

國立台東大學資訊管理學系
環境經濟資訊管理碩士在職專班
碩士論文

指導教授：謝明哲 博士

代理人協商應用於銀行 IT 專案管理之研究
A Study of Applying Agent Negotiation to
Bank IT Project Managment

研究生：吳靜怡 撰

中華民國九十八年七月

NTTU Library



國立台東大學資訊管理學系
環境經濟資訊管理碩士在職專班
碩士論文

代理人協商應用於銀行 IT 專案管理之研究
A Study of Applying Agent Negotiation to
Bank IT Project Managment



研究生：吳靜怡 撰
指導教授：謝明哲 博士
中華民國九十八年七月

國立台東大學
學位論文考試委員審定書
系所別：資訊管理學系

本班 吳靜怡 君

所提之論文：應用代理人協商於銀行 IT 專案管理
業經本委員會通過合於 碩士學位論文 條件

論文學位考試委員會：

陳仲敏

(學位考試委員會主席)

廖國良

謝明昭

(指導教授)

論文學位考試日期：98 年 7 月 17 日

國立台東大學

附註：1. 本表一式二份經學位考試委員會簽後，送交系所辦公室及註冊組或進修部存查。

2. 本表為日夜學制通用，請依個人學制分送教務處或進修部辦理。

博碩士論文電子檔案上網授權書

(提供授權人裝訂於紙本論文書名頁之次頁用)

本授權書所授權之論文為授權人在 國立臺東大學 資訊管理學系碩士班 _____
組 97 學年度第二 學期取得 碩士 學位之論文。

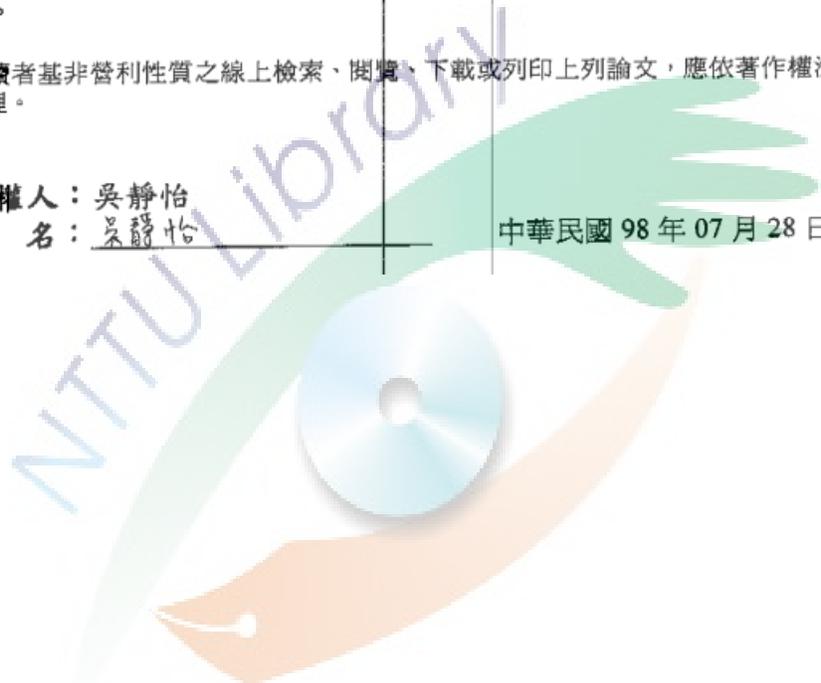
論文題目：代理人協商應用於銀行IT專案管理之研究
指導教授：謝明哲

茲同意將授權人擁有著作權之上列論文全文(含摘要)，非專屬、無償授權國家圖書館及本人畢業學校圖書館，不限地域、時間與次數，以微縮、光碟或其他各種數位化方式將上列論文重製，並得將數位化之上列論文及論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

- 讀者基非營利性質之線上檢索、閱覽、下載或列印上列論文，應依著作權法相關規定辦理。

授權人：吳靜怡
簽 名：吳靜怡

中華民國 98 年 07 月 28 日



博碩士論文授權書

本授權書所授權之論文為本人在 國立臺東大學 環境經濟資訊管理系(所)
組 97 學年度第 2 學期取得 碩 士學位之論文。
論文名稱：應用代理人協商於銀行 IT 專案管理

本人具有著作財產權之論文全文資料，授權予下列單位：

同意	不同意	單位
<input checked="" type="checkbox"/>	<input type="checkbox"/>	國家圖書館
<input checked="" type="checkbox"/>	<input type="checkbox"/>	本人畢業學校圖書館
<input type="checkbox"/>	<input checked="" type="checkbox"/>	與本人畢業學校圖書館簽訂合作協議之資料庫業者

得不限地域、時間與次數以微縮、光碟或其他各種數位化方式重製後散布發行或上載網站，藉由網路傳輸，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

同意 不同意 本人畢業學校圖書館基於學術傳播之目的，在上述範圍內得再授權第三人進行資料重製。

本論文為本人向經濟部智慧財產局申請專利(未申請者本條款請不予理會)的附件之一，申請文號為：_____，請將全文資料延後半年再公開。

公開時程

立即公開	一年後公開	二年後公開	三年後公開
			<input checked="" type="checkbox"/>

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未勾選，本人同意視同授權。

指導教授姓名：游明坤 (親筆簽名)

研究生簽名：吳靜怡 (親筆正楷)

學 號：4396012 (務必填寫)

日 期：中華民國 98 年 7 月 27 日

1. 本授權書 (得自 <http://www.lib.ntnu.edu.tw/theses/> 下載) 請以黑筆撰寫並影印裝訂於書名頁之次頁。

2. 依據 91 學年度第一學期一次教務會議決議: 研究生畢業論文「至少需授權學校圖書館數位化，並至遲於三年後上載網路供各界使用及校內瀏覽。」

授權書版本: 2008/05/29

謝 誌

經過了兩年的學習，終於得以順利完成論文了。首先非常感謝恩師謝明哲教授，在這段期間的細心指導及諄諄教誨，使我受益良多，在此深表感激。

口試期間，非常感謝陳仲儼教授、廖國良教授在百忙之中蒞臨指導，並提供了許多寶貴建議，因為有他們的指正與指導，使本論文可以有更完整的呈現，在此致上最誠摯的感謝。

在此，我也要感謝同班同學的支援，在我寫論文期間常提供資料及建議，這樣的協助讓我得以兼顧學業及工作，並順利完成論文，非常感謝。

最後要感恩的是家人大力支持的心意，在這兩年求學期間，因工作上的需要，必須調整工作型態，因此在學業及工作上的壓力，常使我身心疲憊，也曾感受到挫折而想要放棄，最後讓我堅持下去的力量還是來自於家人的鼓勵。在此必須要感謝家人的支持與鼓勵，才能使我順利地完成學業。謝謝大家，再一次感謝所有關愛我的人，願你們也能與我一起分享這喜悅。

吳靜怡 謹識

于台東大學資訊管理學系

九十八年七月

代理人協商應用於銀行 IT 專案管理之研究

作者：吳靜怡

國立台東大學 環境經濟資訊管理在職進修碩士專班

摘要

資訊安全及系統品質對於銀行業非常重要。但金融 IT 產業爲了節省培養技術人員的成本支出，系統外包已成爲一個趨勢。就現況下，銀行 IT 建立良好的協商過程、有效率地管理專案才能有機會創造雙贏局面。初始化專案時，提出的計畫需做協商，讓參與者都願意朝共同目標前進及努力。接下來，當專案計畫成立後，進行系統開發時也需要管理，例如突發狀況處理及協調，這都是專案管理的任務。無論是計畫建置時亦或進行時，「專案管理」都是一門重要的課題，因此，專案管理者 (Project Manament) 的角色日益重要及不可或缺。

然而，專案管理者的角色並不是萬能的。在考量應用性及實用性的前提下，專案管理者職責重點在統合相關專家們所提供支援來解決問題。專案管理進行時，需考慮到包含人力、成本、範疇、時間…等等多方面的溝通屬性，而這些屬性往往會因爲環境的問題而改變。本研究以代理人協商爲基礎建立一個類似人類進行專案管理時的自動協商機制，建構系統模型以驗證所提之協商機制及基因演算法的可行性，並顯示代理人協商模擬實驗結果，以及協商失敗時，透過基因演算法所提議的最佳專案組合。

※本研究提出兩個核心概念：

1. 將多代理人協商機制運用在專案管理上，賦予代理人不同的協商行為及任務，使系統所創造之協商行為更趨向人性化。
2. 當協商失敗時，將會進行基因演算法，試圖找出最佳提案組合。

目的是將系統代理人應用於「整合型協商機制」，此協商機制讓參與者能依據本身的利益考量做出屬性彈性調整。舉例來說，以 IT 的立場，在協商時，會採取最嚴格的方式來要求系統品質管理，IT 在這方面就不可能做讓步處理。其次，當任何提案皆不能滿足到參與者時，此代理人協商系統會進行基因演算法，運用成本效益值為專案選取的目標最適值，試圖找出最佳專案組合，提供參與者做選擇。

在本研究中提到關於協商失敗時，系統會嘗試找出最佳提案組合給予建議，並記錄結果以提供代理人系統學習案例，逐漸累積思考及決策能力，此模式在目前的相關研究中較少被提及，未來可作為後續研究之方向。

關鍵詞：代理人、自動協商機制、成本效益法、遺傳演算法、專案管理

A Study of Applying Agent Negotiation to Bank IT Project Management

Author : Ching-Yi Wu

Abstract

The information security and the system quality for the banking are extremely important. However, the financial IT industry in order to save the cost of training technical personnel expenditures, system outsourcing has been a trend. On present situation, bank IT has to establish a good consultative process and manage the project efficiently to have the opportunity to create the win-win situation. In the beginning of project, the plan proposed to be done must be drafted in consultation with the participants, so that the participants are willing to move forward toward common goals. When the project plan had been established, systems development also needs management. This is the project management duty for processing and coordination of emergency situation. Whether plan is beginning or proceeding, "project management" is an important issue. Therefore, it makes PM more important and necessary.

However, project managers are not omnipotent in reality. On the premise of application and practicality, resources and suggestion provided by experts to solve the problem is project managers' responsibilities, yet in progress of Project Management. In reality, it needs a lot of management attribute, such as manpower, cost, category, time and so on. These factors often change with environment. Our research established an Automated Negotiation Mechanism that progresses similar to human thinking process in project management on the basis of Multi-agent systems. In this study, we also build the system model to verify the feasibility of our consultation mechanism and genetic algorithm, and show the results of agent

consultation simulation and the best special case combination suggested by the genetic algorithm when the negotiations fail.

✘ This study provides two core concepts:

1. When agent consultative mechanism endows agents with different negotiation behavior and tasks of PM, the automated negotiation system has become more user-friendly.
2. If the consultation fails, there will be a genetic algorithm trying to find the best combination.

The purpose is to apply the agent in an integrated consultative mechanism, so that participants can consider their own interests in accordance with the properties to make adjustments flexible. For example, in the consultation, IT will adopt the most stringent requirements of system quality. In this regard, IT is impossible to make concessions. If all proposals cannot meet the participants' needs, the genetic algorithm will be carried out to find the best combination according to the cost-effective value. Consequently, the best combination is used to offer participants further choice.

In this research mentioned when consultation defeats, the system will try to find the best combination to give advice on the proposal, and to supply a record of the results to the agent system. By this way will accumulate and train thinking and decision-making capacity gradually, the model in the current research less mentioned. In the future may supply direction of the following research.

Key words: agent, automated negotiation, cost benefit principle, genetic algorithm, project management

目次

第一章 緒論	1
1.1 研究動機.....	1
1.2 研究問題.....	2
1.3 研究目的與範圍.....	4
1.4 研究限制.....	4
1.5 研究方法及論文結構.....	5
第二章 文獻探討	7
2.1 專案定義.....	7
2.2 專案管理定義.....	8
2.2.1 專案風險及影響定義.....	11
2.2.2 專案成本管理圖-Control Costs.....	12
2.3.1 代理人和多重代理人的特色.....	14
2.3.2 多代理人系統.....	16
2.3.3 代理人協商.....	17
2.4 協商理論.....	19
2.4.1 協商及談判的定義.....	19
2.4.2 協商類型.....	20
2.4.3 協商機制.....	22
2.5 成本效益分析法.....	23
2.5.1 成本效益分析的理論基礎.....	23
2.5.2 成本效益分析的計算方式.....	24
2.5.3 時間、風險和不確定性.....	25
2.5.4 預算限制和分配效果.....	26
2.5.5 成本效益分析與成本效能分析的比較.....	26
2.6 JADE 代理人發展環境.....	26
2.6.1 代理人架構種類.....	27
2.6.2 JADE 簡介.....	27
2.6.3 AMS(Agent Managemet System)探討.....	30
2.6.4 JADE 的行為表現.....	32
2.7 基因演算法基本概念.....	35
第三章 研究方法	39
3.1 代理人協商環境架構設計.....	39
3.2 代理人協商系統模擬.....	42

3.3 代理人協商系統之基因演算法模式建立.....	49
第四章 建置代理人協商驗證系統與結果分析.....	59
4.1 代理人雛型系統之協商機制模擬.....	59
4.2 雛型協商系統說明與實驗結果分析.....	72
4.2.1 協商雛型系統建置說明.....	72
4.2.2 基因演算法之結果分析.....	77
第五章 研究結論.....	79
5.1 結論.....	79
5.2 未來研究方向.....	80
參考文獻.....	81



圖次

圖 1.1 專案流程圖	3
圖 1.2 本研究的主要流程圖.....	6
圖 2.1 專案整合管理圖(PROJECT INTEGRATION MANAGEMENT OVERVIEW)	10
圖 2.2 專案管理之成本控制圖.....	13
圖 2.3 BDI 代理人架構圖	14
圖 2.4 協商空間觀念圖	22
圖 2.5 JADE 平台執行環境與 CONTAINER 關係圖.....	28
圖 2.6 JADE 平台系統架構.....	29
圖 2.7 DEPICTION OF THE DAGENT MANAGEMENT ONTOLOGY	30
圖 2.8 取得代理人名稱	31
圖 2.9 取得代理人 AID 程式片段.....	31
圖 2.10 AID 組成結構	31
圖 2.11 AGENT THREAD PATH OF EXECUTION	33
圖 2.12 THE JADE BEHAVIOURS HIERARCHY	34
圖 3.1 代理人之關係圖	42
圖 3.2 協商代理人架構圖.....	42
圖 3.3 代理人協商流程設計.....	43
圖 3.4 運用成本效益分析及遺傳演算法為基礎之專案協商成立流程圖	48
圖 3.5 CROSS-OVER 圖解一.....	56
圖 3.6 CROSS-OVER 圖解二.....	56
圖 3.7 CROSS-OVER 圖解三.....	56
圖 3.8 CROSS-OVER 圖解四.....	56
圖 3.9 CROSS-OVER 圖解五.....	56
圖 4.1 使用者代理人進行協商機制-1 循序圖	61
圖 4.2 使用者代理人進行協商機制-2 循序圖	61
圖 4.3 COST 之最大集合表示圖	64
圖 4.4 TIME 之最大集合表示圖	65
圖 4.5 IT 專案管理代理人與使用者代理人之溝通機制	69
圖 4.6 基因演算法示意圖.....	71
圖 4.7 五個協商屬性之區間值.....	73

圖 4.8 五個協商屬性之區間值系統畫面.....	74
圖 4.9 協商第五次成功之表示圖.....	76
圖 4.10 協商第五次成功之系統畫面圖.....	76
圖 4.11 演化世代為 20 的驗證結果.....	77
圖 4.12 演化世代為 50 的驗證結果.....	78
圖 4.13 演化世代為 100 的驗證結果.....	78



表次

表 2.1 四種專案目標對專案管理的影響定義表	12
表 2.2 協商特性	19
表 2.3 代理人協商讓步策略	21
表 2.4 成本效益分析計算方式	24
表 3.1 代理人角色配置	41
表 3.2 設定初始族群	51
表 3.3 計算成本效益值	52
表 3.4 對專案組合進行編碼	53
表 3.5 經選擇後的個數	54
表 3.6 經過選擇後的組成	54
表 3.7 選擇及交配後的族群	57
表 3.8 突變後的族群	57
表 4.1 UA 所設定之屬性數值	62
表 4.2 IA 所設定之屬性數值	62
表 4.3 VA1 所設定之屬性數值	62
表 4.4 VA2 所設定之屬性數值	63
表 4.5 VA3 所設定之屬性數值	63
表 4.6 各提案之屬性值	64
表 4.7 屬性調整後之協商區間值	66
表 4.8 第五次協商記錄表	67
表 4.9 第十次協商記錄表	69

第一章 緒論

1.1 研究動機

近年來，銀行爲了因應資訊化建設，常採用外包的方式來降低系統開發與維護產生的龐大費用。但此方式也讓許多外包風險因而衍生，其風險如：成果未達目標；規劃不佳，導致系統品質不良；或無法有效調度人力等問題。因此在處理各個使用端對系統的需求時，銀行相關部門有責任來評估整個專案並做整合與管理。Project Management Institute(2008)提出專案管理主要工作是運用知識、技能、工具、方法等方式來解決專案需求，並運用起始、計劃、執行、監控、結束等五大程序進行。並定義專案管理九大管理知識領域：整合、範疇、時間、成本、品質、人力資源、溝通、風險及採購管理，以銀行 IT 專案管理這個層面分析，品質方面是很重要的一環，當品質出現瑕疵時，往往銀行所需付出的成本非常高，並且商譽也會受影響。因此銀行 IT 在做專案評估及進行系統開發時，相當注重品質的管理，而在專案管理之九大管理知識領域中，各個領域皆會互相影響。譬如以銀行 IT 的專案管理來看，品質相當重要，而在顧及品質時，相對的，要付出的成本、人力資源、時間也會因此做調整。所以本研究的動機之一，就是將銀行 IT 部門原本運用的傳統人力協商方式轉換成以系統進行，透過代理人的運作達到協商的目的。

一直以來，因銀行 IT 內部未能有一協商平台可協助內部人員統籌及控管各專案的進行，使得 IT 與其它部門的協商程序也變爲較爲繁瑣。必須歷經不斷地開會，才能掌握可用資源及進度，相對地，也造成人力資源無形中浪費，因此創造專案管理協商系統，簡化開會的程序及次數是本研究的動機之二。而由於 IT 內部許多專案多會互相有關聯，或是互相影響，譬如 A 專案與 B 專案必須由 A 專案先進行，才能繼續進行 B 專案。所以 A 專案的如期完成，對於 B 專案相對重要。另一種情況是 A、B 專案必須同時進行時，如有資源需共用，這個因素也必須考慮。較爲單純的情況是 A 專案本身就有因素被限制，如限制成本、完成期限等，皆是專案管理需考量的因素。本研究將會就各個參與者設定其爲代理人，由代理

人依其重視因素設定協商區間值，並進行協商。因此，就銀行 IT 專案管理之探討而言，本研究的最大動機為進行專案中各項影響因素的協商，分析其影響程度，進而完成最多參與者同意的專案計畫書，幫助銀行 IT 人員能更有效掌握各方資源，進行專案管理。而本研究目前進行的協商探討僅為專案管理的小部份而已，尚未完成全自動化協商。所以，以此為基礎，希望未來的研究能觸及更多角度來分析，以期創造更趨向於自動化的協商系統。

而在規劃專案、協調人力時，協商行為扮演一個重要角色。透過使用者、IT、人員、廠商等各方專業領域，由自動協商來達成資源供需均衡並做最有效益的分配。因此，本研究提出以代理人協商系統來支援銀行 IT 人員進行專案管理，在進行專案管理時，經由代理人協商解決各方歧異之意見，進而達成資源配置的最佳化。採用代理人建構協商機制的系統，使資源獲得最有效的管理，並採用代理人技術來完成參與者及 IT 雙方的協商機制。

一般來說，在規劃專案計畫時，實際上所需考量的層面非常廣且複雜，而本研究的重點即是運用這些外在因素轉換其為量化，幫助相關人員瞭解此一動線的優與劣。本研究所建構之代理人系統非決策系統，而是賦予代理人有思考能力，並藉由行為模擬人類社群溝通以達成協商。以專案管理例子來剖析，當以建構之代理人系統實際模擬時，我們將會建立擁有不同行為及立場之代理人，運用其行為來進行溝通，並利用基因演算法進行演化，以繁衍新的子代，建立適者生存的環境，使此系統擁有社群能力，並可實際解決在規劃專案或專案需重新評量時所產生的協商行為。

1.2 研究問題

在 IT 資源配置中，有效整合及運用公司內各種資源，成為專案執行成敗的關鍵，而當整合專案時，實際上是會有著限制的。以下幾項狀況可做說明：

1. 當資源不夠時，運作時間將會延長，而影響完成時間。
2. 整合所有資源完成此專案時，必須考量到成本是否符合，資源是否浪費。

3. 專案計畫範圍的增減，對成本、資源的運用、完成時間，皆會直接被衝擊。

在本研究所提出之應用代理人協商系統於銀行 IT 專案管理，將必須考量以上三點因素。而各種資源中的關鍵，又以人力資源為首要，當新專案推動形成時，如何選才與用才成爲專案主持人要審慎處理的項目。專案開始時，專業人才都將參與專案進行，最主要必須配置其適當的任務來執行，使專案能順利，此外，也必須隨時監控專案進度，在狀況發生或進度落後時，能第一時間處理，其專案流程圖如圖 1.1 所示。

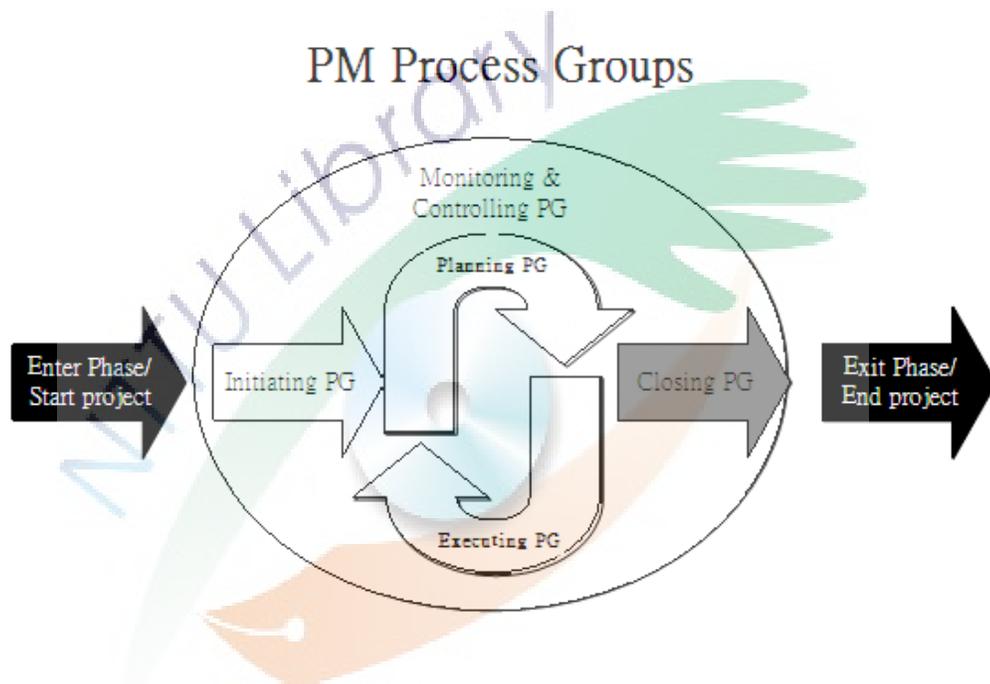


圖 1.1 專案流程圖

資料來源:「Project Management Institute，2008」

1. 專案計畫開始研擬時，會遇到的第一個問題爲收集需求端的意見。
2. 研擬專案計畫時，將會針對使用者提供之需求，分析所需資源，本研究將以範疇、時間、成本、品質、人力資源資源爲首要，配合整合管理，來做需求分析及協商。
3. 突發狀況人力調配問題：假設當專案管理進行到中途時，若有一執行者有突

