必要資源量為3，其可得資源量為R4和R5的資源總和5，故可得資源量大於必要資源量，一決策可產生，表示c3可滿足n3)，接著則可再進一步將最終的服務描述[1] (原模式之解決方案)決定出來。

本論文之MNR決策代理人，是採用集中式系統。每位使用者皆各自擁有一決策代理人，此決策代理人會學習使用者之偏好及興趣，負責和使用者以外的參與者進行溝通，協 調與其他模組之間的互動，以及整合系統之外的各種資源。在本環境架構中，有幾個重要元件， 茲分述如下：

- Task chosen module：在e化服務的環境當中，本研究是以垃圾桶模式來進行決策，在圖2的架構中的task chosen module實作垃圾桶模式中能量之觀念，以可得資源量找出可滿足之需求，再由MNR decision agent來決策出最終服務建議。
- Manifold of needs and resource decision agent（MNR decision agent ）：此決策代理人為本研究之核心部分，使用者的多重需求以及可能參與決策過程的角色為輸入資料；其輸出資料則為 服務類別(service category)和服務描述(service description)。其中包含了三個核心子模組。
  - service category determination agent：依據接收到之多重需求和角色，考量資源的有效利用，來決定出適合的服務類別。

[1] 垃圾桶模式之解決方案(solution) 即為服務分類下之服務描述(service description)。以老人居家照護為例，即可用此該領域的服務本體論(ontology)所定義之概念來表達此服務描述，進一步之解說請見3.3.1節。

- service description determination agent：依前者決定出的服務類別，進一步決定服務描述。
- feedback blame assignment：負責接收使用者的回饋(feedback)並指派給前二項子模組。

- user：為MNR decision agent之服務對象。
- needs & roles assignment：為本研究中之假設部分，負責把多重需求和角色傳送給MNR decision agent。
- service determination：為本研究另一部分假設。MNR decision agent完成一決策過程後，此模組將決定出最終之服務，並告知使用者，記錄使用者的反應並回傳給MNR decision agent。
- user's profile：記錄使用者的相關資訊（e.g., 時間、相關設備、金錢），用來做為決策時的依據。
- resource vector：負責整合外界之資源（e.g., 時間、相關設備、金錢、知識），同樣是做為決策時的依據。

在e化服務的應用環境當中，由於使用服務者是「人」，不同的個體雖然使用相同的服務，但極有可能出現不同的反應，又因為人有多重需求(Alderfer 1969)，使得找尋適合不同個體之服務愈加困難。為使本研究之MNR decision agent能確實符合使用者之喜好，決策過程非採單向式(意即觸發決策的因素發生後，完成決策過程，告知決策結果即結束)；而是整個決策過程是一循環，會將此次決策過程之結果以及使用者之回饋送回系統中，修改決策法則，以使下次決策過程更為精確；而未解決之需求，亦會送回至系統中。

二、Service Category Determination Agent

MNR decision agent為本研究之核心，當其自前端接收到多重需求以及角色後，首先是交由 service category determination agent 來決定出適當之 service category。在本研究中， category必須先被決定之後，才能再決定description。若category在最初即決定錯誤，使用者在收到服務建議時，並不會有想使用之動機，更無法從中得知個體的實際使用結果，故category之決定將會影響到整體的決策品質。為使決策過程中，能將複雜之資源變動以及人類的不同喜好考量進去，本研究在service category determination agent是採用增強式學習(reinforcement learning)故能針對不同個體的不同反應，一次又一次調整決策法則，學習到適合不同使用者的個別決策機制；並在使用喜好或環境有遽烈變動時，能迅速依使用者回饋(feedback)的改變，來修改決策機制。

在增強式學習的研究中，著重於自互動中學習，代理人為依據環境給予之信號(為reward)，來修改其行為。增強式學習在exploitation和exploration間有面臨選擇的挑戰(Sutton 1998)。Exploitation 是在已知的範圍內找尋最佳解；而exploration則是在已知的範圍之外找尋更好的解。若只著重於exploitation，那麼極有可能會陷入區域性解(local solution)，而得不到最好的解；若太注重exploration，雖可以找到全域性解(global

solution)，但會浪費過多的時間或運算能力，造成效率不佳。故必須要在 exploitation和exploration之間有所取捨，找其適當的平衡點。

Reward的計算方式，在增強式學習最初被提出時，reward計算方式是採用折減法(discounted)，之後又有學者（Schwartz 1993）提出平均法(average)，認為尋求最佳解的行為是一有限循環(非在無窮盡之範圍內尋找)，對reward採用平均法才適合。學者（Mahadevan 1996）對average reward用實驗來證明之，發現average reward對exploration有高敏感度，只要略提升exploration degree，即可達更好結果，而且average reward的學習成果較好。增強式學習演算法當中的AH-learning演算法，採用平均法並具有自動做到exploration的能力(Ok 1996)，故本研究選擇使用AH-learning演算法。

（一）決策過程運作方式

Service category determination agent之決策過程，乃結合二種演算法，一為加入垃圾決策模式中的能量概念之task-chosen演算法；一為AH-learning演算法。由於本研究之輸入值為多重需求，故將每個需求視為一個task，並以垃圾決策模式理論為基礎，選擇出一可被滿足之需求，即為task-chosen演算法，其運算方式如表2所示。

表2：task-chosen演算法

1. Choose one task (randomly from the set of tasks) to be the chosen task in the current state *i*.
2. Execute AH-learning algorithm.
3. Let *a* be the action that AH-learning algorithm decides, then check if *AR(t)* $\geqq$ *RR(a)*. Let *a* be the action AH-learning algorithm taken, then check if *AR(t)* $\geqq$ *RR(a)*. If not, ignore *a* (as it exceeds the total resource) and take step2 again (if step2 and step3 are executed less than three times).
4. Let another task be the chosen task in the current state *i* and repeat the algorithm.

在task-chosen 演算法中，current state *i* 與action *a* 為執行 AH-learning 演算法之要素，由task-chosen來判定選擇 *i* 並看是否 *a* 滿足資源需要。要執行本研究中之所有action，皆有一最少所需之資源量，以*RR(a)*來表示；而在當下之時間點，可用的資源量以*AR(t)*表示之。

演算法一開始，先由多個tasks(needs)中以亂數方式選擇一task來滿足current state *i*，接著執行AH-learning演算法，並會得一action *a*。當*a*被選擇出來後，此時加入垃圾桶模式之概念，必須考慮resource是否符合。在本研究中，service category為AH-learning中之action，且每一category 皆有一required resource之數量*(RR(i))*。當action *a*被產生之後，service category determination agent 會去resource vector確認現在可用之resource總量*(AR(t))*，若等於或大於required resource，則採用此action *a*；否則將忽略此action *a*，重新

採取AH-learning演算法來找出下一個可行之action *a*。重複此步驟，直到找到一action *a* 的required resource被滿足為止。若連續三次仍無法找到可滿足之action，則將其他task(也就是need)再亂數擇一來滿足當下之state，重新執行演算法，直到找到一 符合資源需求之action *a*。

AH-learning為自H-learning發展而來，而H-learning之基礎概念為馬可夫決策過程(簡稱為MDP)，並採average reward的觀點（Ok 1996），其定義如下表3所示：

表3：H-learning基本定義

Definitions
S: the set of states.
A: the set of actions.
U(i): the set of actions which are applicable in a state *i*.
$P_{i,j}(u)$: the probability of a given state *i* results in state *j*.
$r_i(u)$: a finite immediate reward for executing an action *u* in state *i* resulting in state *j*.
μ : a policy that map from states to actions.
$r^{\mu}(s_0,t)$: in time *t*, for a given starting state s_0, the reward for using policy μ .
$\rho^{\mu}(s_0)$; for a given starting state s_0, and policy μ, $\rho^{\mu}(s_0)=\lim_{t\to\infty}\frac{1}{t}E(r^{\mu}(s_0,t))$

如果一MDP(Markov Decision Process)是unichain，則所有的states在某policy之下，均可互相communicate(意指某二個states，在某policy之下，可從其中一個state到達另一個state)，但亦可能有少數的states無法彼此communicate。Unichain之MDP在長時間的情況之下，對任何policy而言，每單位時間之average reward乃獨立於任何起始狀態s_0，亦及無論起始狀態s_0為何，average reward，也就是$\rho^{\mu}(s_0)$均相同，稱之為policy μ的“gain”，以$\rho(\mu)$表示之。

