

逢甲大學
交通工程與管理學系碩士班
碩士論文



KSP 演算法於 CORBA-Based 分散式架構
績效評估之研究

Performance Evaluation of KSP Algorithm in
CORBA-Based Distributed Architecture

指導教授：胡大瀛

研 究 生：洪百賢

中華民國九十二年二月

摘要

即時資訊與動態分析是智慧型運輸系統 (Intelligent Transportation Systems, ITS) 中的重要特性，在 ITS 架構中，交通屬性會隨著時間與空間而改變，而負責蒐集交通相關資訊的責任又分屬不同的單位組織。為能整合各類資訊以提供更完善的資訊服務，分散式計算與處理系統 (Distributed Computing and Processing System) 扮演相當重要的角色。以往所使用的架構由於各單位所使用的電腦系統差異愈來愈大，已無法提供有效的解決方案。因此，目前分散式系統已趨向於以 n-tier 架構為主，除增加系統建置彈性外，更降低系統轉移的困難度。

在 ITS 次系統中，路徑選擇功能被廣泛應用於旅行者相關領域，如路徑指引與車輛派遣，其重要基本概念為替代路徑的產生，而替代路徑可經由 K 條最短路徑 (K-Shortest Paths, KSP) 演算法產生。由於 KSP 演算法包含複雜的計算過程，在分散式系統中會對整體執行績效有相當大的影響。為能瞭解其影響因素與影響程度，本研究先定義出四個績效指標以評估系統整體績效，包含回應時間、執行時間、錯誤率、及記憶體使用量，並藉由數值實驗所得結果分析出各項指標之變化與實驗資料之關係。

數值實驗以分散式 GIS-T 及分散式 DYNASMART 兩範例進行測試，其中分散式 GIS-T 系統主要是測試回應時間與錯誤率兩項指標；分散式 DYNASMART 則測試執行時間與記憶體使用量兩指標。實驗結果顯示出兩項結論，一是計算所使用的資料型態會對執行績效有相當大的影響；另一是計算元件增加不會使計算效率成等比例增加，其增加趨勢會隨元件數增加而趨緩，同時降低記憶體使用量的績效。

關鑑字：智慧型運輸系統、分散式架構、CORBA、K 條最短路徑演算法、績效評估

Abstract

Two important characteristics of ITS (Intelligent Transportation Systems) are real-time information and dynamic analysis. With the advancement of communication and computer techniques, several subsystems of ITS, such as ATIS (Advanced Traveler Information Systems), ATMS (Advanced Traffic Management Systems), logistics, and emergency management system, are aiming at real-time response and decentralization. In order to efficiently analyze real-time traffic-related information, several components are essential, including data collection, distributed computing, and two-way communicate ability. The client-server architecture has been used in various distributed architectures; however, the multi (n)-tier architectures (such as CORBA, DCOM) have been developed to overcome the limitation of client-server architecture, such as platform and programming language dependency.

In this research, a general distributed architecture will be proposed and simulated for various ITS applications. In several ITS subsystems, route choice has been widely used to for travelers, such as route guidance and vehicle dispatching. One fundamental concept in route choice is the generation of alternative routes, which can be modeled by KSP (K-Shortest Paths). Due to the complexity of KSP, the computation efficiency highly depends on where the KSP is computed in the distributed architecture. In an n-tier distributed architecture, components could be arranged in several ways according to specified characteristics. One of the main objective of this research is to analyze the performance of computation and communication in different distributed ITS architectures. Two major directions of evaluation methods are considered, namely, the system-oriented and user-oriented, which will be deployed in the development of the system. In order to simulate the specific architecture, modifications will be made based on the general architecture to meet the required demand.

Distributed GIS-T System and Distributed DYNASMART System are two major examples used to evaluate the four performance indexes in the numerical experiments. Major conclusions from the experiments are: 1) data types have significant impact on execution performance; 2) the variation of computing performance is not proportional to the number of computing objects.

Keywords: Intelligent Transportation Systems, Distributed Architecture, CORBA, KSP Algorithm, and Performance Evaluation

目錄

第一章 前言	1
1.1 研究背景	1
1.2 研究動機	2
1.3 研究目標	3
1.4 研究流程	4
第二章 文獻回顧	6
2.1 智慧型運輸系統	6
2.1.1 ITS 架構	7
2.1.2 ITS 未來發展	8
2.1.3 分散式系統失敗案例：LASCAD 醫療資訊系統	10
2.2 分散式架構	11
2.2.1 平行與分散式架構	12
2.2.2 多層分散式架構 (n-Tier Distributed Architecture)	13
2.2.2.1 Common Object Request Broker Architecture, CORBA ...	13
2.2.2.2 Distributed Component Object Model, DCOM	15
2.2.2.3 Enterprise JavaBeans, EJB	15
2.2.3 分散式架構應用實例	16
2.3 K 條最短路徑問題 (KSP)	17
2.3.1 問題定義	18
2.3.2 K 條最短路徑演算法	20
2.3.2.1 序列式 K 條最短路徑演算法	20
2.3.2.2 平行與分散式 K 條最短路徑演算法	21
2.3.3 K 條最短路徑演算法之應用	22
2.4 系統效能評估	23
2.5 小結	24

第三章 CORBA-Based 分散式架構.....	26
3.1 分散式架構於 ITS 之應用.....	26
3.2 CORBA-Based 先進資訊系統架構.....	28
3.2.1 先進旅行者資訊系統：路徑導引與路線查詢.....	28
3.2.2 CORBA-Based 分散式 GIS-T 之應用.....	30
3.3 CORBA-Based 交通管理系統架構.....	31
3.3.1 先進交通管理系統：動態交通模擬指派系統.....	31
3.3.2 CORBA-Based 分散式 DYNASMART.....	32
3.4 系統效能評估方式.....	34
3.4.1 旅行者導向績效指標.....	35
3.4.2 系統導向績效指標.....	36
第四章 CORBA-Based 分散式 GIS-T 數值實驗.....	38
4.1 系統架構.....	38
4.1.1 分散式 GIS-T 之應用.....	38
4.2 程式架構與流程.....	39
4.2.1 程式架構說明.....	41
4.2.2 程式流程說明.....	42
4.2.3 路段成本分類.....	44
4.3 實驗設計.....	47
4.3.1 測試環境.....	48
4.3.2 軟體.....	49
4.3.3 資料來源.....	49
4.3.4 網路架構.....	51
4.4 系統操作流程.....	52
4.5 實驗結果分析.....	54

第五章 CORBA-Based 分散式 DYNASMART 數值實驗.....	57
5.1 系統架構.....	57
5.1.1 ATMS 系統架構.....	57
5.1.2 分散式動態交通模擬指派系統	58
5.2 程式架構與流程	59
5.2.1 程式架構說明	59
5.2.2 程式流程說明	60
5.3 實驗設計	61
5.3.1 測試環境	62
5.3.2 軟體	62
5.3.3 測試資料	64
5.4 系統操作流程	64
5.5 實驗結果分析	66
第六章 結論與建議.....	68
6.1 結論	68
6.2 建議	69
參考文獻.....	71
附錄	76

圖目錄

圖 1.1	研究流程.....	5
圖 2.1	ITS 與其次系統架構及通訊元件	7
圖 2.2	先進大眾運輸系統資訊系統的三層式架構	8
圖 2.3	GIS-T 架構	10
圖 2.4	物件管理架構 OMA (Object Management Architecture)	14
圖 2.5	DCOM 架構	15
圖 2.6	Enterprise JavaBeans 的契約關係	16
圖 2.7	K 條最短路徑範例及其解.....	19
圖 3.1	ITS 架構示意圖	27
圖 3.2	分散式架構於 ITS 之應用	28
圖 3.3	路徑導引與路線查詢功能分散式架構	29
圖 3.4	分散式 EMS 系統架構	30
圖 3.5	分散式 DYNASMART 架構	34
圖 4.1	CORBA 資料庫應用關係圖.....	39
圖 4.2	程式架構.....	40
圖 4.3	分散式 GIS-T 之應用系統流程 (計算元件置於伺服器端)	42
圖 4.4	分散式 GIS-T 之應用系統流程 (計算元件置於客戶端)	43
圖 4.5	固定旅行時間成本設定流程	45
圖 4.6	動態旅行時間成本設定流程	45
圖 4.7	即時旅行時間成本設定流程	46
圖 4.8	混合旅行時間成本設定流程	47
圖 4.9	CORBA-Based 分散式 GIS-T 實驗環境	48
圖 4.10	伺服器應用程式與 ORB 建立連結	52
圖 4.11	客戶端應用程式啟動	52
圖 4.12	伺服器應用程式接收查詢資訊畫面	53
圖 4.13	伺服器應用程式接收路徑查詢資訊畫面	53
圖 4.14	計算元件選擇路段成本種類	54
圖 4.15	客戶端應用程式顯示路徑查詢結果	54
圖 5.1	管線資料流界面型分散式 DYNASMART 執行流程	59

圖 5.2	分散式 DYNASMART 執行流程	60
圖 5.3	分散式動態交通模擬指派系統流程	61
圖 5.4	分散式 DYNASMART 實驗環境	62
圖 5.5	CORBA 程式發展流程.....	63
圖 5.6	伺服器端計算元件與 ORB 建立連結	64
圖 5.7	DYNASMART 主程式啟動畫面	65
圖 5.8	伺服器端計算元件接收計算請求畫面	65
圖 5.9	DYNASMART 主程式接收計算結果畫面	65
圖 5.10	元件個數與計算時間和回應時間之關係	66
圖 5.11	元件個數與傳輸時間和轉換時間之關係	67



表目錄

表 2.1	CORBA 架構於交通運輸之應用.....	18
表 4.1	路段成本分類	47
表 4.3	測試網路架構	51
表 4.4	回應時間測試結果	55
表 4.5	錯誤率比較	56
表 5.1	處理時間與記憶體測試結果	66



第一章 前言

1.1 研究背景

由於全球人口與經濟成長快速，各個國家的交通問題愈來愈嚴重，尤其是與日俱增的車輛，對各大都市的交通環境造成嚴重的衝擊。整體運輸系統因缺乏有效的管理與操作，無法滿足現有的需求。因此各國均開始著手進行新一代運輸系統的研究發展，希望能透過此一研究增加現有運輸系統的運能及安全，並擴大交通資訊的應用範疇；此一研究即是目前被各國政府廣為發展的智慧型運輸系統 (Intelligent Transportation Systems, ITS)。ITS 的主要目標是希望結合先進電子、控制、及通訊等新興技術，以解決各種交通問題，並加強運輸系統的效率與安全。

早期的 ITS 又稱為智慧型車路系統 IVHS (Intelligent Vehicle Highway Systems)，該系統包含四項 IVHS 的主要工作，包含先進旅行者資訊系統 (Advanced Traveler Information Systems, ATIS)、先進交通管理系統 (Advanced Traffic Management Systems, ATMS)、先進車輛控制系統 (Advanced Vehicle Control Systems, AVCS)、及商用車輛營運 (Commercial Vehicle Operations, CVO)。當美國智慧型運輸系統協會 (Intelligent Transportation Society of America, ITS America) 成立後，便將 IVHS 重新命名為 ITS，並加入先進大眾運輸系統 (Advanced Public Transportation Systems, APTS) 與先進郊區運輸系統 (Advanced Rural Transportation Systems, ARTS) 兩項新的工作。到目前為止，ITS 的研究持續在進行中，並延伸擴充其功能。

在真實的交通狀況中，多數的交通屬性會隨著時間維度與空間維度的改變而改變，而負責蒐集交通相關資訊的責任又分屬不同的單位組織。為能整合分佈在不同時間與空間的資訊以提供完善的資訊服務，分散式計算與處理系統 (Distributed Computing and Processing System) 扮演相當重要的角色。藉由分散式計算與處理系統的建立，

雖然不同單位蒐集與處理交通資訊的方式與工具不盡相同，透過分散式系統的處理仍可進行有效率的資訊取得與分析。因此，欲在複雜的交通環境中取得更精確的資訊與更有效的計算效率，必須藉由分散式技術來整合各單位；當資料在分散式系統中透過遠端設備進行存取時，即時性 (Real-time) 資料蒐集與分析的觀念也同時被實作。透過先進電腦與通訊技術，ITS 中的部分次系統，如 ATIS、ATMS、物流 (Logistics)、及緊急管理系統 (Emergency Management Systems) 等都強調即時性的回應與分散處理的觀念。

在交通應用系統中的一個基本且重要的問題類型為路網分析 (Network Analysis)，主要目標在於求解與路網有關的問題，如最佳路線、路網流量等相關問題。早期求解路網分析問題時，僅採用靜態方式求解；而隨著時代演進，求解方式逐漸轉向動態分析方法，亦即動態路網分析。動態路網分析 (Dynamic Network Analysis) 在 ITS 的應用相當廣泛，主要功能可分為路徑查詢與導引及動態交通指派兩種。過去求解靜態路網分析相關問題時，多是以最短路徑演算法 (Shortest Paths Algorithms) 為求解基礎；對於動態路網分析，最短路徑演算法無法符合替代路徑的需求，因此改以 K 條最短路徑 (K-Shortest Paths, KSP) 演算法為求解動態路網分析問題的基礎。

1.2 研究動機

近年來，ITS 的相關應用與需求成長相當快速且愈形複雜，分散式架構常被利用在 ITS 系統中，其目標主要是有效的解決各種交通問題。雖然分散式系統對各單位之間的通訊連結有相當大的幫助，其系統本身仍有效能及研究開發上的限制，若無法有效研發與管理分散式系統，不但無法增進運輸系統的整體效能，更可能會造成更大的損害。在文獻上有一例子，是由英國所發展的 London Ambulance Service Computer-Aided Despatch system (LASCAD) 計畫 (Beynon-Davies, 1999)，該例中系統開發的過程未依照正確的開發流程，並且在系統實際運作時出現人為上的管理疏失，導致該計畫無法達到預期的功能 (Beynon-Davies, 1999)。

由於分散式系統服務的範圍擴大，負荷量也因而升高，現代的軟硬體雖具有強大的功能，但仍有限制；此外，在建立分散式系統時，可選擇的軟硬體相當多，不同的選擇所提供的效能與功能亦不同。因此，在進行分散式系統研究與開發時，需考量系統的功能需求，進而選擇適當的軟硬體搭配進行系統開發，以建立符合需求的分散式系統。

K 條最短路徑雖然較一般最短路徑有更大彈性，而其演算法較最短路徑演算法更為複雜，且其計算步驟繁複。如此複雜的計算過程對整體系統效能有顯著的影響，若使用較低等級的硬體配備或不適當的演算法，將對系統整體效能造成重大的影響。因此，在開發分散式系統過程中，需考量如何配置類似 K 條最短路徑等需要大量計算的計算元件 (Computing component)，以降低其對系統整體的影響。

1.3 研究目標

ITS 架構本身即具有分散式架構之雛型，各個次系統之間都靠有線或無線通訊方式進行連結，但 ITS 的架構僅為一種觀念的表達，實際系統的研究開發仍有相當多種組合可搭配選擇；計算元件在系統中亦是重要的元件之一，其效能對系統整體效能有相當重大的影響。在早期，分散式系統多為兩層式 (Two-tier) 的伺服器端 (Server-Side) 與客戶端 (Client-Side) 架構，但該架構的缺點是必須兩端要使用相同作業平台或相同程式語言，且系統負荷限制較大。因此，三層式 (Three-tier) 或多層式 (Multi (n) -tier) 架構的觀念已成形並快速的取代過去的兩層或單層式架構，其特點是不限制系統所使用的作業平台與程式語言，讓系統開發的彈性大為增加。本研究將整理各種多層式架構在 ITS 的應用方式，並討論計算元件在架構中所扮演的角色。

本研究是以分析 KSP 演算法在分散式架構中的系統績效為主要目標，並找出影響績效表現的因素；進而提供系統管理者與開發者進行相關系統開發時重要的參考依據。為達成此一目標，研究中將包含二項主要工作，分別為 KSP 演算法在分散式架構中之應用與系統績效評估，以下將就各工作項目給予簡短說明。

1. KSP 演算法在分散式架構中之應用

KSP 演算法於 ITS 中之應用大致上可分為兩種：路線查詢與導引及動態交通指派。由此兩種應用所衍生出來的相關系統種類繁多，但其架構差異並不大，且都具有分散式架構之觀念。因此本研究將對兩架構的特性與用途進行說明，

2. 系統績效評估

完成 K 條最短路徑與分散式架構的整理與討論後，本研究將對各類分散式架構進行績效評估，以求得特定類型的分散式 ITS 所適用的分散式架構。數值實驗部分將以一 CORBA-based 分散式 GIS-T 系統進行測試，績效指標將分為系統導向 (System-oriented) 及使用者導向 (User-oriented) 兩類進行績效測量，如執行時間與通訊時間等。分析各項實驗結果後，提出各類分散式架構的適用範圍與其特性。

1.4 研究流程

本研究之流程如下頁圖 1.1，先針對研究範圍內相關的文獻作一回顧整理，包含 ITS 架構與未來發展、K 條最短路徑之演算法比較與動態路網分析相關應用、及分散式系統之架構與績效評估。將 K 條最短路徑之應用與分散式架構的觀念放入 ITS 架構中，建立可實作之 CORBA-Based 分散式架構；架構完成後將建立測試系統以進行數值實驗。測試過程中將依據績效指標來紀錄各種實驗結果。最後進行結果分析與討論，並撰寫本研究之結論與未來研究之建議。

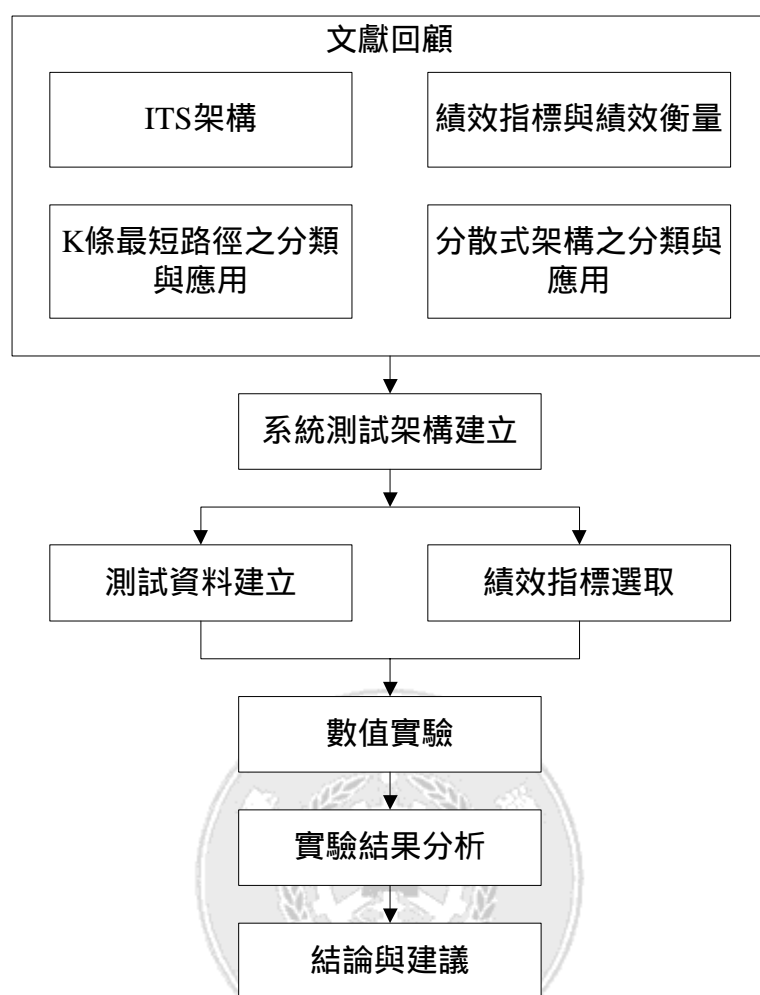


圖 1.1 研究流程

第二章 文獻回顧

2.1 智慧型運輸系統

隨著世界各國經濟與人口的成長，交通需求成長與日俱增，導致各大都市的交通情況逐漸惡化，為解決當前的交通問題，數個先進國家開始著手進行研究對策。在 1980 年代，美國、歐洲和日本等國著手研究 ITS，該系統在歐洲稱為 Prometheus，在日本則稱為 ATIC 及 RACS；ITS 的發展是屬於國家級的規劃，主要是希望將整體運輸環境變的更有效率及具有更大的安全性。無論是任一國家的 ITS 都有類似的目標，就是提供一個更有效率和更具安全的運輸環境，為達到此一目標，在 ITS 中定義了數個次系統，包含 ATIS、ATMS、APTS、ATRS、CVO、及 AVCS。為能建立更有效率的系統，通訊及電腦等相關先進技術將被整合應用於系統開發，其他相關技術亦可輔助開發的過程。ITS 系統開發的詳細資訊可參考美國運輸部所出版之相關著作。

ITS 的主要目標：第 1 目標是增加運輸系統供給與效率，為 ITS 的中心目標，同時也是達成第 2 目標的主要元素；第三個目標主要是希望能改善安全，特別是因為人為疏失所造成的威脅，如 AVCS，透過先進的車輛控制系統，減少駕駛者本身可能的疏失；第 4 目標同時也來自第 1 項，由於系統效率的增加及停等次數的減少，燃料使用得以減少，進而減少對環境的污染，同時降低污染成本。第 5 個目標是由於整體運輸效能提升後，旅客與貨物的運送變得更順暢，對國家整體經濟發展有相當大的助益；第 6 個目標是希望能建立 ITS 的基礎，有助於未來更完善的 ITS 系統的建立。6 項目標列舉如下：(U.S. DOT, 1999)

1. 增加運輸系統的操作效率與容量
2. 加強運輸系統中個人的移動性與便利舒適
3. 改善國家運輸系統的安全性
4. 減少能源消耗與環境成本

5. 增強全體經濟及團體與個人現在與未來的經濟生產力
6. 建立一有助於發展 ITS 的交通環境

2.1.1 ITS 架構

圖 2.1 為 ITS 架構，該架構主要包含了 19 個次系統分別屬於 4 個次系統種類，包含：中心（centers）、路邊（roadside）、車輛（vehicles）、及旅行者（travelers）。管理中心次系統代表具各種功能之行政或管理組織，該組織不需在運輸設施附近，僅需依靠有線通訊方式進行連結即可。路邊次系統具有控制路邊號誌等相關硬體的功能，如號誌、可變訊息號誌（Variable Message Sign）及電子收費掃描器，此類型之次系統通常透過有線的方式與管理中心次系統連結。車輛次系統安裝在各種類型的車輛，其主要功能是以單向或雙向連結的方式進行車輛與車輛或車輛與路邊次系統間的連結。旅行者次系統主要是指旅行者或

