

# 建築管理請照網路作業系統之研究 (二)

內政部建築研究所研究報告

MOIS921024

# 「建築管理請照網路作業系統 之研究（二）」

研究主持人：葉主任秘書世文

共同主持人：郭榮欽、謝尚賢

研 究 員：簡添福、郭明裕

研 究 助 理：林鴻皓、吳雪如、廖麗珠、楊鴻仁

內政部建築研究所研究報告  
中華民國 92 年 12 月

## 目次

圖次	III
摘要	V
英文摘要	VIII
第一章、前言	1
第一節、研究動機	1
第二節、研究範圍	3
第三節、研究步驟	4
第二章 文獻整理	11
第一節、建築物生命週期	11
第二節、XML、XPath、XSL 及相關技術	17
第三節、UML 系統分析語言	24
第四節、.Net 架構探討	29
第三章 建築管理請照資訊系統分析	34
第一節、系統規劃構想	34
第二節、需求分析	39
第三節、使用案例說明	40
第四節、系統規格	46
第五節、系統基本架構	49
第四章 建築管理請照系統設計	56
第一節、使用者介面規劃	56
第二節、系統輸入介面設計考量	60
第三節、系統輸出設計考量	65

第四節、LFT 的運作機制-----	67
第五節、系統靜態結構說明-----	69
第六節、程式封裝及部署的設計-----	75
第七節、軟體安裝及啟用流程-----	76
第八節、縣府端資料接收及轉型的機制-----	77
第九節、系統開發資訊及使用者回饋機制的建立-	79
<b>第五章 建築管理請照系統設計 -----</b>	<b>81</b>
第一節、操作介面-----	81
第二節、系統輸入介面展示 -----	89
第三節、系統輸出介面展示 -----	92
第四節、網路作業-----	94
<b>第六章 結論 -----</b>	<b>96</b>
第一節、使用者的效益 -----	97
第二節、建管部門的效益 -----	99
附錄 A：自訂控制項-----	101
附錄 B：案件列表物件之狀態圖-----	113
附錄 C：編輯介面主要類別圖-----	114
附錄 D：系統安裝流程圖-----	115
附錄 E：系統之 XML Schem-----	116
附錄 F：系統相關軟體量測值-----	126
參考書目 -----	132

## 圖次

圖 1.1、研究步驟流程圖	10
圖 2-1、建築物生命週期的共享資訊層次關係圖	16
圖 2-2、建築物生命週期的資訊鏈	17
圖 2-3、XSLT 的轉換程序	24
圖 2-4、UML 五個不同的模型觀點	26
圖 3-1、建管資訊相關人、事、物的分類觀點	35
圖 3-2、建造執照文件資料解析	38
圖 3-3、建築管理請照系統的使用案例圖	41
圖 3-4、建築管理相關活動圖	42
圖 3-5、建管請照資訊類別圖	45
圖 3-6、資訊「橫向共享」與「縱向共享」	50
圖 3-7、建築物文件資訊之 XML Schema 架構圖	52
圖 3-8、應用層的系統架構圖	54
圖 4-1、系統畫面及功能表內容	58
圖 4-2、Windows Form 的元件階層圖	60
圖 4-3、.Net 所提供的 XML DOM 階層架構	61
圖 4-4、.Net Framework 中 XML 的設計	62
圖 4-5、Crystal Report 的資料存取模式	66
圖 4-6：LFT 的記錄方式概念圖	67
圖 4-7：取出資訊的方法	68
圖 4-8：加入或拿掉事件資訊的方法	68
圖 4-9：DataClass 的類別圖	70
圖 4-10：LFT 的類別圖	70
圖 4-11：ToolsClass 的類別圖	71

圖 4-12 : DataTools 的類別圖-----	73
圖 4-13 : Simple Controls 的類別圖-----	73
圖 4-14 : 建築管理請照網路作業系統的類別圖-----	73
圖 4-15 : 執照申請 2 的類別圖-----	74
圖 4-16 : PopForm 的類別圖-----	74
圖 4-17 : PanelForm 的類別圖-----	75
圖 4-18 : 元件部署圖-----	76
圖 4-19:系統啟用程序示意整個的詳細安裝流程-----	77
圖 4-20 : 資料轉換程式的設計概念-----	78
圖 4-21:系統開發網站-----	79
圖 5-1、系統畫面及功能表內容-----	82
圖 5-2、建造相關案件編輯-----	83
圖 5-3、雜項相關案件編輯-----	84
圖 5-4、其他案件編輯-----	85
圖 5-5、副項案件編修-----	86
圖 5-6、基本資料編修-----	87
圖 5-7、系統設定-----	88
圖 5-8、建造執照申請入口畫面(一)-----	89
圖 5-9、建造執照申請入口畫面(二)-----	90
圖 5-10、起造人編輯畫面-----	91
圖 5-11、列印「建造執照申請書」畫面-----	93
圖 5-12、列印「起造人名冊(一)」畫面-----	93
圖 5-13、列印「建築物概要明細表」畫面-----	94

# 1. 摘要

關鍵詞：建築管理資訊、資料模型、營建資訊運籌管理技術、標準綱目、資訊標準化、建築物生命週期共享資訊鏈、營建資訊運籌管理

## 一、研究緣起

建築管理行政業務長期以來總被民眾詬病為地方政府施政中，效率始終不彰的一環。這些年來，不論中央或地方，行政業務的自動化，一直是政府努力提昇形象的重點工作之一，建築管理行政自動化更是被優先考慮的項目，大家寄望自動化能使行政作業更透明化、更簡便，各級地方政府大多等不及中央的動作，都紛紛自編預算並委外研發相關軟體。多年發展下來，各自研發的系統，在資料格式與系統架構上，可謂形形色色，各異其趣，經費充裕者遠遠走在前面，經費居次者往往只做部分開發，財政拮据之縣市則甚至仍處於傳統手工的作業模式。各行其是的現象一直未見改善，造成營建資訊孤島的情況[1]愈來愈嚴重，歸納原因，未有效統一建築物資訊的標準化是最關鍵的因素。乍看之下，建築物資訊的標準化推展工作好像遙不可及，緩不濟急，但仔細觀察，如果能畢功於此一役(即建築物資訊的標準化)，則建築管理行政自動化的目標就易如反掌了。

(以下略)

## 二、研究方法及過程

在研究方法上，除了以文獻研究法閱讀分析現有相關技術文獻外，參考國內法規及國內、外學者的相關研究，更進一步以專家座談的方式與建管相關人員進行雙向溝通，建構一個可充分共享的 XML Schema，對於系統的開發更是遵循嚴謹的軟體工程方法，依照需求分析、系統分

析、系統設計、測試、部署、維護等步驟，使用物件導向的軟體程序開發完成。(以下略)

### 三、重要發現

建築物生命週期中以工程期間產生的圖文資訊最多，且標準化及共享需求最殷切，然而因工程界各自進行自動化，資料格式沒有標準化，所以溝通困難。而建築物生命週期的建管資訊，其前後相關性密切，許多資訊的共享性很高，在去年的研究案中已將整個生命週期的建築物資訊用 XML Schema 架構在一起，若能在操作介面實作成功，未來將能幫工程界要標準化資料格式的期待加以實現。(以下略)

### 四、主要建議事項

根據研究的過程及與各縣市政府的訪談中發現，各單位對於建築物資訊標準化的需求十分殷切，要達成建築物資訊的標準化，建議中央應進行兩項重要的基礎建設：

- 協助將地方所有舊有建築物資料依統一標準化格式給予數位化建檔(這是舊資料部分)。
- 中央協助開發建管請照業務的軟體，供全國使用(這是新資料部分)。

如果新資料的產生源頭統一標準化了，舊資料又以相同格式建置起來，則馬上顯現出來的成果就是：全國的建築物資料都能輕易地統合，而且資料的查詢擷取與統計將產生極高的效益。這兩件工程都應該由中央主導並提供地方全額補助，才能要求在全國統一的格式之下完成，這是國家資訊基礎建設該下

功夫的首要工作。至於各地方政府的建築管理業務自動化系統，應該在中央提供一個基本資訊規格要求下，任由地方自行委外研發，較能符合地域性各自不同的條件需求，而且系統維護才能即時又有效率。

本研究就是嘗試研發全國通用的建築管理請照系統軟體，企圖幫建築管理資訊的產生源進行標準化的工程。

# ABSTRACT

Keyword: Building Management Information, data model, C-CALS Technologies, standard XML Schema, Information Standardization, Lifecycle Information Sharing for Buildings

The major objective of the proposed project is the development of a web-based system for building permits application for the whole nation. This work will first collect, organize, analyze, and compare the contents in Building Permits Application Forms all over the nation in order to establish a data model for building information management. Standardization of the information for building management will be conducted using the XML technology. The resulted standard XML schema can then be used as a reference for the government to further develop the nation's Building Management Information Standard, . and Finally, based on the data model and XML schema established above, a web-based system for building permits application will be developed.

建築管理請照網路作業系統之研究（二）

出版機關：內政部建築研究所

電話：(02)2736-2389

地址：台北市敦化南路二段 333 號 13 樓

網址：<http://abri.gov.tw>

出版年月：92 年 12 月

版（刷）次：初版

工本費：280 元

GPN: 1009205190

ISBN: 957-01-6312-7

## 第一章、前言

建築物屹立於空間，往往比人的壽命還長，在資訊時代來臨後，人的資訊與土地的資訊，都以國家基礎資訊建設看待，投注大量人力物力進行建置。相對的，建築物的資訊建設似乎仍遲遲未被重視。而建築物相關圖文資訊大多產生於建照申請時，為了實現建築物生命週期資訊共享的理想，先從建築管理請照作業進行全國標準化與自動化應是迫切需要的。本研究即在研究建築管理請照網路作業系統有關文件部分的自動化。

### 第一節、研究動機

建築管理行政業務長期以來總被民眾詬病為地方政府施政中，效率始終不彰的一環。這些年來，不論中央或地方，行政業務的自動化，一直是政府努力提昇形象的重點工作之一，建築管理行政自動化更是被優先考慮的項目，大家寄望自動化能使行政作業更透明化、更簡便，各級地方政府大多等不及中央的動作，都紛紛自編預算並委外研發相關軟體。多年發展下來，各自研發的系統，在資料格式與系統架構上，可謂形形色色，各異其趣，經費充裕者遠遠走在前面，經費居次者往往只做部分開發，財政拮据之縣市則甚至仍處於傳統手工的作業模式。

不論各縣市目前發展的情形如何，比較令人擔憂的是 - 各

行各業自動化的工程已經發展多年，像戶政系統或稅務系統等，就有令人耳目一新的感覺，唯獨建築管理，似乎還處於戰國時代一樣，各行其是的現象一直未見改善，營建資訊孤島的情況[1]愈來愈嚴重，歸納原因，未有效統一建築物資訊的標準化是最關鍵的因素。乍看之下，建築物資訊的標準化推展工作好像遙不可及，緩不濟急，但仔細觀察，如果能畢功於此一役(即建築物資訊的標準化)，則建築管理行政自動化的目標就易如反掌了。

要達成建築物資訊的標準化，中央應進行兩項重要的基礎建設：一個就是協助將地方所有舊有建築物資料依統一標準化格式給予數位化建檔(這是舊資料部分)。另一件基礎建設就是中央協助開發建管請照業務的軟體，供全國使用(這是新資料部分)。如果新資料的產生源頭統一標準化了，舊資料又以相同格式建置起來，則馬上顯現出來的成果就是：全國的建築物資料都能輕易地統合，而且資料的查詢擷取與統計將產生極高的效益。這兩件工程都應該由中央主導並提供地方全額補助，才能要求在全國統一的格式之下完成，這是國家資訊基礎建設該下功夫的首要工作。至於各地方政府的建築管理業務自動化系統，應該在中央提供一個基本資訊規格要求下，任由地方自行委外研發，較能符合地域性各自不同的條件需求，而且系統維護才能即時又有效率。

本研究就是嘗試研發全國通用的建築管理請照系統軟體，企圖幫建築管理資訊的產生源進行標準化的工程。

## 第二節、研究範圍

建築物從規劃開始到報廢拆除，中間經歷了許多有關「興建」與「營運」的過程，這些過程對建築物的「圖」「文」資訊需求都不甚相同，以往因為建築物的「圖」「文」資訊儲存不易，加上人事更迭，及各階段間沒有「圖」「文」資訊傳承的制約，所以，需要使用這些資料時就再重新建置，且在「圖」「文」資訊持有者之間常存在有資料差異的現象，或是實地建築物與現存有關建築物的「圖」「文」資訊之間的不相對應的現象。

這些「圖」「文」資訊大致可以分成「工程用資訊」與「請照用資訊」(或稱建管資訊)兩種。「建管資訊」主要源自設計單位、施工單位與業主，而都集中存放在政府的建築管理單位，設計單位要申請建造執照，或施工單位要申請使用執照時，會將設計圖說與申請書表一併送到建築管理單位，其中的「圖」「文」資訊在審查通過後將留存一份正本在建築管理單位。所附申請書表之內容包括政府法規所訂定需附的有關描述建築物之資訊。以上儲存在建築管理單位的數位化資訊就是構成「建築物生命週期資訊共享」的基礎[2]。將來建築物的改建、增建、修建、變更用途等，都可能在建築物完工使用的許多年以後發生，屆時，建築物依舊，但可能人事皆非。如果政府建

築管理部門的「建築物生命週期的共享資訊」仍保存當初核准建造執照及使用執照時的請照用「圖」「文」資訊，則縱使原設計單位或原施工單位已不復存在，亦可從建築管理單位的資料庫歸檔檔案中之「圖」「文」資訊快速取得原有資料。

如何讓這些重要的「圖」「文」資訊建置能標準化與數位化就是本研究的初始目標，但因「圖」的標準化與建置是一件更為複雜與龐大的工程，需另行規劃研究。本研究的範圍是聚焦在建築管理請照文件資訊自動化網路系統的建置。

從表面上看來，本研究的範圍與焦點雖然是針對建築管理請照文件資訊的開發，但因建管資訊已可以完整反應一棟建築物在其生命週期中異動的所有訊息，因此，本研究在規劃建管的核心資料模型時，不只是以請照案件視之，而是以一棟建築物的基本文書資訊來看待它，因為從資訊技術的角度來重新看待建築物資訊時，不必受到傳統表格紙張尺度的限制，也不必受限於申請案件性質的不同，而可以以描述一棟建築物相關「人」、「事」、「物」的一體性資訊來看待。因此，本研究案的真正範圍內涵，應該是建構「建築物生命週期的共享資訊」的原始建置機制。所以整個研究範圍就可以分成兩大項重點：

1. 組構「建築物生命週期的共享資訊」的資料模型。
2. 研發「建築管理請照網路作業系統」。

### 第三節、研究步驟

本研究的步驟如圖 1.1 所述，計畫推動情形概述如下：

## 一、蒐集並研讀評估資訊技術文獻

本研究為考慮 XML[3,4,5]相關資訊技術會是未來至少十年左右主流趨勢，故延自去年仍持續在這方面的文獻及如何將其引進到本研究的應用實作上費盡心力。文獻與相關研究的蒐集大致如下：

### ①系統研發工具與 XML Web Services 相關技術文獻

Microsoft 的 Visual Studio.NET 系統，已隨其推出 Windows 2003 而更新，將來.NET 的元件庫會隨 Windows 系統同時建置，使.NET 開發之應用系統直接可用。而研發方面，有關 XML 編輯器的使用已能純熟操作，XML Schema[3,4,5]已在去年完成設計。而今年著重與系統研發有關之工具軟體，及 XML Web Services[9]相關技術的文獻，皆有廣泛蒐集閱讀與整理。

### ②資料格式標準化方面的文獻資料

建築管理請照相關表格資料，雖然內政部營建署有提供標準格式，但在全國各縣市政府引用時，仍有許多異同處，很難歸納出簡單的規則性。雖然去年的研究搜集了全國各縣市使用

的表格，但一方面因有些縣市仍沒提供完整而詳細的資料，一方面相異性表格仍有類似處，故在進行標準化時有不少困擾與難度，今年將考慮先以內政部營建署的表格為準，再以滿足宜蘭縣及台北縣的需求為初步目標來進行規劃。

## 二、訪談及蒐集建管請照文件資訊

國內地方政府機構對於建築管理業務，因都市發展的規模不同而互有差異，對不同規模的建管部門在建管行政業務方面的瞭解，仍持續進行中。

本研究延續以前的研究，至今共訪問了台北市、台北縣、花蓮縣、宜蘭縣、南投縣、台中市、台南市等的建築管理單位，而且，針對地域性的差異進行瞭解探詢有關申請表格和作業流程等的意見，並歸類整理，以供標準化的分析設計時擬具更有前瞻性，考慮較為完備的系統做參考。

## 三、建築物生命週期共享資訊總體架構檢討

建築物生命週期中以工程期間產生的圖文資訊最多，且標準化及共享需求最殷切，然而因工程界各自進行自動化，資料格式沒有標準化，所以溝通困難。而建築物生命週期的建管資訊，其前後相關性密切，許多資訊的共享性很高，在去年的研究案中已將整個生命週期的建築物資訊用 XML Schema 架構在一起，若能在操作介面實作成功，未來將能幫工程界要標準化資料格式的期待加以實現。本研究終極目標在將建築物的所

有可能的資訊都匯整在一起，目前則僅考慮建築管理的請照文件資訊，並將去年完成資料流的分析成果，實作請照作業系統。

#### 四、建築管理請照作業平台系統分析

經過建築物生命週期共享資訊總體架構的分析，然後從中切割出初期需要建置的資訊源，亦即建築管理請照部分進行更詳細的分析，主要針對請照資料流的課題，在進行資料流的詳細分析後，設法找出最佳的資料模型架構，讓資料的共享性達到最理想的目標。接著考慮請照作業平台的問題，並繪出整個建築物資訊化、自動化的新架構及如何推動的進程。由於請照操作的本質，及.NET 的技術改進，原來構想的網路連線操作已稍做調整。

在建築管理請照作業系統方面，首先將全國所有請照表格蒐齊整理分類，並將訪談整理得到而原有表格沒有，但具前瞻性、合理性的資料欄位亦考慮進來，破除傳統既有表格的迷失，重新以物件導向技術進行分析組構，分析成果用UML[11,15,19]繪製，供系統設計依循。

#### 五、建築管理請照作業平台系統設計

經標準化後的建管資訊，利用建造執照與使用執照的證照字號(另外在 Schema 中增加 LFT 屬性來控制資料使用狀態)為鍵值，可以將建築物生命週期中所有申請異動的基本資訊串聯起來，而其背後含有原申請的詳細資料，甚至可經網路反鏈結

到資料的起源單位。這個建築物生命週期共享資訊鏈架構建立起來後，因為具全國性的標準化，不論是地方與中央、跨縣市間的資訊交換與蒐集都變成容易。這個「資訊鏈」不僅對進行中的工程資訊傳輸具時效性，更可貴的是對一棟使用期特長的建築物具有歷史性資訊查詢的價值(因理論上，政府機關會一直保有這些資料)。

今年在實作操作介面時，一邊對建築管理資料模型的核心部分仍做更合理的修正，目前已經用.NET 研發出「建築管理請照作業系統」，並提供宜蘭縣建築師公會的部分會員安裝試用中，並已安裝相關軟體元件在宜蘭縣政府建築管理課的主機上，提供宜蘭縣建築師公會更多的會員安裝與測試。

## 六、縣府端資料連接程式的開發

建築管理請照網路作業系統的開發是以.Net 為基礎的系統架構，其中資料的記錄與操作係以 XML 為基礎。這與以往使用 RDB 的方式有很大的差異，因此本系統上傳與匯出的資料，無法直接為縣府端的程式所接受。所以，必須於資料接收端撰寫資料轉換程式做為映對。

## 七、期末報告與報告撰寫：

完成「建築管理請照作業系統」的系統軟體後，進行期末報告，並彙整專家寶貴意見，做為進一步修正之參考。

#### **八、網站之系統部署測試：**

系統研發告一段落後，系統元件部署在伺服器端，並不斷測試中，目前已提供公開下載測試使用，宜蘭縣的案件已可直接上傳案件資料進行申辦請照作業。

#### **九、系統手冊及操作手冊撰寫：**

系統手冊及操作手冊已陸續同步進行寫作及整理中，預計在年底前可完成本系統初版的手冊，並開放置於系統開發網站提供下載。

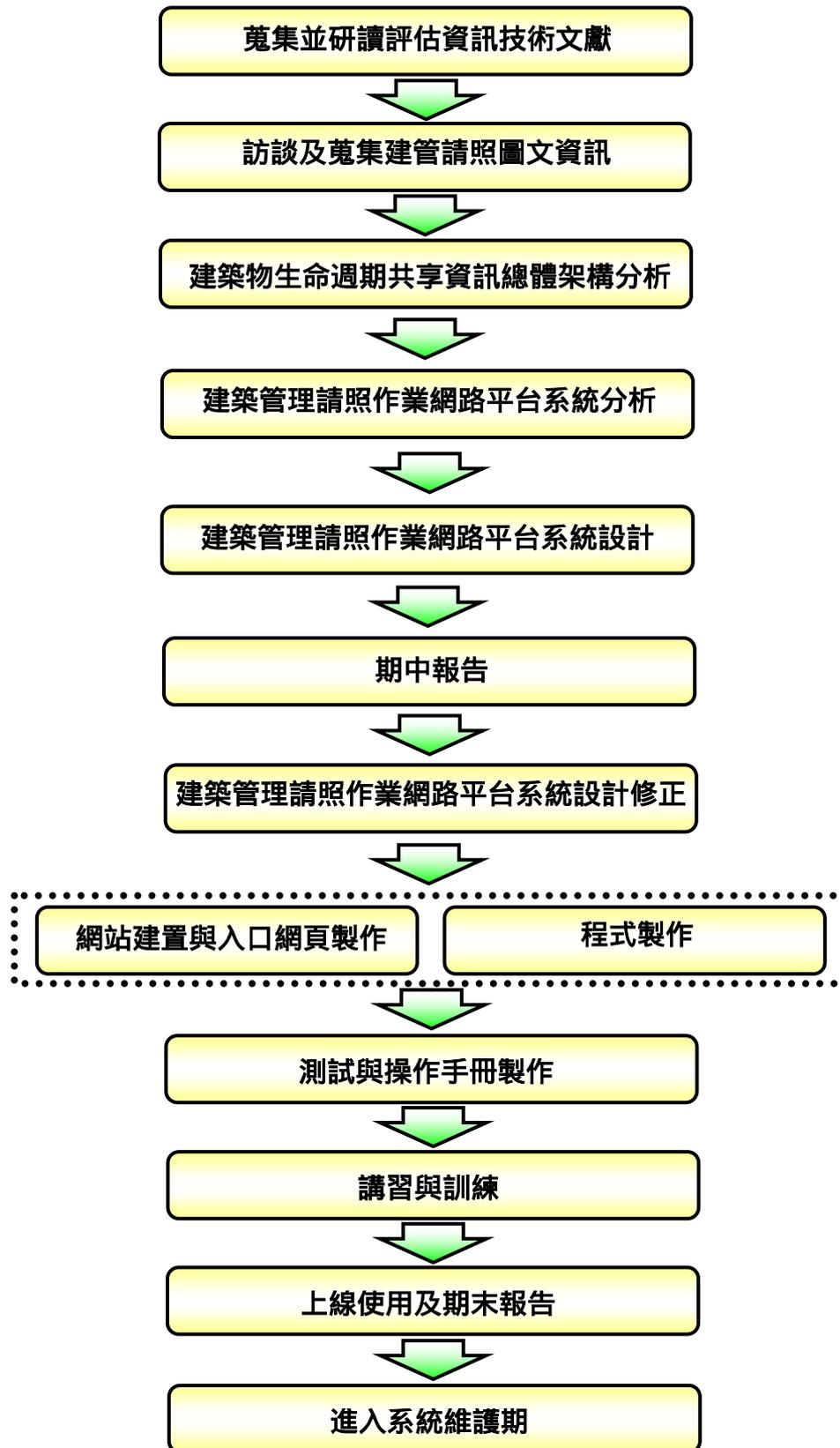


圖 1.1、研究步驟流程圖

## 第二章、文獻整理

本研究除了廣泛蒐集內政部營建署及國內主要縣市之建築管理請照所須之文件表格外，針對資訊新技術在架構建築物生命週期的資訊共享方面，亦不斷的探討其可行性。本研究採用先進的資訊技術－XML 相關標準，及微軟公司最新的系統開發工具－Visual Studio.NET，亦因微軟推出 Windows 2003 而更為篤定。新技術的文獻研讀是本研究頗為吃重的工作之一。本研究在進行之初，同樣的也對相關的研究進行一連串的調查，以確保研究進行的方向正確，以及技術上的可行性與前瞻性。

### 第一節、建築物生命週期

建築物的資訊，不論是描述該建築物的相關文件(例如起造人、座落的土地概要、建築物各樓層的面積與用途等、設計人、承造人)，或是描述該建築物的尺寸與用材的圖說，以目前國內的建管制度而言，這些巨細無遺的資料都會在政府的建管單位留存著。以現有的建管法令，建築物的動態資訊，打從它自平地建起，一直到它的報廢拆除，都要向政府的建管單位報備與申請。理論上，這些資訊應該會持續保存在政府的建管單位，只是以傳統的作業來看，這些資料因存放技術的著眼不同，要

想將同一棟建築物的所有資料在短時間內串聯匯集在一起是件很困難的事，企圖用營建資訊運籌管理技術來解決這件串聯整合資訊的工作，正是本研究的精髓。因此，表面上看來似乎只是在處理建管資訊，但本質上是企圖要建構出建築物生命週期資訊的共享機制。建築管理資訊是建築物生命週期所有資訊的核心所在[1]，有效地掌握這些資訊並提供充分共享，則不論什麼時候要用到建築物的資訊，從建築管理資訊擷取所需，應是值得考慮的。本研究採用文獻中所提出的建築物生命週期資訊共享策略[2]，說明在建管資訊的標準化後，如何進一步地建構出建築物生命週期資訊的共享機制。

