
從增值服務網路的觀點看國家資訊基礎建設

Perspective of Service Networks on National Information Infrastructure

連耀南

摘要

要邁入資訊化社會的境界有賴於國家資訊基礎建設以及增值服務網路的建立。硬體建設容易，軟體建設難。本文分析建立增值服務網路的困難，建議一個開放式架構提供快速更換及建立傳輸網路或服務網路的能力，並介紹一個新的動態網路計算模式，智慧型訊息模式 (Intelligent Messaging Paradigm)，以及相關的重要研發方向。

關鍵詞

個人數位助理
(Personal Digital Assistant; PDA)
個人通訊網路 (Personal Communications Network; PCN)
國家資訊基礎建設 (National Information Infrastructure; NII)
行動計算 (Mobile Computing)
行動電腦 (Mobile Computer)
網路精靈 (Intelligent Message, Intelligent Agent)
增值服務 (Information Service)
Telescript (Operation, Administration, and Maintenance; OA&M)

1. 前言

自從美國副總統 Al Gore 上任以來極力推動 NII (National Information Infrastructure)，全世界有志於發展資訊工業或是運用資訊的國家莫不傾全力跟著發展自己的 NII。我國自也不例外，所有跟資訊有關的產官學研究單位都投入可觀的人力與物力於 NII 有關的研究，即所謂的「國家資訊基礎建設」。目前各方注意的焦點多集中於 NII 的實體架構上，包括資訊高速公路、無線通訊技術、行動電腦、以及資料庫的建立。當這些技術都克服、網路都建好之後，尚需解決一個重要的問題才能徹底發揮 NII 的功能，那就是增值服務的建立。根據過去在通訊界服務的幾年經驗，筆者以為 NII 除了必須提供實體建設之外，還須提供一個好的開發環境給服務提供者，以使增值服務能很輕易的建立。當然，服務提供者必須去尋找能吸引顧客的增值服務，並儘快上市提供給顧客。這兩者皆非易事。在 NII 推行的初期，很可能只有一些免費或是簡單的增值服務可以成功的建立及運轉。

1.1 增值服務網路在資訊化社會中的運用

下面是一個資訊社會中的一個假想的例子：高速公路上發生了一起連環車禍，在現場的一位駕駛人正好有一部具有多媒體設備 (Multimedia) 的 PDA (Personal Digital Assistant)。這位駕駛人立刻用 PDA 上的無線電話功能撥 119 緊急求救，並用 PDA 上所附的照相機把現場情況傳給 119。PDA 上的 GPS (Global Position System) 自動定位系統也自動把所在位置報給 119。119 一看發覺這場車禍不小，立刻成立指揮中心協調救難行動。指揮中心透過無線網路呼叫警車及救護車前往車禍現場搶救。根據現場情況，另外調派消防車攜帶特殊裝備以便快速的分解糾結在一起的車體以搶救困在車中的駕駛人。指揮中心並透過醫療資訊網路瞭解附近大醫院現有的急救容量以安排並指揮各救護車分送傷患至適當的醫院。到達現場的救護車一面急救，一面把傷患隨身的 IC 身份卡插進電腦以獲得傷

患的基本資料，並透過網路傳到救護車上以及醫院的電腦。如發現傷患的血型特殊而且需要輸血，立刻透過醫療網路向特殊血庫請求支援。負責急救的醫師在救護車還沒到達之前就可以經由救護車上的影像設備檢查傷患並指揮急救行動。如需專門醫師會診也可透過資訊高速公路進行。在各方配合之下，很多寶貴的生命可因及時的急救及必要的支援而得以保存。這是我們在未來的生活中很可能會發生的實際狀況。