發將況發生，而必須中斷任務執行，此時，監控專案將啓動並挑選出適妥的方案，繼續執行後續任務。

4. 專案管理進行時廠商與使用者之配合及管理：假設專案進行中與廠商及使用者間的配合有突發狀況，如專案大方向的改變需重新調整專案內容，需緊急調配，以利專案進行。

1.3 研究目的與範圍

代理人協商系統可分析許多協商性的工作，爲解決專案管理中的成本與資源配置問題，並處理許多繁雜且常規的工作，在本研究中會定義個別決策成員能有獨特的屬性及擁有自己的決策行爲模式。以銀行 IT 資源配置可以達到最佳化爲基礎，本研究將採用退讓協商機制與成本效益分析模式，以改善資源不足及不均現象並應用代理人協商來發揮代理人之自主性及協商性進行專案管理活動。具體而言，本研究的主要目的如下：

1. 利用協商機制達到資源配置最佳化建立協商之系統架構。
2. 建構一個採用遺傳基因演算法與成本效益分析理論的代理人協商系統雛型。
3. 代理人在專案管理中，並不能完全來處理專案，必須與各任務相互影響，包括參與人數、執行期間、難易程度。須由自主性的協調及多方面的溝通來進行配置的過程，達到專案管理中人力資源配置合適化。因此，就育才問題，將利用代理人協商的行爲模式，將其中特定代理人導入 BDI(Belief, Desire, Intention)模式，讓代理人依照各自的 BDI 模式以協商化的概念將代理人配置於適當的位置執行專案。
4. 以案例實際模擬協商成功及不成功，所會產生的結果及機制，並做分析。

1.4 研究限制

此研究仍然有很多限制及不足之處，如：

- (1) 如專案複雜度太高，可能造成系統思緒並無人類可以考量的那麼縝密，容易造成區域最佳解的情形。
- (2) 本研究只考量總體利益，如專案本身屬特殊案例，例如說必須在某期限完成，可考慮高成本等問題，亦不在此研究之內。
- (3) 本研究已將各式因素數值化，並且以總成本的方式判斷專案的優劣，如有其它無法數值化的因素，亦不在本研究之內。

1.5 研究方法及論文結構

圖 1.2 及圖 1.3 為本論文之主要流程圖及論文結構，其描述如下：

- (1) 研究問題分析：
希望可以創造由代理人協商系統來幫助人類在專案管理中可以進行更精密的分析，而產生更趨於完美的專案，並且可在專案管理進行時擔任監視與控制的任務，隨時監控專案是否如期如預算的進行。
- (2) 文獻探討：
對與研究相關的文獻做探討，包含基因演算法、成本效益法、專案管理 (PMP)、JADE。
- (3) 建立協商系統架構：
使用成本效益法進行協商，找出最適專案，並引用基因演算法的進化理論，在專案管理者無法提出更好的提案時，代理人會啓用遺傳演算法，進行最佳子代的演化。並在專案管理進行中使用 EVM (Earned Value Management) 監控專案進度，如遇到無法如期如預算進行的緊急狀況時，也會啓動代理人協商機制。
- (4) 進行模擬：
建立雛型系統，進行實例模擬。
- (5) 結論分析與後續發展

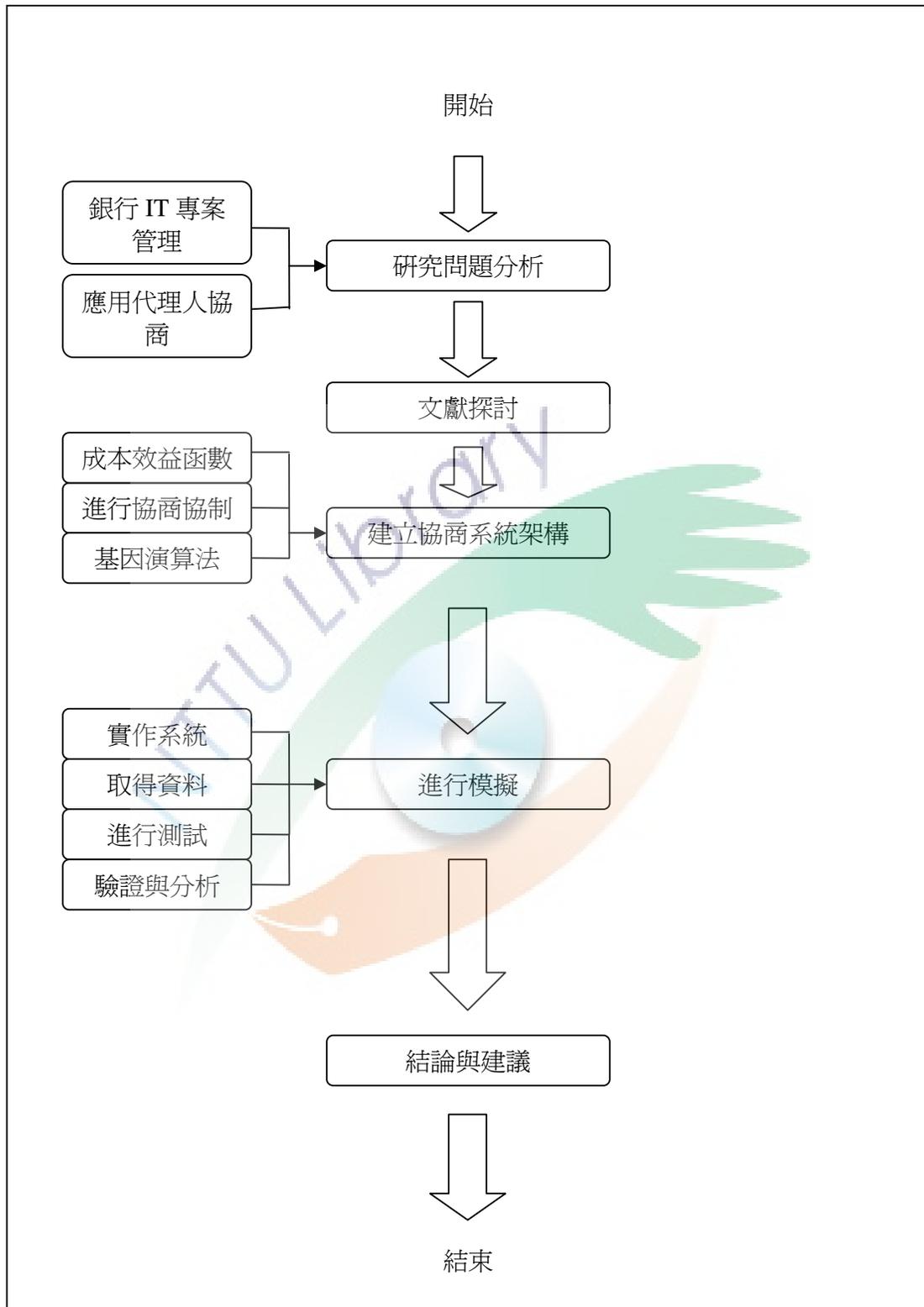


圖 1.2 本研究的主要流程圖

第二章 文獻探討

有關代理人協商的探討文獻，國內外都有許多不同領域的論文研究，這類的問題最主要是解決在電子商務中所遇到的各式問題。因此本研究預定運用專案管理的領域配合代理人協商機制實作一雛型系統，進行協商機制的模擬。

2.1 專案定義

什麼是專案？Project Management Institute(2008)所提出的”PROJECT MANAGEMENT BODY OF KNOWLEDGE”一書中說明了專案的定義。專案可以說是暫時被賦與責任去創造一個獨特的產品、服務或結果。專案本身也具有明確的開始與結束，而專案結束指的是當這個專案的目的已經達到，或是因為目的不能達到或是專案的需求已不存在而中止計畫。在此特別說明一點，專案的特性之一”暫時”並不是指專案的時間很短暫而且也不適用於其產生的產品、服務、或是結果。大部分專案通常可得到一個持久的結果。舉例來說，建一座國父紀念館的影響是好幾世紀，專案也常常會因社會，經濟和環境上的衝擊，產出之結果所造成影響的時間也比專案本身時間長很多。每個專案會創造獨一無二的產品、服務、結果。雖然重複性的元素可能會出現在某些專案中，但這種重複並不會改變每一個專案的獨特性。舉例來說：辦公大樓即使是同一團隊以相同或相似的材質建造，但不同的建築物還是有不同的設計、環境、契約等等。

一個正在進行的工作通常都會反覆重做，因為它是按照一個組織內既存的程式在進行。相反地，由於專案具有獨特性的本質，所以可以新建一個新的團隊來參與專案，執行專案計畫。另外再補充點的是，各種組織都可被賦與專案任務，一個專案可被一個組織介入也可有多個組織參與，就看專案執行所需的資源而定。專案的成立可以有很多樣貌呈現，例如：專案可能是一個產品的部份或是全部，有執行服務的能力(如：有支援產品或銷售的商業功能)，或是結果、文件(例如：研究項目，開發知識，可用於確定是否為一種趨勢或一個新的進展，將有利

於社會)。

2.2 專案管理定義

什麼是專案管理？Project Management Institute(2008)所提出的” PROJECT MANAGEMENT BODY OF KNOWLEDGE”一書中說明了專案管理的定義。專案管理是應用知識、技能、工具、技術等項目來達成專案需求。專案管理是應用 42 個專案管理邏輯程序加上 5 個程序。這 5 個程序包含了有：

- 起始(Initiating),
- 計畫(Planning),
- 執行(Executing),
- 監督與控制(Monitoring and Controlling), and
- 結束(Closing)

而典型的專案管理包括了：

- 定義需求(Identifying requirements),
- 策劃及執行專案計畫(Addressing the various needs, concerns, and expectations of the stakeholders as the project is planned and carried out),
- 平衡專案之競爭因素(Balancing the competing project constraints including), but not limited to:
 - ◇ 範疇(Scope),
 - ◇ 品質(Quality),
 - ◇ 時程(Schedule),
 - ◇ 預算(Budget),
 - ◇ 資源(Resources), and
 - ◇ 風險(Risk).

這之間的關係是當其中一個元素改變時，將會影響至少一個元素的變更。例如當專案時程縮短時，通常預算就要增加以補充資源，使專案可以順利在這麼短的

時間內完成。假設不能增加預算，那範疇和品質也許會受到影響。當專案的需求方改變專案需求，可能會造成額外的風險，所以爲了讓專案可以順利進行及完成，專案管理小組必須評估風險及可行性。

由於有潛在的變化，在整個專案管理生命週期，專案管理計畫是可以經過反復及逐步制定的。逐步制定包括了提升專案計畫、提供更詳細及具體的資訊，並且更準確的估計目標達成所需的資源。

Project Management Institute(2008)提出，專案整合管理包括必要的程序和活動以辨認、定義、結合、統一和協調在專案管理程序內各類的程序和專案管理活動。很顯然地，專案管理整合對各個程序上的互動及影響是非常需要的。舉例來說，對一個意外增加的成本，在估計上就有成本、時間及風險等區域上的考量。對於專案的交付，也許需要結合正在發展中的業務組織、客戶組織、一個長期的計畫，這些在未來都要納入伴隨而來的問題和機會而加以考量，專案整合管理也包括所需的專案文件以確認專案管理的組成及產品的交付。以下圖 2.1 爲專案整合管理概要說明及專案整合管理圖解如下：

專案整合管理(Project Integratio Management)

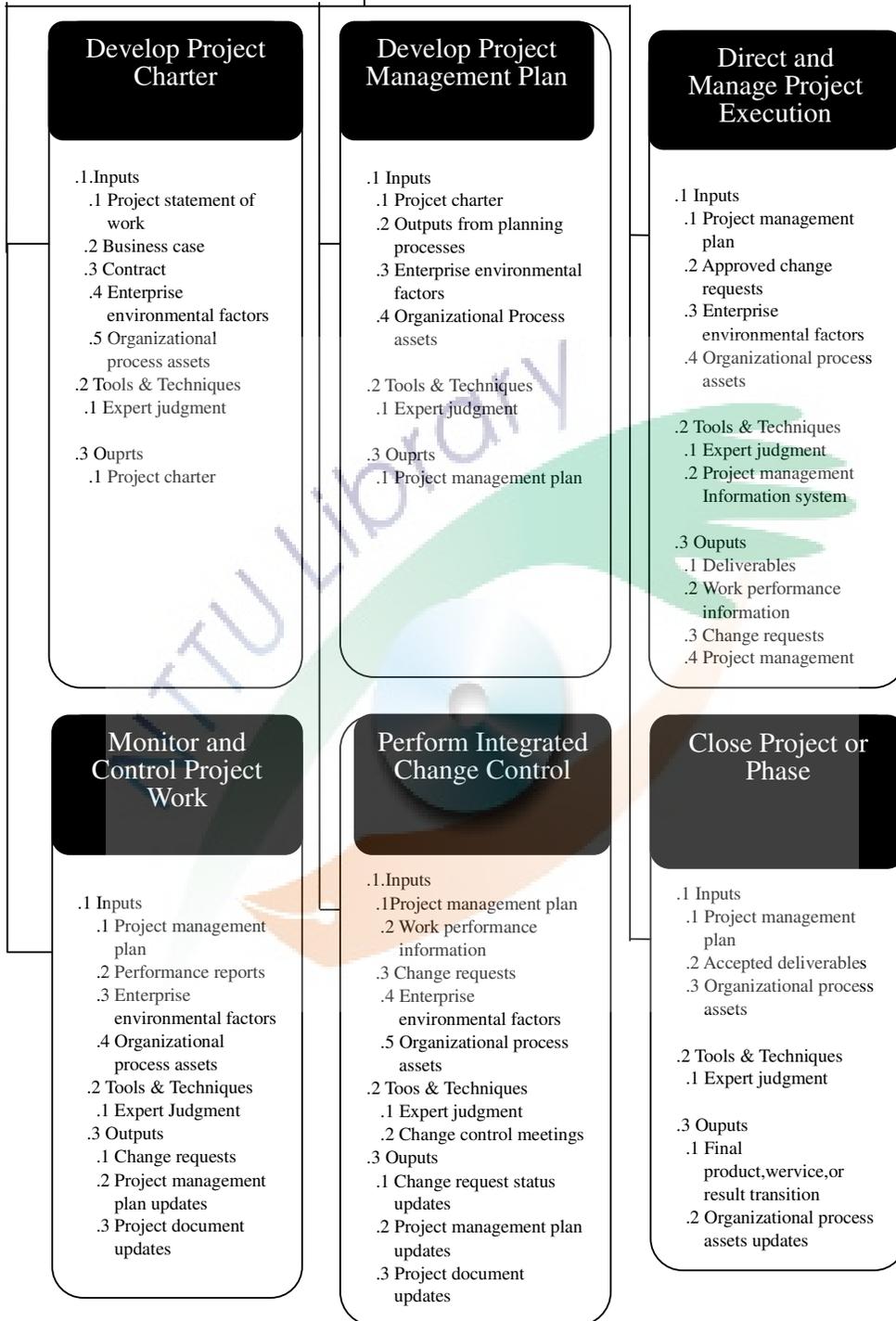


圖 2.1 專案整合管理圖(Project Integration Management Overview)

資料來源: 「Project Management Institute , 2008」

根據上述圖解，專案整合流程共有以下六階段在進行：

1. 專案初始化(Develop Project Charter)－文件形成的過程，文件的內容是正式被核准一個企劃以及紀錄著一開始就滿足資金擁有者的需求其期盼的條件。
2. 專案計畫制定(Develop Project Management Plan)－是一種計畫制定的過程，並準備、整合以及協調所有次要的計畫。
3. 執行(Direct and Manage Project Execution)－執行專案計畫。
4. 監督與控制(Monitor and Control project Work)－追蹤、重新探討、控制計畫的進行和專案管理計畫所訂下的目標。
5. 變動(Perform Integrated Change Control)－重新檢視所有需求變動，考量變動的因素，並做專案計畫的調整。
6. 完成(Close Project or Phase)－計畫正式完成。

2.2.1 專案風險及影響定義

一般定義，品質和可信度的風險進行定性分析的過程需要不同層次的風險的概率和影響加以界定。在計畫風險管理程序，用於執行定性風險分析過程。表 2.1 分別針對四種專案目標定義，說明在不同層度的風險下，這四種專案目標會怎麼做決定。假設在風險程度很低的狀況下，對於成本、時間、範疇、品質所受到的影響非常小，不會影響專案進度。假設情況為中等風險程度，對於成本的影響為增加 10-20%，時間的影響為增加 5-10%，對於範疇的影響謹為主要區域，並不影響全部。假設情況是高度風險，對於這四種專案目標的影響就非常大大，最不想預見的情況，可能就是停止專案的進行。專案管理者必須在專案計畫建立時，就要全面考量，專案進行時，也必須隨時監控進度，儘量避免這種高度風險的情況發生。以下為專案管理風險影響定義表 2.1，可提供專案管理者在做風險評估的範本：

表 2.1 四種專案目標對專案管理的影響定義表

Defined Conditions for Impact Scales of a Risk on Major Project Objectives (Examples are shown for negative impacts only)					
Project Objective	Relative or numerical scales are shown				
	Very low/0.5	Low /.10	Moderate /.20	High /.40	Very high /.80
Cost	Insignificant cost increase	<10% cost increase	10~20% cost increase	20-40% cost increase	>40% cost increase
Time	Scope decrease barely noticeable	<5% time increase	5-10% time increase	10-20% time increase	>20% time increase
Scope	Scope decrease barely noticeable	Minor areas of scope affected	Major areas of scope affected	Scope reduction unacceptable to sponsor	Project end item is effectively useless
Quality	Quality degradation barely noticeable	Only very demanding applications are affected	Quality reduction requires sponsor approval	Quality reduction unacceptable to sponsor	Project end item is effectively useless
This table presents examples of risk impact definitions for four different project objectives. They should be tailored in the Risk Management Planning process to the individual project and to the organization's risk threshold. Impact definitions can be developed for opportunities in a similar way.					

資料來源:「Project Management Institute, 2008」

2.2.2 專案成本管理圖-Control Costs

PMI(2008)提出專案在進行時，專案管理者會隨時監控及控制成本的支出，如遇突發狀況或無法如預算執行，將會採取增加預算及做成本基準線變更。圖 2.2 說明了在評估成本基準、績效報告或變更請求時，如遇變動，將會透過成本控制評量，然後再產出新的專案文件及更新成本基準等文件。這部份最主要是在控制專案進行中，如遇突發狀況，將會重新評估成本，以利專案順利進行。

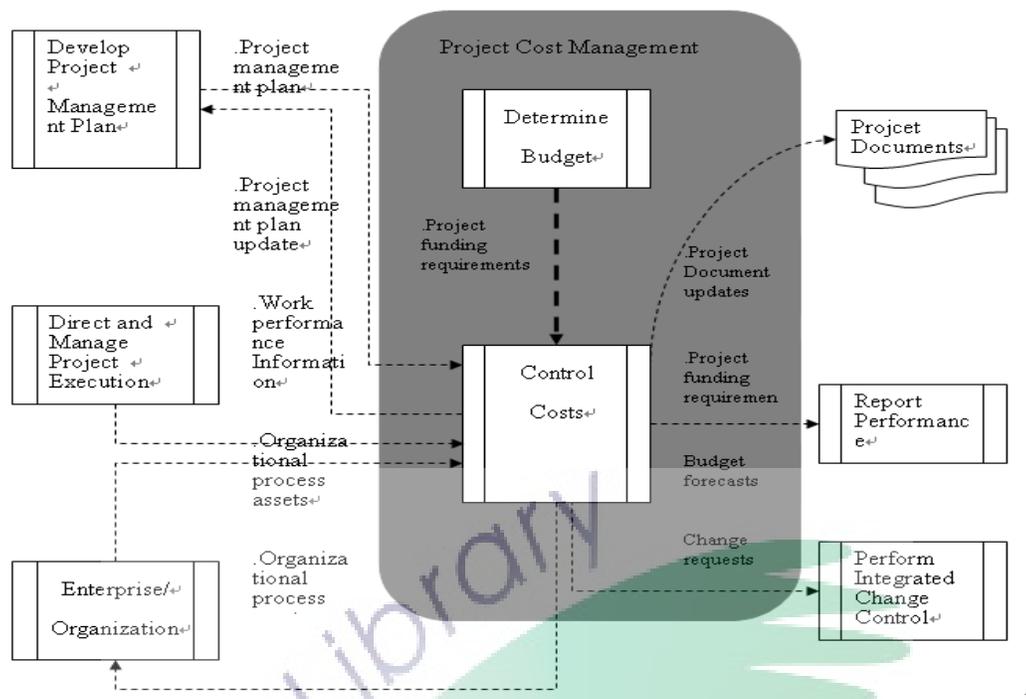


圖 2.2 專案管理之成本控制圖

資料來源:「Project Management Institute, 2008」

2.3 BDI 智慧型代理人與多代理人理論

多重代理人的概念與人類社會的概念相當接近，Zambonelli et al(2000)指出，多重代理人系統像人類社會一樣，它其實就是一個代理人社會，代理人彼此間有組織性或互動關係，彼此有溝通及各自的目標，能對事件作出合理的反應。

綜合不同的學者所提出的見解，BDI 代理人的態度分別有信念、意欲、意圖、計畫、解譯來表示(Rao et al, 1995; Thomas, 1995; Wooldridge et al)。此 BDI 架構為一實務推論的架構，而所謂的實務推論架構就是說其推論過程如人類社群中相仿的決策過程。以下圖 2.3 所表示的核心架構以解譯為主要態度，功用是控制整個代理人的週期運算，首先會以目前環境的情況來更新信念，接著為了滿足環境事實而選出需求集合，之後配合合適的計畫，執行意圖來完成此活動。整個週期架構說明如下：

- (1)信念函數(Beliefs)：在集合中的信念是顯示代理人對目前環境所感知的資訊。
- (2)意欲(Desires)集合：此集合代表代理人可以採取的行動。
- (3)意圖(Intention)集合：意圖代表代理人目前強調之所在，並且這正是代理人當試要實現的、達成的。
- (4)計畫(Plans)：使用者之計畫。
- (5)解譯 (Interpreter)：將投入的信念函數、意欲、意圖、計畫等元素，進行解譯。