## 一、建築物生命週期資訊共享概念

人類聚落的發展，大多依循著建造一座座穩固、美觀舒適的建築物，以方便棲息及活動。而建築物會因為不敷使用、戰爭、天災、破壞、自然老化等等原因被拆除而終止。一般而言，目前極大多數的建築物仍集中建構在人類活動頻繁的都市土地上。同樣的土地空間，可能蓋了又拆，拆了又蓋，週而復始。「建築物」好比一個生命有機體，從誕生、茁壯到終結，有其相當長的活動歷程。在它的生命週期當中，隨著時間，會跟許多的人、事、物構成密切的關係，如何有效掌握它們互動的資訊，是一件值得深思的問題。往昔，科技不發達，資訊的傳遞、交換、延續，有其客觀的限制，因此「建築物生命週期」中與人、事、物間的關聯性，就較難有效掌握。如今，由於電腦軟

硬體的進步，數位化技術日新月異，網際網路的普及，建立上述關聯性及有效掌控，已比以往更為可行。此時以「建築物生命週期」做為探討建築物相關課題的尺度，是適當而有意義的。

人與建築物在整個生命週期中的互動，會產生許多的溝通界面[2]，這些溝通界面是靠「圖」與「文」的資訊來傳達與橋接的。其中，有些「圖」與「文」是隨建築物的生命週期一直都存在而不變的（例如：建築地址、地段、地號、基地地籍圖），而有些則隨著不同的階段而不斷演變（例如：設計圖，設計變更圖等）。由於「建築工程」，是人類社會繁榮進步與文化水準的重要象徵。在其漫長的生命週期中，人民的生命財產、作息活動，都跟它息息相關。一棟建築物的整個生命週期，從規劃設計開始，施工、營運、到報廢拆除，它的相關資訊都應該被詳盡的建檔，並做有效的管理與運用，配合網路技術的不斷進步、普及，整個「建築工程資訊」的建置，近者，可供業主、設計者、營造商查詢與管理之用；遠者，可與業務相關單位(如地政、財稅、消防、水電等政府單位)共享與交換資訊。

建築物生命週期的資訊涵蓋甚廣，大致可以將其分成「興建期」與「營運期」兩個時期，目前國內某些較具規模的縣市政府，都已將原來的單一個建管部門，分割成專司建築物興建期的「建築管理」與專司建築物完工後之「使用管理」兩個部門。而建築物從一開始規劃設計，許多描述建築物的工程

「圖」、「文」資訊就大量產生，建築師為了描述建築物的形狀、尺寸、施作方法、及所需工程經費等等，需繪製工程圖說與製作施工文件、預算書表等，這些圖說文件就是建築物生命週期中的重要資訊之一，然後為配合都市發展與土地有效利用之建築管理的需要，跟建築物附屬相關的資料，例如建築地址、地段地號、起造人、工程造價等等資料都要在申請執照核可時提出申請書表。工程資訊也不是設計完成後就一成不變，例如施工期間，業主隨時會因為資金或需求的改變而變更工程的設計，或是因為實地施工上碰到的困難或技術性問題而變更，甚至將來營運期間也會有修建、增建、改建、室內隔間裝修等行為而變動原有的工程「圖」、「文」資訊。另外，附屬建築物的一般行政資訊也是會不斷的變動，例如施工期間，起造人就有可能變動，甚至連承造人、監造人等的相關資料都可能變動。完工後的使用，建築物所有權人因為買賣或贈與所產生的變動更是經常發生的事，各樓層空間的用途變更等等，以上都是重要的建築管理資訊的一部分。而整個有關建築物生命週期的資訊中，除了門牌整編與土地重劃以外，比較不會更動的資訊是建築地址、地段地號，還有最重要的「建造執照字號」與「使用執照字號」則始終是對應到建築物而不會變動的資訊。

本章內容從建築物資訊的供需問題論起，並以一般性的建築物為對象，從工程的角度，並且考慮政府建築管理制度規定的作業，來探討其在整個生命週期中各種作業行為所產生的資訊與各專業所需求的資訊間的互動情形，企圖理出其規則性，並歸納出可以形成資訊共享的脈絡，然後整理出建築物生命週

期的資訊鏈，再進一步討論資訊共享的可行性。

## 二、建築物資訊供需關係之探討

「工程用資訊」一直到建築物完工啟用後的維護階段仍然有用，建築物的改建、增建、修建，都可能在建築物完工使用的許多年以後發生，屆時，建築物依舊，但可能人事皆非。如果政府建築管理部門的「建築物生命週期的共享資訊」仍保存當初核准建造執照及使用執照時的「請照用資訊」(包括前述審查核准後留存的全數位化「圖」「文」資訊)，則縱使原設計單位或原施工單位已不存在，亦可從建築管理單位的資料庫歸檔檔案中之「圖」「文」資訊快速取得原有資料。

## 三、建築管理資訊鏈

以資料庫技術的觀點來看，「建築物生命週期共享資訊」可以分成幾個層次來看，如圖 2-1 所示，由所有申請案件的收件流水號組成，並以建造執照字號與使用執照字號為關鍵值，構成第一層，就是「建築物生命週期資訊鏈」。在此「資訊鏈」中是靠執照字號及同一建築物申請案件處理時間的先後串聯在一起。

而第二層也是以建造執照與使用執照的核准字號為關鍵值，利用資料庫技術將建築物生命週期中所有證照申請核准的基本資料串接起來的，其中包括建造執照與使用執照的存根資

料。如圖 2-2 所示，所有依「建築管理資訊系統」建置而成的資料，都可以用建造執照字號與使用執照字號串聯在一起，形成「建築物生命週期共享資訊」的第二層資訊。第一層與第二層可以靠案件流水號連結起來。而第三層資訊是整個建管案件申請時所有的「圖」「文」數位資料，其連結方式與第二層相同，只是資料更詳細。至於第四層，是指建置建築物資訊的源頭單位所保存的詳細資訊，將來可配合更進步的資訊與網路技術，以及數位簽章制度，提供連結與共享。

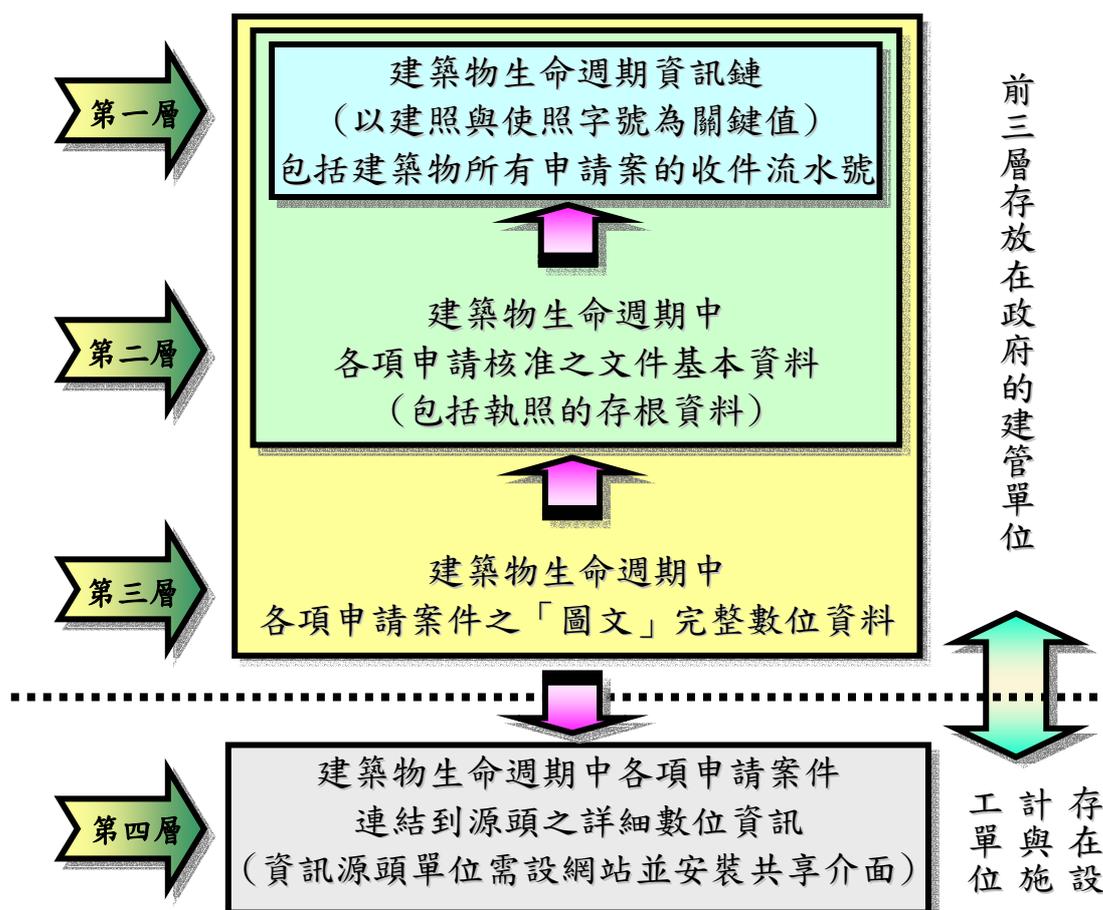


圖 2-1、建築物生命週期的共享資訊層次關係圖

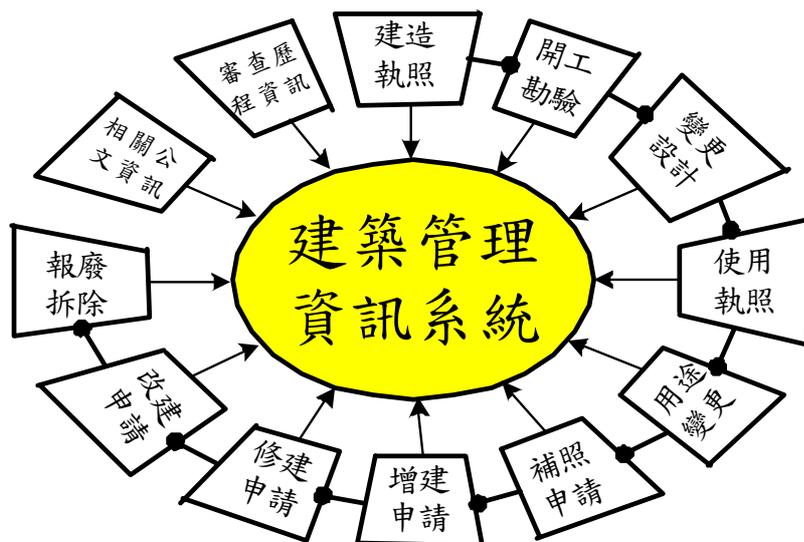


圖 2-2、建築物生命週期的資訊鏈

## 第二節、XML、XPath、XSL 及相關技術

### 一、XML

XML 的基本觀念，簡言之就是文件內容由一組元素單位所構成。這些元素單位係以一系列的標籤 <tag> 加以界定，XML 就是附帶一系列標籤和夾雜資料內容的文件。標籤的使用，類似 HTML，然而由於 HTML 的設計只為了文件在瀏覽器上的呈現，其標籤是固定的，而 XML 的標籤是可以自行不斷延伸及定義的。XML 中的 "X"，所指的就是延伸的可能性。

目前 XML 語言的應用已是資訊界的趨勢，尤其在企業界，企業的資訊系統在應用上，常需要從各個不同的地方收集或分散各種資料，這些不同的系統彼此互相連接，彼此交換及共享資料。而企業之所以使用 XML，最主要是因為 XML 提供了下列兩大功能：

1. 不同應用系統間的資料分享與溝通。
2. 能自訂資料的表現方式。

企業資訊系統發展已有多多年，但在過去的發展當中，不論是採用單層式或多層式的架構，系統和系統之間卻很難進行溝通，而造成一座座的「資訊孤島」。然而，隨著企業的大型化及電子商務的發展，建立一套系統與系統之間能交換資料的方法，變成是一種迫切的需求。在以往同一供應鏈的企業使用 EDI 的方式彼此交換資料，但 EDI 仍是一種封閉式的交換方式，而且所需的軟、硬體設備成本高昂，並非一般企業所能接受，而 XML 的提出，無疑的讓電子商務的未來充滿希望。XML 有別於 EDI，因它是一種開放式的架構，兩系統之間只要依循相同的結構定義(DTD, XML Schema)，彼此便能互相交換資料。因此，和 EDI 相較之下，顯得相當的有彈性。而且，使用 XML 並不需要額外的硬體成本，只需要一個 XML 的解析器，就可

以進行 XML 的解析工作，這也是 XML 另一個誘人的地方。

雖然，XML 有許多的好處，可是要把現有的資料庫及資訊系統轉換成 XML-Based，確實有相當的困難。但是要让現有的資料庫及資訊系統「懂 XML」，卻不是一件困難的事。XML 資料是高度結構性的，從簡單的單一元素，到複雜、多層次的結構。透過解析器的轉換，可以將現有結構化的資料，轉換成 XML 的格式傳送，同樣的，通訊之彼方也可以將之轉換成另一種可接受的型式，而達到資料分享與溝通的工作[10]。

XML 尚可以透過 XSL 處理器及格式化物件(Formatting Object)，將 XML 資料轉換成各種客戶所想要的形式呈現，使得資料的管理顯得非常有彈性及效率，同時這也是 XML 在電子商務上達到客製化的重要技術。

XML(eXtensible Markup Language)語言是當今世上被公認為描述資料模型較理想的語言，它是一個可自行描述資料的語言。本研究採用 XML 作為建築物生命週期共享資訊模型的主要原因有：

- a.XML 具物件導向的特質：在描述建築物生命週期中各項資訊的關係及建構資訊模型時，更能切合實際。

b.XML 符合網際網路傳輸協定：XML 即專為資料要在網際網路上跨平台互相交換而發展出來的。因此，XML 文件可以輕易的穿透防火牆，而不用修改現有的網路安全政策。

c.XML 只描述資料：XML 是一種中性的母語言 (Metadata)，除了能表達資訊的語意以外，不需額外的訊息。

## 二、 XML Schema

如前一節所述兩個資訊系統間只要使用相同的結構描述，便可以互相進行資訊溝通，由此可知相同的結構描述是溝通的基礎，結構描述是供資訊溝通進行中的一個驗證機制。早期在 XML 1.0 的時候，文件的結構描述是透過 DTD 來進行，DTD 雖然提供了強大的基礎，但其本身並非根源於 XML，這也使得其在運用上產生缺陷。所以，W3C 制定了較 DTD 更完備的 XML Schema 定義語言[4,5]，XML Schema 除了提供原來 DTD 的功能之外，XML Schema 本身也是一份遵循 XML 語法格式的 XML 文件，所以 XML Schema 本身也可以接受驗證。

其實資訊科技早就運用到 Schema 的技術理念，例如資料庫的運用。在以往格式的檢查上大都必須經由應用程式本身進行驗證，但這造成了資訊的無法共享，因為除了所使用的特定應用程式外，沒有人會知道它的文件格式為何。XML Schema 的功能即在此透過使用相同的 XML Schema，透過適當的宣告及解析器的運作，不同的應用程式可以知道，並對不同的文件進行查驗工作，進而構成資訊的互換及共享。XML Schema 的運用，也是本研究在進行時的重點。以下即為本研究所建置的 XML Schema 的一小部份：

```
<?xml version="1.0" encoding="utf-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="" id="base">

  <xs:complexType name="日期">

    <xs:sequence>

      <xs:element name="年" type="xs:gYear"/>

      <xs:element name="月" type="xs:gMonth"/>

      <xs:element name="日" type="xs:gDay"/>

    </xs:sequence>

  </xs:complexType>

  .....
```

除此之外，XML Schema 規格中尚包含了在 XML 文件中關連至不同元素或屬性的能力，就像關聯式資料庫透過主鍵與外來鍵的限制提供了聯結資料表的能力一樣。這些功能是透過 unique、key、keyref、selector、field 等元素來達成。

### 三、XPath

如果想要利用 XML 做事，第一件事就是必須要找到所要的資料，而 XML 中 XPath[6]是擔任此一功能的重要角色，在 XPath 尚未發展之前，所有 XML 的應用都只是在資料描述上。因此，XPath 是 XML 在實務運用上，相當重要的一環。利用 XPath 可以找到 XML 文件中的文字資料、元素、屬性、和其他資訊。XPath 是一種精細而複雜的語言，提供詳細的資料查詢機制，它將成為 XML 系統裡的基石。XPath 除了提供資料查詢定位的功能之外，它與 XSL[7]的運作，有很大的關連性。在文件樣式套用的過程中，合理的做法是先要找到被參考到的那一章節之後，再將樣式表套用上去，而「找」的過程，就必須借助 XPath 的功能了。另外一方面，如果要對一份 XML 文件進行運算，也必須要能「找」到相關的資料，才能進行運算。由此可見，XPath 在 XML 的運用上是非常重要的。了解 XPath 對於本研究在進行系統實務設計時，對建築物資訊的查詢及格

式化工作，將有很大的助益。

在 XPath 的運作上，XML 文件係被視為一個家族，其中有父元素也有子元素，父子之間有繼承關係存在，透過對家族的分類，XPath 可以在 XML 文件中進行巡覽及定位，以找到文件中的特定資訊。而其路徑的表示方法，和 Windows 或 UNIX 的路徑表示法是非常類似的。例如：

```
/建築物/id/起造人/*
```

表示建築物節點下的 id 節點之下的起造人節點下的所有屬性。餘此類推，同時 XPath 亦提供許多的運算函數，使得 XPath 的功能就如同一個小型的程式語言一般，關於 XPath 的文獻相當多，詳請參照 W3C 公佈的相關文件。

#### 四、XSL 與 XSLT

在 XML 發展的過程中，W3C 的工作群組發現，要讓 XML 發揮其實務性的功能，就必須要有將 XML 文件轉換成不同展現格式的能力。這造就了後來 XSL[7]及 XSLT[8]的發展，到了 1999 年 11 月，XSLT 1.0 版正式通過 W3C 的建議案，現在大部份的 XML 解析器，都提供了 XSLT 的完整支援。而整個 XSLT

的轉換程序如圖 2-3 所示：

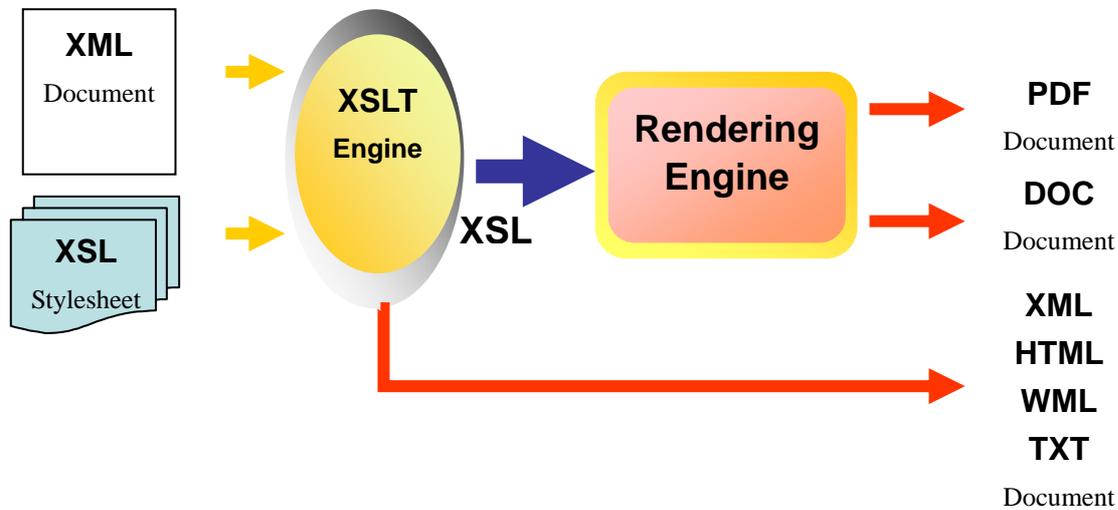


圖 2-3、XSLT 的轉換程序

圖 2-3 中可以看出 XML、XSL、及 XSLT 在資訊轉換過程中各自所扮演的角色。XML 配合所需的樣式表定義經由 XSLT 的轉換可以直接轉換成 XML、HTML、WML、TXT…等文件。如果要轉成印刷格式，則可以再經由格式化物件(FO)及 Rendering Engine 直接轉換成 PDF、DOC…等文件格式。

在 XSLT 中同樣提供了許多的元素，這些元素甚至提供了判斷及迴圈的功能，再配合 XPath 的運作，儼然像一個小型程式語言。而本研究中，XSLT 同樣扮演著重要的角色，在各種動態的元件設計，如文件物件的轉換，輸出報表文件的產生等均採用 XSLT 來達成。

### 第三節、UML 系統分析語言

統一模塑語言 (UML) 是一個典型的視覺化物件導向分析設計之模塑語言，用來指定、呈現、建構和記錄系統化流程中的各種元件，以對軟體進行描述、視覺化處理、構築和建立軟體系統元件的文件。它利用一種統一的圖形表示方法以捕獲對欲建構的系統之各項決定和瞭解，可用於對系統的理解、設計、瀏覽、配置、維護和資訊控制。UML 適用於各種軟體發展方法、生命週期的各個階段、各種應用領域以及各種開發工具，它是一種總結以往模式化技術的經驗，並吸收當今優秀的實作成果的標準模式化方法。UML 涵蓋了概念的語義、表示法和說明，提供了靜態、動態、系統環境及組織結構的模型。它可被交互的視覺化工具所支援，這些工具提供了程式碼產生器和報表產生器。UML 標準並沒有定義一種標準的開發過程，因此，它特別適用於反覆漸進式 (Iterative and Incremental Development Process) 的開發過程。它是為支援大部分現存的物件導向開發過程所設計的。

UML 描繪了關於系統的靜態結構和動態行為。在物件導向的世界中，一個系統可以視為由一些離散且相互作用的物件所

聚集而成的模型結構，其最終有益於外部使用者所用。靜態結構用來定義對系統相當重要的物件屬性和操作，以及這些物件之間的相互關係。動態行為則定義了物件在相對於各時間點所發生的事，和物件為完成目標而相互進行的通信機制。從數種不同但相關的觀點所建立的系統模型，容許系統為各種不同用途所瞭解。

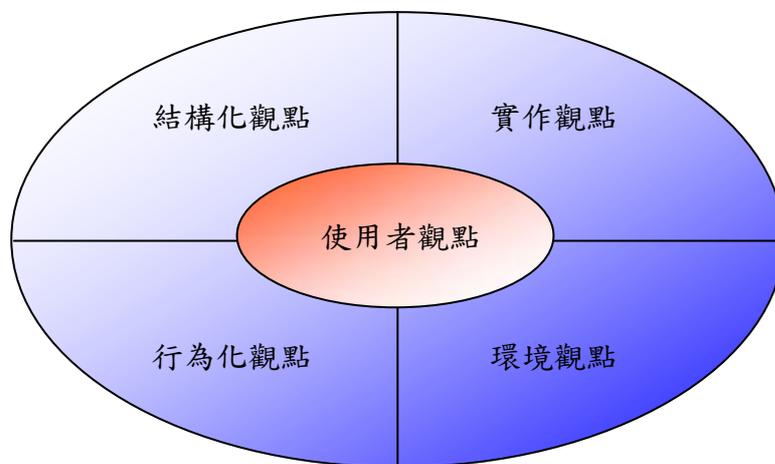


圖 2-4、 UML 五個不同的模型觀點

圖 2-4 表示，UML 提供了五個不同的觀點，以簡化問題解

決過程。這五個不同的觀點分別為：

- 使用者觀點：這個觀點主要利用 Use Case 圖，從使用者的角度來說明使用者的目標及需求。透過這個觀點，系統分析者可以有效的取得使用者的需求，並利用這個簡單的圖形工具，與使用者進行良好的溝通。
- 結構化觀點：本觀點主要的工具為類別圖和物件圖，本觀點主要的目的在利用這兩個圖形工具，針對系統的靜態架構，進行結構化的描述。
- 行為化觀點：顧名思義本觀點意在描繪系統元件的行為特質，主要的圖形工具為循序圖、合作圖、狀態圖、及活動圖。
- 實作觀點：在說明系統在實作結果的結構化及行為化觀點，主要的工具為元件圖。
- 環境觀點：即系統在現實環境中的佈署情形，主要的圖形工具為佈署圖。

UML 還包括有組織的將模型化為套件，以便於軟體小組將大的系統分解成易於處理的塊狀結構，並去理解和控制各個套件之間的依賴關係，以及在複雜的開發環境中管理模型單元。它包含陳述實作這些決定的概念及組織運行時的單元並轉化成元件。

UML 雖然不是一門程式設計語言。但是它和物件導向的程式語法間，是高度相關的。因此，可以透過程式碼產生器這類

工具將 UML 模型轉換為各種程式設計語言，或透過逆向工程將既存的程式代碼轉換為 UML 模型。這使得 UML 的分析工具和傳統的整合開發環境(IDE)有所區別，如果我們將 CASE 依軟體生命週期區分為分析設計和設計整合及測試兩大類的話，UML 是屬於分析設計，而 IDE 則是屬於後者。應用 UML 有利於系統開發人員專注於分析階段，因為所有的工作在分析階段完成的同時，軟體的程式碼也接近完成，不僅可以快速的開發，許多技術性及可重用性的問題，更可在分析工作進行的同時予以解決，使得軟體的架構更加完整，這在軟體服務範圍的日漸廣泛及體積的大型化上，是十分重要的。

