行政院國家科學委員會專題研究計畫成果報告

網路醫學資訊系統之可擴充性與容錯機制

Fault Tolerance and Scalability for Network Medical Information

計畫編號：NSC 89-2219-E-002-003
執行期限：88 年 8 月 1 日至 89 年 7 月 31 日

主持人：郭斯彥　台灣大學電機工程學系
sykuo@cc.ee.ntu.edu.tw

一、中文摘要

本計畫「網路醫學資訊系統」讓醫療服務得以藉由無遠弗屆的網路系統遍佈到世界各地，來協助診療的進行。換句話說，這就是「網路醫院」的具體實現。而在實作上，我們採用分散式物件技術來架構網路醫學資訊系統，除了能增加系統服務效率之外，也將有助於系統的擴充與維護。我們以 Microsoft 的 DCOM 技術為基礎，在系統的實作過程中進一步探討分散式物件在寬頻網路上的各項議題，如服務效率、系統可靠度、容錯機制等等。

關鍵詞：寬頻網路、醫療服務、分散式物件技術、可靠度、可擴充性、容錯機制

Abstract

This project, a network medical information system will make medical treatment services ubiquitous through the widely-spread network systems in the world. In other word, this is the concrete realization of a “networked hospital”. We are going to build the network medical information system by the progressing distributed object technology that will increase system efficiency and be of great help in system extensibility and maintenance.

Our implementation is based on the Microsoft DCOM technology. It is expected that there will be many research issues to be discussed during the development of a distributed object system over the broadband network, such as service efficiency, system reliability, and fault tolerance mechanisms.

Keywords: broadband network, medical treatment service, distributed object technology, reliability, scalability, fault tolerance mechanism.

二、緣由與目的

隨著網際網路以及有線、無線通訊技術的不斷發展與進步，網路使用者的數量與日俱增，架構在網際網路上的資訊服務項目也愈來愈多。為了讓廣大的社會民眾擁有更便利、更快速的醫療服務，「網路醫院」的概念於是誕生。有了網路醫院，醫療服務的提供不再只侷限於醫院本身。透過 Internet，病人可以隨時上網掛號候診，醫生也可立即對病人進行診斷，同時透過網路取得病人的病歷或 X 光片等相關資料以協助診療。除此之外，網路醫療資訊系統也具有教育的功能。醫生可以隨時隨地取得最新的醫療資訊，專業知識由此迅速傳遞，提供了一個便捷的進修管道。對於一般大眾，則也可從中獲得許多有用的醫學常識。

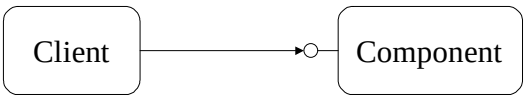
本計畫所將建立的「新世代網際網路上醫學資訊系統」與現有的網路醫院比較起來，二者有許多特性是不一樣的。第一點，本計畫強調系統服務提供的即時性，病人可以透過網路直接與醫生進行互動，而身處各地的醫師們則直接在線上進行會診。第二點，本計畫的目的是希望整合全民的健康資訊，建立完整的資料庫系統，並且實作快速有效的系統可擴充機制，以提供高效能的整合性醫療服務。第三點，本計

畫的實作將採用新興的分散式物件技術，能夠使得資訊系統的發展與維護更具有彈性，而這正是本子計畫的最終目標。

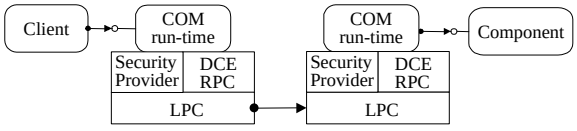
為了達到這個目標，我們仔細研讀了 COM/DCOM、CORBA 等分散式物件技術的規格以及其運作方式，並從中獲得了各項技術的關鍵知識。以下繼續針對分散式物件技術的原理，以 COM/DCOM 為實作架構來詳細說明。