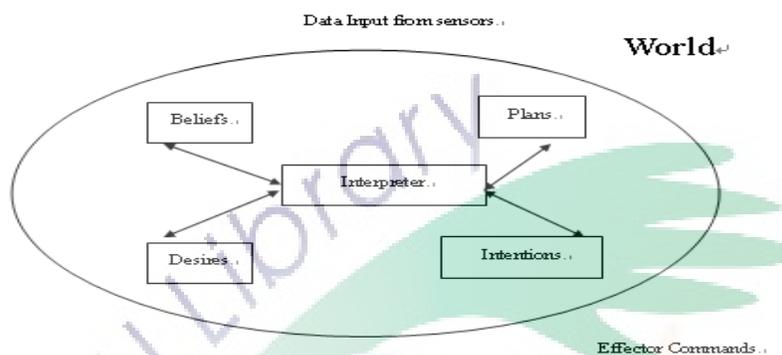


圖 2.3 BDI 代理人架構圖

資料來源:「Georgeff&Lansky, 1987」

這是 BDI 代理人的實務運作。以上這個流程將會運用在本研究所提出之協商機制中。

2.3.1 代理人和多重代理人的特色

Mase(1994)認為「代理人」是一個在動態環境下展現各式複雜行為的計算系統，它可以偵察出環境變化並做出適切的反應，此外代理人可透過執行一連串活動，實現它的目標和動機。一般而言，代理人經常具備以下特性（可只具備其部份特性，不須同時具備所有的特性）：

1. 自發性(Autonomy)

代理人經委派後，不需自然人介入便能自行作業。

2. 社會性(Social Ability)

代理人可和其它各類代理人或自然人互動。

3. 感知性(Pro-activeness)

代理人會主動採取目標導向(goal-directed)的行動，它們會確認目前的意圖並具有確信週遭環境的信念(belief)

4. 反應性(Reactivity)

代理人會察覺環境變化或發生的事件，並適時做出反應。

5. 移動性(Mobility)

代理人可移動到其它環境(例如：透過網路甲電腦移到乙電腦)，且能帶著資料移動。它們可任意漫遊或依照擬定的行程，逐一移動。

6. 脈絡連續(Temporal Continuity)

代理人執行任務時，會持續掌握程序(process)或執行緒(thread)，而不是只做單一運算(one shot)。簡言之，代理人會自己擁有運算資源，以隨意進行各項行動。

7. 適應性(Adaptability)

代理人能透過評斷行為的結果進行學習，因此會隨時間和環境變化調整自己的狀態。

因此根據上述代理人所具備的特色，可將代理人作基本的分類，以釐清所要發展的代理人為何，及其所應具備的功能與特性。

San(1998)所提出對代理人的分類如下：

1. 靜態及行動代理人(Static and Mobile Agents)

依代理人的行動能力(mobility)來看，可分為靜態與行動代理人兩類。靜態代理人固定在一部電腦中執行它的特定工作，以傳統軟體的訊息交換方式來與外界溝通；而動態代理人則會透過網路，游走於其他電腦執行任務，最後回到原動腦，常用於網路上的資訊搜集。

2. 界面代理人(Interface Agents)

界面代理人(Interface Agents)的主要功能是降低界面複雜度，以提高易用性。它提供智慧型(smart)的使用者界面，即時協助使用者解決操作上的問題。

3. 協商代理人(Collaborative Agents)

此種代理人具有與其它代理人共同運作的能力，可以各自分工完成特定的工作，然後再將結果整合起來。

4. 資訊/網路代理人(Information/Internet Agents)

協助使用者過濾網際網路中龐大的資訊，僅將使用者所需要的資訊呈現給使用者。

5. 學習代理人(Learning Agents)

這種代理人具有機器學習(machine learning)能力，可藉由不斷學習外界傳來的資訊改變作業方式，增進執行效率。

2.3.2 多代理人系統

除了以單一代理人為核心的資訊系統外，也可由多個代理人組成更複雜的多重代理人系統(Multi-Agent System)。多重代理人系統是由許多獨立的代理人所組成，整個資訊系統呈現分散式型態。系統效能的優劣，取決於代理人組織架構和協商機制是否設計得當。此外 Dufee(1989)觀察到許多應用系統本質上都有分散式的特色，多重代理人系統適合應用在功能分散，而各個子系統必須群策群力的情形。例如：一群專長不同的專家彼此合作去解決某項困難問題。整體而言，多重代理人有以下優點：

1. 模組性/可擴充性(Scalable)的架構

代理人是獨立的個體，易組合和擴充。每個代理人可隨時加入系統並和其它代理人一同工作，當任務完成時便離開系統而不影響系統的運作。模組化和具擴充性的系統可滿足未來系統漸進式的成長和擴充。

2. 平行處理效能

動態聚集不同能力的代理人以同時處理問題，可共用系統運算資源和展現各代理人獨立執行緒的處理能力。面對分散式處理環境，多重代理人系統的效能會較單一集中式系統更有效率。

3. 自我組織的系統(Self-Configuring System)

一群代理人可在執行時期自行調整內部狀態，以迎合系統面臨各種情況時的

變異。代理人的行動展現類似於自然人，但較自然人具有更佳的記憶、運算能力、毅力、和耐力。除了使系統運算結果接近於真實世界外，也可產生較佳的結果。由於整體系統的行爲來自於各個不同區域或領域代理人的決策，系統可自動隨時間變遷，主動調整系統內的代理人，展現自我組織的特性。

4. 容錯性(Fault-Tolerance)

由於多重代理人系統採用分散式運算，因此當某個代理人發生錯誤或缺席時，可由其它功能類似的代理人更換。換句話說，使用多重代理人系統能降低系統的整體錯誤率。Jennings et al.(1998)認為多重代理人系統具備如下特性：

- (1) 每個代理人擁有解決問題的部份資訊或能力。
- (2) 無整體系統控制能力。
- (3) 資料是非雙中式(Decentralized)。
- (4) 採用非同步(asynchronous)的運算機制。

2.3.3 代理人協商

Nwana et al.(1996)將多重代理人協同合作(coordination)分爲四大類：

1. 組織構建式(organizational structuring)
2. 合約式(contracting)
3. 多重代理人規劃式(multi agent planning)
4. 協商式(negotiation)

Huhns 與 Stephens(1999)指出，多重代理人在互動時，由於所處的環境可能是分散的，各個代理人的目標也可能不盡相同，彼此的行爲可能產生衝突。爲使代理人系統能有一致(coherent)的表現，代理人透過溝通(communication)進行協同合作，以協調彼此的活動與行爲，協同合作方式可分爲合作式(cooperation)及競爭式(competition)二種：在代理人各自具有利害關係(self-interested)的競爭式的協同合作模式中，又以協商的運作機制爲主。

協商理論是由早期經濟學理論中的賽局理論所發展出來的一門分支科學，深深影響人們合作與衝突解決的經濟活動與行為。Raiffa(1982)的著作” The art and science of negotiation” 仍是這領域的代表作，其分別討論兩方協商者(two parties)對於一個議題(issue)、兩方協商者對於多個議題、多方協商者(many parties)對於多個議題情況下，協商所適用的策略；而影響協商的主要因素包括：議題的關聯性、協商次數、協商人數、聯盟(coalition)等。他並指出協商過程需注意下列四個階段的互動：

1. 協商準備(preparing for negotiations)。
2. 開始行動(opening gambits)。
3. 協商與讓步(negotiation dance)。
4. 結束協商(end play)。

即使是個簡單的協商，所達成的協議也常常是次佳的結果。Oliver(1997)研究指出，協商為參與者同時於多維度(multi-dimension)空間搜尋的過程，然後對空間中的某一點達成共識。空間中的維度相當於協商的議題，每個議題可視為二個以上經過排序後的解決方案的集合。而多重代理人的協商亦是一種代理人群體合作及衝突解決的重要方法，代理人各自具有不同目標及利害關係，為了達成自己的目標，他們會彼此溝通的立場及條件，如果發生目標衝突時，則採取讓步策略及其他方法，以更達成共同可接受的協議(agreement)。Kraus(2001)認為代理人針對協商議題採取何種協商策略、如何進行提案溝通、進而達成協議的策略協商模式(strategic-negotiation model)，主要包含下列三部份：

1. 協商協定(protocol)：代理人在協商時，如何提出提案、回應提案、接受或拒絕案、乃至協商結束等整個協商的進行方式。
2. 代理人效用函數(utility functions)：效用函數是代理人在協商時，對於許多不同協議的評估準則。
3. 協商策略(strategies)：協商策略明確說明代理人在協商時，對於不同的處理，如何採取適當的協商行動因應。

Chun 與 Wong (2003)提出一個以代理人為基礎的自動協商架構及演算法，此架構包含了使用者偏好莫式、評估函數、協商協定等相關協商運作的作業模組，以解決專案執行中動態排程的問題。Paulson et al.(2003)針對分散式專案執行過程中，提出一個分散式代理人自動協商的協同運作，以效用移轉的方式，解決各個分包商因資源不足所發生的時程衝突問題。

2.4 協商理論

2.4.1 協商及談判的定義

Lewicki 與 Litterer(1985)認為談判在本質上是一種面對面相互交換、創造及讓步的過程，其目的即在雙方共同努力解決問題，因此積極的說，談判就是使彼此之間能獲得最大的總利益。從消極面來看，談判是彼此讓步以解決爭端的交換過程，換句話說，Druckman(1977)認為談判是一個漸進聚合的過程。Anson 與 Jelasi(1990)將協商定義為解決兩個或兩個以上的成員彼此動突所進行的一連串程序。Lai(1989)則將協商所具有的特定整理如表 2.2 所示：

表 2.2 協商特性

特性	說明
涉及兩個或兩個以上的協商成員	協商活動中至少有兩個成員，當涉及的成員越多，協商過程便越複雜
存在著明顯或潛在衝突或是共同利益	協商成員之間必須要有某種程度以上的衝突或共同利益存在，才能引發協商活動。
協商成員彼此互有依賴性	協商成員需共同努力，才能解決存在於彼此間的衝突，以取得可能的共同利益。
共同解決問題的意願	若協商成員完全沒有共同解決問題的

	意願，則不會有協商活動的空間。
--	-----------------

資料來源：「Lai，1996」

根據上述的協商特性，協商是可以有多位成員參與的，各個成員對相關屬性皆有不同偏好，在協商活動中也會針對各個成員的不同偏好進行調整，在過程中，會有衝突的產生，也會有相互依賴性質的共同利益產生。因此，解決成員間的衝突，以達成協商最大利益是協商的主要目標。本研究將利用上述協商特性，提出一個可以達成 win-win 目標的代理人協商模式，將其應用於專案管理上，並建構雛形系統以驗證此理論的可行性。

2.4.2 協商類型

依照協商者面對協商問題時其參與態度的不同，而可以將協商分為下列兩種類型，由 Bazerman(1986)所提出之兩大協商類型：

1. 分配型協商(Distribution in Negotiation)

所謂分配型協商指的是協商雙方在互相衝突的目標上進行協商的過程。在此協商過程中，協商的雙方是站在對立的關係來思考，若是有一方獲得的利益將會使得另一方利益減少。從賽局理論(game theory)的角度來看，這就是一個零和的遊戲(zero-sum game)。這一個類型的協商關鍵技巧在於估計推測對方底線的點落在哪裡，尋找一個剛好可以讓對方接受的底線，將會使得此次的協商遭致失敗。在這樣的情形之下，如果不是贏就是輸(win-loss game)，將會使得協商結果很難使雙方獲得滿意，減少成功的機會。

2. 整合型協商(Integration in Negotiation)

相對於分配型協商在對立地位上的競爭，整合型協商站在合作的角度上做協商考量。由於此彼對於協商之相關屬性存有不同偏好，賽局的結構已經改變。協商的雙方要找到彼此最大價值的區塊(positive bargaining zone)，就不能只是侷限在每一個議題上都要得到利益，而必須明白的說出自己較重視的

屬性有哪些。有了這些資訊，交易的雙方就能夠在這些議題上做利益交換 (tradeoff)，而找到一個整合的解決方案。由於交易雙方存在互惠的彈性，類似合作的整合型協商，理論上能促成更多成功的協商，也使得彼此的滿意度能提升，達到雙贏的局面。

不同的協商策略會產生不同的結果，根據 McBurney(2003)所提出之觀點，分別將協商定義為獨裁式、半民主式、民主式。如表 2.3 將獨裁式解釋為是可讓決策者得到所有參與者的建議，並擁有主控權，因此決策的準確度也較高，但卻也讓其他參與者無法表達其意見。本研究將採用整合型協商為代理人協商的基礎，目的是要達到 win-win 的結果，因此就不適合使用中央集權的獨裁方式做決策。半民主的協商著重於，只能以現有的意見來做選擇，不讓參與者有反駁的機會，以整合性協商而言，半民主式協商並不適用。而民主式的協商是所有參與者皆可對提案提出正反意見，依照各個參與者所重視的屬性進行協商。不斷地反覆調整，直到所有參與者都能接受該提案並且達到最大共同利益。本研究將會整合以上所提出之協商策略，利用整合型協商及民主式策略協商進行代理人協商應用於銀行 IT 專案管理之研究。

表 2.3 代理人協商讓步策略

協商方式	內 容	問 題
獨裁式	使用中央集權來完成整體的決策。決策者不須透過協商過程就能獲得所有的資訊	有較高的準確度，可是參與的代理人失去了意見表達的能力
半民主式	參與的協商者只能針對目前有的提案做選擇的，無法提出反駁，或進行修改提案的動作	減少代理人相互溝通的成本，快速找出彼此共識。但在意見表達上，只能對於現有的提案做選擇，無法表達更多意見
民主式	參與者可以對決策提出的提案反駁，表達自己的意見。決策者也會依據參與者所提出的意見，對提案做適當修改，透過不斷的協商達到共識	需要大量的雙方資訊交流，在溝通過程花費時間較多成本較高

資料來源:「王亮宗，2007，頁 8」

2.4.3 協商機制

對於協商的過程，目前己有很多學者各自提出他們的定義，Raiffa(1982)就曾提出協商過程共有四階段，分別為：(1)協商準備(preparing for negotiation)：提出各種方案供參與者選擇，定義各自想要達成的目標，及進行各種談判模擬的情境；(2)賽局開始(opening gambit)：決定由誰開始提議賽局，並且篩選出不合理的提案決定後續處理方式；(3)讓步策略(concessing)：如果提案沒有交集而形成差距時，必須考慮讓步的次數、幅度，並同時要預測對方的底限，來做調整；(4)談判結束(end-play)：對於結果的接受，協商者應謹慎地決策，對於協商不成立時，應如何做後續處理。而關於讓步幅度，Raiffa(1982)曾提出協商空間(zone of agreement)的觀念，如圖 2.4 所表示，在產品的價格分布上，買、賣雙方各會保留其價格底限，假設最後的價格成交是落在 x 。那 x 與 s 的差價就是賣方所得的利益。買方利益所得依此類推。

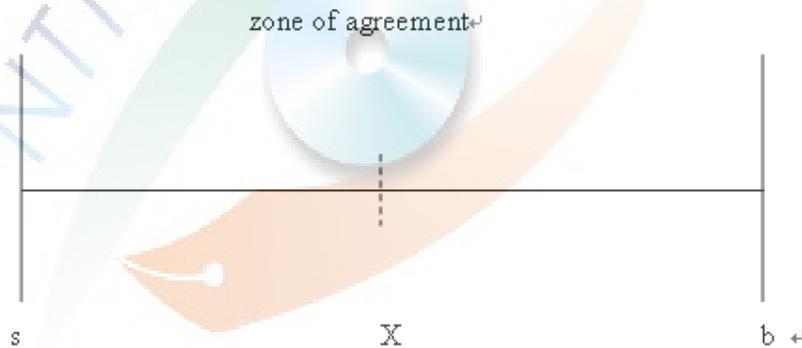


圖 2.4 協商空間觀念圖

資料來源：H. Raiffa, "The Art and Science of Negotiation" Cambridge, MA: Harvard Univ. Press, 1982.

根據楊子青與林啓明(2003)在論文中提出協商活動可看成一個程序，並依照過去文獻提出了(1)問題定義;(2)屬性處理：計算協商空間、協商空間移動和讓步策略的變動；(3)協商評估：進行協商提案的評估，進行協商處理。可接受的讓步策略則在於如何找出交集的點，根據 Liang 等人(1999)提出的理論，議價協商可分為

效用遞增策略(utility increasing strategy)、效用遞減策略(utility decreasing strategy)及效用平均策略(utility neutral strategy)三種策略，之外還還有學者也提出其它理論，大致可分為遞增型、遞減型、固定型、隨機型及互惠型(Raiffa,1989;Kowalczyk,2000)。

在代理人協商探討部份，目前已應用在很多面向，如 zhao 等人(2001)所探討的雙向線上協商模式(Model of bilateral online Negotiation)，主要是兩個代理人間進行和時間無關係的合作雙向協商；朱宏揚(2000)所提出的自動化整合協商(An Automated Integrative Negotiation Agent)，主要是針對電子商務的零售市場，買賣雙方之間所進行的交易協商行為進行自動化的協商。Barbuceanu 等人(2000)提出一個多屬性效用定理(Multi-Attribute Utility Theory)的方式來達到商品的協議。舉例來說，買、賣雙方會先設定低、中、高的範圍效用值，再根據效用值產生各種提案。楊子青與林啓明(2003)所提出之線上協商支援系統，主要是利用協商代理人特性，人們可以不用同時上線即可進行協商，並指定協商模式進行協調，以達到協商合作的目的。

綜合以上文獻所提的協商機制，較少提到利用代理人幫助參與者進行專案管理的協商支援，本篇論文將提出一個代理人架構，藉由此架構完成專案管理的協商。

2.5 成本效益分析法

2.5.1 成本效益分析的理論基礎

Key(1940)曾提出一值得深思的問題：「在什麼基礎上決定分配 X 元於 A 活動而非 B 活動？」(On what basis shall it be decided to allocate X dollars to activity A instead of B?)成本效益分析的理論基礎源自於經濟學的概念，強調於效率概念上，分配資源給創造最高效率的活動與計畫。成本效益分析是經濟可行性分析之工具，其原理在於對方案、計畫作全面性的分析，以求取資源最有效

的運用。

由經濟學的角度觀之，即在求取最大的淨社會效益 (Net Social Benefit)。而求取淨社會效益最大的最終目標下，假定所支付的社會成本固定，則求取最大的社會效益；反之，當社會效益固定時，則求取最小的社會成本。同理，當社會成本及社會效益均不固定時，則以追求社會利益大於社會成本的程度，來判斷資源運用的效率。

2.5.2 成本效益分析的計算方式

Weimer 與 Vining (1992)指出成本效益分析為「以貨幣單位來估算各種政策影響的成本和效益」。關切的重點在於：如何計算成本與效益？

表 2.4 成本效益分析計算方式

計算方式	公式	決策法則	優先次序
淨現值法	$\sum_{t=0}^n \frac{Bt - Ct}{(1+i)^t}$ 或 $\sum_{t=0}^n \frac{Bt}{(1+i)^t} - \sum_{t=0}^n \frac{Ct}{(1+i)^t}$	大於 0 者可投資	正值越大越優先
益本比法	$\frac{\sum_{t=0}^n \frac{Bt}{(1+i)^t}}{\sum_{t=0}^n \frac{Ct}{(1+i)^t}}$	大於 1 者可投資	以增量分析判斷
內部報酬率法	$\sum_{t=0}^n \frac{Bt - Ct}{(1+i)^t} = 0$	$r > \text{MARR}$ 可投資	差距越大越優先
還本期法	$\sum_{t=0}^n \frac{Bt - Ct}{(1+i)^t}$	大於 0 者可投資	正值越小越優先
年值法	$\left(\frac{A}{P, i, N} \right) \times \text{NPV}$	大於 0 者可投資	正值越大越優先

終值法	$\left(\frac{F}{P, i, N}\right) \times NPV$	大於 0 者可投資	正值越大越優先
外部報酬率法	$\sum_{t=0}^n B_j \times (1 + MARR)^{n-1}$ $= \sum_{t=0}^n C_j \times (1 + ERR)^{n-1}$	$r > MARR$ 可投資	差距越大越優先
淨效益－成本比法	$\frac{B - C}{C}$	大於 0 者可投資	以增量分析判斷
盈餘－投資比法	$\frac{P(E)}{P(I) - P(S)}$	大於 1 者可投資	以增量分析判斷

資料來源：「何毓芬，1997」

2.5.3 時間、風險和不確定性

Weimer 與 Vining(1992)指出成本效益分析為：處理時間、風險和不確定性，考慮的問題為：時間因素影響下，如何選擇折現率？風險與不確定性如何處理？是否須要執行敏感度分析？

(1) 折現率的選擇

Weimer 與 Vining (1992)提出，折現率的選擇在成本效益分析的計算過程中相當重要，所謂折現率的意義在於社會大眾對未來與現在所得的時間偏好。由於政策或計畫的成本與效益可能延續非常長久，因此有必要計算不同時間的成本與效益的現值，也就是將不同時間的成本與效益折現至同一時點，始能進行成本與效益的比較。

(2) 風險與不確定性

Dorfman(1972)認為對於無法客觀(objectively)知道機率的不確定狀況，是可以轉換成主觀機率(subjective probability)，沈熾林(1990)提出根據事件發生的機率與規模大小，可計算成本與效益的期望值，但是期望值無法表示事件發

生的變異性與範圍，而且期望值無法顯示個人對風險的偏好，使成本效益分析面臨困境，不論是有客觀機率存在的風險狀況或可主觀認定其機率的不確定性時，在進行成本效益分析時都面臨處理的困境？站在實用角度的立場來看，如果隨機事件(random event)是造成不確定的主要來源，事實上是很難以一個有意義之機率納入分析，因此將其忽略是最直接簡單的方式，誠如 Arrow 與 Lind(1970)所言，公共政策之決定本來就是反應每個人之預期成本(expected cost)與預期效益(expected benefit)，且每個人為風險付出之成本微不足道，因而政府對這種不確定性是可以完全被忽略的(蕭代基等，2002)

2.5.4 預算限制和分配效果

Weimer 與 Vining(1992)指出成本效益分析為考量預算限制與分配效果，以選擇最適當的政策。

2.5.5 成本效益分析與成本效能分析的比較

成本效能分析對政策或計畫所牽涉到之生命價值的評估是由決策者來評判，而在成本效益分析上，這部分的效益評估是由分析者來完成。因此，從事成本有效性分析的分析人員較偏重於技術效率(technological efficiency)，與效益有關之價值判斷則留給決策者，而進行成本效益分析的分析人員，則是採用一種同時考慮成本與效益之經濟效率的分析原則(蕭代基等，2002; Nas,1996)。

2.6 JADE 代理人發展環境

智慧代理人是人工智慧最重要的研究領域之一智慧代理人是軟體實體，能執行相關作業在使用者及其他程式上，其擁有相當程度的獨立性與自主性，使用相關知識或代理人來達到使用者的目標與要求，學者 Wooldridge 與 Jennings(1995)提出：提供代理人是有分「弱」、「強」的兩個觀念。前者是指代理人系硬體或以軟體為基礎的電腦系統，而後者是指包括較弱定義加上如同人類感情的性質等概念。