由 Schema 的特性，要由 UML 的類別模型中產生 XML 的文件架構是可行的，但必須遵循一定的準則，而這正是 OMG 的 XMI(XML Metadata Interchange)所要規範的。在其產生的規則中，根據 Carlson 建議[16]，有 9 條準則必須要注意。

- a. Namespace 的轉換：整個 UML 模型是要轉換成一個 XML Namespace，還是要將每一個 UML 套件轉換為不同的 Namespace。
- b. 元素名稱的唯一性：給定所有 Namespace 轉換與目標文件結構語言後，如何確保所產生的 XML 元素名稱的

- 唯一性。
- c. 元素或是屬性：要將 UML 的屬性與關聯角色轉換成 XML 的元素或屬性。
  - d. 個體數的限制：要如何將 UML 的屬性與關聯角色這兩者的個體數轉換為 XML 文件結構。
  - e. 繼承：如何將 UML 的繼承轉換成 XML 的文件結構定義。
  - f. 內容模型：在定義 XML 元素時，要使用什麼樣的 XML 內容模型來控制元素與文字出現方式 (empty、textOnly、或是 elementOnly)，與元素的群組結構 (sequence、choice、或 all)。
  - g. 元素的順序：如何將無順序的 UML 屬性與關聯角色轉換為 XML 元素的內容模型。
  - h. 資料型別：如何將 UML 模型的資料型別轉換為 XML 元素或屬性的資料型別，那一種限制比較多。
  - i. 連結：如何將 UML 的關聯轉換為在兩個文件之間相連的 XML 連結。

以上 9 點是 UML 在轉換成 XML Schema 時應注意的重點。

## 第四節、.Net 架構探討

### 一、.NET 的基本理念

Microsoft .NET 是微軟公司新近推出其所有軟體技術的核心基礎，它的基本理念，就是將網際網路的技術更全面地跟原有的電腦軟硬體技術做整合，以提供更佳的電腦、裝置、及服務系統。電腦與網際網路技術流行多年後，人們已經可以完全掌控資訊從何時、何地、及如何傳遞給自己，第一代的網際網路技術讓客戶端能獲取伺服器端資訊，第二代的網際網路技術讓客戶端與伺服器端能經人的操作互換資訊，.NET 技術則進入 Web Service 的第三代。

Microsoft .NET 協助策動網際網路上的新轉變，未來將會有更多由 HTML-based 資訊將轉而由 XML-based 代替。XML 是一項目前被廣泛支援的工業標準，其規格由 World Wide Web 協會制定，Web 瀏覽器標準也是由此組織制定。雖然微軟大規模地採用 XML，但它卻不是微軟的技術專利。XML 提供了區分資料本身及資料簡報型態，這是下一代網際網路重點，提供非閉鎖資訊，並可以將其組織化、程式化、與編修；以較有效率的方式傳送資訊於不同數位設備上；更允許不同網站間相互結合、互動的效果，以提供多樣化的網路服務，為使

用者帶來全新的網路應用經驗。

## 二、為何 .NET 如此重要？

.NET 之所以對使用者相當重要是因為它讓電腦更易於使用，並具備更強的功能。值得一提的是，它擺脫了以往硬體的限制：讓外務使用者的資料存在於網路，而非手提電腦中。我們可以透過利用各種平臺，包括桌上型電腦、手提電腦、手機或掌上型裝置(PDA)來達到即時傳遞資訊的目的，更可將這些資訊整合於應用程式中。.NET 能讓使用者輕鬆地連繫並完成交易，免去耗費力氣及重覆鍵入資料的挫敗感。藉著將多種安全性資料整合入單一使用操作介面的方式，或者甚至是一種程式化的自動決策引擎，.NET 架構將使用者從目前網頁應用的資料限制及技術樊籠中跳脫出來，讓使用者可以隨心隨欲地透過任何裝置，在任何時間、任何地點自由地存取資料。

.NET 也對開發人員同樣重要，因為它不僅改變未來應用程式的開發方式，更允許開發人員可以創造出全新的應用程式。這項新發展的核心即是網頁服務(Web Services)的概念。由於架構在網際網路的開放標準 XML(Extensible Markup Language) 之上，透過 SOAP(Simple Object Access Protocol)[10]協定，使得網頁服務成為網際網路上的一項多功

能整合應用服務。

過去，開發人員總要藉助整合本機系統服務來建構應用程式。這種模式使得開發人員需要使用大量的程式設計資源，並要精準的控制這些應用程式的動作。今日，開發人員正在架構一種複合性的多層式架構系統，並在網路上整合所有應用程式，然後創造獨到的價值。這使得開發人員需要關注所要提供的特殊服務而非整個系統的重新建構，這樣的結果將節省產品到市場上的時間、達到更高的開發產能、以及最終的目的：高品質的軟體。

我們正邁入下一新階段的電腦運算階段：一個由網際網路所觸發的階段，此階段充分地運用新的網際網路開放標準技術—XML。藉由創新的軟體技術，XML 允許創造出可讓任何人或在任何地方使用的強力應用程式。它增加了應用程式的取得及與軟體的持續連結。在這種方式中，軟體是一種服務，就像是呼叫服務或付費欣賞的電視節目，都需要經由通訊媒介取得資料。軟體網路化的兩項特質，一項為更具彈性的多層次(n-tier)[11]的運算方式，另一項則是網頁上的訊息導向概念，亦即「網頁服務(Web Services)[12]」，這象徵軟體程式發展的新里程碑。網頁服務(Web Services)，就是利用網際網路的標準協定，例如：HTTP、XML，將軟體的功能完全地展現在網際網路或是企業內部網路上運用的軟體服務，它可以想像是一

種在網頁上的元件編製程序。就概念上而言，程式開發者藉由呼叫應用程式介面（APIs）將應用程式整合於網頁服務(Web Services)中，就像呼叫本機服務的意思一樣。這之間的差別在於網頁服務呼叫可以透過網路來服務位於遠端的系統。例如，Microsoft Passport 可使程式設計者規劃出申請手續的認證。藉由 Passport service 的程式，程式設計者可以利用 Passport 的架構，依靠 Passport 來維護使用者資料庫，確保其正常運作，及資料備份等等。

.NET 即是架構在這樣的網頁服務(Web Services)的原則下，微軟正透過完整.NET 架構平台(包含各項組成部分)，以提供網頁服務(Web Services)穩固的基礎開發架構。下一代的程式發展工具與基礎架構，其中包含 Visual Studio .NET、.NET Framework、Windows.NET 及 .NET Enterprise Services 等，都是設計用來在網頁服務模組上開發應用程式。此外，.NET 區域服務建置(.NET Building Block Services)、新的.NET 裝置支援(.NET device support)、及和即將到來的 .NET 使用者經驗(.NET user experience)三項設計則將提供完整的開發工具及經驗，讓接續的應用程式開發可以真正地利用網頁服務模組所賦予的技術。

## 第三章、建築管理請照資訊系統分析

### 第一節、初步規劃

本研究為打破建築管理資訊之傳統思維，以建構一個可共享的建築物生命週期的資訊鏈為主要設計目標，除了對傳統建築管理資訊全面檢討與解構外，企圖建立一個可長可久，且具彈性擴充潛能的資料模型，也是本研究的追求目標。

本研究首先以內政部最新公佈的申請表格為基礎，利用圖解的方式加以剖析(如圖 3.2)，由圖中可以得知資訊重複使用的情形十分普遍。單以建造執照申請為例，起造人及建築概要等資訊就被許多的表格所引用，而這些資訊在整個建築物生命週期之中，也同時為其他活動所需。以目前的現況，這些資訊在不同的系統之中，由於資訊無法共享，而不斷地被重複建立。所以本研究特別將之以人、事、物[13]等建築相關資訊的抽象特性，將建築資訊予以分類，其分類的概念如圖 3-1 所示：

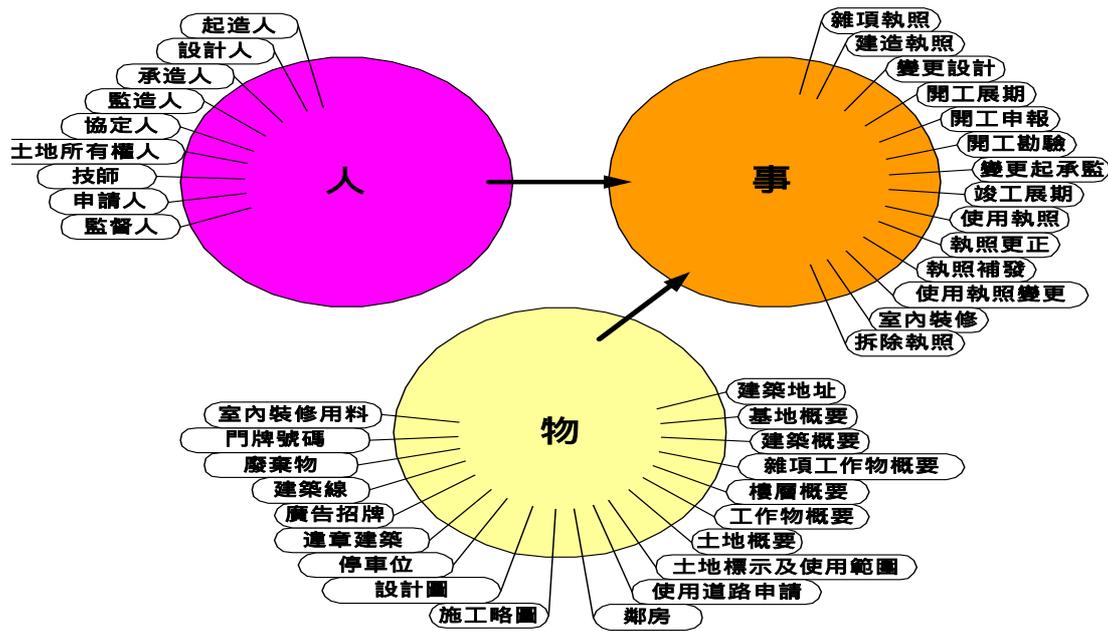


圖 3-1、 建管資訊相關人、事、物的分類觀點

由圖 3-1 可以看出建築物的一般性圖文資訊，可以被拆解為人、事、物三個主要的抽象類別。易言之，整個建築物的資訊可以由人、事、物三個大類別的資訊所組成。其中「人」和「物」的部份可以明確的被定義，雖然其在建築物生命週期之中的資料內涵可能變動，但是它組成的格式與屬性是穩定的，幾乎舉國一致的，故可歸納為靜態資訊。而「事」的資訊，指的是案件屬性（而非直指建築物的屬性）的必要描述，它跟處理該案件之時空資訊有關，亦即與註記該案件之必要的行政資訊有關，這些資訊常因時因地而異，因此規類為動態資訊。

「事」是建築物生命週期中所有「行為」的記錄，例如建造執照申請、開工、勘驗等，這些行為在建築物的生命週期中可能因時空變動而產生「質」與「量」的增減，而且，圖 3-1

僅以建築管理的角度來條列「事」的種類，事實上，戶政的門牌號碼編碼、地政的房屋權狀、稅捐單位的房屋稅籍、水電及銀行等跟建築物都會有「事」的資訊構成；這些「事」的進行，不外乎都需要描述一棟建築物之「人」與「物」的靜態資訊，如果當今這些資訊因其中的某些「事」的進行而變動，卻沒有在相關單位即時更新，那麼持有舊資訊的單位，其資料庫就變成無意義了。而縱觀整個建築物生命週期的「行為」可以發現，許多重要的事件仍跟建築管理最有關係，建管單位擁有建築物最多的資訊應該無庸置疑，是故前述需要建構建築物生命週期正確的共享資訊，而由建築管理機關來擔任應該是最可行的。由此可見，「事」的資訊是屬於動態的，因時空不同會變動的。

事件的發生與政策的施行，通常具有很高的關連性。因此，事件發生的資訊，對於政策的制定而言，是有很高的價值的。如能創造資訊透通的機制，將來對於政策制定的決策支援系統的建立上，更具有莫大的意義。

而由建築物在其整個生命週期的活動進行上，可以得知，由建造執照的請照活動開始，大多數建築物的資訊便被建立起來，接下來的許多資訊，幾乎都以此為基礎而繼續延展增減與變更。因此，整個建築物資訊的建立，如能由建造執照的請照階段便予以標準化，對於日後的資訊共享，將可收事半功倍之效。所以，本研究的基本目標除了建立一套標準化可共享的資料架構之外，建構一套建築管理請照系統，亦是本研究的目標之一。

由圖 3.2 可以看到，右邊為建造執照目前需要的申請表格，而左邊係將這些申請表格中，以「人」、「物」、「事」為主要分類基礎，將其拆離而重新組構；例如「起造人」、「建築地址」等，右邊各表連到左邊的箭頭表示該表的組成資料群有那些。由箭頭集中在各資料群的數量就可以看出它重複的次數多寡；由此可見，一個建造執照的申請，從傳統表格式的作法與思維，前進到用資訊技術的角度來看它時，必要去重新組構，以利未來建築物資訊的提供共享，讓這些資訊有更寬廣及更有效率的用途。

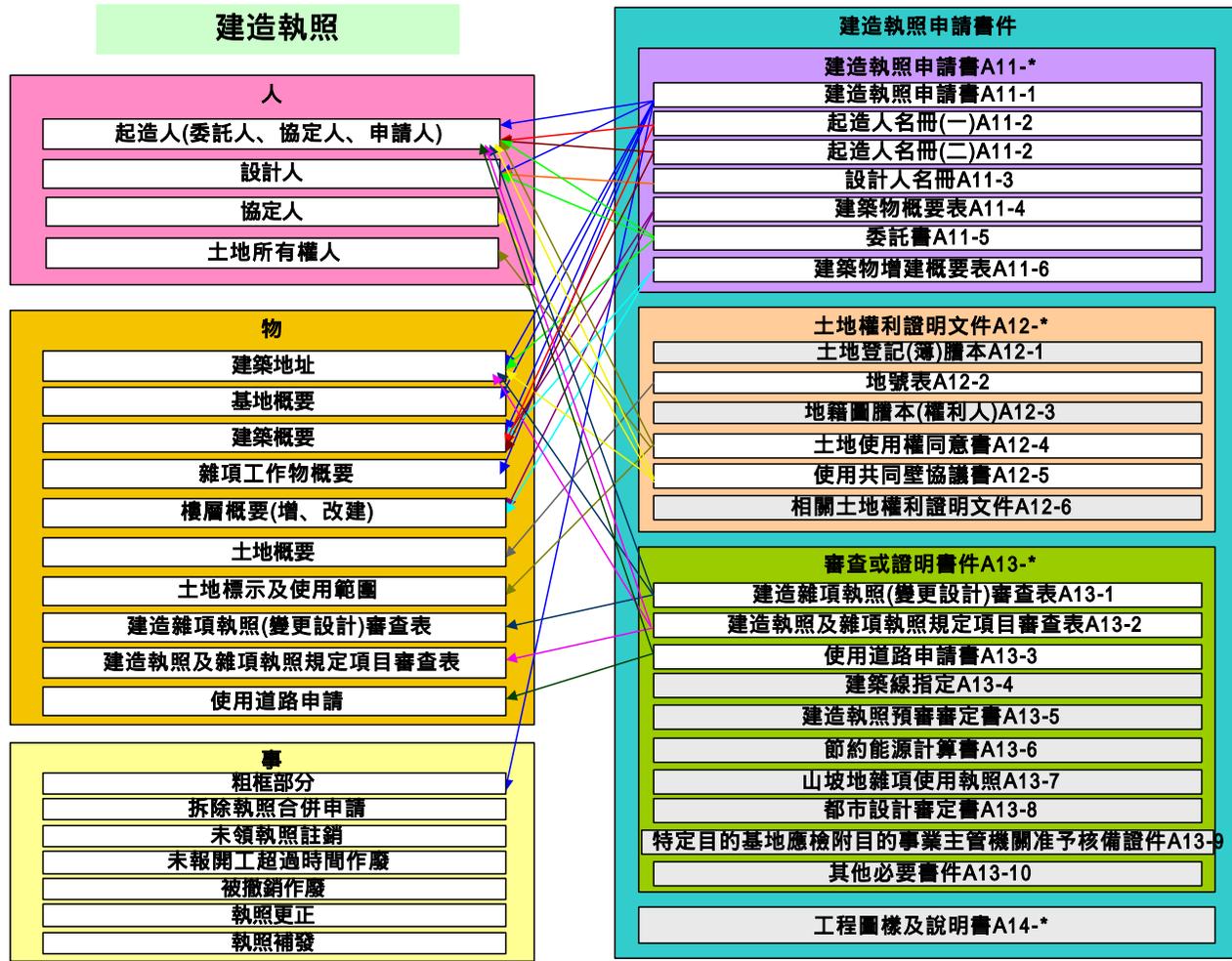


圖 3-2 、 建造執照文件資料解析

## 第二節、需求分析

關於建築物生命週期資訊鏈的形成，在過去的研究中已提出許多詳細的探討及可行性[13]。本研究的最終目的雖然是以建構一套建築管理請照系統，但實際的重點卻是著重於建築物生命週期資訊的建立與共享。因此，資料模型的建構是為本研究的核心，而建築管理請照系統只是整個建築物生命週期資訊鏈建構的引子，也可當成實作的舉證。

本研究旨在建立一套具有擴充性的資料模型，XML 的相關技術，正好可以滿足本研究的需求。而 Microsoft .Net 技術的推出，它所提供的 XML 相關研發資源，更令本研究小組振奮。所以，本研究企圖利用 XML 及 .Net 等新技術[14]，來構築一個建築物生命週期資訊共享的機制。利用 XML 的特性，將傳統資訊系統必須在程式技術層級解決的問題，降到資料組構層級，並使資料的模型更加完整，有意義，且更具有延展性。而藉由 .Net 架構所提供的新世代的系統開發技術，使系統能透過網路快速佈署，並利用網路的運作，實作出一個資訊共享的環境。

本系統在需求方面，主要是在資料模型的建立，其次才是建築管理的請照系統。所以，對於各種建築相關資訊及文件的收集與分析佔了相當大的比例。而在建築管理請照系統的需求

分析上，除了要滿足建築管理上的請照需求外，如何與建築物生命週期資訊鏈產生關連，以達到資訊共享的目的，將是本系統在建置上的一大特點。

### 第三節、使用案例說明

使用案例圖是 UML(統一模塑語言)[15]以使用者觀點來描述系統需求的主要方法，以使用者的角度表示出對於系統的目標與期望[16]。在此我們將整個建築管理請照系統，依其流程特性加以分析得到如圖 3-3 的使用案例圖(資料模型的建立，將在下一章探討)。再配合建築管理請照過程中的各種活動，以行為觀點的活動圖(圖 3-4)來表示其間的行為關係，兩者交互參照，將可對整個建築物生命週期中建築管理請照流程的各項活動，與使用者的需求，有更進一步的了解。圖 3-4 只是應用活動圖表示流程的方法，並未涉及到程式的邏輯。

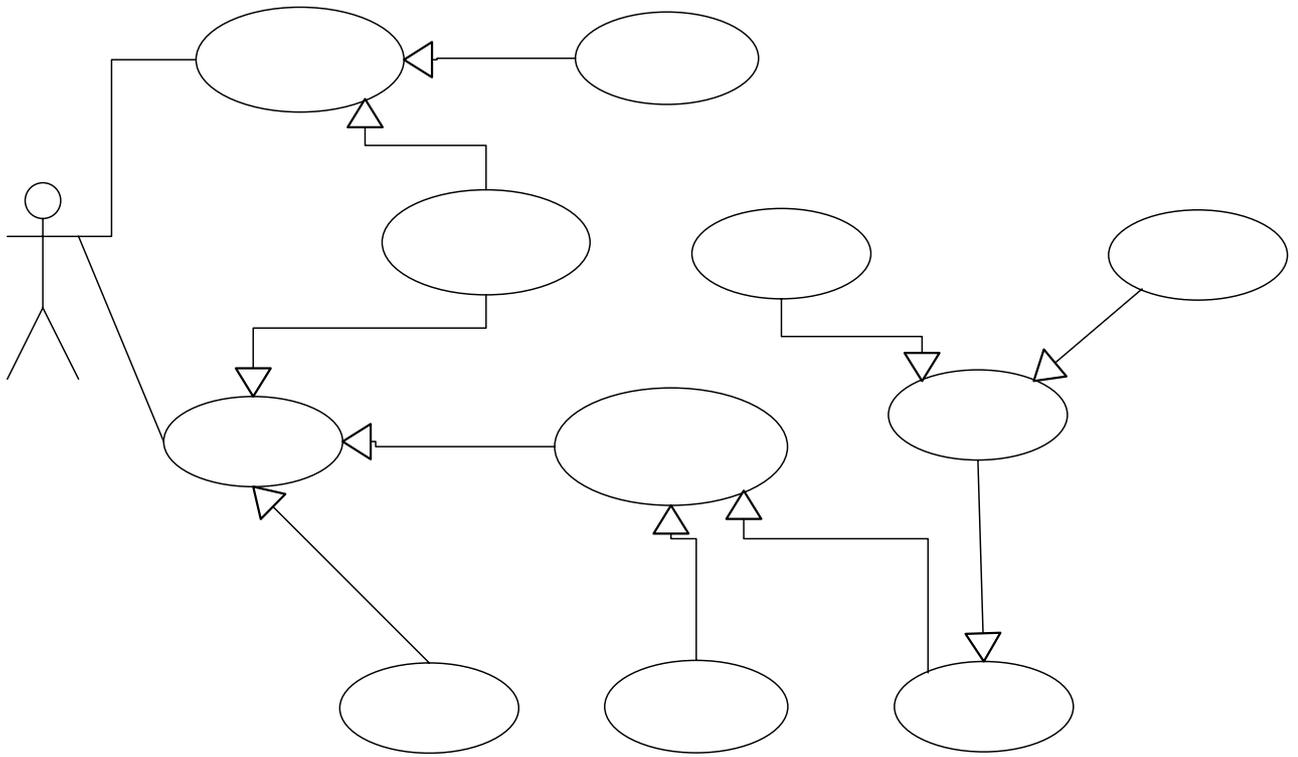


圖 3-3 、 建築管理請照系統的使用案例圖

使用：

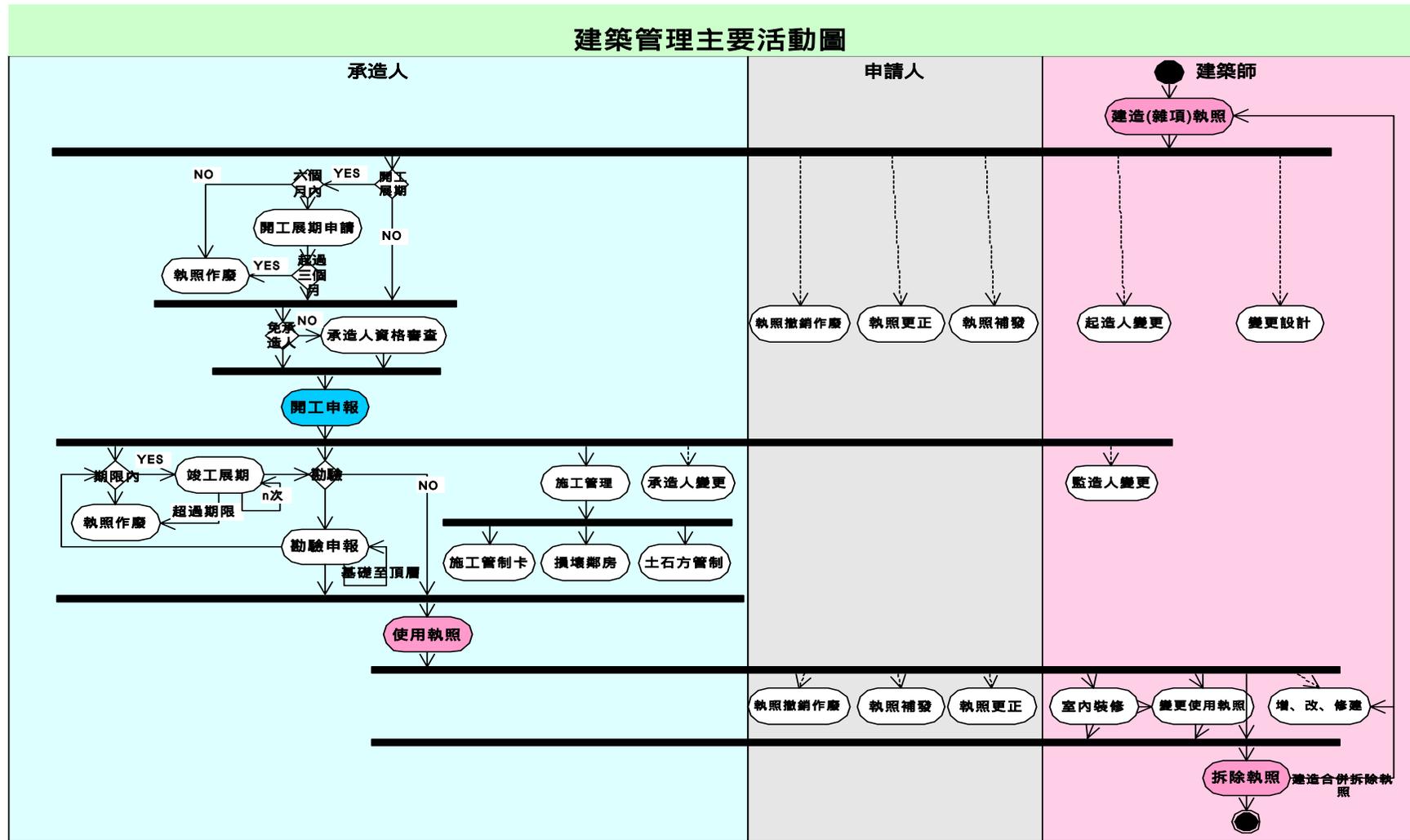


圖 3-4、建築管理相關活動圖

由圖 3-3 中可以概略看出在整個建築管理請照系統中所要執行的工作，其中與系統直接相關的角色，有業主(起造人)、設計人、承造人、及建管單位。在這裡以申請建造執照為例，說明圖形所代表的意義。

