

# 建構以知識本體為基礎之自動化 企業交易協商代理人：以採購電腦零件為例

## Building an Ontology-based Automated Business Transaction Negotiation Agent : Procuring Computer Components as an Example

邱英華

Yin-Wah Chiou

ywchiou@mail.nhu.edu.tw

陳志豪

Chih-Hau Chen

g4141009@mail2.nhu.edu.tw

南華大學資訊管理研究所

Department of Information Management Nan-Hua University

### 摘 要

隨著語意網 (Semantic Web) 技術的提倡，知識本體 (Ontology) 的研究與發展主要是著重在異質系統之間的知識分享與再利用。在智慧型代理人 (Intelligent Agent) 相關領域的研究方面，也隨著知識本體的興起，使得它更具有抽象智慧與進階的處理能力。目前，在企業交易代理人 (Business Transaction Agent) 的研究上大都著重於資料探勘的領域，而欠缺實用的功能性。為了解決此一問題，我們提出以知識本體與協商協定 (Negotiation Protocol) 的概念為基礎，建構出自動化企業交易協商代理人 (Automated Business Transaction Negotiation Agent, ABTNA)。另外，我們也以採購電腦零件為案例，來驗證 ABTNA 系統在實務上的應用。該系統可以促成買賣雙方代理人協商的自動化，因而縮減網路購物流程中資訊篩選與商品議價所耗費的時間與人力。

**關鍵詞：**語意網、知識本體、智慧型代理人、協商協定

## Abstract

As the semantic web technology has been promoted, the research and development of ontology is focused on the knowledge sharing and reusing between heterogeneous systems. Due to the arising of ontology, current researches of intelligent agent have achieved more abstract intelligence and advanced computability. In recent years, most researches about business transaction agent are emphasized on the area of data mining and they have lack of practical functionality. To solve this problem, we propose a concept based on the ontology and negotiation protocol to build an Automated Business Transaction Negotiation Agent (ABTNA). In addition, we use procuring computer components as an example to illustrate the real application of ABTNA System. This system can achieve automated negotiation between buyer agent and seller agent to reduce the time and human intervention for filtering product information and price negotiation.

**Keywords : Semantic Web, Ontology, Intelligent Agent, Negotiation Protocol.**

## 壹、導論

近年來，網路購物活動興盛。在面對資訊繁多與定義不一致的商品時，大量的資訊往往造成消費者個人必須花費時間與網路零售商作確認與過濾，且商家則因為消費者資訊收集不足或是人工回應緩慢而錯失商機。另一方面，消費者所考量的購物決策因素漸趨多樣化，例如：價錢、售後服務、品牌形象、商品品質、交易運送方式、商家信譽、網站的信譽、隱私權與安全性等等，此亦增加消費者做決策時的負擔。因此，我們將傳統企業網路交易模式與採購代理人所遭遇的問題歸納如下：(1) 消費者對於龐大網路資訊的蒐集與篩選往往有所缺失與不足；(2) 消費者得付出較多的時間和成本來進行自身與眾多商家之間的分析與協商；(3) 因為許多限制因素(例如：時間與人力)，致使消費者無法獲得最佳的採購方案，而商家亦無法取得最大的商機；(4) 傳統採購代理人只涉及網路資訊上的擷取與篩選，並未涉及到與商家協商的程序。

為了解決上述的種種問題，我們以知識本體 (Ontology) 為基礎建構自動化企業交易協商代理人 (Automated Business Transaction Negotiation Agent, ABTNA)。透過將領域知識 (Domain Knowledge) 有效地解析成系統可解讀的協商目標 (Negotiation Object) 知識本體，我們更進一步結合協商協定 (Negotiation Protocol) 知識本體來改進傳統代理人協商機制之不足。規則描述在自動化協商機制中，可以用來表達協商協定[1]。自動化協商的主要目的是為了在快速且有系統的協商空間中，獲得較為理想的協商結果[18]；而代理人程式 (Software Agent) 的主要目的則是為求能夠在所處的環境中擁有活動的自主性 (Autonomy) 地活動，以達成其設定的目標。

以消費者的角度來看，ABTNA 系統即為協助使用者篩選產品資訊與採購決策考量，並進一步取代消費者購買網路商品上的手動篩選與協商議價流程。以零售商角度來看，ABTNA 系統則是根據每一商家本身的彈性策略下取代以人力接洽方式來爭

取商機。為了驗證 ABTNA 系統之實用性，我們以採購個人電腦零件為案例來說明該系統所具有的持續性與自動化。在建置 ABTNA 系統時，我們針對買賣雙方分別提出了自動化協商採購代理人 (Automated Negotiation Procurement Agent, ANPA) 與自動化協商拍賣代理人 (Automated Negotiation Auction Agent, ANAA)。ANPA 在實務上可應用於相關網路購物前端產品資訊搜尋平台或產品推薦系統做擴充整合，而 ANAA 在實務上則可運用於網路賣場中賦予零售商自動協商拍賣產品的能力。

本文所開發的 ABTNA 系統，其核心概念是以知識本體來呈現協商目標與協商協定，並進一步結合由智慧型代理人協會 (Foundation for Intelligent Physical Agents, FIPA, <http://www.fipa.org>) 所制定的代理人互動規則與代理人私有策略來達成模擬人類世界買賣雙方協商議價的過程。在協商目標知識本體方面，關於採購個人電腦零件所使用的領域本體 (Domain Ontology)，傳統的 PCDIY (Personal Computer Do It Yourself) 電腦零件知識本體所呈現的知識只在於硬體的分類概念。為了解決此一問題，我們將傳統的知識本體改善之後，本文提出了個人電腦協商採購 (Negotiation for Procuring Personal Computer, NPPC) 知識本體得以實現運用至網路買賣採購等商業交易行為。

此外，在協商協定知識本體方面，我們所提出的改良型協商協定知識本體，主要是以 FIPA 所制定的代理人互動協定規範為基礎並進一步改善林建良[1] 所提出的協商協定知識本體。透過將協商協定知識與協商目標知識的分離，不僅可以促使整體系統更具彈性與降低維護複雜度的優點，並且透過知識本體的運作可以達到更具抽象、豐富的自動化協商能力。而在以

往代理人程式的建構私有策略上所運用的協商策略樣板只針對布林值與整數值作協商判斷參數。為了解決此一問題，我們提出加入字串值等參數運用模式，它可以使得代理人程式在自動化協商過程時擁有更豐富和精確的策略資訊。

關於本文其餘的組織結構如下，我們首先在第 2 節的文獻探討裡介紹 ABTNA 系統所涉及的相關技術，包含：知識本體、自動化協商、代理人程式、與網路購物決策影響因素。其次，我們在第 3 節描述系統分析與設計，包含：建構協商目標知識本體、建構協商協定知識本體、互動協定的描述、協商策略的設計、以及完整的系統架構。之後，我們在第 4 節進行系統實作；使用的工具包含：Protégé (知識本體開發平台)、SWRL (Semantic Web Rule Language, 語意網規則語言)、JTP (Java Theorem Prover)、Jess (Java Expert System Shell, 專家系統推理引擎) 與 JADE (Java Agent Development Framework, Java 多重代理人開發平台)。最後，在第 5 節，我們總結本文的重點並探討未來的研究與發展方向。

## 貳、文獻探討

在本節，我們將介紹本文所開發的 ABTNA 系統所涉及的相關背景技術，包含：知識本體 (Ontology)、自動化協商 (Automated Negotiation)、代理人程式 (Software Agent)、與網路購物決策影響因素。

### 一、知識本體

近幾年來，隨著語意網 (Semantic Web) 概念的興起，其最終目標是著重在電腦能夠利用資料的分類、階層與關係來瞭解資料所代表的語意，而不是單純的做資料呈現的工作。語意網的創新關鍵在於以知識

表達 (Knowledge Representation) 代替檔名，做為網路間訊息交換檢索的依據。因此，未來在語意網上，資源與知識檢索的對象是定義完善，且可由電腦或所謂的代理人程式 (Software Agent) 判讀的知識本體，而知識本體的研究與應用已成為實現語意網的關鍵。

當企業交易代理人彼此間進行溝通協商時，透過知識本體完善的定義規範，代理人之間可以有著共通的領域知識認知，藉此代理人可以擁有更具複雜的交談能力。在知識本體中，概念 (Concept) 是一個特定領域概念的共通詞彙，透過知識本體中專門領域的知識與關係，可以進行不同代理人之間的引導、轉換、轉化與組織。所以，利用知識本體來溝通是降低模糊度最好的方式 [3]。

Guarino [12] 提到知識本體的概念為將真實世界視為許多不同的領域 (domain) 所組成，在不同的領域中，有著不同的領域知識，而根據知識本體的內涵有三種分類：(1) 術語知識本體 (Terminological Ontologies)、(2) 資訊知識本體 (Information Ontologies)、(3) 知識塑模知識本體 (Knowledge Modeling Ontologies)。若是以主要用途來分類則有：上層知識本體 (Top-level Ontology)、領域知識本體 (Domain Ontology) 與作業知識本體 (Task Ontology) 以及應用知識本體 (Application Ontology)。Fensel et al. [9] 認為從代理人程式的角度來看，知識本體是一個理論，探討那些實體能夠存在於智慧型代理人的記憶中。此外，Fensel [10] 亦進一步補充提到知識本體是指在某領域下之共同的認知，目的是為了以明確且正規的定義來表達共同概念化之事物。

## 二、自動化協商

協商可以視為是人類行為中主要用來

解決彼此利益衝突的過程，它亦可以視為所有參與協商者共同搜尋最佳結果的過程 [26]。然而，由於協商過程中變數複雜，維度過高，而造成可能的協商空間 (Negotiation Space) 亦相當寬廣。因此，人類在複雜的多議題協商環境中，時常無法達到最佳的協商結果。為了解決此問題，自動化協商將有助於在協商空間中，較快速且有系統的搜尋較為理想的協商結果 [23]。在自動化協商的研究中，Jennings et al. [15] 提到可以將其分為三個主要的議題：