2. 建立增值服務網路的困難

2.1 賺錢的增值服務不易發掘

除了上節所描繪的遠景之外，NII的鼓吹者在各種媒體上列舉了許多想像的增值服務為全人類描繪了一幅美麗的新世界。事實上，這些增值服務不見得能完全實現，就以目前為各方所看好的「電影點播」(Movie On Demand)而言，它不盡然能為投資者賺進大把的鈔票，筆者在此試以投資者的眼光分別從電話網路及有線電視網路的觀點來盤算一下「電影點播」的投資與報酬。「電影點播」系統中最重要的兩項資源是傳輸網路的頻寬以及視訊流(Video Stream)。在一定的成本下，各用戶所需負擔的費用取決於一起分享資源的人數。如用電話網路傳送影片，因各視訊流係由大量用戶同時分享，各用戶所需負擔之資源主要在於電話頻寬。有線電視網路雖可大為降低頻寬的費用，但一起分享視訊流的用戶卻大為減少，以致大幅提高各用戶所需分擔之視訊流費用。我們先從電話網路的觀點來看，假設一部壓縮過的電影在網路上傳送需耗用3 Mbytes/sec.的頻寬，則傳送一部兩小時的電影共需傳送21.6 GBytes的資料量。以目前國內市內電話五分鐘一塊錢可以傳9600 bps的費率來計算，看一部電影光在資料傳輸上就需花費7500元台幣左右。通訊費用勢必下降數百倍才能與錄影帶出租店競爭，這不太可能在短期內純靠通訊技術的進步可以達成的。如用有線電視網路來傳送影片，因為同時看同一部影片的用戶大為減少，因而由視訊服務

站取得視訊的成本必須由少數用戶來分擔。最壞的情形下，每一用戶各佔用一個視訊流，而必須負擔整個資訊流的費用，這項費用可能跟租用一部PC所需費用相去不遠，算一算可能還是划不來。在成本結構尚未大幅降低之前，Enhanced Pay-Per-View可能是較有競爭力之方式，在這種系統中，有線電視只在固定但是很密集的時段(例如五至十分鐘的間隔)播出正在流行的影片，觀眾只能在固定的時段就播出的影片中挑選，而非隨時可以看任一部影片。如此，分擔每一視訊流之用戶大為增加，因而可以大幅降低各用戶所需負擔的費用。

再以一般的電話服務來看，一個電話交換機所提供的服務機能(feature)，例如「話中插播」及「轉接」，數以百計，但大多數用戶用都到不到十個以上的服務機能。說來難以相信，某一家交換機製造商負責開發服務機能的部門在一項非正式的調查中發現其員工家中有訂「話中插播」的只有三分之一左右，更何況一般的用戶。話中插播是一個簡單而有用的服務機能，而它的使用率仍然這麼低，其它機能的使用率想必就更低了。

目前全球各大公司對於高風險的長期投資日趨謹慎，偏好風險低回收快的短期投資。由這些例子看來，在推展NII的初期，各大公司可能不會貿然投下鉅資去開發新的增值服務，因而很可能是一些免費或很簡單的增值服務有機會活躍在網路上，例如政府機構的資訊、免費的公共資訊、以及email、Internet News、Gopher、WWW(World-Wide Web)等。使用者願意花大錢而不需投資大量資源的服務也將會有機會脫穎而出，例如醫療用的影像傳輸、提供股票市場資訊、以及有助於升學考試的增值服務等。

2.2 複雜的增值服務不易開發

一個增值服務能否成功還得看它的開發成本以及其困難度。近年來由於個人電腦軟體的蓬勃發展使得很多人誤以為設計一個增值服務軟體是輕而易舉的事，或低估了其困難度。其實在公共網路上開發一個新的增值服務並非易事，其難度遠超乎一般人的想像。以電話網路為例，集多年

的經驗、大批優秀的工程師、及還算不錯的設計流程，要在現有的架構上增加一個小服務機能，動輒以百萬美元計，而且常需一兩年以上的開發時間。試以蜂巢式電話為例，在國內每一通電話皆以國內最長的長途電話收費，為何不能以通話距離來計算呢？其實不難理解此種不合理的收費方式，因為即使一個簡單的計費軟體在網路上也不是容易開發的。在大多數電話網路中（例如美國的網路），長途和市內電話是分開計費的，長途電話部份自有長途電話公司去算賬，蜂巢式電話公司只需計次或計時即可。這樣的系統引進國內時，其收費系統勢須作大幅度的改變。在國內這種賣方獨佔並且供需極度失調的情況下，賣方怎會花大錢去開發使他收益會大幅降低的收費軟體呢？一律以最長的長途電話來收費，既省事又賺錢，反正沒人競爭，何苦自找麻煩呢？