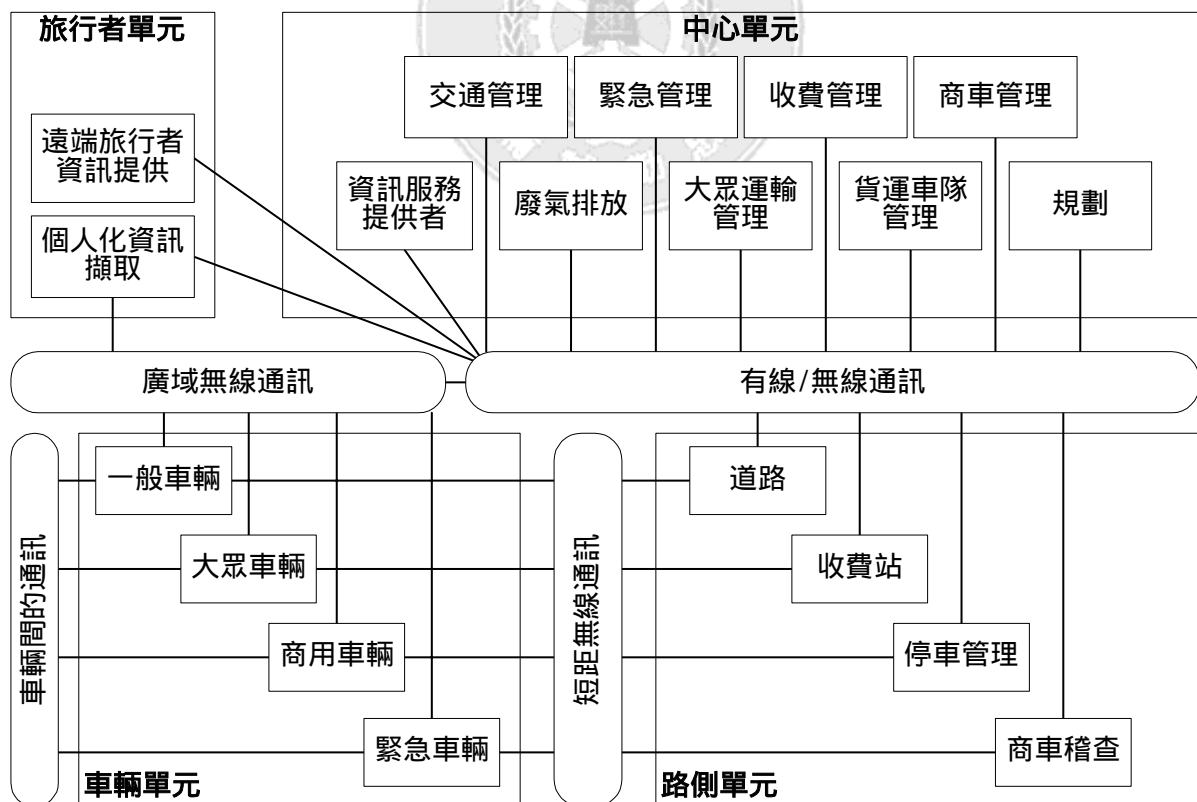


圖 2.1 ITS 與其次系統架構及通訊元件

車輛的操作者，旅行者可透過有線或無線的方式在公開或私人設備上存取管理中心次系統之資訊。

次系統在 ITS 架構中就如同物件導向模式中的物件，次系統代表物件，而現實生活中的實體繼承 (inheritance) 自各種次系統。例如，道路上的車輛是由車輛次系統所繼承；通勤者是繼承旅行者次系統。由於具備此一關係，以 ITS 架構為基礎來發展 ITS 應用系統將變得更有效率 (U.S. DOT, 1999b)。

2.1.2 ITS 未來發展

ITS 是將通訊、電子、資訊等先進技術應用於交通運輸領域，藉由科技的快速發展，ITS 未來的發展亦趨完善。本節將就 ITS 未來發展的重點基礎技術作一說明。

■ 多層分散式架構

ITS 架構本身已具有分散式系統的觀念，因此許多 ITS 系統相關的研究發展亦以分散式架構建立為主。Peng and Huang (2000) 提出先進大眾運輸資訊系統 (Advanced Transit Information System) 的三層式架構，該研究發展以網頁技術為基礎的大眾運輸資訊系統的互動式旅次規劃工具。架構中，網頁瀏覽器為第一層；網頁伺服器為第二層；後端連結資料庫與計算元件等為第三層。透過三層式架構的建立，伺服器與資料分析處理分別位於不同的位置，不但減少伺服器本身的負擔，亦可增加後端資料進行專業處理分析的能力，以提供系統更大的彈性。

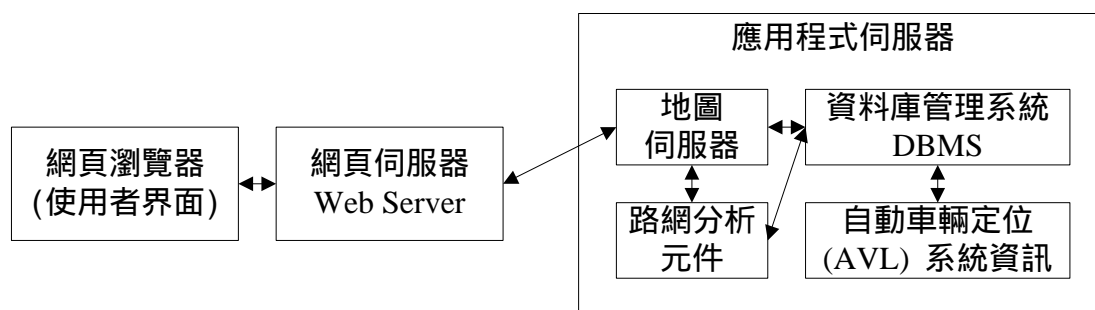


圖 2.2 先進大眾運輸資訊系統的三層式架構

資料來源：Peng and Huang (2000)

■ 無線通訊

在 ITS 架構中，各次系統間的連結通訊方式是相當重要的元件，主要可分為有線及無線兩種連結方式，透過先進的通訊技術，各個獨立的系統可簡單有效率的進行整合。在有線通訊方面，可使用 ATM 或光纖等高速傳輸線路，但有線通訊需鋪設通訊線路，且經鋪設後線路不易變更，因此除非是有大量或頻繁的資料傳輸，避免選擇使用通訊方式。無線通訊方式是目前多數系統傾向使用的通訊方式，在 4 個次系統分類中，旅行者與車輛等系統多數時間是處在移動的狀態下，與該類次系統進行連結必需使用無線通訊方式。在 (Andrisano, Verdone, and Nakagawa, 2000) 中提到，由於資料傳輸量的增加及回應時間要求縮短，高速寬頻通訊方式已成為主要的考慮方案，如行動電話系統，目前所使用的行動通訊為第二代 (2G) 通訊方式，其傳輸速度有限，缺乏使用的彈性；2.5G 的通訊方式已準備普遍化的階段，其速度可達 100KB/s；未來規劃能達到 3G 的通訊方式，其速度將可到達 2MB/s。

■ 地理資訊系統

數位地圖 (Digital Map) 或稱為電子地圖，在許多領域中已有相當廣泛的應用，是目前在 ITS 中用來顯示地圖的主要方式。數位地圖是由數據資料經電腦運算後顯示在螢幕上的地圖，其優點是具較高之精確度、容易修改、使用方便等。地理資訊系統 (Geographic Information Systems, GIS) 則是用來管理與實作數位地圖的系統，地理資訊系統結合數位地圖與相關資料，將資料分析後的結果顯示在數位地圖上。GIS 在規劃、工程、與營運方面都有其相關應用，透過網路技術的輔助，網路地理資訊系統更擴大其應用範圍。Thill (2000) 以交通的觀點描述 GIS 的應用，Thill 認為 GIS 可在交通研究與管理作為一整合性的系統。Ziliaskopoulos (2000) 提出 GIS 可在進行交通模擬 (Traffic Simulation)、交通指派 (Traffic Assignment)、及路徑演算法 (Routing Algorithms) 等工作時作為良好的使用者界面。White et al. (2000) 將使用者地理位置與地圖比對的結果透過 GIS 顯示在個人導航輔助系統 (Personal Navigation Assistants)。

八十年代末期，很多運輸部門正面臨著如何以更有效率的方式獲得交通運輸資訊，以做出有效的決策，此時 GIS 應用成功的例子在國外時有所聞時，再加上運輸路網具有空間資料的特性，使得 GIS 很自然地被運用在運輸上的相關課題，並且被視為運輸界未來有效解決空間資料一致性與正確性問題。

運輸地理資訊系統為結合運輸資料庫與規劃模擬成果，透過地理資訊系統的展示，成為一最佳運輸系統的規劃工具，幫助運輸模擬成果能適當的轉換成資料庫型態，並以 GIS 之圖形方式顯示，將電子地圖結合運輸屬性表達，提昇運輸規劃分析時之效率與資料應用之準確需求，供決策者參考運用。

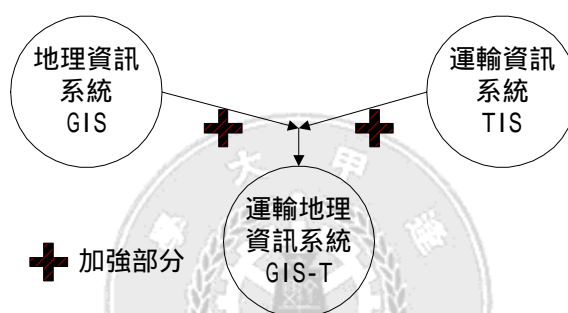


圖 2.3 GIS-T 架構

資料來源：王晉元 (1992)

GIS-T 是以 GIS 為基礎，協助做出有效運輸決策的工具，因此是一套加強的 TIS（運輸資訊系統）與加強的 GIS 的結合，如圖 2.3 所示，其中所謂「加強」則是指為了調和兩系統間在資料庫及分析工具方面的差異而需作適度修改的必要。

2.1.3 分散式系統失敗案例：LASCAD 醫療資訊系統

LASCAD (London Ambulance Service's Computer-Aided Dispatch System) 系統為 1992 年於英國所建立。英國的 LAS (London Ambulance Service) 中心負責運送病患至區內 80 家醫院，平均一年處理 50 萬件意外與緊急事件，運送病人多達 130 萬個旅次。提供這些服務包含有 2700 名工作人員與超過 750 輛的救護車，為全世界規模最大的

救護車服務中心。LASCAD 系統的建立始於 1990 年，由於舊系統無法負荷當時的需求，因此 LAS 著手建立 LASCAD 系統，並於 1992 年完成初步系統建置，並在接下來的幾個月內進行系統測試，在測試過程中，並未進行系統整合測試。約 9 個月後，LASCAD 系統終因無法負荷過多的緊急事件而癱瘓。歸納 LASCAD 失敗的原因是由於系統開發過程中未使用任何一種可接受的系統開發方法，導致系統無法有效整合；以及在管理階層與工作執行階層的溝通不良，導致系統無法有效運作，最後使 LASCAD 系統在短時間內即停止使用。(Hougham, 1996)

2.2 分散式架構

隨著智慧型運輸系統的發展，電腦架構的考慮與應用才逐漸受到相當程度的重視。然而，重點仍然在程式執行效率的改善，採用的方式也大多以平行電腦的應用為主，如 Chang et al.(1994) 於 CM-2 (the connection machine) 所發展的即時性交通模擬模式，模式中利用 C* 的語言來控制 CM-2 中上千個處理器，可以平行化處理大規模路網模擬的問題。Habbal et al. (1994) 也以 CM-2 發展最短路徑的平行化演算法。Mahmassani et al. (1994) 在 CRAY-YMP 上發展 DYNASMART，CRAY-YMP 提供 8 個中央運算單元，並提供 Multitasking 的功能來發展平行化運算模式，但因處理器之間訊息傳遞的 overhead 過高，仍只採用 autotasking 與 microtasking 的方式來加速運算的執行。

但在智慧型運輸系統下另一重要的課題為：如何在即時性交通分析、管理與控制下處理在不同單位、不同電腦系統、不同作業環境下的交通相關資料。在過去十年中，網路的應用已成為資訊科學中最受重視的一環，不僅僅打破使用者之間的空間障礙，也加速資訊的傳遞。本研究希望利用網路的特性建立分散式的模擬指派模式，共用與分享散佈在不同電腦系統的資源，因此交通控制中心可根據最新的資料進行路網車流的模擬，並將結果分享給其他相關的單位，進而規劃適當的交控策略。

一般在分散式的環境下，程式運算的負擔可分散於不同地方的電

腦，以加速執行的效率，或經由分散式的電腦來減輕單一電腦系統的負擔。分散式模式主要以主從 (Client-Server) 架構為主，雖然到目前為止，仍然沒有完整的定義，但一般而言，主從架構需能提供不同機器間之共同合作，用戶端 (Client) 可要求伺服器端 (Server) 的服務，伺服器端也必須能在同一時間內服務不同的用戶端 (Orfali et al., 1996)。Client/Server 也是一種模組化程式的邏輯延伸，這種邏輯延伸使用戶端和伺服器端必須為它們所需的功能搭配適合的軟、硬體。

2.2.1 平行與分散式架構

針對分散式運算環境已有相關系統的發展 (蔡淑蓉, 1996; Orfali and Harkey, 1998), 可提供在工作站群(workstation cluster)上運算的環境，如 PVM(Parallel Virtual Machine)、DCE (Distributed Computing Environment) 等。PVM 由美國田納西大學所發展，軟體中包括程式庫 (subroutine library) 和精靈(daemon) 兩部分，提供在其他機器執行某指定程式的呼叫、收/送訊息給機器中執行的工作 (task)、暫停或中止工作等。PVM 中的機器可以是不同機型的，只要是可以登錄使用者，即可加入 PVM 計算群中。支援的程式語言包括 FORTRAN 和 C。

OSF (Open Software Foundation) 開發的 DCE 包含 RPC (Remote Procedure Call), IDL (Interface Definition Language), Cell and Global Directory Services (CDS and GDS), Security Service, DCE Threads, Distributed Time Service (DTS) 和 Distributed File Services (DFS) 等，並有管理工具來管理以上各部分。DCE 的 RPC 可以轉換不同機器的資料格式，並隱藏機器軟體之間的差異性。

除了上述系統外，所有主要電腦廠商及使用者都在使用物件導向工具和應用程式，若以物件導向來發展分散式計算，其優點幾乎和物件導向本身的優點是一樣的，如封包性 (encapsulation) 和再使用性 (reusability)；而且在異質環境下有一致的介面，不需知道該服務是如何執行 (implementation)，客戶端只需根據伺服器端的服務類別 (class) 的方法來發出要求。

2.2.2 多層分散式架構 (n-Tier Distributed Architecture)

傳統的 Client/Server 架構為二層分散式架構，一層為客戶端，另一層為伺服器；此架構的缺點在於缺乏彈性，無法有效的使用各種資源。將物件的觀念加入分散式架構後，即發展出多層分散式架構。在多層分散式架構中，每一層都是為物件的集合，最簡單的架構：第一層為多個使用者集合的客戶端；第二層為應用程式伺服器 (Applications Server)；第三層為後端資料庫或交易監控系統 (TP Monitor)。多層式架構的優點在於利用物件的觀念大幅增加系統開發的彈性，並有效利用有限的軟硬體資源。現有分散式架構種類繁多，在此列舉較常見的三種：CORBA、DCOM、及 EJB。

2.2.2.1 Common Object Request Broker Architecture, CORBA

目前分散式物件計算技術的標準已由 OMG (Object Management Group) 所訂定，其相關標準正陸續增訂中。OMG 是一個非營利性組織，其目的在於推廣物件軟體標準化，致力於加強軟體的可攜性 (portability)、再利用性、及互通性 (interoperability)，並制定物件管理規格以提供程式發展的共同架構，得以經由各種主要硬體平台及作業系統發展異質計算環境；物件管理也使得程式發展較快、較易維護、再使用性更高。

根據 OMG 之定義，ORB 為一可提供分散式環境上各個物件透明化 (transparent) 的請求服務與回應接收功能的應用程式建構工具。如圖 2.4，OMA (Object Management Architecture) 是 OMG 制定的核心規格，OMA 的四個元件分別為：

- ORB：ORB 是一個仲介軟體 (middleware) 用來確立主從物件主從的關係。ORB 可使客戶端的程式能夠透明地呼叫位於伺服器端的物件，這個物件可以是位於遠方的物件，也可以是同一部機器的物件。ORB 接收請求，並且負責找到一個執行這個請求的物件、

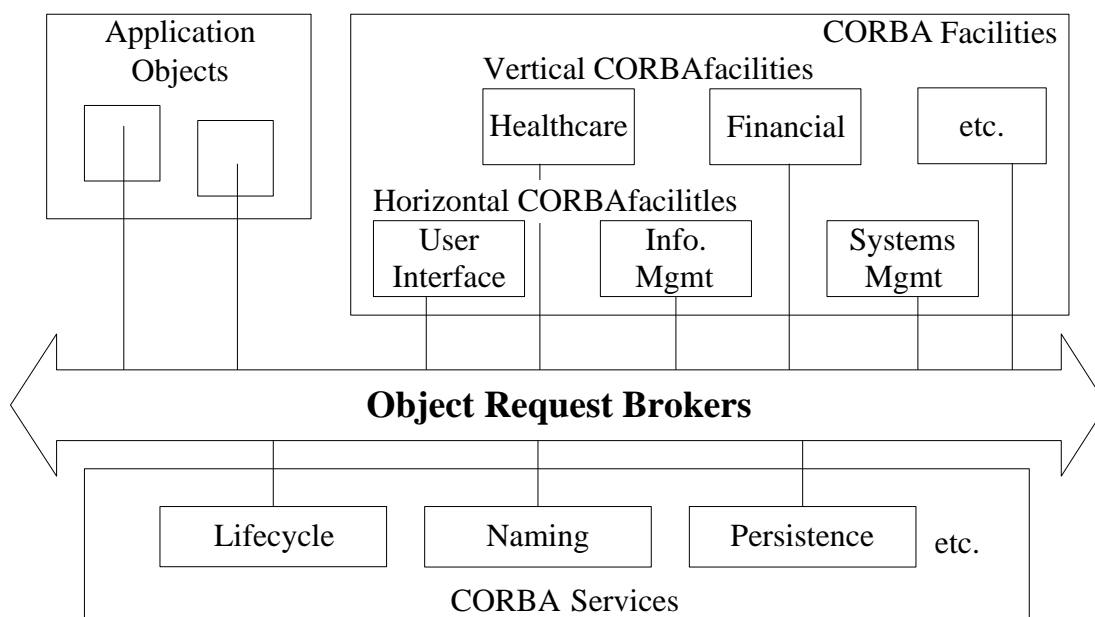


圖 2.4 物件管理架構 OMA (Object Management Architecture)

資料來源：Siegel (1996)

傳給執行參數、呼叫指定的方法、和傳回結果等工作，客戶端可以不知道這個物件的位置、使用的程式語言、作業系統、或是任何和這個物件的界面 (interface) 無關的部份。這樣的作法，使得 ORB 提供在異質性 (heterogeneous) 分散式環境中的應用程式的互通性，並可以連接多個物件系統。

- CORBA Services：根據 OMG 的定義，CORBA 服務為支援物件使用及執行之功能。為了在多部機器之間提供較完美的物件環境，必須有一個機制來經由物件的名稱存取物件、在遠端產生或是修改物件、隨時維護物件的狀態和其他一般的服務。
- CORBA facilities：OMG 定義一些應用程式間共用的界面和統一的語意，使得符合 OMA 的應用程式易於建立。
- 應用軟體物件 (Application Objects)：在 OMA 架構的最上層是應用軟體物件，和下層基本物件不同的是它們直接支援一些商業的活動，它們可以直接被使用者使用，而不是被系統發展人員使用，它最終的目的是允許使用者在一般的應用軟體物件之外自行組合自己的解答。

2.2.2.2 Distributed Component Object Model, DCOM

微軟公司 (Microsoft) 在 Windows 平台所開發的分散式物件模式 DCOM (Distributing Common Object Management) 分散式通用物件模式，其架構如圖 2.5。DCOM 呼叫位在用戶端的 object proxy，該 proxy 便透過網路向伺服器端的 stub 傳遞物件的呼叫，伺服器在判斷物件的呼叫之後便將所需的分散式物件傳回至用戶端的 object proxy，以供應用程式使用。

DCOM 雖然可跨平台使用，但其方便性仍不及 CORBA；而 CORBA 標準可適用於多種作業系統，並且可結合 DCOM，使 CORBA 能在 Windows 作業系統做更進階的使用。CORBA 早在 1990 年便存在了，於 1992 年始應用於商業用途；而 DCOM 卻在 1996 年才有測試版出現，由於開發時間早晚的差異，使得 CORBA 較 DCOM 更為完整。

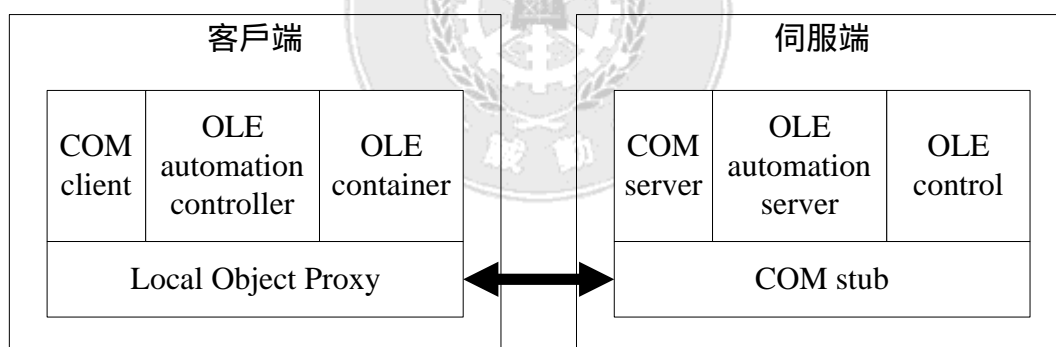


圖 2.5 DCOM 架構

2.2.2.3 Enterprise JavaBeans, EJB

1997 年底，Sun Microsystem 提出 EJB 的架構，該架構之定義為一種元件的架構，用來發展分散式元件導向的企業應用系統，EJB 具有可調性、交易正確性、及多使用者安全性等特點。一般的分散式物件架構為 CORBA 與 DCOM 等標準，而 EJB 則是將分散式物件架構加上元件交易監控伺服器 (Component Transaction Monitor, CTM) 的