H-learning延伸此一概念，認為雖然“gain($\rho(\mu)$)”獨立於起始狀態s_0，但在某時間點t之總期望reward則否。在某時間點t之總期望reward以$\rho(\mu)t+\epsilon_t(s)$表示之，其中$\epsilon_t(s)$為受時間點t之不同而改變的偏移量(offset)，稱之為bias of states，可解釋成“在起始狀態為s的情形下，長時間預期的優勢(advantage)，且大於平均reward值ρ(μ)”。而h(i)-h(j)表示，起始狀態分別為i和j相比較，兩者在長時間之總reward值的平均相對優勢(advantage)。使用某policy μ使得state *i*到state *j*，會放棄原來(若持續留在state *i*)之平均reward($\rho(\mu)$)，但得到一即時reward($r_i(\mu(i))$)，則state *i*與*j*之bias，會滿足$\rho(\mu)+h(i)=r_i(\mu(i))+\sum_{j=1}^{n}p_{i,j}(\mu(i))h(j)$。因此，發展出一H函式理論，定義如表4所示。

表4：H函式理論

For any unichain MDP, there exist a scalar ρ and a real-valued function h over S that satisfy the recurrence relation.

$$\forall_i \in S, h(i) = \max_{u \in U(i)} r_i(\mu) + \{\sum_{j=1}^{n} P_{i,j}(u)h(j)\} - \rho$$

Let μ^* represents the optimal policy that attains the maximum for each state i, and ρ is its gain.

在AH-learning演算法中，即利用H函式來做為選擇適當action的必要運算原理。而此原理可以表5 之AH-learning演算法來運算。在演算法中，*i*為當下之state；而*N(i,a)*為在state *i* 的狀態下， 會執行action *a* 的次數；*T(i,a,j)* 為在state *i* 的狀態下，會執行action *a* 並造成action *j* 的次數。 *GreedyActions*()為一陣列，用來儲存目前之greedy policy。演算法之初始值，將α設為1；其餘皆 設為0，而*GreedyActions*()之初始值則為所有可能的action。

表5：AH-learning演算法

1. Take a greedy action, i.e., an action $a \in U(i)$ that maximizes *R(i,a)* in the current state *i*. Let *k* be the resulting state, and r_{imm} be the immediate reward received.
2. $N(i,a) \leftarrow N(i,a)+1;\ T(i,a,k) \leftarrow T(i,a,k)+1$
3. $P_{i,k}(a) \leftarrow T(i,a,k) / N(i,a)$
4. $r_i(a) \leftarrow r_i(a) + (r_{imm} - r_i(a)) / N(i,a)$
5. $\rho \leftarrow (1-\alpha)\rho + \alpha(r_i(u) - h(i) + h(k))$ If $a \in GreedyActions(i)$, then
6. $\alpha \leftarrow \dfrac{\alpha}{\alpha+1}$
7. $R(i,a) \leftarrow r_i(a) + \sum_{j=1}^{n} \{p_{i,j}(a)h(j)\} - \rho$, where

   $h(j) = \max_u R(j,u)$
8. $i \leftarrow k$

此演算法為一迴圈，當有state輸入時，即觸發此演算法。演算法一開始，首先採取

exploration及greedy(利用效能函式*R(i,a)*，將在自動exploration部分介紹)方式，在當下之state *i*找尋適當之 action *a*。當action *a*被產生之後，*a*會傳至task-chosen演算法來判斷是否符合資源需求。當action *a* 被執行後，service category determination agent會接收到下一state *j*以及reward。接著依接收到之值，更新*N(i,a)*、*T(i,a,k)*；重新計算在state *i*之情況下，會產生state *k*之機率，*pi,k(a)*；並更新 *GreedyActions(i)*陣列；接著重新計算ρ值，並校調α值；使用新ρ值更新*R(i,a)*。當全部步驟完成之後，再將state *k*做為當下之state *i*，繼續重複以上步驟。

（二）Action Model 、Reward Model、State Features

在增強式學習演算法中，因其實際應用之領域不同，其states、actions及reward也隨之不同。為使對本研究的實際運作方式更易了解，將選擇一明確之e化服務情境，來對本研究做詳細解說。在此情境中，整體環境為一居家照護平台，服務對象為老人，此平台使用本研究之decision agent，協助老人在有多重需求時，能在複雜的多種e化服務中，提供適當之服務建議：

- Action

Action 為增強式學習代理人 對環境所做出的反應。在本研究中之 service category determination agent為自前端接到多重需求後，決定一service category，故service category即為本研究的actions。在本研究之service category，為一在相同領域中的統一標準，可能各各域會依不同法則對服務做分類，當分類有所新增或修改時，MNR agent亦會更新actions。以上述之老人居家照護情境為例，可用服務之function來做分類（Chang 2005），形成一階層。其分類之部分階層，如圖4所示。在垃圾桶決策模式的理論中，是認為所有的問題、參與者、選擇機會和解決方案均已存在於垃圾桶中。因此在階層式的分類中，以階層最底部之分類，來做為actions。

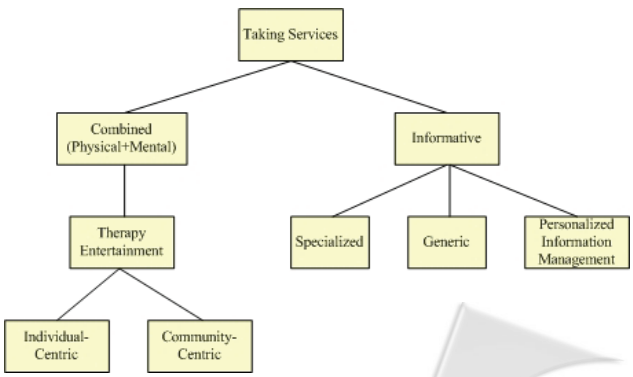

圖4：領域服務功能部份分類

- Reward Model

Reward指的是當增強式學習代理人決定一action，並執行之，環境會有所反應，因此會得到 一reward。在AH-learning演算法中，當在current state *i*之情況下，決定出一action

*a*後，會得到一即時的reward *rimm*，此*rimm*即為由reward model中的reward function所計算出來。在本研究中， 由於注重的是人，因此在提供服務建議後，會紀錄使用者的反應，做為回饋(feedback)成為計算reward的資訊之一。

表6：Reward function定義

Reward function

$$w_1 * R + w_2(w_{21} * D + w_{22} * U)$$

R： reward from resource utility
D： reward from user's decision time
U： reward from user's using time
w_1： weight of R
w_2： weight of $(D+U)$
w_{21}： weight of D
w_{22}： weight of U

此外，由於使用者要使用e化服務時會利用到資源，而資源是有限的，並會隨著時間點的不同而改變。為使資源能被有效利用，在本研究的reward model中，將資源的使用情況亦一同納 入考量。因此，在reward model中，可分為二大要素，一為資源之使用效率；一為使用者之反應，而計算出的值，為介於0～1之間。依據此二大要表，發展了如表6所示之reward function。在上表公式中，R指的是資源之使用效率，而(D+U)則為使用者之反應，並分別給予權重，預設皆為0.5。在使用者反應的部分，又可分為二種，一為D，為提供建議給使用者之後，使用者會決定是否要接受MNR decision agent的建議，此期間之決定時間和結果即為D；一為U，為若使用者接受建議，真正使用服務時的反應，反之，若使用者在提供建議時即拒絕，則此 U值為0。對D和U亦分別給予權重，但因使用情況有多種，故給予不同的權重，俟分別將R、D及U介紹完畢，再對不同情況做討論。

R

R為資源之使用效率。本研究期望能利用最少的資源，達到最大的效用，因此發展了如下表7所示之R計算方式，R值愈大表效率愈好。Rnow為目前可使用的資源總量，Ri是為執行action i (即建議的service category)所需之資源量。二者相減，即為所剩餘之可用資源量，再除以Rnow，看剩餘資源在目前可使用的資源量所佔之比例為何。採用之計算方式，是期望用少量的資源來滿足使用者，則可把多餘資源挪做他用，讓資源做最有效的利用。故R值愈大，表剩下的資源愈多，使用效率也愈好。

