第三章　電腦應用技術之介紹

欲建構訊息傳遞系統，首先必須瞭解一些電腦相關的應用技術，本章主要即是對本研究會使用到的電腦技術，做一個簡單概要的介紹，有助於瞭解其使用的原理及應用的技術，以及說明建構系統所需要的相關軟體資訊，例如：Java、XML、Java 訊息服務、資料庫、網站、程式開發工具等等。

3.1　Java 程式語言之簡介

3.1.1　Java 的起源

全世界使用電腦的人口成長迅速，個人電腦對人們產生極大的衝擊，同時也改變了人們對於組織及商務管理的方式，美國昇陽公司(Sun Microsystems)有鑑於此，於 1991 年成立名為 Green 的內部研究計畫，主要研究方向是消費性電子產品所使用的軟體，如 PDA 等消費性電子的相容軟體。首先遭遇的問題，就是硬體平台的差異，因此開發所使用的程式語言，必須有此特性。但在當時的程式語言裡，找不到適合的工具來開發，最明快的方法，便是自行開發程式語言。該程式擷取其他程式長期累積下來的優點，避免其中一部份語言所具備的不安定性，才能完全符合一個超越平台、硬體設備的要求。因為程式設計師常到咖啡館喝爪哇咖啡而得到靈感，故命名 Java。

1993 年開始網際網路爆炸式的蓬勃發展,Sun 公司看到使用 Java 製作動態網頁的潛在發展商機，拜網際網路之賜，Java 竟然全世界為之轟動，如今 Java 可用於製作動態網頁、企業應用程式設計、全球資訊網伺服器端程式設計等等。[8][9]

3.1.2 Java 的特色

Java 目前已經廣為世人使用，它的應用範圍包括全球資訊網、網路通訊及未來的一些精巧通訊設備，並被視為未來企業資料庫的理想模型。Java 之所以如此快速且廣泛流行的主因，就是其程式只要撰寫一次即可到處被執行(Write Once，Run Everywhere)。

根據昇陽公司的 Java 語言白皮書(The Java Language：A White Paper)中提到 Java 的特色如下：[8][9]

1.簡單(Simple)：

在 Java 發表之前，C++是最流行的程式開發工具，但 Java 號稱比 C++簡單,例如指標和多重繼承通常造成程式設計者的困惑,所以 Java 移除指標並用介面(Interface)取代多重繼承。其次，C++必須由程式設計者動態的分配與收集記憶體，但 Java 卻將記憶體的管理改為自動化，所以 Java 可說是簡單的程式語言。

2.物件導向(Object-Oriented)：

物件導向才是真實世界的寫照，任何東西都可視為物件(Object)，例如人是一個物件、汽車是一個物件、能解一個二元一次方程式的小程式也可稱為一個物件。

3.分散式(Distributed)：

所謂的分散式系統是指數台電腦藉由網路連結而一起工作、分享資源。而 Java 對撰寫網路程式而言，就如同將資料放入檔案，或從檔案取出資料一樣容易。

4.解譯式(Interpreted)：

Java 程式需要解譯器將原始程式解譯為 Java 虛擬機器碼(Virtual Machine Code)，此機器碼稱為 Bytecode。此種 Bytecode 與機器無關，只要此機器具有 Java 解譯器，即能執行 Java 虛擬機器碼。

5.穩健(Robust)：

穩健的程式象徵此程式值得信賴，有許多程式語言均在執行的階段才發生無預警的錯誤導致結束。Java 編譯器在編譯階段費了很大的心思去找出任何可能的錯誤；再者，Java 去掉指標，減少錯誤產生；最後 Java 有一種處理機制，可以攔截使用者的操作錯誤，以確保程式穩健地執行。

6.安全(Secure)：

當 Java 程式執行時，JVM(Java Virtual Machine)可以監視其動作，增添了網際網路程式的安全性。除此之外，Java 設計者也可自行調整安全性的設定，此做法更具有彈性。

7.結構中立(Architecture-Neutral)：

Java 最值得稱道的特性就是它是結構中立的語言，此稱與平台無關(Platform-Independent)的語言，只要寫完程式，就可在任何具有 Java 解譯器的作業系統執行。

8.可攜式(Portable)：

Java 的 Bytecode 不用重新編譯即可在別台電腦執行，也不限定在何種平台上執行。

9.高效率(Performance)：

以往 Java 的 Bytecode 的執行效率總是比不上 C++的編譯式語言，但 CPU 在這幾年已有驚人的進步，加上 Java 編輯器經過改良，加以提高其執行效率，所以 Java 的執行效率是無庸置疑的。