2.6.1 代理人架構種類

學者 Wooldridge 與 Jennings(1995)等人將現行的代理人架構分為邏輯式、反應式、BDI 式、階層式等四大類。以下為四大類架構的簡介：

1. 邏輯式代理人架構：

為目前最傳統的代理人架構。其以邏輯為基礎的方法，透過符號來表達知識並且進行邏輯的推論與證明，然後產生一個代理人系統。

2. 反應式代理人架構：

代理人決策是透過一群已經實作的行為來實現：相對於邏輯式代理人，沒有複雜的符號來表與符號推論，而僅是簡單的輸入對應輸出的關係，因此，可以說是狀態決定行動。

3. 階層式代理人架構：

在代理人「反射」與「主動積極」行為能力的需求，建立分離的系統來處理不同行為的概念。且子系統是經由自然而然的分類與互動，逐漸產生層級的觀念。

4. BDI 代理人架構：

說明在此架構下，代理人具備信念、意欲與意圖的資料結構，以及表示他的深思熟慮和最終使用手法。而其中意圖扮演 BDI 最重要的核心角色，其提供決策的穩定性及實際的行動與行為。

2.6.2 JADE 簡介

JADE(Java Agent Development Framework)是一個代理人軟體發展框架，由 TILAB(Telecom Italia Lab)及 University of Parma 所合作開發的，他適用於開發智慧型多代理人系統和應用程式。JADE 提供了一系列的函式庫和類別讓程式開

發人員能夠很方便的開發屬於他們的多代理人系統。JADE 完全以 Java 語言開發而成，所以代理人設計者便可以 Java 語言來撰寫其代理人程式。在 JADE 套件中包含兩個主要的套件：(1)一個符合 FIPA 標準的代理人平台，提供執行時期的環境讓使用 JADE 開發的代理人能夠存活在使用者的系統中。(2)一系列的函式庫和類別，讓程式開發人員能夠很方便的使用 JAVA 開發屬於他們的多代理人系統。此外 JADE 平台還提供了一系列的圖形化工具讓使用者可以管理及監視正在執行的代理人程式。GUE 的環境提供代理人開發者進行遠端主機上，代理人的產生及啟動代理人的執行。

從圖 2.5 可以得知，JADE 多代理人平台可以分散於許多不同主機，每台主機必須有 JAVA 虛擬機器在執行。代理人系統執時必須在一個主要容器（Jade Main-container），在 JADE Main-container 上主要是提供代理人的 AMS、DF 及 ACC 服務，Main-container 內允許多個代理人執行。除了 Main-container 外，同時亦允許多個 container 在同一個代理人系統中，同時每一個 container 容許許多個代理人同時執行。

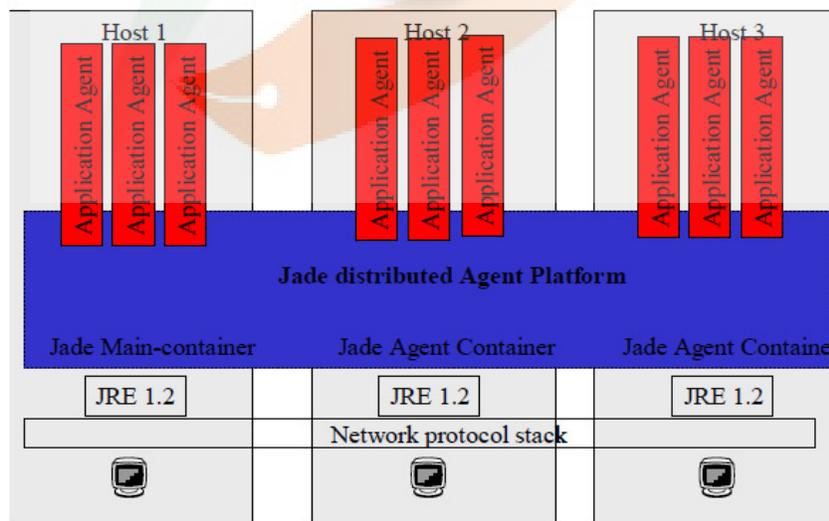


圖 2.5 JADE 平台執行環境與 container 關係圖

資料來源：「Fabio Bellifemine and Gioveanni Caire and Dominic P. A. Greenwood, 1996」

由此可知，JADE 代理人系統可以很簡單的建構出分散式環境，雖然外表上看，代理人是在不同的主機上執行，但就邏輯層面而論，代理人是在同一系統中的不同容器裡執行。圖 2.6 為 JADE 多代理人平台各個功能模組及代理人管理描述圖，簡介如下：

1. AMS(Agent Management System):

主要掌管代理人產生與消滅，同時給予每一個代理人一個唯一的識別名稱。

2. DF(Directory Facilitator):

簡稱目錄導引器，用於登錄、註銷或搜尋每個代理人可以提供的服務。

3. ACC(Agent Communication Channel):

代理人訊息傳送系統，負責分配代理人平台內的所有訊息傳送，也包含控制與遠端平台之間的訊息傳送。

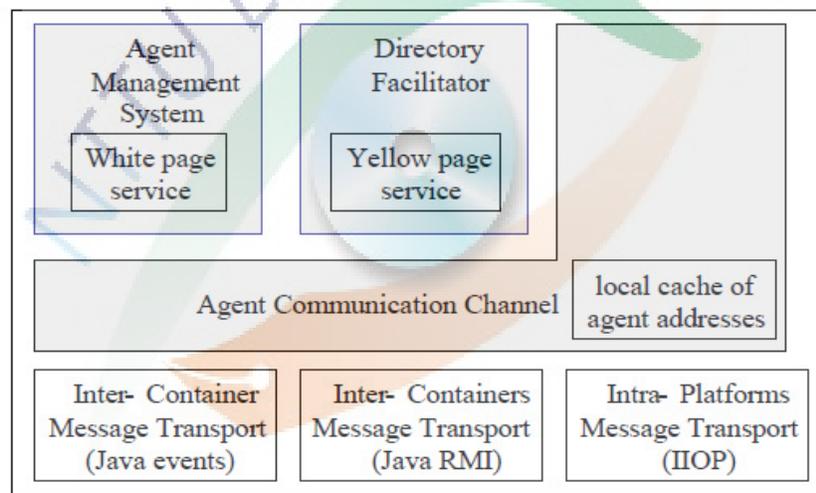


圖 2.6 JADE 平台系統架構

資料來源：「Fabio Bellifemine and Gioveanni Caire and Dominic P. A. Greenwood, 1996」

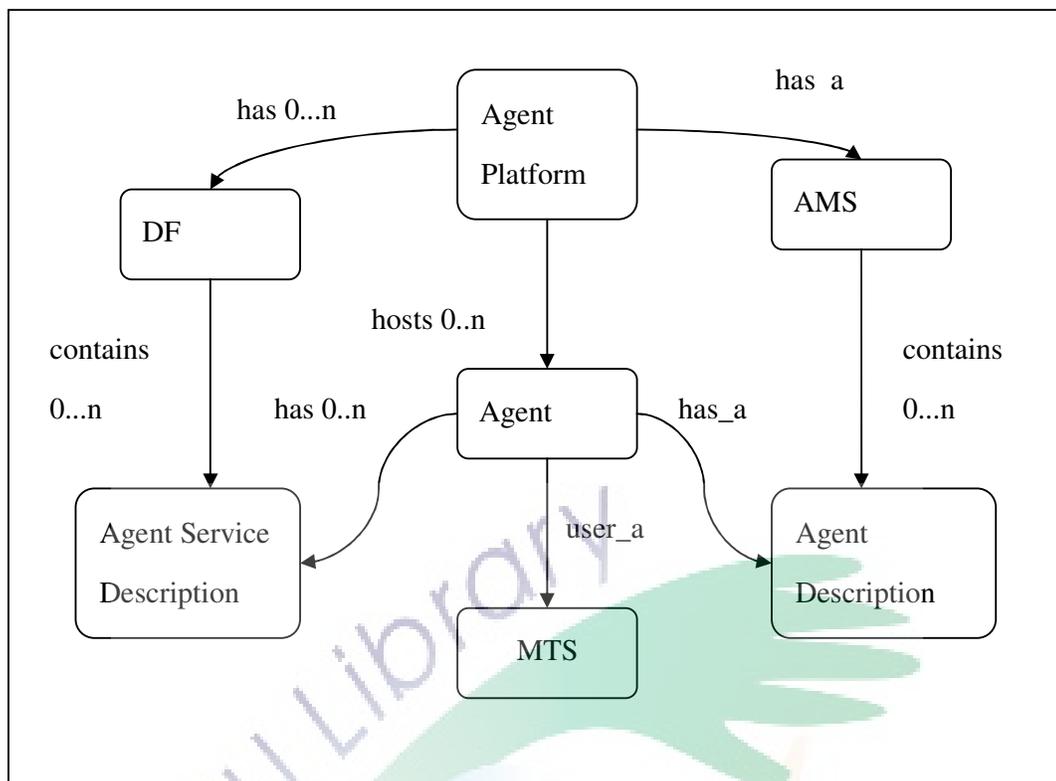


圖 2.7 Depiction of the agent management ontology

資料來源：「Fabio Bellifemine and Gioveanni Caire and Dominic P. A. Greenwood, 1996」

2.6.3 AMS(Agent Management System)探討

JADE 之 AMS(Agent Management System)代理人管理系統，他主要的功能是行使對代理人平台存取及使用的監控制工作，代理人產生時必須向 AMS 取得一個不與其他代理人相同的名稱作為系統識別。因此 AMS 維護及管理在此代理人平台下每一個代理人的識別 AID(Agent Identifier)，透過 AID 目錄管理，代理人可向代理人平台註冊傳送位址，因此，每一個代理人必須跟 AMS 註冊，以便獲得有效的 AID。

至於代理人如何得知他自己的代理人名稱?我們可以透過圖 2.8 所示的程式碼片段範例得知：

```
import jade.core.Agent;
public class BookBuyerAgent extends Agent {
    protected void setup() {
        // Printout a welcome message
        System.out.println("Hallo! Agent "+getName()+" is Ready.");
    }
}
```

圖 2.8 取得代理人名稱

資料來源：「Fabio Bellifemine and Gieveanni Caire and Dominic P. A. Greenwood, 1996」

由程式碼可得知只要呼叫 Agent 類別之 GetName()方法，就可以得知代理人的名稱。但是在多代理人系統中，不同平台間的代理人會有相同的名稱，所以才需要 AMS 系統去產生唯一的識別 AID，而代理人如何去得知自己的 AID 呢？只要透過圖 2.9 的程式碼，就可以得知。

```
String agentname = "myname";
AID id = new AID(agentname, AID.ISLOCALNAME);
```

圖 2.9 取得代理人 AID 程式片段

資料來源：「Fabio Bellifemine and Gieveanni Caire and Dominic P. A. Greenwood, 1996」

JADE 的 AID 組成如圖 2.10 所示，類似 e-mail 信箱地址的組成方式，以@為分隔，左邊為代理人名稱；右邊為代理人所處平台的平台名稱。

```
<agentname>@<platform-name>
```

圖 2.10 AID 組成結構

資料來源：「Fabio Bellifemine and Gieveanni Caire and Dominic P. A. Greenwood, 1996」

2.6.4 JADE 的行爲表現

代理人具有的行爲和執行的動作都定義在 `Behaviour` 類別中，我們可以將任務或者動作代碼放在 `Behaviour` 的 `action` 方法中，`action` 方法是必須要有的。`Behaviour` 有很多子類，分別對應着不同類型的 `Behaviour`，包括 `SimpleBehaviour`, `SequenceBehaviors`, `ParallelBehaviors`, `CyclicBehaviors` 等。一個 `agent` 的行爲表示它可以執行的任務，以繼承 `jade.core.behaviours.Behaviour` 實現。然後在 `agent` 類中通過 `addBehaviour()` 方法將行爲加入。當一個 `agent` 啓動時（通過 `setup()` 方法）後，行爲可以在任何時間加入進來。

要定義 `Behaviour` 必須實現其 `action()` 方法，它定義了 `agent` 的執行時的實際動作，而 `done()` 方法指明了一個行爲是否已執行完畢，是否要從行爲中刪除。一個 `agent` 可以同時執行多個 `behaviour`。透過 2.11 及 2.12 的圖可以得知，每個 `agent` 執行過程如下：

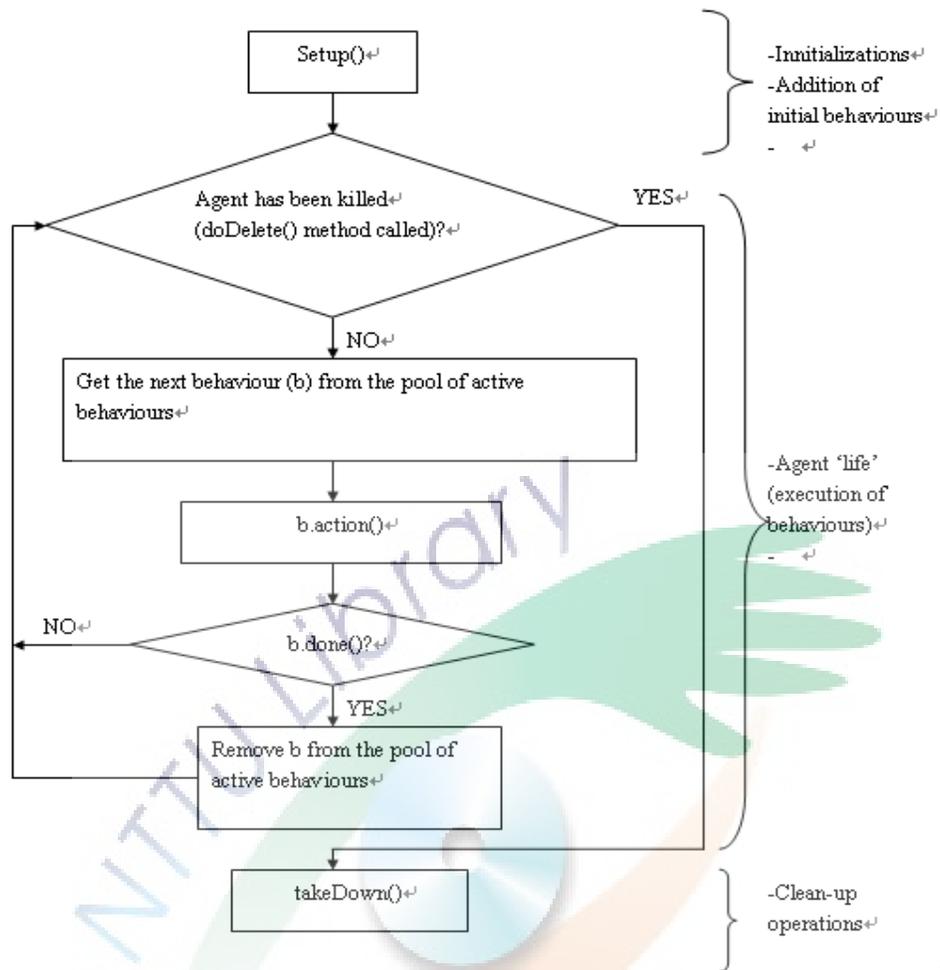


圖 2.11 Agent thread path of execution

資料來源：「Fabio Bellifemine and Gioveanni Caire and Dominic P. A. Greenwood, 1996」

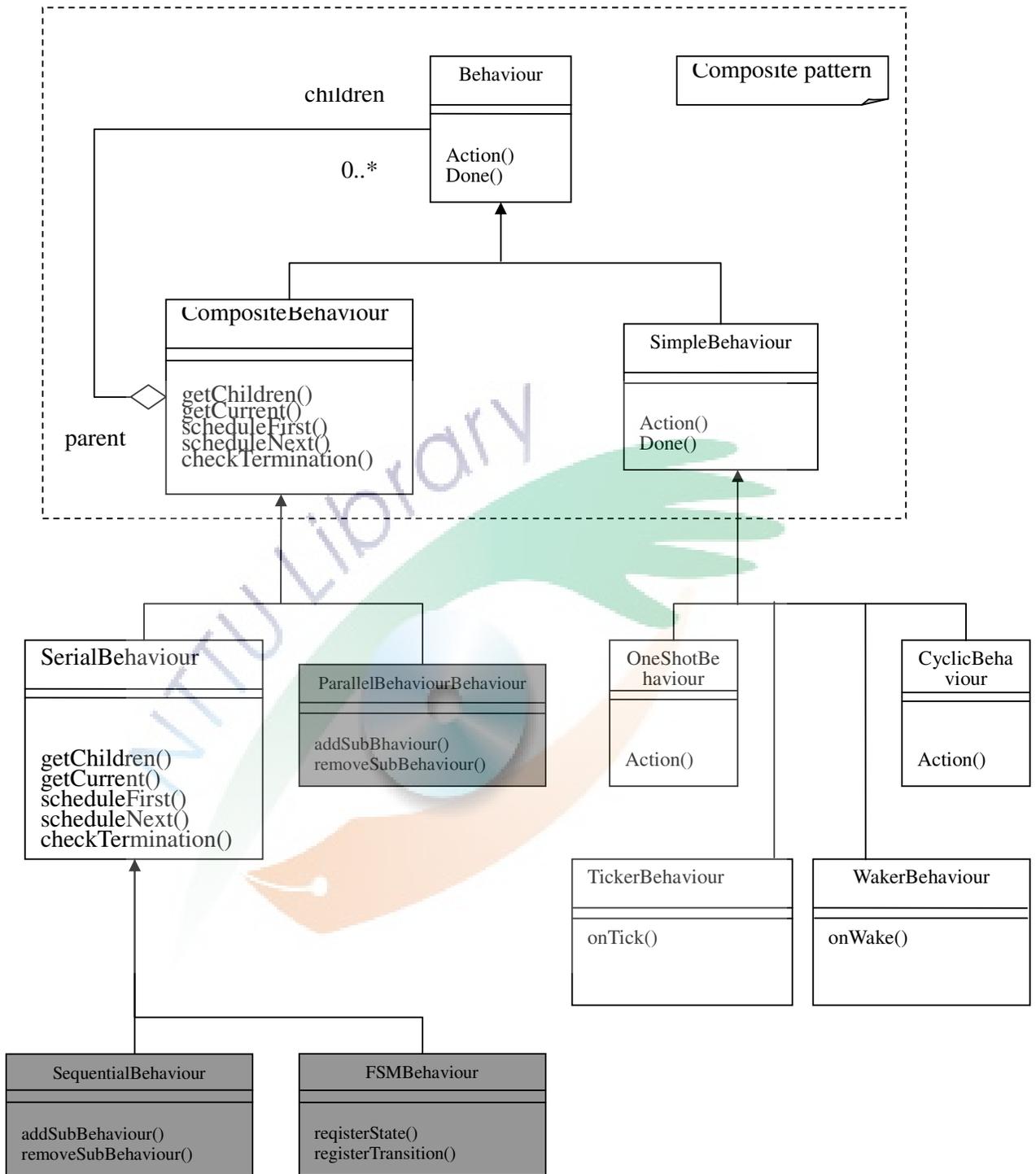


圖 2.12 The JADE behaviours hierarchy

資料來源：「Fabio Bellifemine and Gioveanni Caire and Dominic P. A. Greenwood, 1996」

2.7 基因演算法基本概念

John H. Holland於1975年大作 *Adaptation in Natural and Artificial Systems* (Holland, 1975) 提出基因演算法的概念，其精神是以達爾文的生物進化論「物競天擇、適者生存」而來。Holland是由自然界生物不斷競爭與繁殖所產生的基因原理中得到靈感，提出了基因演算的方法，用以模擬自然環境與生物的演化。Holland認為，人工或自然環境，都可將其值依屬性進行如基因編碼 (coding and representation)，並藉由編碼的交配及突變繁衍出新一代。透過函數設計可以遴選適合環境的下一代繼續參與繁衍，以此獲得較適合環境的物種。Holland 在該書中除了說明基因演算法的基本概念之外，亦奠定了基因演算法的方法原型。Holland指出基因演算法的三項基本操作 (Genetic operators)，即「交配」(Cross-over)、「反轉」(inversion) 與「突變」(Mutation)。所以至今，GA 已經被成功的應用於各種領域，並成功地可被用來解決許多問題。以下為基因演算的原理，詳述如下：

(1) 設定初始族群：

選擇研究範圍內的物種。

(2) 編碼(Encoding)

編碼的用意在於將問題的可能解以染色體的型式表達在基因演算法中，因為我們將問題的解比喻在自然界的生物，而生物的組成架構即為染色體，因此編碼的設計就如同決定生物的染色體一樣重要。在進行編碼前必須先確定問題的搜尋空間大小，最常用的編碼方式為二進位編碼(Binary coding)，即以0 和1 將問題的解編成n 個位元的字串，一個字串即代表一個染色體，每一個位元即代表一個基因。除了二進位編碼外，基因演算法還可以接受以整數(Integer)、實數(Real) 及文字(Alphabet)的編碼方式。

(3) 適應函數(Fitness Functions)

每一個物種在自然界中都有不同的適合性。適應函數，可以被認為是一個大自然的環境，用來決定每一個染色體的適合度。當演算進行時，整個編碼的方向會依著最佳的適合度做選擇。而適合度視問題特性而定，可能求最大值抑或是最小值。

(4) 選擇(Selection)

選擇(或複製)的目的，係模仿自然界中生物同性生殖的特性，將父代優良的染色體完全保留至子代，此程序可以保留上一代優良的染色體，淘汰適應值低的染色體，並產生新的族群(Population)，再繼續演化。選擇的機制有很多種，最常使用的方法為輪盤法(Roulette Wheel)，即將所有族群中的染色體依適應值的高低排序，將適應值視為一虛擬的輪盤，適應值高的染色體佔輪盤的面積較大，選取的方式如同旋轉輪盤一般，佔輪盤面積越大的染色體被選取到參與交配(Crossover)機率也越高。另一種常見的方式為競賽式選擇法(Competition selection)，通常依據競賽的觀念，會以選擇物種較為優秀的來進行，此即為適者生存，不適者淘汰的精神所在。

(5) 交配(Crossover)

對染色體進行交配的目的在於產生與父代不同基因的新子代，經由選擇(Selection)後保留下來的父代，經過兩兩配對，重新組合基因後，會產生新的子代(Offspring)，交配的方法為從父代的基因中隨機挑選數個交配點，將位於交配點的基因互換後，即產生兩個新的染色體，稱為子代。不同的編碼會有不同的交配方法，在此只列舉常見的二進位編碼的交配法：

- a. 單點交配(One-point crossover)：隨機挑選一個交配點，並以交配點為準，將交配點之後的基因互換，產生子代。
- b. 雙點交配(Two-point crossover)：隨機挑選二個交配點，並將兩個交配點之間的基因互換，產生子代。
- c. 多點交配(N-point crossover)：隨機挑選N 個交配點，並比較各交配點間的長度，將相同長度區段的基因互換，產生子代。

- d. 均勻交配(Uniform crossover)：染色體上的每個基因都視為交配點，並隨機挑選需互換的基因，產生新子代。
- e. 算數交配(Arithmetic crossover)：藉由某些數學運算來產生新子代。

(6) 基因的突變(Mutation)

在自然環境中，基因的突變是不可預期的，但也就因為這樣可能造成物種更強壯或更弱，目前常見的突變方式有：

- a. 選取一個位置做突變點。
- b. 增加或減少一個或多個字串值。
- c. 反轉式的突變。

(7) 停止條件(Termination Condition)

基因演算法的執行過程是迴圈，因此必須設定一個停止條件，判斷演化過程將繼續或是停止，常用的停止條件包括設定固定代數(fixed generations)，設定適應值目標值(fitness target)，適應值收斂程度評估(fitness convergence)，多樣化收斂評估(diversity convergence)。