近年來智慧型網路的發展其目的是在改善系統的軟體架構以加速服務機能的開發，CCITT 及 BellCore 定下了一個理論上極為漂亮的 Service Creation Environment (SCE) 架構[1]，要讓服務設計者用內建的基本功能模組(Service Independent Building Blocks) 來組合新的服務機能，但截至目前為止所有的 SCE 都只能紙上談兵設計一些以查表為主的簡單服務機能，尚未有任何較為複雜的服務機能是用 SCE 設計出來的。單以服務邏輯(service logic) 的開發而言，例如呼叫處理(Call Processing)，目前 SCE 的作法多半是以 Finite State Machine (FSM) 為架構設計一個服務的邏輯。SCE 所提供者不過是一個 GUI 畫面，讓使用者很方便的畫出 State Diagram 進而自動產生程式。所省者不過是一點寫程式的時間而已，對整體的設計工作並無太大的幫助。一個通訊系統的服務邏輯最難處理的部份，像是機能衝突 (Feature Interaction Problem)[2]，多半要在系統工程階段處理好，而非留到寫程式的階段才來解決。此外，除了呼叫處理之外，OA&M 所耗費的人力更為可觀。總而言之，SCE 尚未發揮它預期的功效。

近幾年前，AT&T 原想花五年時間投資七十

億美元改善它的長途電話網路軟硬體架構力圖與 MCI 在開發服務機能的速度上一較長短，可是就在花了兩三年時間連系統工程都尚未完成之前便知難而退。由這些例子看來，要開發一個網路上的加值服務絕非易事。

為了減低在 NII 上開發加值服務的困難，筆者認為「服務網路整體架構設計」、「服務邏輯設計環境」、以及「網路執行環境」，是幾個很重要的研發題目。以下各節分別對各個題目作簡單的探討。

3 · 整體服務網路架構

類似於智慧型通訊網路的設計目標，服務網路整體架構設計之主要目標是要在不影響網路效能之下提供一個架構以使各種加值服務能在其邏輯設計好之後很輕易的加進網路。（所不同者，智慧型網路是為通訊網路提供連接服務，而本文所談的加值服務則是一般的資訊服務，兩者的研究方向並不相同。）

3.1 架構設計原則

從加值服務的角度來看，我們認為 NII 整體架構的設計必須符合下面幾個原則才能提供必要的彈性給各式各樣的加值服務設計者：

原則一：實體的傳輸網路必須與邏輯網路（服務網路）分離。如此，傳輸網路可隨時汰舊換新而不致影響服務網路的設計。其實，服務網路本身也應採用分層分工的方式以便每一階層都可變動其設計而不會影響到其他層的設計。

原則二：採用開放式的架構。新的加值服務可隨時加入服務網而不需作大幅度的調整，新的傳輸網路也可隨時加入，而且一個服務網路可以透過不同的傳輸網路提供服務。例如，一個資料庫可以透過 Internet、或電話網路、或有線電視網來提供服務，而不需限定於某一種傳輸網路。

原則三：採用分散式管理，盡量避免中央集中管理的需求。一般而言，中央集中式的管理較有效的運用資源，也有較高的可靠度及服務品質，可是比較沒有彈性，任何變動經常需時甚

久。相反的，分散式的管理則能擁有很大的彈性，但是較不易達到有效的資源管理，也較不易保證服務的品質。茲舉電話網路及 Internet 為例，電話網路採用集中式管理所以非常可靠，但要設計一個新的服務卻不免過於昂貴而且需時甚久。而Internet 採用分散式管理則彈性非常大，新的服務要加入 Internet 非常容易，不需經過冗長的申請手續，但是所提供的及時性 (real-time) 服務像 Telnet 則無法達到像電話網路那麼高的完成率

(Call Completion Rate) 或是所謂的服務品質(Quality of Service)。在開放式架構之下，各服務網本身可以自成一個獨立網路，而可以根據自己的需要去定義自己的可靠度、彈性，以及服務的品質。可靠度要求不高的服務便可縮短開發時間，盡快加入服務網。

3.2 一個開放性架構

本架構是由一群服務站及兩種獨立的網路，服務網路及傳輸網路所構成的。服務站 (Server)