元件模型的標準，與分散式物件架構最大差異在於 EJB 能有效的管理和監控系統中各元件的交易狀況。EJB 架構中主要包含三個觀念，分散式物件、伺服器端元件、及元件交易監控伺服器，其中分散式物件技術為 EJB 架構的基礎；伺服器端元件表示 EJB 架構符合元件模型的特性，而其元件為伺服器端提供之服務；元件交易伺服器為 EJB 架構之核心，可視為 TP Monitor (Transactions Processing Monitor) 與 ORB 之綜合體，主要是提供管理監控大量交易的功能，也是 EJB 與一般分散式物件架構最大的差異。

圖 2.6 為 EJB 的契約關係 (Contracts)，以客戶端的觀點來看，Enterprise bean 被容器 (Container) 所包裹，就如同一般元件一樣，其連結方式亦較有彈性。Enterprise bean 與容器之間亦有一層契約，該契約能夠讓容器將 Enterprise bean 以元件的方式跟客戶端相連結。

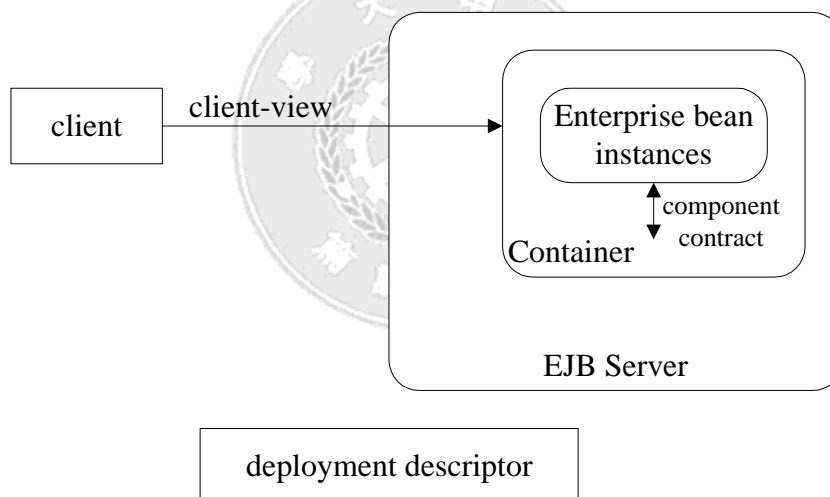


圖 2.6 Enterprise JavaBeans 的契約關係

資料來源：Sun Microsystems (2000)

2.2.3 分散式架構應用實例

美國 NEMA (National Electrical Manufacture Association) 組織為整合交通的通訊技術提出一共同的標準 NTCIP (National Transport Communication for ITS Protocol)，透過此一標準可讓各項軟硬體有共同的標準規格，大幅提高使用彈性與相容性；國內同時遵循 NTCIP

的標準建立一套 NTCIP-like 做為國內的共通標準,而其介面標準化則是採用 CORBA 與為 Profile 標準之一 (張堂賢、韋理為, 2001)。陳俊良等在 1999 年提出一多階層式架構,以 CORBA 技術發展一交通管理中心資訊,該架構具有較高的系統維護與擴充能力,並可加速建立國內 ITS 資訊共享之基礎平台 (陳俊良、蕭偉政、林俊彥, 1999; 何金駒、林富泰, 2001)。

許多以 CORBA 為基礎,包含了以下領域的的計畫及應用軟體已被成功發展,諸如:航空太空、國防、金融/銀行、化學/石油化學、顧問公司、教育單位、電子商務、政府單位、醫療照顧/保險、人力資源、生產製造業、出版業/多媒體業、不動產、研究、零售業、軟體/硬體公司、電信業、交通運輸/旅遊、公用事業。在本節,主要將探討交通運輸相關的計畫,並將相關應用軟體作一重點整理,如下頁表 2.1。

2.3 K 條最短路徑問題

最短路徑 (Shortest Path) 問題為一般路網問題求解的基礎,主要的求解目標是兩點之間的最少成本的路徑。但在實際應用上,單一最短路徑並無法滿足系統與旅行者的需求,因此有了計算多條最短路徑的想法,產生次佳路徑,並依照成本大小排列,讓使用者有較多的選擇,並可依不同的需要調整 k 值的大小。

K 條最短路徑的演算法大致上可分為直接求解與間接求解兩種,直接求解即是直接對路網中的兩個點進行 k 條最短路徑的計算;間接求解是先計算出最短路徑,再從最短路徑的結果計算第 2 至 k 條最短路徑。由於近年來電腦科技的發展,平行 (parallel) 和分散式 (distributed) 的演算法也逐漸受到重視,依照平行或分散式處理架構建立之演算法可大幅提升運算的效率,透過效率的提升可使該演算法所應用的範圍更廣。

表 2.1 CORBA 架構於交通運輸之應用

主要的組織	使用工具	問題	效益
美國航空 AA (American Airlines)	Iona Technology Orbix Broadvision One-to-One	在 SABRE 訂票系統及 HTTP 網路伺服器間，設計一個以 CORBA 為基礎的網站，並以可重複操作的中介層為工具，廣泛的使用一對一架構，以提供個人化的內容服務	減少使用者接觸介面及複雜性
美國航空 AA (American Airlines)	1. Iona Orbix CORBA 2. Windows NT 3. TCP/IP	航空貨物管理：一個即時決策支援系統，藉由提供即時的航班及貨物警報、貨物清單的資訊及全美七個航空貨物轉運站的資源分配，以幫助貨物處理部門運作。	1. 改善作業表現 2. 增進航空貨運運轉效率
Ascom Autelca	Ascom Fare Collection CORBA	運輸收費系統	一個完整的分散式系統，以元件為組成基礎
Eurocontrol Experimental	1. Iona's Orbix 2. CORBA 3. Oasis	歐洲航空管制系統 (ATC)	在不同的航空管制系統間提供完整的相互可操作性
聯邦快遞 FEDEX	Smalltalk	一個新的高容量分散式遞送追蹤應用軟體	1. 從中央辦公室到地方追蹤貨物情形 2. 可以進行即時的出貨分析 3. 改善全體系統的操作表現
法國里昂大眾運輸當局	APPLI-BUS ORBIX CORBA for ATM switches Windows NT	交通狀況的監控與管理控制系統	1. 提供照相機及錄影帶的操作彈性 2. 分散資訊節點的透明化
新加坡港務局	APPLI-BUS	藉由終端機提供貨物處理過程的自動化操作	提供彈性管理界面
美國汽車協會	Inprise's Visi-Broker	旅遊預先訂位系統	增強使用者界面

資料來源：<http://www.corba.org/>

2.3.1 問題定義

如同其他的路徑產生問題，K 條最短路徑除了基本定義外，依照不同的問題類型亦有不同的定義。為能瞭解不同的 K 條最短路徑演算法所適用的問題類型，將針對各問題定義作一詳細說明。本節將介紹

基本定義與不同類型問題的定義，並比較其間的差異。

KSP 問題為一般最短路徑 (Shortest Path, SP) 問題的延伸，一般 SP 問題為求解路網中任意兩點之間的最短路徑；而 KSP 問題除求解最短路徑外，另需求出多條次佳路徑。KSP 問題之基本定義為「求解路網中所給定的兩點之間所有的路徑，並將所有的路徑依照成本由小到大排序，從 1 排到 k ，其中 $k > 1$ ， k 值可依需要改變大小 (Shier, 1979)」。

本節藉由一 K 條最短路徑之簡例說明其基本觀念與解法。假設有一虛擬路網，其中包含 5 個節點與 8 條路段，路段的成本標示在線上。最簡單的 K 條最短路徑是找出路網中給定兩點間的第 1 條、第 2 條、第 K 條最短路徑，本例中是要找出點 S 與點 E 之間的 K 條最短路徑。所有可能路徑及其通過的節點與路徑成本標示於圖 2.7。

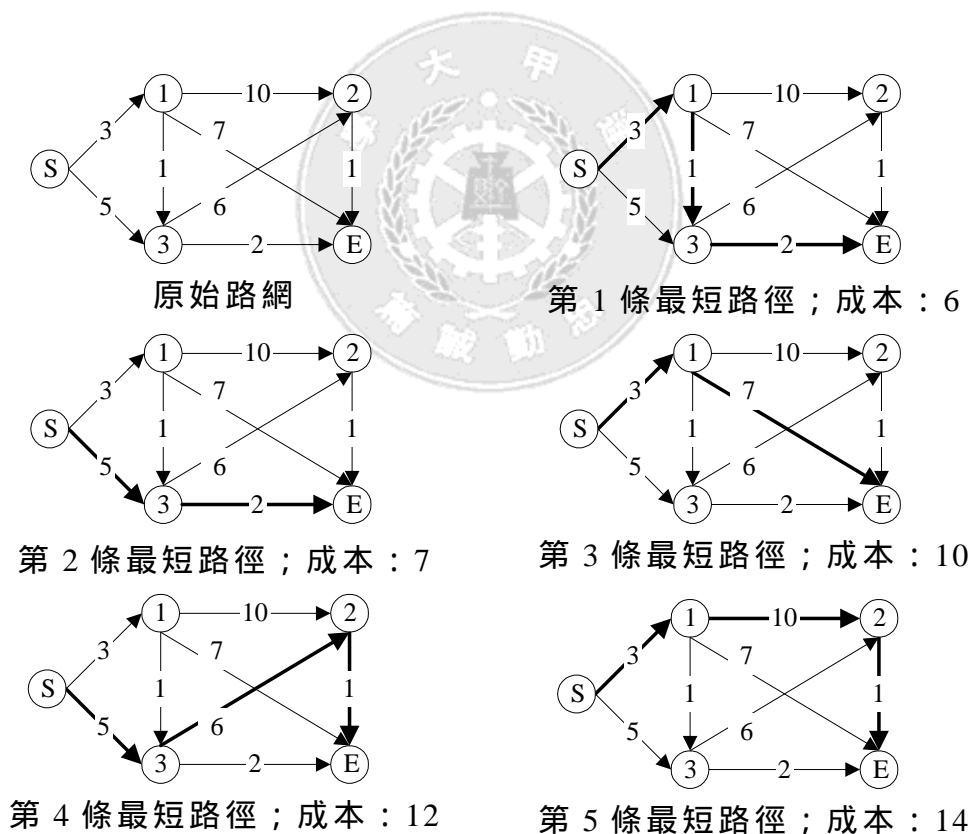


圖 2.7 K 條最短路徑範例及其解

本例以窮舉法列出點 S 到點 E 之間所有可能的路徑並計算其成本，在結果中，路徑 $S-1-3-E$ 為點 S 到點 E 的最短路徑，路徑 $S-3-E$

為第 2 條最短路徑，其餘依此類推，本例顯示出 K 條最短路徑的基本觀念，並詳細列出其結果。路徑 S-1-3-E 為最短路徑，在一般的最佳化路徑問題中應用相當廣泛，當最短路徑因道路成本的不確定性而轉變為無效或不正確之路徑時，其餘路徑均為可能替代路徑；在本例中，路徑 S-3-E 為第 2 選擇方案。

2.3.2 K 條最短路徑演算法

KSP 演算法最早被提出是由 Hoffman 在 1959 年所提出，最常見的演算法為 Yen 在 1971 年所發表的 K 條最短路徑演算法，大多數與 KSP 相關的演算法及其應用都是以 Yen 的演算法為基礎，(Hadjiconstantinou, 1999)。

2.3.2.1 序列式 K 條最短路徑演算法

本節將介紹研究中作為 KSP 計算元件的演算法，DoubleSweep 演算法，該演算法於 1976 年由 Shier 所提出，為標籤修正法 (Label correcting method) 的一種。DoubleSweep 演算法允許路網中存在負成本的路段，而不允許存在負成本迴圈，其計算複雜度為 $O(kn^3)$ 。以下將說明該演算法之流程。

定義：假設一路網 $G=(N, A, c)$ ，其中 N 為節點的集合、 A 為節線的集合、 c 為節線成本的集合。令 C 為路網中點對點的成本矩陣 $C=c_{ij}$ ，再將 C 分為兩個矩陣 $L=(l_{ij})$ 為 C 的下三角矩陣及 $U=(u_{ij})$ 為 C 的上三角矩陣。

步驟 1：設定 $v = 0$ ， $d_j^v, \forall j$ 。

步驟 2： $v = v + 1$ ，計算 $d_j^v = [d_j^{v-1} \otimes u_{.j}] \oplus d_j^{v-1}$ for $j = 1, 2, \dots, n-1, n$

步驟 3：若 $d_j^v = d_j^{v-1}, \forall j$ ，到步驟 6。

步驟 4： $v = v + 1$ ，計算 $d_j^v = [d_j^{v-1} \otimes l_{.j}] \oplus d_j^{v-1}$ for $j = n, n-1, n-2, \dots, 1$

步驟 5：若 $d_j^v = d_j^{v-1}, \forall j$ ，到步驟 6；否則到步驟 2。

步驟 6：停止。

除了 Shier 的演算法之外，尚有許多 K 條最短路徑演算法被提出。以下列出其他 K 條最短路徑演算法：

- a. Hoffman and Pavley's algorithm (Hoffman and Pavley, 1959)
- b. Yen's algorithm (Yen, 1971)
- c. Simplified Double-Sweep Method (Rink, Rodin, and Sundarapandian, 2000)
- d. Dreyfus' algorithm (Dreyfus, 1969)
- e. Martins' Algorithm (Martins, 1984)

2.3.2.2 平行與分散式 K 條最短路徑演算法

許多路網問題因為求解的規模限制無法對大規模問題進行求解，而無法應用在實務上，為能增加演算法的求解規模，有許多的研究將原本演算法做修改，以平行計算的方式提升計算的效率。卓訓榮 (民國 82) 討論程式語言不同的平行最佳化，發現向量化的平行處理方式較純量化及平行化處理為佳。本節將以 Gerriero 所提出之平行 K 條最短路徑演算法為例，說明平行與一般 KSP 演算法的差異。

Guerriero (2000) 提出一平行 K 條最短路徑演算法，其想法為將原本 Label- correcting 方法修改為平行處理之演算法。該演算法雖為平行處理，但其處理過程為非同步 (asynchronous) 平行處理，亦可視為分散式演算式。其演算法如下：

步驟 1. 自 V^ξ 移除點 i , $\forall j \in N : (i, j) \in A, j \neq s$, 找出最小標記 $\bar{k} \leq K$:

$$d_j^{\bar{k}} > d_i^\xi + p_{ij},$$

$$d_i^\xi + p_{ij} \neq d_j^\psi, \quad \psi = \bar{k} - 1, \bar{k} - 2, \dots, 1.$$

步驟 2. 令

$$d_j^{\bar{k}} \leftarrow d_j^{\bar{k}-1}, \forall \bar{k} = \tilde{k}, \tilde{k} - 1, \dots, \bar{k} + 1,$$

$$d_j^{\bar{k}} \leftarrow d_i^\xi + p_{ij},$$

當 \tilde{k} 為 $\bar{k} \leq \tilde{k}$ 情況下的最大值且 $d_j^{\tilde{k}} < \infty$ 。將點 j 加入 $V^{\bar{k}}, V^{\bar{k}+1}, \dots, V^{\tilde{k}+1}$ 。

步驟 3. 若 $V^\xi = \emptyset$, 停止；否則回到步驟 1。

該演算法將步驟 1 中之 V^k 集合的計算過程修改為平行處理方式，亦即點 i 的所有相鄰點以非同步平行方式處理。透過平行化處理可明顯加快求解的速度，在求解大規模路網問題時，平行處理可加速 3 至 5 倍。

2.3.3 K 條最短路徑演算法之應用

在前一小節已提到 K 條最短路徑在 ITS 中有廣泛的應用，本小節將作更詳細說明。KSP 問題的觀念主要是提供除最佳解以外的替代方案，因此在實務上有相當廣泛的應用，許多的問題都是採用 KSP 的解題方式或觀念，包括基本的替代路徑選擇、替選方案的排序...等相關應用。本節將針對不同的 KSP 應用作一簡單說明。

KSP 的基本觀念即是求解最短路徑的替代路徑，這也是多數路網問題所需要的功能之一，因此有許多相關的研究針對此一課題進行研究。許多實務上的研究，原本的問題可以路網的形式呈現，以求解路網問題的方式來解原始問題。

Bala (1995) 將線性光波網路 (Linear lightwave network) 中的路徑選擇方式以 KSP 等演算法進行求解；Carrier (1997) 在大型電信網路的可靠度評估中，以 KSP 演算法計算光纖網路中多條重新路由 (Rerouting) 的路徑；Tang and Abdel-Malek (1996) 以 KSP 演算法求解製造過程中生產點平面配置問題 (Cellular Layout Problem, CLP)；Akgün et al. (2000) 以 KSP 的觀念求解路網中不相似路徑 (dissimilar paths)。Chen and Feng (2000) 將 KSP 演算法應用在隨機/動態用路人最佳路徑選擇問題 (stochastic/dynamic user-optimal route choice problem)；Solanki et al. (1998) 應用在求解大規模公路路網設計的近似最佳解；Fu and Rilett (1998) 以 KSP 演算法求解動態隨機最短路徑問題 (Dynamic and Stochastic Shortest Path Problem, DSSPP)。Koncz et al. (1996) 求解大眾運輸路網中多重最佳路徑，提供旅行者路徑選擇方案，以增加旅行者選擇的彈性。在路線導引上的應用主要是提供旅行者多條路徑選擇，當遇到路段上因車流變化或意外發生而導致原路徑無法使用時，旅行者可選擇次佳替代路徑。Lee et al. (1998)、

Narasimhan et.al (1999) 以 KSP 演算法計算即時交通控制系統 (Real-Time Traffic Control) 中自動車輛導引系統 (Multiple Automated Guided Vehicle Systems, MAGVS) 所需的路徑表。

2.4 系統效能評估

建立分散式系統過程中，效能 (Performance)、可靠度 (Reliability)、和成本 (Cost) 三者無法兼顧，因此需藉由績效評估的方式來決定三者之間的平衡點；透過績效評估結果可發現系統的效能瓶頸，並針對問題加以改善。績效測量的主要目標是讓管理者決定系統在現在或未來所需達成的績效目標，為此需求得兩種不同的績效值，一種情況是在已知的負載下，系統真正的績效值；另一種情況是估計在有潛藏的負載狀況下系統績效值，並估計提供服務品質的風險 (Langsford, 1993)。在進行績效測量之前，必須先選擇績效指標，適當的績效指標可提供更精確可靠的測試結果。

根據 Langsford (1993) 與 Hirano et al. (1998) 之研究內容可將分散式系統績效評估項目分為四大類：服務水準、通訊時間、執行時間、及錯誤率；服務水準代表系統整體回應時間；通訊時間表示資料在兩機器間之傳輸時間，由於分散式系統的傳輸距離較長，通訊時間對系統有很大的影響；執行時間指當客戶端送出請求或伺服器接收並處理請求的時間，複雜的計算元件需要較長的執行時間，對系統效能也有相當的影響；錯誤率，顧名思義是指系統傳回或執行錯誤結果的比率，由於分散式系統較為複雜，錯誤率較一般電腦系統更高，為避免失去系統原有用途，必須將錯誤率控制在可接受範圍內。詳細績效指標的訂定由上述四大項衍生而來，依照不同的架構亦有不同的績效指標。

Urban 將 CORBA 應用在工程設計，並針對系統進行效能評估。作者選擇檔案、封裝 (Wrappers)、及使用者存取 (Multi-user access) 作為評估項目；檔案傳輸分為不同大小的檔案，以瞭解檔案大小對傳輸效能的影響；封裝是探討客戶端存取伺服端的過程是否有經過封裝對效能的影響；使用者存取是研究單一使用者與多使用者進行存取動

作之效能差異 (Urban, 1999)。Gokale 等人探討在高速網路中, CORBA 協定的各項通訊連結績效指標 (Gokale et al. 1998)。

效能最佳化是相當複雜的步驟, 前節已回顧各種績效指標, 而最佳化的目的在於將現有的系統效能, 經過調校後, 達到整體效能最佳化。最佳化目標可分為三種: 通訊系統最佳化、執行系統最佳化、及整體效能最佳化; 設計者可根據需求來決定最佳化的種類。以下分別說明各個最佳化目標: (Langsford, 1993)

1. 通訊系統最佳化

一般而言, 通訊系統是所有分散式系統的效能瓶頸, 其最佳化的目標為減少延滯與增加傳輸量, 調整方式可選擇參數修改或選擇其他通訊協定以達到最佳化的目標。

2. 執行系統最佳化

執行系統的效能通常是受到軟體元件配置的影響, 調整的方式可藉由適當的資源分配來達到。而軟體元件的配置通常是由記憶體的使用、處理時間、處理優先次序等影響, 當系統為靜態系統時, 則調整的過程必須在系統正式啟動前調整完畢; 若為動態系統, 則可在系統啟動後進行動態調整, 如動態負載均衡方法 (Dynamic Load Balance)。

3. 整體效能最佳化

主要是探討系統在單位時間內所能提供的服務量, 可藉由增加個別系統元件或容錯 (Fault Tolerance) 系統的可靠度進行最佳化的調整。系統元件可藉由穩定環境因素 (如溫度、濕度) 或更換系統元件來提升可靠度; 而容錯系統可透過備份資源的提供來增加可靠度。

2.5 小結

本章就 ITS 架構與應用、分散式系統架構、K 條最短路徑演算法、與系統效能評估等四項重點進行相關文獻回顧。ITS 目前的發展方向朝著分散式系統發展與無線通訊的使用, 透過這兩種技術可大幅提昇