10.多執行緒(Multi-Threaded)：

多執行緒是指一個程式具有同時執行多個工作的能力，在現代的程式裡，同時要做很多事是必備的功能，Java 當然也提供此多功的能力，圖 3-1 即為多執行緒示意圖。

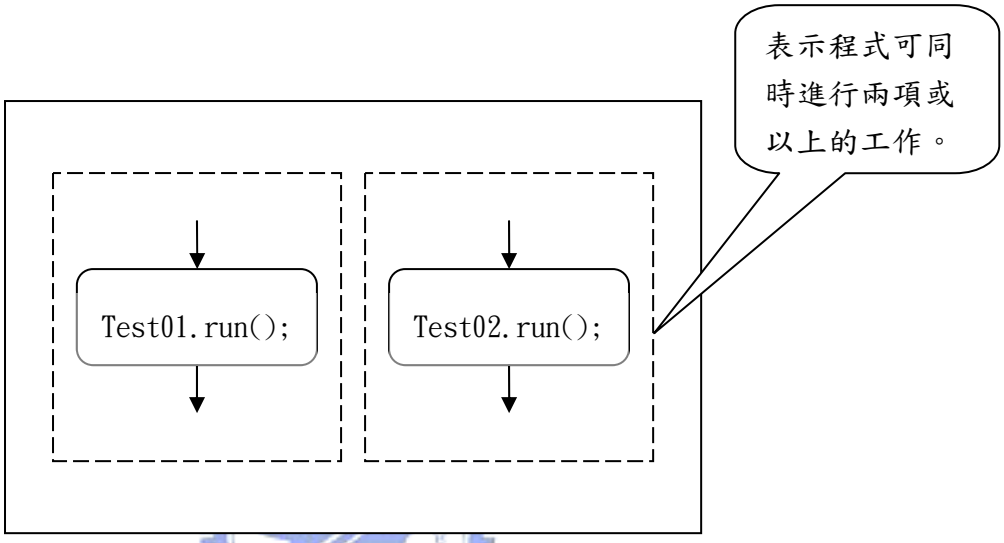

圖 3-1 多執行緒示意圖

11.動態(Dynamic)：

Java 是設計來適合會進化的環境，它可以自由地在原類別中新增屬性或方法而不影響原程式的進行。

3.1.3 Java 的使用方式

欲使用 Java 所開發的程式須先於平台上安裝 JDK 或 Java 執行環境(JRE，Java Runtime Environment)，示意如下圖 3-2：

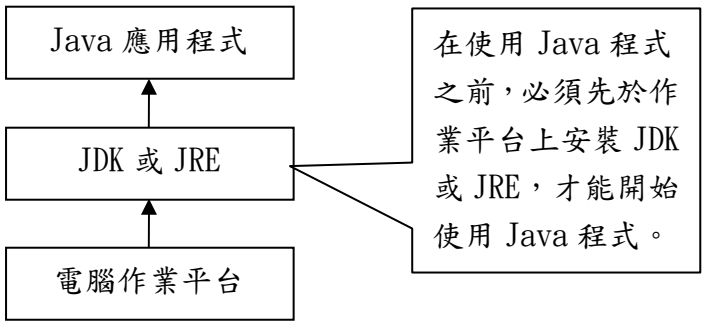

圖 3-2 Java 之使用示意圖

Java 是將程式解譯成與電腦作業平台無關的機器碼，所以 Java 所開發的應用程式，必須於電腦平台上安裝能解讀機器碼的工具，才能使程式於平台上運作。也因為有此過程，使得 Java 才能具備跨平台的特色，因為只要在各式平台上安裝能解讀機器碼的工具，就能使 Java 應用程式運作，而解讀機器碼的工具即指 JDK 或 JRE(Java 執行環境)。Sun 發表的 Java Development Toolkit，簡稱 JDK，內含 Java 的編譯器、Applet 瀏覽器及許多有用的工具，這些都是免費的，可於 Sun 公司的網站上輕易的取得，如此也加速 Java 的流通與普遍，而至今 Java 已是開發跨平台程式的主要程式語言。

3.2 XML (eXtensible Markup Language)

3.2.1 XML 的起源

早期標記語言(Markup Language)的產生，是用於印表機工作時利用不同的標記來控制文章；後來出現 SGML(Standard Generalized Markup Language)，專門為文件交換與處理所訂定的標準，雖然功能強大，但卻過於複雜，因此並未被廣範使用。隨著網際網路的進步，標記語言有了新的應用範圍，也就是設計網頁用的 HTML(Hypertext Markup Language)，透過 HTML 可以控制網頁之排版，HTML 成功的地方主要是其規格簡單，很容易被網頁設計者所接受，但 HTML 有一些缺點如：標籤固定、偏重內容的顯示、原始碼不易閱讀、不適合自動化資料處理、語法不嚴謹等等，使得 HTML 的應用也到達了瓶頸。

SGML 太複雜，HTML 不夠用，於是就有 XML 的誕生。XML 是由 W3C(World Wide Web Consortium)在 1998 年 2 月發展制定出來的，XML 為 eXtensible Markup Language 的簡稱，中文稱為「可延伸標記語言」。XML 的出現就是為了要應用於各個不同的層面，可依據應用領域的不同，而以不同方式來描述文件。簡單來說，XML 就是以一種簡單、標準並可擴展的方式，將各種資訊如文字、表格，甚至圖形等以原始資料(Raw Data)的方式儲存，在儲存的過程中，加入一些可供辨識的標籤(Tags)，藉此辨識資料含意。

3.2.2 SGML、HTML 與 XML 的比較

SGML、HTML 與 XML 優缺點比較，如表 3-1 所示[10]：

表 3-1 SGML、HTML 與 XML 的比較

語言	描述	優點	缺點
SGML	描述文件語言	功能完整、具延伸性。	複雜。
HTML	超連結標記語言	簡單易用。	呈現資料，無法延伸。
XML	可延伸標記語言	簡單易用、具延伸性。	需要結合其他技術。

由上可知，XML 取 SGML、HTML 之優點，彌補其不足，可說是新一代的技術發展。

3.2.3 XML 的制定目標

制定 XML 的委員會提出 10 點制定目標，如下所示[11][12]：

1.能直接應用在 Internet：

雖然 SGML 也能被應用在全球資訊網，但它的高複雜性會使得這種應用付出更多的代價，況且 SGML 並沒有支援相關 Web 的通訊協定，例如 SGML 就無法有超連結到遠端資源的能力。所以 XML 要克服上述的問題，就必須具有以下的能力：

- 能支援相關 Web 的通訊協定。
- XML 要盡量簡化，不可像 SGML 如此龐大且複雜。

2.能被各式的應用軟體使用：

同一份 HTML 文件被不同的應用軟體來解讀後，其擷取出的資料可能都會不盡相同，除了是因為 HTML 的語法不夠嚴謹外，也可能因該份 HTML 文件使用了非標準的控制標籤的緣故，如：Microsoft、Netscape、Oracle、Louts 等等公司，為了特殊用途會自建一些控制標籤，這些控制標籤並沒有通過 W3C 的認可，只能在各家公司的軟體才能解讀，無法被廣泛的應用軟體所解讀。

XML 文件被建立後，可透過 XML 剖析器(Parser)先來解讀該份 XML 文件，確定是否合法正確，如果是正確才可被其它應用軟體來使用，所以只要 XML 剖析器有根據 XML 規範中的要求來設計，就能發揮檢驗 XML 文件的功用。

3.能與 SGML 相容：

因為有很多大企業或政府部門已投入 SGML 應用多年，所以他們是不可能全數放棄重頭再來，因此 XML 文件必須也能夠被 SGML 的應用軟體所解讀。但相反的，XML 的應用軟體未必要能解讀所有的 SGML 文件，如果 SGML 文件中有使用到超出 XML 規範的格式，XML 應用軟體就無法成功解讀該份 SGML 文件了。

4.能輕易發展 XML 相關軟體：

因為 SGML 過於龐大且複雜，所以要發展 SGML 的相關軟體是要付出相當高的代價，且開發的時間也會很長。在簡化 SGML 創造 XML 的過程中，也需要將如何降低開發 XML 應用軟體成本的因素加入考量。