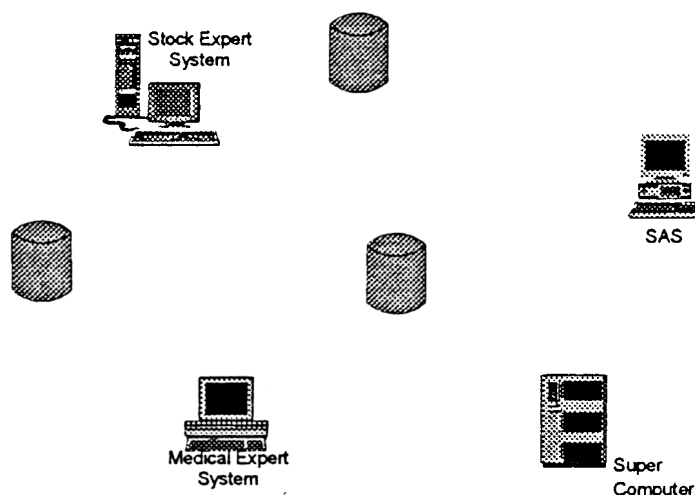


圖 3.2-1 服務站

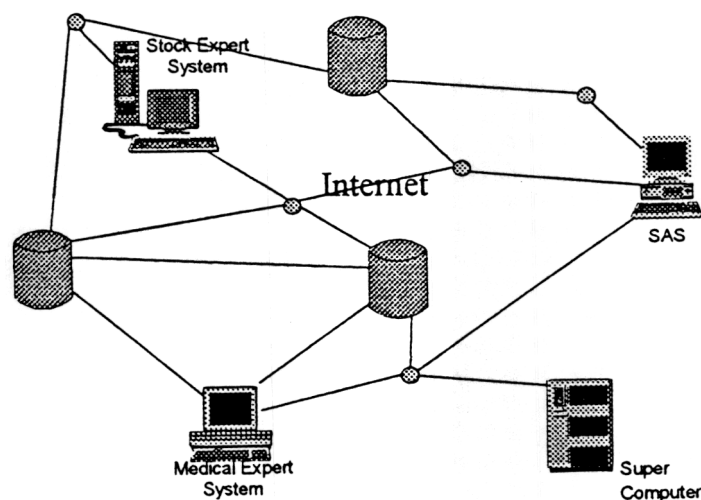


圖 3.2-2 服務站經由Internet互相連結

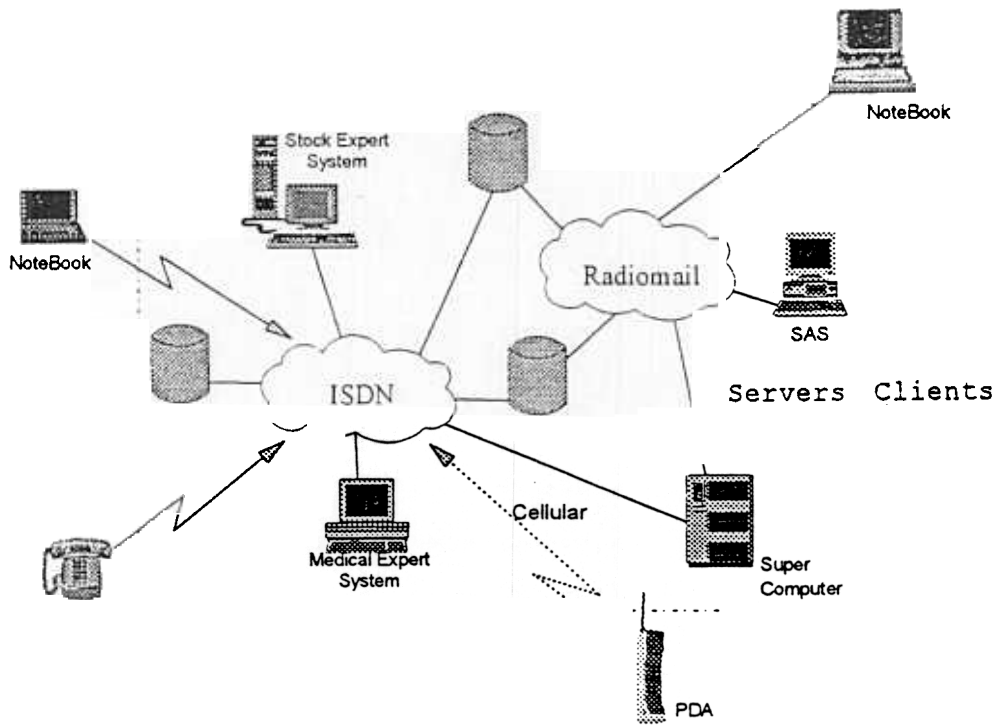


圖 3.2-3 服務站經由其它網路互相連結

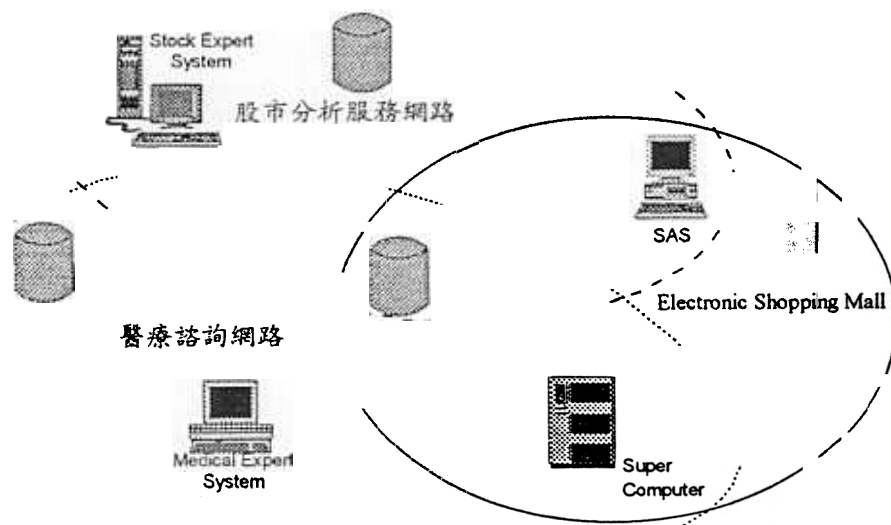
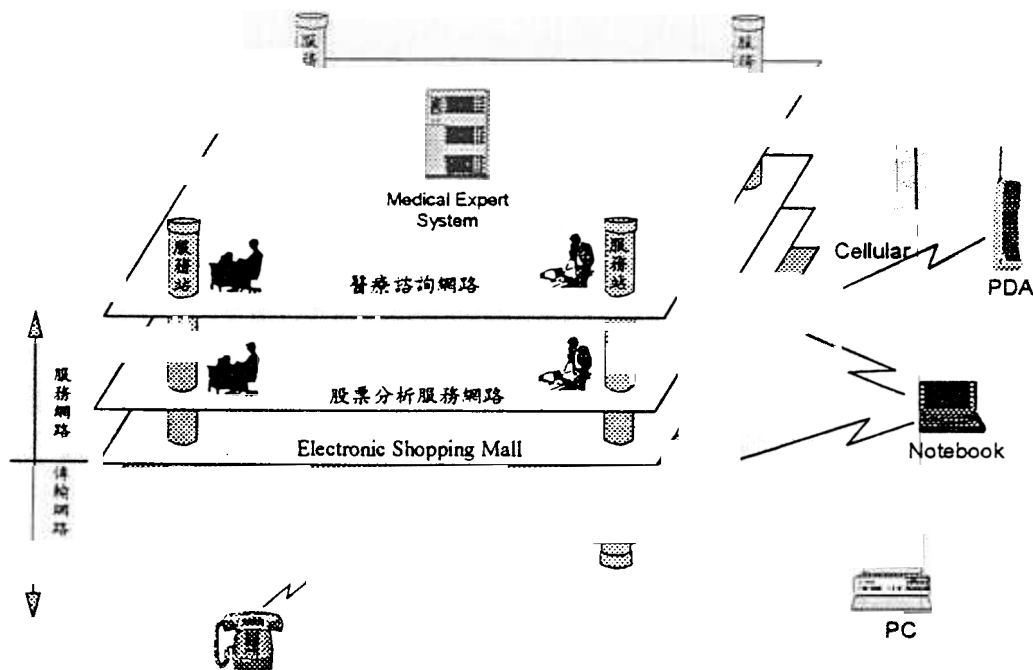


圖 3.2-4 各服務站可自由加盟不同的服務網路



提供基本的資訊或計算服務，例如資料庫、超級電腦、股市專家系統、醫療專家系統等。一個服務網路可以聯合任意數目的服務站提供某一類的服務，例如醫療諮詢服務網路、股市分析服務網路、考前加油服務網路等。服務網路的數目沒有一定的限制，一個服務站也可以同時加盟好幾個不同的服務網路，每一個服務網路提供服務給它的用戶(client)，一個用戶使用終端機經由傳輸網路連到服務網路上，傳輸網路提供服務站之間以及附屬的終端機間的通訊。一個傳輸網路可以是Internet、可以是電話網路、可以是數據網路、也可以是像 Ardis 一樣的無線網路。如果是Internet，它的終端機很可能是一部電腦。如果是電話網路，則它的終端機就可能是普通電話、無線電話、或是PDA。每一個服務站可以連接好幾個不同的傳輸網路，一個用戶可以經由不同的傳輸網路跟服務網路溝通。例如，一個用戶用他的PDA經由無線電話在車上送一個電子信(message)進入一個旅行訂位服務網路去要求訂位服務，回辦公室之後再用電腦經由Internet去收回