系統功能彈性。地理資訊系統亦是 ITS 應用趨勢之一，由於地理資訊系統提供更友善的地圖顯示方式，相關系統開發亦較傳統軟體更為方便。ITS 架構中已包含分散式架構的觀念，因此相關系統開發可朝向分散式架構；目前的分散式架構大多以物件為基礎觀念，增加系統開發的方便性與擴充能力。

K 條最短路徑演算法是動態路網分析的求解基礎，而 K 條最短路徑演算法的求解過程複雜，在計算過程會耗費大量時間，系統需針對避免增加計算處理時間。因此，數值實驗中以兩種 KSP 主要應用進行測試，進而瞭解系統績效之影響因素。

系統效能評估可檢測系統效能瓶頸並且提供系統設計的參考依據，管理者必須針對需求挑選適當的效能評估方式。為能有效進行系統評估，必須選取與系統有關之績效指標，進而藉由績效指標對不同系統設計進行評估。本研究以過去的研究為基礎，先行定義所需要的績效指標，再予以搭配研究中之系統架構定義各項績效指標的內容。



第三章 CORBA-Based 分散式架構

ITS 架構中包含一般使用者、用路人、控制中心等單元；透過各種通訊技術的連結，各個單元均可彼此連結及傳遞訊息，儼然形成一個大型分散式架構。而在 ITS 架構中，相關系統的建置所花費的成本都相當高，更由於交通與民眾生活息息相關，系統穩定性必須具備相當高的標準方能被接受。因此，一個應用系統的建置必須經過有系統的分析與測試方能正式上線運作。研究中將探討系統的目標需求、計算效率、傳輸效能等績效指標，希望能藉由績效指標評估的結果獲得系統建置的參考依據。

本章將進一步說明分散式架構與 ITS 之關係，以及作為數值實驗之範例系統和架構說明。3.1 節介紹分散式架構於 ITS 之應用，並結合 CORBA 協定之通訊元件成為 CORBA-based 分散式架構。3.2 節探討目前 ATIS 與 ATMS 系統的應用方向與一般架構。3.3 節以兩個系統實例分別代表 ATIS 與 ATMS 的應用，實例將以兩個 CORBA-based 分散式架構的應用，探討本研究所建立之各種分散式架構。3.4 節說明效能評估的方式及績效指標的選取。

3.1 分散式架構於 ITS 之應用

交通環境中包含旅行者、車輛、路側設施、及控制中心等四項元件，在 ITS 架構下，彼此間藉由各種有線或無線傳輸方式進行溝通。為能達到 ITS 的目標，交通資訊扮演相當重要的角色，一般用路人需要交通資訊作為各種交通行為決策的參考；研究中心需要交通資訊作更深入的研究及產生其他有價值的資訊；政府單位需要交通資訊作為交通相關決策的考量依據。而在此一複雜的分散式環境中，交通資訊的傳遞與分析過程往往會因為資料量過於龐大或處理過程繁複，而導

致效率低落以致於無法達到原訂的系統目標。

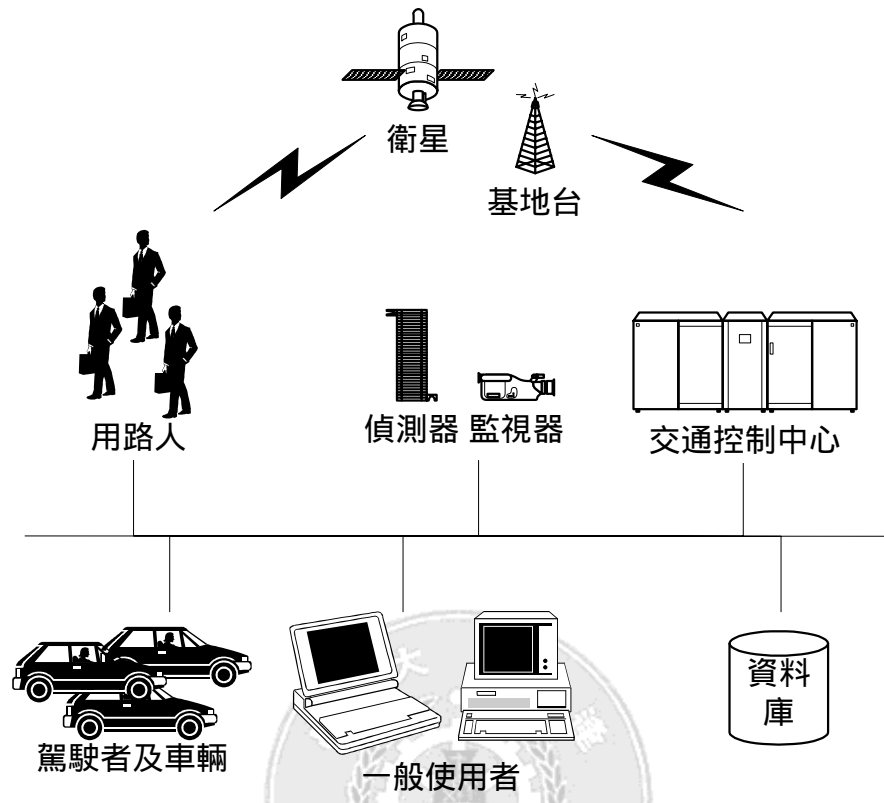


圖 3.1 ITS 架構示意圖

將上述 ITS 架構予以簡化得圖 3.2，該圖為分散式架構於 ITS 之應用，架構中四個元件透過中央的通訊元件進行連結。此架構之優點為各項元件不需個別建立連結至其他所有元件，僅需連結至中央的通訊元件即可與其他元件進行通訊。此外，各元件所採用之系統亦不盡相同，包含電腦作業平台、程式語言、及資料庫建置方式等，為能連結各種不同系統，需採用易擴充、維護的通訊元件。為建立此一架構，本研究採用 CORBA 協定作為中央通訊元件，透過 CORBA 的分散式處理及跨平台、與程式語言無關的特性，各元件可在作小幅度的修改後，即可與其他元件進行連結，同時也可透過通訊元件與其他 ITS 系統建立連結。

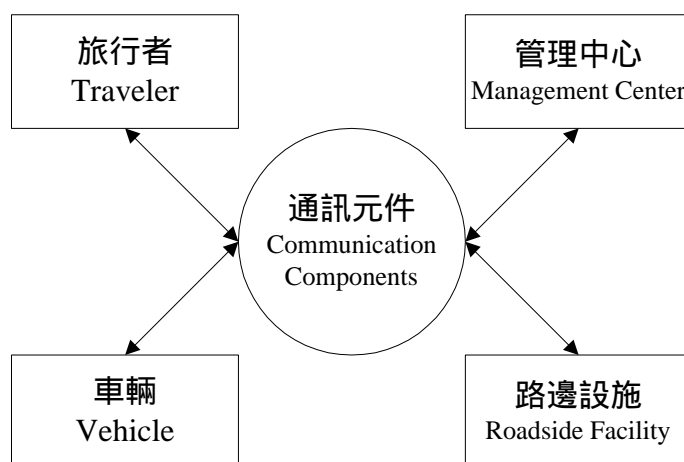


圖 3.2 分散式架構於 ITS 之應用

以 CORBA 建立的系統可藉由協定中的 ORB 與其他 ITS 子系統建立連結，ORB 可將客戶端元件的請求 (Request) 傳送至適當的伺服器端，並將結果傳回。透過此一連結方式，各個 ITS 系統間都可進行連結，以共享資料與分析工具，將可大幅增加系統擴充的彈性；此一優點將可減少系統開發時間，並以最大效益使用各項資源，將可加速建置整體 ITS 系統。

3.2 CORBA-Based 先進資訊系統架構

本節將針對 ATIS 中資訊的管理與提供來探討 K 條最短路徑的應用與未來分散式系統架構建立的方式。在文獻回顧中已提及動態路網分析應用在 ITS 中的應用相當廣泛，而 K 條最短路徑演算法是動態路網分析的主要求解基礎之一。因此，本節將 KSP 演算法應用於 ATIS 中的路徑導引與路線查詢，並建立其分散式架構，以作為效能評估的基礎。

3.2.1 先進旅行者資訊系統：路徑導引與路線查詢

圖 3.3 為路徑導引與路徑查詢之分散式架構，主要包含三個部

分，第一部分為使用者端，包含使用者與其使用之路線產生系統；第二部分為 ORB，處理使用者之請求送出與應用程式伺服器端之結果回傳；第三部分為應用程式伺服器端，此部分又包含三大元件，應用程式、計算元件、與路網資料。在早期的架構中，應用程式伺服器端之三大元件包含在同一部電腦裡；而在目前的分散式架構裡，該元件可分散於不同位置，以提高資源使用效率與減少資料維護負擔。架構主要可分為兩種，最主要的差異在於計算元件的位置（圖 3.3 中之虛線物件）。無論是應用程式伺服器端或使用者之路線產生系統皆有能力執行計算元件，但效率會有所不同。架構的主要流程為使用者送出請求，透過路線產生系統連結至 ORB，而 ORB 自動尋找適當之伺服器，並送出請求至應用程式伺服器端；應用程式再根據請求尋找計算路徑時所需的路網資料，將資料傳給計算元件並進行路徑運算，最後將結果傳回予使用者。

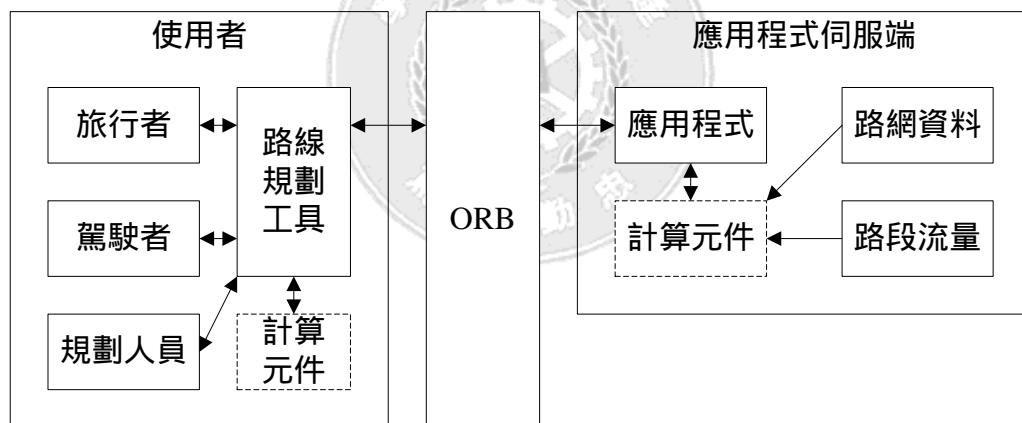


圖 3.3 路徑導引與路線查詢功能分散式架構

資料來源：Hu (2001)

由於路徑導引與路線查詢的應用廣泛，實際建立分散式架構的方案選擇亦較多。以通訊方式為例，除旅行中路線規劃必須使用無線通訊方式外，其他應用皆可考慮無線或有線通訊方式進行連結，更增加系統開發的彈性，也因此需要更詳細之效率評估來決定適合的系統開發方式。

3.2.2 CORBA-Based 分散式 GIS-T 之應用

本研究以一緊急醫療救護應用系統作為 ATIS 數值實驗範例，該系統以 CORBA-Based 的分散式 GIS-T 架構為主，希望克服動態資訊的傳遞問題，配合 GIS-T 的特性達成人員、車輛、儀器、及醫院的決策參考資料，未來可以結合派遣、調度之觀念成為即時性調度派遣的系統。CORBA-Based 分散式 GIS-T 於緊急醫療救護之資料蒐集包含醫療救護 GIS 與運輸資訊兩部分（架構如圖 3.4）。為能查詢不同醫院間架構於不同軟硬體配置的資料庫，CORBA ORB 伺服器扮演溝通的角色，藉由其跨平台、與跨程式語言的架構，系統可與其他不同軟硬體之電腦進行溝通，達到分散式資料共享的目標。

GIS-T 中基本的運輸屬性資料包括當地主要道路與路網之基本屬性，包括：道路長度、寬度、幾何線型等；與交通流量調查部分，包括：不同時段之主要路口交通流量、主要道路之行駛速率、路邊停車情形、道路服務水準等。

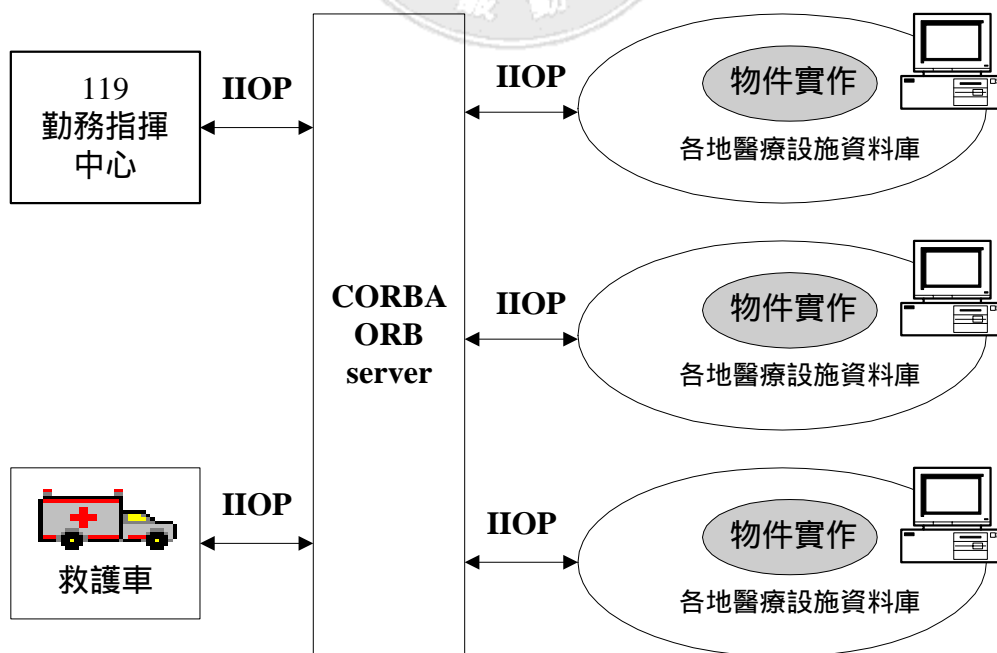


圖 3.4 分散式 EMS 系統架構

在架構中以 GIS 軟體作為處理地理空間圖形之工具，將資料以資料庫軟體建構資料庫，再以一程式查詢介面驅動 GIS 軟體及資料庫，即可由查詢介面提供動態醫療設施資料查詢功能。各元件之間透過 IIOP (Internet Inter-ORB Protocol) 與 ORB 伺服器進行連結。資料庫、GIS 軟體、程式查詢介面間相互獨立，並可因應環境之需要置換合適之軟體，整體架構替代彈性大、限制需求小。未來更可易於結合其他模式庫作為應用發展。

3.3 CORBA-Based 交通管理系統架構

KSP 演算法是動態路網分析的主要求解基礎之一，該演算法於 ATMS 中亦有廣泛的應用，主要可用於路徑規劃相關模式計算，因此本節以 ATMS 中的動態交通模擬指派系統說明 KSP 的應用實例，並以一分散式動態交通模擬指派系統作為數值實驗測試之用。

3.3.1 先進交通管理系統：動態交通模擬指派系統

ATMS 系統主要目的是藉由整合新發展及現存的交通管理及控制系統，以期能在給予各種運具的服務反應出動態的交通狀況。為能提供動態交通狀況，需透過區域性監測設備測得地區性即時交通狀況，以做為動態分析之用。由於所蒐集之即時資訊通常包含許多相關單位，且資料量亦較大，因此 ATMS 系統需能在短時間內處理不同單位的資料。藉由先進通訊與資訊處理技術，可以分散式架構建置 ATMS 系統以達到快速處理不同單位資料的目標。

而在 ATMS 系統中，交通指派為一個重要的方法之一，其演算過程較為繁複。交通指派 (Traffic Assignment) 或稱路網指派 (Network Assignment) 可定義為 (Sheffi, 1985)：根據指派的規則 (或路徑選擇原則)，將起迄旅次分配到一交通路網，此路網包含旅行時間函數。

透過交通指派的應用，可產生路網內各路段上之流量，用於不同之規劃分析。然而，靜態交通指派模式仍有其缺點，包含完全忽略依時性的因素、及道路績效函數之不合理。相對的，交通模擬應用數值模擬的方法，結合車流行為之數學模式，以巨觀或微觀的方式來觀察車流在不同假設下之變化。交通模擬模式在智慧型運輸系統的應用上，最主要的問題在於路徑選擇行為無法反應在模擬中，一般模擬都是以路口轉向比值計算車輛移動。一般來說，模擬適用於交通分析與控制策略的研擬；傳統靜態交通指派較適用於長程的規劃。

模擬指派模式係根據路徑選擇原則，將依時性的 OD 依次分配到一交通路網。車輛移動的過程以模擬方式配合交通控制策略進行。車輛所使用的路徑，為駕駛者行為決策下之結果。

3.3.2 CORBA-Based 分散式 DYNASMART

在分散式 DYNASMART 的建構中，因不同的單位所使用的作業平台和程式語言不盡相同，若僅依靠 TCP/IP 通訊協定在各單位間進行資料傳遞，將造成使用上之不易。DYNASMART 雖已可由 TCP/IP 通訊協定與其他單位之電腦進行資料共享，但仍僅限於靜態資料傳輸，無法控制其他電腦上之程式，也未達成真正即時的動態資料擷取功能。本系統提出以 CORBA 協定建構分散式 DYNASMART，此一分散式運算具有下列幾項優點。

(1) 連絡不同電腦上之程式

交通相關資料通常是由不同單位的電腦所產生，各單位依照現實情況隨時更新其資料，CORBA 協定可連結不同電腦上之程式，使分散式系統能達成真正即時的動態資料擷取功能。

(2) 跨平台且相容於各種程式語言

CORBA 協定可相容於各種作業系統及程式語言，可使某一種程

式語言直接控制另一種程式語言，進而減少系統整合所耗費的成本。

CORBA 協定下之分散式 DYNASMART 可透過 ORB (Object Request Broker) 物件代理控制、分配及整合位於不同電腦上之程式運算與資料，並可減少主伺服器之運算負荷。此一架構利用 CORBA 之物件導向及跨平台特性，可於不同作業平台間進行分散式運算 (Distributed Computing)，將所有資料視為物件處理，並進行資料共享與使用。

DYNASMART 採用模擬指派 (Simulation-Assignment) 的方式來指派依時改變的交通需求，利用交通模擬來建立相對應的交通型態，並用以評估交通路網在不同假設下的績效。迄今，DYNASMART 的應用，主要是以描述與評估在 ATIS/ATMS 架構下路網車流可能的變化；考慮的因素包含了交通資訊的產生與供給、交通控制策略、路線導引與駕駛者個體對系統的反應與決策。DYNASMART 包含了車流模式的應用，路徑資訊產生、個體決策行為模式與可能的資訊供給法則。在這個架構下，旅次產生者所決定的出發時間 (Departure Time)，運具 (Model) 與使用路徑 (Route)，構成了一個三維的依時 OD 矩陣。DYNASMART 依時間次序，將私人車輛 (含汽、機車與貨車) 與公車陸續載入交通路網中，並按各車種的特性來移動車輛，直到車輛抵達目的地為止。在車輛移動的過程中，交通資訊的傳輸與使用，亦將針對車輛特性加以區分 (胡大瀛等，1996)。

該系統採用物件運算方式，架構如圖 3.5 所示，分散式 DYNASMART 中假設有三個中心：先進交通管理中心 (ATMS)，交通號誌控制中心，與資訊控制中心。其所負責的工作包括：

- 1、先進交通管理中心：負責交通模擬運算。

運算包括車輛於路段移動中計算，與車輛於路口中的分配與行進。計算過程中視需要向交通號誌控制中心要求即時性號誌時制分配資料或向路徑資訊控制中心要求提供即時性路徑供駕駛者參考。

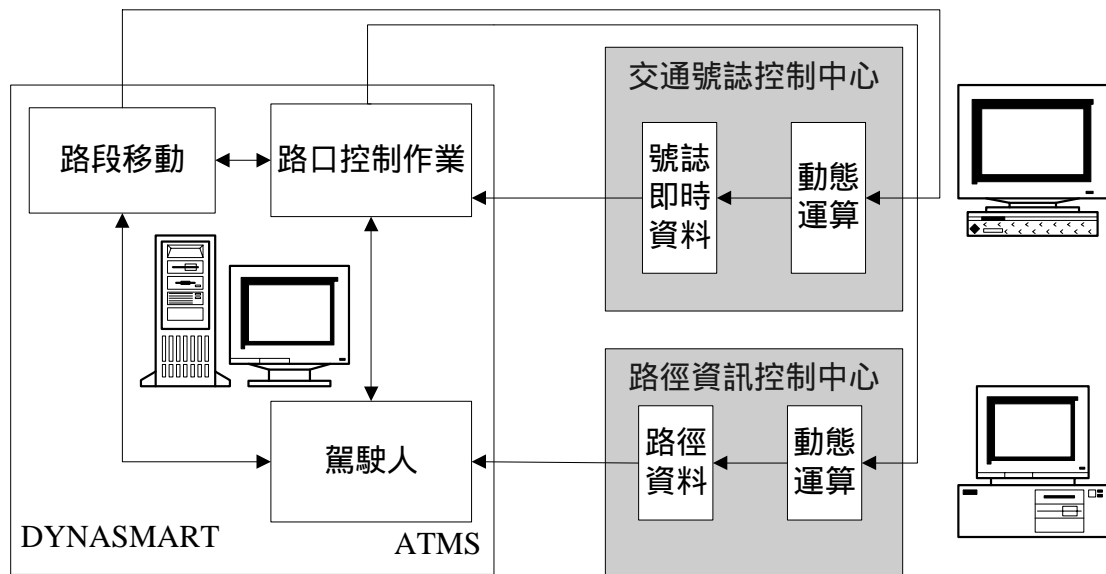


圖 3.5、分散式 DYNASmart 架構

2、交通號誌控制中心：負責號誌運算。

根據設置的號誌型態進行最佳化的運算，所產生的時制資訊除可直接應用於號誌上，並可提供給模擬運算進行交通管理之策略研擬，同樣的模擬運算的結果也可以用於交通號誌動態運算上。

