

植基於 Java/CORBA 物件式 Web 運算模式之研究： 以多媒體視訊物件之訊息交換為例¹²

李昇墩 林金木 張國盛 李天祥
國立高雄第一科技大學資訊管理系
高雄縣燕巢鄉大學路一號
Tel: (07) 6011000-4111
Fax: 07-6011042
Email: stli@ccms.nkfu.edu.tw

摘要

本研究探討如何引進橋接技術以建構新一代的網際網路應用系統，並以多媒體視訊物件之訊息交換為研究對象，討論在一實際系統發展過程中，所面臨的各項異質元件整合的困難與挑戰。在解決方案上，則以 Java 與 CORBA 為多層式主從架構之主軸，配合分散式物件導向的設計理念及委任式的使用者圖形介面整合成一物件式 Web 的分散式異質運算環境。根據實驗結果指出，在該環境下的物件可以有效地跨過電腦硬體平台、作業系統、程式語言甚至視覺訊息型態的藩籬而相戶溝通，並進而達成開放式系統之隨插即用型的目標。

關鍵詞：Java、CORBA、物件式 Web 運算、多媒體應用、異質環境運算

一、研究動機

隨著資訊科技之創新與多樣化的發展，提昇了電腦在我們日常生活中的重要性。舉例來說，功能強大的個人電腦的流行、訊號處理及多媒體技術的快速進步，將應用程式的發展推到了極致，諸如隨選視訊 (VOD)、視訊會議 (video conference) 甚至視覺化查詢法 (visual query) 的發展，

改變了我們日常的生活型態 [1]。在這之中，以結合視訊攝影機 (video camera) 所發展之保全監視服務 (surveillance service) 是一個相當有趣的應用，它的基本功能在於允許使用者藉由一般瀏覽器之一致性存取方式，從遠端觀看某一特定地點的即時影像資訊，並進而擷取可疑的歷史影像，在緊急情況時，尚可將相關資訊立即透過電子郵件或呼叫器傳撥出去。近來，WWW 的快速發展與 Java 技術的出現，兩者之間的緊密結合使得監視服務的附加價值得以大幅提昇；前者提供一個跨平台式的一致性視覺化使用者介面以利資訊的存取，而後者允許應用程式碼 (applets) 能從網際網路上被下載及執行於任何支援 Java 的網頁瀏覽器 [6]。雖然目前存在若干 Web 版的 video cameras 支援監視服務，但大多數都是應用於娛樂及實驗性質方面，功能過於簡單；另一類的應用系統，諸如 RealVideo、VivoActive、Web Theater 及 Oracle 8 等，則不是操作使用上太過複雜就是成本過高，一般使用者無法負擔其成本。

為了提供符合成本效益且多功能的 Web 版保全監視服務，吾人需考慮實務上的若干限制與技術上的相關議題。首先，一個 Web 版的監視服務在理想上應該可以傳送即時的視訊資料，然而高頻寬的需求使得在網際網路上要傳送這類資料，以目

¹ 本研究承國立高雄第一科技大學育成中心與一揚科技公司支持 (87B22)，謹此致謝。

² 本研究成果雛型榮獲「經濟部 87 年度 Java 應用軟體程式競賽—JavaCup」校園組優等獎。

前的網路頻寬來說是較不可行的，因此傳送序列性的靜態影像(still image)提供了另一個可行方案。此外，較高的資源的需求，諸如 CPU、獲得影像訊息所需的記憶體及用來處理服務請求的 HTTP daemon 等，使得爲了要更有效率地擷取影像訊號並對影像進行編碼，因而往往需要一台獨立的機器來處理這些事。尤有甚者，要爲視訊或靜態影像的擷取提供一個跨平台的解決方案是一個艱鉅的挑戰，因其涉略到驅動低階設備[3]。因此，很多 Web-camera 系統往往求助於使用一般如 C 或 C++的高階程式語言，來實作影像擷取的元件，這也往往是考量到低階驅動程式的存取及運作時的執行效能。這樣一來，就牽涉到網際網路系統開發的一項重要任務—異質環境之系統合作式整合 (interoperability)。