結果。如此，他經由不同的傳輸網路在不同的時間不同的地點用不同的終端機去辦一件事情。這個架構請參考附圖3.2-1至圖3.2-5。

服務網路的分開考慮對我們的設計工作幫助極大，因為現有的傳輸網路，例如公眾數據網路、無線網路、或是Internet等，都可派上用場，不必另起爐灶。而且終端機的行動能力(mobility)不是一個簡單的問題，就讓傳統的傳輸網路業者去傷腦筋。我們只需考慮用戶在服務網路上的行動能力即可。

4 · 服務邏輯開發環境

4.1 計算模式

在計算理論方面，筆者建議仍然採用傳統的計算模式，頂多容許平行計算(parallel computing)而避免用分散式計算，因為非常複雜的Clock Synchronization, Byzantine Agreement, Termination Detection, Mutual Exclusion, Leader Election, Deadlock Detection[3], 等問題都還停留在理

論階段，以目前的軟體工程技術是無法承擔這樣複雜的控制程式的開發。雖然一個服務站本身可能可以由一個區域網路上的分散式系統來擔任，但是從服務網路的觀點來看，那還是一個單純的服務站，整個服務網路本身要由一個分散式系統來提供整體性服務還是不太實際的。至於偏向人工智慧式的 Intelligent Agent 則比分散式計算更難實現。筆者認為還是實際一些比較恰當。

4.2 服務邏輯的抽象模式

最近軟體工程界正在提倡 Middleware 介於應用程式及基本軟體模組之間，可以加速應用程式的發展。這 Middleware 要如何設計，眾說紛云並沒有定見。筆者在此提出一個看法。回想過去數十年間計算機能力的運用得以大幅提昇泰半依賴高階語言的發展，由組合語言而 FORTRAN，C 進而至 Object Oriented Languages 等。更貼切的說法應該是說愈高階的語言提供更高階的抽象模式給使用者。例如 UNIX 作業系統把一個資訊系統看成檔案 (files) 以及處理檔案的許多工具。這些工具又可以很快的組合成新的工具來幫使用者做事，這個簡單的模式大幅提高了大家運用資訊系統的能力。再以資料庫為例，Edgar F. Codd 提出的 Relational Model 把以往複雜的資料庫組織成一些 Tables 而大幅提昇使用者運用資料庫的能力。因此，如要好好運用網路上的資源，我們必須提供適當的抽象模式給服務邏輯的設計者。例如說，我們可以把整個資訊網路看成一個 Active Database，可能是 Relational，也可能是 Object Oriented，使得設計服務就像設計 Active Database 的應用程式一樣容易。對很多簡單的服務而言，這可能已經足夠了。筆者必須在此提醒有志於從事這方面研究的先進，在考慮一個抽象模式的時候必須特別小心，以免建立在一些過於樂觀的假設上。過去有些領域對於計算能力的進步作了太過樂觀的假設，以至於提出的觀念不易或無法實現，因而浪費了許多資源，走了許多冤枉路。我們提出的抽象模式必須能在適當時間內得以實現。因限於時間及人力，我們尚未在這個方向作很深入研究。

5 · 一個動態的網路計算模式

NII 的建立與行動式計算環境的蓬勃發展使得將來的資訊使用者可以在任何地方任何時間利用有線或無線的通訊技術去運用網路上的資源。在這樣一個動態的環境下去運用網路的資源需要有不同的網路計算模式。在行動計算環境中，由於無線通訊的非持續性，一個使用者不會持續的連接在網路上，因此一個加值服務不能要求使用者持續連接在網路上一段相當長的時間。以目前相當風行的 World-Wide Web (WWW) 而言，我們如果用傳統的 Client-Server 模式去搜尋資訊，像是 Mosaic 一般，就非常不適合在行動計算環境中使用。Mosaic 的使用會以 Client 為中心向外送出並接收很多訊息 (message) 以獲得網路上的資訊。這樣子不但傳輸的費用太貴，而且使用者在行動計算環境中也不太容易長時間的連在網路上。因此我們需要有一個動態的網路計算模式來支持這樣的計算環境。（註：行動計算環境是在強調用戶在無線網路中的移動，而本節所指的動態環境範圍比較廣泛，如果一個用戶在不同的時間在不同的服務站作同一件事，雖然不是透過無線網路來作，也可以涵蓋。）