3、資訊控制中心：負責路徑資訊之提供。

控制中心獲得模擬資料、號誌資料後進行路徑產生計算。

3.4 系統效能評估方式

多數 ITS 應用系統可依據服務對象區分為旅行者與系統管理人員兩大類，由於服務對象不同導致系統需求有所差異，可藉由使用者的使用狀況與目的大致區分系統需求的差異。一般而言，旅行者使用 ITS 應用系統可分為兩種情況，一為旅行中旅次行為決策的參考依據，此種情況因為旅行者使用該系統時已在旅程中，若此時需要系統的決策輔助，系統必須在短時間內給予有效的回應方能達成旅行者要求。另

一項為旅行前的旅次規劃，旅行者準備即將發生的旅次行為時，先行透過 ITS 應用系統規劃該旅次行為，此種情況旅行者依據 ITS 應用系統提供的建議進行規劃，若系統所提供的輔助無法給幫助旅行者作出有效的決策，旅行者將對此類型系統失去信心。

另一種類的 ITS 應用系統是以系統管理人員為主要服務對象，由於多數屬於控制管理的應用系統必須處理分析大量的資料或以複雜的模式或演算法進行計算。若與旅行者所使用的應用系統比較，差別在於管理系統的處理時間較長，不若旅行者應用系統需在短時間內回應（通常為數秒內），管理系統僅需在合理時間內完成工作即可。雖然管理系統對回應時間的要求並不高，大量資料處理或複雜的計算過程卻可能將回應時間放大至無法接受的範圍，因此，為能讓系統有更佳的處理能力，除改進模式或演算法外，可透過系統架構的改善以達到更好的效率。同時由於管理系統多是以管理中心為系統核心，無論是硬體的設備或軟體的設計都可有較大的彈性，可使用較高階的硬體或較複雜的架構。

綜上所述，可歸納出針對旅行者為服務對象的應用系統應以快速回應及正確性高為首要條件；而以系統管理人員為服務對象的應用系統應以合理反應時間與軟硬體需求為首要條件。本研究提出兩種主要的分散式架構，用以討論 K 條最短路徑運算元件的配置與系統建立選擇方案對系統整體效能的影響。在績效指標（Performance Indexes）的選取方面，將績效指標區分為以控制管理為主系統導向（System-oriented）及以服務為主的旅行者導向（User-oriented）兩大類。本節將介紹各項績效指標之定義。

3.4.1 旅行者導向績效指標

■ 回應時間（ $T_{response}$ ）

根據 3.3 節敘述之分散式架構，回應時間應包含三大種類：資料傳輸時間、資料轉換時間、與資料計算時間。資料傳輸時間（ $T_{transmit}$ ）是

指自使用者發出請求至收到系統回傳結果之間，資料在不同電腦之間的總傳輸時間，此處不將各段傳輸時間分開討論主要是因為此類型應用系統的服務對象通常遍佈在不同的地理位置，使用的設備亦不相同，導致可能的傳輸路徑非常複雜，因此採用所有旅行者傳輸時間的平均數。而過去不常被討論的資料轉換時間 ($T_{transform}$)，因為部分系統需要即時資訊的應用，使得系統必須等待資料再次轉換後方能進行下一步驟。此外，由於資訊種類不同所需轉換時間亦不同，必須將資料轉換時間與資料計算時間分開討論。最後一項為資料計算時間 ($T_{compute}$)，是指路徑計算元件求解的時間。回應時間的計算方式表示如式 3.1 所示。

$$T_{response} = T_{transmit} + T_{transform} + T_{compute} \quad (3.1)$$

■ 錯誤率 R_{miss}

錯誤率定義為獲得系統有效回應數佔發出請求總數的百分比。如式 3.2， N 為向系統發出請求且系統有記錄的總數； r 為獲得系統有效回應的數量。若使用者端因非屬系統錯誤導致無法將請求送至系統端，則試驗不列入樣本計算。

$$R_{miss} = \frac{N - r}{N} \times 100\% \quad (3.2)$$

3.4.2 系統導向績效指標

■ 處理時間 $T_{process}$

此處時間的定義與旅行者應用系統的回應時間定義類似，差異在於處理時間是以系統執行期間，程式中完成某一特定或複雜的方法或函數所必須花費的時間。

$$T_{process} = T_{transmit} + T_{transform} + T_{compute} \quad (3.3)$$

■ 記憶體使用 M_{usage}

管理系統在執行過程中通常需要計算分析大量的資料，必須使用大量程式記憶體以執行模式或演算法。而在分散式架構的各項優點中，各元件的單獨管理是相當重要的特性之一，若因記憶體需求較高而要求其他元件配合昇級或有所改變，則失去原本分散式架構的意義。因此將系統架構中所有電腦執行該系統所使用記憶體的總合作為硬體設備的主要績效指標。定義表示如式 3.4。

$$M_{usage} = m_1 + m_2 + \dots + m_n = \sum m_i, i = 1, 2, \dots, n \quad (3.4)$$

此處所選取的四個績效指標為通用性的績效指標，而各個指標之間並無絕對的重要性排列順序，其間關係可由系統開發者自行界定各指標不同的重要程度。透過四個指標的績效評估，可獲得系統的穩定度、負載能力、及基本配備需求，提供系統開發者於開發過程中相當重要的參考依據。



第四章 CORBA-Based 分散式 GIS-T 數值實驗

先進旅行者資訊系統 (ATIS) 是 ITS 的一個子系統，其主要目的是透過取得、分析、傳遞、及呈現資訊，以輔助旅行者獲得更佳的安全、效率、及舒適性。該系統主要以旅行者為服務對象，提供各種資訊服務，包含路徑、導引、號誌、及安全等資訊。本研究以一 ATIS 系統實例數值實驗，藉由不同系統架構進行測試，根據績效指標與實驗結果分析瞭解各種系統需求所適合的方案。

4.1 系統架構

ATIS 主要功能為提供旅行者作為行前或旅行中資訊的查詢工具，其目的在於讓使用者能作出更有效的決策或作為導引的工具。當系統普及後，系統所必須負荷的使用者將會提高許多，此時系統的負載能力將成為系統的效率門檻。此外，由於系統的使用者均由系統外部透過 Internet 連結，使用者可能會透過不同網路連結方式與系統溝通。因此，本章將以緊急救護車輛導引系統為例，以其急迫性及大量負載的特點表現出分散式架構對於快速反應及負載承受的處理能力。

4.1.1 分散式 GIS-T 之應用

本研究所使用的 ATIS 系統實例為一個 CORBA-Based 的分散式 GIS-T 的系統，於系統中可查詢相關的醫療設施，並將模擬數個醫療設施的物件來展現本系統的特色。該系統應用 GIS-T、資料庫、程式開發、網路通訊等關鍵性技術。

該系統中救護車上的編組人員，除了照料病患之外，並可藉由配備於車上之筆記型/手提式電腦，利用無線通訊方式，在圖形界面上將所搭載病患之狀況點選輸入，資料以物件方式傳遞至 ORB 代理伺服器向醫療院所及交通機構檔案伺服器提出查詢廣播，要求傳遞最新

正確資料，各地之檔案伺服器再回報現存之最新資料給予 ORB 代理伺服器，ORB 代理伺服器將查詢結果以物件方式傳回救護車上電腦，完成整個查詢動作。

由於網路節點之特性為不受外界災害影響，只要對外連結系統正常工作，任何一地之資料庫皆可上線供查詢，使用者也可由單機上網查詢緊急醫療救援系統資料庫之最新資料。ORB 代理伺服器即使因重大災難不能正常運作，仍然可以從系統中任何一部電腦立即架起另外一部 ORB 代理伺服器，迅速接替前者之工作；另外 ORB 跨平台的特性也可整合其他作業系統達到資源共享，符合分散式架構之精神。

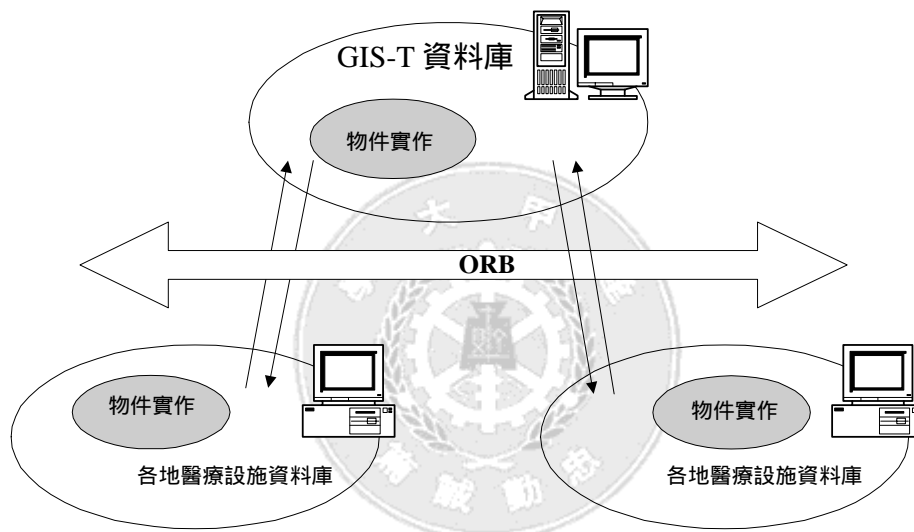


圖 4.1 CORBA 資料庫應用關係圖

資料來源：廖彩雲等 (1999)

4.2 程式架構與流程

本節將以圖 4.2 說明 ATIS 數值實驗的程式架構與流程。架構說明是以資訊的流動為主旨，可從中瞭解系統中各項資訊的流動；流程說明是以程式執行的步驟為主旨，透過圖示說明可輕易瞭解程式的執行細目與檢視效率瓶頸的問題。

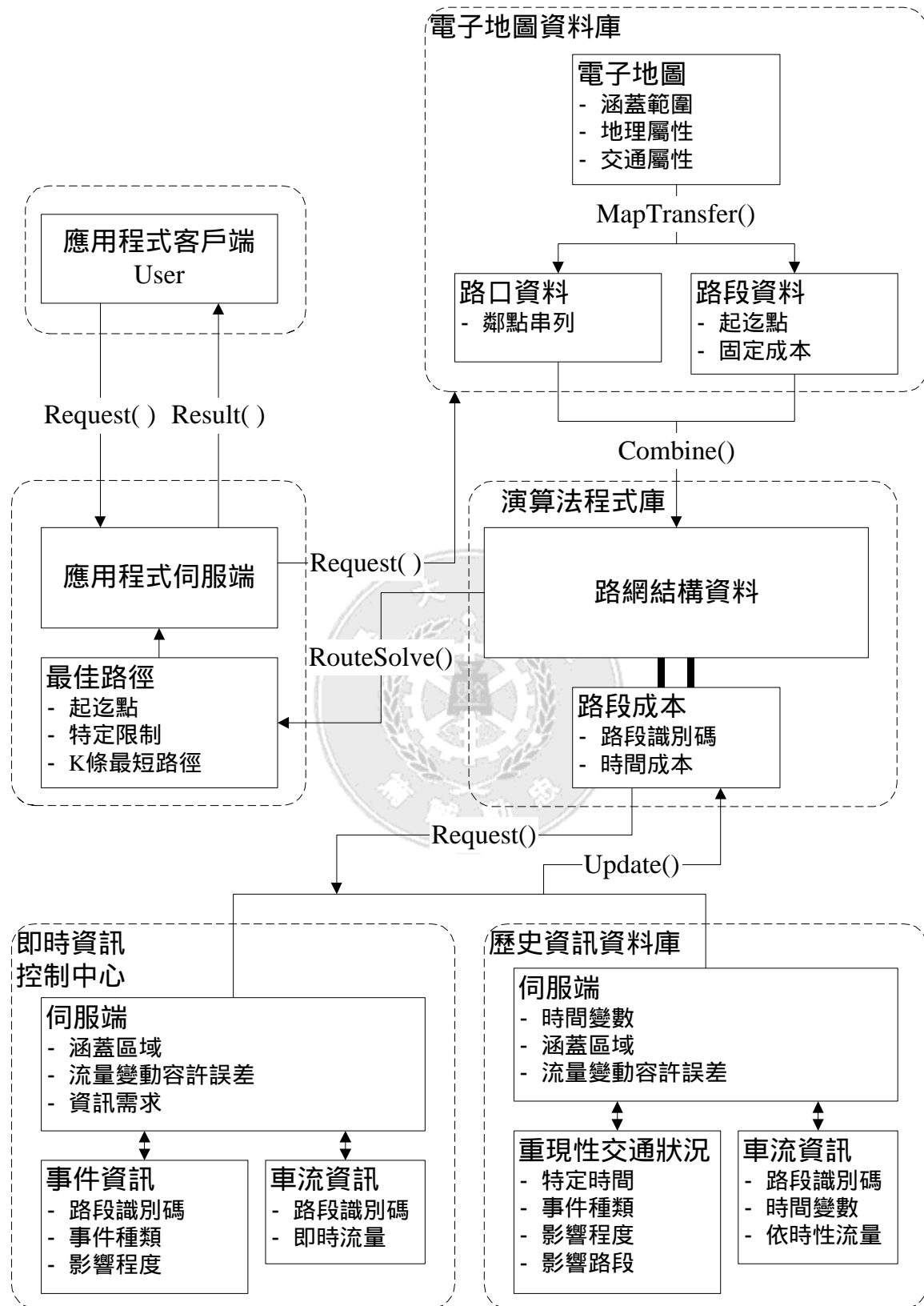


圖 4.2 程式架構

圖 4.2 中的演算法程式庫即為 KSP 計算元件，該元件的功能主要是計算出使用者所需要的路徑，其組成可分為三部分。

1. 演算法

此為計算元件的核心，透過演算法可求得使用者所需的結果。在元件中，此部分通常是執行效能的瓶頸，若欲大幅改善執行效能或更改系統需求，可修改或使用其他演算法。本研究是以 DoubleSweep 演算法做為求解 KSP 的主要演算法。

2. 輸入及轉換

包含基本資料及參數的設定，在本研究中資料輸入包含路網結構、路段成本、K 值、起迄點、成本種類。當元件開始進行求解的動作，所有的資料輸入行為都將被禁止，以免影響計算結果。轉換功能為將資料格式轉換至可由演算法核心進行計算的資料結構，主要是為了讓系統在開發過程中有更大的彈性，不會因為系統兩端使用不同的資料結構而需面對的資料溝通問題。

3. 輸出及其他

輸出功能為將結果傳回予使用者，若使用 one-to-one 演算法僅需將結果傳回即可；若使用 one-to-all 或 all-to-all 演算法則需自一組結果中將適當的結果挑選出來並回傳使用者。其他功能則包含單一節線或節點的增刪修改、路網之基本資料顯示、路網結構再確認...等。

4.2.1 程式架構說明

數值實驗所使用之程式架構如圖 4.2 所示，圖中虛線表示獨立的電腦，客戶端先將查詢路徑請求送至伺服器應用程式，而後伺服器通知電子資料庫索取目標區域之電子地圖並傳送至計算元件。電子地圖傳遞至計算元件後，向即時資訊資料庫與歷史資訊資料庫要求路段成本更新，路段成本設定完成後便以路徑求解演算法計算結果並傳回客戶端。其中，各電腦之間的通訊均透 IIOP 通訊協定與 ORB 進行連結。

經過初步實驗後，發現電子地圖資料的容量太大，無法在合理時間內產生回應，導致系統效率低落，主要原因是地圖資料量過大所致，因此實驗中假設電子地圖已存在計算元件中。

4.2.2 程式流程說明

本節以圖 4.3、圖 4.4 說明程式流程，分為計算元件置於客戶端與伺服器端兩種，當計算元件置於客戶端，系統可減少由計算元件傳遞資料至客戶端的時間；同時卻也增加即時資訊傳遞至計算元件的時間。在一般情況下，客戶端（亦即旅行者）通常只使用一部電腦設備，因此計算元件與客戶端以一部電腦代表之。

1. 計算元件置於伺服器端

步驟 1：客戶端將使用者之需求轉換為文字資料，並傳送至 ORB。

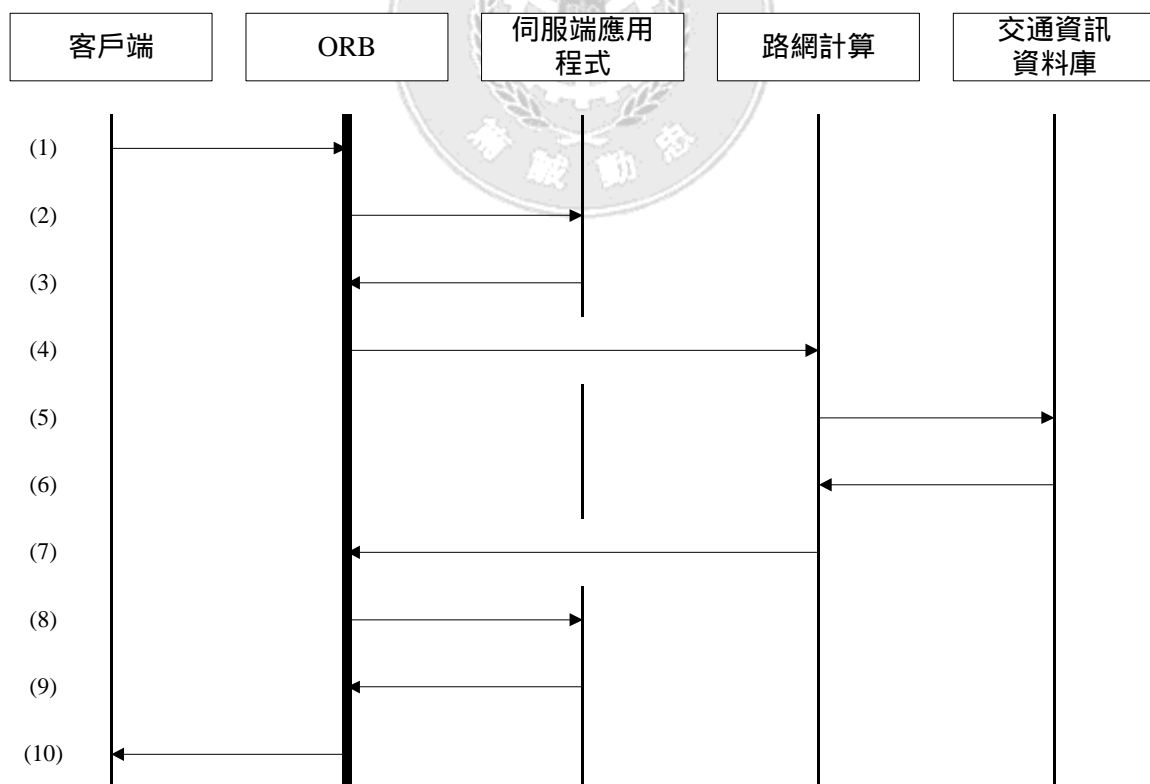


圖 4.3 分散式 GIS-T 之應用系統流程（計算元件置於伺服器端）

步驟 2：ORB 將需求資料傳送至適當之伺服器端。

步驟 3：伺服器端接收資料後，根據資料查詢其資料庫，並結合查詢結果與需求資料以文字資料傳送至 ORB。

步驟 4：ORB 將資料傳送至路網計算元件。

步驟 5：路網計算元件向交通資訊資料庫索取資料。

步驟 6：交通資訊資料庫將資料傳回予路網計算元件。

步驟 7：路網計算元件依照資料計算適合路徑資訊，並將結果以數字型態傳送至 ORB。

步驟 8：ORB 將路徑資訊傳回至伺服器應用程式。

步驟 9：伺服器應用程式將結果路徑傳至 ORB。

步驟 10：ORB 將路徑資訊傳回客戶端。

2. 計算元件置於客戶端

步驟 1：客戶端將使用者之需求轉換為文字資料，並傳送至 ORB。

步驟 2：ORB 將需求資料傳送至適當之伺服器端。

步驟 3：伺服器端接收資料後，根據資料查詢交通資訊資料庫。

步驟 4：交通資訊資料庫將所需資訊傳至伺服器端。

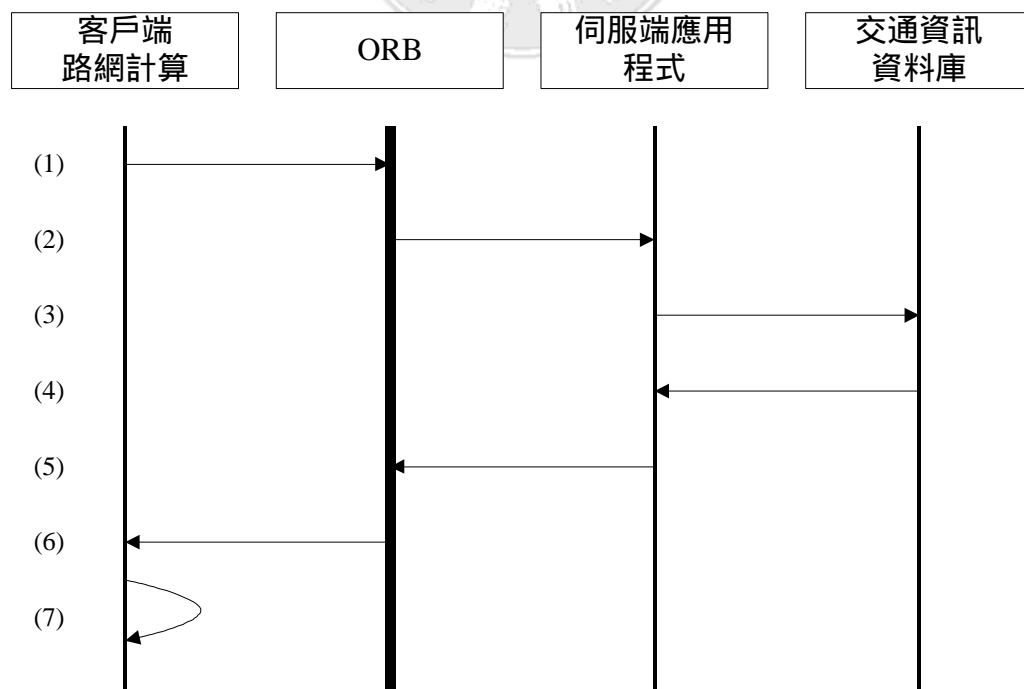


圖 4.4 分散式 GIS-T 之應用系統流程（計算元件置於客戶端）

步驟 5：伺服器端將資訊傳回至 ORB。

步驟 6：ORB 將路徑資訊傳回客戶端。

步驟 7：路網計算元件計算路徑查詢結果並顯示於使用者界面。

4.2.3 路段成本分類

路段成本可分為兩類，包含距離成本與時間成本，距離成本為靜態資料，可由電子地圖中之線段元件基本屬性中獲得；時間成本為變動成本，隨車流量改變即會造成路段旅行時間改變。此兩種資訊分別有其優缺點，靜態資訊的使用較為方便，直接讀取線段基本屬性即可，不需另外增加資料欄位儲存路段成本，缺點則是靜態資料無法反應即時性的交通狀況，所得結果正確性較低；動態資料的優點即在於可反應即時性交通狀況，而缺點即在於資料的處理速度，必須新增資料欄位以儲存旅行時間成本，並且在每一次進行動態路網分析前，需向遠端交通管理中心請求即時車流資料，以作為旅行時間估計之依據，為此將增加系統回應時間。因此，為兼顧系統目標與建置成本，必須選擇適當的分散式架構以達到效果最大、成本最小的目標。