1. **協商協定** (Negotiation Protocol)：它主要是規範代理人程式之間的互動規則，亦可以視為一系列事前滿足與事後回應的規則。協商協定是用來管理協商參與者彼此之間的互動方式，其中包含可允許的參與者類型、協商狀態、導致協商狀態改變的事件以及特定狀態中的合法行動。
2. **協商目標** (Negotiation Object)：它主要是描述必須達成協定的議題；協商可能是單一議題或是多議題的協商，而協商議題的內容可以是固定的或是可臨時更動的，這些都會對協商的複雜度造成很大的影響。
3. **決策制定模型** (Decision Making Model；亦即，協商策略)：它主要是根據個別協商參與者私有的發揮空間來制定協商策略 (Negotiation Strategies)；而協商策略是由協商協定來決定，在既定的協商協定中，協商參與者根據本身的策略來達到各自的目標。

此外，Lomuscio et al. [18] 認為自動化協商系統主要包含：協商協定及協商策略兩個基本元件。在協商協定方面，Lei et al. [17] 提到協商協定描述了互動的規則。有許多方法可被用來設計協商協定，而協商協定限制了整體的協商流程。在協商策略方面，Oliver [21] 提到協商策略表達了由

個別議價所組成的議價次序。Jennings et al. [15] 認為協商策略為一個複雜的運算過程，在每回合的協商過程中，代理人程式的效益評估功能需大量的計算，以提出最佳的議價計畫。而 Lei et al. [17] 則提到協商策略決定特定狀態下的行為活動，並且可驅使協商狀態的改變。

### 三、代理人程式

近幾年來，代理人程式(Software Agent)不斷地被發展與廣泛地應用。在代理人程式的研究中，許多文獻對代理人程式所提出的定義都不盡相同。Maes [19] 提到代理人程式是一個接受委託的程式，它與傳統程式不同之處在於代理人程式具有的特性包含：個人化(Personalized)、不斷地執行(Continuously Running)、半自主性(Semi-Autonomous)。

此外，代理人程式亦被廣泛地運用在許多資訊處理與管理的工作上。Etzioni and Weld [8]認為一個代理人程式必須具備下列特性：擬人化(Personality)、持續性(Temporal Continuity)、自主性(Autonomy)、溝通能力(Communication Ability)、調適(Adaptability)與機動性(Mobility)。Fitzmaurice [11] 提到代理人程式設計的理念為目標導向，只要指示一個「目標」給代理人程式，透過代理人程式內建的規則與智慧，不須教導其何時與如何去作，代理人程式即可完成交付之任務。Woodridge and Jennings [29] 則認為一個智慧型代理人應具備下列特性與能力：(1)反應力(Reactivity) 智慧型代理人能夠認知到環境的變化，並採取適當的行動以因應環境的改變；(2)主動能力(Proactiveness) 智慧型代理人能夠以採取符合自己的設計目標的活動，來顯示其行為是以目標為導向(Goal-Directed Behavior)；(3)社交能力(Social Ability) 智慧型代理人能夠與其

他的代理人進行溝通，以達到其設計的目標。

另外，多代理人系統 (Multi-Agent System)是由許多獨立的代理人程式所組成，整個系統呈現分散式的型態，而系統效能則取決於代理人程式的組織架構和協商機制之設計。Jennings et al. [16] 認為多代理人系統是由一組透過溝通、協調與競爭方式來解決單一問題之系統所組成。Weiss [28] 更進一步指出：假使多代理人在一個有限資源的環境中，代理人之間便可以透過協調的方式將資源做有效的配置，以達到個體的利益或整體的目標；此時必須使代理人之間有相同的溝通語言，並且對於溝通的事物要有共同的認知以及代理人的互動協定。

### 四、網路購物決策影響因素

近幾年來，隨著網路的普及，網路拍賣購物行為亦逐漸成熟興盛。目前各類型網路拍賣平台琳瑯滿目，對於消費者而言，隨著網路拍賣購物的出現，不僅打破了時間與地點的限制；對於商家而言，網路拍賣市場的低門檻進入條件與巨大商機更吸引無數的商家投入。面對網路上所販賣的眾多商品，消費者對於在網路上採購的決策考量因素與商家對於消費者所重視的資訊呈現與服務項目亦有別於傳統的商務。

Turban et al. [27]在消費者採購決策制定流程中指出商家可操控的變數包含：價錢、商標、多樣化、宣傳、產品品質、產品可得性、客製化。林政瑩[2]指出影響拍賣網站購物意願的因素有八項：商品品質可靠、商品種類及價格、交易安全、購物流程便利、溝通管道便利、購物成本、購物規則、網站提供資訊豐富化。此外，在消費者進行購物決策的思考上會產生所謂的考慮集合(Consideration Set)[25]，它是指

當消費者在做購買決策時，可能會考慮購買的產品品牌的集合。因此，從消費者所產生的考慮集合概念上，可以進一步看出產品品牌的形象對於消費者採購決策上有著重大的影響。在面對眾多品牌的競爭下，良好的品牌形象與信譽會造就消費者

優先考慮的認知經驗。而對於消費者的產品知識，可以解釋為消費者對產品的熟悉度、產品的專業知識與產品的經驗[6, 7, 22, 24]。產品的熟悉度是指：自我評估對產品知識瞭解自信的程度；而自我認知的產品熟悉度亦會影響購物決策的結果。

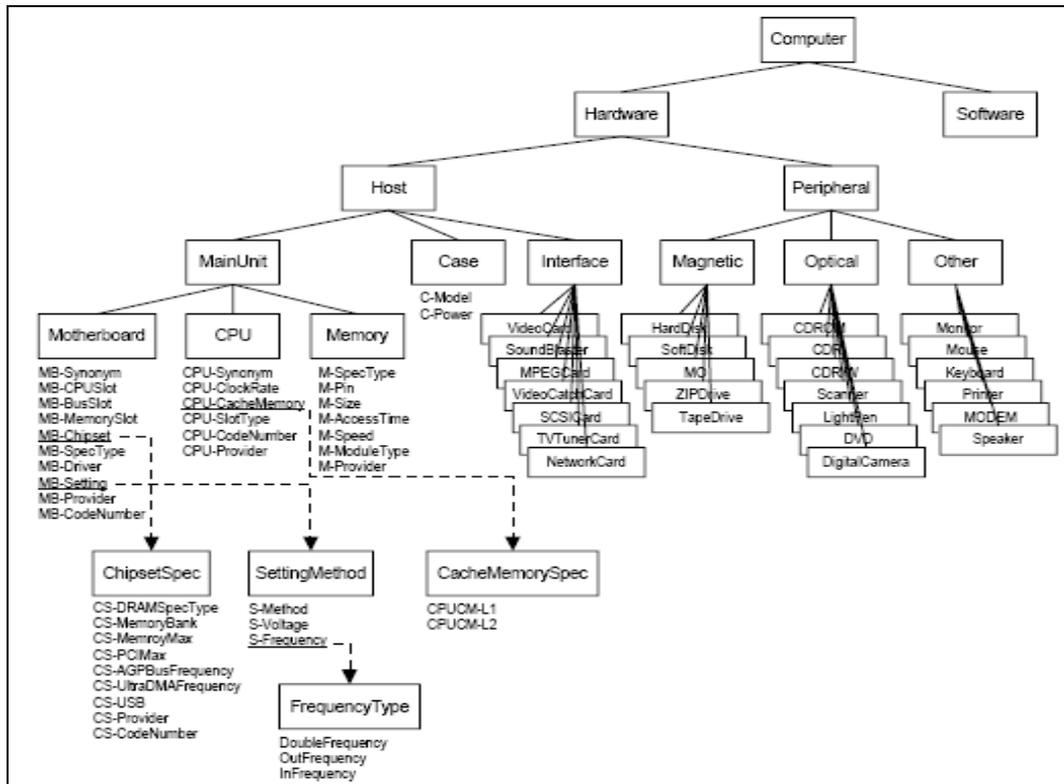


圖 1 傳統 PCDIY 知識本體的基本架構

(資料來源：楊勝源 等人[4])

在上述文獻所呈現的各類網路購物決策因素中，我們歸納出應用於 ABTNA 系統中消費者於協商活動中可以選擇的採購考量策略，包含：價錢策略、產品效能策略、品牌策略、保固策略、交易運送策略。

### 參、ABTNA 系統分析與設計

在本節，我們說明以知識本體為基礎所建構的自動化企業交易協商代理人 (Automated Business Transaction Negotiation Agent, ABTNA) 的系統架構與運作流程。我們首先在 3.1 節呈現 ABTNA

系統所使用的協商目標知識本體，而該系統所採用的協商協定知識本體則在 3.2 節有完整的描述。另外，互動協定的描述呈現在 3.3 節，而針對協商策略的設計則在 3.4 節裡亦有詳盡的說明。之後，我們在 3.5 節呈現出完整的 ABTNA 系統的代理人自動化協商環境與系統架構。

### 一、建構協商目標知識本體

圖 1 顯示出楊勝源[4]所提出的傳統 PCDIY 知識本體的基本架構，它明確定義出電腦零組件的組成概念與各項零組件的

內容屬性。利用該特性結合代理人程式的運用，代理人程式可以清楚地辨識各類零組件的組成關係與相關零組件的知識來源。然而，傳統 PCDIY 知識本體的提出只在於重視電腦零組件概念的分類，而不具有任何協商概念屬性。例如：在 CPU(中央處理器)此類別中，其相關屬性(如：Synonym、ClockRate、CacheMemory)皆在解釋 CPU 本身名稱、規格與製造商。

| Motherboard                                                                                                                                     | CPU                                                                                                                                                                                                                                                 | Memory                                                                                                                                                                                                               | VideoCard                                                                                                                                                                                                                                                                       |
|-------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MB-Synonym<br>MB-CPLSlot<br>MB-HusSlot<br>MB-MemorySlot<br>MB-Chipset<br>MB-SpecType<br>MB-Driver<br>MB-Setting<br>MB-CodeNumber<br>MB-Provider | CPU-Synonym<br>CPU-ClockRate<br>CPU-CacheMemory<br>CPU-SlotType<br>CPU-CodeNumber<br>CPU-Provider<br>CPU-Buyer<br>CPU-Seller<br>CPU-Warranty<br>CPU-CommonPrice<br>CPU-MinPrice<br>CPU-SuggestionPrice<br>CPU-TransportType<br>CPU-AcceptRangePrice | M-SpecType<br>M-PinM-Size<br>M-AccessTime<br>M-Speed<br>M-ModuleType<br>M-Provider<br>M-Buyer<br>M-Seller<br>M-Warranty<br>M-CommonPrice<br>M-MinPrice<br>M-SuggestionPrice<br>M-TransportType<br>M-AcceptRangePrice | VC-Synonym<br>VC-SlotType<br>VC-GPUClockRate<br>VC-GPUCodeNumber<br>VC-MemorySpeed<br>VC-MemorySize<br>VC-DRAMSpecType<br>VC-Provider<br>VC-Buyer<br>VC-Seller<br>VC-Warranty<br>VC-CommonPrice<br>VC-MinPrice<br>VC-SuggestionPrice<br>VC-TransportType<br>VC-AcceptRangePrice |