5.1 智慧型訊息模式 (Intelligent Messaging Paradigm)

針對這樣的環境，近年來有一套新的計算模式叫做 Intelligent Messaging Paradigm（暫譯：智慧型訊息模式）剛被提出來以解決這種問題。簡單的說，使用者把每一個帶有電腦程式的電子訊息稱為「網路精靈」(Intelligent Message)，送到網路上去執行，這訊息所帶的程式可以在網路上的服務站被執行，而且可以要求轉到另一個服務站去作下一階段的工作。在這個模式下工作，使用者在送出網路精靈之後不必持續的連在網路上，網路精靈本身在完成任務之後會自動把結果帶回使用者手中。下面的程式是一個假想的網路精靈被送到一個旅行訂位服務網路中要求定飛機票，旅館，及作租車安排。

1. Global \$DATE = 8/1/94
2. GOTO reserve@airlineA.com
3. Flight = RESERVE first class from New York to Chicago on \$DATE after 12PM
4. GOTO reserve@chicago.avis.com
5. Rental_car = RESERVE 1 midsize for 2 days on \$DATE at Flight.arrive + 00:30
6. GOTO reserve@chicago.hilton.com
7. Lodging = RESERVE single room for 1 night on \$DATE at Flight.arrive + 02:00
8. GOTO \$HOME
9. SAVE Flight, Lodging, Rental_car into Trip
10. DISPLAY Trip

指令 1 設定旅行日期

指令 2 要求進入某航空公司 (AirlineA) 的定位系統

指令 3 要求定八月一日中午以後從紐約到芝加哥的頭等艙機位

指令 4 要求進入 AVIS 租車公司的預約系統

指令 5 要求預定一部中型車，取車時間為飛機到達後半小時，使用期兩天

指令 6 要求進入芝加哥希爾頓旅館的定房系統
(事實上全美各地的希爾頓可能會共用一個全國性或全球性的定位系統，所有的訂位要求都被送到同一個地方去，使用者可能完全不知道)

指令 7 要求定一個晚上的單人房，到達時間為飛機到達後兩小時

指令 8 要求回到客戶的終端機 (PC 或 PDA 等)

指令 9 要求把結果存檔

指令 10 要求顯示結果

跟一般的 remote batch execution 的程式看起來是差不多，可是上面這個程式不會固定在一個服務站執行，而會被輾轉送到不同的服務站去執行。這種模式在人工智慧的領域中也已經被討論了很長的一段時間，最常見的名稱為 "Intelligent Agent" 或是 "Knowledge Robot"，其重點在盡情發揮一個 Intelligent Agent 的潛力在網路上擷取資訊，並且作人工智慧式的決策。而在 Internet 中也

有所謂的 "Software Robot" 可以在網路中自動作一些簡單的搜尋與計算。在 1994 年七月份的 Communications of ACM 上有專文介紹 [5][6]。在分散式計算的領域上，類似的觀念也不少，例如 "Remote Programming"。但其實用性仍然侷限於小區域的網路上，在廣域網路上尚未達實用的階段。有鑑於大型軟體系統開發不易，我們所討論的網路精靈所具有的計算能力跟一般的傳統的程式一樣。如果將來人工智慧或分散式計算的技術成熟了，我們再來作更進一步的引進。其實如果從基本的計算理論來看，網路精靈的計算能力跟一般程式的計算能力並沒兩樣，同是出於程式人員的設計，只不過一個是在固定的電腦上執行，而另外一個是在網路上變換執行的電腦而已。不過，計算理論雖然一樣，在實現的過程中，卻有一些問題需要特別考慮。在傳統的計算環境中，使用者可以很方便的掌握一個程式的執行情況，但是在智慧型訊息模式中，一個程式即網路精靈會在不同的電腦上執行，它的執行情況及行蹤就需要費事去找尋。再者，當一個正在執行的程式需要跟使用者溝通的時候，或者要回報執行的結果卻找不到原來的主人，也是挺麻煩的。更麻煩的是，如何執行一個跨服務站的交易 (Transaction)？如果智慧型訊息模式要達到實用的階段，這些問題都必須一一克服。