在交通路網最短路徑的計算過程中，路段成本通常是以路段長度來表示，其最佳解為車輛的最短行駛路徑。然而，此最佳解實際運用的限制相當大，其假設為每一路段的單位距離旅行時間皆相等，即表示不會因時間地點不同而有所差異。因此，以距離轉換成旅行時間最短路徑在實務上僅可作為參考路線，若使用在需考慮即時資訊的狀況下，將無法滿足時間的需求。

為解決此一問題，在決定路段的旅行成本時，為能求得較符合真實交通狀況的最佳路徑，旅行成本以行駛時間來估計。而旅行時間的估計方法有多種方法，本節將針對不同旅行時間成本的估計與最短路徑應用的差異進行說明與比較。數值實驗中將以三種不同路段成本來進行實驗，以瞭解不同種類成本對系統目標與成本的影響差異，三種路段成本包含固定旅行時間成本、動態旅行時間成本、以及即時旅行時間成本。以下就三種路段成本作一詳細說明。

1 固定旅行時間成本

此一種類的路段成本與距離成本相當類似，各路段的成本不會因時間不同而變動。其主要差別在於路段因等級不同造成行駛速率不同，計算旅行時間成本時，等級不同的路段即使長度相同，旅行時間亦不會相同。此一成本類型可反應路段等級對行駛效率的影響。圖 4.5 說明旅行距離與旅行時間成本設定流程。

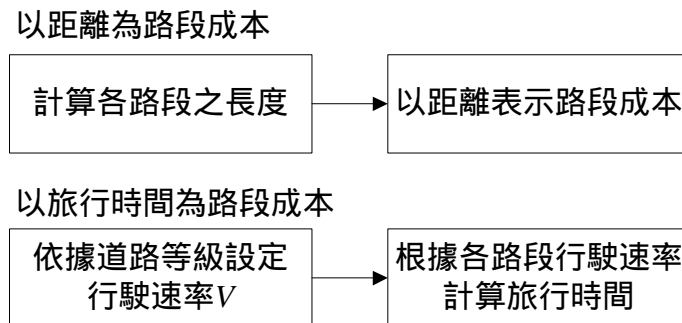


圖 4.5 固定旅行時間成本設定流程

2 動態旅行時間成本

在實際狀況中，車輛行駛時間會因時間地點不同而有所差異，此外，事故或施工亦會影響實際旅行時間。因此，為使結果更能符合實際交通狀況，採用依時間不同而改變的變動旅行時間成本，其方法是在計算最佳路徑時，以最接近的車流狀況來估計時間成本。

歷史資料是將過去所蒐集的車流資料透過分析方法求得該時間點的車流狀況，此一方式可減少資料蒐集的難度，僅需在系統使用前將歷史資料存入系統，再進行求解即可。其缺點即在於歷史資料仍舊無法反應即時性的交通狀況，由於此一限制，歷史資料適用於交通量穩定，且事故與施工路段較少的區域。圖 4.6 說明歷史資訊與路段成本設定流程。

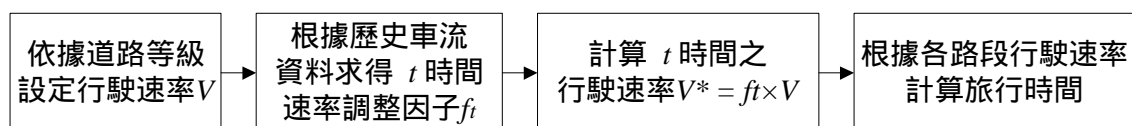


圖 4.6 動態旅行時間成本設定流程

3 即時旅行時間成本

即時旅行時間成本亦屬動態旅行時間成本之一種，是以即時交通

資訊計算該時間點之旅行時間，其中，又可根據資訊的取得與使用方式分為完全即時與半即時兩種。以下將說明此兩種成本類型。

(a) 完全即時車流資料

即時車流資料是在求解最佳路徑前，先將即時交通資料傳回系統端，待系統將路段成本更新過後，再進行求解步驟。即時資料可能包含車流狀況、事故、意外等資訊，透過此一類型成本所求解的最佳路徑準確性較高。但其缺點在於即時資料須在使用時向交通控制中心要求獲得即時資訊，若在移動中之運輸工具使用即時資料，則必須使用無線傳輸的方式進行資料傳遞。由於需要具無線傳輸功能的設備，為使用即時車流資料，系統設置成本將提高；此外，由於即時資料在使用前須先將所有的路段成本更新過一次，會增加求解的時間。圖 4.7 說明完全即時旅行時間成本設定流程。

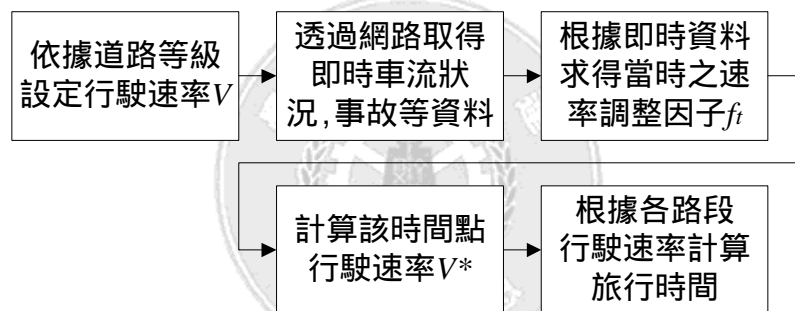


圖 4.7 完全即時旅行時間成本設定流程

(b) 混合車流資料

混合車流資料是希望能同時結合歷史資料與即時資料的優點，既能使用即時資料，亦可增加系統使用的效率。主要可分為兩類：一是降低即時資料傳輸量，一是減少成本改變路段。降低資料傳輸量的作法是以歷史資料為基礎，並取得部分即時資料，用以推估當時車流狀況；減少成本改變路段的作法是在獲得即時資料後，比較各路段的即時資料與最近的歷史資料，若差異在容許誤差內則不作改變，反之則以即時資料覆蓋歷史資料，以進行求解，此法可降低穩定車流狀況下，修改成本的路段數，以減少因成本改變路段數過多對計算效率造成的影響。圖 4.8 說明混合旅行時間成本設定流程。

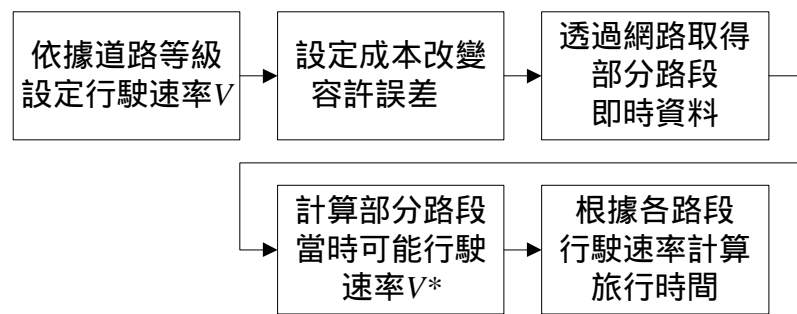


圖 4.8 混合旅行時間成本設定流程

表 4.1 將上述各種成本予以列表說明比較，從表中可得知目前所有資訊均可獲得，對於本研究路網模式計算的功能提供更大的彈性，可針對不同需求使用不同資訊。此外，更可將資訊應用於其他領域，如車輛導航、商用車輛派遣、旅行者資訊系統...等。

4.3 實驗設計

實驗設計中共包含測試環境、軟體、網路架構、測試資料、及 KSP 演算法，由於目前各項技術都有許多選擇方案，各方案亦有其特性及相互差異，因此本節將詳細說明各子元件的選擇方案及其特性。由於可選擇的方案相當多，為求降低將來實務上建立系統的難度，將以普遍常用的方案為主要測試元件，過於特殊或需要相當專業知識的方案則在研究範圍之外。

表 4.1 路段成本分類

成本屬性	資料種類	資料屬性	資料來源	資料蒐集方式
固定成本	距離	靜態	電子地圖路段元件長度屬性	不需要
	旅行時間		向主管機關索取各道路限制速率	人工
變動成本	歷史資訊	動態	各地方政府交通控制中心及調查報告 ¹	人工或網路傳輸(定期)
	完全即時資訊	即時	公路主管機關及各地方政府交通控制中心 ²	授權網路傳輸(即時)
	混合即時資訊			

4.3.1 測試環境

本章數值實驗以 10/100Mbps 區域網路為基礎網路架構，共以 5 部個人電腦及工作站相互連結進行測試。如圖 4.9(a)，若將計算元件置於伺服器端應用程式的後端，則電腦 A 負責執行客戶端應用程式；電腦 B 執行 ORB 伺服器、及伺服器端應用程式；電腦 C 為虛擬醫院資料庫；電腦 D 為虛擬即時資訊資料庫；電腦 E 儲存歷史資訊並執行計算元件。若將計算元件置於客戶端，則其架構如圖 4.9(b)所示，與圖 4.9(a)的差異在於電腦 A 負責執行客戶端應用程式以及儲存歷史資訊並執行計算元件；刪除電腦 E。

電腦 A：P-III 1.2GHz, 384MB ram

電腦 B、D：P-III 550MHz, 128MB ram

電腦 C：P-III 450MHz, 128MB ram

電腦 E：P-4 1.7GHz, 768MB ram

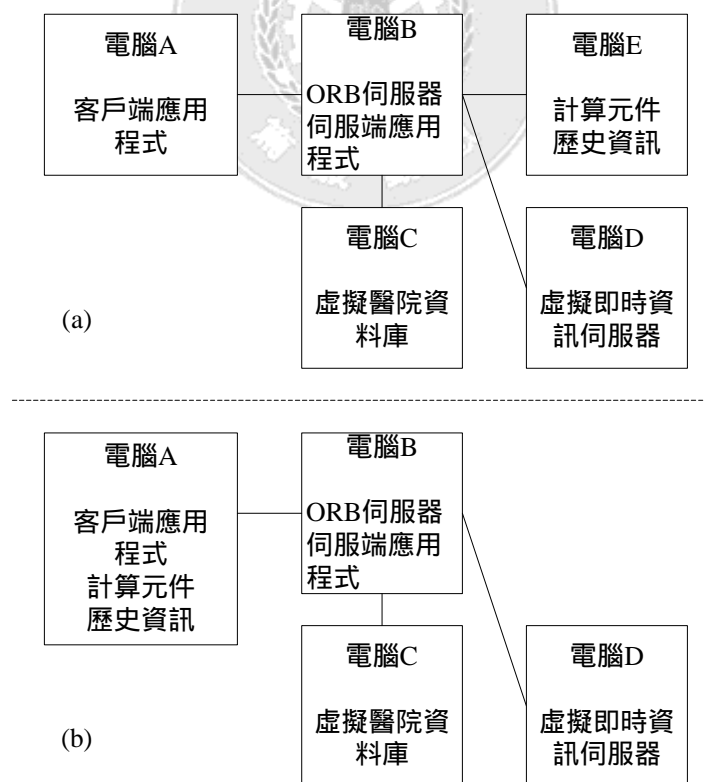


圖 4.9 CORBA-Based 分散式 GIS-T 實驗環境

4.3.2 軟體

軟體部分包含程式開發語言、ORB 伺服器、資料庫等軟體，其中 ORB 伺服器使用上較為困難亦牽動系統整體架構，因此本研究僅選擇 CORBA 單一 ORB 伺服器作為測試之用。其餘兩項軟體於目前有相當多的方案可供選擇，在同樣為降低系統建構的困難度的目標下，實驗中將選擇數種較常用的軟體作為測試元件，選用軟體如下。

(1) 程式開發語言

程式開發語言選擇常見的 C++ 及 Java 語言，此兩者目前皆有多套視窗設計軟體，可用於設計簡單易用之使用者介面。此外，用於測試之 DYNASMART 系統原為 Fortran 語言所撰寫，需以 C++ 或 Java 將該程式包裝成一元件，此部分細節將於後說明。

(2) ORB 伺服器

由於 ORB 伺服器軟體較為昂貴，因此以有發行測試版的軟體為主要考量。ORB 伺服器選擇 Inprise 所發行之 VisiBroker for C++/Java。

(3) 資料庫軟體

由於模擬醫療資料庫並未在本研究範圍內，因此僅選用一般的資料庫軟體，如 Microsoft 所發行之 Access 2000 軟體。

4.3.3 資料來源

實驗所需的資料來源主要分為真實資料與虛擬資料，其中真實資料包含作為地圖顯示基礎的電子地圖，及各醫療院所資料；虛擬資料包含旅行時間成本、客戶端需求、及虛擬大型路網結構。由於資料取得時均為原始資料，需作前置處理後始可用於實驗中測試，以下針對各項資料的來源及前置處理作一詳細說明。

(1) 電子地圖

交通路網數值地圖包含道路線形之空間資料與道路屬性資料。空間資料包括國道、省道、縣道、鄉道、產業道路、都市道路及其他道

路（原紙圖上未標示路名之道路）等交通路網；屬性資料內容包括道路各路段之代碼（含道路分級、道路編碼、道路所屬縣市別等資訊）、都市道路名稱、道路共線關係及節點類別（交流道、圓環、十字路口）等資訊。路線編碼方式採國土資訊系統交通網路資料庫之路線編碼原則予以編號。

電子地圖採用交通部運輸研究所發行的「新世紀台灣地區交通路網數值地圖」，該版電子地圖之原始格式為文字檔，需經由檔案格式轉換程式轉換成其他 GIS 軟體可讀取之格式。地圖中之路網圖層包含道路名稱、編號、等級等屬性資料，長度則為線段元件的基本屬性。實驗中，電子地圖用於客戶端輸入請求及結果顯現功能。

(2) 路網結構資料

實驗中之路網分析模式所使用的路網結構為真實路網，其格式即由電子地圖之道路圖層資料轉換而來，但由於運研所提供之電子地圖其路段與節點分開為兩圖層，欲得到之完整之路網結構需先經過兩圖層之比對，將點圖層之路口屬性資料與線圖層之線段端點進行比對，始得完整之路網結構。此外，運研所之電子地圖均依照區域分為許多小圖，在使用前需先將所有涵蓋目標區域的小圖結合成一目標區域圖，再行使用、轉換。以下詳列路網結構之準備作業：

1. 決定目標區域，挑選出所有涵蓋目標區域的小圖。
2. 以格式轉換程式修改檔案格式供 GIS 軟體讀取，再以 GIS 軟體將各小圖結合成一目標區域圖。
3. 以自行撰寫之圖層比對程式比對目標圖中的路線及節點圖層，以完成完整之路網結構。

(3) 醫療院所資料

各都市中之醫療院所單位眾多，礙於人力、物力的限制僅挑選區域醫院及教學醫院，小型醫院及診所並未在研究範圍內。資料蒐集方式以網路查詢及電話詢問兩種方式，透過此兩種方式蒐集醫療院所的基本資料，包含位置、醫護人員數、病床數、救護車數、特殊醫療器材等靜態資訊。

(4) 路段即時資訊

雖然國內已有許多提供交通即時資訊服務的單位，至目前為止仍沒有單位可提供特定範圍的完全即時交通資訊，因此實驗中將以虛擬的即時資訊作為資料來源。即時資訊設定方式為，首先令某路段之限制速率依據道路等級設定為 V ，一隨機變數（範圍：-0.2 ~ +0.2），則該路段之即時速率 $V^* = (0.8 + \beta) \times V$ 。實驗過程中，虛擬即時資訊伺服器以每 30 秒更新一次的頻率提供即時資訊。

4.3.4 網路架構

由於目前網路架構種類相當多，可從傳輸界面分有線、無線；亦可從速度區分寬頻、窄頻，其設置成本與特性亦不相同，本研究將根據系統建立目標與限制，選擇適合的方案進行測試，以瞭解效能與不同架構之間的關係。與測試硬體同樣考量，網路架構的選擇亦為常見的方案，若方案需要高度專業或高額成本，將在研究範圍外。經初步篩選後，共選擇兩種無線通訊與三種有線通訊網路架構，下表 4.3 分別將兩種網路架構作一摘要說明。

表 4.3 測試網路架構

網路架構		傳輸速度 (秒) (大約值)	特點
有線	56K 撥接	40k	<ul style="list-style-type: none"> ● 連接任一電話線路即可 ● 傳輸速度慢
	ADSL	1.5M(D)/384k(U)	<ul style="list-style-type: none"> ● 需使用固定電話線路 ● 傳輸速度較快
	固接網路	10/100M	<ul style="list-style-type: none"> ● 線路無法輕易變更 ● 傳輸速度快且穩定
無線	GSM	9,600b	<ul style="list-style-type: none"> ● 移動性相當高 ● 傳輸速度過低 ● 連線成本較高
	GPRS	100k	<ul style="list-style-type: none"> ● 與 GSM 同樣具有高度移動性 ● 傳輸速度大幅提升 ● 連線成本亦高

無線通訊之設置共分為三部分，包含行動電話為通訊工具之無線通訊、ORB server、和醫療院所資料庫伺服器端 (Hospital Database Server) 及查詢客戶端 (Query Client)。由於無線通訊技術較為複雜，以及一般撥接速度過慢，數值實驗中將不考慮該類通訊方式，僅保留 ADSL 與固接網路為主要網路測試環境。

4.4 系統操作流程

本節將分別以客戶端及伺服器端之電腦螢幕畫面輔以文字敘述說明系統操作流程。

如圖 4.10 系統開始執行時，首先將伺服器端之應用程式與 ORB 伺服器建立連結，以等待客戶端應用程式的呼叫。

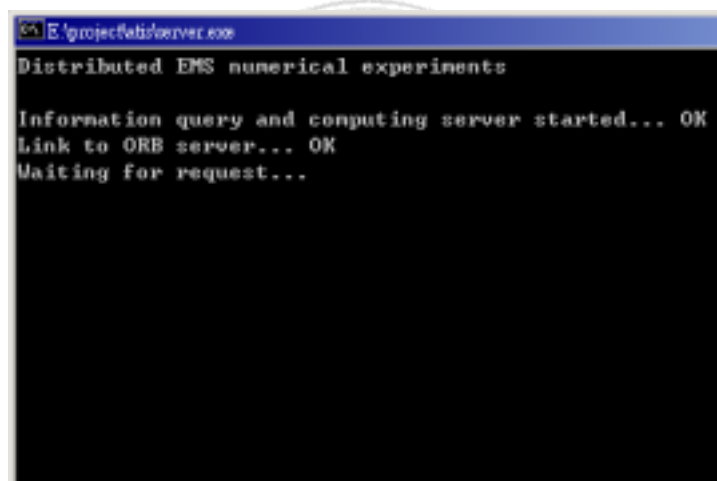
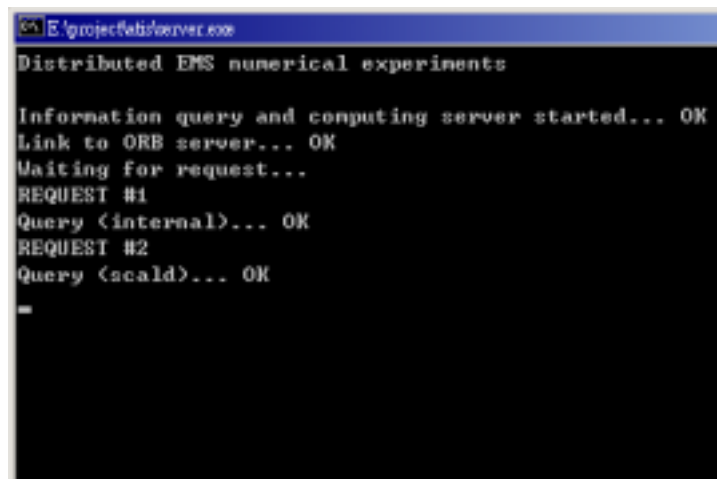


圖 4.10 伺服器應用程式與 ORB 建立連結



圖 4.11 客戶端應用程式啟動

接著啟動客戶端應用程式，程式主畫面如圖 4.11 所示，畫面中央為 MapX 元件所呈現之電子地圖，使用者可於地圖下方之選項填寫欲查詢醫院的資訊。按下「查詢」按鈕後，客戶端應用程式將查詢資料透過 ORB 伺服器傳送至伺服器應用程式，如圖 4.12 所示。

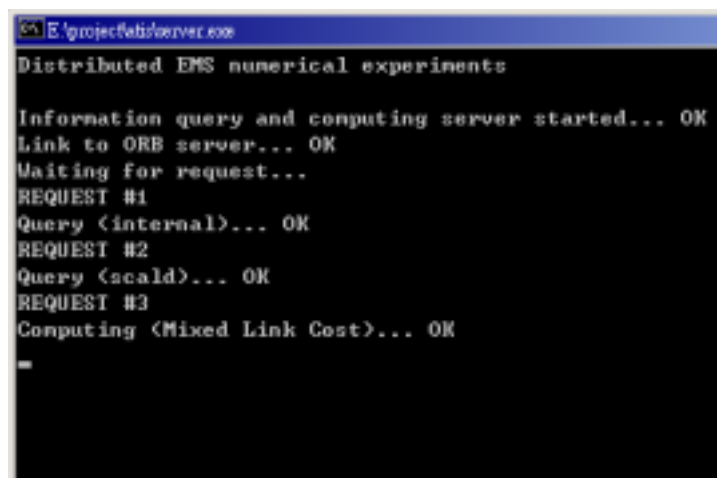


```
E:\project\server.exe
Distributed EMS numerical experiments

Information query and computing server started... OK
Link to ORB server... OK
Waiting for request...
REQUEST #1
Query (internal)... OK
REQUEST #2
Query (scald)... OK
=
```