從資訊系統開發的角度來看，由於具有一致性之存取方式，以 Web 爲運算平台（包含 CGI）的機制已廣被應用，然而因其屬於「無狀態式」(stateless) 之協定以及客戶端物件無法直接與伺服器端溝通，造成執行效率上之缺失，此類運算模式已不足因應新一代主從架構系統之需求 [10]。Java 運算模式的興起爲此項缺失帶來一線曙光，而透過與 CORBA 分散式技術的密切結合，將主從式系統帶入所謂的「物件式 Web」(Object Web) 的新時代[10]。在物件式 Web 環境裡，應用程式或系統都將被包裝爲元件(component)，各元件間透過標準介面(interface)相互溝通。如此一來，系統的開發將呈現「隨插即用」(plug-N-play) 的方式進行，此即爲開放式系統的理想境界。SUN 公司新近發表之 Jini 技術亦即本此精神而研發。

本論文將探討如何應用「橋接技術」(bridging technologies) [2] 建立一個物件式 Web 的運算平台；尤其著重於 CORBA/RMI/Java/Web 的緊密結合，以進行多媒體視訊物件之訊息交換，並將其應用在支援整合型監視服務上。本論文的章節架構如下：第二節說明整合異質元件所面臨之課題與 Java 解決方案之可行性，第

三節則探討有關多媒體物件溝通上之型態議題，第四節介紹一實驗性整合型保全監視服務系統並討論元件間的互動關係，第五節則爲結論。

二、基於 Java 異質性計算環境之合作式整合

近幾年來，Java 儼然成爲網路語言的代名詞，由於 Internet 與 WWW 的因緣際會，使得 Java 綜合了 C/C++、p-PASCAL、LISP、Guarded Command 等多種傳統語言的諸多特性，例如：位元組碼 bytecode（跨平台性）、多執行緒 multithread（並時處理）、遠端物件方法呼叫 RMI（分散性）、抽象視窗工具 AWT（使用者圖形介面）、物件序列化 object serialization（物件永續性）、可下載式程式碼 downloadable applet（移動性）、嚴謹之安全機制 (security model)、介面化 (interface)之物件溝通等等 [9]。文獻[1]更指出，Java 將被選爲發展以網路運算爲中心的企業應用系統之基礎建設。例如，美國昇陽電腦公司之資訊技術部門即建置一個基於 Java 多層次服務環境，以支援散佈在 55 個國家中的 108 個地點之兩萬個員工[4]。

一般而言，以 Java 技術在異質性計算環境中，進行合作式系統整合（如本研究之保全監視服務）的方案有三：第一種方法是藉由 Java Native Interface (JNI) 將原生碼(native code)嵌入 Java 程式碼中。然而，在「視訊擷取」元件中包含了過多以 Windows 環境有關的原生碼，使得 Java 應用程式與原生碼的整合複雜化。第二個方法就是觀望的解決方案 (wait-and-see)。近來，SunSoft 與 IBM 合作之 Java Media Framework (JMF)工作小組，正努力克服 JMF 跨平台的障礙[11]。正如該小組所允諾的，新版的 JMF API 將會擴充先前由 SUN、Intel 及 SGI 所提出的建議，將允許系統發展者利用跨平台的工具集來控制具有時間性的媒體 (time-based media)，包含 video capture 的功能。第三，從軟體發展的觀點來看，

最可行的方法就是應用橋接技術整合異質性的計算機硬體、作業系統及程式語言等藩籬。OMG 的 Common Object Request Broker Architecture (CORBA) [8] 與 Web 的結合可說是橋接技術的一個成功典範[2]。此外，Java RMI 也是另一種備受重視的橋接技術，雖然它並不支援跨語言的特性。正如文獻[2]指出，RMI 與 CORBA 的結合將會是未來網際網路應用程式設計的主要潮流；而 Java 與 CORBA 的結合，正是促成物件式 Web 運算模式的主要功臣。