5.能簡化 SGML：

在 SGML 標準中，有許多功能都是選擇性的，也就是說這些功能可以被設定使用或不使用，甚至對一般 Web 上的應用者來說是很少用到或根本不會使用到。XML 規範盡可能將這類選擇性的功能減到最少。

6.XML 文件可讀性高：

所謂的高可讀性是針對人來說的，也就是說希望人也可以很輕易的解讀 XML 文件，當然這需要 XML 文件有高的結構性。如 SGML 的高複雜性與 HTML 的語法不夠嚴謹，都是不利人來解讀的。

7.XML 規範能盡速完成：

因為 HTML 已是無法滿足在全球資訊網的應用，但在全球資訊網如此蓬勃發展下，各大軟體廠商對於新世代的標記語言期望更是殷切。所以 XML 開發小組為了避免軟體廠商各自制定新的標記語言，並且希望先發制人獲取先機，故期望能盡速完成 XML 規範的制定，以作為各軟體廠商依循的標準。

8.XML 規範須簡潔：

為了減少學習 XML 規範的時間，決定採用 EBNF(Extended Backus-Naur Format)格式來展現 XML 規範。

9.能輕易建立 XML 文件：

XML 文件可以包含任何設計者所自定的元素，只要該份 XML 文件符合規範的要求即可。

10.語法不可模糊不清：

HTML 的語法不夠嚴謹，是因 HTML 允許設計者可以省略某些控制標籤，但容易造成文件語意上的模糊不清，因設計者可能有的省略，有的卻沒省略，這對解讀的應用軟體來說是一種困擾，更何況是人來閱讀該份文件。所以 XML 文件不允許這類省略的現象發生。

3.2.4 XML 的格式定義與解析方式

一份 XML 文件可以區分兩種格式定義，如下圖 3-3 所示：

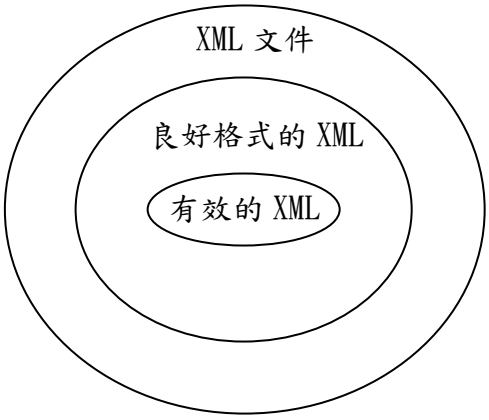

圖 3-3 XML 的格式定義

良好格式的 XML 文件(Well-Formed XML)不需要另外以格式定義的檔案或內容規範，只要文件結構能符合 XML 文件的規定即可，如圖 3-4 所示即為一份良好格式的 XML 文件：

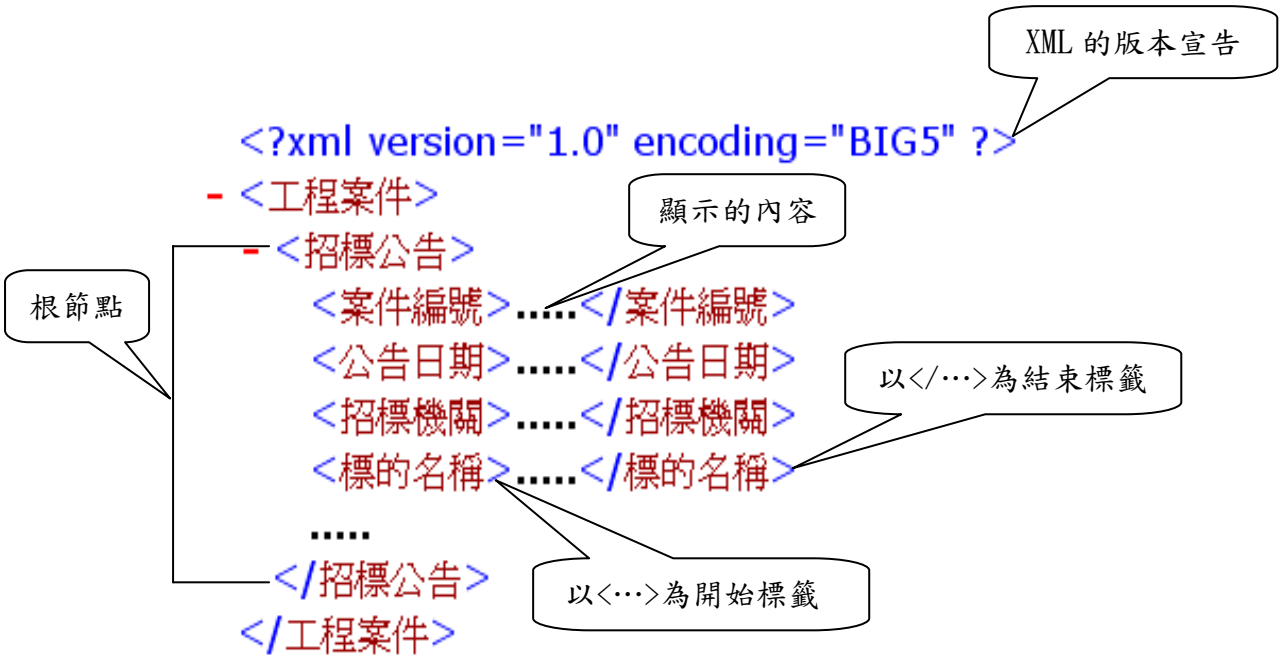

圖 3-4 一份良好格式的 XML 文件

而有效的 XML 文件(Valid XML)則必須同時在良好格式的情況下，另外以格式定義的檔案或內容規定。有效的 XML 文件在現階段的使用中採取了 DTD 以及 XML Schema 兩種方式定義。而本研究實作的系統中採用良好格式的 XML 文件進行開發。

在接收到一份良好格式的 XML 文件後，必須針對文件裡的資料內容加以處理，才能真正得到需要的資訊。XML 資料的處理方式，是透過 DOM 解析器的解析，將資料分解成樹狀的結構，如圖 3-5 所示：