【角色】起造人、設計人、建管單位

【型態】主要的

【描述】在一棟建築物生命週期的開始，起造人得委託設計人向建管單位提出建造執照申請，並詳附各種相關資料及審核書表。而在這個使用案例中，部份案件係屬於雜項工作物的範疇，即改申請雜項執照；因此，建造執照也可能換成雜項執照的申請。

在建築物的興建階段，各種活動之間本有一定次序的行為關係存在，因此以物件導向技術的活動圖來加以表示。圖 3-4 即表示建築管理行政中較重要的執照申請審查發照活動之間的順序及控制流程的關係。而三個不同的隔間(游泳水道)，正表示三個不同角色所進行的活動及其需求。

接著，進一步以圖 3-1 - 建管資訊相關人、事、物的資訊分類理念，重新拆解組合而形成圖 3-5 之建管請照資訊類別圖。圖中實際上已隱含一棟建築物的各項執照申請（包括建造執照、變更設計、變更起承監、使用執照、開工、勘驗、補照、

變更改用途等)所需之大部分資料模組。

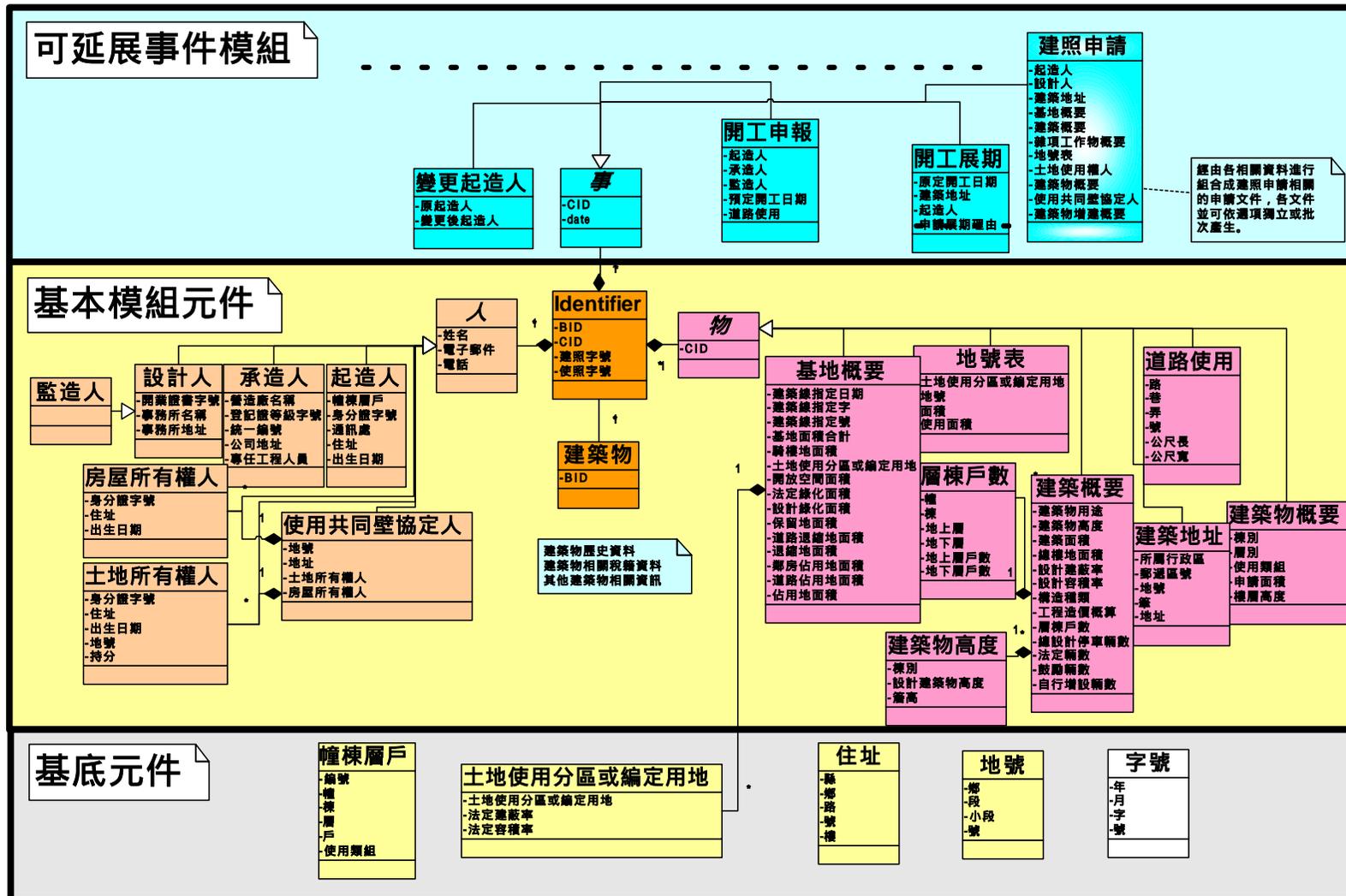


圖 3-5、建管請照資訊類別圖

## 第四節、系統規格

系統規格為一種針對使用者需求分析之後的結果，予以彙總並將使用者的需求詳細定義而得的文件報告。通常，在這裡會記錄兩種型態的資訊。一是使用者對系統的抽象描述；一是詳細的描述系統功能。前者可謂系統概觀及非功能性需求的描述；而後者則為功能性需求的細部描述。接下來即針對本系統的非功能性需求作一描述。

非功能性需求一般很難以具象的方式實際的描述出來，根據宜蘭縣建築師公會及宜蘭縣政府建管課，及與各縣市相關單位的訪談中，可以歸納出，本系統的非功能性需求如下：

友善的操作介面：介面的友善程度，攸關系統的親和力，許多設計良好的系統，因介面的不友善，最後淪為失敗的命運。本系統的使用者多半非電腦方面的專長，因此介面的設計上必須考量到系統的直接目標、使用者的專業領域知識及操作習慣，以提高介面的友善度。

良好的系統效能：使用者多半希望系統的回應時間越快越好，因此系統的設計，應避免讓使用者有過久的等待時間。

系統的穩定性：不穩定的系統著實令人頭痛，系統的不穩定，常是使用者的夢魘。因此，系統的穩定性亦是本系統在設計時優先考慮的重點。

系統的安全性：系統必須提供相當程度的安全性，以保持系統的安全，及資料的隱密。

系統佈署的簡易性：前面提及本系統的使用者大都非電腦專才，而且亦無專業的資訊部門，因此系統佈署的簡易性是必須的。由於本系統的服務範圍十分廣大，因此如何利用網路及.Net的特性，快速的完成佈署及維護是系統後續運作十分重要的考量。

彈性的系統設計：本研究在建構系統時，為了因應各地區的特殊需求，應特別要求系統的彈性，以符合各地方的需求，並保持資料的穩定性。

而在功能性需求方面，本系統眼前主要的功能在協助建築管理的請照工作。因此，系統的功能性需求十分簡單。但是這只是表面簡單的目標，如果遇上多變的環境，情況亦會變得十分複雜。本系統為了解決這個問題，並達成資訊共享的目的，因此花費了相當多的心力在資料模型的建立上。以使用共同而具有彈性的資料結構，來解決這些問題。本系統中建造執照請照的活動典型，其他的請照活動，皆與此類似，以下分別針對各項目進行說明。

使用者管理功能：提供使用者管理帳號及權限的管理功能，讓系統具有一定的安全管理。

使用者登入功能：提供系統使用者登入使用系統之用，登入的密碼必須予以隱藏，且必須對密碼的長度進行限制，避免使用者使用過於簡單的密碼。

使用者操作及登入記錄：系統必須要能記錄使用者的登入

及操作，並提供記錄檔的檢視功能。

案件新增：案件的新增功能，必須提供一個良好的輸入介面，以人、事、物等各大類的方式輸入資料，保持資料的完整性。

案件編輯：案件的編輯功能，案件的修改，通常離不開人、事、物的變更，是故案件的編輯亦應以資料的原有形態來進行編輯的功能。

案件刪除：起造人在進行申請的動作之前有可能取消了欲申請的案件，所以系統必須提供刪除案件的功能。

案件查詢：使用者可能因需要而必須查詢某案件的資料，而系統不論在進行編輯、刪除、列印、或上傳的功能時，皆必須經過查詢來取得案件資料。在案件的查詢上，系統將提供彈性多關鍵字查詢能力。

申請文件的列印：除了資料的編輯之外，各式申請文件的列印是本系統的重點之一，配合 XSL、及 XSLT 的運作，系統必須提供完整格式的報表產生、列印、及預覽能力。

案件的上傳與下載：由於本系統強調資訊的共享，所以資訊的上傳與下載能力是必須具備的。

上面是簡單的系統規格描述，接著歸納出其基本架構。

## 第五節、系統基本架構

經過以上許多的分析與探討，針對本研究的需求，將本系統的架構分為資料層與應用層兩部分。基本上仍以三層式架構來思考，上述的應用層即包括程序服務層與操作服務層，再加上資料服務層，就是所謂三層式架構(Three tiers)。本研究實以資料服務層的設計為首要任務。

### 一、資料模型

如前一章所述，本研究的主要精神在於建構一個建築物生命週期資訊共享的環境。並企圖利用 XML 的特性，將資料迴歸其本質。並利用物件導向的分析及開發方法(OOAD)[17]，將建築物視為一個完整物件，而其週遭相關的人、事、物，亦為一個個的完整物件。將建築物及相關人、事、物視為完整物件的好處是，以物件的觀點來看，將更符合真實世界直觀的看法，而且也使得資料本身的架構，更具有意義。本研究所提出共享的觀點以圖解如圖 3.6 所示：



圖 3-6、資訊「橫向共享」與「縱向共享」

所謂的橫向共享意謂案件內的資訊共享，例如建造執照申請書與起造人名冊和建築物概要表內的起造人資料的資訊本是共享的，不應再重複建置。而縱向共享意謂建築物生命週期的資訊共享，例如建築物在建造執照申請與使用執照申請中，均必須用到起造人的資料，這些資訊不斷的會在建築物生命週期中持續出現與引用。在過去的研究及現況的觀察均指出 [2,13,18]，在整個建築物的生命週期中，這些相同的資訊不斷的重複建置，而且資料的型態由於系統獨自開發的結果，不但系統平台各異，資料結構與格式無一定的標準，更使得資訊的共享變得不可能。基於這個原因，本研究嘗試著利用物件導向

及 XML 的技術，讓資料回歸物件的本質，並透過 XML Schema 的建置，產生一個共通而具有擴充能力的資料模型，並將傳統在技術層級解決的許多問題，企圖將之降至資料層級予以解決。

經過傳統資料表格的蒐集彙總與整理，資料模型基本上決定以「人」、「物」、「事」三大項作析離分類，然後再依性質拆分出基底元件及基本模組元件。基底元件及基本模組元件是整個 XML 資料模型的核心元件層(如圖 3-7)，圖 3-7 將整個建築物文件資訊之資料模型分成兩大層：核心元件層與應用元件層。核心元件層由基底元件和基本模組元件組成，主要是描述建築物有關的「人」與「物」的「圖」、「文」資訊(本研究暫不考慮「圖」部分)。這一類的資訊格式需求在建築物生命週期中多數無地域上的差異性，且較為穩定，很少會自行變動，因此將之歸類為靜態模組。而事件的產生，除了事件本身資訊的加入以外，連帶的可能引發相關「人」、「物」的變動，其在整個生命週期中也會隨著時空背景的轉變，而有所改變，所以將之歸為動態模組。

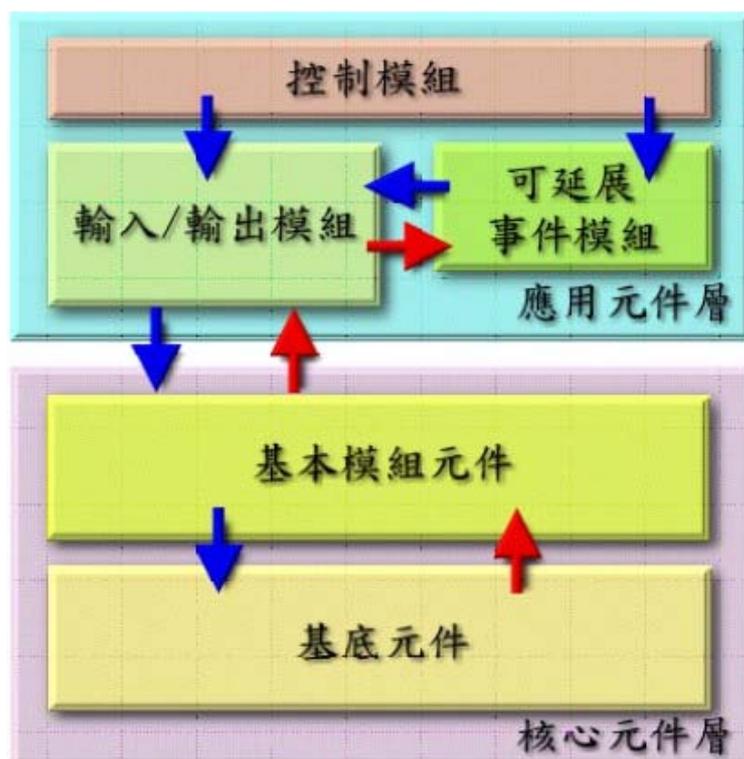


圖 3-7、建築物文件資訊之 XML Schema 架構圖

在資料層的規劃當中，除了對資料的本質加以研究，將其回歸應有的模式外，如何有效的將之組構起來，也是相當重要的一件事。在 XML Schema 中可以如傳統資料庫的一般，表現出資料關連關係，更有甚者亦可以表現出物件關連關係，所以在運作上，應該較傳統資料庫更有彈性。以下是建築物文件資訊之 XML Schema 架構的主要規劃：

- a. 基底元件：例如「住址」、「地號」等基本的資料類別。在這裡將之定義為一個基本的資料型別為其他資料所引用。
- b. 基本模組元件：例如「起造人」、「建築地址」等靜態

模組的資料。將之宣告為一元素型態，並建立各模組元件間的關連，以使資料的運作更加完整順暢。本研究將基本模組元件視為建築物生命週期共享資訊的核心模組元件，甚為重要。

c. 輸出模組：例如「建築執照申請書」、「起造人名冊一」、「起造人名冊二」等，為實體要產生的文件。將透過 XSL 相關的技術來產生，並利用 Crystal Report 工具設計傳統表格供輸出。

d. 輸入模組：近年來的應用程式開發，流行採用 Web-based 介面，透過 HTTP 進行資料編輯與存取操作。本研究經再三討論，認為請照管理系統對操作者而言，大部分時間都在進行各自案件資料編輯與列印，案件資料亦只限在自己客戶端，只有在必要上傳資料或系統更新時才需網路，其所需上網時間與資料操作所需時間相對甚少，因此決定整個系統的運作環境在 Windows Form 下進行，再配合局部上網功能，可喜的是 .NET 的 ADO.NET 的離線式資料庫處理功能正符合所需。透過 .NET 的機制，以其所提供之豐富而多樣的介面元件來設計輸入介面，而 .NET 所提供的非連接導向的程式運作環境，程式無需透過網路來回奔波，有助於效能的提高。

e. 可延展事件模組：例如「建造執照申請」、「使用執照申請」即為各式各樣的事件驅動模組，透過可延展的事件模

組，可以使程式的反應更具彈性，亦增加了系統生命週期的延續力。

## 二、應用層系統架構

在應用層的系統架構中，主要描述系統在實作時的系統架構。根據前面的需求分析得知，本系統主要的服務對象為對建築請照申請者，系統提供一個以元件為基礎的運作環境，為了元件的管理，將統一由建研所的應用程式伺服器進行元件的佈署工作(初期先架在宜蘭縣政府建築管理課)，使用者依其需求下載所需的元件。而各元件間的資料溝通，乃是透過 XML 文件為基礎。預計系統的架構圖如圖 3-8 所示：

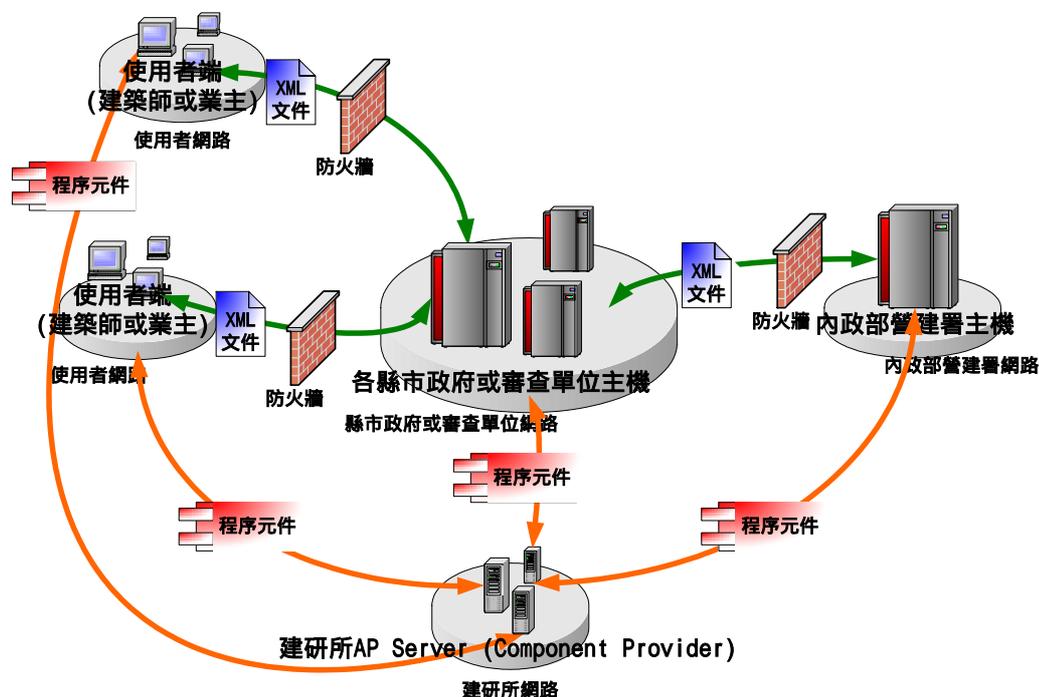


圖 3-8、應用層的系統架構圖

由圖 3-8 可見系統透過建研所的伺服器提供使用端進行下載元件與佈署工作，使用者依其需求，向建研所的伺服器下載相關元件，待相關的元件佈署完備之後，各元件間共同資訊的共享，則可經由共同定義使用相同 XML Schema 的 XML 文件來達成，如此，便可建構一個資訊共享的環境。

## 第四章、建築管理請照系統設計

本研究在 XML Schema 的建構，因碰到許多的瓶頸而花了相當多的心力，至今仍不敢輕言完成，但確信主要的建構原則已趨於成熟。也因此在此系統軟體的設計規劃的進程方面有所壓縮，但整體而言應屬值得，下列即針對請照管理系統的規劃與設計過程加以介紹。

### 第一節、使用者介面規劃

#### 一、設計理念及訴求重點

本系統主要係提供建築師事務所及營造廠使用，因使用對象分散各地，且多數使用者對電腦應用的能力較弱，故系統的親和性及佈署安裝都是設計的重點，下列為本系統設計列為重點考量的項目：

1. 操作的親和性：雖然圖形化操作介面(GUI)及視窗的操作方式廣泛為目前大家所接受且使用方便，但在大量的資料輸入時有時卻略顯麻煩(可能需要滑鼠到處點來點去)，擬考慮系統各項主要功能皆加上快捷鍵功能，系統中的每個視窗均有一定通用操作方式，例如按下 ESC 按鍵可立即關閉視窗等。必須要輸入的項目，擬採取顏色標註的辨識方式，讓使用者可以在視覺上得知其重要

性。

2. 視覺化的操作示意框：使用者在輸入資料或修改資料時常有不知道該欄位必須填寫什麼的情況發生，若有此狀況發生，使用者可將滑鼠停留在該欄位上約不到一秒鐘的時間，系統會出現一個黃色小框，顯現該欄位所需填寫的資料為何，並且附帶填寫範例說明，輔助使用者填寫該欄位資料。
3. 資料異動提醒：若資料有異動或發生變更，使用顯著的色標標註出異動的資料位置。
4. Step By Step 精靈式操作導覽介面：可讓第一次申辦案件的用戶能夠藉由系統上一步一步的詳細說明，一邊輸入申辦案件所需的各項資料，一方面了解其行政作業流程。
5. 資料查核機制：資料填寫的合理性檢查，資料填寫漏失的示警。
6. 資料搜尋：盡可能提供多樣化的參數查詢機制，讓使用者在對欲找尋的資料特徵極少的情況下亦能找到。
7. 使用權限的規劃：提供系統管理者的功能，以利管制系統的使用者。
8. 佈署安裝的方便性：.NET 在軟體元件的佈署安裝方面已進步許多。除了考慮軟體元件輕便化，不會有網路下

載負荷過大現象，亦考慮系統元件與資料架構元件分開包裝，以避免元件更新時的資料覆蓋問題。

## 二、主功能架構考量

系統主功能畫面及各功能表內容如圖 4-1 所示

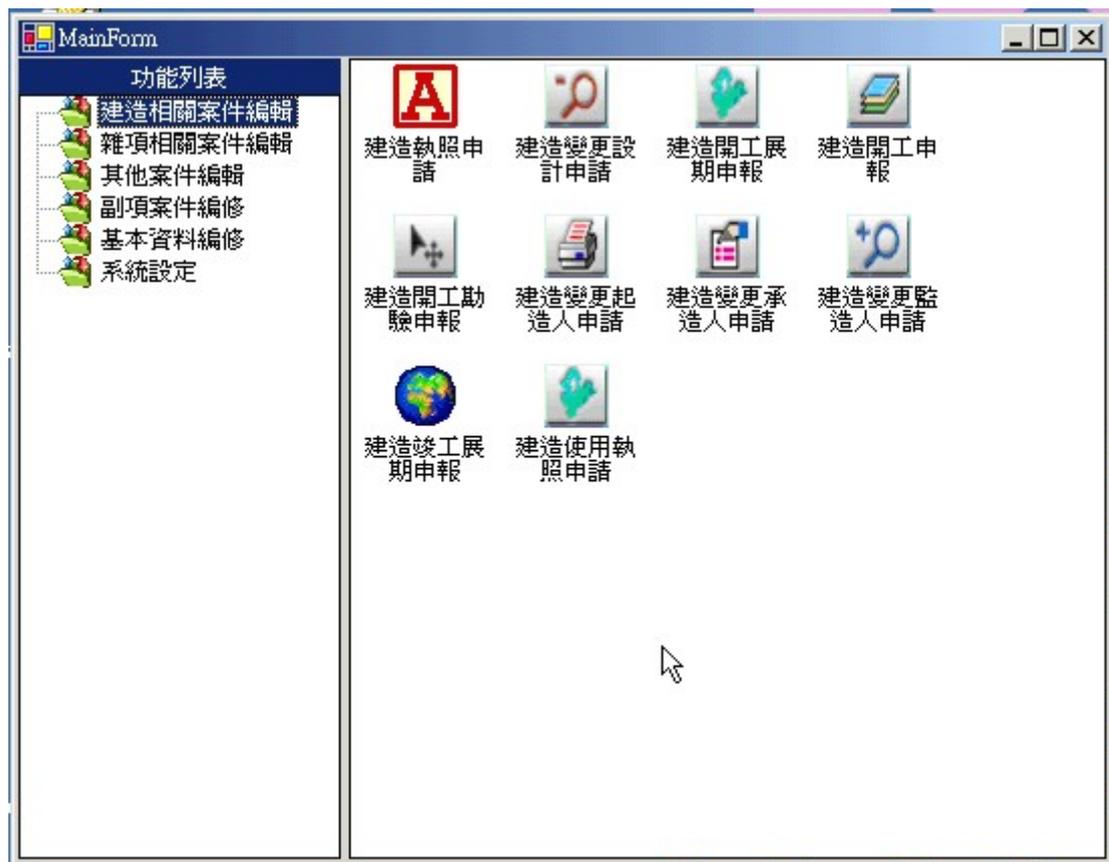


圖 4-1、系統畫面及功能表內容

系統主功能表的規劃考量，著重多數使用者在傳統填表動作的調整與適應，雖然本研究將背後處理的建築物請照資料做了相當大的整頓，但為了讓實務使用上不會有太大的落差，因此，主功能表就直截了當，一進來就出現最常被使用的建造執照申請之編輯，如圖 4-1 所示。各主要功能表的內容如下：

1. 建造相關案件編輯：包括建造執照、變更設計、使用執照、開工、勘驗、展期等編輯功能。
2. 雜項相關案件編輯：包括雜項執照、雜項變更設計、雜項使用執照、開工、勘驗、展期等編輯功能。
3. 其他案件編輯：包括用途變更、拆除執照、室內裝修、執照更正、執照補發等編輯功能。
4. 副項案件編修：開工報告、開工展期、勘驗報告、竣工展期等申請作業相關資料之編修與維護。
5. 基本案件編修：建築師、簽證技師、營造廠等基本資料的維護。
6. 系統設定：縣市設定與標準化資料內容的維護。

## 第二節、系統輸入介面設計考量

在輸入介面的設計上以 Windows Form[21]為主，利用 Windows Form 多樣的互動式元件，構成一個良好的操作環境。並以 VB.Net[22]實作 Data Island 的構想，摒除傳統網路應用程式需要在客戶端與伺服器來回往返的問題，同時提高效能及避免網路頻寬的消耗。.Net 所提供的 Windows Form 可運用之元件如圖 4-2 所示：