圖 2 NPPC 知識本體的屬性

相對於這些屬性中，我們發覺其中並無交易協商中消費者所注重的協商概念屬性。

為了解決上述的問題，我們提出在代理人協商過程中所採用的協商目標知識本體，除了以傳統 PCDIY 知識本體為基礎外，在每一零組件概念中皆增加數項協商概念屬性。透過我們對於傳統 PCDIY 知識本體的改良之後，不僅使其知識本體具有記載產品交易明細的屬性來做為協商的根據，亦促成自動化協商概念的實現。而改良後的協商目標知識本體，我們將其命名

為個人電腦協商採購 (Negotiation for Procuring Personal Computer, NPPC)知識本體。亦即，NPPC 知識本體是整合了傳統 PCDIY 知識本體與協商概念。

在 NPPC 知識本體中，針對每一零組件概念所增加的數項協商概念屬性上，我們以 CPU(中央處理器)、Motherboard(主機板)、Memory(主記憶體)、VideoCard(顯示卡)為範例來描述其各零組件的完整屬性。在圖 2 所示的 NPPC 知識本體的屬性，我們以實線筐起的屬性為傳統 PCDIY 本身所具備的零組件特性，而虛線筐起的

屬性則是我們參考網路購物決策影響因素之相關文獻後所制定設計的協商概念屬性，其主要應用於協商過程中代理人程式互動協商的依據。針對圖 2 中的虛線筐起的八項屬性，我們分別說明如下：

1. Buyer (消費者)：此項屬性主要用於描述參與協商之消費者名稱且亦對零售商公開的資訊。
2. Seller (零售商)：此項屬性主要用於描述參與協商之零售商名稱且亦對消費者公開的資訊。
3. Warranty (保固期)：此項屬性主要用於描述該零件廠商所提供的保固時間，此屬性亦為對消費者公開的資訊。
4. CommonPrice (一般價格)：此項屬性主要用於零售商制定零件的起始價格，對於消費者而言則是用於協商初始階段時願意購買的價格支出。
5. MinPrice (底限價格)：此項屬性是對於零售商制定該零件最低可接受賣出的價格，以利進行議價協商，對於消費者而言則用於制定最高可出價購買的底限支出。
6. AcceptRangePrice (可接受再議範圍價格)：此項屬性為消費者與零售商雙方所具有，它主要用於雙方協商議價在尚未達到最佳結果時，決定是否可以進一步協商議價的彈性範圍價格。
7. SuggestionPrice (建議價格)：此項屬性為消費者與零售商雙方所具有，它主要用於雙方首次協商未達到雙方可接受的最佳結論，但是仍屬於可接受再議範圍價格時，則代理人程式可以採用建議價格運用於下一回合的協商議價程序。
8. TransportType (運送型態)：此項屬性主要用於制定針對該零件零售商所提供之運送方式，此屬性亦為對消費者公開的資訊。

## 二、建構協商協定知識本體

在多代理人(Multi-Agent)的互動過程之中，代理人程式本身需要了解其他代理人程式在溝通的訊息中所表達的意思。在本文，我們將代理人程式的知識分成協商目標以及協商協定兩個部份。在協商目標的部份，我們以 NPPC 知識本體做為代理人程式之間共同認知；而協商協定中的互動協定雖然包含規則化的知識，但需要以規則語言來描述。

在開放式的環境之下，代理人程式需要與其他代理人程式之間進行規則知識的分享，而此時也就會發生代理人程式之間語意認知的問題。因此，有些文獻認為需要將規則語言加入概念的描述，由這些概念做為規則描述的來源[13, 14]。協商協定知識本體的主要作用在於：透過建構協商協定知識本體將規則知識轉換成具有抽象概念與一致性的描述就能有效地解決代理人程式之間語意認知的問題。我們舉例說明如下：假設 A 代理人程式透過協商協定知識本體得知在流程規則下他應該等待接受 B 代理人的 Inform 型態訊息；此時，除非 B 代理人傳送 Inform 型態訊息給 A，否則 A 會視 B 為違反溝通行為規定。

本文所建構的協商協定知識本體除了參考 FIPA 相關規範之外，尚且改良了林建良[1] 所提出的協商協定知識本體。在他所提出的協商協定知識本體中，我們修正了多餘無效的屬性並且新增協商發起者與協商參與者的概念，其主要目的是考量在多代理人環境中，多重協商活動裡不同代理人程式代表不同協商角色上所產生的協商活動與該對應的代理人程式辨識的問題。

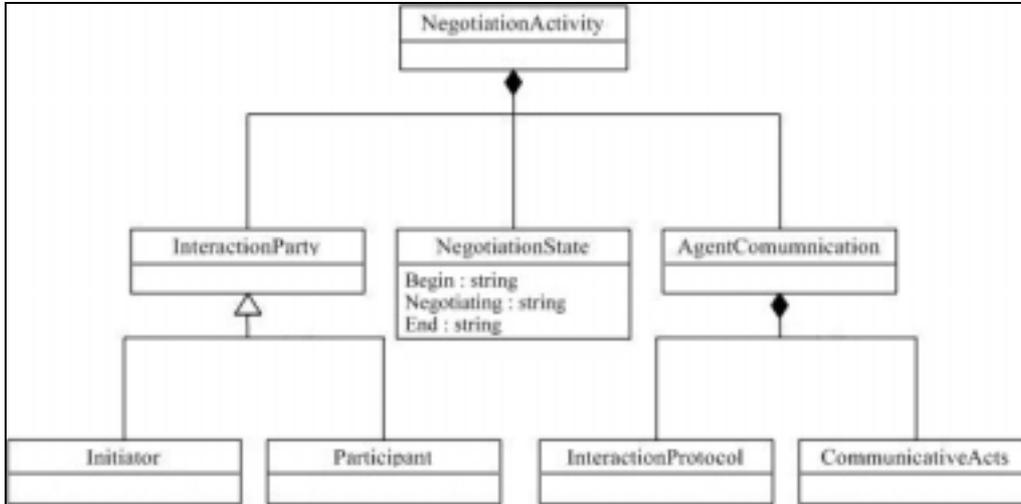


圖 3 改良型協商協定知識本體基本架構

另外，我們分離了其中與協商目標知識本體相關的類別，這是為了達成協商目標與協商協定彼此獨立的概念，並且保持維護知識本體的效率性與複雜度的考量。在圖 3，我們呈現出改良型協商協定知識本體的基本架構，它包含下列八個主要的類別：

1. NegotiationActivity (協商活動)：整個協商活動包含參與協商互動的參與方、代理人溝通、以及協商狀態。藉由 hasInteractionParty (擁有協商互動參與方)、hasAgentCommunication (擁有代理人溝通)與 hasNegotiationState (擁有協商狀態) 表達 InteractionParty、AgentCommunication 與 NegotiationState 三個概念的關係。
2. Initiator (協商發起者)：在整個協商過程中，代表協商互動參與方中的協商發起者角色，此類別當中可記載特定的協商發起者 ID 與相關細節資訊，藉此可達到篩選、限制特定協商發起者的目的。
3. Participant (協商參與者)：在整個協商

過程中，代表協商互動參與方中的協商參與者角色，此類別當中可記載特定的協商參與者 ID 與相關細節資訊，藉此可達到篩選、限制特定協商參與者的目的。

4. InteractionParty (協商互動參與方)：整個協商過程中所有參與協商的代理人程式所具有的身份；藉由 hasInitiator (擁有協商發起者)與 hasParticipant (擁有協商參與者)來表達擔任 Initiator 的自動化協商採購代理人程式和擔任 Participant 的自動化協商拍賣代理人程式。
5. InteractionProtocol (互動協定)：存在於代理人程式之間的互動協定，主要為表達代理人程式之間互動的流程規範。
6. CommunicativeActs (溝通行為)：當不同的代理人程式之間在交換訊息時，CommunicativeActs 為該訊息所代表的行為。
7. AgentCommunication (代理人溝通)：代理人程式協商溝通包含代理人之間

的互動協定與協商過程中的事件與行動；藉由 hasInteractionProtocol (擁有互動協定) 可以表達 InteractionProtocol 的概念，而 hasPreEvent (擁有事前滿足事件)與 hasPostAction (擁有事後反應行動)則表達同樣以 CommunicationActs 來表達行為的事前滿足事件與事後反應行動。

8. NegotiationState (協商狀態): 在協商進行的狀態，我們以 Begin、Negotiating 與 End 分別來表示協商開始、協商進行中以及協商結束這三項狀態。

### 三、互動協定的描述

在傳統代理人程式的互動溝通上，以往並沒有將溝通流程與代理人程式加以區

分，造成程式過於複雜而導致維護困難且缺乏彈性溝通的能力。有鑒於此，FIPA 在制定代理人程式的互動流程規範上，致力於統一代理人程式的溝通協定，並且促進代理人程式之間能夠擁有更具廣泛的溝通議題與抽象化的交談能力。