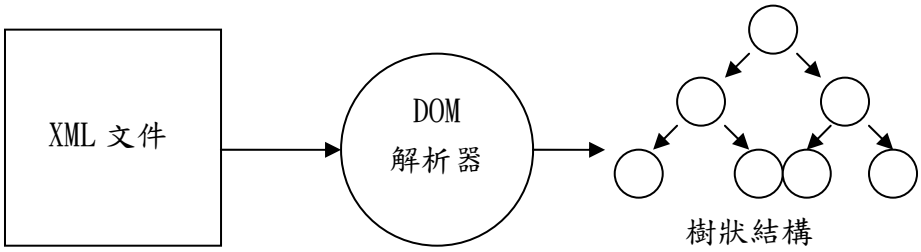

圖 3-5 XML 文件的 DOM 解析方式

當資料解析為樹狀結構後，就可依其相關位置，很輕易的擷取需要的資料來加以運用，或是加入其他相關資料，此即 XML 可延伸、易擴充的原則。

3.2.5 XML 的優勢

XML 之所以能讓網際網路上的資料更能互相流通，讓文件的內容更能顯而易懂，一定有它過人之處，其優勢如下所示[12]：

1.延伸性：

由於 XML 可以自由創造新的標籤來配合運用使得 XML 的應用層面具有無限的延伸性，可以稱為超語言(Meta-Language)。

2.簡單易懂：

XML 文件內容都是以文字來表示，所以利用一般的文字編輯器就可以編輯修改，而且表達十分直覺化，既簡單又容易了解語意。

3. 異質系統間的資訊傳遞：

在目前的資訊社會中，存在各式各樣的資訊產品，各產品間的系統也大不相同。藉由 XML 的特色，為不同系統間提供了一個溝通的平台，扮演著一個資訊傳遞的媒介。

所以利用 XML 於異質系統傳遞資訊，可如圖 3-6 所示：

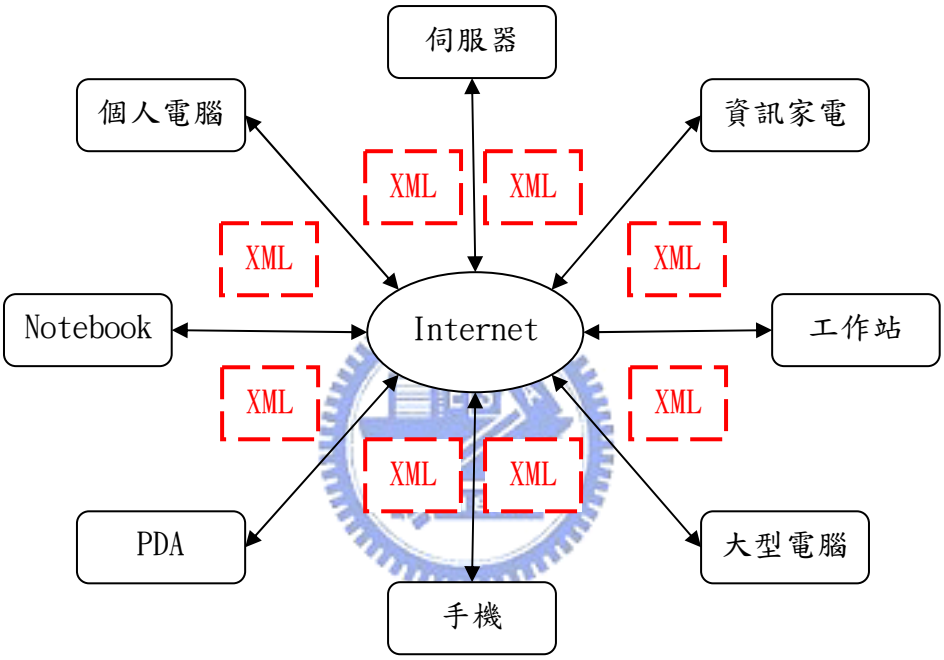

圖 3-6 異質系統的 XML 溝通平台

4. 國際化：

XML 支援多語系文件與 Unicode，在設計 XML 的一開始，設計者便考慮到，資訊流通不該只受限於一個國家或一塊區域，且隨著網際網路在全世界的普及，將一份文件能夠以多種不同的語言來呈現，是 XML 必須擁有的能力，所以便將 XML 建構在 Unicode 之上。

3.3 訊息傳遞機制之探討

訊息傳遞指使用訊息收送系統在網路上交換訊息，電子郵件是目前最廣為使用的訊息收送系統，透過電子郵件的傳遞，人與人之間的溝通更為便利。各企業間常常需要做資料的轉移以及交換，這樣的工作，也可視為是一種訊息的傳遞，而傳遞的訊息即為欲交流的資訊或資料。所以了解訊息傳遞的機制與傳送的方式，實在刻不容緩。

3.3.1 訊息傳遞的架構

訊息傳遞的架構可分為兩種，分述如下[13]：

1.分散式架構：

如同點對點的傳遞，點與點之間都存在一條專屬的道路專供聯繫。這種系統在節點較少時也許能發揮不錯的效果，但是若節點較多時勢必產生負荷過於沉重的狀況。節點為了傳遞訊息所要具備的能力，也將必須隨著網絡的複雜而增加，畢竟每一次的運作都必須在節點端進行，因此必須考量系統中成員的組成是否適當。

2.中央集權式架構：

所有的節點都透過一個核心媒介的運作，並且將屬於各節點的相關資訊，在系統中作雙向傳輸，達到資訊交流的目的。此核心媒介的存在，就如同一個代理人的角色。一個專司訊息控制的代理人，可以接

受來自四面八方的訊息，並且將訊息無誤傳送到指定的目的地。因此就算是節點再多，也能夠一目了然系統其中的運作。同時，節點端的軟弱並不會影響到整體系統的表現，畢竟真正處理訊息傳遞的工作是落在媒介身上。