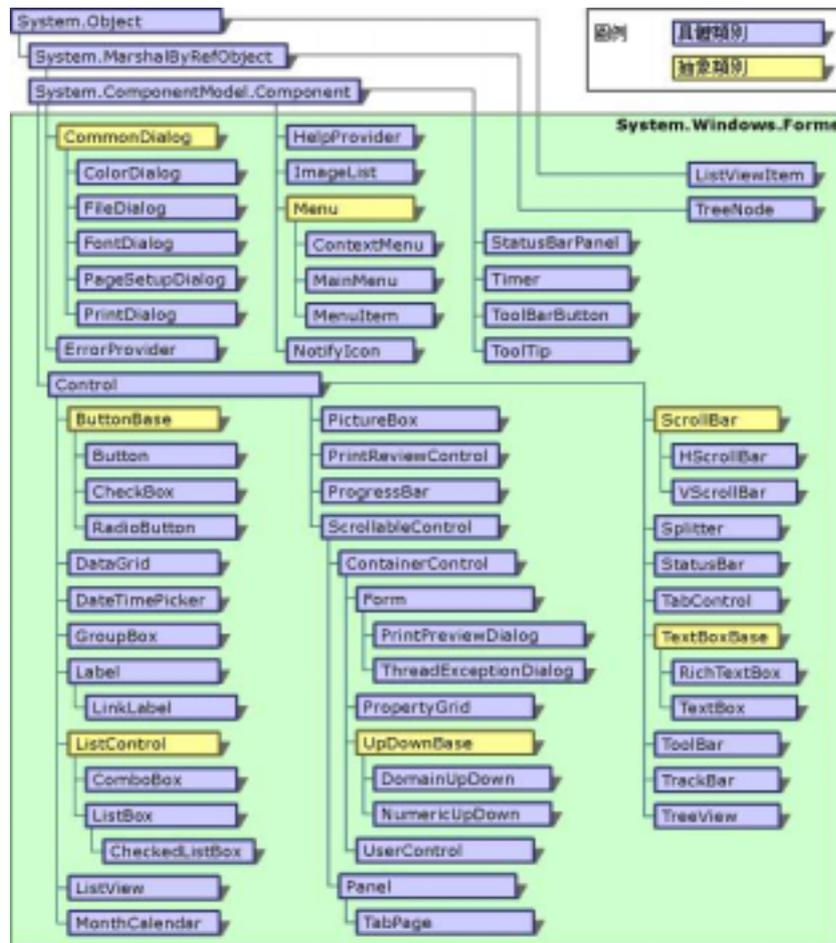


圖 4-2、Windows Form 的元件階層圖

至於在 XML 的存取上除了 MSXML 所實作的 DOM 介面及 SAX 外，.Net 提供了相當多的 XML 支援，.Net 所提供的 XML 命名空間如圖 4-3 所示：



圖 4-3、.Net 所提供的 XML DOM 階層架構

微軟公司之 .Net Framework 是整個 .NET 應用程式運作的核心，其系統架構將是未來微軟的 Windows 作業系統的運作主體，故它對以 XML 為資料交換底層的設計目標為：

1. 與 W3C 標準相容
2. 可擴充性
3. 可外掛架構
4. 效能
5. 與 ADO.Net 緊密整合

根據前述目標及所提供的工具，可用圖 4-4 表示：

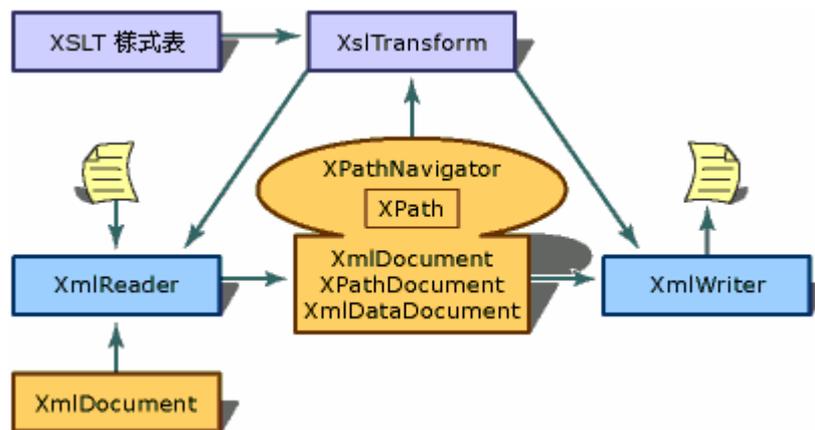


圖 4-4、.Net Framework 中 XML 的設計

而根據本研究的需求，主要應用的有 XmlDocument、XPathDocument、及 XsltTransform，使用 XmlDocument 為 XML 容器的主因，在於這個元件可以與 ADO.Net 的 DataSet 同步處理 XML 文件，它可以利用局部的 XML Schema 對 XML Data 進行框取的動作，並且會維持框

選外的 XML 文件不被改變，同時也可以使用 DataSet 的 DataTable 及 DataRow 物件進行存取。而使用 XPathDocument 為 XML 容器，在於這個元件可以針對 XPath 進行最佳化的處理，可以做為 XsltTransform 的前置處理，以提高 XSLT 的效能。

### 一、自訂控制項的建立

雖然 Windows 提供了許多好用的控制項，但是為了達到本系統所要的目標---資料的標準化及方便的操作性，仍嫌不足。因此，在設計上有許多地方是延續原本的控制項，再增加其功能而成的。所幸，拜 VB.Net 物件導向化之賜，開發自訂控制項變成一個簡單的工作。透過繼承機制，可以很輕鬆的擴充原來控制項的功能，來達成所要的目的。

在本系統中所設計的控制項，介面相關的控制項大都包含於 simple controls 的函式庫裡面，而與資料操作相關的則包含於 data tools 的函式庫裡面。介面相關的控制項主要的目的在於資料的標準化，及增加操作的便利性方面。例如：地址控制項、地號控制項、...等。而與資料操作有關的如資料瀏覽控制項、及編輯控制項等。其中比較具有代表性的控制項，其設計的原委及設計考量，分別介紹如下：

1. 地址控制項：在過去由於地址的輸出，並未經過格式化的限制。所以，在資料的搜尋上產生了許多的問題。例如：村里沒有寫、4-1 號與 4 之 1 號和四之一號...等。因此，在本系統中特別針對這方面做了這樣的一個控制項，由介面下手來達成資料格式標準化的目的，除此之外，也可由介面技術來增加系統的操作性。
2. 土地使用分區及編定用地控制項：在以往的系統中土地使用分區及編定用地也具有和地址相同的問題，而不管是土地使用分區或編定用地，所使用的名詞都是經由正式公告的，因此本系統將之歸納分類，變成一層層的選單，除了可以增加系統的操作性之外，也可以確保使用者輸入名詞的正確性。
3. 資料編輯控制項：傳統的資料編輯控制項，只包含簡單的資料瀏覽及編輯功能。在本系統中重新設計的資料編輯控制項則根據使用者的需求，增加了複製新增及連續新增的功能，並針對使用者的操作邏輯，對於可能的錯誤操作，進行了設計即所謂的防呆處理。複製新增主要針對相似程度較高的資料而設計，系統會自動複製目前這一筆資料並進行新增，如 20 筆地號中只有少部份不同其餘均同的情況下，這個功能就相當需要。而連續新增則是簡化連續的新增動作，當使用

連續新增時，系統在您輸入完本筆資料並按下確認鍵之後，會自動進入新增狀態。如上例所言就可以透過兩者的交互作用(同時使用連續新增及複製新增)，來簡化操作步驟。

### 第三節、系統輸出設計考量

系統輸出的效率與品質問題也是整套系統是否成功的重要一環，本研究預定的輸出係考慮不必讓使用單位再購買額外軟體工具配套的規劃。因此後續的研究需要在 XSLT 與報表產生相關技術上下工夫。

在 XSLT 方面可以發現 .Net Framework 已提供了相當多的支援，而在報表工具上，Visual Studio.Net[23]亦提供了相當方便的工具—Crystal Report[24]，更可貴的是它是免費提供了部份的基本功能。

在資料的存取方面，Crystal Report 提供了擷取和推入兩種模式，如圖 4-5 所示，它並可結合多種資料來源，透過 ADO.Net[25]更可直接存取 XML 資料。

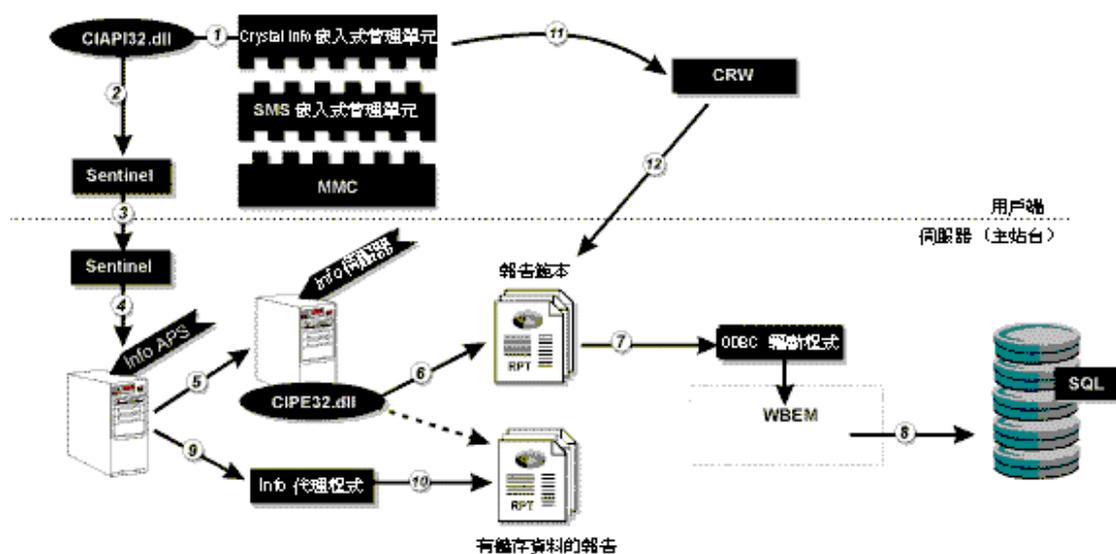


圖 4-5、Crystal Report 的資料存取模式

為了方便報表的設計及資料存取的效能，本研究採用如第 4 章所述的輸出模式，即先依各報表所需的資料建立局部 XML Schema，並以此 Schema 為基礎建立 DataSet 以為報表的資料來源。而資料在進入這個 DataSet 之前，即已透過 XslTransform 進行 XSLT 轉換，將之轉換為較簡單的模式，再進入報表的資料來源(DataSet)中，以提高報表產生的效能。

在報表的佈署方面，若將 managed.msm 及 managed\_cht.msm 合併模組包涵在 MSI 專案中，執行報表所需的所有檔案都會隨著設定檔案正確散發。

使用 Crystal Report 除了簡化了報表產生的問題之外，亦

提供了多種的報表匯出格式，如 PDF、DOC、RTF、XLS…。但是其在匯出成 Word 格式時，尺寸控制上尚有一點小瑕疵。在之前的研究中，曾以 Word 為輸出格式，在使用的彈性上，也獲得了相當的好評，但限於效能的問題有待突破。當然為了部份使用者的需求，本系統亦不排除直接以 Word 格式輸出的可能。

#### 第四節、LFT 的運作機制

LFT(Life Cycle Table)的使用是本系統的主要特色之一，透過這個由一連串二進位成的 64 位元長度的編碼表，將建築物生命週期的資訊記錄其中，而這 64 位元長度的整數值可以完整表示 2 的 64 次方減 1 種的現實狀況，在使用上應是足夠。

其記錄的方式表示如圖 4-6 所示：

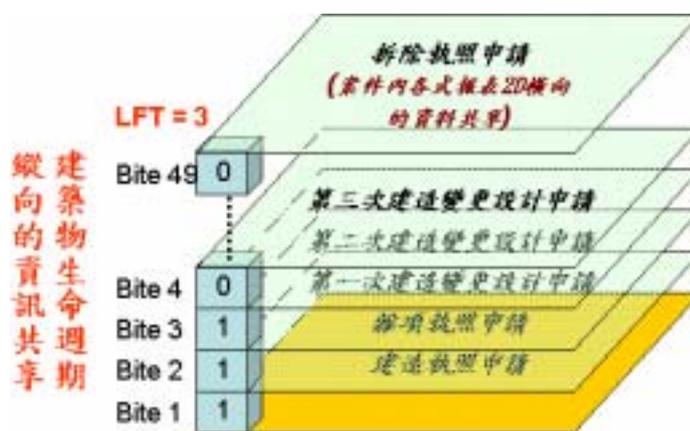


圖 4-6：LFT 的記錄方式概念圖

LFT 的設計概念源自於 IP 的遮罩運算，透過特殊設計的遮罩，經由簡單的位元運算(and)，得以將該位置所記錄的資訊予以取出。取出的方式如圖 4-7 所示：

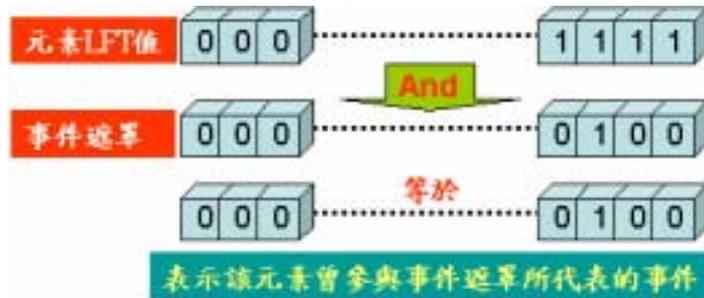


圖 4-7：取出資訊的方法

如果要將事件資訊加入或拿掉，也只需使用簡單的加減法運算即可。使用簡單的運算方法可以有效的提升演算法的運算效率，進而提升系統的運作效能。加入或拿掉事件資訊的方法如圖 4-8 所示：



圖 4-8：加入或拿掉事件資訊的方法

## 第五節、系統靜態結構說明

所謂靜態結構在 OO 中指的就是類別結構，在 UML 的分析方法中有一個重要的圖形在描述物件類別的關係，就是類別圖，類別圖和系統實作間的關係頗大，對於整個系統各類別間的關聯關係，都可以在這個圖形中得到解答，所以在系統完成後的維護及測試階段，類別圖所能提供的功用是相當大的。

在物件導向設計的範疇，可重用性一直是一個重要的課題，一個設計良好的物件模型，是穩定的、可以因應多變的需求環境的。因此，有人將許多 OO 的解決方案將之歸納成為許多典型的樣版，稱之為設計樣版(Design Pattern)[26]。這些設計樣版都是經過特別挑選的，而設計樣版的使用除了可以加快分析的速度外，最重要的是可以讓模型更精煉、穩定、具可重用性。在本系統設計之初，雖然沒有完全按照這樣的規則，但就在模型不斷演進的過程中，也使用了一些 Pattern，如：Abstract Factory Pattern、Template Method、Singleton 等，以提高系統的穩定性，與元件的可重用性。

另外，拜 .Net 之賜，各種視覺元件(控制項)的設計變得相當簡單。為了資料格式的統一，設計了許多的控制項，這些控制項將在 4-2 詳細說明。

以下的說明將系統區分為兩個部份，資料元件及使用者界面元件分節描述之：

### 一、資料元件設計

在本系統中資料元件計有 DataClass、LFT、ToolsClass 等三個程式包裹，這三個程式包裹所包含的類別分別為：DataClass 為對應資料模型及存取資料所需的類別，使用繼承的方式及 Abstract Factory 的設計樣式所組成。LFT 為主要的控制工具包裹，包含了 LFT 所需的各種演算法，及工具，XML 檔案的分割、合併、及篩選的工具，及 BID 的產生器等。而 ToolsClass 則是各種環境變數的設定，在這裡為了確保前後的一至性，主要的類別採用了 Singleton 的設計樣式。

三個程式包裹的類別圖如圖 4-9、4-10、4-11 所示：

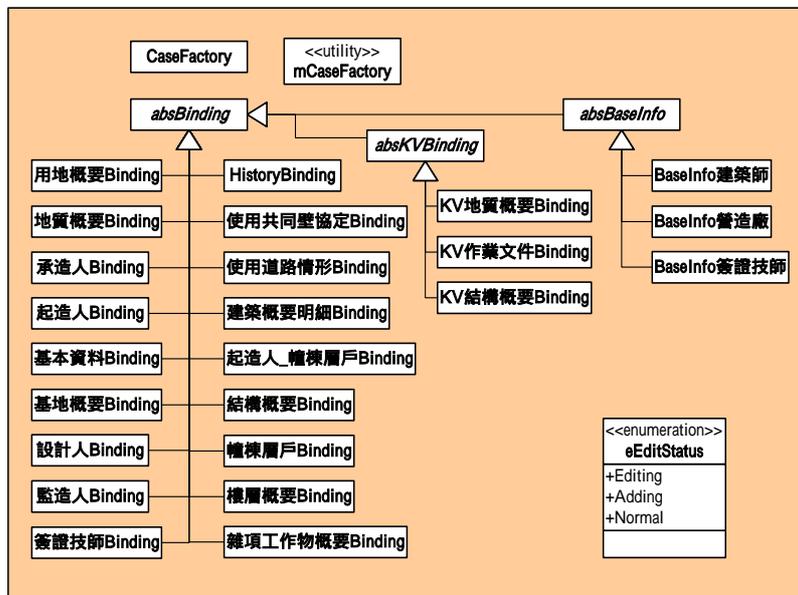


圖 4-9：DataClass 的類別圖

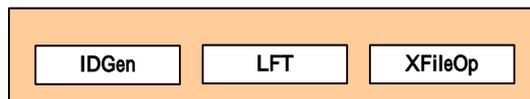


圖 4-10：LFT 的類別圖

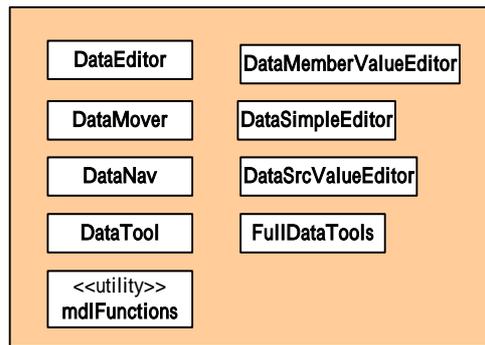


圖 4-11：ToolsClass 的類別圖

Abstract Factory 是以同一個介面來建立一整族相關或相依的物件，而不需點明各物件真正所屬的具象類別。由圖 4-9 中可以看到一整個系列的繼承自 `absBinding` 的各種類別，形成一整族的物件，而這些物件的產生則是交由 `CaseFactory` 這個類別來產生。透過繼承的機制，這些同族的類別有一個共通的介面，而各自相異的實作則交由子類別來實現。透過統一的產生機制，在產生這些類別的物件時，並不需點明各物件真正所屬的具象類別。因此以後如果需要新增同族的類別，只需針對介面加以實作即可，可重用性非常高，而且具有相當的擴充性。例如：在起造人當中要增加一個法定代表人的欄位，只需在相對應的 `FrmPnl` 起造人及起造人 `Binding` 類別即可，要增加新的元件時也是相同的動作即可進行擴充。在圖 4-9 中因為圖形太過複雜所以將之產生的關係移除，所以看不到這個關聯的存在。

另外，要特別說明的是 `absBinding` 這個類別因具有 `LIST` 的特性所以也實作了 `Ilist` 的介面。如此在存取這些 `Binding` 物件時便可以直接使用 `For Each` 的敘述。

Singleton 是一種確保類別只會有一個物件實體存在，並提供單一存取窗口的樣版。透過靜態方法的宣告，並將類別的建構子設為 Private，使得該類別只能透過靜態方法產生實例，而不能由外部直接產生物件實例，這樣的設計可以確保類別只會有一個物件實體存在，這在處理物件內容的唯一性時是很重要的。在環境變數的處理上，理當是一個單一窗口的服務形式，所以使用 Singleton 是一種必要的設計考量。

## 二、使用者界面元件設計

為了方便使用者的輸入，使用了許多的視覺元件及特別的設計方法。例如，TreeNode、NodeList、停駐式 Panel、動態載入的 Panel、自訂控制項等，無不為了使用者的方便著想。在使用者界面方面包含有：DataTools、PanelForm、PopForm、SimpleControls、建築管理請照網路作業系統、執照申請 2 等程式包裹。而 DataTools 及 SimpleControls 兩個程式包裹在 4-2-2 裡有詳細的說明。PanelForm 為各個資料單元的編輯畫面，在程式執行階段程式會動態的載入所需的 Form，以減輕系統資源的消耗。而 PopForm 則是各資料單元的下層節點的彈出式視窗，這兩個程式包裹都實作了樣版方法(Template Method)。而建築管理請照網路作業系統、執照申請 2，主要是程式的操作畫面。各個程式包裹的類別圖如圖 4-12、4-13、4-14、4-15、4-16 等。

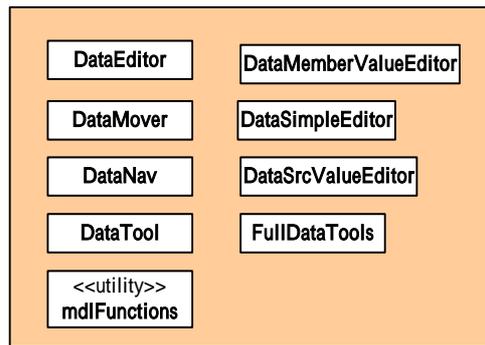


圖 4-12：DataTools 的類別圖

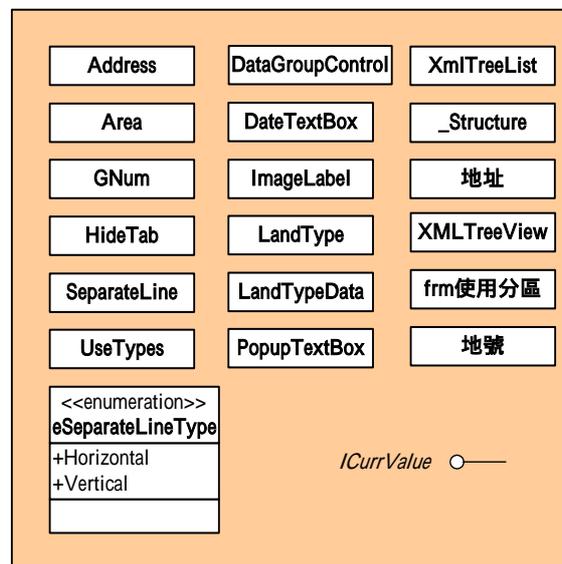


圖 4-13：Simple Controls 的類別圖

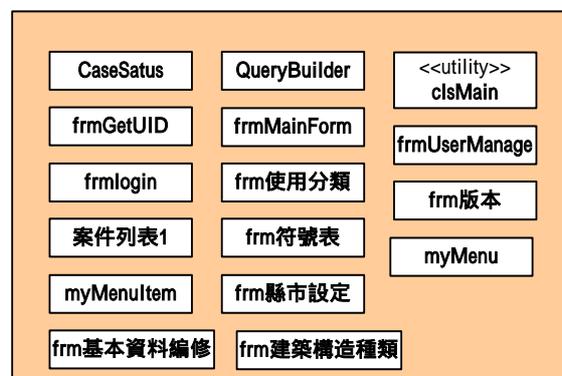


圖 4-14：建築管理請照網路作業系統的類別圖



圖 4-15：執照申請 2 的類別圖

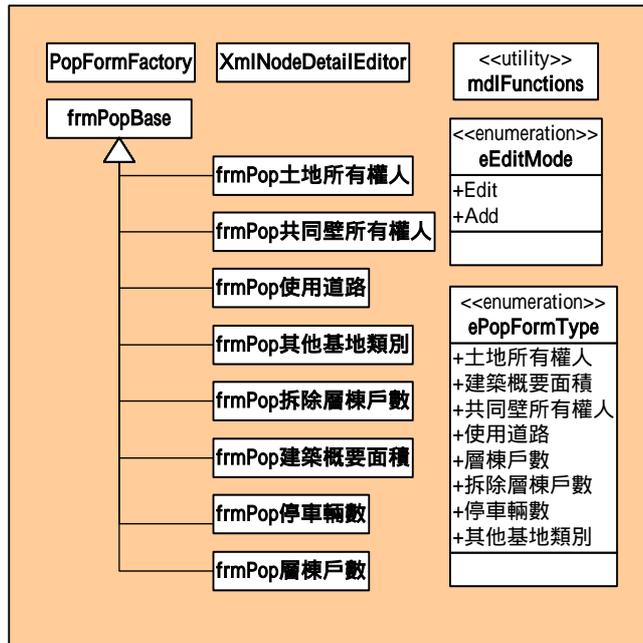


圖 4-16：PopForm 的類別圖

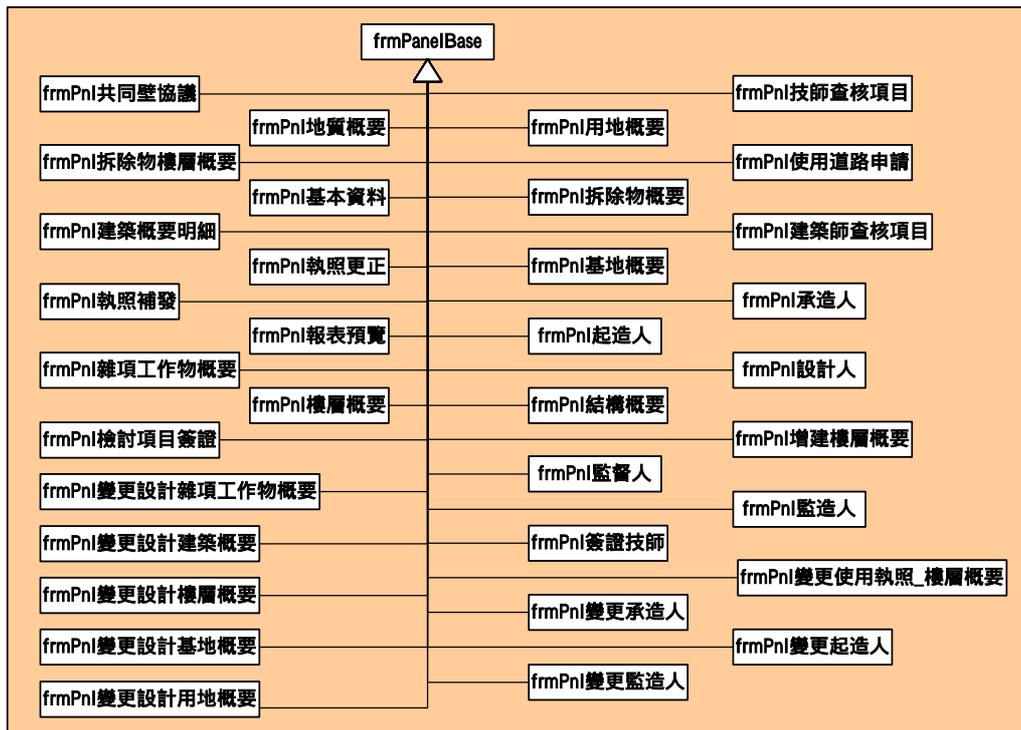


圖 4-17：PanelForm 的類別圖

樣版方法主要是對於操作，只先定義好演算法的輪廓，某些步驟則是交由子類別去填補，類似繼承的目的。以 PanelForm 為例 frmPanelBase 中只定義基本的操作，其餘的實作則交由子類別。