表7：R之定義

Definition of *R*

$$R = \frac{R_{now} - R_i}{R_{now}}$$

R_{now}： the amount of current available resource
R_i： the required resource of action *i*

D

D為使用者在決定是否要接受建議時反應。在此分為二種情況，一為使用者接受；一為使用者拒絕。由於決定結果的不同，造成reward有正負之值。D之計算公式如下表8所示。

表8：D之定義

Definition of *D*

$$D = \{\pm(1 - \frac{DT_E - DT_S}{E(DT)})\}/2 + 0.5 \quad \begin{cases} \text{If accept，} & \text{positive (+)} \\ \text{If reject，} & \text{negative (-)} \end{cases}$$

DT_S： suggestion provision time
DT_E： user's decision time
E(DT)： worst expect time

在公式中，DT_S為提供使用者建議的時間點；DT_E為使用者下決定的時間點；二者相減，則為做出決定所需花費的時間，*E(DT)*為預期使用者會花費的最長時間。對計算reward而言，值愈大通常代表reward愈佳。但對決定時間的長短而言，若考慮時間短，表示建議佳，不需多做考慮即可下決定；若建議不甚理想，則需多花時間來考慮。因此，為使D值愈大表示reward愈高，故用1來扣除$\frac{DT_E - DT_S}{E(DT)}$。若使用者接受建議時，考慮時間長，表建議佳；時間短，則表建議不甚理想。但在使用者最後拒絕的情況下，考慮時間短，表示建議極差，故馬上拒絕；若考慮時間長，則代表建議尚有可取之處，才需多花時間來思考。故拒絕之情況的reward恰與接受之情況相反，故當使用者拒絕時，需以負值來表示之，故D介於-1～1之間。為使D值如R與U值般，皆介於0～1之間，接著進行轉換步驟，將上述算出之值，先除以2再加上0.5。

U

U代表使用者實際使用時的反應。若使用服務時間愈長，表示服務愈令使用者滿意，故U值愈大愈好。計算U之公式如表9所示。其中UT_S為開始使用服務的時間點；UT_E則為終止服務的時間點；二者相減為使用服務的總時間長度；E(UT)為預期的服務使用時間長度。同樣地，將實際使用時間除以預期時間，以取得一介於1～0之間的值。

表9：U之定義

Definition of U

$$U = \frac{UT_E - UT_S}{E(UT)}$$

UT_S： start using time
UT_E： end using time
$E(UT)$： expected using time

由於使用者反應的reward，是由決定服務和使用服務的反應計算得來，故二者之權重需被考慮，但會受到情況之不同，權重之分配亦不同。使用的各種情況，如表10所示。

若情況為考慮時間短，使用時間長，則表示建議良好；若為考慮時間長，使用時間短，則表示建議不好。但若是二者時間皆長或皆短時(表中“?”)，則產生了矛盾情形。在此情形下，有可能是雖然對服務建議不滿意，但使用後卻發現不如自己想像中差，故使用時間長；亦有可能是對建議滿意，但真正的使用過程卻不如自己相像中好，故中斷了使用。

當有上述二情形發生時，若D與U之權重仍為0.5，則算出的最終R值會有誤差，無法真實反應出使用的偏好。舉例來說，系統提供的服務建議不符合使用偏好，預期的結果應為0.2；使用者在系統提供建議的當下認為此建議是自己想要的，則D值可能為0.8；但使用者在實際使用服務時，發現服務並不如預期中的好，則U值可能為0.2；在權重皆為0.5時，算出的結果為0.5，與預期的0.2差距過大，無法表示服務建議不適合使用者。為因應此情形之發生，本研究利用調整D和U之權重來加以解決。分配權重之原則，為當D與U之間的差異愈大時，則愈偏重U值，因使用者真正使用的情況，為較正確之reward。依此原則，當U值非0時，即使用者有接受服務建議的情況下，採用如表11所示之權重定義。

表10：權重考慮情況

	Using time		
Decision time (DT_S - DT_E)		High (better)	Low (worse)
	High (worse)	?	√
	Low (better)	√	?

表11：權重之定義

Definition of weight：

If |D-U| ≤ 0.3 and U≠0 Then $w_{21} = 0.5$，$w_{22} = 0.5$
If 0.3 < |D-U| < 0.7 and U≠0 Then $w_{21} = 0.3$，$w_{22} = 0.7$
If 0.7 ≤ |D-U| and U≠0 Then $w_{21} = 0.1$，$w_{22} = 0.9$

- State Features

在增強式學習中，環境的整體變化，即為各個state。而本研究之目的，為提供e化服務建議，以滿足使用者多重需求，故基本的states，即為使用者的多重需求。依據需求之不同，來學習正確的action 。除了有自前端接收之使用者的多重需求為states 外，亦有由service category determination agent自前感測的環境變化，來做為states。表12為本研究之state feature。其中NON-NEED及USE_SERVICE為service category determination agent可自行感測之state，而NEEDS則為自前端接收得來。依運用之領域不同，而有不同之states。NEEDS中將包含多個task，一個task即為一個need，當環境中之狀態為NEEDS時，將由task-chosen演算法來找出於current state *i*可被滿足之task。以前述之老人居家照護情境為例，NEEDS即老人之多重需求，如找伴之需求、外出資訊之需求、家人關心之需求等。

表12：State features

Stateee	Description
NEEDS	User's multiple needs which are received from front-end.
NON-NEED	Receiving nothing from front-end.
USE_SERVICE	User is using some services now.

三、Service Description Determination Agent

當MNRdecision agent自前端接收到多重需求與角色後，先交由service category determination agent決定出適當之service category，接著將決定出之service category傳至service description determination agent，由service description determination agent來進行下一步的決策過程。換言之，service description determination agent則是要決定出更詳細的服務建議，也就是service description。每次決定出的結果並非只有單筆資料，因為是要描述服務，故需用多筆資料才能完整描述出一服務。

在過去，一般的軟體技術可解決簡單問題、提升效率，但當環境愈趨於動態發展、問題更加複雜之後，傳統的資訊科技已無法滿足人的需要。之後，不斷有人尋求解決之道，當中一位學者(Bratman 1987)，將系統視為一有理性的代理人，並有如人心智上的看法。之後，逐漸架構成BDI agent，包含三大部分，Belief、Desire以及Intention，使得系統能容易地和使用者以及其他的代理人進行溝通，並且容易建置、維護及稽核（Rao 1995）。由於service description determination agent的決策環境複雜，必須考慮使用者之興趣及偏好，同時也須和其他的代理人做溝通。因此，本研究選擇採用BDI架構，來運用在service description determination agent 的決策過程中。

（一）Service description

本研究選擇採用了本體論(ontology)來對服務做描述，各領域皆會有各自對其服務進行描述的本體論(ontology)，使得決策環境中的所有元件，特別是對service description determination agent和service determination(後端之假設模組)而言，能以統一之方式來做溝

通。以老人居家照護為例，service description即可用此領域的服務本體論(ontology)所定義之概念表達，如圖5所示（Chang 2005）。若是一個提供天氣資訊的服務，則其service description為：

Scope = scope→heterogeneity→geographical matched

Type = type→personal information→weather

type→information→general

（二）BDI Architecture

Belief

Belief為BDI agent對其環境之認識，以及對行為上的邏輯理論等。在本研究中，決策環境中最重要之基本知識為service category及service description，必須擁有此兩樣知識，才有辦法決定出適當的服務描述，故本研究的belief為service category和service description。以老人居家照護為例，belief則為其服務階層分類，以及其服務本體論(ontology)，如圖5所示。

Desire

Desire為agent想達到之目標(goal)。本研究之最終目的，為提供適當之服務建議給使用者， 而在service description determination agent部分，是負責決定service description。故本研究BDI 架構中的desire，為依自service category determination category傳送之service category，使用者之偏好、興趣以及使用服務後的反應等，來決定出適當的service description。