- a. 設定固定代數(fixed generations)：基因演算法會演化至事先設定的代數即停止。
- b. 設定適應目標值(fitness target)：基因演算法會演化至適應值優於某一給定的適應值為止。
- c. 適應值收斂程度評估(fitness convergence)：當演化至族群裡染色體的適應值標準差均低於某一給定的適應值收斂程度時，表示族群中的染色體均已收斂至某一程度，此時即停止演化。
- d. 多樣化收斂評估(diversity convergence)：類似於適應值收斂程度評估，當族群中染色體的多樣性低於某一給定的多樣性收斂程度時，即停止演化。

(8) 參數設定

爲了最佳化基因演算法的搜尋效能，適當的參數設定是必要的，常設定的控制參數有族群大小(Population size)，交配率(Crossover rate)，交配點數，及突變率(Mutation rate)等。

a. 族群大小(Population size)：

族群大小指的是同一代族群中染色體的個數，族群的大小會影響搜尋的效能與效果，如果族群設的較小，每一代訓練的時間也較少，但容易產生區域最佳解的情況，使搜尋成效降低，若族群設的較大，計算成本也越高，雖然搜尋的時間會增加，但較不容易落入區域最佳解的情況，因此在基因演算法中，搜尋效能和效果間存在一取捨(Trade-off)的關係，而實際上族群大小的設定，則須依照實際問題以及經驗法則，選擇一個適當的族群大小，才能兼顧搜尋效能和效果。

b. 交配率(Crossover rate)：

交配率指的是每一代演化中，染色體發生交配的機率，交配率設的越高，交配會發生的越頻繁，搜尋的廣度也會增加；和族群大小類似，若交配率設的太低，則容易落入區域最佳解，但若交配率設的太高，也可能讓優良的基因消失，反而降低搜尋成效。舉例來說，若族群大小每一代中有500個染色體，而交配率設爲0.7，則表示每一代中有350個染色體需要發生交配。

c. 突變率(Mutation rate)：

突變率指的是每一代的染色體發生突變的機率，常用的突變方法是位元反相運算，即將基因從0變爲1，1變爲0，突變是一項非必要的運算，因爲突變很可能使好的基因遺失，因此一般而言，突變率發生的機率都遠低於交配率。

第三章 研究方法

本研究提出以代理人協商為基礎的智慧型 IT 專案資源配置系統，在進行資源配置時，使專案管理在進行遴選人才及效用分析時，參與者可以被假設為代理人 (agent)，依據各自的 BDI 模式，藉由自主、協同、溝通、生存、活動等特性，來配置專案資源。

為簡化表達在本研究協商模式運作中，不同代理人所扮演的角色，我們定義：進行專案管理時，參與者將對各項元素設定效用函數，當專案需求提出時，代理人即會開啓協商機制，稱為啓始代理人，本研究將針對所提出的協商機制與成本效益分析方法論進行整體需求設計，並探討代理人在此協商模式下，運用各種元素之效用函數進行多代理人之間的協商，而且判斷當協商不成立時，須要進行基因演算法演算，並提出以成本效益最大為基礎的專案計畫書。

3.1 代理人協商環境架構設計

本系統將利用 JADE 程式分析銀行 IT 如何達到資源配置之最佳化，指派代理人對使用者及 IT 人員尋求適當的解決方案，應用協商代理人的協調功能及工程師代理人的技術分析，讓雙方進行協商，期望可使 IT 的資源配置更有效率。

本研究依據表 3.1 代理人角色配置、圖 3.1 代理人之關係圖以及圖 3.2 協商代理人架構圖分析的結果，整理出代理人需求。此處在於分析代理人的主要工作、目標、行為。利用這些特性將代理人分為：IT 專案管理代理人、工程師代理人、技術人員代理人、顧問代理人、廠商代理人、使用者代理人、協商代理人等。各個代理人皆會賦與行為能力及協商執行能力，針對這些代理人，以下將會個別分析其特質：

1. IT 專案管理代理人：

目標：接收使用者代理人投入的需求與之協商及委派其它代理人任務。

行爲：

- (1) TickerBehaviour 爲每隔固定時間就會執行行爲定義。
- (2) Negotiation with UserAgent behaviours 爲接收使用者的需求時，所啓動的的機制。

工作：

- (1) 接收使用者代理人投入的需求。
- (2) 做資料分析，依據結果與使用者代理人進行協商。
- (3) 如有需要則會委派其它代理人共同進行協商。
- (4) 回覆協商結果。

2. 工程師代理人、技術人員代理人、廠商代理人：

目標：接收各個代理人詢問並提供本身資源。

行爲：

- (1) TickerBehaviour 爲每隔固定時間就會執行行爲定義。
- (2) Provide Resource Message With Other Agent。

工作：提供本身資源狀況。

3. 顧問代理人：提供 IT 專案管理人專業建議。

4. 使用者代理人：

目標：與 IT 專案管理人進行協商，並視其需要與其它代理人進行共同協商。

行爲：

- (1) TickerBehaviour 爲每隔固定時間就會執行行爲定義。
- (2) Negotiation with ITAgent behaviours 爲接收其它代理人的訊息時，所啓動的的機制。

工作：

- (1) 接收 IT 專案管理代理人回覆的訊息進行協商。
- (2) 做資料分析，依據結果再次提供提案進行協商。
- (3) 如有需要則會加入其它代理人共同進行協商。

4. 協商代理人：接收所有代理人訊息，進行共同協商，並在協商不成功後進行

基因演算法，可代替當專案管理之協商不成功時，藉助基因演算法找出替代方案。

表 3.1 代理人角色配置

角色配置	Description
(1)IT 專案管理代理人	此角色的代理人將必須能有管理專案的能力，以及對任務能充分瞭解與掌握。
(2)工程師代理人	此角色的代理人擁有特定任務的專長與知識。
(3)技術人員代理人	此角色的代理人對於某一方面的任務有其專業知識。
(4)顧問代理人	提供專案管理人建議(專案代理人專屬顧問)
(5)廠商代理人	此角色的代理人對於提供銀行 IT 專案主持人相關的資訊有其重要的可用性。
(6)使用者代理人	此角色的代理人為擁有專案相關內容探討及建議的能力。
(7)協商代理人	探討及建議的能力。

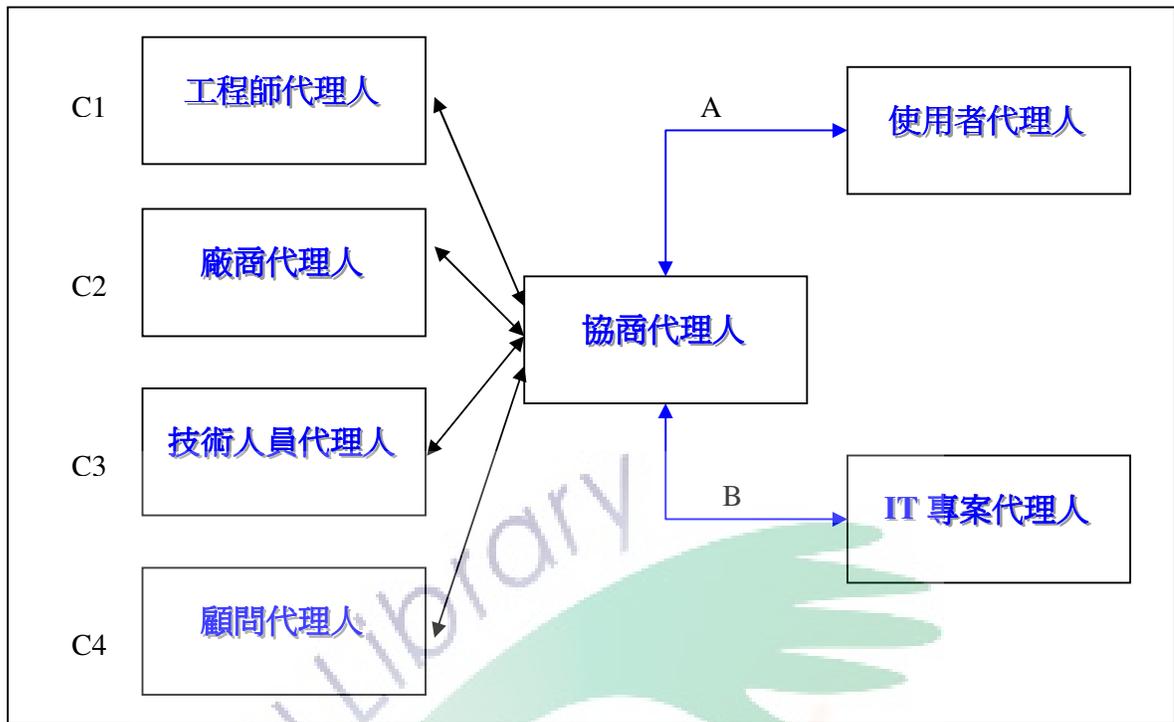


圖 3.1 代理人之關係圖

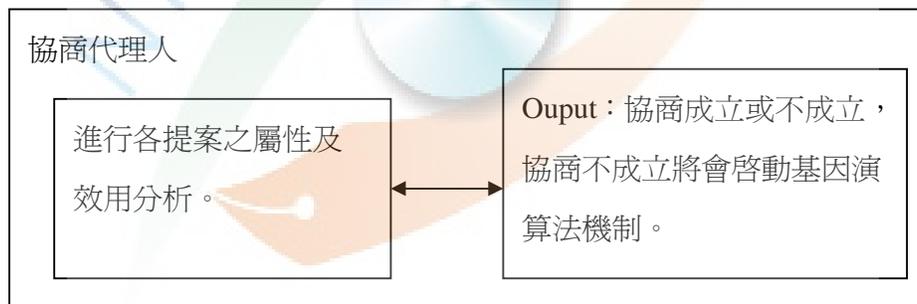


圖 3.2 協商代理人架構圖

3.2 代理人協商系統模擬

本研究之協商機制會先設定各個提案的屬性值，並設定專案元素的效用值，當協商成功時，即建立專案，當協商不成功時，會導入效益成本分析法配合遺傳演算法，運用遺傳演算法之物競天擇、適者生存理論，進行演化，取得效益越高、

成本越小的個體進行採用。本研究將會以楊子青與林啓明(2003)所提出之三個協商階段為基礎，在第二章文獻有提到問題定義階段、屬性處理階段以及協商評估階段並運用其公式並行協商。其中三階段的實際運作方式將會以本研究為基礎進行修改。大致方向改為(1)提案需求定義；(2)屬性及效用值建立；(3)進行協商；(4)協商成立與否判斷，成立代表參與者已接受該提案為專案；(5)進行基因演算法。可參考圖 3.3 為本系統之協商流程：

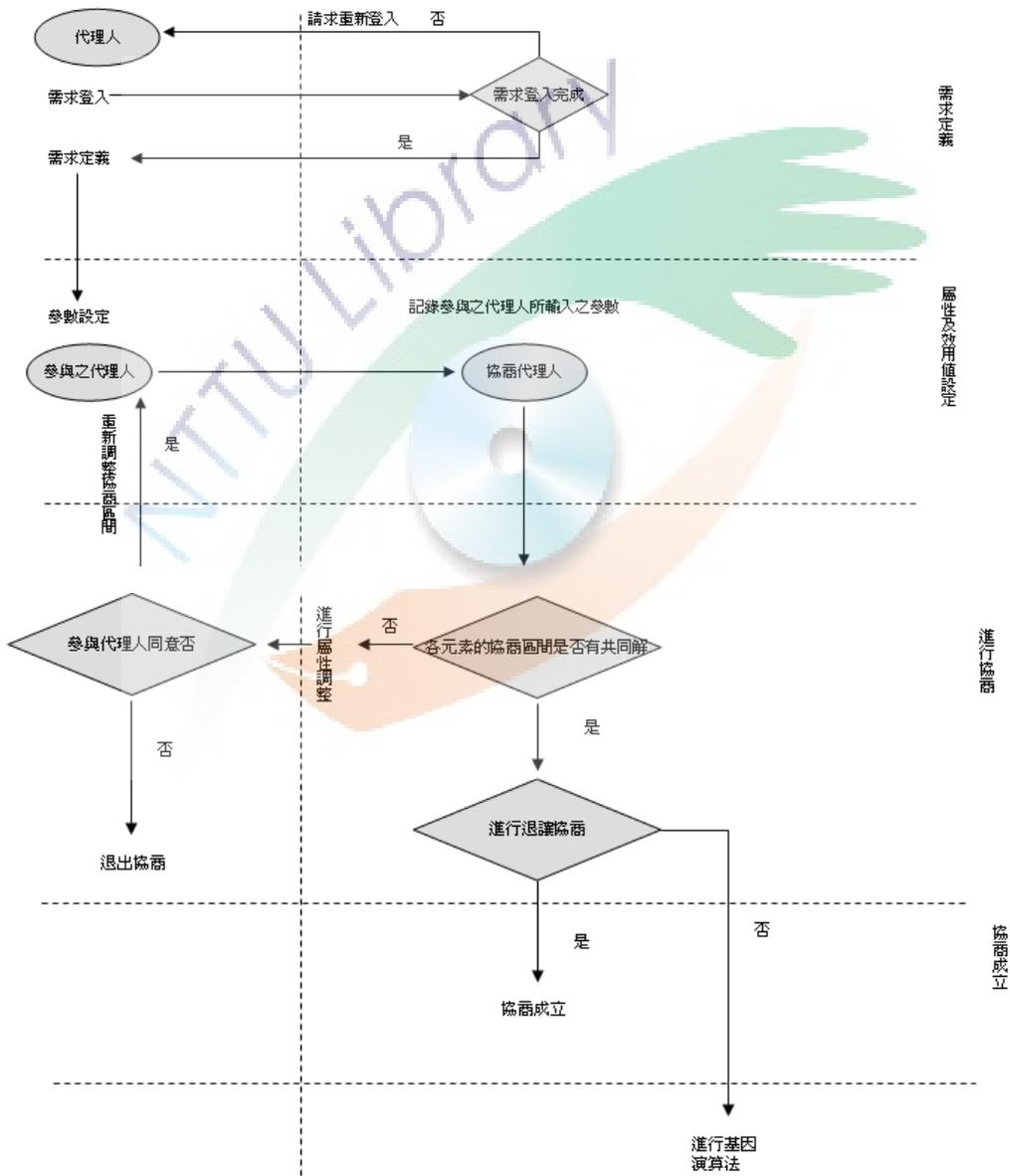


圖 3.3 代理人協商流程設計

第一階段：需求定義

在需求定義階段，將會由一名代理人啟動協商機制，當代理人登入協商平台並登入需求時，協商系統將會發出訊息給各個代理人，如使用者代理人、IT 專案管理代理人、廠商代理人、技術支援代理人、工程師代理人、顧問代理人，通知他們有一需求要處理，於是相關代理人將會進入協商平台，分別把自己的相關的資訊提供給協商系統。

第二階段：屬性以及效用值處理

本研究以楊子青與林啓明(2003)文獻中所提到的屬性最大交集蒐尋、屬性彈性調整為基礎，進行公式運算。

(1)屬性最大交集蒐尋：針對每一協商者的協商區間，進行搜尋，假設區間下限是(Lx)或區間上限(Ux)，則搜尋機制為以自己為準，如果其他協商者的上下限是否小於此值，如果都是則將之淘汰，如果不是，則記錄下來，反覆此動作，直到找出交集的最大集合。

(2)屬性彈性調整：當元素最大交集並未包含所有協商者時，系統會使用協商者設定之「彈性屬性」(Rx)來調整協商區間。首先會先計算群體「期望目標值」平均數(M)和需調整的「期望目標值」(Mx)的差值，再乘上彈性屬性來彈整。公式為：

$$\text{某調整者其彈性調整數值}(S_x) = |M_x - M| * R \quad \text{公式(3.1)}$$

$$\text{新協商空間值} = (L_x - S_x, U_x + S_x) \quad \text{公式(3.2)}$$

第三階段：進行協商

根據第二章文獻探討，本研究將參考讓步策略的五大類型做為協商策略，將考量相關環境因素與平均型退讓函數配合進行退讓協商，因此讓步策略是採取讓步皆相等的方式進行。公式如下：

屬性為正相關之效用值 =

$$\text{最大值} - (\text{最大值} - \text{最小值}) * \text{第幾次協商} / \text{總協商次數} \quad \text{公式 (3-3)}$$

屬性為負相關之效用值＝

$$\text{最小值} + (\text{最大值} - \text{最小值}) * \text{第幾次協商} / \text{總協商次數} \quad \text{公式 (3-4)}$$

屬性為無相關之提案值計算如下：

$$\left\{ \begin{array}{l} \text{當協商次數為奇數，提案值} = \text{期望值} - (\text{值望值} - \text{最小值}) * \frac{\text{協商次數} + 1}{\text{協商總數}} \quad \text{公式(3-5)} \\ \text{當協商次數為偶數，提案值} = \text{期望值} - (\text{最大值} - \text{值望值}) * \frac{\text{協商次數}}{\text{協商總數}} \quad \text{公式 (3-6)} \end{array} \right.$$

第四階段：協商成立與否

成立：即產生專案計畫書

不成立：進行基因演算機制

(1) 提出專案之編碼

在本研究中決定採用二進位編碼將所有輸入值由產生後，然後以一個染色體的形式來做計算。利用二進位編碼將基因串起來，建構一個染色體。染色體組合如下：

例長度為 7 的染色體：1001001

(2) 引用郭昱瑩著(2007)提出之成本效益理論，利用其成本效益分析選出專案的適應函數。運用公式計算出此專案之效益及設定適應函數找出協商中達成效益目標的專案。使用計算公式為：淨現值法、外部報酬率法兩種計算方式

淨現值法：

$$\text{公式: NPV} = \sum_{t=0}^n \frac{Bt - Ct}{(1+i)^t} \quad (3-7)$$

NPV: 計畫方案的整體淨效益

B_t:第 t 年之總效益

C_t:第 t 年之總成本

n : 計畫預期使用年限

i : 折現率(discounting rate)

外部報酬率法：

$$\text{公式:FWOC}=\sum_{t=0}^n B_j \times (1 + MARR)^{n-1} \quad (3-8)$$

R_j:各期收益

MARR:最低可接受的報酬率

n : 計畫預期可使用年限

$$\text{公式:FWOC}=\sum_{t=0}^n C_j \times (1 + ERR)^{n-1} \quad (3-9)$$

C_j : 方案在第 j 期的現金流出 (或投資)

ERR:外部報酬率

n : 計畫預期使用年限

合併二式即可獲得外部報酬率：

(3-3)=(3-4) 原始投資的終值

$$\sum_{t=0}^n B_j \times (1 + MARR)^{n-1} = \sum_{t=0}^n C_j \times (1 + ERR)^{n-1}$$

(3) 成本效益分析及遺傳演算法為基礎之專案協商成立流程圖

本研究在專案管理協商不成立時，運用成本效益分析及適應函數區塊中利用協商代理人的協商機商分別向各個代理人要求所以參數及資料，分析成本與效益，然後再把分析結果釋放出來，在這個架構當中協商代理人扮演極其重要的角

色，各個代理人皆需向其提供專業資料，由協商代理人統一評估專案可行性。此次研究會保留不符合之專案規劃，提供分析資料給使用者，讓使用者可以清楚的了解專案缺失及需改善部位。

系統實作部份會依此流程圖如下：



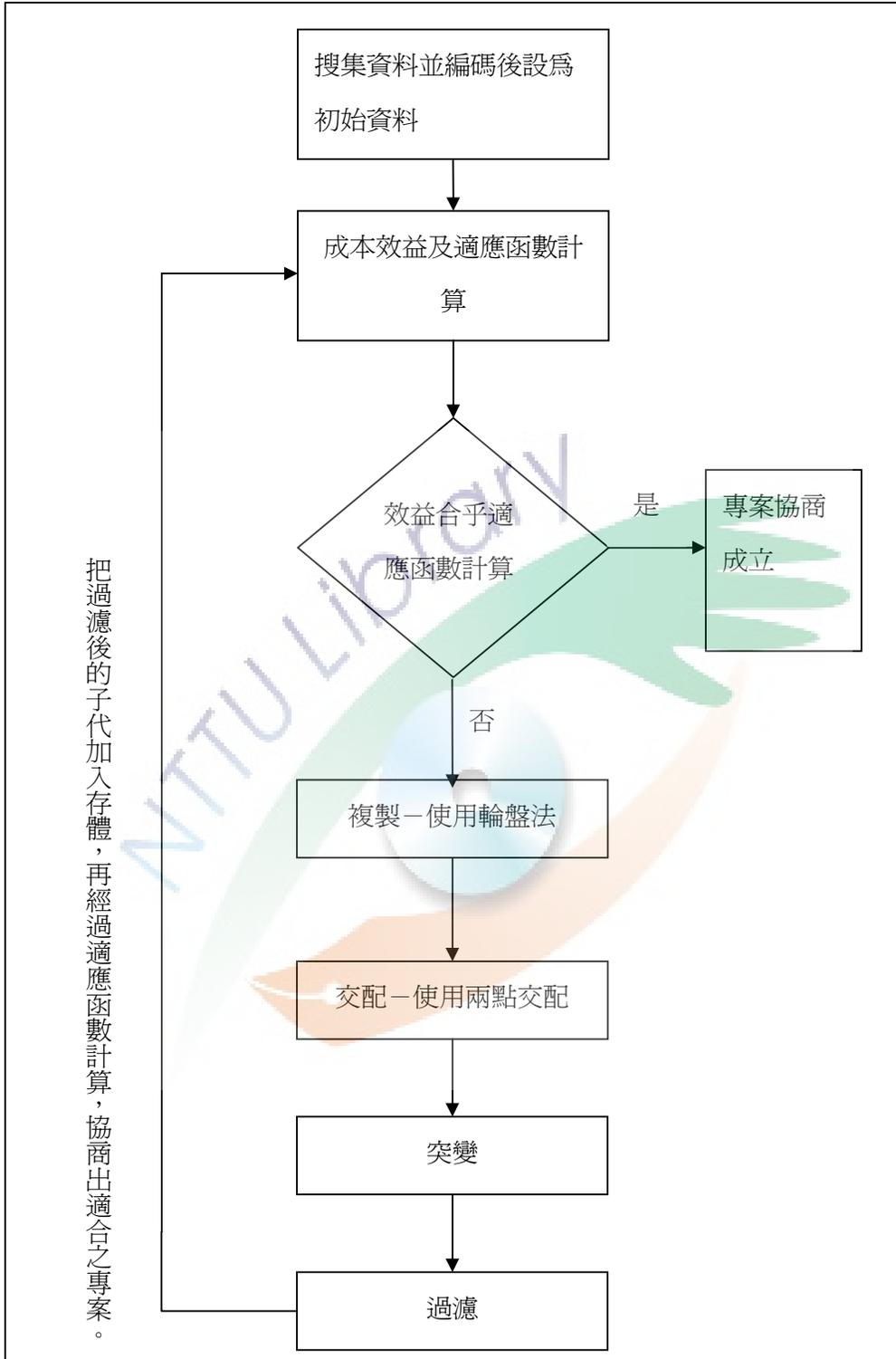


圖 3.4 運用成本效益分析及遺傳演算法為基礎之專案協商成立流程圖

3.3 代理人協商系統之基因演算法模式建立

1. 參數初始化：在演化之前，必須先設定好控制參數。
 - a. 最大演化世代，演化世代分為兩種：
 - a1.符合適存值即停止。
 - a2.可能系統設定演化世代。
 - a3.本範例採取符合適存值即停止。
 - b. 編碼字串長度
 - c. 變數個數
 - d. 交配率
 - e. 突變率
 - f. 族群大小