在 FIPA 所制定的互動協定 (Interaction Protocol) 中，每一個協定所表達的是一個互動流程，它是由雙方代理人程式在協商互動過程中每一個階段裡所能採取的行為所組成，而組成互動協定的行為則是根據 FIPA 所制定的溝通行為 (Communicative Acts)。在 FIPA 所發表的互動協定中，我們以 Request Interaction Protocol (請求互動協定) 為例來解釋互動流程。

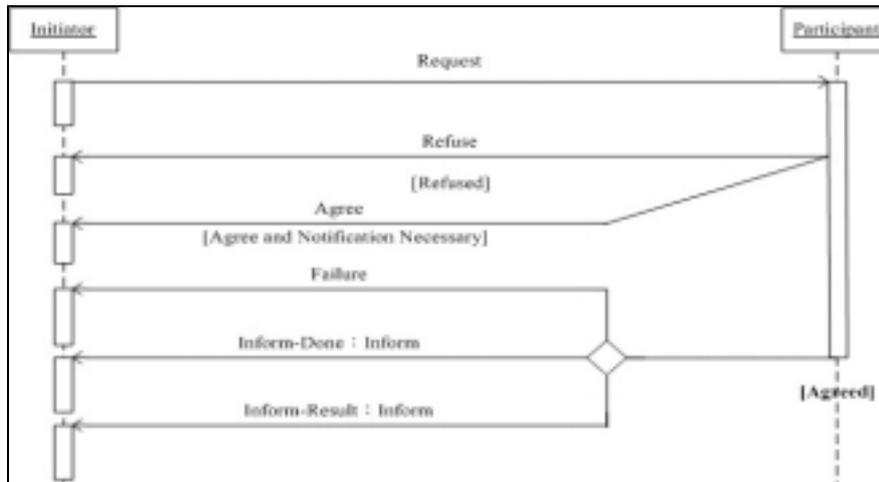


圖 4 FIPA Request Interaction Protocol

(資料來源：<http://www.fipa.org>)

如圖 4 所示，Request Interaction Protocol 是應用於 Initiator (協商發起者) 請求 Participant (協商參與者) 啟動特定協商作業的互動程序，整體流程我們說明如下：

1. 當 Participant 在收到 Initiator 所送出的請求 (Request) 的訊息後依內部考量

因素來決定是否發出拒絕 (Refuse) 或是同意 (Agree) 的訊息。

2. 當 Initiator 接收到拒絕訊息時則代表請求互動作業結束且無任何請求被接受；同意訊息則進一步視代理人程式是否存在於特定環境中，因為相關環

境條件因素會形成特定的同意訊息型態,亦可稱為必要性通知 (Notification Necessary)訊息。

舉例來說,如果 Initiator 要求 Participant 於特定時間內完成相關活動,而且此要求被 Participant 接受後,Participant 可於執行作業前先行回覆同意訊息給予 Initiator,然後等待作業完畢後再回覆給 Initiator 相關的作業結果,而回覆的作業結果則有下列三種不同的回覆訊息:(1)Failure (失敗) 表示此 Request (請求)作業失敗,並回覆失敗訊息給予 Initiator;(2)Inform-Done (回覆完成) 表示此作業成功完成但只回覆成功通知訊息給予 Initiator;(3)Inform-Result (回覆結果) 表示此作業成功完成,除了回覆成功通知訊息,也回覆作業結果給予

Initiator。

不同互動協定的流程彼此間有所差異,透過以規則語言表達出來之後,代理人程式即可藉由互動協定推論引擎 (Interaction Protocol Reasoning Engine)來推論出協商當時每一個階段所可以採取的行動有哪些,而代理人程式再根據協商目標的內容來判斷要採取哪一個行動,也就是決定所發送的訊息是哪種行動 (Communicative Act)。在圖 4,我們將代理人程式在其中的每一個階段可以採取的行動以及其判斷的依據,以協商協定知識本體中所制定的概念及屬性來描述互動協定。因篇幅所限,在表 1,我們只能呈現出部份 Request Interaction Protocol 結合多代理人程式 (Request Interaction Protocol

表 1 RIPMA 設計概念的案例

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 協定內部溝通行為-1                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| If (NegotiationActivity.hasInteractionParty.hasInitiator = ID)<br>And (NegotiationActivity.hasNegotiationState = Begin)<br>And (NegotiationActivity.hasAgentCommunication.hasInteractionProtocol = RequestInteractionProtocol)<br>Then (NegotiationActivity.hasAgentCommunication.hasPostAction = Request)                                                                                                                                                         |
| 協定內部溝通行為-2                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| If (NegotiationActivity.hasInteractionParty.hasParticipant = ID)<br>And (NegotiationActivity.hasNegotiationState = Negotiating)<br>And (NegotiationActivity.hasAgentCommunication.hasPreEvent = Request)<br>And (NegotiationActivity.hasAgentCommunication.hasInteractionProtocol = RequestInteractionProtocol)<br>Then (NegotiationActivity.hasAgentCommunication.hasPostAction = Refuse)<br>Or (NegotiationActivity.hasAgentCommunication.hasPostAction = Agree) |

with Multi-Agent, RIPMA)協商環境的設計概念。我們對於 RIPMA 互動流程中的兩個協定內部溝通行為之設計概念說明如下：

- 協定內部溝通行為-1：當互動協定為 Request Interaction Protocol 且協商狀態為開始時,具有特定 ID 的 Initiator (協商發起者)可向具有特定 ID 名稱的

Participant (協商參與者)請求特定活動。

- 協定內部溝通行為-2：當互動協定為 Request Interaction Protocol 且協商狀態為協商進行時，當具有特定 ID 的 Participant 察覺到請求時，該 Participant 可向此具有特定 ID 名稱的 Initiator 回應拒絕或是同意請求的訊息。

#### 四、協商策略的設計

代理人程式在建構與評估對方議價的策略過程時，可透過邏輯關係的表達得以發展出各種可能的競價策略。因此，在協商策略的設計上，我們依據 Meister[20]所提出的協商議題類別為基礎，並且改善楊紹明[5]所提出的協商策略樣版類型。傳統的協商策略樣版類型只以布林值與整數值作策略推論判斷，雖然有利於系統程式快

速且明確的邏輯判斷，但面對真實商品的複雜明細記載卻明顯缺乏相關決策資訊。因此，我們加入以字串值的呈現方式來達到更具周全的協商策略設計，並且透過協商目標與協商協定知識本體的結合運用，這不僅讓代理人程式於協商過程中所重視的協商議題能夠針對價錢來進行策略規劃，更能以字串比對的方式來進行商家、品牌、運送方式與產品細節的策略規劃。

另外，協商策略樣版類型又可分為三種策略意向，分別為堅持議題、非重要議題與讓步還價議題。堅持議題是我方所堅持己見的議題項目，例如：我方對於該次協商中的某些協商議題，表示欲堅持我方意見的態度時，這些議題就可稱之為堅持議題。

表 2 堅持議題樣版(單一議題)

| if         | Item(x)屬於協商目標知識本體描述範圍 |                     |                                             |
|------------|-----------------------|---------------------|---------------------------------------------|
| antecedent | 單一議題                  |                     |                                             |
|            | 議題一                   | 偏好表達-布林值            | Value(y) { ≡, ≠ } 我方數值                      |
|            |                       | 偏好表達-整數值            | Value(y) { ≡, ≠, ≤, ≥, <min,...,max> } 我方數值 |
|            |                       | 偏好表達-字串值            | String(z) { ≡, ≠ } 我方意見                     |
| then       |                       |                     |                                             |
| consequent | 單一議題                  |                     |                                             |
|            | 議題一                   | 接受對方所提出的意見或數值       |                                             |
| else       |                       |                     |                                             |
| consequent | 單一議題                  |                     |                                             |
|            | 議題一                   | 以我方意見或數值為退還待議訊息回傳對方 |                                             |

(資料來源：修正自楊紹明[5])

因篇幅所限，在表 2 中，我們修正了楊紹明[5]所舉例的堅持議題樣板中的單一議題，並以 Item(x)代表對方協商議題、Value(y)代表對方議價數值，以及 String(z)

代表協商物品細節屬性。由表 2 我們可知，前提條件(Precondition)是當 Item(x)屬於協商目標知識本體描述範圍時，我們針對單一議題來看，該議題底下所包含的對

方偏好表達是否符合我方數值或是意見標準範圍，以此進行系統判斷進而在回覆訊息中表達接受對方數值與意見或是回覆我方建議數值與退還待議的訊息。

## 五、ABTNA 系統架構

經由上述各節的分析之後，我們呈現出自動化企業交易協商代理人(ABTNA)完整的系統架構(如圖 5 所示)。在 ABTNA 系統中，自動化協商採購代理人(Automated Negotiation Procurement Agent, ANPA)能夠在開放的環境之下與不特定的自動化協商拍賣代理人(Automated Negotiation Auction Agent, ANAA)進行協商。除了兩者都要具備協商相關的領域知識外，還必須能夠使用不同的互動協定以保持代理人的彈性。Jennings et al.[15]認為自動化協商包含協商目標、協商協定以及協商策略三個元件。參與協商的 ANAA 與 ANPA 雙方代理人透過協商目標知識本體可讓代理人之間皆有相同的領域認知，而協商協定知識本體則讓代理人程式之間有相同的協商協定認知。在協商策略方面，我們根據協商目標知識本體對於協商議題的標準化語意，並視情況需要搭配各種邏輯策略，來進行私有協商策略的建構工作。我們所設計出的 ABTNA 自動化協商系統(見圖 5)，包含下列七個主要元件：