Intention

Intention 為如何達到目標的方法，亦可稱為plan。在本研究中，此部分採用知識庫(knowledge base)，來將接收到之service category與service description對應(mapping)起來，較複雜(牽涉到使用者之喜好)之部分，則用使用者之回饋(feedback)來幫助決策。由於是使用知識庫(knowledge base)，故本研究之intention依應用領域之不同，內容也就隨之改變。 以老人居家照護為例，依belief中之服務階層分類及服務本體論(ontology)，可分為scope、source 以及type三大類，故將intention分為三個sub-plan，表13為部分之plan，而表中“＊”之符號，代表必須使用回饋(feedback)來輔助決策，將在下段做詳細說明。

表13：部分intention plan

Sub-plan 2　find suitable Service Source
1)　*if* community centric
then　Source = source　social network　relationship proximity
1.1)　*if*　home movie
then Source = source　electronic media　＊

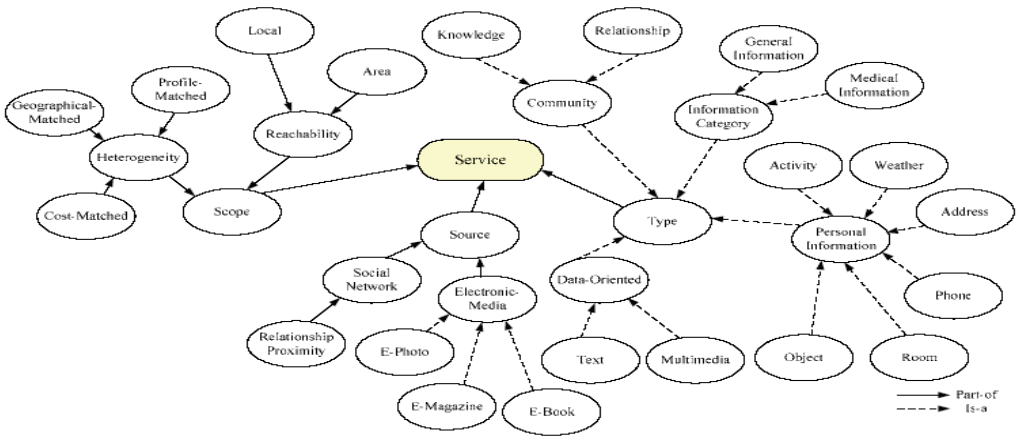

圖5：Ontology of service concept

（三）Feedback

在決定service description時，有些並無法用知識庫(knowledge base)可完成對應(mapping)。如圖3中的electronic-media部分，底下有e-photo、e-magazine、e-book，而此三種會因為使用者的不同，其所偏好之electronic-media就不同，因此必須將不同使用的偏好考慮進去，故採用了使用者之回饋(feedback)來做輔助。

若是必須考慮使用者偏好之description，均給予一係數w，w定義如表14所示，w為一介於0～1間的數值，第一次決策時，在同一分類之下的description中隨機選取，此時所有description之w初始值皆為0.5。當一description被決定後，會得到使用者之回饋(feedback)，即U值，定義如表14所示。接著採用簡單移動平均法(simple moving average)，將最近的5次之U取平均值(若不足5次則以目前的所有U值做平均)，取代原先之w。必須等在同一分類底下之所有description皆被決定過，且得到新的w值後，之後又須對應(mapping)到此分類時，則選取有最大w值的description。

表14：w之定義

Definition of w	
U： user's feedback	$w=\frac{1}{n}\sum_{i=1}^{n}U_i \quad (n\leqq 5)$
U_1： the first U received recently	
U_n： the n-th U received recently	

四、系統流程

本論文之目的，是希望當使用者有多重需求產生時，MNR決策代理人能依照使用者的偏好，在資源有限的情況之下，提供適當的服務建議給使用者。圖6為MNR決策代理人之系統運作流程，以UML中的順序圖來表示之。以下即以條例方式來對流程加以說明：

1. 當使用者有多重需求產生時，則會啟動MNR系統，此時不僅是使用者的多重需求會

傳遞給MNR系統，同時所有可能參與決策過程的角色(外部參與者)也會傳送給MNR系統。

2. 在MNR系統中，首先由Task-Chosen Module來處理使用者的多重需求，此時會先亂數選擇出單一需求，並傳送給AH-Learning Module。
3. AH-Learning Module會根據接收到的需求，使用AH-Learning演算法來選擇出服務類別，並回傳給Task-Chosen Module。
4. Task-Chosen Module收到服務類別後，會將服務類別和角色傳送至User Profile Module查詢現在可得的資源量。
5. User Profile Module回傳現在可得的資源量。
6. 若可得資源量無法大於或等於最低的服務類別資源限制時，則重覆3.，選擇另一服務類別；若連續三次找出的服務類別皆無法滿足資源限制，表示當下的資源無法滿足被選擇的需求，故重覆2.，選擇出另一需求；若可得資源量為大於或等於資源限制，則AH-Learning Module會將服務類別傳送給BDI Module。
7. BDI Module會依照接收到的服務類別，採用BDI架構來決定出服務描述。
8. BDI Module結束決策程序後，會啟動UI Module。
9. BDI Module將服務類別和服務描述傳遞給UI Module。
10. UI Module產生一介面，將決定出的服務建議呈現給使用者。
11. 當使用者做出決定時，即會將其反應回送給UI Module。
12. UI Module接收到使用者的反應後，會分別將使用者反應傳送給AH-Learning Module和BDI Module，使其能修改內部決策程序。

當1.至12.完成，MNR系統即完成一回合之決策，當使用者又有多重需求產生時，流程會自1.重新開始。MNR系統即藉由多次回合的流程循環，學習到使用者的偏好，才能決定出適當的服務建議。

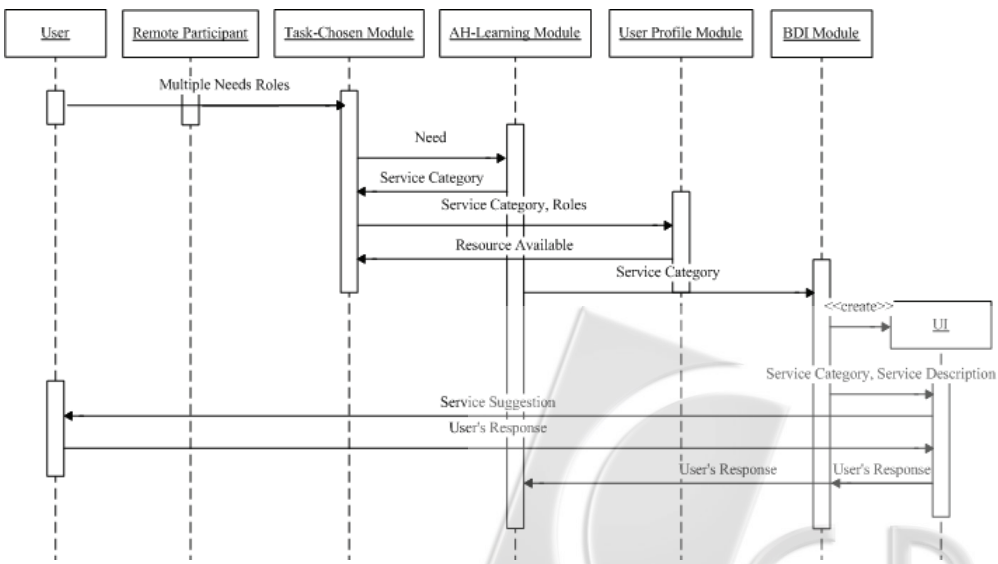

圖6：MNR系統順序圖

肆、研究方法評估

本研究之目的，是希望在有多樣e化服務的環境中，提供適當的服務建議給使用者，並且考量資源狀態和使用者偏好。為了驗證上述目的是否能達成，本研究採用模擬(simulation)的方式進行，分析是否針對不同使用者皆能符合其偏好，並且和其他方法相比較，評估是否能提供更適合的服務建議給使用者。

本研究是老人居家照護例子為模擬環境，在此環境中，已有許多存在的e化服務，當老人的多重需求產生時，照護平台會將多重需求交由本研究之決策代理人(MNR decision agent)，決定出適當的服務建議後再提供給使用者。本研究希望透過微觀與巨觀的兩層次，對實驗模擬出的結果加以分析，以回答下述問題：