圖 3-7 與圖 3-8 即為分散式架構與中央集權式架構示意圖：

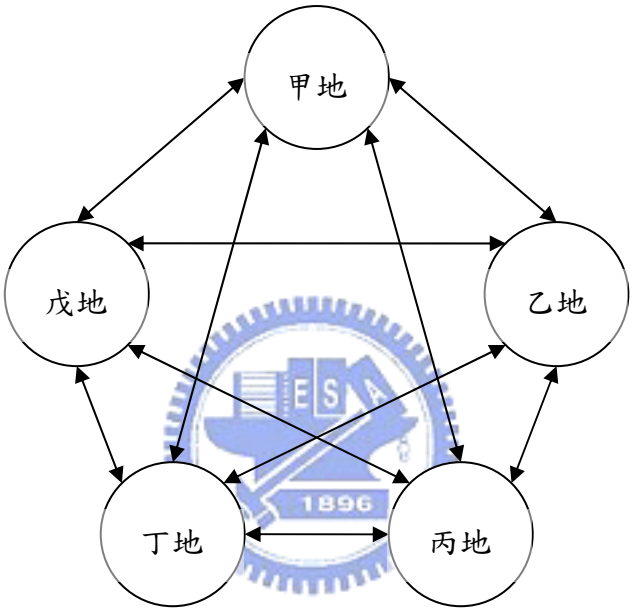

圖 3-7 分散式架構

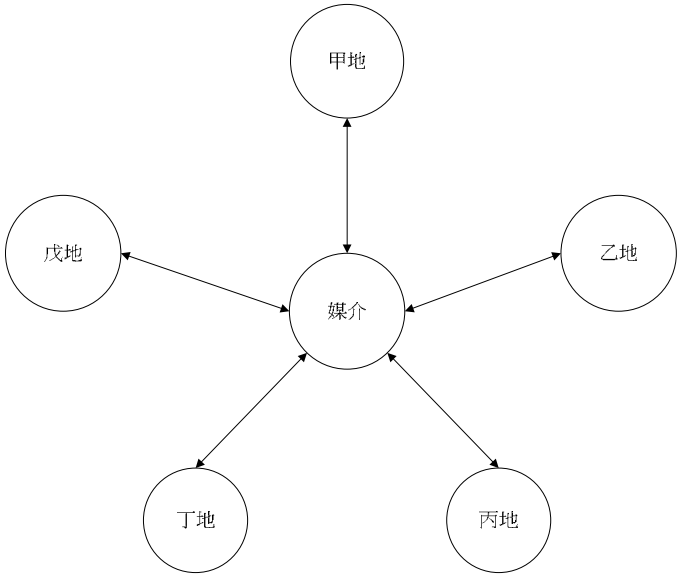

圖 3-8 中央集權式架構

由訊息傳遞的架構來看，可以了解到效率的差別，分散式架構的效率勢必較低，因為事必躬親。若是網絡上的各個成員效能落差很大，必定將拖累所有成員，所以必須對成員在軟體或硬體規格上要求，而必須付出升級的成本代價。如果採用中央集權架構的方式，則只要針對媒介的部分加強效能，對於訊息傳遞的效率並不會因為成員的差異，而互相影響到整體系統的營運。

比較過優缺點後，本研究將採用中央集權式的架構，進行程式的開發。而中央集權式架構中，扮演媒介的角色稱為訊息導向中介軟體(MOM，Message-Oriented Middleware)，或稱為企業訊息系統。在企業訊息系統中，訊息的用途在於用來通知另外一個系統發生了特定的事件。MOM 的介紹與必須具備的功能，後面將陸續介紹。

3.3.2 訊息導向中介軟體

訊息導向中介軟體(MOM，Message-Oriented Middleware)是指可讓兩個或以上的程式，透過訊息來交換資訊，或稱訊息伺服器。當使用 MOM 的時候，MOM 會確定訊息能夠適當的傳送。此外，MOM 通常會有容錯、負載平衡、規模可變，以及交易的支援，如此一來，企業才可能依賴其訊息交換的機制。

MOM 傳送的重要概念為「將訊息從某系統用非同步化的方式，透過網路傳送到另一個系統」，所謂的非同步化訊息傳送，意味著傳送端把訊息丟出去後，就可以回來繼續做其他的事情，不需等待訊息被接受或處理完畢。

MOM 必須具備以下幾個特點[13]：

1. 訊息系統是一種 Middleware，相較於傳統的 Client-Server 的架構，Middleware 用來隔離用戶和伺服器，可讓用戶端程式不必擔心通訊和資料傳送等事宜，它會負責效率和可靠性，如此可以減少用戶端程式的複雜度。
2. 相異系統之間的整合，意謂不同平台、程式語言、通訊協定可以整合在一起，MOM 可提供簡單且具彈性的解決方案。
3. 資訊交換往往需要保證訊息能無誤傳送與接收，即使系統失敗，也必須保證資料能被重新傳遞。
4. 當訊息傳送到 MOM 後即可不理，自動商業處理的程序可以減少直接和使用者的互動，這可避免因等待彼此的回應而影響效率，使 MOM 能處理更多的用戶需求。
5. 保證系統的可靠性，當一個系統失敗時，可將訊息自動導向到另一個系統而繼續訊息的傳遞。

MOM 不是一個新的概念，這樣的產品已有不少，包括：SonicMQ、IBM 的 MQSeries、微軟的 MSMQ 等等，下圖 3-9 為 MOM 示意圖，即表示用戶程式利用連結 MOM 後來傳遞訊息。

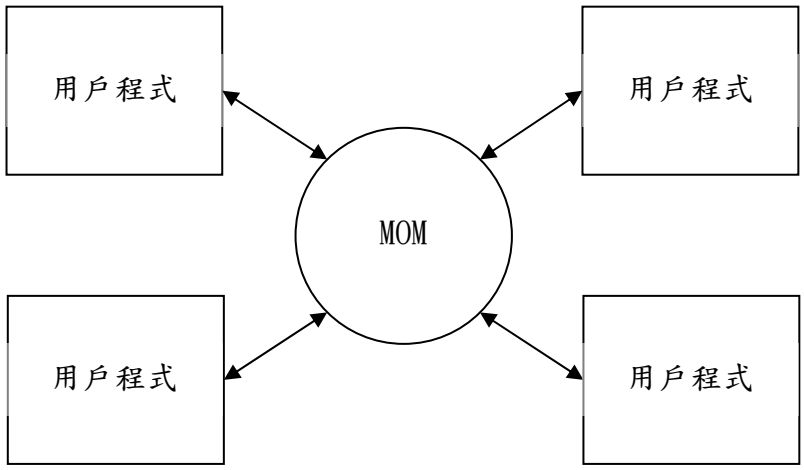

圖 3-9 MOM 示意圖

而本研究採用 SonicMQ 作為 MOM。SonicMQ 是最早開發 MOM 的廠商之一，它具備的功能完整，並且主要支援可傳遞 XML 格式的訊息，故選擇 SonicMQ 做為本研究的訊息伺服器。