1. Domain Ontology (領域知識本體)：在消費者與零售商協商目標知識本體的設計，我們以 PCDIY 為基礎並進一步結合協商概念屬性改良成為適用於協商環境的 NPPC 知識本體。
2. Negotiation Ontology (協商協定知識本體)：透過 FIPA 所制定的代理人協商互動協定，我們進一步結合自動化協商所設計的協商協定知識本體，可讓代理人之間有著共通的協商協定規則，以利代理人彼此的互動協商作業的一致認知。
3. Logic Reasoning Engine (邏輯推論引擎)：為了使代理人程式之間溝通訊息有相同的認知，透過知識本體可以將我們對事物的認知以抽象化的概念來表達。透過邏輯推論引擎引用協商目標知識本體與協商協定知識本體的步驟，不僅可以將開放式協商環境中不同的外部代理人訊息推論轉換成內部代理人自身所認知的訊息，亦可進行篩選參與協商活動的特定代理人程式。
4. Interaction Protocol Reasoning Engine (互動協定推論引擎)：為了使代理人程式之間有著共同遵循的協商互動步驟，透過互動協定推論引擎引用互動協定的限制與描述，代理人之間有著限定規範的作業活動可供選擇運用，亦使得雙方代理人協商協定有著明確步驟的認知。
5. Negotiation Strategy Reasoning Engine (協商策略推論引擎)：為了使代理人自動化協商獲得最大的利益，協商策略推論引擎運用代理人私有所制定的協商策略與對方協商訊息來進行對比推論作業，透過代理人之間的接受、拒絕與退還議價等作業活動以期獲得最佳的協商結果。
6. Negotiation Strategy Repository (協商策略儲存體)：為儲存 ANPA 與 ANAA 代理人程式各自私有的協商策略，其協商策略是由雙方協商人員遵循協商策略規則樣板，並視情況搭配各種邏輯策略來進行零售商私有協商策略的建構工作。
7. Interaction Protocols (互動協定)：為規範代理人程式之間特定事件觸發(Event Triggering)時可採用的步驟流程，我們採用 FIPA 官方所制定的統一

規範並結合多代理人協商概念來制定其協定內容。

#### 肆、系統實作

為了進一步驗證本文所開發出的自動化企業交易協商代理人(ABTNA)系統的實用性，我們在本節主要是實作關鍵性的系統設計原理。首先，我們呈現 ABTNA 系統所使用的協商目標與協商協定知識本體，之後再以 JTP (Java Theorem Prover)邏輯推論的方式實作不同代理人之間的概念推論。而我們所設計的協商協定知識本體，是透過 SWRL (Semantic Web Rule

Language, 語意網法則語言)來實作描述各項互動協定的概念。經由 SWRL 的描述能力，並利用以 FIPA 官方所制定的 Request Interaction Protocol(請求互動協定)為例，來呈現以 SWRL 描述後的互動協定與運用在代理人回應動作上的協商策略。接著，我們將 SWRL 轉換成 Jess (Java Expert System Shell, 專家系統推理引擎)可解讀的格式，藉此達成利用 Jess 來進行協商推論目的。最後，我們以 Jess 推論引擎結合 JADE (Java Agent Development Framework, 代理人開發平台)來達成多代理人的協商溝通。

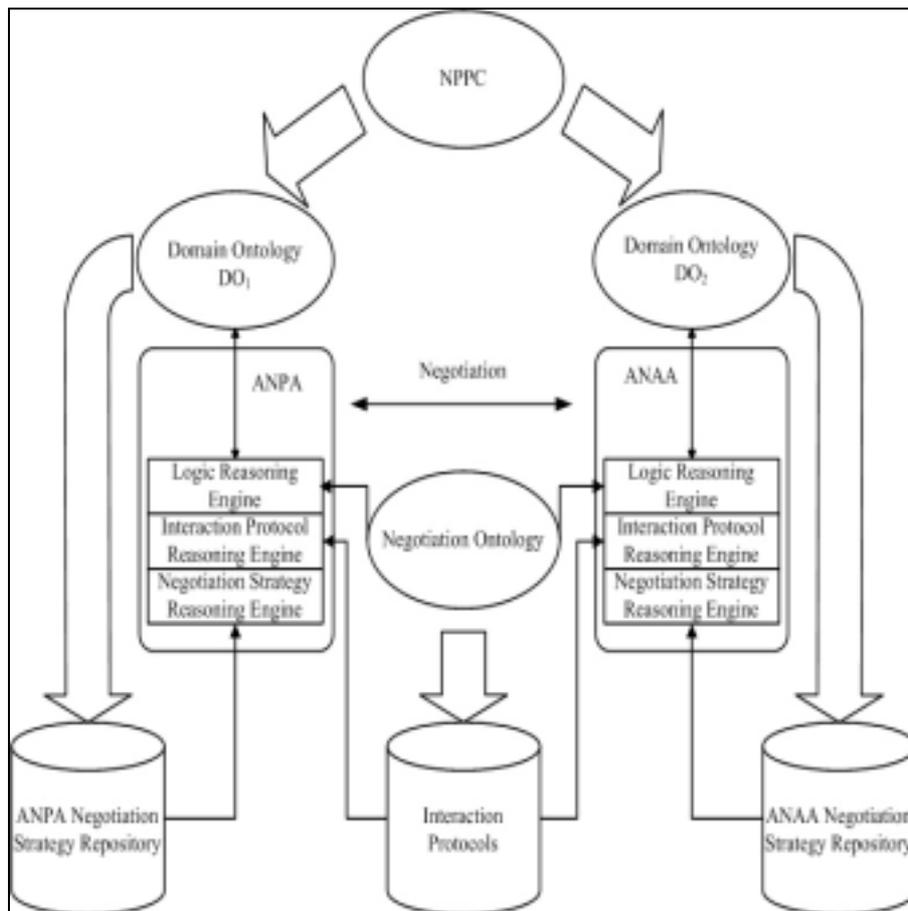


圖 5 ABTNA 自動化協商系統架構

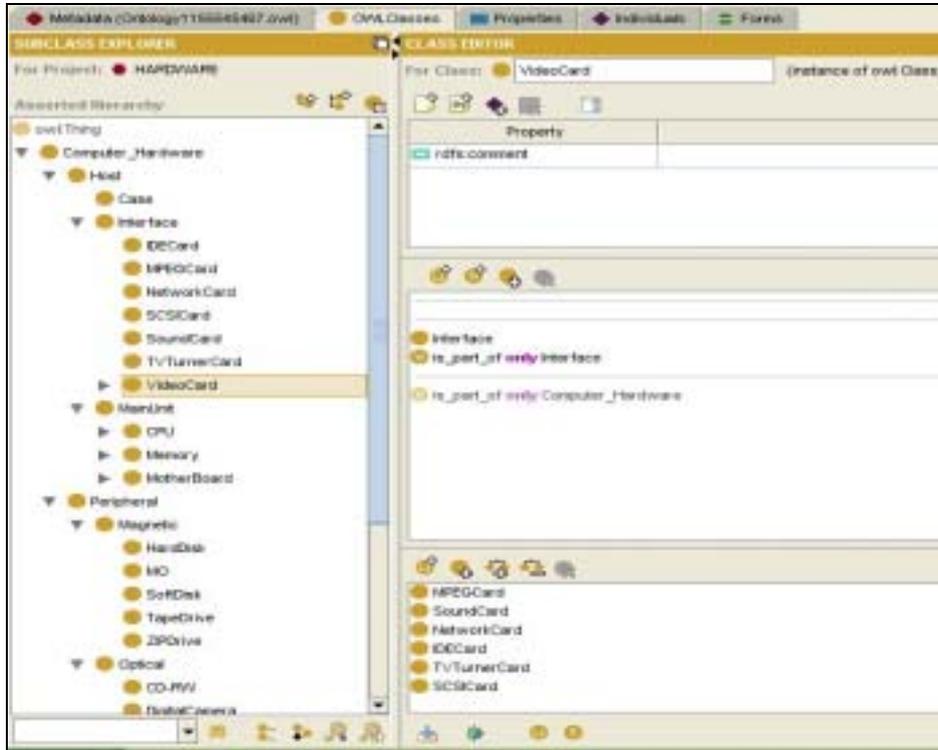


圖 6 以 Protégé 建置協商目標知識本體

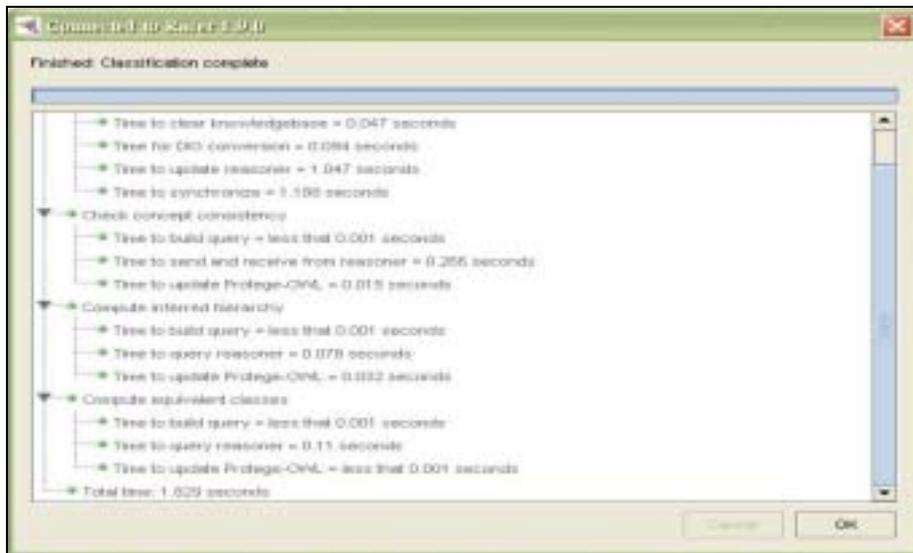


圖 7 Racetrack Pro 驗證結果

### 一、以 Protégé 實作協商目標知識本體

在本節，我們以 OWL (Web Ontology Language, 網路本體語言) 來描述 ABTNA 系統所使用的協商目標知識本體 (亦即, NPPC 知識本體)。目前在開發知識本體的工具中, 多數以 Protégé (<http://protege.stanford.edu>) 為主要的開發工具。該工具除了支援 OWL 語法外, 亦支援 RDF (Resource Description Framework, 資源描述架構) 格式。透過 Protégé 所設計的協商目標知識本體的類別與屬性的階層架構, 顯示於圖 6。例如: 在 Computer\_Hardware 類別底下包含 Host 與 Peripheral 兩個子類別, 而在 Host 類別底下又可分為 Case、Interface 與 MainUnit 三個子類別。