（1）MNR模式在不同使用者的情況下是否能提供符合使用者偏好的服務建議？

（2）MNR模式是否能夠找出全域性解？

（3）MNR是否能在使用者資源有限的情況之下仍然能提供服務決策建議？

（4）MNR是否能提高服務的使用率？

而實驗模擬中，微觀面是分析三種典範型態(stereotype)的使用者是否皆能達到相同目的；而巨觀面則是將本研究方法，即以垃圾桶模式為基礎之MNR模式與其他二種決定服務的方式進行比較。而另外二種的決策方式：一是在不知道何種服務可滿足自身需求的情形下，隨機從已知的所有服務中做選擇，稱為random模式；另一種是雖然不知道何種服務最適合，但在所有服務中，有某幾項服務是較感興趣的(稱為候選服務)，因此在候選服務中隨機選擇，稱為greedy模式（此乃經常見到的系統模擬使用者之可能決策模式，其是在模擬環境中，則是從所有偏好值大的服務中亂數選擇）。因此，巨觀面將分析比較MNR、random和greedy三種決策模式的結果。

一、實驗設計

本研究是以前老人居家照護例子為模擬環境，在此環境中，已有許多存在的e化服務，當老人的多重需求產生時，照護平台會將多重需求交由本研究之決策代理人(MNR agent)，決定出適當的服務建議後再提供給使用者。以下針對各模擬部分加以說明：

- 使用者

為了要能驗證本研究之方法是否能針對不同使用者的偏好來提供適合每個人的服務建議。為了模擬出老人的偏好，本人對生活週遭大於65歲以之老人作觀察，發現有些老人足不出戶，極少與家人之外的人有互動；有些老人則是積極參與社交活動，喜歡與同年紀的人一同上課、出遊等；有些老人則是對新事物有高敏度，會使用電腦上網，甚至是使用P2P軟體。因此在模擬環境中，本研究由觀察結果歸納而得出三種典範型態(stereotype)的老人，描述如下：(1) Stereotype A – 活潑： 此種典範型態的老人性格外向，樂於與人交際，喜歡和外界接觸。(2) Stereotype B – 內向：此種典範型態的老人性格較內向，喜歡獨自活動，極少和外接觸。(3) Stereotype C – 創新：此種典範型態的老人除了性格外向之外，更樂於接觸新事物，喜歡嚐試以往未曾體驗過的事。此三種典範型態

(stereotype)的老人之服務使用偏好差異分佈廣，因此對驗證本研究方法是否可提適當的服務建議上具有代表性。

- 需求、服務分類與描述

由於本研究是以使用者的多重需求為輸入值，輸出值為服務的分類和描述，必須將需求、服務分類與服務描述給予具體化，於是以學者(Chang & Yuan 2005)提出之智慧型老人居家照護服務分類為例（圖7），並略為縮小範圍和修改，以符合本研究的模擬情境，分述如下：(1) 需求：在本研究的模擬情境中共有三類需求：family、creative life、social connection，在模擬過程中將會有不同的多重需求組合出現。(2) 服務分類：如前所述，本研究的服務分類為圖7服務分類階層的最底層，但本研究之目的為提高老人生活品質，較屬於心靈層面，故將telemedicine刪除；而在mental分類之下，選擇了較易聯想出相對服務的home movie為代表；informative分類之下則挑選了與外界資訊較相關的specialized和generic。經由簡化，本研究模擬情境共有七種服務分類，並搭配一種可能的服務 giving individual[2]（如部落格）、giving community（如線上學院）、home movie（如影片或照片播放）、taking individual（如個人益智遊戲）、taking community（如遠距聊天室）、info-specialized（如最新電影資訊）、info-generic（如天氣資訊）。(3) 服務描述：在本研究方法的BDI部分，使用知識庫(knowledge base)方式來進行描述的決定，但部分描述無法以簡單的if-then規則就可決定，必須考慮使用者偏好。在本研究的模擬情境中所採用的service ontology為對學者(Chang & Yuan 2005)所提出之ontology(圖8)略加修改，結果如下圖所示，圖中有顏色的描述表示要考慮使用者的偏好，且為本研究增加之部分。若服務分類為giving community，則必須依使用者偏好從static、active、mix中選擇；若服務分類為home movie，則必須依使用者偏好從photo、video中選擇。

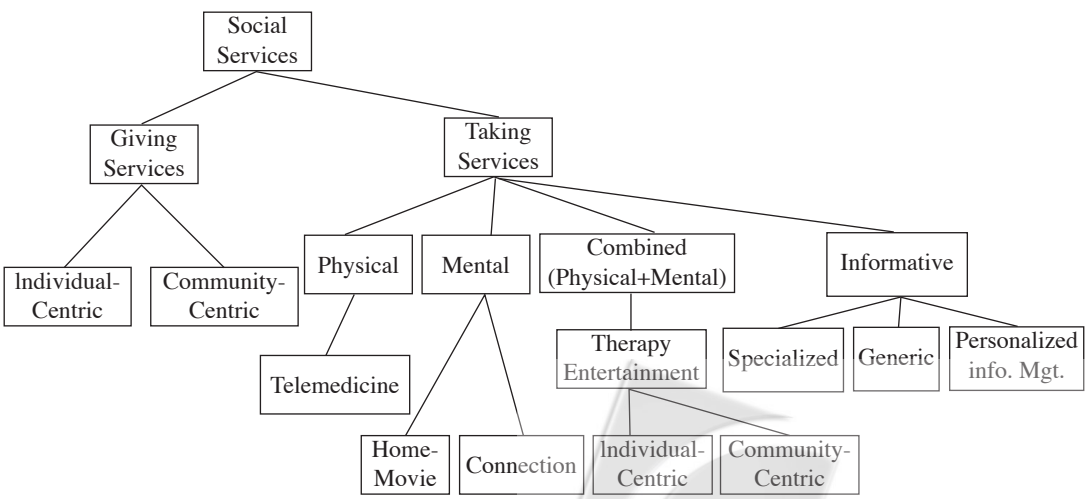

圖7：老人居家照護服務分類範例

2 “giving individual”乃是一個簡稱，其代表於圖7中屬於“Giving Services”服務分類下之“Individual-Centric”之子服務分類。其他六種服務分類之簡稱亦是類似。

圖 8：模擬設定-服務描述之service ontology

● 使用者偏好

依據上述之不同使用者典範型態、需求、服務分類和描述，可定義出使用者在不同需求情況下的偏好(見表15)，當本研究的決策代理人提供服務建議後，本研究將會依照使用者偏好來模擬使用者的反應(1為偏好低，5為偏好高)。在表15中，若以老人A為例，由於其個性屬外向活潑，當其需求為social connection時，會希望是能多與他人接觸，而非只是外界資訊的取得，故服務類別為taking-community的偏好指數為最高。而老人C是喜好較具有創新力的服務，因此在線上學院服務細部內容的偏好程度上，會較偏好有動、靜兩種混合型態的內容，其次為以動態為主的內容，故為「混>動>靜」。

● 資源

本研究在資源整合部分的模擬，將使用服務假設為需要付費，並且有可能是由老人的親友來共同支付，除此之外，某些服務是需要與他人有互動，因此要將對方參與者的時間也納入考量，而且對方參與者與老人也必須有相當程度的認識，才能在使用服務過中有良好互動。基於上述假設和考量，本研究將資源分為時間、金錢和熟識程度，而每個服務分類都有最低的資源限制(見表16)，必須要達到限制才可使用服務。

● 其他參與者

本研究中的角色，除了使用者本身之外，還有外部的參與者，即使用者的親戚或朋友，而這些外部參與者都有各自所擁有的資源(見表17)，如以老人A為例，由於個性屬外向，因此在朋友數的設定上比老人B多，而其家人恰為一家庭(老人A兒子、媳婦、孫子)，與老人A關係良好，其中家人1為其兒子，故熟識程度高，由於需投入工作，雖然擁有的金錢資源多，但是能夠花在老人A的時間就少。本研究會模擬在決策代理人的前端