3.3.3 Java 訊息服務(JMS，Java Message Service)

雖然 MOM 在商業系統的整合有其實用性，但系統間並沒有一個通用的程式介面，來自不同公司的訊息系統會有不同的介面，所產生的訊息程式只能在某個廠商的系統上執行，這造成往後轉移的問題。於是 Sun 公司和主要 MOM 廠商於 1998 年提出 JMS 的標準，又稱 Java 訊息服務(JMS，Java Message Service)。

JMS 本身不是一套訊息系統，而是一組抽象的介面與類別，用來提供 Java 程式發展者開發通用的訊息傳遞程式，可和 MOM 連接以傳送或接收訊息，並減少程式發展者需要了解不同 MOM 特性所花費的時間，且增加程式的可攜性。圖 3-10 所示，意指具有訊息傳遞功能的 Java 應用程式就是經由 Java 訊息服務介面(JMS API)和各 MOM 廠商產生連結來執行訊息的傳送與接收。

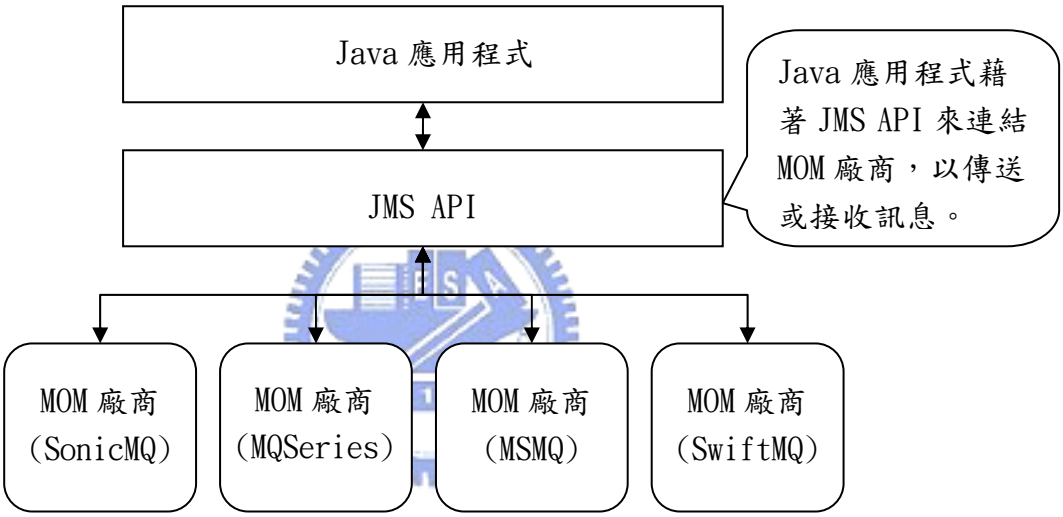

圖 3-10 JMS 的使用方式示意圖

3.3.4 JMS 的訊息傳遞模型

JMS 提供兩種訊息傳遞的模型，分別敘述如下[14]：

1.點對點(PTP，Point-To-Point)：

每個訊息只有一個消費者，目的地為 Queue(佇列)。訊息發送者將訊息傳入至 Queue 中，而訊息接收者到 Queue 中接收屬於自己的訊息，訊息將存留在 Queue 中，直到過期或被接收。

2. 出版與訂閱(Pub/Sub，Publish-and-Subscribe)：

每個訊息容許有多個消費者的存在，目的地為 Topic(主題)。出版者將訊息出版到 Topic 中，而一位或以上的訂閱者到 Topic 中訂閱屬於自己的資料。而訂閱方式有「非持久的訂閱」，意指出版者及訂閱者必須同步上線，才能完成交易。另為「持久訂閱」，指訊息將存在 Topic 中，直至過期或被接收掉。

圖 3-11 為點對點訊息模型，將訊息存於 Queue(佇列)中，只允許一位接收者來接收訊息。

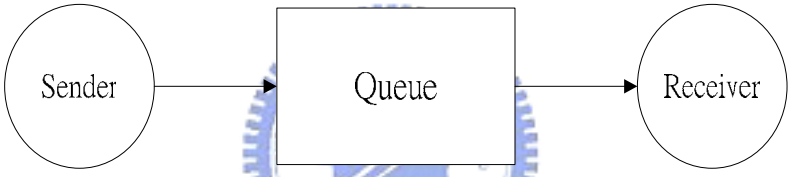

圖 3-11 點對點訊息傳遞模型

圖 3-12 為出版與訂閱訊息模型，將訊息放於 Topic(主題)中，讓一個以上之接收者來收取訊息。

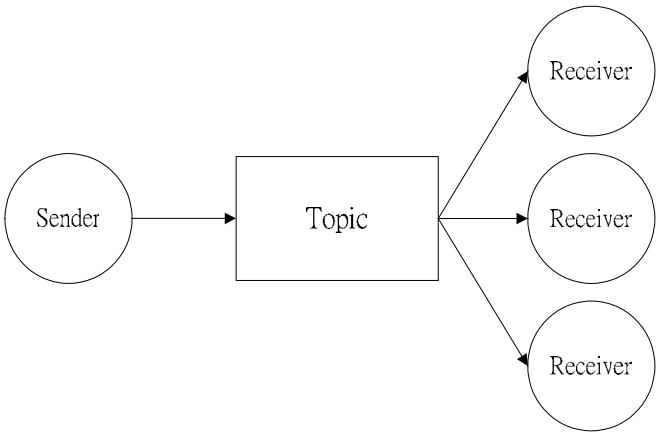

圖 3-12 出版與訂閱訊息傳遞模型

3.4 資料庫的應用

3.4.1 資料庫(Data Base)之簡介

所謂資料庫(Date Base)，即整合文書軟體、試算表等作業軟體，將大量類似格式資料提供儲存、輸入、查詢、並與相關資料相互連結於同一視窗或表單(Form)，並依需求提供報表(Report)、列印等等，多功能於一完整作業系統中。其特色為減少重複輸入、不同作業環境連結、類似性質交叉連結、資料格式化、錯誤警示、單一操作環境等等，系統背後需由專業人員針對使用者做程式設計。

3.4.2 實作系統之資料庫

本研究使用之資料庫軟體為「Microsoft Access」，共有兩個資料庫，一為工程資料庫，儲存各專案的相關資訊，另一個為廠商資料庫，作為存放廠商相關資料用，例如招標資訊、報價單等等。