## 第六節、程式封裝及部署的設計

在網路作業系統中，軟體元件的封裝與部署之間有很大的關聯性。因為軟體元件在實際運作的時候，是在各個不同的地方。因此，如何能保持軟體的功能，又能使程式封包減到最小，是一項在設計上必須考量的事情。圖 4-18 即是本系統在實際

部署的元件分佈情形。

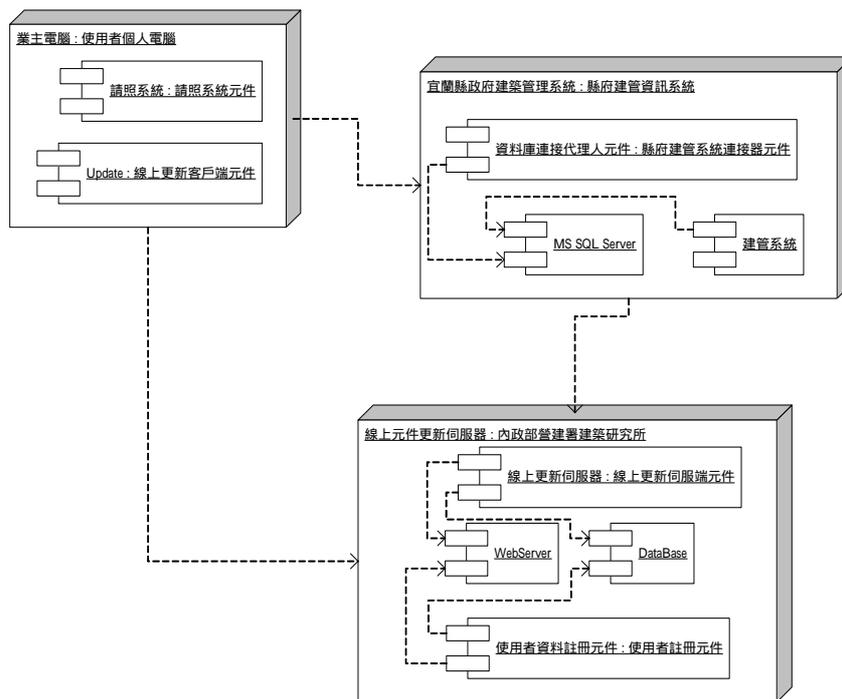


圖 4-18：元件部署圖

由圖 4-18 可以了解本系統在建置及部署上大概可分為三個方面，第一個是使用者端，主要為請照系統元件及線上更新元件。第二個是各縣市政府，資料庫連接代理人元件，此元件主要負責接收使用者上傳案件，並將之轉換之後再進入縣府現有資料庫。第三部份為內政部建築研究所部份，此部份主要為使用者註冊元件及線上更新系統元件。

## 第七節、軟體安裝及啟用流程

由於本軟體的設計是值基於 .Net Framework 之上，所以在安裝前，系統必須先安裝 .Net Framework 1.1 版的套裝。這是一個免費散發的函式庫，可以從微軟的網站下載安裝。為了方

便使用者，本軟體的開發網站及下載點亦提供下載連結。

拜.Net 的技術之賜，本軟體在封裝及安裝的過程，較前版簡化許多，許多擾人的設定問題，在這個版本中，將不復存在。而且整個的系統大小，亦由原來的數十 MB，大幅縮小到 12MB，安裝檔更是小於 10MB。

基於管理的理由，本軟體在下载安裝之後必須透過網路進行啟用的動作，目前由下載至啟用的流程，可以圖 4-19 說明。

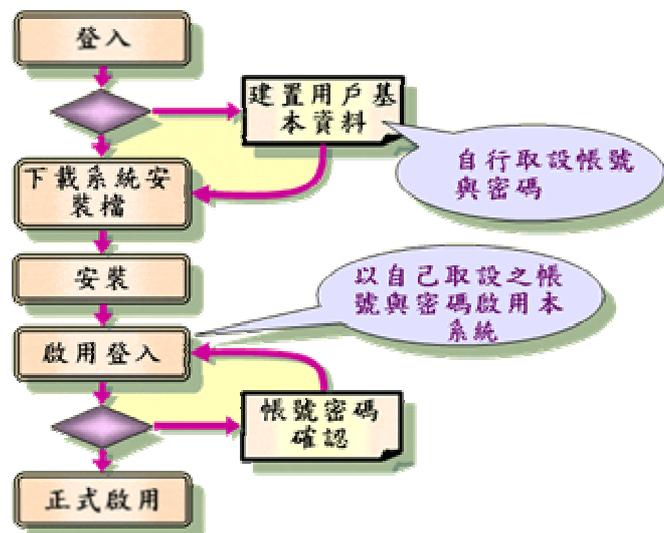


圖 4-19:系統啟用程序示意整個的詳細安裝流程(詳見附錄)

## 第八節、縣府端資料接收及轉換機制

異質資料的轉換問題，亦是異質系統資訊交換不易處理的問題之一。本系統透過使用共同的 Schema 來達成資料格式的完整性及一致性，再利用 XSLT 及 XPath 的技術，來達成異質資料庫之間的傳輸問題。在本計畫中特別以宜蘭縣政府為例，展示了這樣的特性。原來在宜蘭縣政府已經行之有年的請照系

統，其資料的傳遞是經由傳遞系統所建立的 Access 資料庫為主。而縣府的掛號系統即是將之讀入再轉至縣府對應的 SQL 及 Access 資料庫之中。因此，本系統也於縣府端建立了對應的轉換程式。本程式的設計在設計上也保留了相當的彈性，其設計概念如圖 4-20 所示：

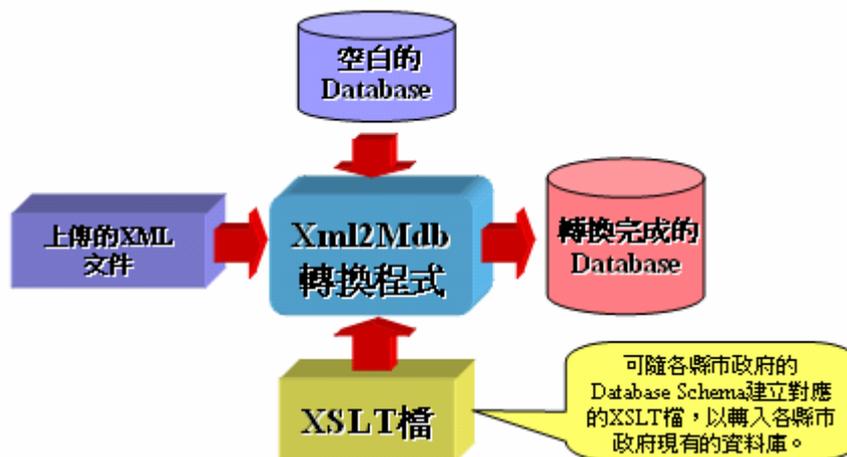


圖 4-20：資料轉換程式的設計概念

該程式可以透過針對不同縣府的 DataBase Schema 所建立的 XSLT 檔，以進行資料的轉換工作。對於 XSLT 技術的實作，更證明了本計畫最初希望能在不更動各縣府的原系統，並且可與未來可能的系統變動的情況下，使用同一個請照系統的可能性。

## 第九節、系統研發資訊及使用回饋機制的建立

本系統專案的研發，因目標在全國建管資訊的標準化，而建管行政作業長期存在有地域的部分差異性，因此，初期在推廣本系統時，將系統研發之理念及系統架構之技術與原理公諸大眾，並尋求使用者的廣泛回響實有其必要。在本系統開發到達到初步可用的程度後，除了透過專家座談，直接與使用者和專家進行分享與討論外，亦透過系統開發網站的建立，直接讓使用者了解系統開發的現況、問題的解決，並尋求回饋以利修正系統功能，使臻於完善。



圖 4-21:系統開發網站

圖 4-21 為本系統開發網站的首頁內容。主要的功能有：

1. 系統發展的理念：讓使用者了解系統的發展背景。

2. 系統發展狀況：讓使用者了解系統目前的發展概況，及進度。
3. 相關技術網站：對於本系統所使用技術的相關網站，讓使用者更了解技術的背景。
4. .Net Framework 下載：由於本系統在安裝之前都必須安裝這個套件，為了避免使用者找尋這個套件，直接提供連結，讓使用者方便下載。
5. 說明文件下載：這是系統操作說明文件的下載位置，目前系統操作文件正陸續完成中。
6. 留言板：提供使用者意見直接回應的地方，也可以提供使用者心得交換。
7. FAQ：提供一些常見問題的解決方法。
8. 寫信給我：提供使用者意見直接以 Email 回應的地方。

## 第五章、系統展示

本研究在 XML Schema 的建構，因碰到許多的瓶頸而花了相當多的心力，致使操作介面一改再改，目前整個系統的建構可謂完成初稿。除了系統主體提供請照操作的介面為份量最重以外，申請者將檔案資料上傳到縣府，以及縣府將此 XML 格式的資料再轉換到原建築管理資訊系統中的 RDB 資料庫，都包含在今年的研發範圍內，下列即針對請照管理系統的相關功能加以介紹。

### 第一節、操作介面

#### 一、基本構想

由於本系統以提供請照案件資料編輯為重點，編輯資料的使用時間遠高於資料在網路上傳及系統元件下載所花的時間，加上 .NET 新技術中，ADO.NET 對資料庫的處理已大幅改革，方便離線操作，因此，本系統在資料編輯的考量，採用離線操作，這大大改善操作畫面切換的效率。而上傳與下載再即時連線作業。

輸出報表方面採用 Crystal Report，亦使系統在處理申請表報上有更高的效率。

## 二、功能目錄

系統畫面及各功能表內容如圖 5-1 所示



圖 5-1、系統畫面及功能表內容

預計系統的初步規劃，各功能表的內容如下：

1. 建造相關案件編輯。

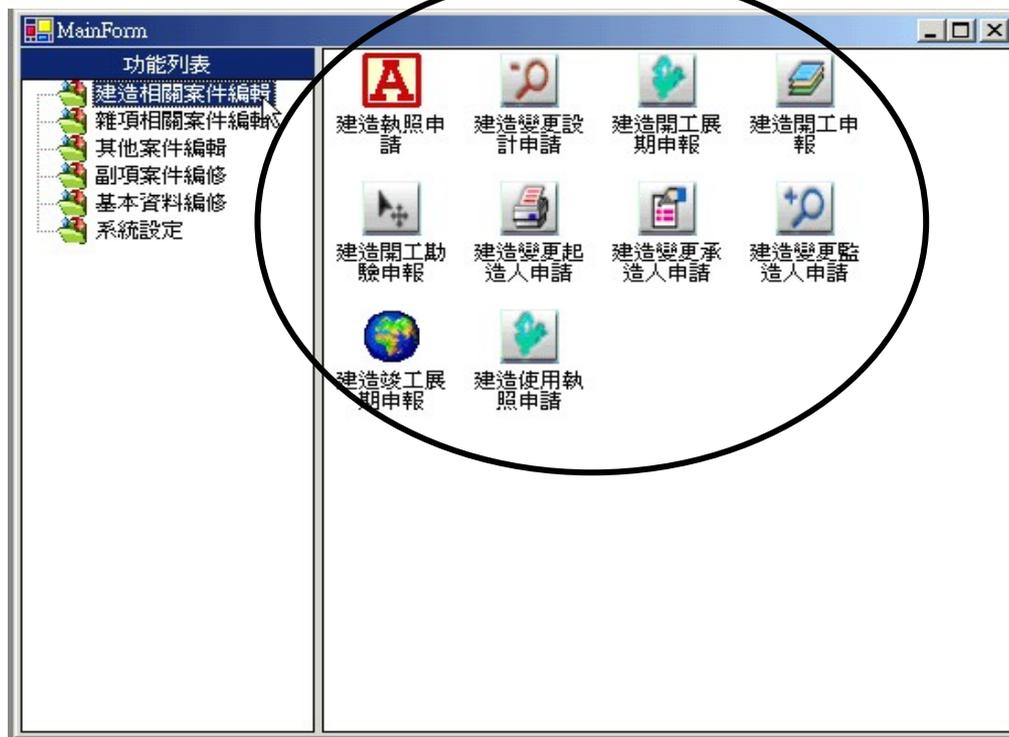


圖 5-2、建造相關案件編輯

2. 雜項相關案件編輯。



圖 5-3、雜項相關案件編輯

3. 其他案件編輯。

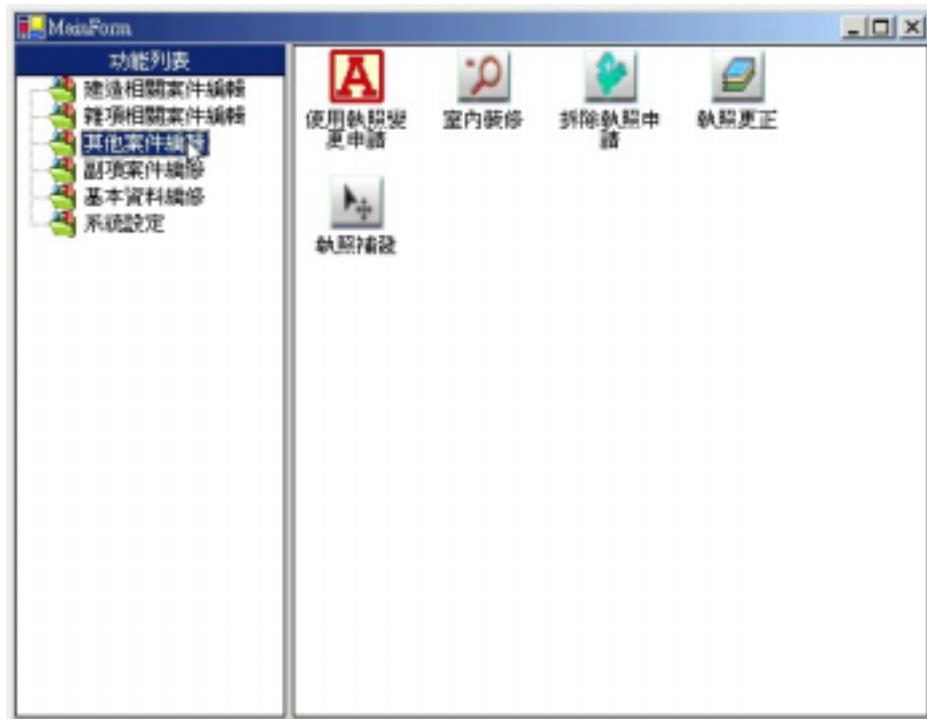


圖 5-4、其他案件編輯

#### 4. 副項案件編修



圖 5-5、副項案件編修

## 5. 基本資料編修

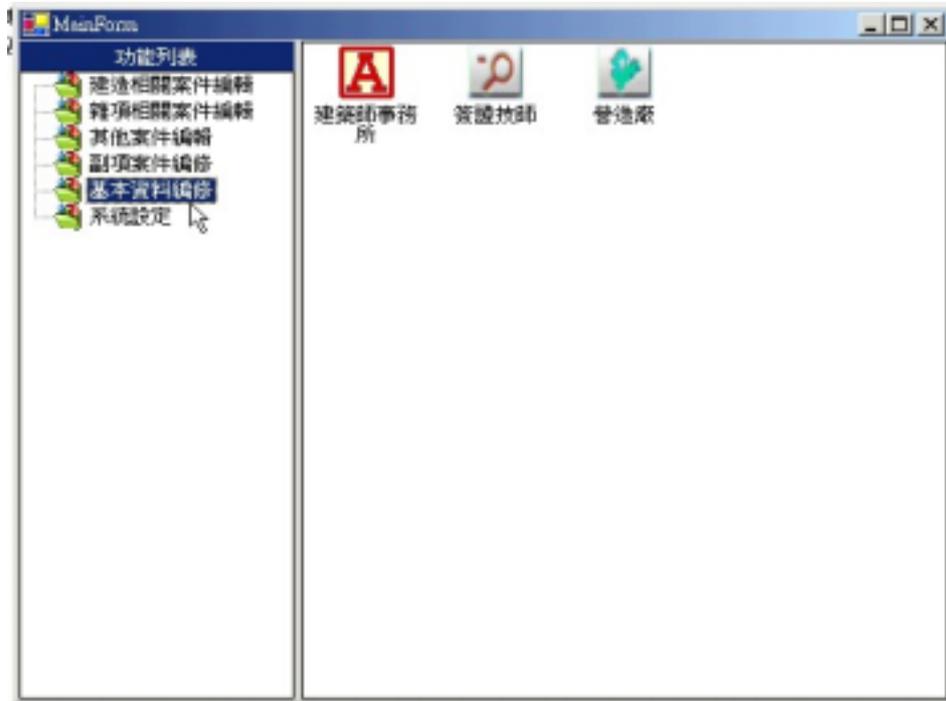


圖 5-6、基本資料編修

## 6. 系統設定

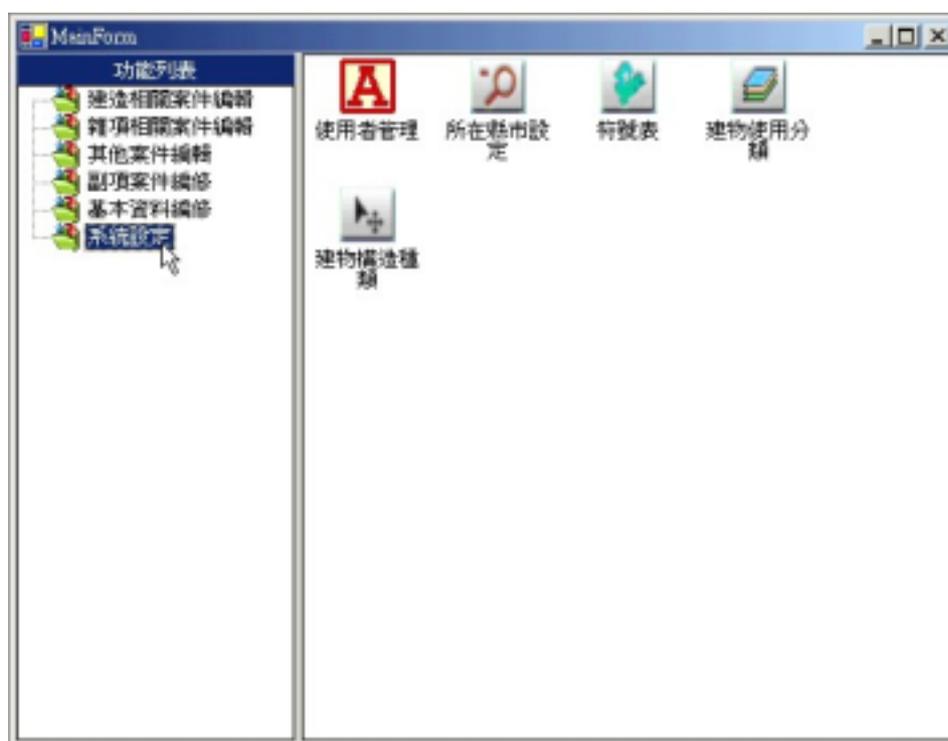


圖 5-7、系統設定

## 第二節、系統輸入介面展示

輸入介面是本系統使用程式最多的部分，包括 Windows Form、Control Panel，以及資料的繫結等。由於考慮一般事務所處理的建築物量在一千件左右，以目前 XML 格式檔案尚無類似 RDB 的資料庫引擎，搜尋時效率不佳，因此，採取一開始就切割出單一建築物的資料來編輯，最後編輯完再寫回去，避免編輯當中不斷做存取動作會有時間延遲的感覺。圖 5-8 為選擇「建造執照申請」圖示選項後會出現案件操作狀態的選項。

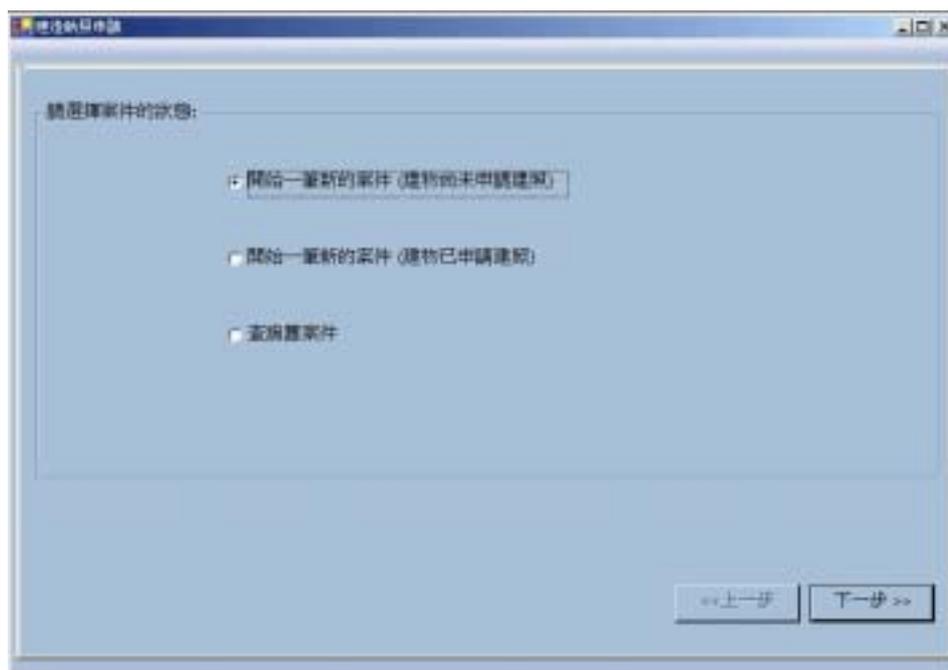


圖 5-8、建造執照申請入口畫面(一)

選擇好舊有建築物後會出現圖 5-9，該建築物生命週期中已申請過的案件種類都會出現。



圖 5-9、建造執照申請入口畫面(二)

圖 5-10 為輸入畫面中有關起造人的基本資料編輯畫面，由於起造人跟建築物的幢棟層戶有不可分的關係，而且是「多對多」的對應關係，因此在起造人的操作介面上用了甚多的功夫將兩者之間的操作介面表現出來。這個對應關係將會反應在起造人名冊一與起造人名冊二的輸出表報上。

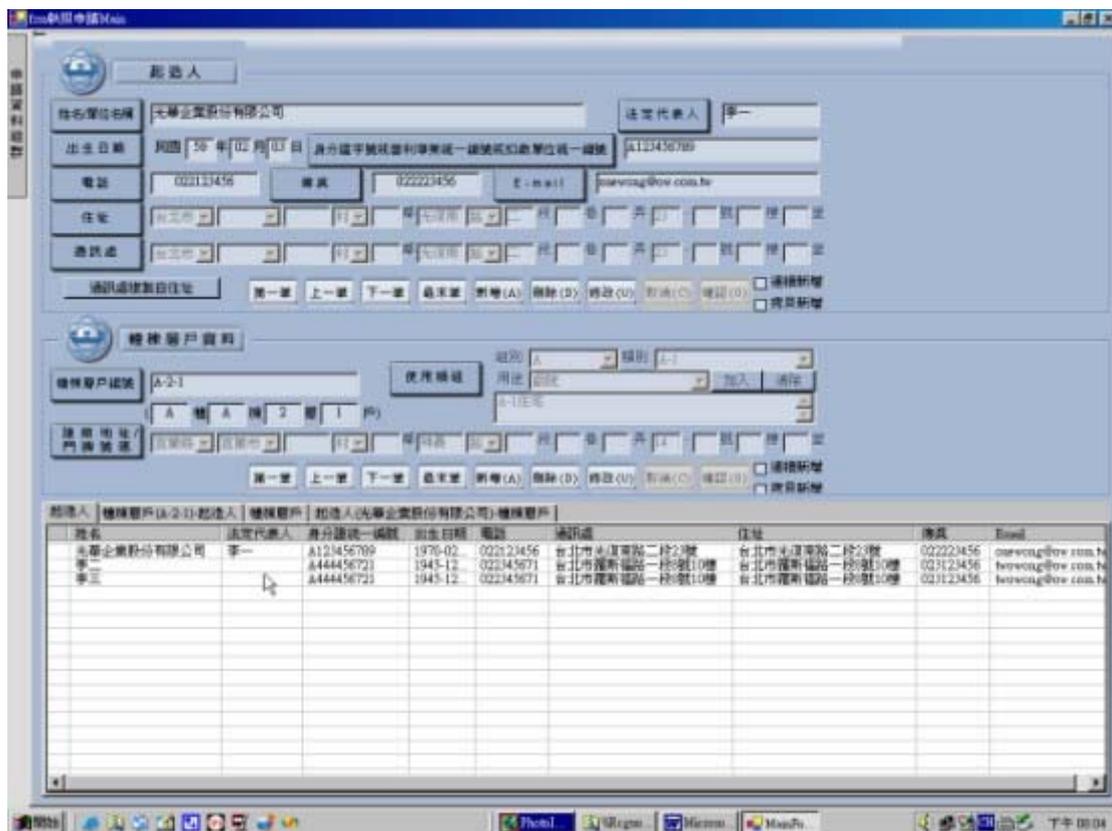


圖 5-10、起造人編輯畫面

### 第三節、系統輸出介面展示

系統輸出的效率與品質問題也是整套系統是否成功的重要一環，本研究預定的輸出係考慮不必讓使用單位再購買額外軟體工具配套的規劃。因此後續的研究需要在 XSLT 與 Crystal Report 之報表產生相關技術上下工夫。