COM 代表 Component Object Model。它定義了元件及其客戶端之間的互動。這個定義使得客戶端與元件之間的連接不需要任何居間的系統元件；客戶端可無額外負擔地呼叫 (call) 元件所定義的方法 (method)。圖一是 COM 元件在相同程序 (process) 的示意圖。

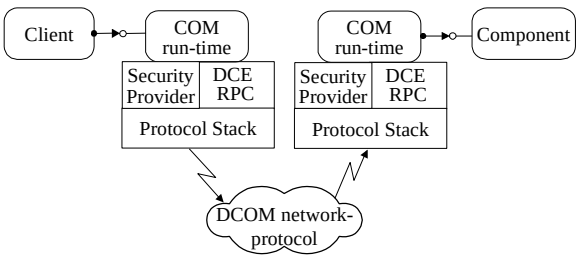
在現今的作業系統，不同的程序之間通常會彼此隔開，所以若某客戶端需要與在另一個程序中的物件進行溝通，則需要使用作業系統所提供的某種程序間(inter-process)溝通機制。COM 提供了一套這樣的溝通機制，使得客戶端完全不須知道其中是如何運作的：它把從客戶端往元件的呼叫攔截下來，再將這個呼叫繼續傳遞給在另一個程序的元件。圖二為上述說明的示意圖。



圖一：在相同程序的 COM 元件



圖二：不同程序中的 COM 元件



圖三：DCOM 在不同機器上的 COM 元件

以上我們說明了 COM 的運作機制，現在我們將 COM 延伸到分散式的環境，也就是所謂的 Distributed COM (DCOM)。當客戶端與元件不在同一台機器上的時候，DCOM 就以網路協定來把程序間的溝通機制給取代。客戶端與被呼叫的元件都將不會發覺它們兩者之間的距離變得遠了些。

圖三是 DCOM 的整體架構：COM run-time 對客戶及元件提供了物件導向的服務；而使用 RPC(Remote Procedure Call) 與 security provider 來產生適合於 DCOM 標準通訊協定的標準網路封包。

三、結果與討論

提供一個全方位的網路醫學資訊系統是項非常具挑戰性的工作，它必須是不會停下來的服務，否則將會對醫生或病人造成重大影響，甚至危害生命安全。另外效能的表現與可擴充性也是必須包括的，因為系統必須能夠隨著醫院及病人之增加，處理大量同時連線的客戶請求。原有的網路醫學資訊系統最缺乏的是在客戶端的發展，若是沒有改善並加強使用者介面，則與系統間的交談無法得到改進，這就限制網路醫學資訊系統的發展。

所以，我們將設計的主要目標設定在如何提供簡單且有效的存取介面，如何將目前最新的多層式分散式技術應用於醫學資訊系統。進而提供可擴充性及容錯機制，使得客戶與伺服器都能感到效能提升及可靠度提高。

分散式元件物件模型的特徵包括了介面與實作的徹底分離、支援物件具有多重介面、語言中立、即時二位元可執行碼的重複使用、元件位置透通性、可延伸架構、間接存取、版本管理及伺服器生命週期管理。也就是說以 DCOM 為發展分散式程式的平台，則研究者及研發者可以專心於對他們程式比較重要的課題，而不需將大部分的努力投資在建立支援的基礎建設。接下來，我們將具體說明 DCOM 的主要特點：透通性、更新版本及伺服生命週期。