下圖 3-13 為工程資料庫中各資料表的關聯圖：

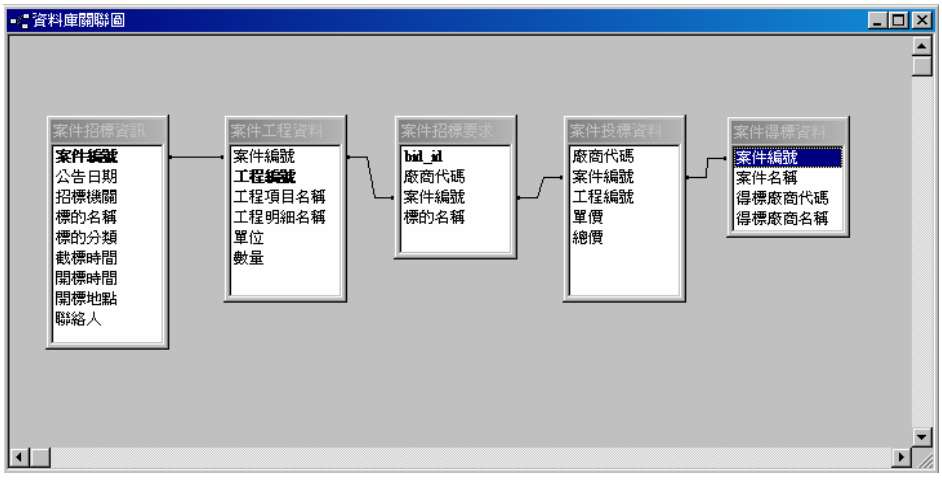

圖 3-13 工程資料庫之資料表關聯圖

廠商資料庫之資料表關聯圖如下圖 3-14 所示：

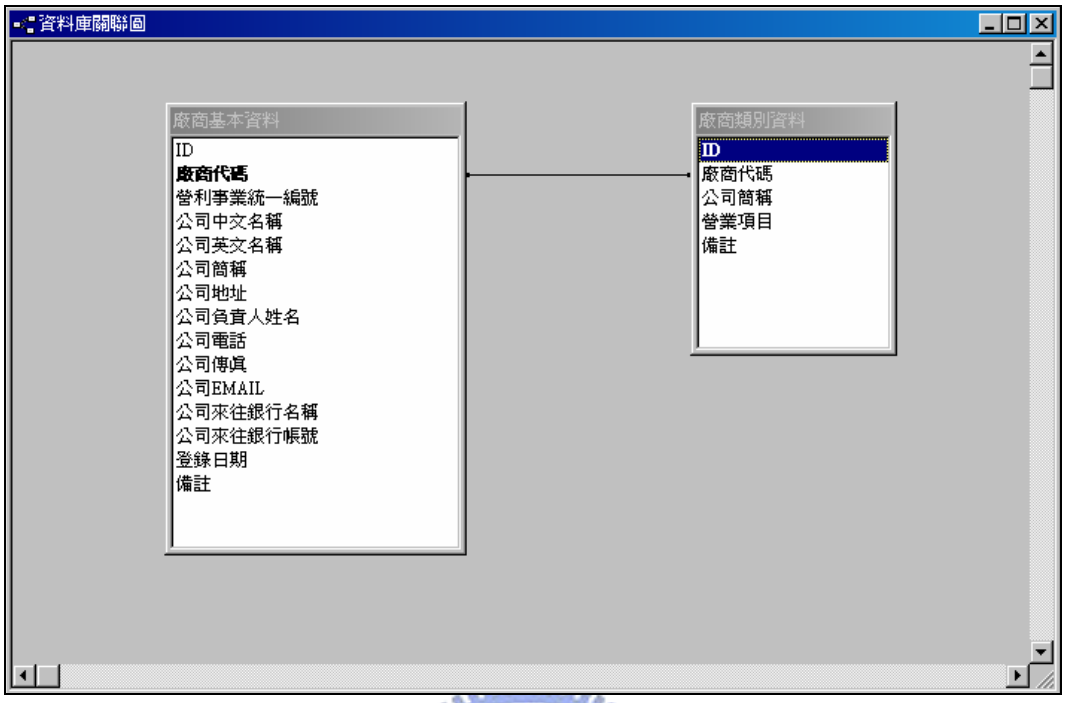

圖 3-14 廠商資料庫之資料表關聯圖

3.5 網站的應用

本研究需要架設一網站，用以提供廠商先註冊登記相關的廠商資料，然後取得本研究開發之訊息傳遞程式來使用。

網站亦設有兩個資料庫。一是密碼資料庫，用來儲存廠商帳號及密碼，是在網站修改廠商資料所需要用到的；二是廠商資料庫，主要儲存廠商的基本資料和各個廠商所登記的營業工程項目種類。此資料庫和本研究開發的訊息傳遞程式是共用的，裡面存放的廠商資料將是採購機關用來選擇傳遞訊息給廠商的根據，只有在網站上註冊並填好廠商基本資料的供應商，才能使用本程式來接收訊息。

3.6 程式開發工具－Borland JBuilder

要開發一個良好的程式，開發工具的選擇是非常重要的。本研究欲開發 Java 的訊息傳遞程式，而開發工具種類繁多且效能不同，有鑒於 Borland 公司所推出的 JBuilder 功能非常強大且實用，於是選擇使用 JBuilder 來開發本研究的訊息傳遞程式。

JBuilder 提供的功能有很多，最主要是有強大的程式除錯器、專案管理模式與最重要的使用者視覺化設計介面，圖 3-15 所示即為 JBuilder 之使用者視覺化設計介面，可利用此介面設計程式的外觀。