圖 4.12 伺服器應用程式接收查詢資訊畫面

伺服器應用程式查詢結束後，將查詢結果循原路徑送回客戶端應用程式並於畫面上顯示。使用者再選擇其中一家醫院按下「路徑查詢」按鈕，此時客戶端應用程式將醫院、救護位置、最短路徑數 K、及路段成本種類等資訊傳送至伺服器，如圖 4.13 所示。



```
E:\project\server.exe
Distributed EMS numerical experiments

Information query and computing server started... OK
Link to ORB server... OK
Waiting for request...
REQUEST #1
Query (internal)... OK
REQUEST #2
Query (scald)... OK
REQUEST #3
Computing (Mixed Link Cost)... OK
=
```

圖 4.13 伺服器應用程式接收路徑查詢資訊畫面

路徑計算元件依照查詢資訊選擇合適的路段成本種類進行計算，圖 4.14 顯示客戶端要求以混合路段成本進行計算。計算結果傳回客戶端後將於電子地圖上呈現，如圖 4.15 所示。

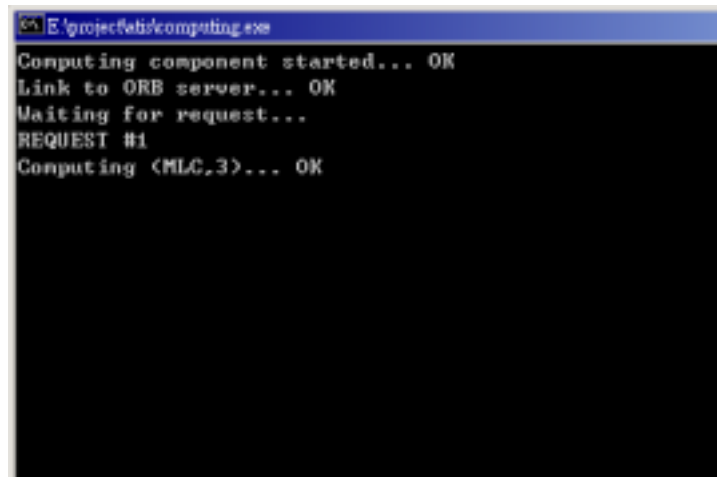


圖 4.14 計算元件選擇路段成本種類

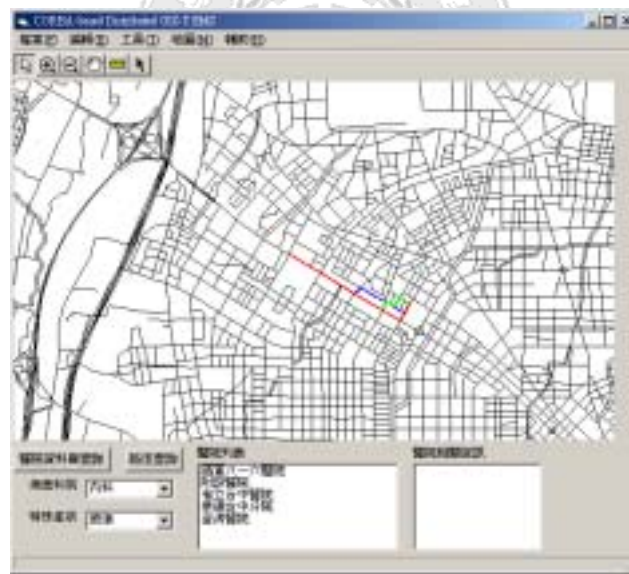


圖 4.15 客戶端應用程式顯示路徑查詢結果

4.5 實驗結果分析

1. 回應時間績效

根據下表 4.4 實驗結果可發現，即時資訊的傳輸時間佔用系統全

部資料傳輸時間的 65%，而在不使用即時資訊的狀況下，整體系統資料傳輸時間小到可忽略不計。當系統設計計算元件置於客戶端時，此一情況更為明顯，主要原因在於客戶端連結後端資料庫的網路頻寬小於連結的網路頻寬，導致客戶端取得即時資訊時需要更長的時間。

即時路段成本所需的時間最大，主要是因為資料轉換時間佔用相當大的比例，由於即時資訊是在系統接收使用者請求時向交通資訊提供者要求給予即時資訊，當系統接收全部即時資訊後始可修改原有路段成本，因此，即時資訊的傳輸及路段成本的修改大幅增加系統回應時間，導致系統效率降低。混合路段成本因有容許誤差值存在，可減少部分即時資訊的傳輸時間與路段成本修改的時間成本，但相對亦減少即時資訊的準確性。管理單位可根據系統目標與效率限制選擇合適之方案。

表 4.4 回應時間測試結果

計算 元件	成本種類	$T_{transmit}$ (sec)	$T_{compute}$ (sec)	$T_{transform}$ (sec)	$T_{response}$ (sec)	K-SP (sec)
位於 伺服器端	固定路段成本 (FLC)	0.004	2.023	0.085	2.721	1. 1489.88 2. 1552.52 3. 1698.99
	動態路段成本 (DLC)	0.004	2.109	1.585	4.120	1. 1229.17 2. 1334.31 3. 1498.94
	即時路段成本 (RLC)	0.910	2.078	5.844	9.357	1. 1034.55 2. 1108.47 3. 1149.86
	混合路段成本 (MLC)	0.352	2.051	3.189	6.078	1. 1129.23 2. 1209.55 3. 1271.53
位於 客戶端	固定路段成本 (FLC)	0.004	2.564	0.044	3.121	1. 1489.88 2. 1552.52 3. 1698.99
	動態路段成本 (DLC)	0.948	3.112	2.239	6.720	1. 1229.17 2. 1334.31 3. 1498.94
	即時路段成本 (RLC)	2.712	2.507	8.426	13.957	1. 1002.37 2. 1061.58 3. 1112.92
	混合路段成本 (MLC)	0.998	2.551	5.007	8.678	1. 1037.59 2. 1093.22 3. 1163.48

2. 錯誤率績效

由表 4.5 可知，兩種實驗的錯誤率都在 1% 以下，與計算元件的設計位置並無太大關係，主要是造成系統產生無效回應的原因為伺服器的負載過高導致部分程序消失。因此，若系統上線時發現有無效回應，則必須檢討是否伺服器已超過負荷，需要以增加伺服器或提高硬體設備方式提高伺服器的總負載能力。

表 4.5 錯誤率比較

計算元件配置	記錄請求 總數	無效回應	錯誤率
位於伺服器端	987	5	0.51%
位於客戶端	992	8	0.81%

客戶端發出請求總數：1000

第五章 CORBA-Based 分散式 DYNASMART 數值實驗

先進交通管理系統 (ATMS) 是 ITS 的一個子系統，主要目的是藉由整合新發展及現存的交通管理及控制系統，以期能在給予各種運具的服務反應出動態的交通狀況。為能提供動態交通狀況，需透過區域性監測設備測得地區性即時交通狀況，以做為動態分析之用。由於所蒐集之即時資訊通常包含許多相關單位，且資料量亦較大，因此 ATMS 系統需能在短時間內處理不同單位的大量資料。藉由先進通訊與資訊處理技術，可以分散式架構建置 ATMS 系統以達到快速處理不同單位資料的目標。在系統規劃過程中，軟硬體的选择與網路架構會因不同的系統需求而有不同的最佳方案，因此本章將透過 CORBA-Based 分散式 DYNASMART 實例進行數值實驗，透過績效指標與實驗結果分析得知不同系統架構下之系統績效。第一節將說明測試系統架構；第二節介紹實驗所使用之各種測試元件，包括軟硬體、資料來源、網路架構等；第三節將說明系統績效指標與實驗結果分析。

5.1 系統架構

由於 ATMS 負責整合各項運具間的服務，並以先進管理控制系統蒐集即時性交通資訊，以反應出動態交通狀況，例如區域性交通狀況監測系統、快速事故回應系統、交通策略最佳化等，因此在 ITS 架構中，ATMS 系統是最基本的子系統，其他子系統均需以 ATMS 為基礎進行系統建置。

5.1.1 ATMS 系統架構

ATMS 包含下列幾項主要特性：

- 蒐集即時性交通資訊。
- 使用依時性交通管理策略反應隨時間變動的交通狀況。
- 區域性調查與監測系統。
- 整合不同管理功能，包括運輸資訊、需求管理、公路匝道儀控、電子收費、及幹道號誌控制。
- 交通管理單位相互合作以作出最佳化策略達到改善車流的目標。
- 事故管理快速反應策略。

交控中心與其他相關單位分別蒐集所需的即時性交通資訊，並視需要由交控中心單獨或與其他單位合作提出最佳化控制策略，控制策略更新頻率可依據系統需求設定即時、秒、分、日、月等時間單位。

5.1.2 分散式動態交通模擬指派系統

本研究中，將以一分散式動態交通模擬指派系統做為 ATMS 系統的實例，該系統的主程式為 DYNASMART，以 Fortran 語言進行撰寫、編譯，卻也由於 CORBA 標準至今仍未將 IDL 加入對應 Fortran 的規格，使得在發展 CORBA 架構下分散式運算系統，面臨一些困難。在 CORBA 標準中，因 C++ 程式語言為完整的物件導向程式語言，是常被用來開發的程式語言之一，加上 IDL 亦有相對應 C++ 的規格，因此本研究以 C++ 語言包裹 DYNASMART 主程式，並以管線資料流 (Pipes and Streams) 包裝 (Wrapper) 方式，做為 C++ 語言與 DYNASMART 主程式之間的溝通界面 (Interfaces)。

此種類型的界面設計，最主要在於將 C++ 程式與 FORTRAN 程式兩者之間建立起連絡的管線 (Pipes)，而資料流 (Streams) 便可以經由管線做訊息資料的傳遞。同樣地，在執行 C++ 包裝程式 (Wrapper Application) 的同時也會驅動外部 DYNASMART 主程式的執行，根據 C++ 語言的特性，每當執行一外部命令時，在 C++ 程式本身會以程序 (Process) 物件對應此外部命令做為識別、控制之用，並由程序物件產生一輸出資料流 (OutputStream) 銜接至該外部命令的標準輸入，以及

產生一輸入資料流 (InputStream) 銜接至該外部命令的標準輸出。

如圖 5.1 所示，在分散式 DYNASMART 執行的過程中，由 DYNASMART 主程式由標準輸出送出請求，C++ 程式便可輸入資料流接獲請求，並向 ORB 智慧代理商尋找適合的伺服器端，將所需要的資料再由 ORB 智慧代理商傳回 C++，此時已傳回的資料可立即由輸出資料流傳回 DYNASMART 主程式的標準輸入，DYNASMART 亦可由標準輸入讀取之方式完成資料的更新。上述的操作步驟亦將反覆地執行，直到 DYNASMART 主程式已完成所需的模擬運算時段才停止。

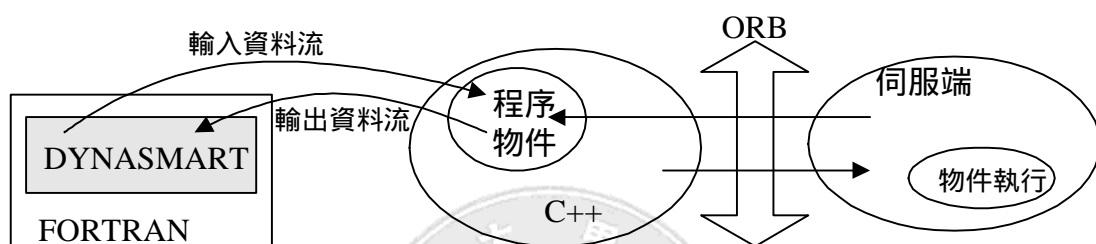


圖 5.1 管線資料流界面型分散式 DYNASMART 執行流程

資料來源：Mowbray and Ruh (1997)

5.2 程式架構與流程

如 3.3.2 節所述，DYNASMART 軟體執行時，其最大的計算步驟為路網中 All-to-All 的 KSP 計算，若能有效此一部份的計算負擔，將可增加其執行效率。因此，本研究將每一遞子迴 (iteration) 中所需計算的 All-to-All KSP 問題切成 One-to-All 的數個計算單元，並將計算單元同時給予不同元件計算，希望能減少 KSP 的執行時間。本節將分別說明程式的架構與流程。

5.2.1 程式架構說明

本研究所採用之 ATMS 數值範例—分散式動態交通模擬指派系統之架構圖如圖 5.2 所示。DYNASMART 主系統首先將計算 KSP 的請求與計算元件個數傳送予程式控制介面，該介面再將請求透過 ORB 傳

送至伺服端。此時伺服端根據使用者所設定的分散計算元件個數，將計算 KSP 的工作逐一派給不同電腦上的計算元件執行。當所有計算工作完成後，伺服端將結果傳送回 DYNASMART 主系統並繼續進行下一遞子迴。

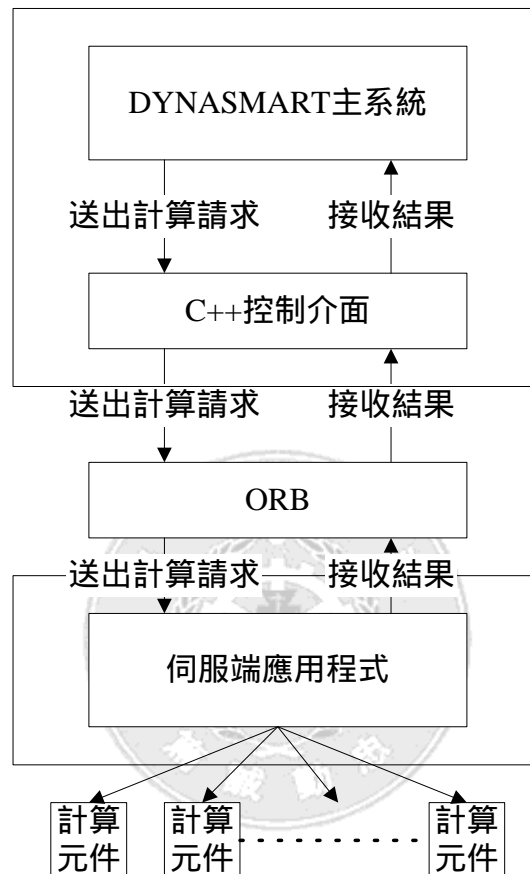


圖 5.2 分散式 DYNASMART 執行流程

5.2.2 程式流程說明

DYNASMAT 在模擬過程中會依據使用者的設定於特定模擬時段進行路網中 All-to-All 的 KSP 運算，此部分為目前最耗時間的運算步驟。圖 5.3 之程式流程即為系統要求計算一次 KSP 的流程，整個模擬過程中將多次執行此流程。主系統先將要求計算的命令傳送至伺服端，之後再分配至路網計算元件；計算完成後，元件將結果循原路徑傳回至主系統。實驗中所比較之回應時間是指完成一次 KSP 運算的時間，從主系統發出命令開始，至取得該次計算結果所需的時間。

- 步驟 1：DYNASMART 將計算的命令、及計算元件個數傳送至 ORB。
- 步驟 2：ORB 將命令傳送至應用伺服器。
- 步驟 3：應用伺服器將資料傳送至 ORB。
- 步驟 4：ORB 將資料傳送至路網計算元件。
- 步驟 5：路網計算元件依照請求資料計算出路徑資訊後，將路徑資訊傳回至 ORB。
- 步驟 6：ORB 將路徑資訊傳回至應用伺服器。
- 步驟 7：應用伺服器將資料傳送至 ORB。
- 步驟 8：ORB 將路徑資訊傳回 DYNASMART 主系統，並等待下一個命令。

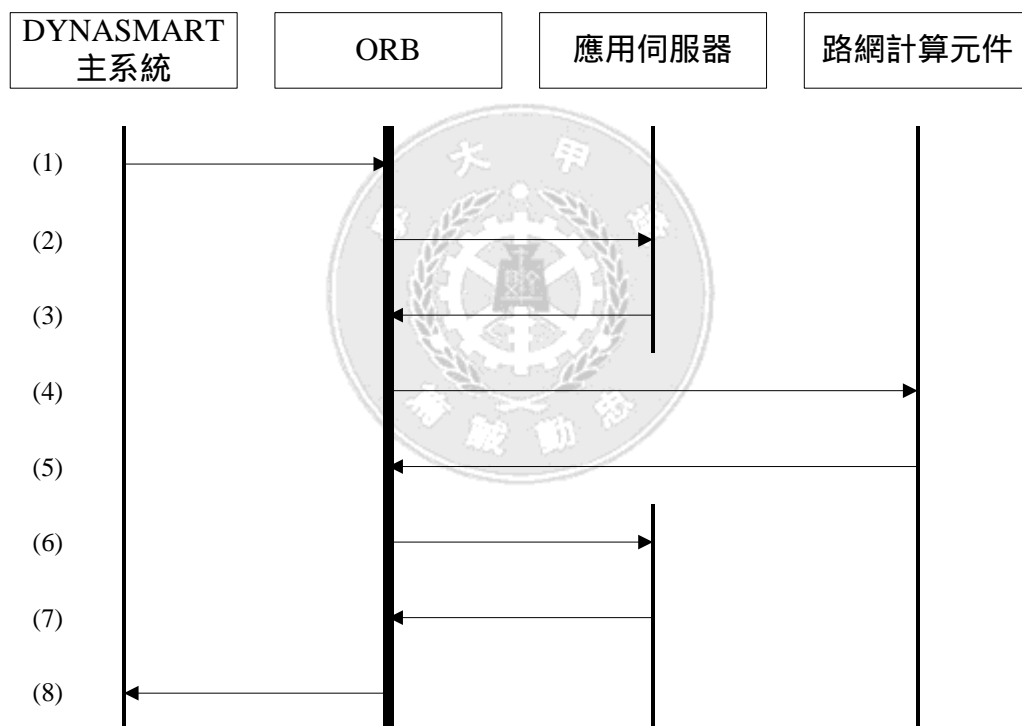


圖 5.3 分散式動態交通模擬指派系統流程

5.3 實驗設計

實驗設計中共包含測試環境、軟體、測試資料，由於各方案有其特性及相互差異，因此本節將詳細說明各子元件的選擇方案及其特性。由於可選擇的方案相當多，為求降低將來實務上建立系統的難度，

將以普遍常用的方案為主要測試元件，過於特殊或需要相當專業知識的方案則在研究範圍之外。

5.3.1 測試環境

本章數值實驗同樣以 10/100Mbps 區域網路為基礎網路架構，共以 5 部個人電腦及工作站相互連結進行測試。如圖 5.4，電腦 A 負責執行 DYNASMART 主程式；電腦 B 執行 ORB 伺服器、伺服端應用程式、及計算元件-1；其餘三部電腦 C, D, E 執行計算元件-2, 3, 4。

電腦 A：P-4 1.7GHz, 768MB ram

電腦 B、D：P-III 550MHz, 128MB ram

電腦 C：P-III 450MHz, 128MB ram

電腦 E：P-III 1.2GHz, 384MB ram

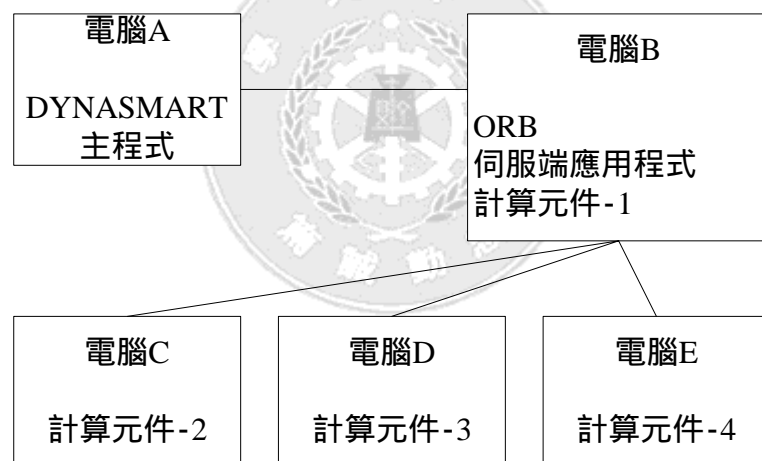


圖 5.4 分散式 DYNASMART 實驗環境

5.3.2 軟體

IDL 與應用程式的開發如圖 5.5。IDL 檔經過 idltocpp 編譯器編譯過後，產生連結客戶端與伺服端的 stub 與 skeleton，物件執行程式碼可利用 skeletons，經 C++編譯器編譯產生一可執行之程式，客戶端應用程式碼可利用 stubs，經 C++編譯器編譯產生一可執行之客戶端程式。

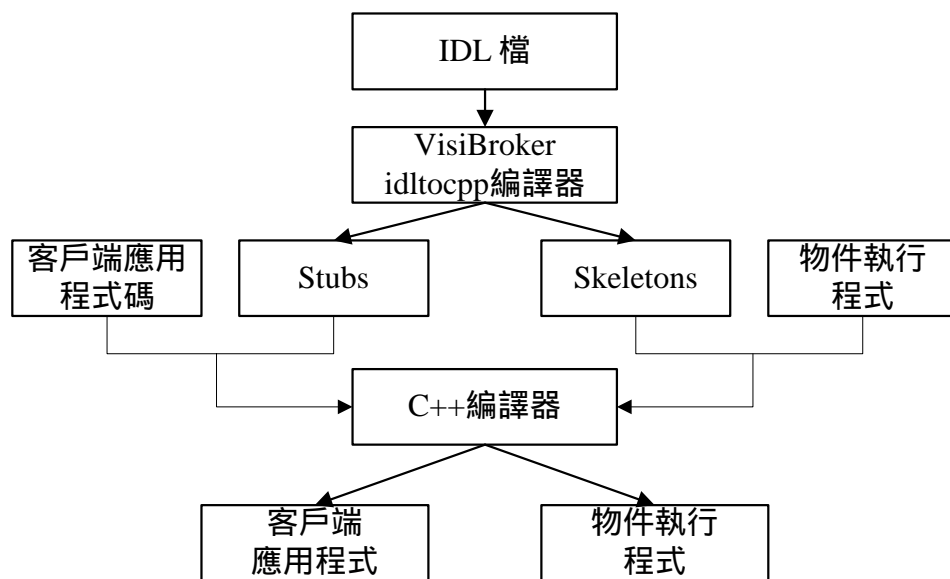


圖 5.5 CORBA 程式發展流程

資料來源：U.S. DOT (1999b)