假設部分，每次可能參與決策過程的外部參與者之組合皆不同，因此每次的可使用資源總量也不同。

表15：模擬設定-使用者偏好

	需求	giving-individual 部落格	giving-community 線上學院 (動,靜,混)	home-movie (video, photo)	taking-individual 個人益智遊戲
老人A (活潑)	family	2	3	4	1
	creative life	2	3(動>混>靜)	2(V>P)	1
	social connection	1	4	1	1
老人B (內向)	family	2	1	5	2
	creative life	2	1(靜>混>動)	3(V>P)	5
	social connection	1	1	3	4
老人C (創新)	family	3	4	5	1
	creative life	5	4(混>動>靜)	2(V>P)	1
	social connection	4	5	1	1

	需求	taking-community 遠距聊天室	info-specialized 最新電影	info-generic 天氣
老人A (活潑)	family	5	2	1
	creative life	4	5	1
	social connection	5	2	3
老人B (內向)	family	4	2	3
	creative life	2	2	4
	social connection	3	2	5
老人C (創新)	family	5	3	1
	creative life	3	3	2
	social connection	3	4	1

表16：模擬設定-最低資源需求

	時間	金錢	熟識程度
giving-individual	中	中	無
giving-community	高	高	中
home-movie	中	中	無
taking-individual	中	低	無
taking-community	高	中	高
info-specialized	低	中	無
info-generic	低	低	無

表17：模擬設定-外部參與者資源

		時間	金錢	熟識程度
老人A	朋友1	高	中	高
	朋友2	中	低	中
	朋友3	低	中	中
	家人1(工作)	低	高	高
	家人2(學生)	低	低	高
	家人3(家管)	中	中	中
老人B	朋友1	低	中	高
	朋友2	中	低	中
	家人1(工作)	低	高	中
	家人3(家管)	中	低	高
老人C	朋友1	中	中	高
	朋友2	中	低	中
	朋友3	低	高	中
	家人1(工作)	低	高	高
	家人3(工作)	低	高	中

二、實驗結果

本研究希望透過微觀與巨觀的兩層次，對實驗模擬出的結果加以分析，茲將結果說明如下：

實驗一、MNR模式是否能符合使用者偏好之評估

本實驗之目的，是在測試本研究所提出之MNR模式是否在e化服務環境當中，能夠提供適當的服務建議，而此服務建議要能符合不同使用者的喜好，並且比傳統決策模式更能滿足使用者。經由微觀面與巨觀面的實驗結果分析，針對三種不同典範型態使用者，在在評估MNR模式是否能符合使用者偏好時，不能只單從D與U值進行分析，還需分析整體reward值的實驗結果。MNR模式的平均D、U、reward值，趨勢皆為在學習初期快速上升，之後則趨於平緩，但仍在提升當中，其值均可到達0.5之上，整體平均reward值更是可接近於0.7（如圖9所示）。而在服務接受率上，三種典範型態使用者在第21回合之後，服務接受率均達到72%以上（如表18所示）。與傳統模式相較而言，傳統模式的平均reward值僅能達到0.4。綜合上述實驗結果，顯示MNR模式所提出之服務建議，比傳統模式更能滿足使用者（如圖10所示），且能針對不同使用者，來學習其不同的偏好。

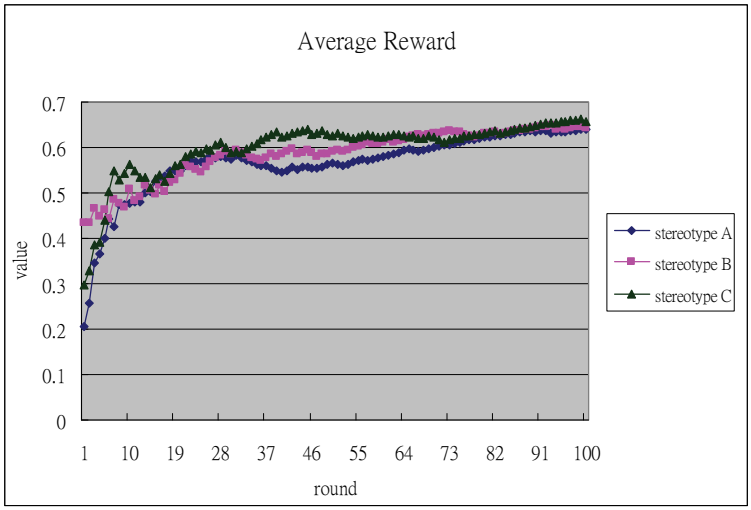

圖9：MNR模式實驗結果-average reward

表18：MNR模式下的使用者服務接受率

	Stereotype A	Stereotype B	Stereotype C
Before 20 round	0.35	0.4	0.55
After 20 round	0.775	0.7375	0.75

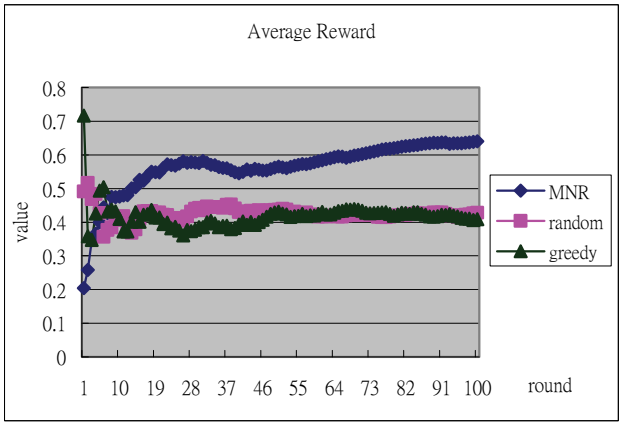

圖10：三種模式之average reward

實驗二、MNR模式是否能夠找出全域性解

本實驗之目的，是希望能藉由實驗分析的結果，探討MNR模式是否能有具備有explore的能力，經過幾回合的運作就會去找尋是否有其他更好的服務，避免陷於區域性最佳解中。綜合以上所述，本研究所提出之MNR決策模式確實能有效率地提供服務建議給使用者。

MNR模式的核心為AH-learning演算法，此演算法的一大特色為可達到自動explore的能力，不會只侷限在已知的action，會去嚐試其他的action，因此可避免產生區域性解(local solution)，若不考慮時間或運算成本，則能找到全域性解(global solution)。在評估MNR模式的效率時，可從MNR模式是否真正具備explore的能力來分析，若有explore的能力，提供的服務建議才是最適合使用者，才能有效地進行決策程序。圖11為Stereotype A典範型態使用者，採取MNR模式進行100回合實驗後的每回合reward結果。

本研究設計的reward值為介於0與1之間，若reward值若可到達0.5，則視為在平均水準之上，屬於「佳」的決策結果。圖11，顯示出每一回合所得到的reward，由於受到資源的限制，當無法滿足使用者偏好程度最高的服務類別之最低資源限制時，則會選擇偏好程度次高的服務類別，故所得之reward會略低，整體而言，每回合所得的reward均可達到0.8左右。從圖中亦可看出，經過幾回合的運作之後，會出現低於0.5的reward，若將此現象對應至詳細的實驗結果(表19)[3]，Stereotype A的需求為family時，已連續好幾回合的解均為homeMovie，到第90回合時，MNR模式會嚐試homeMovie之外的解，但選擇了infoGeneric後得到的reward僅0.51，較homeMovie要低，故下次決策又回到原來的解homeMovie。從此結果中可發現到MNR模式找出的解不會一直侷限在已知的最佳解中，而是經過幾回合後會嚐試其他的解，而其他的解確實無法得到更好的reward，因此下次運作仍會選擇已知的最佳解。從上述實驗結果分析中可得知，MNR模式確實具有自動explore之能力，可避免限於區域性解的情況中。

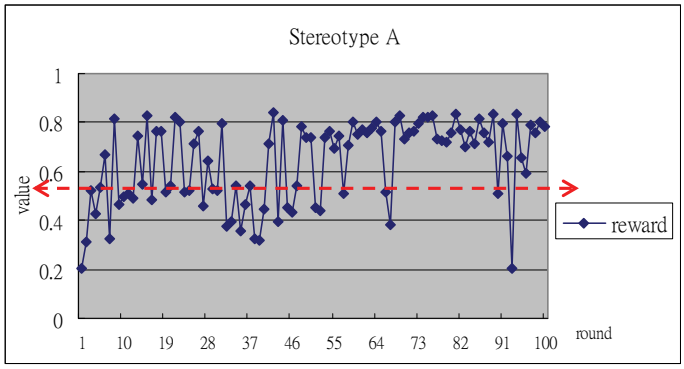

圖11：MNR模式實驗結果-reward (Stereotype A)