在以 Protégé 初步建置完成後, 我們可以透過 Racer Pro 來驗證整體建置過程中是否有邏輯錯誤存在。Racer System 公司的 Racer Pro 軟體為德國漢堡大學所開發, 它可以與 Protégé 結合做為驗證知識本體的一致性與簡單推論的工具。而透過呼叫 Racer Pro 來驗證協商目標知識本體邏輯正確的結果, 如圖 7 所示。

### 二、以 Protégé 實作協商協定知識本體

我們同樣也以 Protégé 來建置改良型協商協定知識本體 (如圖 8 所示), 透過 Protégé 我們可以清楚地制定完整的協商協定中各個類別元素與關係。此外, 在協商協定知識本體的開發過程中, 我們亦將以 SWRL (Semantic Web Rule Language, 語意網規則語言) 來進行協商協定知識本體各類別與關係的流程描述。以知識本體為基礎的規則語言目前只有 SWRL, 它是結合 RuleML (Rule Markup Language, 規則標記語言) 與 OWL (Web Ontology Language, 網

路知識本體語言) 演變而來, 兩者同樣都是以 XML 為基礎的法則格式, 具有人機可讀的優點。然而, SWRL 相較於 RuleML 則進一步擁有利用知識本體中的元素來進行編輯規則的特點。此外, Protégé 亦整合了 SWRL 的描述編輯功能。我們透過 Protégé 來啟動 SWRL 的編輯規則介面, Protégé 就會自動引用由 W3C (World Wide Web Consortium, 全球資訊網協會) 所規範的 SWRL 與 SWRLB (Semantic Web Rule Language Built-ins) 兩項 Namespace (名稱空間)。在圖 8 的左側粗線框部份, 顯示出 SWRL 整體概念主要由 Imp、Atom、Variable 與 Builtin 四種元素所組成。Imp 表示法則中的 head 和 body 部分, 而 head 和 body 中的敘述子句則是由 Atom 所提供; 法則中所使用的變數部分則記錄於 Variable 中; 在 Builtin 中記錄了 SWRL 可以使用的邏輯比較關係。

### 三、使用 SWRL 描述互動協定與協商策略

針對我們所改良的協商協定知識本體以及 FIPA 官方所發布的互動協定內容, 我們利用 Protégé 中的 SWRL 編輯功能來描述相關互動協定, 其目的是為了達成 ABTNA 系統內部互動協定與協商策略推論引擎可分析解讀的語意格式。因此, 我們以列於第 3.3 節的表 1 中對於 FIPA 官方所制定的 Request Interaction Protocol (請求互動協定) 設計概念為例來說明實作。在表 1 的案例中, 當協商活動中的互動方為協商發起者且擁有合法的 ID 名稱, 而協商狀態為開始且協商互動協定為 Request Interaction Protocol, 則相對於協商發起者的後續活動為 Request (請求)。

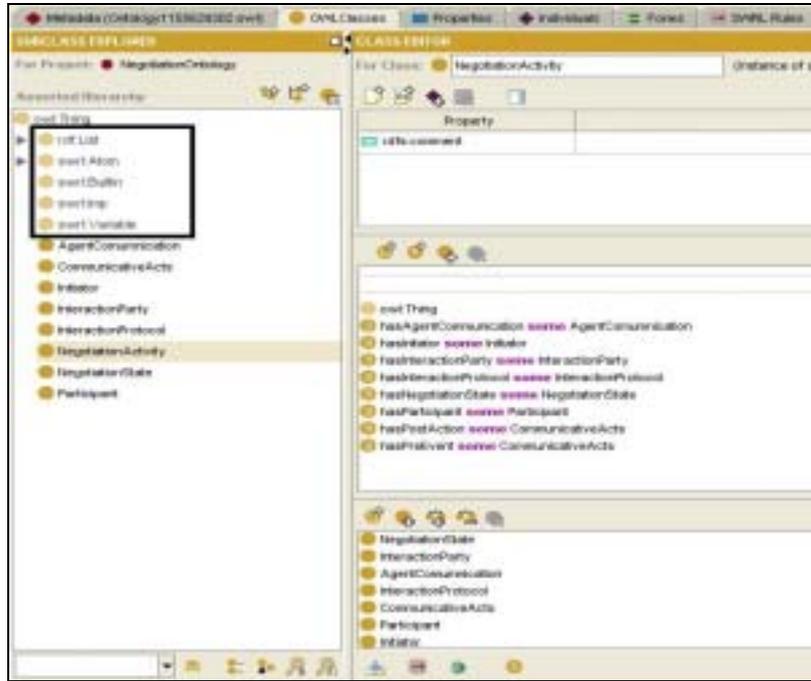


圖 8 以 Protégé 建置協商協定知識本體

利用 Protégé 所編輯的 SWRL 互動協定 (如圖 9 所示), 在 Edit SWRL Rule 對話視窗中所呈現的內容, 我們說明如下:

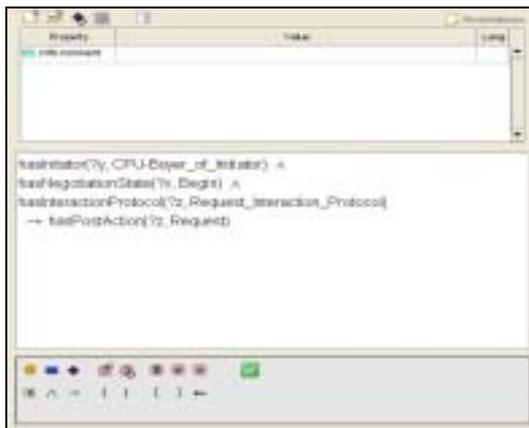


圖 9 以 SWRL 描述互動協定

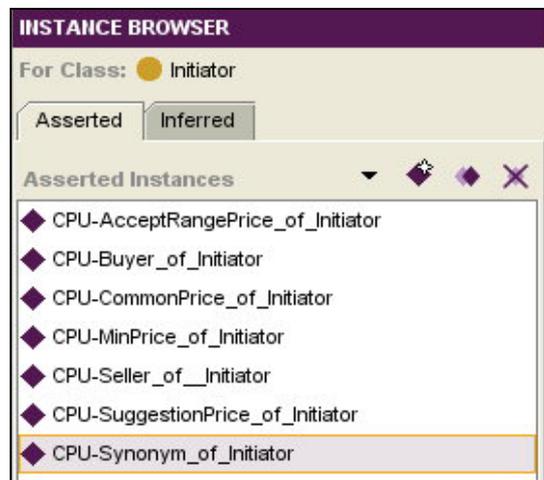


圖 10 引用協商目標知識本體的協商策略屬性之實例建置

- hasInitiator (擁有協商發起者)的關係概念與 CPU-Buyer\_of\_Initiator (亦即,代表購買 CPU 的消費者協商發起者)屬性數據來判斷是否為具有合法的 ID 名稱。
- hasNegotiationState (擁有協商狀態)關係概念上則是用以判定該協商狀態是否為 Begin (開始)狀態。
- hasInteractionProtocol (擁有互動協定)關係概念上判斷該代理人所溝通的互動協定種類是否等於 Request Interaction Protocol (請求互動協定)屬性數據。

如果上述三個關係概念皆成立的話,我們就給予一個推論結果為:有一個 hasPostAction (擁有事後反應行動)關係概念代表該回應動作為 Request (請求)。

如圖 9 所示, CPU-Buyer\_of\_Initiator 是屬於紀錄協商發起者 ID 名稱的字串欄

位。ABTNA 系統中所使用的協商策略必需與互動協定結合推論。在我們所制定的 SWRL 互動協定描述上,我們必需引用協商目標知識本體來做為互動流程中代理人協商策略制定的來源。對於協商協定知識本體中所描述的協商策略在 Initiator (協商發起者)與 Participant (協商參與者)兩個類別下,區分為協商發起者協商策略與協商參與者協商策略。在圖 10,我們經由 Protégé 引用協商目標知識本體的 CPU 零件概念裡的屬性做為實例建置來達成協商策略的依據來源。

經由我們引用協商目標知識本體來進行協商策略實例的建置後,我們以 SWRL 結合互動協定與協商策略來描述所有於協商活動中可能發生的互動流程與私有策略,並且以 3.4 節中所提到的堅持議題樣板中的單一議題為例來思考協商策略的制定。

表 3 以堅持議題樣板(單一議題)為例制定協商策略

|      |            |             |               |                          |
|------|------------|-------------|---------------|--------------------------|
| if   |            | 零售商擁有合法ID名稱 |               |                          |
|      | antecedent | 單一議題        |               |                          |
|      |            | 議題一         | 偏好表達-字串值      | 零售商販賣CPU名稱 ≠ 消費者欲採買CPU名稱 |
| then |            |             |               |                          |
|      | consequent | 單一議題        |               |                          |
|      |            | 議題一         | 拒絕對方所提出的CPU產品 |                          |

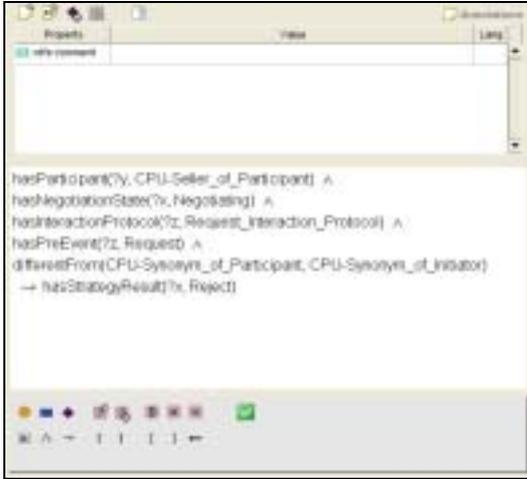


圖 11 以 SWRL 描述協商策略

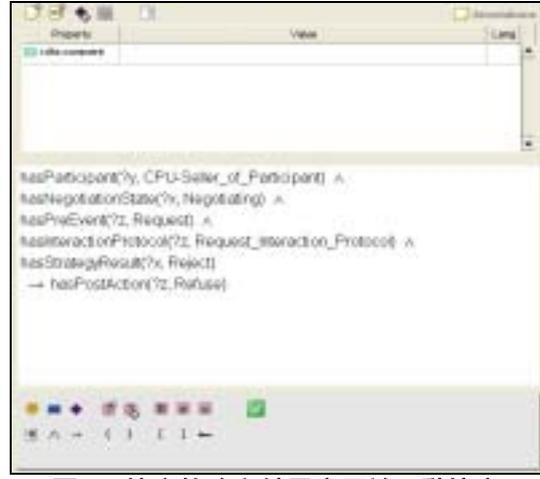


圖 12 協商策略之結果應用於互動協定

在圖 11 中，我們主要以 SWRL 語法來描述表 3 所制定的協商策略。針對上述因 CPU 產品名稱不符的策略而推論出的 Reject 結果，我們進一步將其結合於互動協定流程中來作為代理人程式在協商過程中使用邏輯策略推論引擎來進行推論的判斷依據。如圖 12 所示，代表消費者的自動化協商採購代理人(ANPA)若欲推論在此階段互動流程的後續活動為 Refuse(拒絕)，則必須符合下列四點條件：