2. 設定初始族群

設定好參數後，可利用電腦亂數挑選出與正在協商之專案同一類型的專案，直到滿足族群大小為止。本研究將會設定族群固定為 10 組，包括原本進行協商的專案組合。表 3.2 是初始族群的設定，代表基因演算法所必須產生的初始方案，並且設定第 3 組為本次協商專案。以下為 Cost、Time、Scope、Quality、Human Resource 元素之間的關係及變化說明：

- a. **Scope**:專案的範疇，本研究將會定義範疇為大、中、小，在族群內也會挑選同樣範疇的專案進行演化。
- b. **Human Resource**:人力資源的變化將會直接影響專案的成本及時間運作。將人力資源分為強、中、弱三種狀態，以利分析。
- c. **Time**:同樣範疇內的專案會受預定時間的長短造成成本及品質的變化。將時間分為長、中、短三種狀態，以利分析
- d. **Quality**:品質會受到成本及時間所影響。將品質由小到大分為 1~5，共五種等級，以利分析

- e. **Cost:**當同樣範疇內的專案時間非常短暫時，將會造成成本增加，或預算不夠時，也會造成專案時間拉長或品質降低。
- f. 針對表 3.2，提供初始族群的十組設定之案例說明：
- 第一組：專案範疇很大，但屬於時程不趕的專案類型，人力資源也很強大，成本可以控制在預算內。
- 第二組：專案範疇很大，但屬於時程不趕的專案類型，唯人力資源不比第一組強，成本雖可控制在預算內，但品質可能受影響，但專案進行中也發生突發狀況，造成本增加。
- 第三組：專案範疇很大，但屬於時程不趕的專案類型，唯人力資源很弱，屬於超過預算的專案組合。
- 第四組：專案範疇很大，但屬於時程不趕的專案類型，唯人力資源不比第一組強，成本雖可控制在預算內，但品質可能受影響，爲了不使品質受影響，新增不在專案內的人力支援。
- 第五組：專案範疇很大，屬於時程很趕的專案類型，但人力資源很強，成本也會相對增加。
- 第六組：專案範疇很大，屬於時程很趕的專案類型，但人力資源並不比強，雖然成本增加，但無法找到人力支援，所以品質也會受到影響。
- 第七組：專案範疇很大，屬於時程很趕的專案類型，但人力資源很弱，成本增加幅度稍大，專案品質才能達成正常水準。
- 第八組：專案範疇很大，屬於有足夠時程的專案類型，人力資源很強，成本可控制在預算內。
- 第九組：專案範疇很大，屬於有足夠時程的專案類型，人力資源足夠，成本可控制在預算內。
- 第十組：專案範疇很大，屬於有足夠時程的專案類型，人力資源不足，成本可控制在預算內，但還是有潛在的風險，須盡量早點安排額外人力支援。

表 3.2 設定初始族群

專案規劃 組合	Cost	Time	Scope	Quality	Human Resource
第一組	200 萬元	middle	大(200 萬以 上)	5	Strong
第二組	210 萬元	middle	大(200 萬以 上)	4	Balance
第三組	250 萬元	middle	大(200 萬以 上)	2	Weak
第四組	220 萬	middle	大(200 萬以 上)	5	Balance
第五組	260 萬	short	大(200 萬以 上)	4	Strong
第六組	270 萬	short	大(200 萬以 上)	3	Balance
第七組	300 萬	short	大(200 萬以 上)	4	Weak
第八組	200 萬	long	大(200 萬以 上)	5	Strong
第九組	200 萬	long	大(200 萬以 上)	4	Balance
第十組	200 萬	long	大(200 萬以 上)	3	Weak

3. 計算成本效益值：

由表 3.3 所示，由於建構系統所產生的利益較為無形，所以本研究假設如果未建構系統時，所須承受的人事成本費用應用在專案收益上，所以此專案預

估未來收益有 400 萬。在系統上可設定折現率，本專案將設折現率為 20%。

第一年之成本效益值公式如下，以萬元為單位，NPV 如有小數點，保留至第五位，並做四捨五入，B/C 如有小數點，保留至第 1 位，並做四捨五入：

$$\text{NPV: } \sum_{t=0}^n \frac{Bt - Ct}{(1+i)^t} \quad (3-7)$$

$$\text{公式：B/C Ratio: } \sum_{t=0}^n \frac{Bt}{(1+i)^t} \bigg/ \sum_{t=0}^n \frac{Ct}{(1+i)^t} \quad (3-10)$$

表 3.3 計算成本效益值

專案規劃組	第一組	第二組	第三組	第四組	第五組
合					
NPV(成本 效益值)	166.66667	158.33333	125	150	116.66667
B/C(經濟益 本比)	2	1.9	1.6	1.8	1.5
專案規劃組	第六組	第七組	第八組	第九組	第十組
合					
NPV(成本 效益值)	108.33333	83.33333	166.66667	166.66667	166.66667
B/C(經濟益 本比)	1.48	1.3	2	2	2

4. 對專案組合進行編碼：

如表 3.4 所示，為了分析各個元素的關係，在進行編碼時，分別針對各個元素定義編碼個數。

- a. Cost: 在本研究中限制專案成本最大到一千萬，所以 cost 的位元最大會到十位。

- b. Time:時間將會由短到長，分別設 1 Short、2 Middle、3 Long。
- c. Scope:範疇將會由小到大，分別設 1 Small、2 Middle、3 Big。
- d. Quality:品質將會由小到大，分別設 1、2、3、4、5。
- e. Human Resource:人力資源將會由弱到強，分別設 1 Weak、2 Middle、3 Strong。

表 3.4 對專案組合進行編碼

專案規 劃組合	專案組合 二進位編碼
第一組	0011001000 10 11 101 11
第二組	0011010010 10 11 100 10
第三組	0011111010 10 11 010 01
第四組	0011011100 10 11 101 10
第五組	0100000100 01 11 100 11
第六組	0100001110 01 11 011 10
第七組	0100101100 01 11 100 01
第八組	0011001000 11 11 101 11
第九組	0011001000 11 11 100 10
第十組	0011001000 11 11 011 01

5. 選擇機制：

選擇機制是模仿生物界「適者生存，不適者淘汰」的方法，它的主要功能是用來確定某個染色體被選擇的個數；與適應值互相配合，即可實現複製基因演算法的族群個體。

本研究應用輪盤法實現此一概念。當適應值越高則所佔的輪盤面積也就越大，如此被選擇的機率就越高，直到交配池的數量等於母體數的染色體，其中個體被選到的機率，公式為：

$$\text{選擇機率} = \text{個體適應值} / \text{所有適應值的總和} \quad (3-11)$$

假設表 3.5 的族群大小=10，經由計算得到十個個體的適應值，分別如下所示：

表 3.5 經選擇後的個數

專案規劃組合	第一組	第二組	第三組	第四組	第五組	第六組	第七組	第八組	第九組	第十組
計算適存值	2	1.9	1.6	1.8	1.5	1.48	1.3	2	2	2
存活數目	1	1	2	1	1	0	1	1	1	1
選取率	11.4%	10.8%	9.1%	10.2%	8.5%	8.4%	7.4%	11.4%	11.4%	11.4%

6. 經過選擇後的組成，如表 3.6 所示：

表 3.6 經過選擇後的組成

專案規劃組合	成本效益值	二進位編碼	適存值
第一組	166.66667 萬元	0011001000 10 11 101 11	2
第二組	158.33333 萬元	0011010010 10 11 100 10	1.9
第三組	125 萬元	0011111010 10 11 010 01	1.6
第四組	125 萬元	0011111010 10 11 010 01	1.6
第五組	150 萬元	0011011100 10 11 101 10	1.8
第六組	116.66667 萬元	0100000100 01	1.5

		11 100 11	
第七組	83.33333 萬元	0100101100 01 11 100 01	1.3
第八組	166.66667 萬元	0011001000 11 11 101 11	2
第九組	166.66667 萬元	0011001000 11 11 100 10	2
第十組	166.66667 萬元	0011001000 11 11 011 01	2

7. 交配機制：

經過選擇機製挑選染色體後，繼續要進行交配。如圖 3.5、圖 3.6、圖 3.7、圖 3.8、圖 3.9 所示，交配機制是基因演算法重要的方法之一；它將兩個母代的部份數值做交換，而產生兩個子代；交配就好像在做大幅度的搜尋。母代是否要進行交配取決交配率，本研究考量於主題較特殊且專案集合體的個數不多，故決定用百分之百的交配率，做為交配的運算機制且本研究設定之染色體為各屬性的集合，所以只能依各屬性做單點交配。以人類行為來看，在專案管理進行中，專案管理者會因大環境的影響而做出抉擇，而在代理人協商系統亦是，本研究以專案管理中的成本為主，故成本不會參與交配，但會因其它因素如時間、品質、人力資源的交換而有所改變。至於專案管理集合，也會在協商不成功或交配不理想，為了避免產生最佳區域解，而做單點突變。這將會在表 3.7 說明。為配合實驗，在本研究中，設定以原來成本為基準，當各個因素有增減時，會各自加減十萬，在交配後，成本會自動重算。

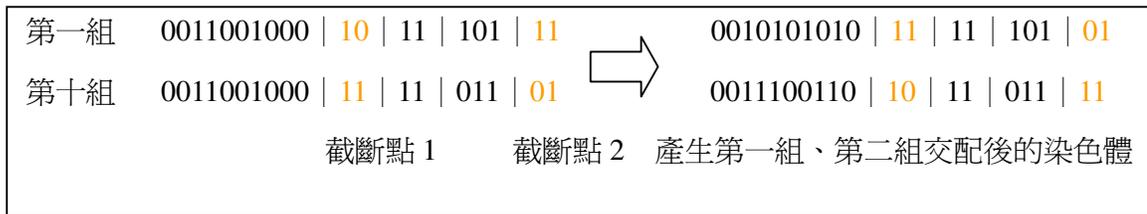


圖 3.5 Cross-over 圖解一

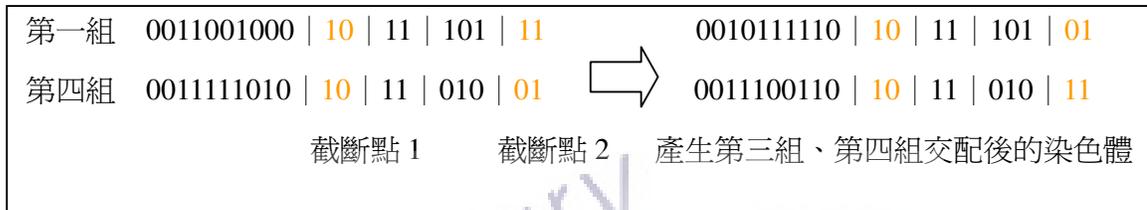


圖 3.6 Cross-over 圖解二

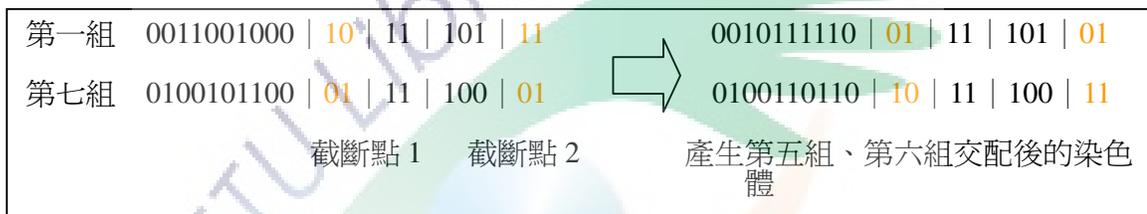


圖 3.7 Cross-over 圖解三

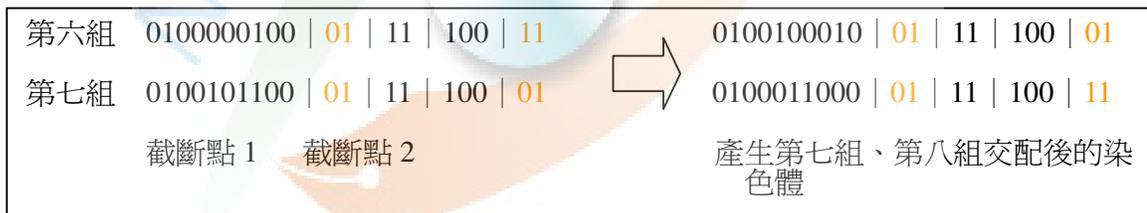


圖 3.8 Cross-over 圖解四

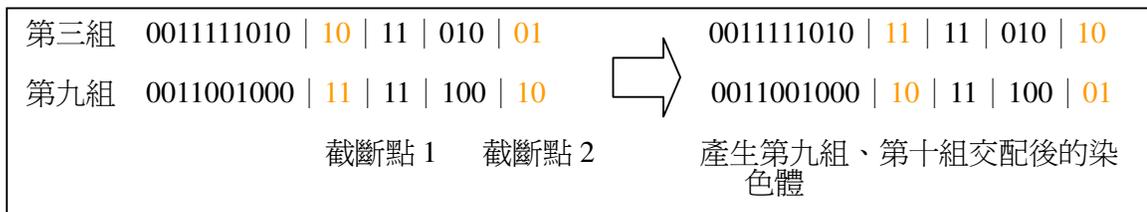


圖 3.9 Cross-over 圖解五

8. 經過交配後的組成，如表 3.7 所示：

本研究以分析成本為主，在經過交配後，符合適存值的組合有第一組、第三

組、第五組、第十組，以達到最佳適存值為基準，當演化到第二世代，收斂在最好的專案組合為第一組。

表 3.7 選擇及交配後的族群

專案規劃組合	成本	二進位編碼	適存值
第一組	170 萬元	0010101010 11 11 101 01	2.35
第二組	230 萬元	0011100110 10 11 011 11	1.74
第三組	190 萬元	0010111110 10 11 101 01	2.11
第四組	230 萬元	0011100110 10 11 010 11	1.74
第五組	190 萬元	0010111110 01 11 101 01	2.11
第六組	310 萬元	0100110110 10 11 100 11	1.29
第七組	290 萬元	0100100010 01 11 100 01	1.38
第八組	280 萬元	0100011000 01 11 100 11	1.43
第九組	250 萬元	0011111010 11 11 010 10	1.6
第十組	200 萬元	0011001000 10 11 100 01	2

9. 突變機制，如表 3.8 所示：

使用單點突變，應用在交配後的專案組合，即使成本符合需求，但品質不盡理想時，會啓動突變機制，並制定突變率為 0.05，表示每筆被選中的機率有 5%，突變後將會改變原先的染色體結構，其目的為避免區域最佳解及尋求最佳解。

表 3.8 突變後的族群

專案規劃組合	適存值	二進位編碼
第一組	2.35	0010101010 11 11 101 01
第二組	1.74	0011100110 10 11 011 11
第三組	2.11	0010111110 10 11 101 01

第四組	1.74	0011100110 10 11 010 11
第五組	2.11	0010111110 01 11 101 01
第六組	1.29	0100110110 10 11 100 11
第七組	1.38	0100100010 01 11 100 01
第八組	1.43	0100011000 01 11 100 11
第九組	1.6	0011111010 11 11 010 10
第十組	2	0011001000 10 11 100 01



第四章 建置代理人協商驗證系統與結果分析

為驗證本研究所提出模型的可行性和實用性，第四章針對第三章所提出之架構，進行系統實作。並進行測試，測試範例將會以銀行 IT 之專案進行。範例中包含了專案名稱、專案完成時間、成本、人力資源、專案範圍等資訊，文中將統計其各項因素所產生之成本作為評定指標。為了驗證研究方法之可行性，本章將先建立代理人協商系統，以其達到驗證效果。

4.1 代理人雛型系統之協商機制模擬

為了驗證專案管理代理人協商機制的運作是可行的，因此本節將引入一個案實驗，模仿人類實際進行協商時的表現。在協商過程中，本研究將採用 AUML 循序圖(Agent Unified Modeling Language Sequence Diagram)來模擬各代理人之間的溝通協定及行為表現，然後以 JADE 實際進行協商機制，之後進行的實驗都將會利用 AUML 及 JADE 這兩種工具互相配合完成協商。

需求說明：

1. 透過 Web 將 Local PC 的 Raw data 上傳 Incentive System database，上傳時須指定 Rev 年份／月份，Raw data 檔案明細。
2. 須提供重複上傳的機制，User 如重複上傳同一月份的 data，則系統須用新的 data 取代。
3. Database output format，所需欄位共 10 個。
4. SD、ETF、DCD revenue 如為外幣，換算為台幣之邏輯：按交易日(Transaction Date)之中價匯率換算為台幣。
5. Revenue Report。
6. 刪除” AO Performance Details Report” 報表中原 Rev 數字，另加(Total Rev)欄位。

7. 新增兩個上傳檔以計算 Revenue。

問題定義：

本研究採用整合型協商，進行問題解析及定義，在文獻中有提到，相對於分配型協商在對立地位上的競爭，整合型協商站在合作的角度上做協商考量，所以可以針對專案中的各個元素進行協商。接著，在需求說明所提到的各項需求，將會在此節定義：首先使用者進行登入需求的循序圖如圖 4.1 及圖 4.2：

第一階段：使用者代理人，簡稱：UA

UA 進行協商機制-1 循序圖說明：UA 以提出者進協商機制登入需求並且設定參數，UA 針對專案管理集合中的各項元素設定屬性。在此 UA 針對 Cost 設定 0，表示數值屬性表示成本越小越好；Time 設定 0，數值屬性表示時間越少越好；Scope 設定 0 表示範疇越接近需求越好；Quality 設定為 5，數值屬性表示品質越高越好；Resource 設為 3，數值屬性，表示人力資源越強越好。於是 UA 向 IA(IT 專案管理代理人)發出 request 訊息，內容包含使用者輸入的需求及參數。

第二階段：IT 專案管理代理人，簡稱：IA

IA 接收請求，並接著發出 request 給 VA(廠商代理人)，完成之後 VA 會傳送兩種訊息回 IA。IA 接收 VA 傳回訊息後，即起動協商代理人，將各個代理人的參數傳入協商系統。

第三階段：廠商代理人，簡稱：VA

VA 接收請求，並接著回覆資訊給 IA，讓 IA 啟動協商代理人。本研究設定 VA1、VA2、VA3 共有三組共同參與協商。

第四階段：協商代理人，簡稱：NA

NA 接到 IA 提出之 request，依照個別代理人之屬性值開始進行協商。表 4.1、4.2、4.3、4.4、4.5 為各提案之屬性值。

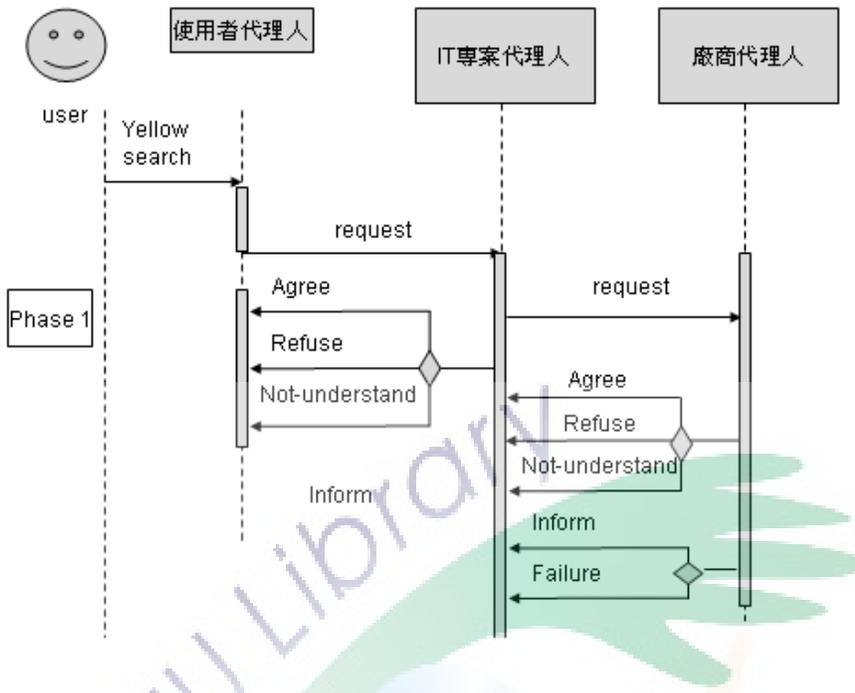


圖 4.1 使用者代理人進行協商機制-1 循序圖

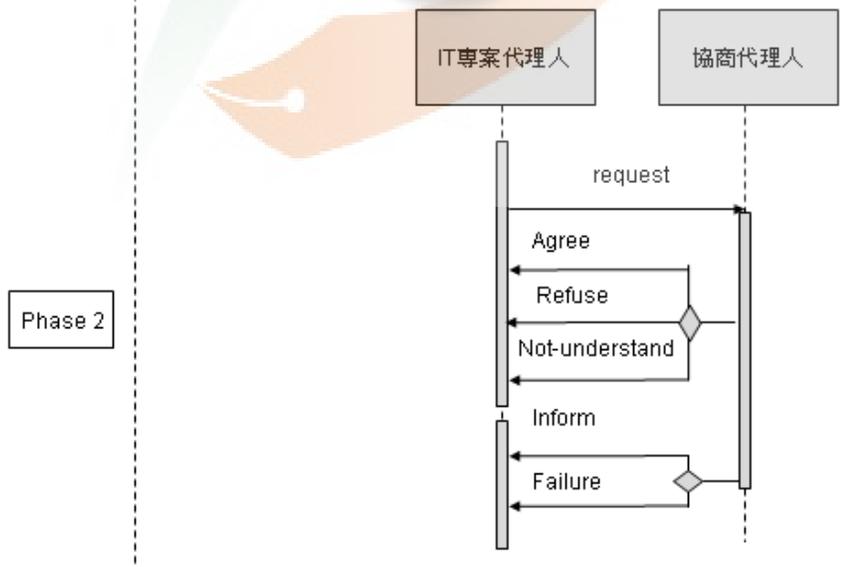


圖 4.2 使用者代理人進行協商機制-2 循序圖

表 4.1 UA 所設定之屬性數值

UA	屬性值設定			
討論項目	屬性種類	期望目標值	協商區間值	彈性指標
Cost	數值	20 萬元	[15,25]萬元	0.2
Time	數值	30 天	[24,35]	0.2
Scope	數值	1	[1,2]	0.1
Quality	數值	4	[4,5]	0.1
Human Resource	數值	3	[1,3]	0.3

表 4.2 IA 所設定之屬性數值

IA	屬性值設定			
討論項目	屬性種類	期望目標值	協商區間值	彈性指標
Cost	數值	18 萬元	[12,25]萬元	0.1
Time	數值	26 天	[20,28]	0.1
Scope	數值	1	[1,2]	0.1
Quality	數值	5	[4,5]	0.1
Human Resource	數值	3	[1,3]	0.2

表 4.3 VA1 所設定之屬性數值

VA1	屬性值設定			
討論項目	屬性種類	期望目標值	協商區間值	彈性指標
Cost	數值	25 萬元	[20,30]萬元	0.3
Time	數值	32 天	[30,40]	0.3
Scope	數值	1	[1,2]	0.1