表19：整理後之部分實驗結果 (Stereotype A)

回合數	需求	服務分類	Reward
74	family	homeMovie	0.818581
75	family	homeMovie	0.81981
79	family	homeMovie	0.719273

[3] 表19為將原始實驗結果重新排序整理後的部分呈現。

回合數	需求	服務分類	Reward
80	family	homeMovie	0.755789
83	family	homeMovie	0.703111
86	family	homeMovie	0.813241
90	family	infoGeneric	0.510769
92	family	homeMovie	0.661

實驗三、資源有限環境下之效能評估

本實驗之目的是在評估MNR模式是否能處理資源不足的限制，期望能在資源有限情況下提供可使用的服務建議，而非提供無法使用的服務建議，造成可得資源完全被浪費。從實驗數據的分析中顯示，傳統決策模式(random、greedy)選擇出的服務極有可能會無法使用，造成資源浪費；但MNR模式則可完全避免此情形的發生，讓使用者皆可使用服務來滿足自身需求。總結來說，本研究提出之MNR模式，確實能解決資源有限的情況。在本實驗是使用巨觀面方式來進行，對同一使用者stereotype A分別以MNR、random、greedy模式模擬，每種模式皆連續進行100回合，並記錄每回合的結果。當可得資源量無法滿足服務分類的最低資源限制時，決定出的服務分類則無法使用，故所有reward值皆為0。

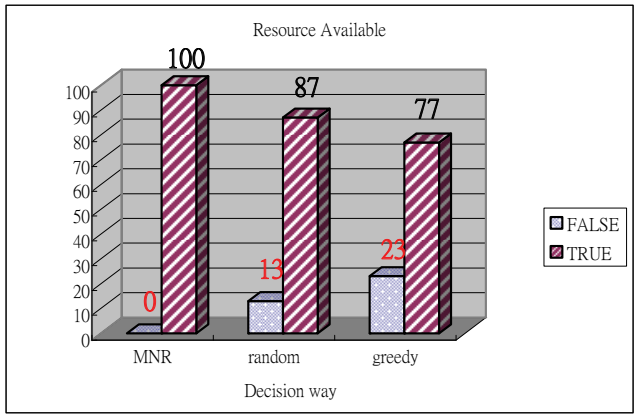

圖12：資源有限環境之下的實驗結果

圖12中呈現的是三種決策模式在連續100回合實驗中，可得資源量足夠與否的次數，點狀長條代表的是此回合可得資源量無法滿足選出服務類別的最低限制；斜線長條代表的是此回合可得資源量可滿足選出服務類別的最低限制，長條上之數字為總次數。由圖中結果可看出，random模式下資源不足的情形有13回合，greedy則是23回合，在這些回合當中，使用是無法使用服務，也就無法滿足任何需求；反觀MNR模式，其所選擇出之服務建議在每一回合皆能滿足最低的資源限制，只要使用者接受服務建議，就不會有無法使用服務的情形發生。

實驗四、MNR模式服務使用率之評估

當決策代理人提供服務建議後，使用者可選擇接受並使用服務，或是拒絕使用服務，而本研究設計之MNR模式的其中一個目的，即希望能提高服務的使用率，故設計此一實驗來與其他決策模式進行比較，分析MNR模式是否有較高之服務使用率。本模擬使用者部分，會依使用者的偏好設定而有不同的接受服務機率，當偏好指標愈高時，使用服務的機率就愈高。在此實驗中，為了比較MNR模式是否有較高的服務使用率，故採取巨觀面的實驗方式，對同一典範型態的使用者(stereotype A)，分別以MNR模式、random模式、greedy模式各別進行連續100回合的模擬，將每一回合的服務接受情形記錄下來並分析之。

由於MNR模式在第20回合前是學習初始期，自第21回合後決策結果開始趨於穩定，故以巨觀面分析三種模式的服務使用率時，會以第20回合做為分段點。分析結果如表20所示，而服務使用率的計算方式如下[4]：

(區段內有使用服務的總回合數) / (區段內總回合數)

服務使用率與服務接受率不同，服務接受率指的是決策代理人提供服務建議後，使用者接受服務的比率；服務使用率是指使用者真正有使用服務的比率。即使使用者接受服務建議，若資源不足會造成無法使用服務，因此服務接受率和服務使用率是不同的。由於MNR模式中不會有無法滿足資源限制的情形，故服務接受率與服務使用率是相同的。

表20：三種決策模式之服務使用率

	MNR	Random	Greedy
Before 20 round	0.35	0.3	0.6
After 20 round	0.775	0.325	0.5125

MNR代理人不僅是提供適當服務建議給使用者，來滿足其多重需求，更希望能藉由MNR決策過程的使用，使得e化服務的使用率能夠提升，本實驗之目的，即在評估MNR模式是否能提高服務使用率。由實驗數據分析後的結果得知，在過去傳統的決策模式下，服務使用率僅能分別達到30%與55%，而MNR模式在學習穩定之後，服務使用率則能達到77%。和傳統決策模式相較，MNR模式確實能提高服務的使用率。

三、討論

在充滿e化服務環境的情境之下，當人有多重需求時，會不知道那個需求應該先被滿足，也無法從眾多的e化服務中找出最適合自己的服務來使用，此時正處於無政府狀態(organized anarchy)之下，而MNR決策代理人可在得知使用者有多重需求產生時，即運

[4] 在Random模式下，在第1至第20回合的服務使用次數 = 6 ， 故服務使用率 = 6 / 20 = 0.3；在第21至100回合的服務使用次數 = 26 ， 故服務使用率 = 26 / 80 =0.325

用垃圾桶決策模式之概念，根據可被某服務分類解決的需求、可參與某服務分類的角色以及角色可提供的總資源量，當服務分類的可得資源量大於或等必要資源量時，則一決策產生，此部分決策程序在MNR中是以task-chosen演算法及增強式學習演算法來運作，考量了可得資源量及使用者偏好；接著再以BDI架構來決定出最終的solution，即服務描述；最終可將服務建議提供給使用者。而從上述所有實驗評估的結果可得知，MNR決策模式在20回合左右時，就可針對不同使用者皆符合其不同偏好，決策結果並不會陷入區域性解，故決策出的服務接受率為73%以上。若與其他決策模式相較(random、greedy)，MNR更能符合使用者偏好，且能減少資源不足使用的情形以及提高服務使用率。總結而言，MNR模式不僅是在無政府狀態的混亂情況下進行自動化決策，此一決策模式更具備有高效能和高效率，能夠在使用者有需求產生時，即自動啟動垃圾桶決策模式，快速地將最適當的服務建議提供給使用者，滿足其需求。

然而對於本研究決定服務描述所採用之BDI架構，由於其是採用知識庫(knowledge)及本體論(ontology)，具有領域相依性(domain-dependent)，在不同應用領域下內容就不同，而本論文實驗是以模擬之方式，服務的本體論以及知識庫範圍皆有限，故在BDI部分之實驗結果並不明顯，但並非表示BDI不會影響到決策的效能。BDI架構部分的驗證，需要更大量且完整的應用領域模擬資源，方適合進行更進一步的實驗評估，此部分為本研究的後續實驗目標。

伍、結論

數位e化生活，是人類社會的必經之路。嶄新的生活方式，相對也帶來新的問題與挑戰。 在有多重需求的情況下，人不知道該如何選擇適當的e化服務來滿足需求，此種情境符合了無政府狀態(organized anarchy)的三特性：模糊的偏好、不明的決策技術、流動的參與者，因此本論文提出一決策機制，稱為MNR決策模式，以垃圾桶決策模式之概念為基礎，將原模式中的能量概念擴展成資源，並採用二階段決策方式，第一階段的目的是決定出服務類別，先以task-chosen演算法從多重需求中擇一，接著再由增強式學習中的AH-Learning演算法，依使用者的喜好，找出適當的服務類別，接著task-chosen再確認當下可得資源量是否可滿足服務類別的最低資源限制；第二階段的目的則是決定出服務描述，採用BDI架構，其中intension的部分是使用知識庫(knowledge base)，故會隨著應用情境的不同而有差異。本研究期望每位使用者皆擁有一決策代理人，當使用者有多重需求產生時，決策代理人就會自動啟動MNR決策模式，提供適當的服務建議給使用者。