三、異質性計算環境之多媒體物件之交換

另一個有關在分散式異質環境之合作式整合性的難題，是如何確保多媒體訊息型態間之交換的一致性。儘管有若干個基本的與使用者自訂的資料型態已被先前所提之橋接技術所支援，但如何進行如影像及視訊等複雜型態的視覺化物件間的溝通，絕非是個簡單的工作，因其往往是與作業系統及程式語言相關。為了處理這類異質性的問題，吾人有必要詳細剖析視覺化物件的原貌。在 RMI 中，遠端物件可以藉由 call by reference 或 call by value 來傳送。為了傳送非原始的 (non-primitive) 資料型態之值，物件在被傳送之前，需經由 RMI 系統中的傳輸層將其序列化 (serialized) 成位元組資料流型態後方能傳送。在接收端收到該位元組資料流後，需將其解序列化 (deserialize) 後才能得到一份該遠端物件的複製。自 JDK1.1 版的 Object Serialization API 提供了將物件轉成位元組資料流及將位元組資料流還原為原物件的機制，程式設計者只需為欲序列化的 Java 物件，實作 java.io.Serializable 界面即可。然而，基於安全性的考量，並非所有的 Java 物件都可以被序列化；例如 JDBC ResultSet 物件及 image 物件就不能被序列化。一個解決 RMI 傳送 image 物件之限制的可行方案，就是萃取出 Java image 物件的圖素值 (pixels)，並將其存入一個整數陣列中。Client 端接收到此整數陣列後再將其還原為原來的 Java image 物件即可 [7]。

四、物件式 Web 運算模式之保全監視服務

在此研究中，我們建構了一個實驗性質的物件式 Web 運算模式之保全監視服務系統。圖一說明了整個系統架構，其中 Capturer Server、Surveillance Server 及 Client Monitor 為三大關鍵元件。以下將簡要地說明每個元件之功能與運作原理。

(一) Capturer Server

在本系統中，Capturer Server 裝配有一套遵循 H.263 標準的 video camera。由於 video 訊號擷取需用到較多 CPU 與記憶體資源，而促使得我們將 Capturer Server 獨立於另一台機器上執行，用來擷取影像訊號並進行編碼。使用者可自行設定影像擷取的時間間隔、JPEG 的影像品質及加入影像中的時間標誌 (time stamp) 格式。Capturer Server 中的即時影像擷取 (Snapshot Grabber) 目前是利用 MFC (Microsoft Foundation Class) 來實作。為了能和 Surveillance Server 元件產生合作式整合，Snapshot Grabber 藉由 CORBA IDL 來進行包裝。由於 IDL (為一種中介宣告語言、如同獨立於作業系統的 ORB) 只能提供有限的資料型態，例如 char、boolean、short、long、struct、sequence 等，但並不提供 video 及 images 等視訊型態的資料格式。因此，Snapshot Grabber 將以固定時間週期擷取某一特定地點的即時影像訊號，並將之轉成整數陣列的格式。

(二) Surveillance Server

Surveillance Server 扮演一個介於 Capturer Server 與 Surveillance Monitor 之間的中介角色，以減輕 Capturer Server 擷取即時影像的工作負擔。它完全是以 Java 所撰寫 (pure Java)，因此可以被安裝在任何平台上並執行。Surveillance Server 提供了多種服務，包括了壓縮影像的組成、Web 服務、呼叫器服務、使用者登錄服務及調閱歷史影像資料服務等。Image Composer 接收從 Capturer Server 傳來的整數陣列之即時影像格式，再把它傳給 Snapshot Feeder

以提供 Client 端觀看即時影像。此外, Image Composer 也將接收到的整數陣列還原為 JPEG 影像, 再將它送給 Archiving Server, Archiving Server 會把收到的影像依照擷取

(三) Surveillance Monitor

Surveillance Monitor 提供了多種不同的服務, 總的來說, 包含了瀏覽即時影像及歷史影像、儲存影像檔到 Local 端的檔案系統中、讀取 Local 端的檔案系統、在緊急情況下可立即透過數據機, 緊急傳呼呼叫器 (BBC) 及將影像以 email 方式透過 SMTP 協定傳送出去。Surveillance Monitor 也是完全由 Java 寫成(applets), 可以在任何支援 Java 的瀏覽器上執行, 例如: Netscape Communicator 4.07, 且不需安裝任何外掛程式 (plug-in)。圖三為 Surveillance Monitor 的使用者介面外觀。為了減少 Java applet 在安全性方面的限制, 本研究中的 applet 已經過認證, 而且並用 JAR 格式將它包裝起來, 可大幅縮短 applet 的下載時間並簡化安裝移動式程式碼 (mobile code) 到客戶端的程序。當下載此經過認證的 applet 時, 使用者端會接收到一個提示訊息, 用來確認此 applet 的證書。使用者端可選擇允許、拒絕或檢視此證書。