在 XSLT 方面可以發現 .Net Framework 已提供了相當多的支援，而在報表工具上，Studio.Net 亦提供了相當方便的工具 - Crystal Report，更可貴的是它是免費提供了部份的基本功能。

在資料的存取方面，Crystal Report 提供了擷取和推入兩種模式，它並可結合多種資料來源，透過 ADO.Net 更可直接存取 XML 資料。

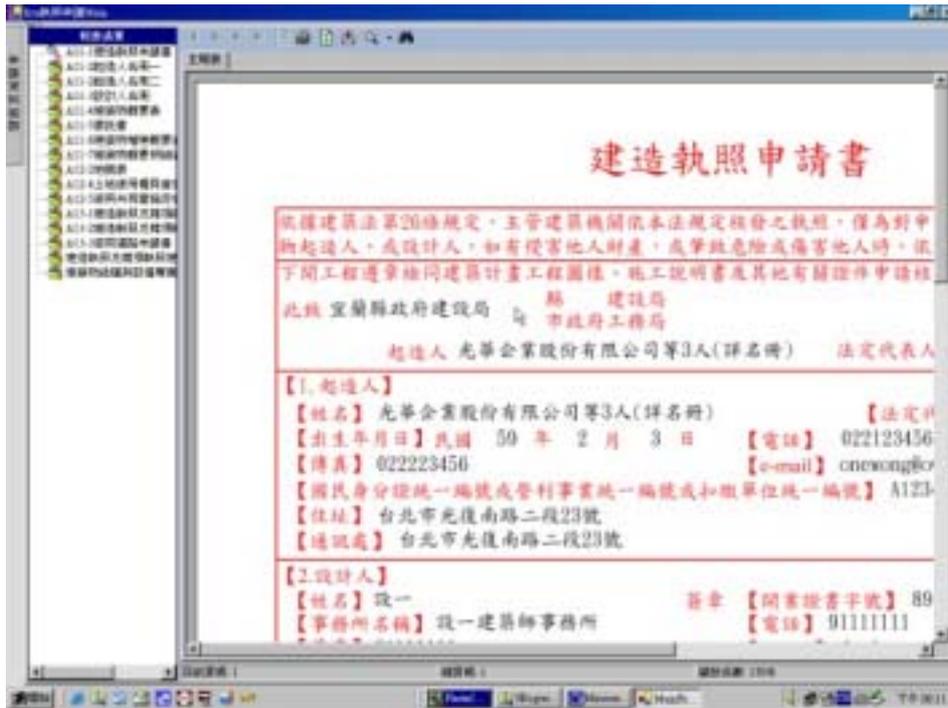


圖 5-11、列印「建造執照申請書」畫面



圖 5-12、列印「起造人名冊(一)」畫面

**建築物概要明細表** A11-7 (建照)

建築類型	新建	
建築字號	2501-01-01建管建字第01號	
建築地點	地址	宜蘭縣神農街神農小莊3號第20層(詳建照)
建造人姓名	光華企業股份有限公司 法定代理人：曹一 曹2,4(詳建照)	
建造人地址	台北市光復南路二段23號	
設計人姓名	曹一	曹四 第一建築師事務所
建築字號	2502-02-03九一建管字第1號	
施工類別	增設既有建物內隔工、隔工日起 2501建管建字第01號	
土地座落位置	一樓商業區第四層	樓層戶數 2 樓上 樓地上 第層-地層 1層-地 30層 12戶
建築物用途	公有辦公用途	構造種類 鋼筋混凝土加強磚 上林估價 1,254,507,000 元
基地面積 (㎡)	樓面面積 (㎡)	容積率 (%)
特種用途	其他	法定
2222.88	12284.12	11.80
建築面積 (㎡)	12284.12	容積率 (%)
3樓特種用途面積 (㎡)	12284.12	法定容積率 (%)
容積率 (%)	274	法定容積率 (%)
容積率 (%)	343	容積率 (%)
容積率 (%)	343	容積率 (%)
容積率 (%)	343	容積率 (%)
容積率 (%)	343	容積率 (%)
容積率 (%)	343	容積率 (%)

圖 5-13、列印「建築物概要明細表」畫面

## 第四節、網路作業

.NET 技術在網路作業的系統運作有突破性的作法，它的 .NET Framework 是一個建置、佈署及執行 XML Web Services 與應用程式的多語言環境。其中 Common Language Runtime 對本研究在網路作業平台的規劃設計非常重要。Common Language Runtime 是元件的執行時期和開發過程

中的重要角色。元件執行時，Runtime 負責管理記憶體配置、啟動及停止執行緒和處理序 (Process)，並且執行安全原則，同時還要滿足元件對其他元件的相依性。開發過程中，Runtime 的角色稍有改變；由於執行許多自動化 (例如記憶體管理)，因此讓開發人員覺得較簡單，尤其是和 COM 比較之下更為明顯。尤其是反映等功能，大幅減少開發人員將商務邏輯轉換為重複使用元件時所需撰寫的程式碼數目。

本系統將安裝於建築研究所的主機，提供全國建築師、營造廠、民眾及政府建管單位的下載，將來會有元件的修正與更新，亦將長期提供下載服務。而使用者在利用此系統完成建築物請照案件編輯並列印傳統表報外，尚可供上傳該案件資料給政府部門。而在政府部門也需安裝一 XMLtoMDB 的轉換程式，供轉換成「建築管理資訊系統」能接受的格式，以併入原資料庫續作自動化審照處理。

## 第六章 結論

「建築管理請照網路作業系統」在多位參與人員一年多來，無數的爭議、閱讀技術文獻、跟建築師與建管人員不斷的討論、加上許多許多次幾乎翻案的修改，終於已大致研發完成，雖然研發小組有稍釋重負的感覺，但相信在全國大量的人使用時，真正的挑戰才要開始。整個研發過程可謂千辛萬苦，尤其系統中蘊含許多新的理念與嘗試，可謂摸黑前進以致撞得鼻青臉腫；心中的困頓與挫折，擔心無法達成的心路歷程，實不足為外人道。

新理念與新資訊技術都是為了建構建築管理資訊更宏觀的價值，目的除了證明這些觀念可行，研發成果還要真正被使用。因此，後續的推廣及維護此系統，使其成長茁壯，還需更多的努力。

本系統雖經千辛萬苦才研發成功，但仍深深瞭解必有諸多缺失及沒有考慮到的細節，故待改善的空間仍大。

如同內政部營建署設計提供全國的請照新表一樣，本研究將去年完成的資料模型架構，企圖引為全國建築管理行政中建築物資訊的標準，先從建築師請照介面的實作開始，接著並據以規劃設計各縣市的建築管理資訊系統，以及所有舊有的建照存根建檔之標準資料結構。如此一來，我國的建築物資訊就有了一致性的起始標準，對於後續的任何有關建築物資訊的引用及交換就容易暢通了。

本系統研發作業，實際上在期中報告前已大致告一段

落，其中雖然還有部份問題，並於專家座談與試用說明會中，得到許多寶貴的建議，據以修正改進，為了更接近使用者，研究小組亦建立了系統研發網站，企圖讓系統的演進公諸大眾，期吸取更多改進意見，使系統更趨實用價值。

為因應使用者時空的差異及變動，本系統規劃之初，即考量系統功能模組延展性，及動態模組的可更換特質。針對各縣市現有的請照需求，進行客製化的動作，而且是在需要的時候才進行下載相關設定檔的動作。這不但大幅的縮小了使用者端的程式大小，而且可以動態的因應各縣市的需求變動(當請照需求改變時，只需刪除原來的設定檔後，重新下載即可)。目前已完成宜蘭縣及台北市的輸出模組實作。

## 第一節、使用者的效益

本系統除了可以進行傳統的請照報表列印外，亦可透過網路傳輸的方式進行線上申請。利用離線編輯的特性，除了大幅減輕伺服器端的負擔之外，亦可減少網路間來回往復傳輸資料的效能損失。而且使用者亦可擁有自己的案件原始檔案，而不用處處受限於伺服器端的限制。另外，當網路狀況不良的情況下，亦可利用離線的方式以檔案申請。

除此之外，就操作便利性而言，新系統的設計除了安裝簡易外，大量使用 Windows Form 介面所提供的控制項，也讓人機介面操作上更添許多親和力。以往使用 Web Based 的操作畫面，由於可使用的控制項較少，操作的便利性上較難發揮。所以考慮到使用者操作的便利性及效率，系統研發之初，在廣泛的研讀有關 .NET 新功能之後，毅然決定使用 Windows Form，並使用 .Net 離線資料繫結及物件導向控制

項的擴充能力，不足的部份再自行研發控制項，以增添系統操作的便利性。

在本系統中幾乎將所有 .Net 平台所提供的介面元件都使用上了，此外，更自行研發了 23 種自訂控制項。這些控制項的設計，大都是根據建築管理這個領域量身訂作的，例如：起造人及幢棟層戶編輯、土地使用分區或編定用地、建築構造、…等。另有一部份是針對資料瀏覽及操作便利而設計的，如地號、地址、編輯工具列、瀏覽及編輯工具列…等。還有一部份是為了改進資料顯示所做的控制項，如 XMLTreeView、XMLListView、可變形式的功能表、停駐式工具列…等(所有的自訂控制項清單，詳請參閱附錄)。

除了使用許多增加操作便利及資料格式標準化的控制項之外，本系統也應用動態載入的新技術，透過這個技術的實作，可大幅減少程式的複雜度，同時又可以保持原來的便利性，這也是本系統的特色之一。

即名為網路作業系統，網路的傳輸能力自然也是設計的重點之一。透過物件導向技術將 IE 內嵌至本系統之中，以及 .Net 所提供的網路連結功能，讓本系統能在資料編輯完成後，可以透過網路進行線上申請。當然，這必須同時在縣府端設計對應的轉換機制之後方可進行。將來更可透過 Web Service 的方式，進行網路線上申請。

除了對建築物生命週期的實作，讓使用者可以最少的資料 Key in，以及最簡單明確的資料輸入，和簡潔、可客製化的操作環境，這些都是以往的請照系統所缺乏的。

本系統雖然盡力的使操作簡化，但由於使用的技術及觀念較為新穎，且由於使用新技術所產生的許多未知問題，尚需解決；因此，在說明文件的建立及系統研發資訊的暢通

上，特建立專屬技術網站，供使用者下載系統研發資訊等相關文件，並與系統研發人員直接進行互動。系統研發網站亦已於本年度 10 月份正式上線，內容正積極更新擴充中。

## 第二節、建管部門的效益

就建管單位而言，本系統可在不更動目前縣府建管資訊系統的情況下，完成網路請照的目的。其中，只需在建管部門的主機加裝由 XML 轉成原建管資料庫格式的轉換元件即可。目前宜蘭縣政府的建管資訊系統已經完成轉換的設置，轉換程式為 Xml2Mdb。本系統上傳的資料是一個 XML 檔案，而且所用的 Schema 與原資料庫的 Schema 有很大的差異，而宜蘭縣政府建築管理課使用磁片檔案進行請照作業，使用的檔案格式為 Access 的 MDB 檔。Xml2Mdb 即負責將 XML 檔轉換成可以順利掛號及線上審查的 MDB 檔。依照這樣的經驗，全國各縣市政府既有的建管系統所用的資料庫平台（大多為 Oracle 或 SQL Server、Access 或 Sybase 等）都可依此技術達成轉換工作。

理想的網路請照作業，必須結合使用者端與各縣府端同時俱備連線與接收能力方能完整。但是，由於各縣市對請照的文件需求皆稍有差異，而且部份較為先進的縣市政府，業已建置自己的建築管理系統使用多年，並累積了許多寶貴的資料，這產生了諸多資料交換與彙整的困擾。

而本系統建置的主要效益，應在於使用一個共同而完整的資料模型(XML Schema)，經由共用的 Schema 及 XML 對資料的解釋能力，讓本系統可透過 XSLT 轉換資料的型態，和各種傳統的資料庫進行交換。而使用共同的資料模型，做

為中介的傳輸介面，不僅可以在使用者間通行無阻，對於不同縣市間的資料交換，更可以透過電腦與電腦之間的溝通，而減少許多人力作業的疏失及時間的浪費。

## 參考書目

- [1] 王明德 ( 1997 ) , ” 建築生命週期資訊管理 ” , 營建資訊運籌管理 (Construction CALS)研討會論文集 , 民國八十七年十月 , 台灣大學 , C1-C12。
- [2] 郭榮欽(2000), 建築物生命週期資訊共享之研究, 國立台灣大學土木工程學研究所博士論文。
- [3] W3C(2000), *Extensible Markup Language (XML) 1.0 (Second Edition)* , <http://www.w3.org/TR/REC-xml> , 經濟部標準檢驗局電子商務 / 資料交換標準分組委員會 ( TEC ) : [http://www.twtec.org.tw/XML\\_spec.htm](http://www.twtec.org.tw/XML_spec.htm)。
- [4] W3C(2001), *XML Schema Part 1: Structures*, W3C Recommendation 2 May 2001.
- [5] W3C(2001), *XML Schema Part 2: Datatypes*, W3C Recommendation 02 May 2001.
- [6] W3C(1999), *XML Path Language (XPath) Version 1.0*, W3C Recommendation 16 November 1999.
- [7] W3C(2001), *Extensible Stylesheet Language (XSL) Version 1.0*, W3C Recommendation 15 October 2001.
- [8] W3C(1999), *XSL Transformations (XSLT) Version 1.0*, W3C Recommendation 16 November 1999.
- [9] Kevin Hoffman(2001),*Professional .NET Framework*, Wrox Press Ltd.
- [10] XML 技術中心譯, Steven Holzner 著(2002), *XML 深入剖析:實務應用與探討*, 維科圖書有限公司。
- [11] 周瑞(1999), “OOAD 及開發 N-Tier 系統的實務技巧和思維,” *MISOO 物件導向雜誌*, 第 12 期。
- [12] Microsoft, <http://www.microsoft.com/Taiwan/msdn/Main.htm>.
- [13] 劉慶男, 郭榮欽, 謝尚賢 ( 2001 ) , ” 建築管理請照網路作業系統之研究 ( I ) ” 期末報告, 內政部建築研究所。
- [14] 位元文化 ( 2002 ) , ” 最新標準 XML 技術實務 ” , 文魁資訊股份有限公司。
- [15] Booch, G., Rumbaugh J., and Jacobson, I.(1999), *The Unified Modeling Language User Guide*, Addison-Wesley, Reading, Mass.
- [16] David Carlson(2001), *Modeling XML Applications with UML: Practical e-Business Applications*, Addison-Wesley, Pearson Education.
- [17] 周思畏, 詹雅璟, 張春霞, 簡杏玟 (2002), 物件導向系統分析與設計:使用 UML 與 C++, 全華科技圖書股份有限公司。
- [18] 郭榮欽, 秦壁如, 黃宏偉, 楊承剛 (1999), ” 宜蘭縣政府建管資訊系統使用

手冊”,宜蘭縣政府建設局建築管理課,宜蘭市。

- [19] Brend Oestereich ( 2003 ), 鄧文彥譯, "UML 軟體發展實務 - 物件導向設計方法的練習", 培生教育出版集團。
- [20] Dan Wahlin(2001), *XML for ASP.NET Developers*, Addison-Wesley, Pearson Education.
- [21] Chris Payne( 2002 ), *Sams Teach Yourself .NET Windows Form in 21 Days*, Sams.
- [22] 曹祖聖等 5 人 ( 2002 ), "Visual Basic.NET 程式設計經典", 松崗電腦圖書股份有限公司。
- [23] Microsoft ( 2002 ), "Visual Studio.NET 逐步解說", Microsoft.
- [24] 鄭淑芬 ( 2002 ), ".NET Crystal Report 實戰演練", 學貫行銷股份有限公司。
- [25] 張世敏譯, Rebecca M.Riordan ( 2002 ), "Microsoft ADO.NET 程式設計之鑰", Microsoft Press.
- [26] William Stamatakis( 2001 ), "VB 高手救命十招-善用 Design Pattern", 華彩 ( 昱威資訊有限公司 )。

## 附錄 A：自訂控制項

為了資料輸入的格式化，自訂控制項的使用是一個良好的策略，透過視覺控制項的限制輸入格式，可以使資料的輸出格式化。以下是本系統所設計自訂控制項的詳細說明。

Name Space	說明
DataTools	凡是會操作到 absBinding，且與 UI 有關之物件，歸為此類
SimpleControls	凡是不會直接操作 absBinding，且與 UI 有關之物件，歸為此類

Object Name	SimpleControls.Address
Object Intent	讓 User 輸入地址的物件
Description	
Property	CurrentValue:String(直接指定地址如：基隆市安樂區樂利三街 46 巷 45 號. 會自動拆解至相對位置)
method	Nil
資料檔	Xml/縣市.xml (Env.DataSet 縣市)
畫面	

Object Name	SimpleControls.Area
-------------	---------------------

建築管理請照網路作業系統之研究 (II)

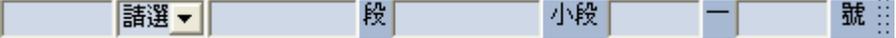
Object Intent	讓 User 輸入土地使用分區或編定用地
<b>Description</b>	
Property	CurrentValue:String
method	
資料檔	Xml/使用分區或編定用地.xml (Env.DataSet 使用份區)
畫面	

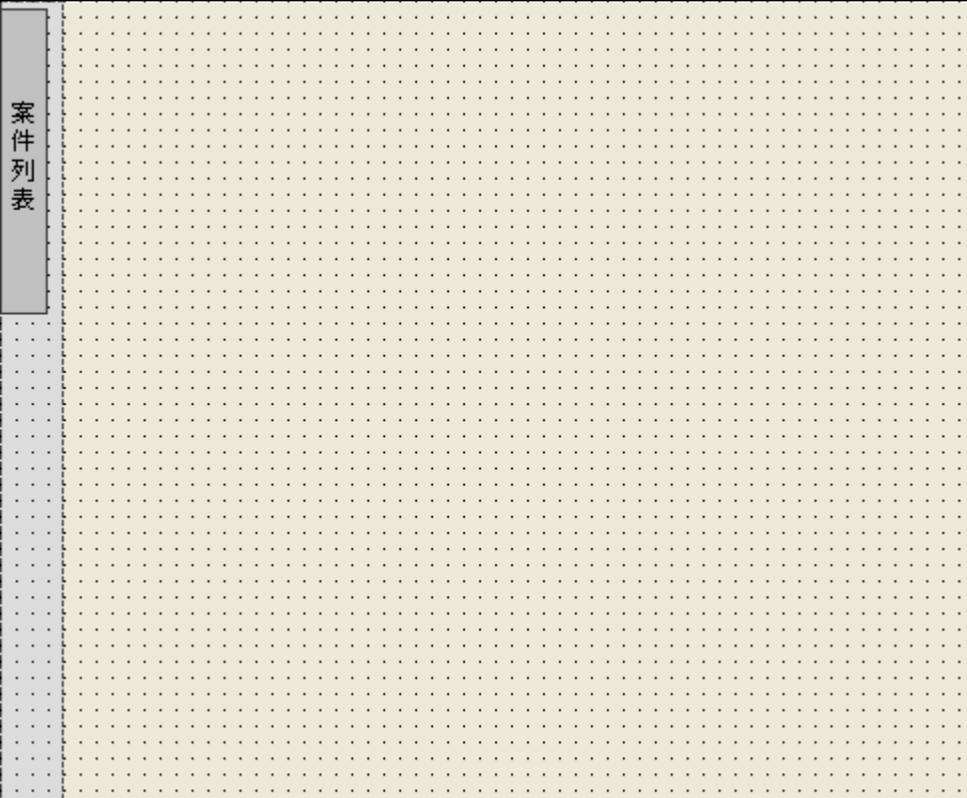
Object Name	SimpleControls.DataGroupControl
Object Intent	讓 User 單選或複選之控制項
<b>Description</b>	
Property	<p>CurrentValue:String(傳回或設定所選之項目之名稱如：“新建” 或 “增建,改建”)</p> <p>Columns:Integer (傳回或設定一行中有幾個控制項，以下就是 Columns=4 的表現)</p> <p>DataList:String (傳回或設定所要讓使用者選擇的項目如：“新建,增建,改建,修建”)</p> <p>MarginXXX(設定上下左右的邊界)</p> <p>ReadOnly(設定唯讀狀態)</p> <p>Style(設定顯示為 CheckBox or RadioButton)</p> <p>Title(設定標題)</p>

Object Name	SimpleControls.DataGroupControl
Object Intent	讓 User 單選或複選之控制項
<b>Description</b>	
method	Nil
畫面	

Object Name	SimpleControls.DateTextBox
Object Intent	民國年的輸入控制項，資料儲存為西元年、顯示為民國年
<b>Description</b>	
Property	CurrentValue:String(傳回或設定日期—西元年 YYYY/MM/DD ) ReadOnly(設定唯讀狀態)
method	Nil
畫面	

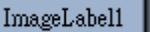
Object Name	SimpleControls.GNum
Object Intent	民國年的輸入控制項，資料儲存為西元年、顯示為民國年
<b>Description</b>	

Object Name	SimpleControls.GNum
Object Intent	民國年的輸入控制項，資料儲存為西元年、顯示為民國年
<b>Description</b>	
Property	CurrentValue:String(傳回或設目前的地號值) ReadOnly(設定唯讀狀態)
method	Nil
畫面	
Object Name	SimpleControls.HidTab
Object Intent	一個類似 .Net IDE 介面中可以收回，展開控制項的控制項
<b>Description</b>	
Property	AllowHiding:Boolean (設定被指定之控制項是否可以收回) HideControl:Control(設定所要隱藏的控制項) HideSteps:integer (設定收回的步數) isHide:Boolean (取得目前是否為隱藏) ShowSteps:Integer(設定展開的步數)
method	Nil

Object Name	SimpleControls.GNum	
Object Intent	民國年的輸入控制項，資料儲存為西元年、顯示為民國年	
<b>Description</b>		
畫面		

Object Name	SimpleControls.ImageLabel1
-------------	----------------------------

建築管理請照網路作業系統之研究 (II)

Object Intent	一個類似 .Net IDE 介面中可以收回，展開控制項的控制項
<b>Description</b>	
Property	Text:String (所要顯示的字串)
method	Nil
畫面	

Object Name	SimpleControls.LandType
Object Intent	土地使用分區
<b>Description</b>	
Property	未定!!
method	Nil
畫面	都市計劃別： <input type="text" value="ComboBox1"/> 使用分區別： <input type="text" value="ComboBox1"/> 分區細類： <input type="text" value="ComboBox1"/>

Object Name	SimpleControls.SeparateLine
Object Intent	就是一條線
<b>Description</b>	
Property	Style:{Vertical, Horizontal} (設定為垂直 or 水平線)
method	Nil
畫面	Nil

Object Name	SimpleControls.Structure
Object Intent	選擇構造種類
<b>Description</b>	
Property	CurrentValue:String
method	Nil
資料檔	Xml/建築構造.xml (Env.DataSet 建築構造)
畫面	

Object Name	SimpleControls.UseTypes
Object Intent	使用類組
<b>Description</b>	
Property	CurrentValue:String
method	Nil
資料檔	Xml/使用類組.xml (Env.DataSet 使用類組)

建築管理請照網路作業系統之研究 (II)

Object Name	SimpleControls.UseTypes	
Object Intent	使用類組	
<b>Description</b>		
畫面		
Object Name	SimpleControls.XmlTreeView	
Object Intent	將 XmlNode 的樹狀結構顯示在 Tree View 中	
<b>Description</b>		
Property	DataNode:Xml.XmlNode Caption:String	
method	Nil	
Event	NodeClick(Xml.XmlNode)	
資料檔	Xml/使用類組.xml (Env.DataSet 使用類組)	
畫面		Xml 格式說明

Object Name	SimpleControls.UseTypes	
Object Intent	使用類組	
Description		
		<ol style="list-style-type: none"> <li>1. 要有 ShowInTree="true" Attribute 之 node 方會顯示，且其父節點也要有同樣的 Attribute</li> <li>2. Icon="IconName.XXX" 可以設定該 node 所要顯示的圖示名稱。圖示皆放在 Env.IconPath 的路徑中</li> </ol>