Quality	數值	5	[4,5]	0.1
Human Resource	數值	2	[1,2]	0.2

表 4.4 VA2 所設定之屬性數值

VA2	屬性值設定			
討論項目	屬性種類	期望目標值	協商區間值	彈性指標
Cost	數值	35 萬元	[33,38]萬元	0.3
Time	數值	30 天	[30,40]	0.3
Scope	數值	1	[1,2]	0.1
Quality	數值	5	[3,5]	0.1
Human Resource	數值	2	[1,3]	0.2

表 4.5 VA3 所設定之屬性數值

VA3	屬性值設定			
討論項目	屬性種類	期望目標值	協商區間值	彈性指標
Cost	數值	28 萬元	[24,32]萬元	0.3
Time	數值	30 天	[30,40]	0.3
Scope	數值	1	[1,2]	0.1
Quality	數值	4	[4,5]	0.1
Human Resource	數值	2	[1,2]	0.2

第五階段：進行協商屬性調整

為保持 IA 的主導性，只有 IA 可以看到各個提案的所有屬性值，而會公開的資訊只有期望值，其它如彈性指數、協商區間都是屬於私密資料，故不公開。當協商

者輸入參數後，系統會利用第三章之研究方法找出提案各屬性交集最多的集合，依照表 4.1、4.2、4.3、4.4、4.5，五個提案集合的屬性分別為表 4.6 所示：

表 4.6 各提案之屬性值

屬性值	Cost	Time	Scope	Quality	Human Resource
UA	[15,25]萬元	[24,35]天數	[1,2]	[4,5]	[1,3]
IA	[12,25]萬元	[20,28]天數	[1,2]	[4,5]	[1,3]
VA1	[20,30]萬元	[30,40]天數	[1,2]	[4,5]	[1,2]
VA2	[33,38]萬元	[30,40]天數	[1,2]	[3,5]	[1,3]
VA3	[24,32]萬元	[30,40]天數	[1,2]	[4,5]	[1,2]

爲了讓後續的計算易於了解，分別將屬性值給與代號，條列式如下：

Cost :

元素屬性：負向屬性，數值越大，表示越不讓決策者接受。

UA(C1,C2)、IA(C3,C4)、VA1(C5,C6)、VA2(C7,C8)、VA3(C9,C10)由小排到大爲 C3<C1<C5<C9<C2=C4<C6<C10<C7<C8，計算 Cost 的最大集合，落在 UA、IA、VA1、VA3。

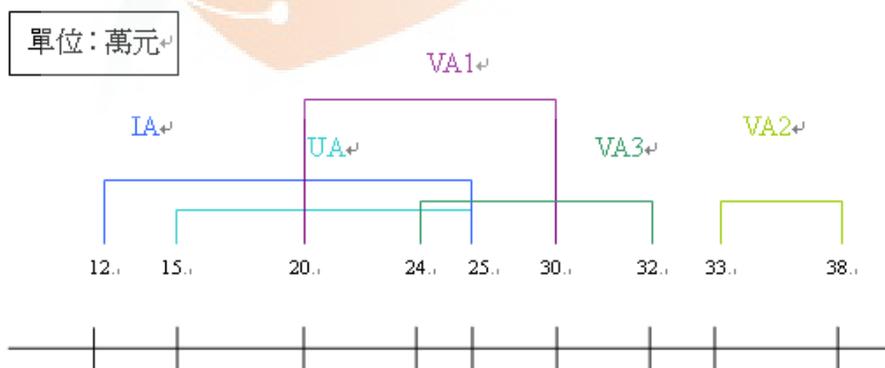


圖 4.3 Cost 之最大集合表示圖

Time:

如圖 4.4 所示，元素屬性：正向屬性，時間越短，會造成成本相對增加，決策者會依狀況考量。UA(T1,T2)、IA(T3,T4)、VA1(T5,T6)、VA2(T7,T8)、VA3(T9,T10) 計算 Time 的最大集合，落在 VA1、VA2、VA3、UA。

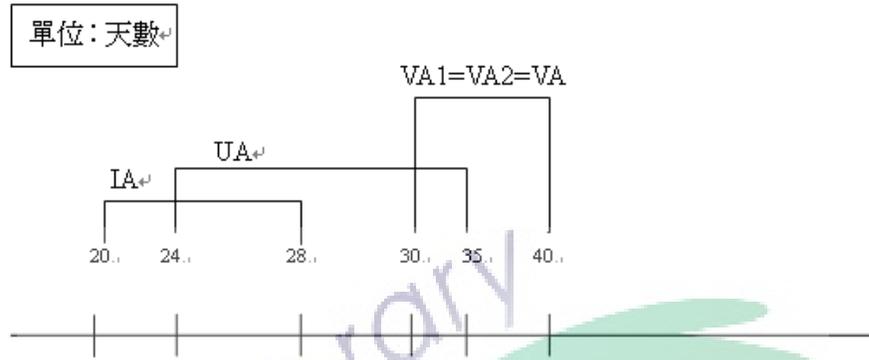


圖 4.4 Time 之最大集合表示圖

Scope:

元素屬性：無相關

UA(S1,S2)、IA(S3,S4)、VA1(S5,S6)、VA2(S7,S8)、VA3(S9,S10)，計算 Scope 的最大集合，落在 UA、IA、VA1、VA2、VA3。

Quality:

元素屬性：正向屬性，數值越大，表示越能被決策者接受。

UA(Q1,Q2)、IA(Q3,Q4)、VA1(Q5,Q6)、VA2(Q7,Q8)、VA3(Q9,Q10)，計算 Quality 的最大集合，落在 UA、IA、VA1、VA2、VA3。

Human Resource:

元素屬性：正向屬性，數值越大，表示越能被決策者接受。

UA(H1,H2)、IA(H3,H4)、VA1(H5,H6)、VA2(H7,H8)、VA3(H9,H10)

計算 Human Resource 的最大集合，落在 UA、IA、VA1、VA2、VA3。

根據實例，除了 Cost、Time 未能全部交集，其它元素皆有全部交集，根據第三章研究方法 Cost、Time 需做協商區間調整。

1. VA2-Cost 屬性調整：

期望目標值平均數： $M=(20+18+25+23+35)/5=24.2$

數值調整： $135-24.21*0.3=1.74$

新的協商區間： $[31.26,39.74]$

2. UA-Time 屬性調整：

期望目標值平均數： $M=(30+26+32+30+30)/5=29.6$

數值調整： $130-29.61*0.2=0.8$

新的協商區間： $[23.2,35.8]$

3. IA-Time 屬性調整：

期望目標值平均數： $M=(30+26+32+30+30)/5=29.6$

數值調整： $126-29.61*0.1=0.36$

新的協商區間： $[19.64,28.36]$

如表 4.7 所示，根據實例經過調整的協商調整過的區間值，Cost 的集合更新為 UA、IA、VA1、VA2、VA3，Time 的集合更新為 UA、VA1、VA3，但是由於 IA、VA2 雖然被通知需調整協商區間，但由於 IA 及 VA2 考量後，覺得不符合原先期待，所以不同意調協商整區間值，最終進入退讓協商的集合會為 UA、VA1、VA3。

表 4.7 屬性調整後之協商區間值

屬性值	Cost	Time	Scope	Quality	Human Resource
UA	[15,25]萬元	[23.2,35.8] 天數	[1,2]	[4,5]	[1,3]
IA	[12,25]萬元	[19.64,28.36]天 數	[1,2]	[4,5]	[1,3]
VA1	[20,30]萬元	[30,40]天數	[1,2]	[4,5]	[1,2]
VA2	[31.26,39.74]萬	[30,40]天數	[1,2]	[3,5]	[1,3]

	元				
VA3	[24,32]萬元	[30,40]天數	[1,2]	[4,5]	[1,2]

第六階段：進入退讓協商

計算各個元素的效用值，並且各個提案的元素效用值都要與其它提案的元素相符合，才會接受提案，根據方法論，本研究之退讓值採用平均型讓步策略。此退讓協商在第五次，協商第一次成功，之後進行到第十次，協商都未能成功，所以這次協商成功，會成為專案的是 VA1 所提出來的提案。

公式為：

屬性為正相關之效用值 = 最大值 - (最大值 - 最小值) * 第幾次協商 / 總協商次數

屬性為負相關之效用值 = 最小值 + (最大值 - 最小值) * 第幾次協商 / 總協商次數

屬性為無相關之提案值計算如下：

$$\left\{ \begin{array}{l} \text{當協商次數為奇數，提案值} = \text{期望值} - (\text{值望值} - \text{最小值}) * \frac{\text{協商次數} + 1}{\text{協商總數}} \\ \text{當協商次數為偶數，提案值} = \text{期望值} - (\text{最大值} - \text{值望值}) * \frac{\text{協商次數}}{\text{協商總數}} \end{array} \right.$$

表 4.8 第五次協商記錄表

屬性值	專案元素	期望目標值	協商區間值	效用值	提案拒絕或接受
UA	Cost	20 萬元	[15,25]萬元	20	提案被拒絕
	Time	30 天	[23.2, 35.8] 天數	29.5	
	Scope	1	[1,2]	1	
	Quality	4	[4,5]	4.5	
	Human Resource	3	[1,3]	2	

VA1	Cost	25 萬元	[20,30]萬元	25	接受提案
	Time	32 天	[30,40]天數	35	
	Scope	1	[1,2]	1	
	Quality	5	[4,5]	4.5	
	Human Resource	2	[1,2]	1.5	
VA3	Cost	28 萬元	[24,32]萬元	28	提案被拒絕
	Time	30 天	[30,40]天數	38	
	Scope	1	[1,2]	1	
	Quality	4	[4,5]	4.5	
	Human Resource	2	[1,2]	1.5	

第六階段：分析協商結果：

以第三章協商架構圖分析，判斷協商是不是成功，如協商成功，即無需進入基因演算法，視同專案已成立。不成立，即進入基因演算法，提出最佳效益之專案計畫書給決策者做決策，並不視同專案成立。以此案例，協商是成功的，所以不需要進入遺傳演算法。假設協商成立，在專案管理進行中仍需監控各項進度，所以圖 4.4 表示目前各代理人在專案進行中的溝通機制，IT 專案管理代理人將會負責統籌管理各個代理人的訊息及專案進度。此部份只提供訊息 IT 專案管理代理人在專案進行中的資訊，假設需要技術支援代理人提供技術支援，即會啟動系統，由 IT 專案管理人發出 request 給技術支援代理人，並得到從技術支援代理人得到相關資訊，在本研究中並無做分析及開發。圖 4.5 為一代表，如 IT 專案管理代理人須與其它代理人溝通，只須替換使用者代理人即可。

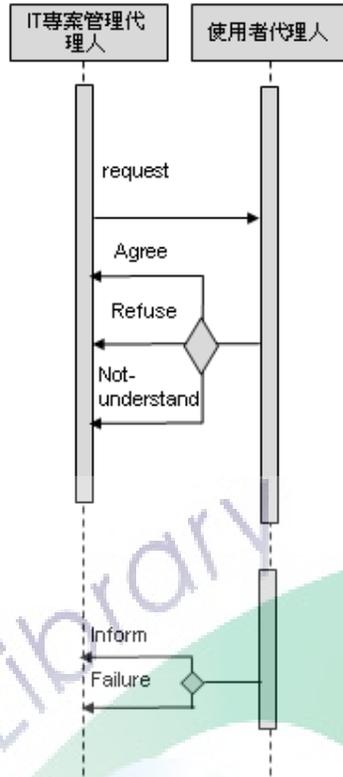


圖 4.5 IT 專案管理代理人與使用者代理人之溝通機制

第七階段：模擬協商未成功的情況，進行基因演算法進行演化，以一樣的實例進行模擬，為了進行實驗，本例將 UA 的 Quality 協商區間值定為[3,4]，此值的改變對於協商結果會有很大的改變，表示 UA 為了進行協商，不斷地提高成本，但時間卻同時只能縮短，但這種情況下，專案品質必定會受影響，所以提案不被接受。而 VA1、VA3 所提供的提案成本又太高，無法符合 UA 預算，基於這幾項因素，協商最終沒有成立，如表 4.9 及圖 4.6 所示。

表 4.9 第十次協商記錄表

屬性值	專案元素	期望目標值	協商區間值	效用值	提案拒絕或接受
UA	Cost	20 萬元	[15,25]萬元	25	提案被拒絕
	Time	30 天	[23.2,35.8] 天數	23.2	

	Scope	1	[1,2]	1	
	Quality	4	[3,4]	3	
	Human Resource	3	[1,3]	2	
VA1	Cost	25 萬元	[20,30]萬元	30	提案被拒絕
	Time	32 天	[30,40]天數	30	
	Scope	1	[1,2]	1	
	Quality	5	[4,5]	4	
	Human Resource	2	[1,2]	1	
VA3	Cost	28 萬元	[24,32]萬元	32	提案被拒絕
	Time	30 天	[30,40]天數	40	
	Scope	1	[1,2]	1	
	Quality	4	[4,5]	4	
	Human Resource	2	[1,2]	1	

Inupt:

P:專案的集合， $P=\{P_1, P_2, \dots, P_m\}$ ，專案 P_m : 專案 m 的集合

Output:

FN:最佳專案集合。

Tasks:

1.編碼:

//益本比計算

```
public double BC(int Bt,int Ct,int n,int dis)
```

```
{
    double sum=0,sum1=0,sum2=0;
    for(int i=0;i<n;i++) //Bt sum
    {
        double t=Math.pow((1+dis),(double)i);
        sum1+=(Bt/t);
    }
    for(int i=0;i<n;i++) //Ct sum
    {
        double t=Math.pow((1+dis),(double)i);
        sum2+=(Ct/t);
    }
    sum=sum1/sum2;
    return sum;
}
```

//益本比編碼

```
for (int i = 0; i < project.length; i++) {
    int p[i] = project[i];
    System.out.println(Integer.toBinaryString(p[i]));
}
```

2.計算各個提案的適存值

3.select

4.crossover

5.突變

6.依演化世代重複進行以上流程

圖 4.6 基因演算法示意圖

4.2 雛型協商系統說明與實驗結果分析

在本研究 4.1 節中爲了驗證協商機制的可行性，已實際演算出協商的結果。因此，本節將詳細說這個由 JADE 及 JAVA 來實作的雛形協商系統。其中包含了基因演算法的運作，此演算法已在第三章研究方法介紹過，當協商不成功時，會啓動基因演算法，試圖找出最佳專案組合。在過程中，將會以 4.1 節的實例來進行詳細的系統模擬說明。

4.2.1 協商雛型系統建置說明

在進行協商機制的過程中，需將協商參數輸入，在 4.1 有介紹過 ua 代表使用者代理人、ia 代表 IT 專案管理代理人、va1 代表廠商 1、va2 代表廠商 2、va3 代表廠商 3。因此爲了方便說明，後續陳述皆以代號表示。在此，特別說明的一點是參數的順序是成本期望值，成本下限，成本上限，成本彈性調整值，時間期望值，時間下限，時間上限，時間彈性調整值，範疇期望值，範疇下限，範疇上限，範疇彈性調整值，品質期望值，品質下限，品質上限，品質彈性調整值，人力資源期望值，人力資源下限，人力資源上限，人力資源彈性調整值。以下爲程式碼：

```
//宣告物件
```

```
NegotiationCounters ua = new NegotiationCounters(20,15,25,0.2,30,24,35,0.2,1,1,2,0.1,4,4,5,0.1,3,1,3,0.3);  
NegotiationCounters ia = new NegotiationCounters(18,12,25,0.1,26,20,28,0.1,1,1,2,0.1,5,4,5,0.1,3,1,3,0.2);  
NegotiationCounters va1 = new NegotiationCounters(25,20,30,0.3,32,30,40,0.3,1,1,2,0.1,5,4,5,0.1,2,1,2,0.2);  
NegotiationCounters va2 = new NegotiationCounters(35,33,38,0.3,30,30,40,0.3,1,1,2,0.1,4,3,5,0.1,2,1,3,0.2);  
NegotiationCounters va3 = new NegotiationCounters(28,24,32,0.3,30,30,40,0.3,1,1,2,0.1,4,4,5,0.1,2,1,2,0.2);
```

爲了計算出各個代理人所提出的五個協商屬性的最大集合，以便找出最多代理人認同的屬性區間，程式利用以下這段程式碼來找出每個屬性最大集合的下上限。最大集合之協商區間值如圖 4.7 及圖 4.8，因此就可以計算出各個代理人的屬性區間值是否有符合，舉例說明，如 ua 的成本協商區間值爲[15,25]，而成本屬性的最大集合爲[24,32]，因此就可以演算出 ua 的成本區間值有落在最大集合內。

```
for (int i=0;i<5;i++){ //進行五回合比較
```

```

for (int j=0;j<5;j++){ //每一回合區間 i 與除了自己之外的其他四個區間 j 比較
    if (j!=i) {
        tmp_upper=interval[i][1]; tmp_lower=interval[i][0];
        if (interval[j][0]<interval[i][1]&&interval[j][1]>interval[i][0]){
            tmp_intersection++; //計算交集數
            if (interval[j][1]<tmp_upper) tmp_upper=interval[j][1];
            if (interval[j][0]>interval[i][0]) tmp_lower=interval[j][0];
        }
    }
} //for j
if (!initial){ //若為第一回合則不用比較
    if (tmp_intersection>intersection){ //交集數最大者優先存入長期記憶
        intersection=tmp_intersection;
        upper=tmp_upper;
        lower=tmp_lower;
    }
    else if (tmp_intersection==intersection){ //交集數相同者取大區間存入長期記憶
        if ((tmp_upper-tmp_lower)>(upper-lower)){
            upper=tmp_upper;
            lower=tmp_lower;
        }
    }
}
else initial=false;
//System.out.println(intersection + "," + upper + "," + lower);
} //for i

```

輸出 - ProjectAgent (run)

```

run:
cost下限:24.0,上限:32.0
time下限:30.0,上限:40.0
scope下限:1.0,上限:2.0
quality下限:4.0,上限:5.0
hresource下限:1.0,上限:2.0

```

圖 4.7 五個協商屬性之區間值

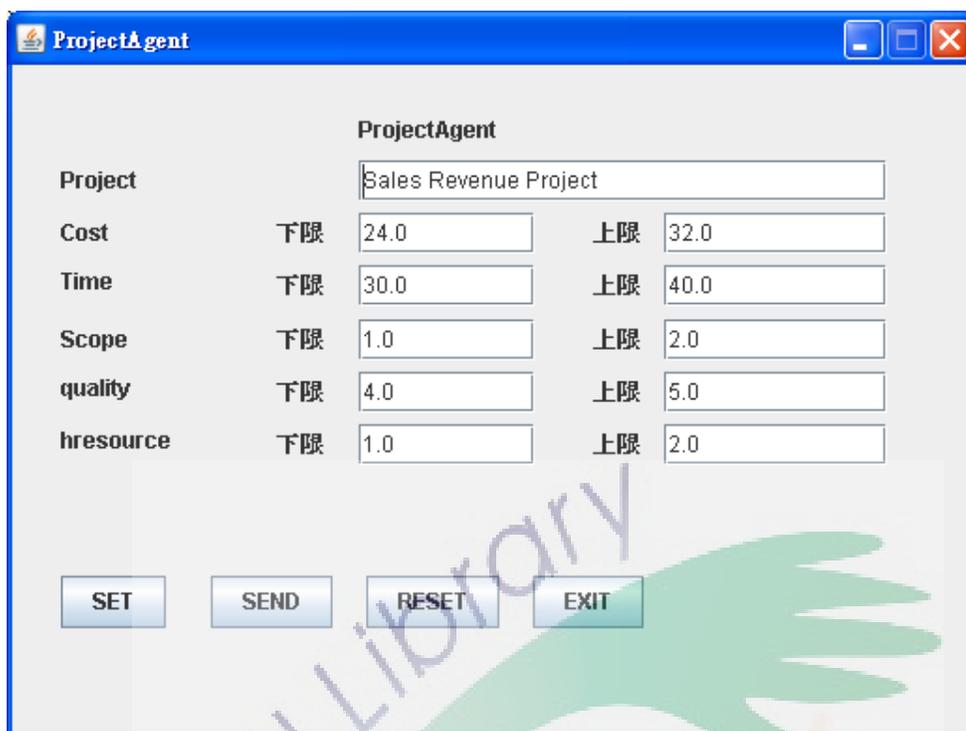


圖 4.8 五個協商屬性之區間值系統畫面

依上述說明類推，可演算出本次實例中需做彈性調整的分別有 $va2_Cost$ 、 ua_Time 、 ia_Time ，利用第三章研究方法之公式 3.1 及公式 3.2 進行區間值調整，程式碼如下：

```
double mean_cost = (ua.getCostExpectedValue()+ia.getCostExpectedValue()+
va1.getCostExpectedValue()+va2.getCostExpectedValue()+va3.getCostExpectedValue())/5;//期望值之平均值
double cost_adj_factor= (va2.getCostExpectedValue()-mean_cost)*va2.getCostElasticity(); //公式 3.1
double getCostLowerUpper [] =
{ va2.getCostLower()-Math.abs(cost_adj_factor),va2.getCostUpper()+Math.abs(cost_adj_factor)};//公式 3.2
```

在進行第一次屬性調整後， ia 的 $Time$ 屬性及 $va2$ 的 $Cost$ 屬性仍沒有落在最大集合，如代理人接到通知後仍要進行屬性調整，即會有第二次屬性調整，但本實驗設定 ia 及 $va2$ 因此放棄協商，因此最終進入協商的有 $ua, va1, va3$ 。
`yieldNegotiationupright` 這個方法的參數共有六個，`Lower` 為各個代理人提出的屬性下限、`expected` 為期望值、`Upper` 為屬性上限、`slcountname` 為進入協商的代理人名

稱、negotiationtime 為協商總次數、slcount 為進入協商的代理人個數，退讓協商程式碼如下：

```
//定義退讓協商方法
yieldNegotiationupright(double [][] Lower,double [][] expected,
                        double [][] Upper, String [] slcountname, int
negotiationtime, int slcount)
for (int ll=0; ll<(Lower.length+Upper.length)/2; ll++) {
    for (int sl=0; sl<slcount; sl++) {
        if (ll==0){ utility = Lower[ll][sl] + ((Upper[ll][sl] - Lower[ll][sl])*((double)negotiationtime/10));
        }
        if (ll==1 || ll==3 || ll==4){ utility = Upper[ll][sl] - ((Upper[ll][sl] -
Lower[ll][sl])*((double)negotiationtime/10));}
        if ((negotiationtime/2) !=0){
            if (ll==2){ utility = expected[ll][sl] + (expected[ll][sl] -
Lower[ll][sl])*((double)negotiationtime+1/10);}}
        if ((negotiationtime/2) ==0){
            if (ll==2){ utility = expected[ll][sl] + ((expected[ll][sl] -
Lower[ll][sl])*((double)negotiationtime/10));}}
        for (int llck=0; llck<slcount; llck++) {
            if (utility <= Upper[ll][llck] && utility >= Lower[ll][llck]) {count[ll][sl]++;}
            if (count[ll][sl] == slcount) {realize[sl] = realize[sl] + count[ll][sl];}
            (negotiationtime==5){System.out.println(ll+", "+sl
+ ", "+count[ll][sl]+ ", "+realize[ll]);} //debug
            if (realize[sl]==5*slcount) {rzname = rzname + " ; " + slcountname[sl];}
        }
    }
}
```