五、結論

隨著網際網路快速地發展, 電腦硬體功能日益強化, 傳統的 Web-CGI 運算環境已無法滿足新一代網路應用系統的需求, 本研究嘗試導入物件式 Web 與橋接技術, 特別是 Java 以及 CORBA 建構一滿足隨插即用開放式系統。為此, 我們以保全監視服務系統為個案研究, 成功地整合了異質軟體元件, 並克服了電腦硬體、作業系統與程式語言的藩籬, 更重要的是, 較易被忽略的多媒體物件資料型態的障礙, 亦一併得到解決。在這種物件式 Web 環境下所開發之保全監視服務, 具有多種符合成本效益之應用, 例如, 校園電腦教室與實驗室的監控管理、醫院的加護病房監控、交通流量監控及小型遠距教學系統等。

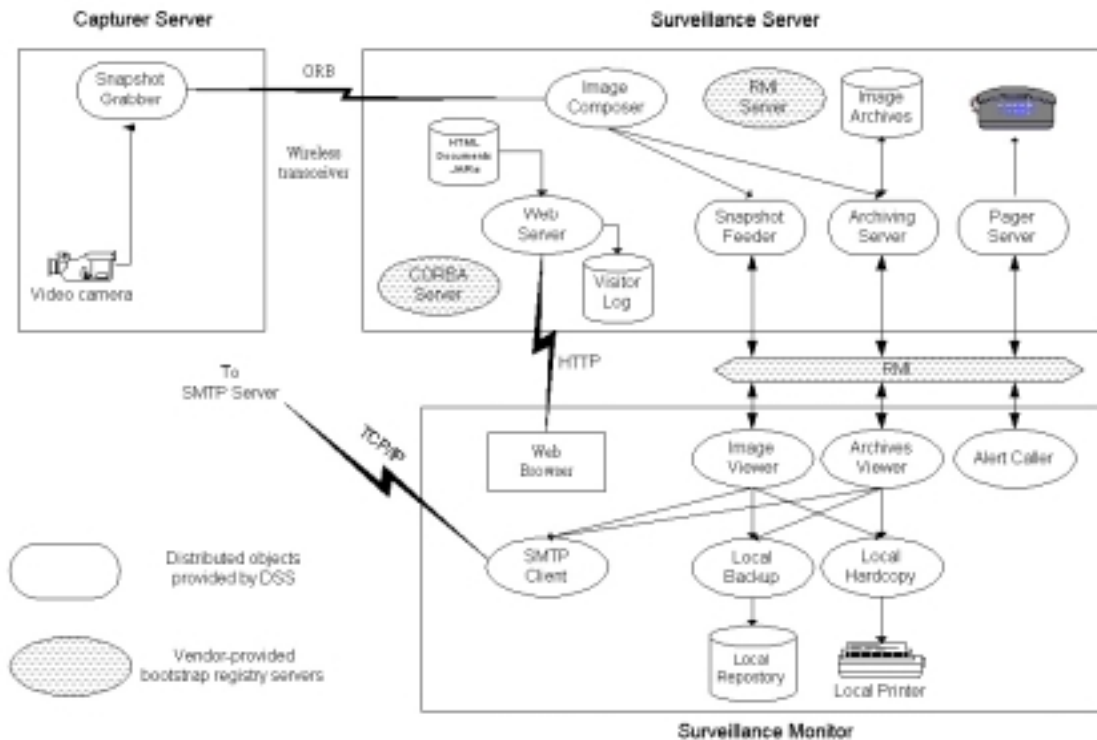
時間作分類目錄儲存 (Image Archives)。Surveillance Server 可以決定這些歷史影像要保留多久且可以建立工作排程來定時地清除歷史影像 (如圖二)。