〔一〕透通性

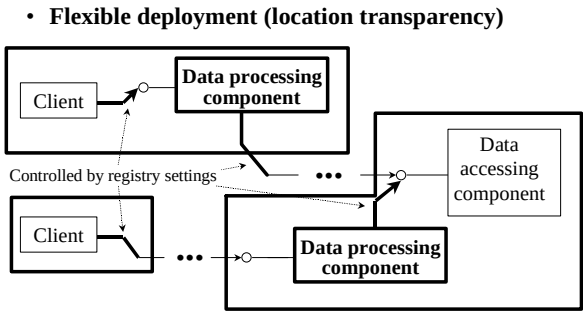
在啟動時，DCOM 同時支授非透通及透通模式。在非透通模式中，客戶端可以明確地指出伺服元件所在的位置。相反地若選擇透通模式，則讓 DCOM 自行詢問登錄資料庫來決定以上屬性。一旦機器名稱確定，DCOM 將會利用已存在的物件實體來完成任務。若是物件實體不存在，則由 DCOM 自動找出伺服器實作檔產生一個新的物件實體。為了要在方法呼叫上提供透通性，伺服器僅傳回物件實體的參照(reference)，它包裝了所有由客戶端到伺服物件之間所有的連線訊息，一般來說包含了 IP 位址、埠號，以及物件位址。當物件參照傳到客戶端時，客戶端的 DCOM 便會解開它，傳回介面的指標。當客戶端透過這指標呼叫時，這個呼叫只傳給所指的物件介面而不會打擾伺服器其他物件。其實這也就是間接存取的應用之一。圖四表示了元件位置的透通性。

〔二〕更新版本

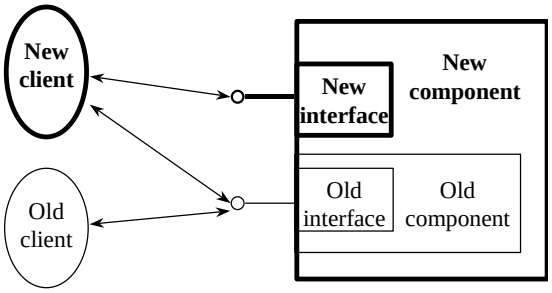
DCOM 的方法是基於下列三個需求。首先，任何原有的介面是不能被變更的，若要擴充則必須支援之前已存在的介面。最後，任何客戶必須利用詢問介面的方法與伺服器交談。如此才能允許客戶端與伺服器端獨立發展軟體。假設在特定的機器上，伺服器軟體在客戶端軟體更新之前更新，因為新的伺服器支援所有舊有的介面，舊客戶仍然可以保有所有的指標並正常工作。當客戶軟體更新時，新客戶將會詢問新的介面 ID 來享受新機能。相反地假如客戶端軟體先更新，則新客戶將試著詢問舊伺服器而得到失敗的結果。這個程序將使新客戶以舊有的機能來處理這個失誤，但不會使它當掉。圖五表示更新版本處理的架構。

（三）伺服生命週期的管理

DCOM 支援各種形態的管理。基本上，被啟動的伺服器產生一個類別工廠來管理所有支援的物件。物件實體隨著客戶端的請求而產生，參照計數是用來管理伺服器物件的生命週期。為避免不正常的結束，DCOM 提供了自動 pinging 的機制。



圖四：位置透明化



圖五：更新版本的處理

在物件參照解開後，客戶端開始送出週期性的心跳信號給伺服器。當客戶端斷線時，就停止送出心跳訊號，一段時間後若沒收到心跳，伺服器便會自動切斷連線。

本計畫的最終目的是建構一個具有高擴充性、高穩定性的通訊環境。為達此一目標，我們仔細研讀了 DCOM 等分散式物件技術的規格以及其運作方式，從中獲得了各項技術的關鍵知識。

我們將系統開發與實作的重心放在通訊協定的使用方面。包括：通訊管道的建立，通訊管道的維持，以及後續資料處理。

在通訊管道的建立方面，我們採用兩台串接在 100BaseT 的乙太網路上的個人電腦模擬真實通訊情況下的主從兩端。為顧及與相容性與擴充性，我們並不預設這兩台個人電腦何者為伺服端、何者為客戶端。如此一來在實際執行時不僅更具有彈性，同時也能顧及使用人員操作不當所造成的錯誤。

由於通訊管道是採用雙方互相協商的方式所建立的，因此通訊管道的維持就必須由雙方共同監視。我們預設一些錯誤情況的處理規則，例如錯誤重置、重複建立連線、以及意外斷線等。