5.2 Telescript Programming Language

自從個人數位助理 (PDA) 的風潮在 Apple Computer Inc. 發表的 Newton 帶動之下，各大廠商或多或少都注意到智慧型訊息模式的重要。其中 General Magic 發表了一套語言叫做 Telescript 以求智慧型精靈訊息模式的實現。很多 PDA 廠商競相加入 General Magic 聯盟以引用 Telescript 在他們的產品中。由於僧多粥少，只有少數大廠能獲得 General Magic 優先接納，其他小廠只有眼睜睜的看著眾家大廠搶盡先機，並期望 General Magic 聯盟能盡快擴大陣營來接納眾家小廠以免落後太多。

如今之計，我們應該全力投入這方面的研發，提高我們的競爭力，一方面避免任人宰割，

一方面或許可以先期掌握關鍵技術來宰制他人。下節即在嘗試對智慧型訊息模式作一個簡單的探討，並討論可能需要解決的OA&M (Operation, Administration, and Maintenance) 問題。

5.3 尚待解決的問題

無論是用哪一種模式，一個服務網路要做到安全可靠必須要有很強的OA&M，否則只能算是一個實驗室的產物而已。OA&M的問題千頭萬緒極為複雜。本節列舉一些跟異常有關係的問題，希望有興趣的先進一起來解決。我們按 Telescript 的習慣把一個網路精靈稱為一個 agent。

1. 如何知道一個進行中的 agent 的所在位置？
2. 如何知道一個進行中的 agent 的狀態 (status)？
3. 如何控制一個進行中的 agent？
4. 如何知道一個 agent 的執行過程？
5. 如何支援交易 (transactions)？
6. 如果一個 agent 可以產生另外一個 agent，那整個計算模式會如何改變？
7. 如何把結果收回來？
8. 如何處理執行中的異常現象？例如：
 - (a) 程式有錯
 - (b) 正在執行一個 agent 的服務站當機
 - (c) 要進入的服務站當機
 - (d) 原來送出 agent 的客戶不見了
(例如電池沒電了)

當然，實際上的問題還有很多，有待大家共同努力來一一克服。

6 · 結語

國家資訊基礎建設的工作正在如火如荼的展開，它對整個社會的影響可能不下於第二次工業革命對於人類的影響。資訊工業的發展到目前為止是對人類社會產生了極大的貢獻，可是很多資訊運用還是比較局部性的，對人類的生活方式及做事方式並沒有多大的影響，只不過很多瑣碎的小事因為電腦的幫助而作起來比較有效率而已。資訊科學家的夢想，要對人類的做事及生活方式做出結構性的改變，仍然還有一段距離。過去的

一些理想，像人工智慧，已經不再被寄予厚望，而高速電腦和分散式計算只在提供計算平台而已。國家資訊基礎建設的推行將是把資訊科技盡情發揮以對人類的生活方式作結構性改變的良機。我們從Internet 在這一兩年的急速擴張，便可看出端倪。本文嘗試指出可能的困難以及應該努力的方向，並建議一個簡單的開放性架構，希望各種增值服務能夠很快很順利的加進去。但願不久的將來大家的期望就可以實現。

計畫相關資訊

本文係工研院電通所執行經濟部委託之個人數位助理 (35N7200) 計畫成果之一。

感謝

本文承電通所軟體技術組同仁劉沅、李鎮樟、李淑慧及蕭信煌斧正，呂豐昌繪圖，特此致謝。

參考文獻

1. J. Garrahan, P. Russo, K. Kitami, and R. Kung, "Intelligent Network Overview," IEEE Communications, pp. 30-36, March 1993.
2. E. Cameron, et al, "A Feature-Interaction Benchmark for IN and Beyond," IEEE Communications, pp. 64-69, March 1993.
3. Michel Raynal, "Distributed Algorithms and Protocols," John Wiley & Sons, 1988.
4. P. Maes, "Agents that reduce work and information overload," CACM, pp. 30-41, July 1994.
5. Michael Genesereth and Steven Ketchpel, "Software Agents," CACM, pp. 48-53, July 1994.
6. James E. White, "Telescript Technology: The Foundation for the Electronic Marketplace," General Magic, Inc.
7. M. Weiser, "The computer for the 21st century," Scientific America, pp. 94-104, 1992.

作者簡介：

連耀南



工研院電通所電腦軟體技術組副組長，普渡大學電機工程博士，曾任美國俄亥俄州立大學資訊科學系助教授及貝爾實驗室研究員。專長於資料庫管理系統，分散式計算，電腦軟硬體系統架構及通訊系統。曾經參加過交換機系統軟體架構改造及智慧型網路架構設計等計畫。