由於 Windows 2000 版的 DYNASMART 以 Digital Visual Fortran 6.0 編譯，而 CORBA 2.0 尚未制定 IDL 對應 Fortran 的規格，因此數據機實驗中使用 C++ 程式包裹 DYNASMART，並以檔案檢查的方法控制 DYNASMART 的執行。

在分散式 DYNASMART 中，伺服器端安裝 Inprise 所發展的 VisiBroker Smart Agents，智慧代理商 (Smart Agents) 延伸 CORBA 規格，可加速獲得物件參考指標 (Object References)，智慧代理商會自動因伺服器端執行失敗而將無法連結的客戶端重新連結至其他適合的伺服器端。它提供客戶端應用程式及物件執行部分一個便利的動態分散式位址服務，當客戶端應用程式呼叫任一物件時，智慧代理商找出指定的執行和物件的位置，在客戶端應用程式與執行之間建立一個連結。

智慧代理商可於任何一部主機執行，客戶端應用程式及物件執行可對區域網路內各部機器發出一個 UDP (User Datagram Protocol) 協定的廣播訊息 (Broadcast message)，第一個產生回應的智慧代理商即表示連線成功，在確定智慧代理商的位置後，即建立一點對點 (point-to-point) 的 UDP 連線，因為 UDP 協定比 TCP 協定佔用較少的網路資源，且註冊資訊與請求位置都是動態更新，不需維護其設定。

當不同主機同時執行智慧代理商時，每一個代理商都會對可獲得的物件及其他代理商可獲得的物件重新進行組織。

5.3.3 測試資料

由於 DYNASMART 的路網資料需要較詳細的交通屬性，如車道數、速限、時制計畫等，且需要交通分區及 OD 等資料，而目前新路網並無上述相關資料，因此實驗中仍採用舊有的台中市路網及 OD 資料，該路網中包含 1800 條路段與 534 個節點。

5.4 系統操作流程

如同 4.4 節，本節同樣以之電腦螢幕畫面輔以文字敘述說明此一分散式交通模擬指派系統操作流程。

如圖 5.6 系統開始執行時，首先將伺服器端計算元件與 ORB 伺服器建立連結，以等待 DYNASMART 主系統的呼叫。

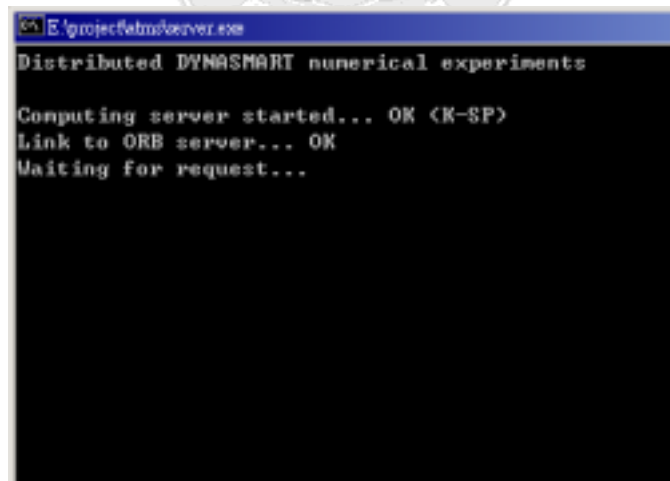
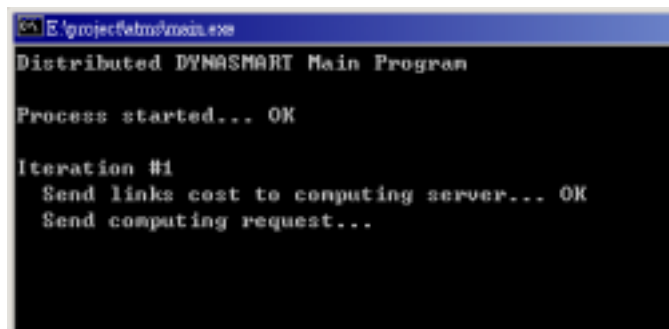


圖 5.6 伺服器端計算元件與 ORB 建立連結

圖 5.7 顯示系統主程式畫面，開始執行後 DYNASMART 有一段時間進行初始化的動作，接著於特定模擬時段發出計算 KSP 的請求。



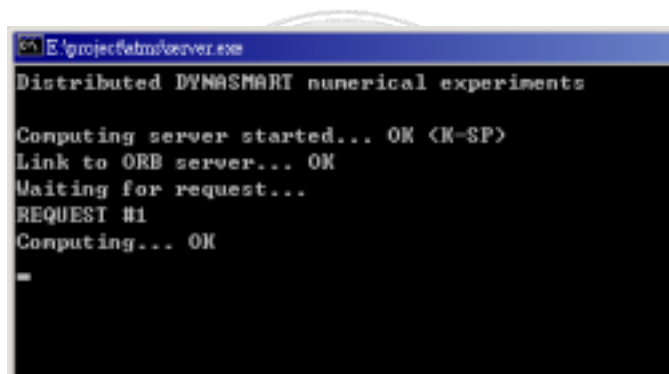
```
E:\project\dm\main.exe
Distributed DYNASMART Main Program

Process started... OK

Iteration #1
  Send links cost to computing server... OK
  Send computing request...
```

圖 5.7 DYNASMART 主程式啟動畫面

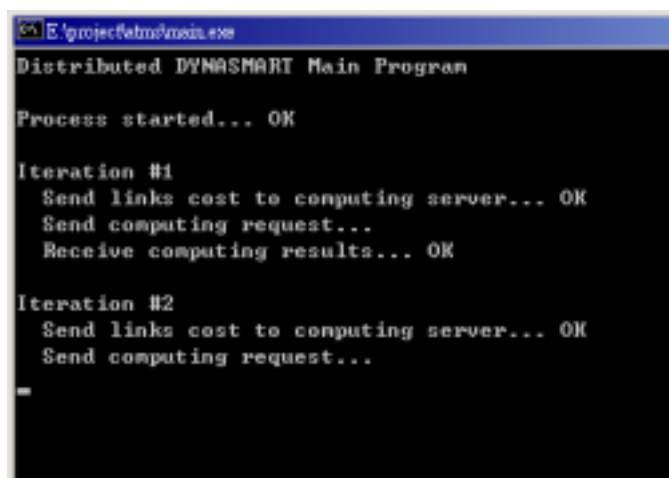
計算元件接收請求後同時開始接受主程式所產生之即時路段資訊以進行 KSP 之計算，計算完成後將結果傳回主程式，如圖 5.8、圖 5.9 所示，並等候下一個模擬時段的呼叫。



```
E:\project\dm\server.exe
Distributed DYNASMART numerical experiments

Computing server started... OK (K-SP)
Link to ORB server... OK
Waiting for request...
REQUEST #1
Computing... OK
-
```

圖 5.8 伺服端計算元件接收計算請求畫面



```
E:\project\dm\main.exe
Distributed DYNASMART Main Program

Process started... OK

Iteration #1
  Send links cost to computing server... OK
  Send computing request...
  Receive computing results... OK

Iteration #2
  Send links cost to computing server... OK
  Send computing request...
-
```

圖 5.9 DYNASMART 主程式接收計算結果畫面

5.5 實驗結果分析

1. 回應時間績效

由下表 5.2 可得，處理時間會因為路網計算元件的增加而減少，但可由圖 5.10 得知減少的趨勢亦會因為元件增加而減緩。主要原因在於計算元件增加雖可有效增加系統總體計算效率，但原本單機架構使用的演算法為 All-to-All 演算法，而分散式架構是採用 One-to-All 演算法，演算法效率下降，以致增加計算程序總量。此外，由於採用分散式架構進行運算，路段成本必須跟隨系統發出的命令傳送至計算元件，此步驟亦明顯增加資料傳輸時間。

本節所進行之數值實驗中，雖然傳輸時間與轉換時間各有其變化趨勢（如圖 5.11 所示），但由於在系統回應時間中僅佔有相當小的比例（不足 1%），因此對於系統回應時間的變化並無顯著影響。

表 5.1 處理時間與記憶體測試結果

計算元件 個數	$T_{transmit}$	$T_{compute}$	$T_{transform}$	$T_{process}$	記憶體使用 (MB)
1	0.012	284.144	0.219	285.753	97.55
2	0.033	168.835	0.200	170.911	99.42
4	0.071	113.274	0.213	114.854	104.31

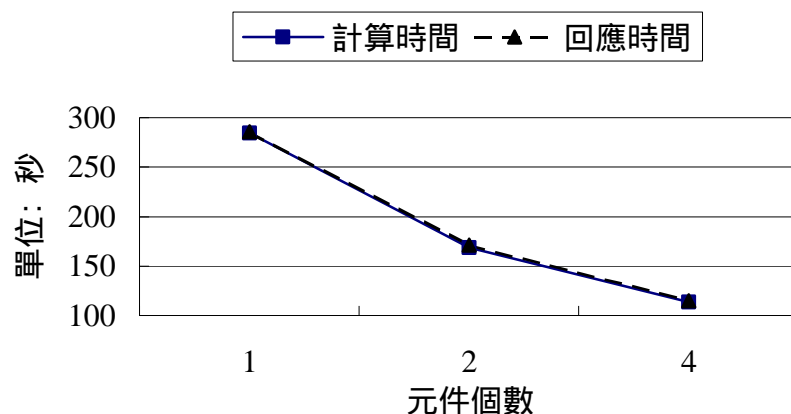


圖 5.10 元件個數與計算時間和回應時間之關係

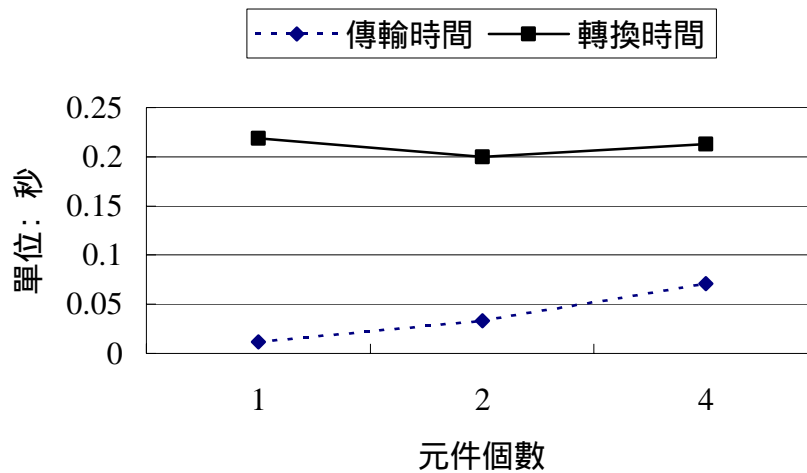


圖 5.11 元件個數與傳輸時間和轉換時間之關係

2. 記憶體績效

從表 5.1 中可知記憶體的用量會與計算元件個數成正比，但增加幅度並不明顯，主要是因為 DYNASMART 主程式本身已耗費大量的記憶體（約 96MB），且數值實驗中的台中市路網僅有 600 個點，計算元件僅需使用約 1.5MB 的記憶體進行計算。另外，與處理時間績效的比較，可看出兩者是約略成反比的關係，因此，在開發過程中需同時針對系統設計對兩績效的影響作一評估，以求得系統最大效能。

第六章 結論與建議

本章節將彙集研究的各項成果與數值實驗結果以作出本研究的結論，並針對研究不足之處加以說明以作為後續研究參考之用。

6.1 結論

■ 分散式架構於 ITS 之應用

文中首先將現行 ITS 架構簡化成為一包含四項元件，旅行者、管理中心、車輛、路側設施，之間以通訊元件彼此連結的分散式架構。透過 ATIS 與 ATMS 的相關討論，將具備跨平台與跨程式語言特性的 CORBA 分散式架構作為 ITS 的應用系統架構。

在實際系統建置過程中，分散式系統通常需要不同單位之間進行溝通，若欲進行整合則需一較高層級的單位負責主導，以避免各單位溝通不良的情況發生。以美國為例，多數分散式系統的開發是由運輸部 (Department of Transportation) 或各州政府運輸部門主導整個計畫；相對而言，國內則可由交通部、部屬運輸研究所、或各縣市交通局進行整合。

■ 績效指標選取

為能有效分析系統效能，第三節中根據 ITS 架構的討論及 ATIS/ATMS 子系統分析選取四個績效指標，包括回應時間、錯誤率、處理時間、記憶體使用，並且將上述指標按照系統服務對象區分為旅行者導向與系統管理導向兩大類，以作為系統效能分析的依據。

■ 路段成本種類對系統效能之影響

根據數值實驗中 ATIS 範例系統的測試結果可知，系統中所使用的路段成本種類不同會造成回應時間增加 1.5~2.5 倍。因此，在設計

系統過程中需詳細瞭解系統目標，以選擇使用符合目標需求且有較高效率的資訊種類。

■ 計算元件個數對系統效能之影響

在 ATMS 的範例測試中，KSP 計算步驟同時以 1 至 4 個元件進行運算效率測試，雖然測試結果顯示出總體計算效能會因計算元件增加而提昇，相對的記憶體使用量亦與計算元件個數成正比，且透過分散式架構進行即時運算所需額外的資料傳輸時間亦會造成總體計算效能降低。因此，分散式計算架構的使用，必須在單機的計算時間大於分散式架構下傳輸時間與計算時間總合的情況下，且記憶體的增加需在可接受的範圍內。

6.2 建議

■ 績效指標的分析標準

研究中所訂定的績效指標僅作為系統效能評估之用，未來應對各指標建立一標準，以作為系統設計的參考依據。

■ 混合路段成本改變門檻值

根據測試結果可知混合路段成本較即時路段成本有較佳的計算效率，且根據一般經驗可知混合路段成本在流量變化大或有事故影響的區域仍可提供較佳的資訊正確性。唯在本研究中僅將成本改變門檻值設為定值 (0.8)，後續研究可對值的設定及其影響作更深入的研究。

■ 分散式演算法的使用

在 ATMS 數值實驗中，僅以一般演算法進行 KSP 運算，因此效率提昇受到限制；而在考量分散式架構的特點後，建議後續研究可以分散式 KSP 演算法進行求解，應可提高系統總體計算效率。

■ 廣泛的軟硬體測試

由於本研究受限於軟硬體的缺乏，僅以一般系統建置軟硬體進行

測試，測試結果的應用範圍因此受到限制。未來可增加更多軟硬體設備進行測試，則測試結果將可應用更為廣泛，且可獲得較精確之分析。



參考文獻

- 王晉元（1994），運輸地理資訊系統資料格式標準化之可行性研究，交通大學運輸工程與管理學系。
- 何金駒、林富泰（2001），「以先進交通管理系統需求制定不同交控中心間之通訊協定」，交通部 90 年智慧型運輸系統研討會論文集，頁 96-109。
- 卓訓榮、林文斌（1991），「YEN 第 K 條最短路徑演算法及其應用」，交通運輸，第 13 期，頁 201-215。
- 卓訓榮、陳信雄（1993），「電腦平行處理技術在 YEN 第 K 條最短路徑演算法之應用」，土木水利，第 19 卷，第 4 期，頁 21-33。
- 胡大瀛、陳建緯、洪百賢、陳麗雯（1997），「網際網路下分散式模擬指派模式之建立」，第 2 屆運輸網路研討會論文集，頁 243-252。
- 胡大瀛、廖彩雲、洪百賢（1998），「CORBA 於分散式模擬指派模式之應用」，第 3 屆運輸網路研討會論文集，頁 1-9。
- 張堂賢、韋理為（2001），「我國現行交控通訊協定與 NTCIP 間界接與訊息交換之研究」，中華民國運輸學會第 16 屆論文研討會論文集，頁 731-741。
- 陳俊良、蕭偉政、林俊彥（1999），「以多階層式架構與 CORBA 技術發展交通管理中心資訊交換模型」，第 1 屆台灣智慧型運輸系統國際研討會論文集，頁 B1-11~25。
- 廖彩雲、蔡昭平、洪百賢、胡大瀛（1999），「CORBA-Based 分散式 GIS-T 於緊急醫療事件之應用」，中華民國運輸學會第十四屆論文研討會論文集。

蔡淑蓉 (1996) , 分散式物件導向設計在可調性計算之應用 , 逢甲大學資訊工程研究所碩士論文。

Akgün, V., Erkut, E., and Batta, R. (2000) “On Finding Dissimilar Paths”, *European Journal of Operation Research*, Vol. 121(2), pp. 232-246.

Andrisano, O., Verdone, R., and Nakagawa, M. (2000) “Intelligent Transportation Systems: The Role of Third-Generation Mobile Radio Networks”, *IEEE Communications Magazine*, Vol. 38(9), pp.144-151.

Bala, K., Stern, T. E., Simchi-Levi, D., and Bala, K. (1995) “Routing in a Linear Lightwave Network”, *IEEE/ACM Transactions on Networking*, Vol. 3(4), pp. 459-469.

Beynon-Davies, P. (1999) “Human error and information systems failure: the case of the London ambulance service computer-aided despatch system project”, *Interacting with Computers*, Vol. 11(6), pp. 699-720.

Carlier, J., Li, Y., and Lutton, J. -L. (1997) “Reliability Evaluation of Large Telecommunication Networks”, *Discrete Applied Mathematics*, Vol. 76, pp. 61-80.

Chang, G. L., Junchaya, T., & Santiago, A. J. (1994) “A Real-Time Network Traffic Simulation Model for ATMS Applications: Part I- Simulation Methodologies,” *IVHS Journal*, Vol. 1(3), pp. 227-241.

Chen, H. K., and Feng, G. (2000) “Heuristics for the stochastic/dynamic user-optimal route choice Problem”, *European Journal of Operation Research*, Vol. 126, pp. 13-30.

- Fu, L., and Rilett, L. R. (1998) "Expected Shortest Paths in Dynamic and Stochastic Traffic Networks", *Transportation Research Part B*, Vol. 32(7), pp. 499-516.
- García-Ortiz, Amin A., S. M., and Wootton, J. R. (1995) "Intelligent Transportation Systems--Enabling Technologies", *Mathematical and Computer Modelling*, Vol. 22(4-7), pp. 11-81.
- Gokhale, A. S., and Schmidt, D. C. (1998) "Measuring and Optimizing CORBA Latency and Scalability Over High-Speed Networks", *IEEE Transactions on Computers*, Vol. 47(4), pp. 391-413.
- Hadjiconstantinou, E. and Christofides, N. (1999) "An Efficient Implementation of an Algorithm for Finding K Shortest Simple Paths", *Networks*, Vol. 34(2), pp. 88-101.
- Habbal, M., Koutsopoulos, H.N., and Lerman, S. (1994) "A Decomposition Algorithm for the all Pairs Shortest Path Problem on Massively Parallel Computer Architectures", *Transportation Science*, Vol. 28. No.4, pp. 292-308.
- IVHS America 1992. Strategic Plan for Intelligent Vehicle Highway Systems in The United States.
- Koncz, N., Greenfeld, J., and Mouskos, K. (1996) "Strategy for Solving Static Multiple-optimal Path Transit Network Problems", *Journal of Transportation Engineering-ASCE*, Vol. 122, No. 3, pp. 218-225.
- Langsford, A., J.D. Moffett. (1993) *Distributed Systems Management*, Addison-Wesley.
- Lee, J. H., Lee, B. H., and Choi, M. H. (1998) "A Real-Time Traffic Control Scheme of Multiple AGV Systems for Collision Free Minimum Time Motion: A Routing Table Approach", *IEEE Transactions on System, Man, and Cybernetics - Part A*, Vol. 28(3), pp. 347-358.

- Mahmassani, H.S., Hu, T.Y., Peeta, S. and Ziliaskopoulos, A., Development and Testing of Dynamic Traffic Assignment and Simulation Procedures for ATIS/ATMS Applications, Technical Report DTFH61-90-C-00074-FG, Center for Transportation Research, The University of Texas at Austin.
- Mowbray, T. J., and Ruh, W. A. (1997) Inside CORBA: Distributed Object Standards and Applications, Addison-Wesley.
- Object Management Group, CORBA Website. <http://www.corba.org/>
- Orfali, R., Harkey, D., and Edwards, J. (1996) The Essential Client/Server Survival Guide. 2nd edition, John Wiley & Sons, Inc., pp. 676.
- Peng, Z. R., and Huang, R. (2000) "Design and development of interactive trip planning for web-based transit information systems", *Transportation Research Part C*, Vol. 8, pp. 409-425.
- Siegel, J. (1996) CORBA Fundamentals and Programming, John Wiley & Sons, Inc.
- Sheffi, Y. (1985) Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods, Prentice-Hall, Englewood Cliffs.
- Shier, D. R. (1979) "On Algorithms for Finding the K Shortest Paths in a Network", *Networks*, Vol. 9, pp. 195-214.
- Solanki R. S., Gorti J. K., and Southworth F. (1998) "Using Decomposition in Large-Scale Highway Network Design with a Quasi-Optimization Heuristic", *Transportation Research Part B*, Vol. 32(2), pp. 127- 140.

- Sun Microsystems (2000) Enterprise JavaBeans Specification, Version 2.0.
- Tang, C. and Abdel-Malek L. L. (1996) “A Framework for Hierarchical Interactive Generation of Cellular Layout”, *International Journal of Production Research*, Vol. 38(8), pp. 2133-2162.
- Thill, J. C. (2000) “Geographic Information Systems for Transportation in Perspective”, *Transportation Research Part C*, Vol. 8, pp. 3-12.
- U.S. Department of Transportation (1999) “ITS Mission Definition”, Federal Highway Administration, Washington, DC.
- U.S. Department of Transportation (1999) “National ITS Architecture: Theory of Operations,” Federal Highway Administration, Washington, DC.
- White, C. E., Bernstein D, and Kornhauser A. L. (2000) “Some map matching algorithms for personal navigation assistants”, *Transportation Research Part C*, Vol. 8, pp. 91-108.
- Ziliaskopulos, A. K., and Waller, S. T. (2000) “An Internet-based Geographic Information System that Integrates Data, Models and Users for Transportation Applications”, *Transportation Research Part C*, Vol. 8, pp. 427-444.