1. 當外部代理人角色為 CUP-Seller\_of\_Participant (亦即，代表販售 CPU 零售商的協商參與者)且假設根據協商策略推論可知此協商參與者具有合法的 ID 名稱。
2. 對於整體協商狀態為 Negotiating (協商進行中)且互動協定為 Request Interaction Protocol (請求互動協定)。
3. 相對於整體流程中的前一個活動為協商發起者所提出的 Request (請求)。
4. 代理人程式所制定的私有協商策略被觸發(見圖 12 )且回傳其推論結果為 Reject (拒絕)。

#### 四、以 JTP 實作知識本體的概念推論

在 ABTNA 系統中，我們所設計的邏輯推論引擎，其主要功能在於提供數個不同外部代理人程式用來進行數個協商目標知識本體之間的概念協調，藉此提供代理人程式之間相同的協商目標知識；透過一致性的概念認知，代理人彼此間才能夠進行溝通與推理。在透過以 OWL 語法來表達協商目標知識本體的基礎之下，我們更進一步利用 JTP 來進行概念推論的工作。JTP 是由美國史丹佛大學知識系統實驗室針對 DAML+OIL 與 OWL 所提出來的查詢答覆系統，當中包含邏輯推論引擎以及相關的 JAVA API (應用程式介面)。

在進行概念推論之前，我們假設雙方協商者分別使用 Protégé 開發各自的協商目標知識本體，在過程中皆引用 NPPC 知識本體為其上層概念，並且假設雙方代理人的系統開發人員皆各自對電腦零組件名稱有自己的考量依據而有所變動，但在零組件概念上則皆遵循 NPPC 知識本體電腦零組件概念關係規範。最後，我們將雙方協商目標知識本體分別命名為 USER1 與 USER2。在圖 13 (a)與圖 13 (b)，分別顯示

出 USER1 與 USER2 知識本體。

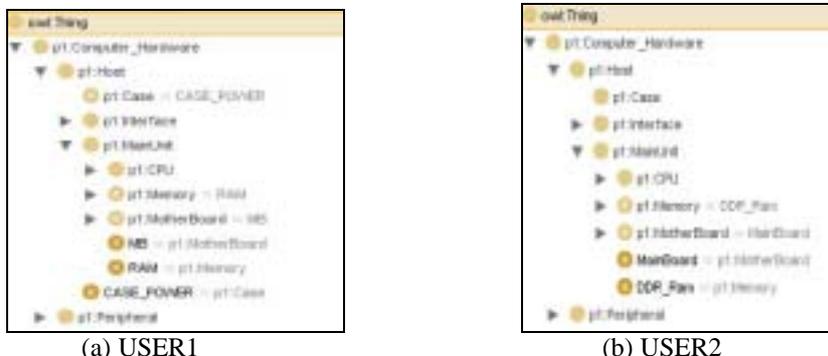


圖 13 協商目標知識本體與 NPPC 知識本體之概念關聯

如圖 13 所示，我們說明如下：

- USER1 協商目標知識本體定義 MB 全等於 NPPC 知識本體的 MotherBoard 零組件概念。
- USER1 協商目標知識本體定義 RAM 全等於 NPPC 知識本體的 Memory 零組件概念。
- USER1 協商目標知識本體定義 CASE\_POWER 全等於 NPPC 知識本體的 Case 概念。
- USER2 協商目標知識本體定義 MainBoard 全等於 NPPC 知識本體的 MotherBoard 零組件概念。
- USER2 協商目標知識本體定義 DDR\_RAM 全等於 NPPC 知識本體的 Memory 零組件概念。

在我們完成建置雙方協商目標知識本體與上層 NPPC 知識本體的關係之後，在 ABTNA 系統運作中，雙方代理人會在協商溝通前啟動 JTP 概念推論引擎。在啟動 JTP 後，代理人會以 load-kb 指令來針對 USER1 協商目標知識本體、USER2 協商目標知識本體與 NPPC 知識本體進行讀取至 JTP 記憶體的工作。透過知識本體的讀取與對比，JTP 方可進行概念推論。在將雙方知識本體讀取完畢後，代理人會以 ask 指令下達下列問句：

(|http://www.w3.org/2002/07/owl#::equivalentClass|http://www.owl-ontologies.com/Ontology1155713868.owl#::|MB| ?x)

```

query succeeded.
bindings 1:
  ?x = |http://www.owl-ontologies.com/Ontology115545467.owl#::|MotherBoard|
bindings 2:
  ?x = |http://www.owl-ontologies.com/Ontology1155713868.owl#::|MB|
bindings 3:
  ?x = |http://www.owl-ontologies.com/Ontology1155716183.owl#::|MainBoard|
>
    
```

圖 14 以 JTP 推論知識本體之間零組件概念的相對關係

此指令問句主要在於詢問 USER1 協商目標知識本體中的 MB 零組件概念是否與

其他已被載入的知識本體有相同的概念存在，而透過 JTP 詢問的結果如圖 14 所示。

推論後的結果顯示：NPPC 知識本體的 MotherBoard 概念同等於 USER1 協商目標知識本體的 MB 概念，也同等於 USER2 協商目標知識本體的 MainBoard 概念。

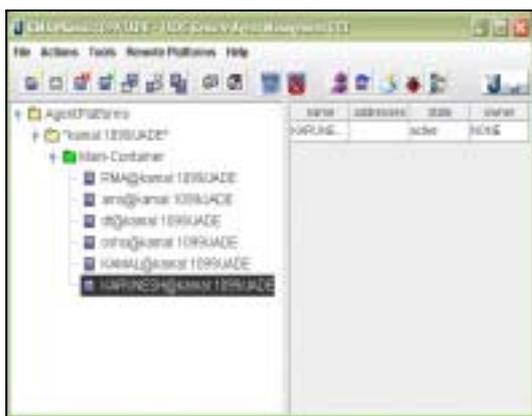


圖 15 JADE 多代理人程式管理開發介面

### 五、結合 JADE 與 JESS 推論引擎的多代理人協商環境

在 ABTNA 系統中，除了在 4.4 節中所呈現的邏輯推論引擎外，另外在考慮協商過程中推論互動流程與策略運用的角色上，則是交由互動協定推論引擎與協商策略推論引擎來負責。在實作上，我們將兩種概念整合規劃並以 SWRL 加以描述而成，其主要目的在於提供 Jess 推論引擎所使用的法則依據。Jess 本身可以用來發展以規則為基礎 (Rule-based) 的專家系統。而 JADE 為一個依循 FIPA 規格並以 Java 程式語言所開發實作的多代理人

(Multi-Agent)管理開發介面，它提供了圖形化的工具幫助我們去做 FIPA 代理人程式的開發與除錯。JADE 的系統管理畫面，如圖 15 所示。

此外，JADE 亦提供與 Jess 推論引擎結合的能力，使其開發出的代理人程式具有推論能力。在 ABTNA 欲使用 Jess 來進行推論互動流程與協商策略工作前，系統必須先行建立 Jess 事實庫 (Facts Base)與推論法則。事實庫的推論法則來源則需導入我們所建置的協商協定知識本體與我們以 SWRL 所描述的互動協定與協商策略。在完成導入現有推論法則進入 Jess 事實庫後當中，之後就可由現有的法則推論出新的事實(亦即，協商結論)。而所有 Jess 推論法則可由 JessTab (亦即，將傳統 Jess 推論引擎與 Protégé 對應整合的附加套件)的法則列表欄中進行查詢，如圖 16 所示。