參考文獻

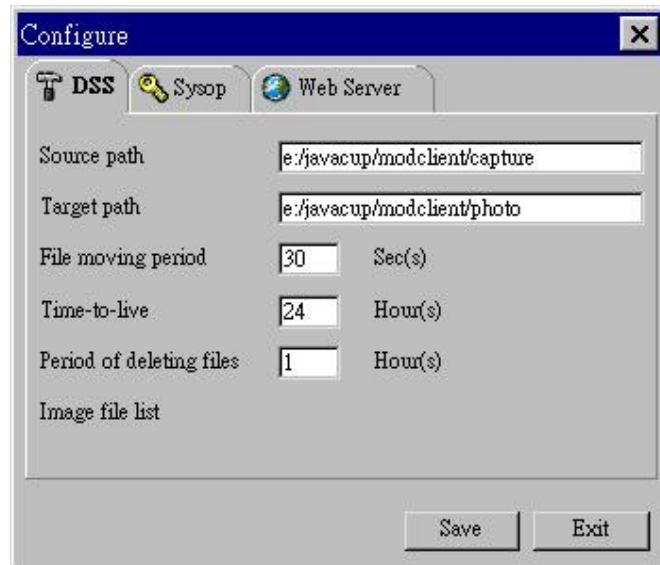
1. S. Ashok and V. K. Bansal, "Java: Network-centric Enterprise Computing," Computer Communications, Vol. 20, pp. 1467-1480, 1998.
2. S. Baker, V. Cahill, and P. Nixon, "Bridging Boundaries: CORBA in Perspective," IEEE Internet Computing, pp. 52-57, Sept-Oct., 1997.
3. P. Cadolini, F. Davoli, G. Marescotti, and P. Maryni, "Developing a Distance Learning System using Java Applets," Procs. of IEEE Global Telecommunications Conference, pp.63-66, 1996.
4. A. Gupta, C. Ferris, Y. Wilson, and K. Venkatasubramanian, "Implementing Java Computing: Sun on Architecture and Applications Deployment," IEEE Internet Computing, March-April 1998, 60-64.
5. JavaSoft Company, "Java Remote Method Invocation," <http://www.javasoft.com/products/jdk/rmi/>.
6. S.-T. Li and M.-Y. Shih, "A Multi-tier Java-Enabled Paradigm of Intelligent Message Exchange in Cyberspace," Procs. of the Pacific Workshop on Distributed Multimedia System, July 1998.
7. S.-T. Li and W.-S. Chen, "Java-centric Distributed Objects-based Paradigm for Surveillance Services and Visual Message Exchange," Accepted for publication in the Special Issue on Distributed Multimedia Information Retrieval, Int'l Journal of Visual Languages and Computing, 1999.
8. OMG, "CORBA," <http://www.omg.org/corba/>.
9. R. Orfali and D. Harkey, Client/Server

- Programming with Java and CORBA, John Wiley & Sons, Inc., 1997.
10. R. Orfali , D. Harkey, and J. Edwards, “CORBA, Java, and the Object Web, ” Byte, Oct. 1997, 95-100.
11. SUN Press Release, “SUN and IBM

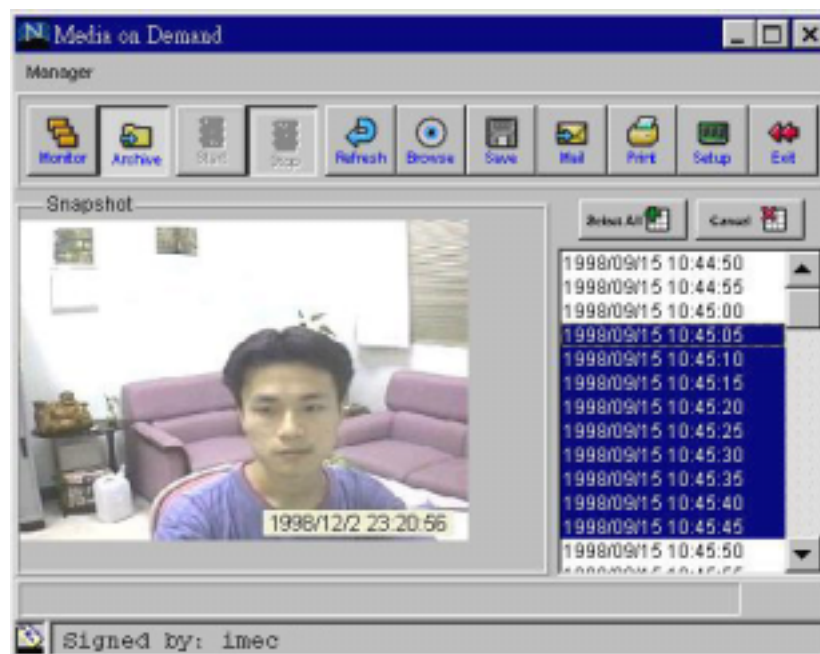
Collaborate on Java Media Technologies,” Internet World, <http://www.javasoft.com/pr/1998/10/pr981007.html>, Oct. 7, 1998.



圖一、物件式 Web 運算模式之保全監視服務系統架構



圖二、Surveillance Server 的控制面板



圖三、Surveillance Monitor 的使用者介面外觀