由實驗數據經分析評估後，MNR模式可得下列之結果：

- 能針對不同使用者，皆符合其偏好：每位使用者都有其自己的偏好，而MNR模式可以針對不同的使用者，學習其不同的偏好。若與傳統決策模式(random及greedy)相較，MNR模式明顯有較高的使用者回饋，以及高服務接受度，顯示MNR模式更能符合使用者偏好。
- 能夠找出能跳脫區域最佳解：無論使用對象是誰，雖然在學習初期的結果並不穩

定，但在第20回合左右MNR模式即可學習到使用者偏好，決策結果趨於穩定。此外，MNR模式不會只侷限在現行解中，而是會再去尋找是否有其他的最佳解。

- 在資源有限的情況下，可將資源有效利用：使用服務必須耗用掉一些資源，但資源並非無限的。另二種決策模式(random及greedy)，常出現資源量無法滿足決策出的服務分類資源限制，造成無法使用服務和資源浪費的情形。但在MNR模式中，當使用者的可得資源量較為不足時，仍會找出既滿足最低資源限制，又能符合使用者偏好的服務分類，故可將資源有效利用。
- 提高服務使用率：影響服務使用率的因素有二，一為資源量是否足夠；一為使用者接受或拒絕服務建議。在MNR模式中，決策時將使用者偏好及資源有限情形皆納入考量中，故服務的使用率可高達77%，明顯比傳統模式(random及greedy)要來的高。

本研究之未來可供後續發展之方向包括：

- 在本研究的實驗情境部分，是在服務分類的個數及需求種類的個數均固定不變的情形下，來驗證決策模式的效率，在未來研究中，可嘗試增加服務種數和需求個數，分析是否會影響決策效率，或是影響要達到穩定的平均reward所需的回合數。
- 本研究決策模式的提出，是希望能提供符合使用者偏好的偏好，而使用者的偏好並非長久不見，而會隨著時間的演進，可能受到外在環境的影響，使得偏好與過去不同。在本研究的模擬情境中，使用者的偏好為固定不變，未來可在連續回合的實驗中，改變使用者的偏好(例如在第30回合時更改偏好指數)，驗證MNR模式是否能處理使用者偏好改變的情形。
- 由於e化老人居家照護為一仍在發展中的產業，許多適合老人的e化服務尚未成熟和完整，故本研究模擬情境僅為初步實驗資料，雖然足夠驗證增強式學習部分的有效性，但在BDI部分僅知不會影響到增強式學習的效能，但BDI的有效仍待增加完整模擬資料後，方可進行完整驗證和分析。
- 本研究決策模式的啟動，是在使用者有多重需求產生，且不知該先滿足哪個需求時，由MNR代理人依據使用者偏好和資源可用量，建議某一服務來滿足使用者的單一需求。在未來，可試著擴展現行決策模式，在決策時可同時考慮多種服務和使用者的所有需求，讓提供出的服務建議為複合式服務(composite service，整合多種服務成單一服務)，並能同時滿足所有的需求。

參考文獻

1. Alderfer, C. P.“An Empirical test of a New theory of Human Needs,”*Organizational-Behavior-and-Human-Performance* (4:2), 1969, pp. 142-175.
2. Bratman, M. E.“Intention, Plans, and Practical Reason,”Harvard University Press, Cambridge, MA, 1987.
3. Bromiley, P.“Planning Systems in Large Organizations: Garbage Can Approach with Applications to Defense PPBS,”*Ambiguity and Command: Organizational Perspectives*

on Military Decision Making, 1985, pp. 120-139.

4. Busetta, P., Ronnquist, R., Hodgson, A., and Lucas, A. "JACK Intelligent Agents - Components for Intelligent Agents in Java," *AgentLink News Letter* (2), Jan. 1999 (available online at http://www.agent-software.com.au)
5. Chang, W. L., and Yuan, S. T. "Ambient iCare e-Services for Quality Aging: Framework and Roadmap," *7-th International IEEE Conference on E-Commerce Technology*, 2005, July, 19-22, Munich, Germany.
6. Clark, D. L. "New Perspectives on Planning in Educational Organizations," *Technical Report*, Far West Laboratory for Educational Research and Development, San Francisco, CA, USA, 1980.
7. Cohen, M., March, J., and Olson, J. "A Garbage Can Model of Organizational Choice," *Administrative science quarterly* (17), 1972, pp.1-25.
8. Denning, P. J. "A New Social Contract for Research," *Communications of the ACM* (40:2), 1997, pp.132-134.
9. Kingdon, J. W. "Agenda, Alternatives, and Public Policies," New York: Harper Collins, 1984.
10. Kingdon, J. W. "Agenda, Alternatives, and Public Policies 2nd ed.," New York: Harper Collins, 1995.
11. Kinny, D., Georgeff, M. P. and Rao, A. "A Methodology and Modelling Technique for System of BDI Agents," *Proceedings of the Seventh European Workshop on Modeling Autonomous Agents in a Multi-Agent World*, 1996, Eidhoven, The Netherlands.
12. Lavitt, B., and Nass, C. "The Lid on the Garbage Can: Institutional Constraints on Decision Making in the Technical Core of College-Text Publishers," *Administrative Science Quarterly*, June 1989, pp. 190-207.
13. Lin, D., Wiggen, T. P., and Jo, C. H. "A Restaurant Finder Using Belief-Desire-Intention Agent Model and Java Technology," *Computers and their Application*, 2003, pp. 404-407.
14. Mahadevan, S. "Average Reward Reinforcement Learning: Foundations, Algorithms, and Empirical Results," *Machine Learning* (22), 1996, pp.159-195.
15. Markus, M. L., Majchrzak, A., and Gasser, L. "A Design Theory for Systems that Support Emergent Knowledge Processes," *MIS Quarterly* (26:3), 2002, pp. 179-212.
16. Ok, D., and Tadepalli, P. "H-learning: A Reinforcement Learning Method for Optimizing Undiscounted Average Reward," *Technical Report*, 94-30-1, Department of Computer Science, Oregon State University, USA, 1994.
17. Ok, D., and Tadepalli, P. "Auto-Exploratory Average Reward Reinforcement Learning," *Proceedings of Thirteenth National Conference on Artificial Intelligence*, 1996, Portland, Oregon, USA.
18. Rao, A. S., and Georgeff, M. P. "BDI Agents: From Theory to Practice," *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, 1995, San

Francisco, USA.

19. Schwart, A. "A Reinforcement Learning method for Maximizing Undiscounted Rewards," *Proceedings of the Tenth Machine Learning Conference*, 1993, Aberdeen, Scotland.
20. Simon, H. A. "The Sciences of the Artificial," MIT Press, 1981.
21. Sproull, L. S. "Organizing an Anarchy: Belief, Bureaucracy, and Politics in the National Institute of Education," University of Illinois Press, 1978.
22. Sutton, R. S., and Barto, A. G. "Reinforcement Learning: An Introduction," MIT Press, Cambridge, MA, 1998.
23. Takahashi, K. "Decision theory in Organizations," Tokyo: Asakura Shoten. (in Japanese) , 1993.
24. Takahashi, K. "A Single Garbage Can Model and the Degree of Anarchy in Japanese Firms," *Human Relations* (50), 1997, pp. 91-108.
25. Tsichritzis, D. "The Dynamics of Innovation," *Beyond Calculation: The Next Fifty Years of Computing*, Copernicus, 1997, pp. 259-265.
26. Walls, J. G., Widmeyer, G. R., and El Sawy, O. A. "Building an Information System Design Theory for Vigilant EIS," *Information Systems Research* (3:1), 1992, pp. 36-59.

C.E.P.S.