在 JADE 管理平台上所開發的 ABTNA 系統過程中，我們利用 Java 語言所開發的 ANPA 與 ANAA 代理人在溝通上皆屬於訊息導向。每一個訊息都被封裝在 jade.lang.acl.ACLMessage 物件裡，而圖 17 則呈現 ABTNA 系統運行過程中，代表協商發起者的自動化協商採購代理人(ANPA)與代表協商參與者的自動化協商拍賣代理人(ANAA)彼此溝通的 ACLMessage 訊息內容。

```

(defrule Rule-10 (hasInitiator ?y CPU-Buyer_of_Initiator) (hasNegotiatorState ?x Negotiating) (hasInteractionProtocol ?z Request_Interaction_Protocol) (hasPreEvent ?z Failure) == (assert ()
(defrule Rule-11 (hasInitiator ?y CPU-Buyer_of_Initiator) (hasNegotiatorState ?x Negotiating) (hasInteractionProtocol ?z Request_Interaction_Protocol) (hasPreEvent ?z Inform) == (assert ()
(defrule Rule-9 (hasParticipant ?y CPU-Seller_of_Participant) (hasNegotiatorState ?x Negotiating) (hasInteractionProtocol ?z Request_Interaction_Protocol) (hasPreEvent ?z Request) (name
(defrule Rule-2 (hasParticipant ?y CPU-Seller_of_Participant) (hasNegotiatorState ?x Negotiating) (hasPreEvent ?z Request) (hasInteractionProtocol ?z Request_Interaction_Protocol) (has
(defrule Rule-6 (hasParticipant ?y CPU-Seller_of_Participant) (hasNegotiatorState ?x Negotiating) (hasInteractionProtocol ?z Request_Interaction_Protocol) (hasPreEvent ?z Agree) (hasSt
(defrule Rule-1 (hasInitiator ?y CPU-Buyer_of_Initiator) (hasNegotiatorState ?x Begin) (hasInteractionProtocol ?z Request_Interaction_Protocol) == (assert (hasPostAction ?z Request)) )
(defrule Rule-5 (hasInitiator ?y CPU-Buyer_of_Initiator) (hasNegotiatorState ?x Negotiating) (hasInteractionProtocol ?z Request_Interaction_Protocol) (hasPreEvent ?z Refuse) == (assert ()
(defrule Rule-4 (hasInitiator ?y CPU-Buyer_of_Initiator) (hasNegotiatorState ?x Negotiating) (hasInteractionProtocol ?z Request_Interaction_Protocol) (hasPreEvent ?z Agree) == (assert ()
(defrule Rule-3 (hasParticipant ?y CPU-Seller_of_Participant) (hasNegotiatorState ?x Negotiating) (hasPreEvent ?z Request) (hasInteractionProtocol ?z Request_Interaction_Protocol) (has
(defrule Rule-8 (hasParticipant ?y CPU-Seller_of_Participant) (hasNegotiatorState ?x Negotiating) (hasInteractionProtocol ?z Request_Interaction_Protocol) (hasPreEvent ?z Request) (diffe
(defrule Rule-7 (hasParticipant ?y CPU-Seller_of_Participant) (hasNegotiatorState ?x Negotiating) (hasInteractionProtocol ?z Request_Interaction_Protocol) (hasPreEvent ?z Agree) (hasSt
    
```

圖 16 Jess 推論引擎的法則列表

```

構勤 ABTNA
[ACLMessage begin]
From: Initiator
To: Participant
Performative: request
Protocol: fipa-request
CID: 1A2020688723_0
Content:
  <?xml version="1.0"?>
<rdf:RDF
  xmlns:owl="http://www.owl-ontologies.com/Ontology1155545467.owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns="http://www.owl-ontologies.com/Ontology1155713868.owl#"
  xml:base="http://www.owl-ontologies.com/Ontology1155713868.owl">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://www.owl-ontologies.com/Ontology1155545467
    .owl"/>
  </owl:Ontology>
  <owl:Class rdf:ID="UGn_Card">

```

圖 17 ABTNA 內部代理人溝通的 ACLMessage 訊息內容

## 伍、結論

我們以知識本體結合代理人程式與自動化協商機制開發出自動化企業交易協商代理人 (ABTNA) 系統，而該系統平台具有下列優點：

- ABTNA 透過知識本體與議價機制的運用，使得代理人更加深入人類對於採購買賣商品的思考運作模式。
- 我們將協商領域知識與協商協定以知識本體來進行抽象化概念的呈現，讓代理人程式與協商知識分離，使得代理人程式的維護與知識的運用更具有彈性。
- 協商協定知識本體亦提供代理人之間更具動態的協商作業流程。
- 我們以 FIPA 所制定的規範來描述互動協定，使得代理人程式互動協定更具有標準一致性。
- 代理人協商策略的推論能力，促成了

買賣雙方達成符合彼此利益的交易結果。

在代理人程式的研究中，如何更深入取代人類的作業思考模式亦是專家們不斷努力的方向。在有關語意網技術應用於企業交易代理人的學術研究與實務方面，本文所發展出的 ABTNA 系統已向前邁進了一大步。本文的主要貢獻總結如下：

- 在學術的貢獻上，我們以知識本體的方式來呈現代理人的協商協定與協商目標並結合協定描述，主要是為了提昇代理人協商範圍與推論能力，並促進代理人協商環境更具有彈性與抽象化。
- 在實務的貢獻上，我們所發展出的 ABTNA 系統可以有效地解決網路購物商品資訊篩選與協商議價等複雜的人工程序。

在未來展望方面，本文亦可做為日後網路購物交易模式之代理程式自動化應用

方面的研究基礎。在議價技術上可以發展出複雜多樣化的多重商品案例式結合推銷議價方法或是更加細微精確的抬價與減價的演算法。此外從知識本體的角度而言，面對資訊產品的推陳出新，如何將本系統所牽涉到的協商目標知識本體賦予自動化新增商品實例與新產品概念歸納的能力亦是將來可以研發的方向。

### 參考文獻

- [1] 林建良,「以知識本體提供代理人建構共通之協商環境 - 以生產排程協商為例」, 中原大學資訊管理學碩士論文, 2005。
- [2] 林政瑩,「影響拍賣網站使用者到拍賣網站購物意願因素之研究」, 國立中山大學企業管理系碩士論文, 1999。
- [3] 洪懷謙,「以本體論考量之教材數位版權管理之研究」, 國立高雄師範大學資訊教育研究所, 2004。
- [4] 楊勝源、朱毓君、何正信,「以本體論強化網路 FAQ 系統之解答整合能力」, 第六屆人工智慧與應用研討會, 高雄, 台灣, 2001。
- [5] 楊紹明,「建構語意網規則導向之電子契約自動化協商機制之研究」, 中原大學資訊管理學碩士論文, 2004。
- [6] Alba, L. W., and Hutchinson J. W., "Dimensions of Consumer Expertise" , Journal of Consumer Research, vol. 13, no. 4, 1987, pp. 411-54.
- [7] Brucks, M., "The Effect of Product Class Knowledge on Information Search Behavior" , Journal of Consumer Research, vol. 2, no. 1, 1985, pp. 1-16.
- [8] Etzioni, O., and Weld, D. S., "Intelligent Agents on the Internet: Fact, Fiction, and Forecast" , IEEE Expert, August , 1995, pp. 44-49.
- [9] Fensel, D., Benjamins, V. R., Motta, E., and Wielinga, B., "UPML : A framework for knowledge sytem reuse" , In Proceedings of the 16th International Joint Conference on AI , Sweden, 1999, pp. 16-21.
- [10] Fensel, D., "Ontologies and Electronic Commerce" , IEEE Intelligent System , Volume 16, no. 1, 2001, pp. 8-14.
- [11] Fitzmaurice, G., " Form-Centered Workflow Automation Using an Agent Framework" , Brown University, USA, 1991.
- [12] Guarino, N., " Formal Ontology, Conceptual Analysis and Knowledge Representation" , International Journal of Human-Computer Studies, 43, 1995, pp. 625-640.
- [13] Grosf, B. N. and Poon, T. C., " SweetDeal: Representing Agent Contracts withExceptions Using Semantic Web Rules, Ontologies, and Process Descriptions" , International Journal of Electronic Commerce, Vol. 8, No. 4, 2004, pp. 61-97.
- [14] Horrocks, I., Patel-Schneider, P. F., Bechhofer, S. and Tsarkov, D., "OWL Rules: A Proposal and Prototype Implementation" , Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 3, No. 1, 2005, pp. 23-40.
- [15] Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Parsons, M. and Sierra, C., "Automated Negotiation: Prospects, Methods and Challenges" , Group Decision and Negotiation,.Vol. 10, No. 2, 2001, pp. 199-215.
- [16] Jennings, N. R., Sycara, K., and Wooldridge, M., "A Roadmap of Agent Research and Development" , Journal of Autonomous Agents and Multi-Agent Systems, Vol. 1, No. 1, 1998, pp. 7-38.

- [17] Lei, Y., Feng, Y. Q., Wei, Q. and Yang, L., "Adding Argument in Automated Negotiation System to Support Flexible Negotiation" , In Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, China. November, 2002, pp. 1419-1422.
- [18] Lomuscio, A. R., Wooldridge, M., and Jennings, N. R., "A Classification Scheme for Negotiation in Electronic Commerce" , International Journal of Group Decision and Negotiation, Vol. 12, No. 1, 2003, pp. 31-56.
- [19] Maes, P., "Modeling Adaptive Autonomous Agents" ,In Artificial Life: An Overview, Cambridge, MA: MIT press, 1994, pp. 135-162.
- [20] Meister, D. B., "Prescriptive Analysis Negotiator Support System" , Ph.D. Thesis, Department of Management Sciences, University of Waterloo, 1993.
- [21] Oliver, J. R., "A Machine-Learning Approach to Automated Negotiation and Prospects for Electronic Commerce" , Journal of Management Information System. 13.3, 1997, pp. 83-122.
- [22] Park, C. W., Feick, L., and Mothersbaugh, D. L., "Consumer Knowledge Assessment : How Product Experience and Knowledge of Brands, Attributes, and Features Affects What Think We Know" , Advances in Consumer Research, vol. 19, no. 3, 1992, pp. 193-98.
- [23] Raiffa, H., "The Art and Science of Negotiation " ,Cambridge, MA. : Harvard University Press, 1982.
- [24] Schmidt, J. B., Spreng, R. A., "A Proposed Model of External Consumer Information Search " , Journal of Academy of Marketing Science, vol. 24, no. 3, 1996, pp. 246-56.
- [25] Shocker, Allan, Ben-Akiva M., Boccara B., and Nedungadi P., "Consideration Set Influences on Consumer Decision -Making and Choice: Issues, Models and Suggestions" , Marketing Letters, 2 (3), 1991, pp. 181-97.
- [26] Tu, M. T., Kunze, C. Kunze and Lamersdorf, W., "A rule management framework for negotiating mobile agents" , In Proceedings of the 4th International Enterprise Distributed Object Computing Conference, Makuhari, Japan, 2000, pp. 135-143.
- [27] Turban, E., King, D., Viehland, D., and Lee, J., "Electronic Commerce : A Managerial Perspective" , Prentice-Hall, New Jersey, USA, 2006, pp. 140-141.
- [28] Weiss, G., "Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence" , MIT Press, Cambridge, Massachusetts, USA, 1999.
- [29] Wooldridge, M., and Jennings N. R., " Intelligent Agents: Theory and Practice " , Knowledge Engineering Review, Vol. 10(2), 1995, pp. 115-152.