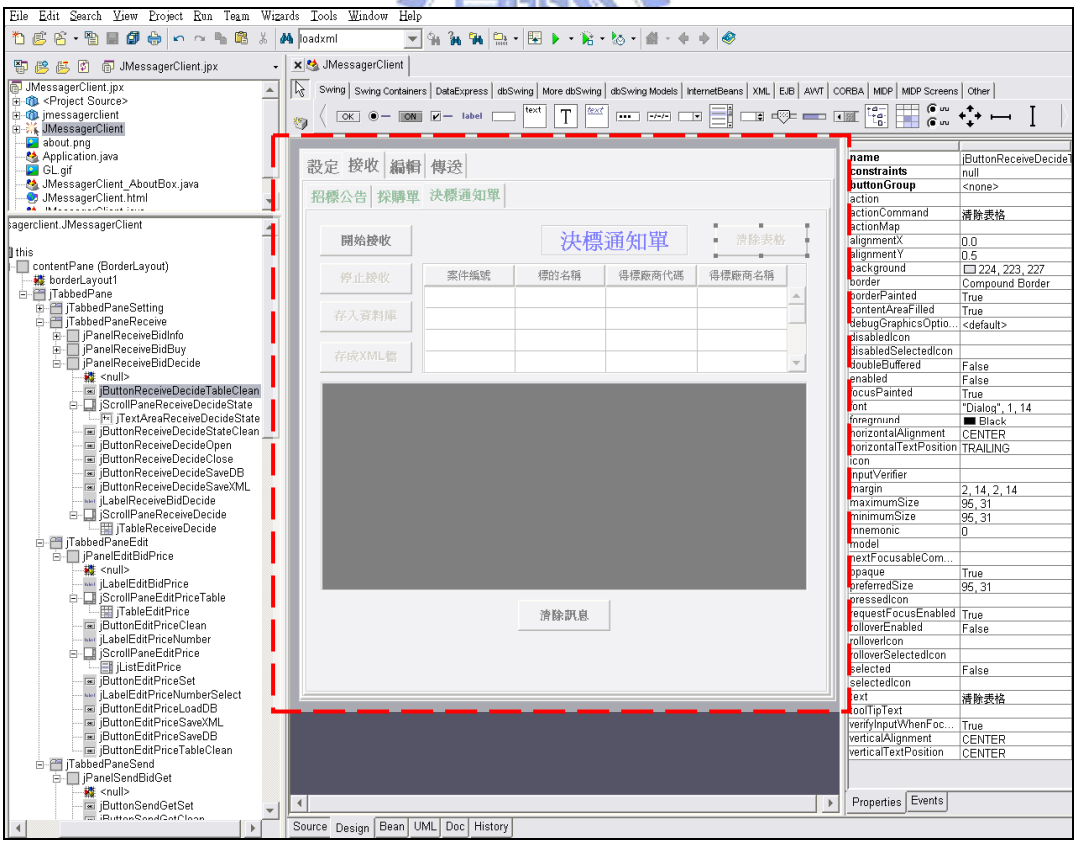

圖 3-15 JBuilder 之使用者視覺化設計介面

利用圖 3-16 所示之 JBuilder 程式碼設計環境來設計程式碼，有助於加快程式撰寫的速度，減少程式碼的混淆與錯誤。

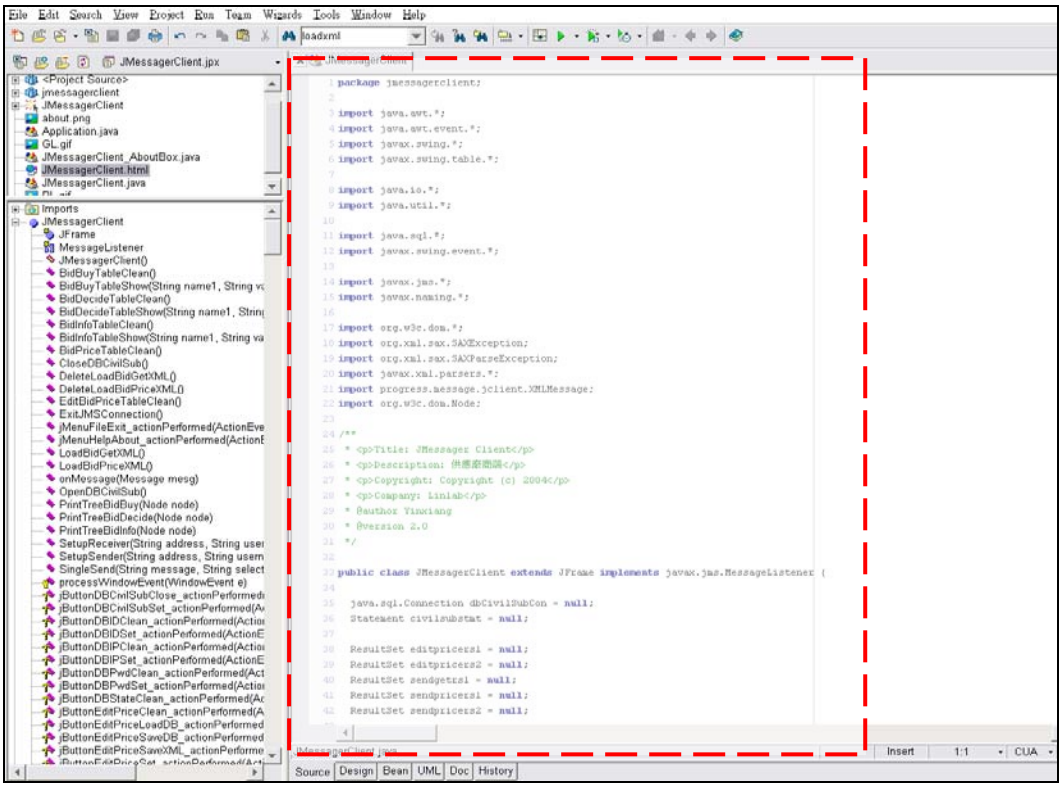

圖 3-16 JBuilder 之程式碼設計環境

Borland JBuilder 是全球第一的跨平台 Java 開發環境，可建構符合業界標準之 Java 應用程式，雖然 JBuilder 工具非常龐大且複雜，但如果充分運用其提供的功能，不但可設計專業的 Java 應用程式，快速的建置程式外觀介面，並且更方便未來程式功能的擴充和修改，相信欲開發 Java 程式，JBuilder 不失為一個良好的開發工具。

3.7 本章總結

本章探討欲建構系統所必須先瞭解的電腦知識，並且將其加以應用。Java 與 XML，是 21 世紀最強大的組合，Java 有助於跨平台，是易於處理物件的應用程式語言；而 XML 資料格式則是容易存取資料，替 Java 提供了最佳的溝通管道。XML 不能單獨使用，但 Java 可以為 XML 提供易於處理資料的程式碼;XML 則為 Java 提供通用的資料來源格式。XML 與 Java 是天生一對、相輔相成，兩者都將是網路分散式運算的關鍵技術。

根據上述的優點，本研究將選擇 Java 與 XML 作為主要技術，JBuilder 為開發工具，設計一套訊息傳遞程式，結合 Java 訊息服務(JMS，Java Message Service)，並建置網站與資料庫，整合成一套訊息系統，期望能發揮最大功效，有助於營建電子化。