後續資料處理此一部份的功能屬於系統中的應用程式層級，考慮到系統開發的時效性，以及本系統將來可能面臨的規格修改，我們在此一部份並未實作完整的人機界面程式，而以較為簡單的交談式界面、傳輸 ASCII 資料。完整的應用程式層級也將是未來開發的重點。

依據上述規範，大致可以規劃出系統藍圖。系統中通訊管道的兩端皆具備主動建立通訊管道、與被動告知通訊管道建立的能力。也就是說，雙方各自具備一資料傳送界面物件，以及一資料接收界面物件。在實際執行時，通訊管道可建立於以兩機器已執行之行程之間。程式行程開始時，資訊通訊界面並未立刻產生物件實體（Instance）。當雙方之一產生建立通訊界面之要求時，對方立即會在行程之中產生一對應的物件實體；在這個過程中 DCOM 即扮演通訊埠監視、訊息指派（Dispatch）與物件管理的角色。此外，我們採用了 DCOM 的一些先進功能，讓系統能夠動態載入程式而不必預先執行。若被連線端的程式行程尚未被執行，DCOM 本身即會要求系統建立行程。在行程建立以後，DCOM 便會把程式執行權限轉移給行程本身。

經由第一年度的實作經驗累積，在第二年度我們開始將所需的軟體製作成分散式軟體元件，並配合其它子計畫將軟體製作成分散式軟體元件，如此可以增加系統的可擴充性。在分散式的環境中，應該要盡量善用各個伺服器的資源，而不該將要求只送給某些伺服器而造成那些伺服器的負擔遠比其它伺服器高。我們可利用物件中介者(object broker)來管理物件的生成與分配，以達成伺服器的負載平衡。

而物件中介者也可提供其它子計畫的軟體元件集中註冊的機制。物件中介者必須定時偵測伺服器的狀態，因此客戶端就不會連結到無效的伺服器。因此，物件中介者提供了元件層次的容錯機制。加入上述功能後，我們可以明顯看出分散式物件技術可使得系統整合與維護更容易，以加速系統的開發，提高系統的可擴充性，使得系統的更新較為快速、穩定。

四、結果自評

我們預期此計畫的實作將可縮短學術研究和產業界在實作能力的差異。而參與之工作人員可獲得物件導向程式設計能力、分散式物件技術的實作能力，以及容錯應用能力。

除此之外，本計畫對於建立完整的醫療體系也將具有以下的貢獻：第一點，本計畫提供系統服務的即時性與穩定性。第二點，本計畫的最終目的是希望整合全民的健康資訊，建立完整的資料庫系統，並且實作快速有效的系統可擴充機制，以提供高效能的整合性醫療服務。第三點，本計畫的實作採用新興的分散式物件技術，能夠使得資訊系統的發展與維護更具有彈性，也因此計畫的具體實現，為發展其他網際網路上資訊系統服務奠定了穩固的基礎。

五、參考文獻

[1] Dale Rogerson, "*Inside COM*", Microsoft Press, 1996.

[2] Don Box, "*Essential COM*", Addison Wesley, 1998.

[3] Stanley Lippman, "*Inside the C++ Object Model*", Addison Wesley, 1996.

[4] Microsoft Corporation and Digital Equipment Corp., "*The Component Object Model Specification*".

[5] Y. Huang and C. Kintala, "*Software implemented fault tolerance: Technologies and experience*", in Proc. IEEE Fault-Tolerant Computing Symp., pp. 2-9, June 1993.

[6] E. Seligman and A. Beguelin, "*High-level fault tolerance in distributed programs.*" Tech. Rep. No. CMU-CS-94-223, Dept. of Computer Science, Carnegie Mellon University, 1994.

[7] B. Narendran, S. Rangarajan and S. Yajnik, "*Data Distribution Algorithms for Fault-Tolerant Load Balanced Web Access*", in Proc. of the IEEE Symposium on Reliable Distributed Systems, October 1997.