圖 4.9 及圖 4.10 為本實驗之驗證結果，設定之總協商次數為 10 次，因此在第五次協商即產生第一次協商成功，本例進行到第十次協商仍只有一次成功，因此 va1 是本次實驗中唯一符合期待的專案組合，但如果協商有第二次成功仍可比較前後次的成功協商最符合期待。

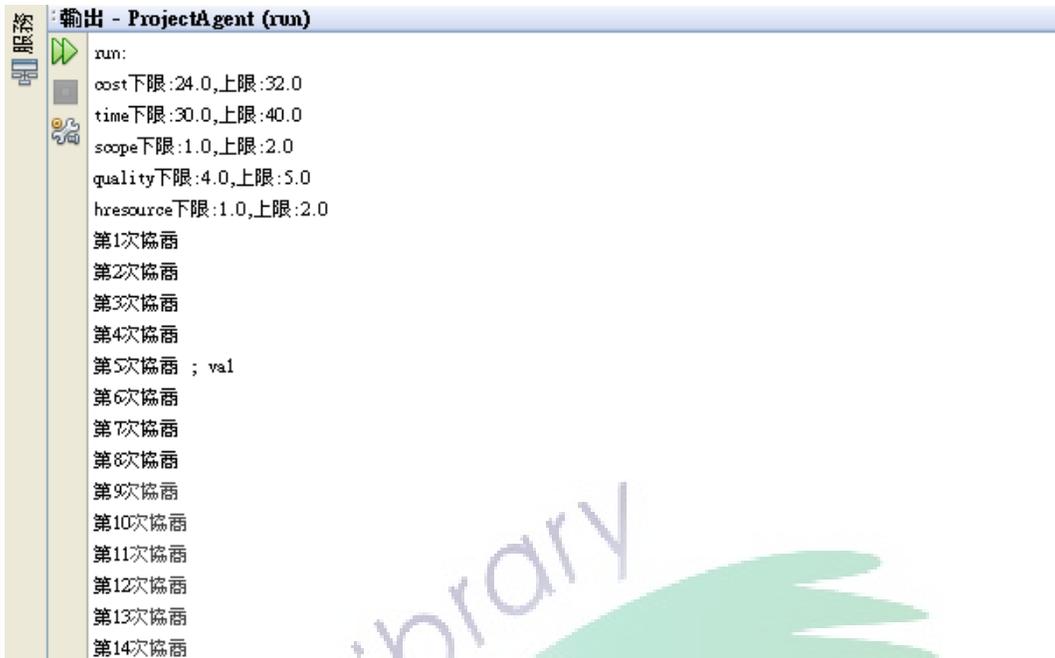


圖 4.9 協商第五次成功之表示圖

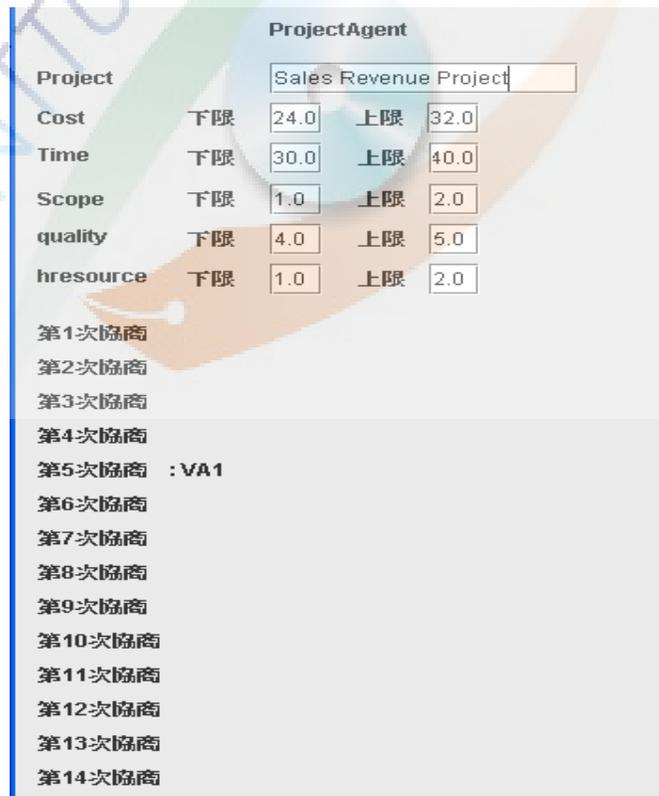


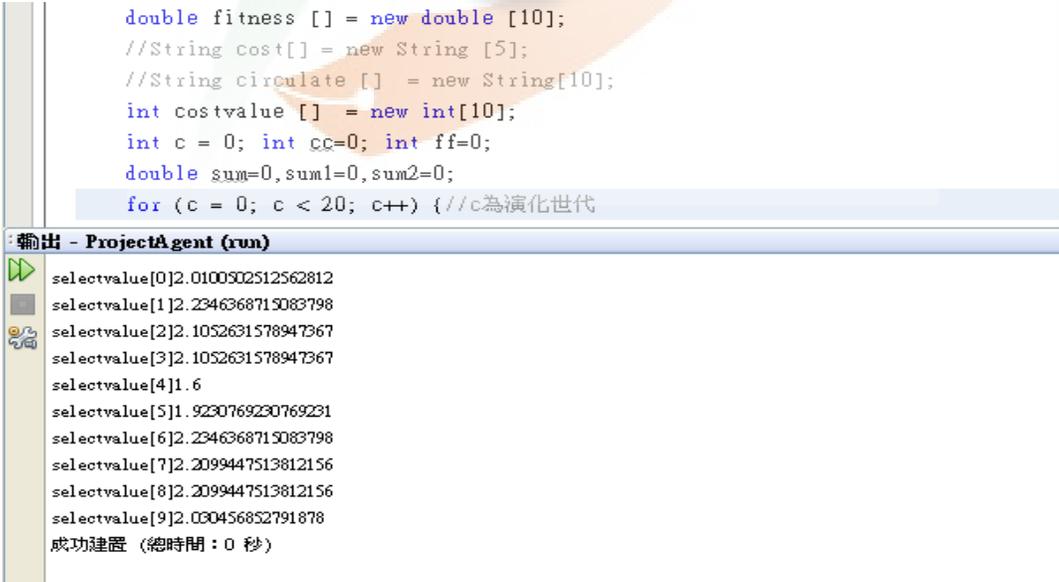
圖 4.10 協商第五次成功之系統畫面圖

4.2.2 基因演算法之結果分析

由於在第三章研究方法中有提到，當協商不成功時，系統會啟動基因演算法機制，運用物競天擇的方式找出最佳專案組合。在此小節延用第三章研究方法中的模擬例子，以程式產生演化世代 20、50、100 的最佳染色體，發現同樣連續各進行十次的結果，在第 100 代的結果較為穩定，最佳成本效益之益本比都在 2.0~2.5 之間，在第三章研究方法中設定的模擬例子之基因演算法參數為交配率=1、突變率=0.05，因此在此條件下，圖 4.11 為演化世代為 20 的最佳值極不穩定，落差也比較大，而圖 4.12 的演化世代為 50，最佳值漸趨於穩定，但仍未達到完全穩定，以圖 4.13 來看，所被設定之演化世代為 100，其產生的最佳值已非常穩定，所以本模擬例子的實驗結果為演化世代為 100，這個世代產生的專案組合較優秀。

在基因演算法機制中，所產生的最佳專案組合僅提供參與者參考，因此必須將所有符合的最佳專案，以報表呈現方式提供給參與者，本研究探討的是運用成本效益法則是否能有效的產生讓參與者滿意的提案，所以報表的呈現未在此列出，在此僅做簡單的敘述。

```
double fitness [] = new double [10];
//String cost[] = new String [5];
//String circulate [] = new String[10];
int costvalue [] = new int[10];
int c = 0; int cc=0; int ff=0;
double sum=0, sum1=0, sum2=0;
for (c = 0; c < 20; c++) { //c為演化世代
```



```
selectvalue[0]2.0100502512562812
selectvalue[1]2.2346368715083798
selectvalue[2]2.1052631578947367
selectvalue[3]2.1052631578947367
selectvalue[4]1.6
selectvalue[5]1.9230769230769231
selectvalue[6]2.2346368715083798
selectvalue[7]2.2099447513812156
selectvalue[8]2.2099447513812156
selectvalue[9]2.030456852791878
成功建置 (總時間: 0 秒)
```

圖 4.11 演化世代為 20 的驗證結果

```
Start Page x GAsselect.java x Negotiation.java x
double fitness [] = new double [10];
//String cost[] = new String [5];
//String circulate [] = new String[10];
int costvalue [] = new int[10];
int c = 0; int cc=0; int ff=0;
double sum=0, sum1=0, sum2=0;
for (c = 0; c < 50; c++) { //c為演化世代

輸出 - ProjectAgent (run)
selectvalue[0]1.8518518518518519
selectvalue[1]2.2222222222222223
selectvalue[2]2.272727272727273
selectvalue[3]2.272727272727273
selectvalue[4]2.272727272727273
selectvalue[5]1.8867924528301887
selectvalue[6]2.272727272727273
selectvalue[7]2.2222222222222223
selectvalue[8]2.272727272727273
selectvalue[9]2.272727272727273
成功建置 (總時間：1 秒)
```

圖 4.12 演化世代為 50 的驗證結果

```
Start Page x GAsselect.java x Negotiation.java x
double fitness [] = new double [10];
//String cost[] = new String [5];
//String circulate [] = new String[10];
int costvalue [] = new int[10];
int c = 0; int cc=0; int ff=0;
double sum=0, sum1=0, sum2=0;
for (c = 0; c < 100; c++) { //c為演化世代

輸出 - ProjectAgent (run)
selectvalue[0]2.3391812865497075
selectvalue[1]2.3391812865497075
selectvalue[2]2.3391812865497075
selectvalue[3]2.3391812865497075
selectvalue[4]2.3391812865497075
selectvalue[5]2.3391812865497075
selectvalue[6]2.3391812865497075
selectvalue[7]2.3391812865497075
selectvalue[8]2.3391812865497075
selectvalue[9]2.3391812865497075
成功建置 (總時間：1 秒)
```

圖 4.13 演化世代為 100 的驗證結果

第五章 研究結論

5.1 結論

由於專案可以說是暫時被賦與責任去創造一個獨特的產品、服務或結果。而專案屬於“暫時”性的活動，”暫時”並不是指專案的時間很短暫時，而是指臨時被賦與責任來完成一個特定的計畫。而不是常態性的活動，因此必須在特定的時間內組成一個團隊，進行資源的整合及調度，在這期間，必須進行相當多次的協商，才能確保資源已整合，進而運用。

本研究提出之代理人協商應用於銀行 IT 專案管理之研究，主要是針對當銀行 IT 部門接到需求，而需成立專案時，在建立專案計畫書前，利用協商代理人及其它相關代理人的協商溝通機制。以協調、調整、退讓等模仿人類行為的方式進行，而協商機制是透過這個多人同意的方式，而挑選出符合眾多數參與者同意之提案，而成立專案。目的為在成立專案時，當進行多人協商時及眾多協商屬性時，代理人能透過協商機制，找出被最多人接受的提案。在第二章文獻中有提到整合型協商被定義為彼此對於協商之相關屬性存有不同偏好，因此協商雙方會試圖找到雙方的最大利益。因此本研究的目的之一是找到一個方式，增加雙方都可以互惠的機會，希望能促成更多的協商成功，達到 win-win 的局面。

另一研究目的為當透過代理人協商機制，仍無法協商成功時，本研究運用基因演算法試圖找出以成本效益法則為基礎的最佳組合，提供給相關人員參考。一方面是希望產生的成本效益最佳提案組合能被相關參與者參考與接受。另一方面是期待在未來的研究能有相關的突破，應用參與者的回饋及意見，建立一個知識儲存系統。代理人協商系統可與之配合，當代理人協商系統接收到更多意見時，可做為資訊儲存，並運用在以後的協商機制上的參考依據。並利用遺傳演算法做更多屬性的進化，促使代理人協商系統更趨於人性化。

5.2 未來研究方向

本研究對於後續之開發及研究有以下幾點建議：

- (一) 由於本研究並沒有對協商屬性設定權重、影響性的分析，往後研究可進行這方面的分析，探討如有設定時，對於專案的成立有多少影響，有哪方面影響等面向探討。
- (二) 本研究目前只進行 IT 專案管理代理人、使用者代理人、廠商代理人來進行多人協商，而這些代理人所關心的元素都一樣，並無差別，未來研究可加入其它代理人，如顧問代理人、技術支援代理人，工程師代理人，或是贊助者代理人，這些代理人關心的元素可能就有所差別，如工程師代理人會關心的是專案時間是否足夠讓他做開發工作，以不同比例的元素來做協商，相信代理人協商機制會更趨於人性化。
- (三) 本研究在第二章文獻探討時，有討論到分配型協商及整合型協商，前者是完全在對立的角度上思考及進行協商，後者則是因為注重的因素不同也可以進行以彼此雙方皆互惠的協商，目前本研究之探討中只有做整合型協商的實例驗證，並未做兩者之間的分析及探討，未來研究也可進行此方面的探討。這在人類協商之間也會有此現象，相信也可以促使協商機制能有多更發展。

參考文獻

一、 中文部份

- [1] 王世甫 (2006). Mareg—以行動代理人與網格經濟為基礎之網格資源管理模型. 資訊管理研究所. 中壢市, 國立中央大學. 碩士論文.
- [2] 王東星 (2007). 考慮勞工能量消耗與時間雙目標最佳化之工作排程研究. 營建工程系碩士論文班. 台中縣, 朝陽科技大學. 碩士論文.
- [3] 王亮宗 (2007). 數位家庭資源管理系統. 資訊工程學系. 台北縣, 輔仁大學. 碩士論文.
- [4] 朱玉芬 (2000). 應用基因演算法在專業 ic 設計業的供應鏈生產排程之研究. 資訊管理學系. 台北縣, 輔仁大學. 碩士論文.
- [5] 朱宏揚 (2000). 自動化之整合協商代理人. 資訊管理研究所. 台北市, 國立臺灣大學. 碩士論文.
- [6] 吳宗翰 (2002). 代理人為基的供應鏈供需自動協商系統. 工業工程學研究所. 台北市, 國立臺灣大學. 碩士論文.
- [7] 吳振嘉 (2003). 應用基因演算法規劃具間歇性負載工廠之變壓器容量. 電機工程研究所. 中壢市, 中原大學. 碩士論文.
- [8] 李文政 (2004). 結合基因演算法與 SIRMs 模糊控制器於倒單擺系統控制. 機械與機電工程學系研究所. 高雄市, 國立中山大學. 碩士論文.
- [9] 李彥邦 (2005). 智慧型代理人與資訊分享研究 ——以虛擬寵物為例. 國家發展研究所. 台北市, 臺灣大學. 碩士論文.
- [10] 林宗政 (2005). 遺傳演算為基的研究所入學考試時間表及考場排程法. 工業工程學研究所. 台北市, 臺灣大學. 碩士論文.
- [11] 林俊雄 (2008). 設計鏈作業參考模式之多代理人系統開發. 工業管理系. 台北市, 國立臺灣科技大學. 博士.
- [12] 林啓明 (2003). 以代理人為基礎的線上協商支援系統. 資訊管理學系研究所. 台中縣, 靜宜大學. 碩士論文.

- [13] 張烈堂 (2005). 基因演算法於建構臺指選擇權投資組合的應用. 資訊管理研究所. 中壢市, 國立中央大學. 碩士論文.
- [14] 張紹勳 (2000). 研究方法. 台北: 滄海書局.
- [15] 許誌雄 (2003). 應用顧客關係管理制定電子商務代理人之議價策略. 工業工程與管理系碩士論文班. 台中縣, 朝陽科技大學. 碩士論文.
- [16] 郭昱瑩 (2007). 成本效益分析. 台北, 華泰文化事業股份有限公司.
- [17] 陳以明、王世昌 (2006). "專案協同合作之代理人補償式之策略協商模式."
Journal of e-Business 8(13): 415-446.
- [18] 曾偉育 (2004). 智慧型交通模擬系統之物件導向分析與設計. 資訊管理系碩士論文班. 台中縣, 朝陽科技大學. 碩士論文.
- [19] 楊景隆 (2006). 導入習慣領域學說及智慧型代理人於知識管理系統之研究.
資訊管理系. 台南, 南台科技大學. 碩士論文.
- [20] 盧賢豪 (2003). 知識管理系統中運用智慧型代理人之研究. 資訊管理學系研究所. 高雄市, 國立中山大學. 碩士論文.
- [21] 賴郁玲 (2005). 以基因演算法求解最少延遲工作數下總延遲時間最小化之單機排程問題. 工業管理研究所. 台南, 南台科技大學. 碩士論文.
- [22] 謝穎欣 (2005). 應用田口方法於基因演算法輸入參數設計--以求解多模式專案排程下資源撫平為例. 工業管理研究所. 中壢市, 國立中央大學. 碩士論文.
- [23] 魏成瑋 (2007). 以代理人社群為基礎解決人力資源配置問題. 工業工程與管理學系. 中壢市, 元智大學. 碩士論文.
- [24] 石育樑 (2005). 動態平行機雙準則排程啟發式演算法之研究. 工業管理系. 台北市, 國立台灣科技大學. 碩士論文.
- [25] 劉邦權 (2005). 開發以 Belief-Desire-Intention 為基礎的代理人於 ebXML CPA 協商, 中壢市, 中原大學. 碩士論文.

二、英文部份

- [1] Anson, R. and M. Jelassi (1990). "A development framework for computer-supported conflict resolution." European Journal of Operational Research 46(2): 181-199.
- [2] B.LEE, F. N. K. H. (2002). Foundations of Behavioral Research. 台北, 華泰文化事業公司.
- [3] Bazerman, M., B. Sheppard, et al. (1986). Research on negotiation in organizations, JAI Press, Greenwich, Conn.
- [4] Bellifemine, F. L. C., Giovanni;Greenwood, Dominic (2007). Developing Multi Agent Systems with JADE, John Wiley.
- [5] Chun, A., H. Wai, et al. (2003). "Optimizing agent-based meeting scheduling through preference estimation." Engineering Applications of Artificial Intelligence 16(7-8): 727-743.
- [6] Chun, H. and R. Wong (2003). "N —an agent-based negotiation algorithm for dynamic scheduling and rescheduling." Advanced Engineering Informatics 17(1): 1-22.
- [7] Durfee, E., V. Lesser, et al. (1989). "Trends in cooperative distributed problem solving." IEEE Transactions on Knowledge and Data Engineering 1(1): 63-83.
- [8] Heerkens, G. R. (2001). "Project Management." from <http://www.ipc.itri.org.tw/content/menu-sql.asp?pid=122>.
- [9] Holland, J. (1975). "Adaptation in natural and artificial systems. ." Ann Arbor MI: University of Michigan Press.
- [10] Holland, J. (1992). Adaptation in natural and artificial systems, MIT Press Cambridge, MA, USA.
- [11] Hsu, Y.-C. (2006). Countering Adversarial Strategies in Multi-Agent Virtual Scenarios, TSING HUA UNIVERSITY. Dissertation.

- [12] Huhns, M. and L. Stephens (1999). Multiagent systems and societies of agents, Multiagent systems: a modern approach to distributed artificial intelligence, MIT Press, Cambridge, MA.
- [13] Institute, P. M. (2008). A Guide to the Project Management Body of Knowledge, Project Management Institute.
- [14] Keesoo, K., C. Boyd, et al. (2000). Compensatory negotiation for agent-based project schedule optimization and coordination.
- [15] Kraus, S. (2001). Strategic negotiation in multiagent environments, The MIT press.
- [16] Kuo, C.-Y. (2001). An Internet Agent-Based Communication System for Construction Project Scheduling, National Taiwan University. Dissertation.
- [17] Lai, H. (1989). "A theoretical basis for negotiation support systems." Unpublished doctoral dissertation, University of Purdue, August.
- [18] Lee, S.-J. (2007). Role-Based Contracting with Risk-Enabled Reputation Model in Agent Organizations, National Central University. Dissertation.
- [19] Liang, T. and H. Doong (1999). Effect of bargaining in electronic commerce.
- [20] Liang, Y.-F. (2003). A BDI-Based Multiagent System Development Environment, National Chiao Tung University. Dissertation.
- [21] Marsh, P. (2001). Contract negotiation handbook, Gower Publishing Company, Limited.
- [22] Nien, C.-T. (2006). Social Constraints Generation by Social Motivation for Multi BDI Agents, TSING HUA UNIVERSITY. Dissertation.
- [23] Oliver, J. (1997). "A machine-learning approach to automated negotiation and prospects for electronic commerce." Journal of Management Information Systems 13: 83-112.
- [24] Raffia, H. (1982). The art and science of negotiation, Harvard University Press, Cambridge.

- [25] Raiffa, H., J. Richardson, et al. (2002). Negotiation analysis: the science and art of collaborative decision making, Belknap Press.
- [26] Russell, D. P. C. L. D. (1997). Molecular Biology Made simple and fun, Cache River Press.
- [27] San, M. a. C. (1998). " Intelligent agents on the Internet and Web." IEEE Region 10 International Conference on GlobalConnectivity in Engery, Computer; Communication 1: 97-102.
- [28] Tsai, P.-L. (1999). Design and Implementation of a Versatile Mobile Agent System, National Taiwan University. Dissertation.
- [29] Wooldridge, M. and N. Jennings (1995). "Intelligent agents: Theory and practice." Knowledge engineering review 10(2): 115-152.
- [30] Ye, Y.-H. (2006). Development of a Multi-agent Software Platform for Context-aware Digital Home Applications and Its Environment Simulator, National Cheng Kung University. Dissertation.
- [31] Zhao, L., W. Ng, et al. (2001). Cooperative multi-attribute bilateral online negotiation for E-Commerce, IEEE Computer Society Washington, DC, USA.
- [32] Nwana, H., S. (1996). The Potential Benefits of Software Agent Technology to BT, Internal Technical Report, Project NOMADS, Intelligent Systems Research, AA&T, BT Labs,UK.
- [33] Nwana, H., S. and Wooldridge, M. (1996). Software Agent Technology, BT Technology Journal, Vol. 14, No. 4, pp.68-78.
- [34] Nwana, H., S. (1996). Software Agents: An Overview, The Knowledge Engineering Review, Vol. 11, No. 3, pp.205-244.
- [35] Huhns, M. N., and Stephens, L. M. (1999). Multiagent system and societies of agents, Multiagent systems: a modern approach to distributed artificial intelligence, The MIT Press, Cambridge, MA, USA, 79-120.

- [36] Kim, K. W., M. Gen, and G. Yamazaki. (2003). Hybrid Genetic Algorithm with Fuzzy Logic for Resource-Constraint Project Scheduling, *Applied Soft Computing*, Vol. 2, No.3, pp.174-188.
- [37] Weimer, David Leo. and Aidan R. Vining. (1992). *Policy analysis*. Englewood Cliffs, N.J: Prentice Hall.
- [38] Maes, P. and R. Guttman "Moukas. A.(1999). Agents that buy and sell: Transforming commerce as we know it." *Communications of the ACM* 42(3): 81-91.
- [39] McBurney, P. and S. Parsons (2003). *Democracy in open agent systems*, ACM New York, NY, USA.