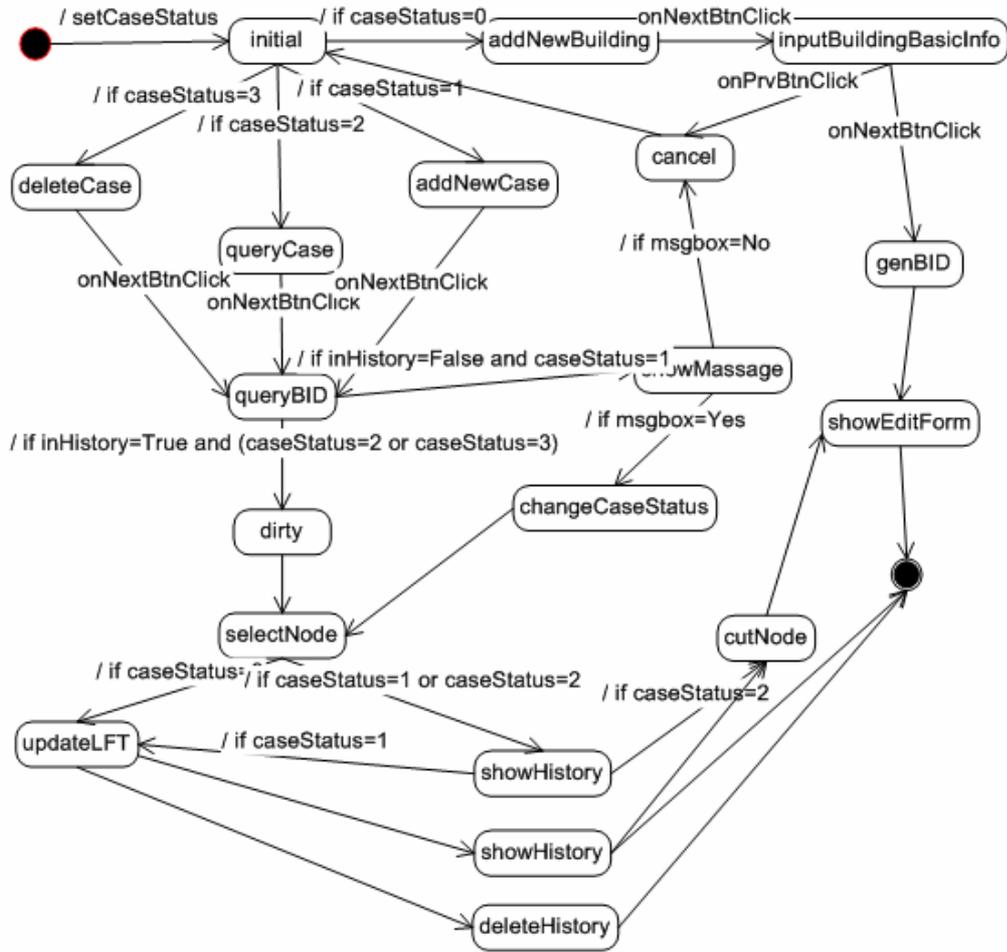
建築管理請照網路作業系統之研究 (II)

Object Name	PopForm.XmlNodeDetailEditor
Object Intent	將 XmlNode 的資料顯示在 ListView 中，並配合 PopForm 可以對該 Node 進行新增、修改的功能。並可對 Node 的資料做刪除。 <a href="#">詳細說明請看類別圖</a>
Description	
Property	DataSource:Xml.XmlNode 所要進行顯示、編輯的 Node Text:String 標題 CurrentNode:Xml.XmlNode 目前所點選的第一個節點 PopUpForm:ePopFormType 彈出 Form 的種類 FieldName:String 指定目前
method	Nil
Event	Nil
資料檔	Xml/使用類組.xml (Env.DataSet 使用類組)

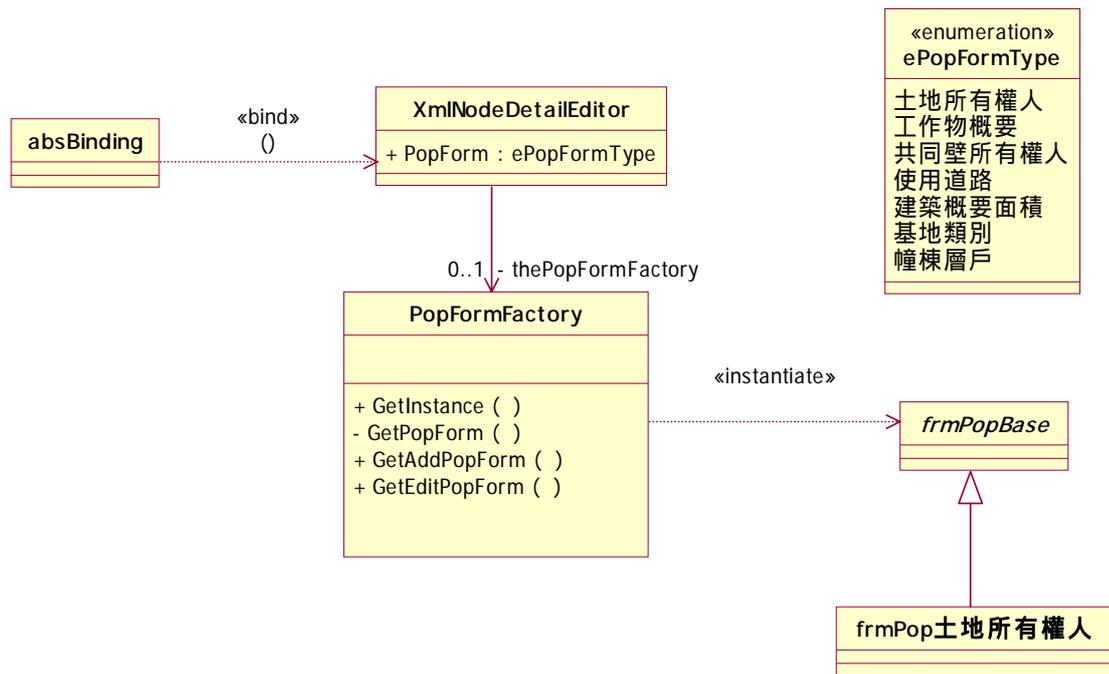
<b>Object Name</b>	<b>PopForm.XmlNodeDetailEditor</b>
<b>Object Intent</b>	將 XmlNode 的資料顯示在 ListView 中，並配合 PopForm 可以對該 Node 進行新增、修改的功能。並可對 Node 的資料做刪除。 <a href="#">詳細說明請看類別圖</a>
<b>Description</b>	
<b>畫面</b>	 <p>The screenshot shows a window titled 'XmlNodeDetailEditor'. It contains a list view with several empty rows. At the bottom right, there are three buttons: '新增 (A)', '刪除 (D)', and '修改 (U)'. The list view area is currently empty, suggesting it is ready to display data.</p>

Object Name	DataControl.FullDataTools	
Object Intent	操作 absBinding 或 BindingManagerBase 物件，提供資料瀏覽、編輯等操作	
Description		
Property	BindingObj	指定所要 Binding 之 absBinding 物件
	Datasource	指定所要 Binding 之 BindingManager 物件
	FirstControl	指定在新增、修改時所要 Focus 的第一個控制項
method	Nil	
Event	Nil	
資料檔	Xml/使用類組.xml (Env.DataSet 使用類組)	
畫面		

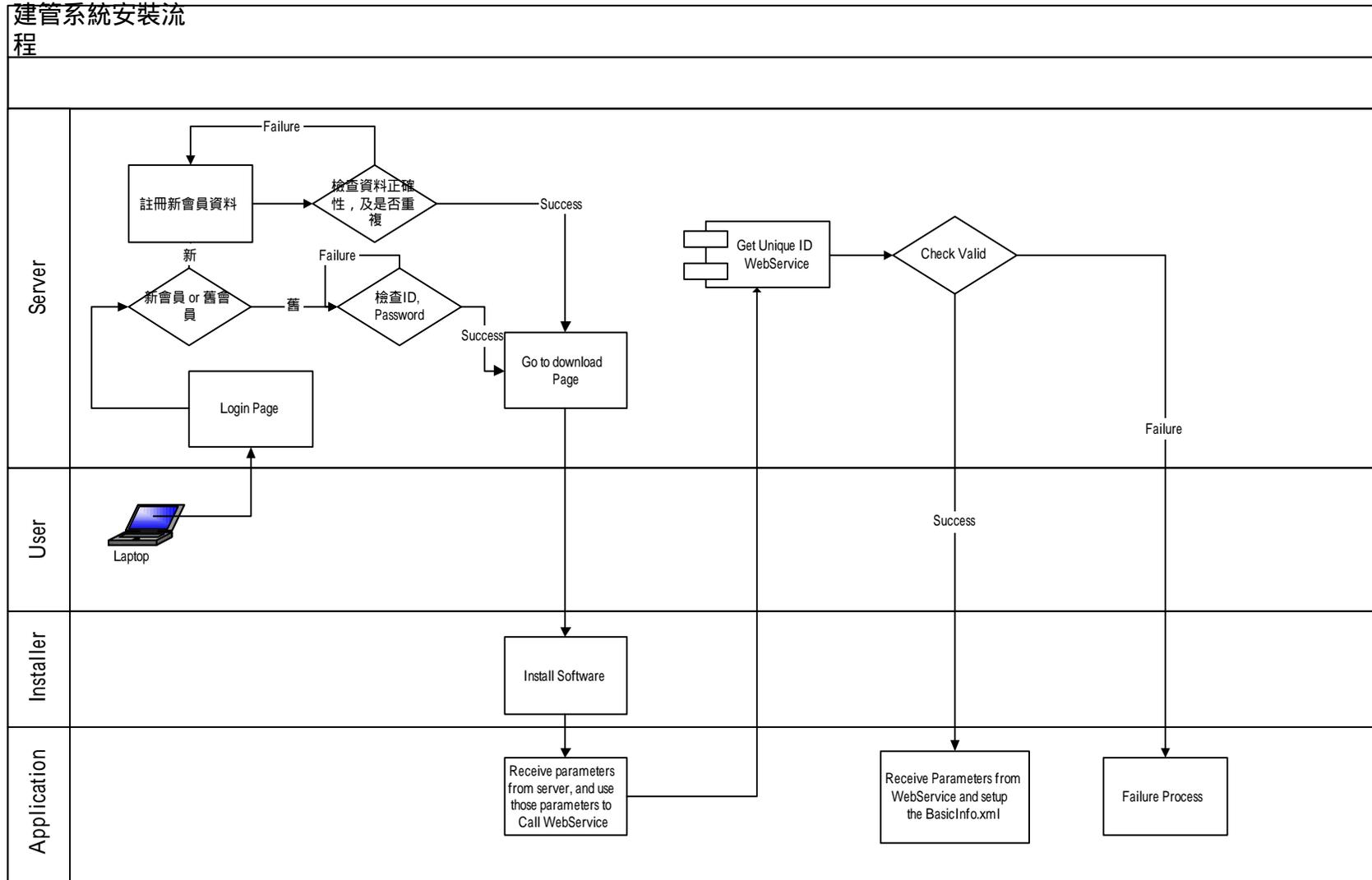
## 附錄 B：案件列表物件之狀態圖



## 附錄 C：編輯介面主要類別圖



## 附錄 D：系統安裝流程圖



## 附錄 E：系統之 XML Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 U (http://www.xmlspy.com) by jj (j) -->
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by jj (jj) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" elementFormDefault="qualified"
attributeFormDefault="qualified" version="1.0">
  <xs:element name="root">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="建築物" type="建築物" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
    <xs:key name="rootKey1" msdata:PrimaryKey="true">
      <xs:selector xpath="//建築物"/>
      <xs:field xpath="BID"/>
    </xs:key>
  </xs:element>
  <xs:complexType name="建築物">
    <xs:sequence>
      <xs:element name="BID" type="xs:string" minOccurs="0"/>
      <xs:element name="臨時建築" type="xs:boolean" minOccurs="0"/>
      <xs:element name="管理單位" type="xs:string" minOccurs="0"/>
      <xs:element name="建築物名稱" type="xs:string" minOccurs="0"/>
      <xs:element name="建築類別" type="xs:string" minOccurs="0"/>
      <xs:element name="建築物使用性質" type="xs:string" minOccurs="0"/>
      <xs:element name="所屬行政區縣" type="xs:string" minOccurs="0"/>
      <xs:element name="所屬行政區鄉" type="xs:string" minOccurs="0"/>
      <xs:element name="郵遞區號" type="xs:string" minOccurs="0"/>
      <xs:element name="建造執照日期" type="xs:string" minOccurs="0"/>
      <xs:element name="建造執照字" type="xs:string" minOccurs="0"/>
      <xs:element name="建造執照號" type="xs:string" minOccurs="0"/>
      <xs:element name="使用執照日期" type="xs:string" minOccurs="0"/>
      <xs:element name="使用執照字" type="xs:string" minOccurs="0"/>
      <xs:element name="使用執照號" type="xs:string" minOccurs="0"/>
      <xs:element name="開工日期" type="xs:string" minOccurs="0"/>
      <xs:element name="施工期限" type="xs:string" minOccurs="0"/>
      <xs:element name="歷程表" type="歷程表" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="起造人" type="起造人" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="起造人_幢棟層戶" type="起造人_幢棟層戶" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="幢棟層戶" type="幢棟層戶" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="設計人" type="設計人" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

```

<xs:element name="監造人" type="監造人" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="承造人" type="承造人" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="簽證技師" type="簽證技師" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="基地概要" type="基地概要" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="建築概要" type="建築概要" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="樓層概要" type="樓層概要" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="地質概要" type="地質概要" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="結構概要" type="結構概要" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="用地概要" type="土地標示及使用範圍" minOccurs="0"
maxOccurs="unbounded"/>
  <xs:element name="使用共同壁協定" type="使用共同壁協定" minOccurs="0"
maxOccurs="unbounded"/>
  <xs:element name="使用道路情形" type="使用道路情形" minOccurs="0"
maxOccurs="unbounded"/>
  <xs:element name="雜項工作物概要" type="雜項工作物概要" minOccurs="0"
maxOccurs="unbounded"/>
  <xs:element name="作業文件" type="作業文件" minOccurs="0" maxOccurs="unbounded"/>
  <xs:element name="BIDRelation" type="BIDRelation" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="起造人">
  <xs:sequence>
    <xs:element name="序號" type="xs:string" minOccurs="0" msdata:AutoIncrement="true"/>
    <xs:element name="姓名" type="xs:string" minOccurs="0"/>
    <xs:element name="法定代表人" type="xs:string" minOccurs="0"/>
    <xs:element name="身分證統一編號" type="xs:string" minOccurs="0"/>
    <xs:element name="出生日期" type="xs:string" minOccurs="0"/>
    <xs:element name="電話" type="xs:string" minOccurs="0"/>
    <xs:element name="通訊處" type="xs:string" minOccurs="0"/>
    <xs:element name="住址" type="xs:string" minOccurs="0"/>
    <xs:element name="傳真" type="xs:string" minOccurs="0"/>
    <xs:element name="Email" type="xs:string" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="LFT" type="xs:long"/>
  <xs:attribute name="Parent" type="xs:integer"/>
  <xs:attribute name="Status" type="xs:integer"/>
</xs:complexType>
<xs:complexType name="起造人_幢棟層戶">
  <xs:sequence>
    <xs:element name="起造人" type="xs:string"/>
    <xs:element name="幢棟層戶" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="LFT" type="xs:long"/>

```

## 建築管理請照網路作業系統之研究 (II)

```
<xs:attribute name="Parent" type="xs:integer"/>
<xs:attribute name="Status" type="xs:integer"/>
</xs:complexType>
<xs:complexType name="幢棟層戶">
  <xs:sequence>
    <xs:element name="序號" type="xs:string" minOccurs="0" msdata:AutoIncrement="true"/>
    <xs:element name="編號" type="xs:string" minOccurs="0"/>
    <xs:element name="幢" type="xs:string" minOccurs="0"/>
    <xs:element name="棟" type="xs:string" minOccurs="0"/>
    <xs:element name="層" type="xs:string" minOccurs="0"/>
    <xs:element name="戶" type="xs:string" minOccurs="0"/>
    <xs:element name="門牌號碼" type="xs:string" minOccurs="0"/>
    <xs:element name="使用類組" type="xs:string" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="LFT" type="xs:long"/>
  <xs:attribute name="Parent" type="xs:integer"/>
  <xs:attribute name="Status" type="xs:integer"/>
</xs:complexType>
<xs:complexType name="設計人">
  <xs:sequence>
    <xs:element name="序號" type="xs:string" minOccurs="0" msdata:AutoIncrement="true"/>
    <xs:element name="姓名" type="xs:string" minOccurs="0"/>
    <xs:element name="設計人開業證書日期" type="xs:string" minOccurs="0"/>
    <xs:element name="設計人開業證書字" type="xs:string" minOccurs="0"/>
    <xs:element name="設計人開業證書號" type="xs:string" minOccurs="0"/>
    <xs:element name="事務所名稱" type="xs:string" minOccurs="0"/>
    <xs:element name="事務所地址" type="xs:string" minOccurs="0"/>
    <xs:element name="電話" type="xs:string" minOccurs="0"/>
    <xs:element name="傳真" type="xs:string" minOccurs="0"/>
    <xs:element name="Email" type="xs:string" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="LFT" type="xs:long"/>
  <xs:attribute name="Parent" type="xs:integer"/>
  <xs:attribute name="Status" type="xs:integer"/>
</xs:complexType>
<xs:complexType name="監造人">
  <xs:sequence>
    <xs:element name="序號" type="xs:string" minOccurs="0" msdata:AutoIncrement="true"/>
    <xs:element name="姓名" type="xs:string" minOccurs="0"/>
    <xs:element name="監造人開業證書日期" type="xs:string" minOccurs="0"/>
    <xs:element name="監造人開業證書字" type="xs:string" minOccurs="0"/>
    <xs:element name="監造人開業證書號" type="xs:string" minOccurs="0"/>
    <xs:element name="事務所名稱" type="xs:string" minOccurs="0"/>
    <xs:element name="事務所地址" type="xs:string" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="LFT" type="xs:long"/>
  <xs:attribute name="Parent" type="xs:integer"/>
  <xs:attribute name="Status" type="xs:integer"/>
</xs:complexType>
```

```

    <xs:element name="電話" type="xs:string" minOccurs="0"/>
    <xs:element name="傳真" type="xs:string" minOccurs="0"/>
    <xs:element name="Email" type="xs:string" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="LFT" type="xs:long"/>
  <xs:attribute name="Parent" type="xs:integer"/>
  <xs:attribute name="Status" type="xs:integer"/>
</xs:complexType>
<xs:complexType name="簽證技師">
  <xs:sequence>
    <xs:element name="序號" type="xs:string" minOccurs="0" msdata:AutoIncrement="true"/>
    <xs:element name="姓名" type="xs:string" minOccurs="0"/>
    <xs:element name="執業執照日期" type="xs:string" minOccurs="0"/>
    <xs:element name="執業執照字" type="xs:string" minOccurs="0"/>
    <xs:element name="執業執照號" type="xs:string" minOccurs="0"/>
    <xs:element name="內政部許可文號日期" type="xs:string" minOccurs="0"/>
    <xs:element name="內政部許可文號字" type="xs:string" minOccurs="0"/>
    <xs:element name="內政部許可文號號" type="xs:string" minOccurs="0"/>
    <xs:element name="事務所名稱" type="xs:string" minOccurs="0"/>
    <xs:element name="事務所地址" type="xs:string" minOccurs="0"/>
    <xs:element name="事務所電話" type="xs:string" minOccurs="0"/>
    <xs:element name="傳真" type="xs:string" minOccurs="0"/>
    <xs:element name="Email" type="xs:string" minOccurs="0"/>
    <xs:element name="簽證項目" type="xs:string" minOccurs="0"/>
    <xs:element name="計畫書" type="xs:string" minOccurs="0"/>
    <xs:element name="說明書" type="xs:string" minOccurs="0"/>
    <xs:element name="圖樣" type="xs:string" minOccurs="0"/>
    <xs:element name="備註" type="xs:string" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="LFT" type="xs:long"/>
  <xs:attribute name="Parent" type="xs:integer"/>
  <xs:attribute name="Status" type="xs:integer"/>
</xs:complexType>
<xs:complexType name="承造人">
  <xs:sequence>
    <xs:element name="序號" type="xs:string" minOccurs="0" msdata:AutoIncrement="true"/>
    <xs:element name="營造業名稱" type="xs:string" minOccurs="0"/>
    <xs:element name="統一編號" type="xs:string" minOccurs="0"/>
    <xs:element name="負責人" type="xs:string" minOccurs="0"/>
    <xs:element name="登記證等級日期" type="xs:string" minOccurs="0"/>
    <xs:element name="登記證等級字" type="xs:string" minOccurs="0"/>
    <xs:element name="登記證等級號" type="xs:string" minOccurs="0"/>
    <xs:element name="專任工程人員" type="xs:string" minOccurs="0"/>
    <xs:element name="專任工程人員證書字號日期" type="xs:string" minOccurs="0"/>
  </xs:sequence>

```

## 建築管理請照網路作業系統之研究 (II)

```
<xs:element name="專任工程人員證書字號字" type="xs:string" minOccurs="0"/>
<xs:element name="專任工程人員證書字號號" type="xs:string" minOccurs="0"/>
<xs:element name="電話" type="xs:string" minOccurs="0"/>
<xs:element name="營業地址" type="xs:string" minOccurs="0"/>
<xs:element name="傳真" type="xs:string" minOccurs="0"/>
<xs:element name="Email" type="xs:string" minOccurs="0"/>
</xs:sequence>
<xs:attribute name="LFT" type="xs:long"/>
<xs:attribute name="Parent" type="xs:integer"/>
<xs:attribute name="Status" type="xs:integer"/>
</xs:complexType>
<xs:complexType name="土地所有權人">
  <xs:sequence>
    <xs:element name="序號" type="xs:integer" minOccurs="0" msdata:AutoIncrement="true"/>
    <xs:element name="姓名" type="xs:string" minOccurs="0"/>
    <xs:element name="出生日期" type="xs:string" minOccurs="0"/>
    <xs:element name="身分證統一編號" type="xs:string" minOccurs="0"/>
    <xs:element name="電話" type="xs:string" minOccurs="0"/>
    <xs:element name="住址" type="xs:string" minOccurs="0"/>
    <xs:element name="權利範圍" type="xs:double" minOccurs="0"/>
    <xs:element name="傳真" type="xs:string" minOccurs="0"/>
    <xs:element name="備註" type="xs:string" minOccurs="0"/>
    <xs:element name="Email" type="xs:string" minOccurs="0"/>
    <xs:element name="同意使用年期" type="xs:integer" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="使用共同壁協定">
  <xs:sequence>
    <xs:element name="序號" type="xs:integer" minOccurs="0" msdata:AutoIncrement="true"/>
    <xs:element name="地號" type="xs:string" minOccurs="0"/>
    <xs:element name="地址" type="xs:string" minOccurs="0"/>
    <xs:element name="鄰基地境界線" type="xs:string" minOccurs="0"/>
    <xs:element name="所有權人" type="所有權人" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="LFT" type="xs:long"/>
  <xs:attribute name="Parent" type="xs:integer"/>
  <xs:attribute name="Status" type="xs:integer"/>
</xs:complexType>
<xs:complexType name="基地概要">
  <xs:sequence>
    <xs:element name="序號" type="xs:string" minOccurs="0" msdata:AutoIncrement="true"/>
    <xs:element name="基地狀況" type="xs:string" minOccurs="0"/>
    <xs:element name="建築線指定日期" type="xs:string" minOccurs="0"/>
    <xs:element name="建築線指定字" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

```

<xs:element name="建築線指定號" type="xs:string" minOccurs="0"/>
<xs:element name="法定建蔽率" type="xs:double" minOccurs="0"/>
<xs:element name="法定容積率" type="xs:double" minOccurs="0"/>
<xs:element name="所在區域" type="xs:string" minOccurs="0"/>
<xs:element name="土地使用分區或編定用地" type="xs:string" minOccurs="0"/>
<xs:element name="基地面積合計" type="xs:double" minOccurs="0"/>
<xs:element name="騎樓地面積" type="xs:double" minOccurs="0"/>
<xs:element name="有效基地面積" type="xs:double" minOccurs="0"/>
<xs:element name="法定空地面積" type="xs:double" minOccurs="0"/>
<xs:element name="實際空地面積" type="xs:double" minOccurs="0"/>
<xs:element name="法定綠化面積" type="xs:double" minOccurs="0"/>
<xs:element name="實際綠化面積" type="xs:double" minOccurs="0"/>
<xs:element name="佔用地面積" type="xs:double" minOccurs="0"/>
<xs:element name="退縮地面積" type="xs:double" minOccurs="0"/>
<xs:element name="其他基地面積" type="xs:double" minOccurs="0"/>
<xs:element name="其他基地類別" type="類別面積" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="LFT" type="xs:long"/>
<xs:attribute name="Parent" type="xs:integer"/>
<xs:attribute name="Status" type="xs:integer"/>
</xs:complexType>
<xs:complexType name="建築概要">
  <xs:sequence>
    <xs:element name="序號" type="xs:string" minOccurs="0" msdata:AutoIncrement="true"/>
    <xs:element name="建築物用途" type="xs:string" minOccurs="0"/>
    <xs:element name="設計容積率" type="xs:double" minOccurs="0"/>
    <xs:element name="設計建蔽率" type="xs:double" minOccurs="0"/>
    <xs:element name="工程造價概算" type="xs:double" minOccurs="0"/>
    <xs:element name="建築面積" type="xs:double" minOccurs="0"/>
    <xs:element name="總樓地板面積" type="xs:double" minOccurs="0"/>
    <xs:element name="容積總樓地板面積" type="xs:double" minOccurs="0"/>
    <xs:element name="開放空間面積" type="xs:double" minOccurs="0"/>
    <xs:element name="防空避難室面積" type="xs:double" minOccurs="0"/>
    <xs:element name="其中兼停車空間" type="xs:double" minOccurs="0"/>
    <xs:element name="高層建築物緩衝空間" type="xs:double" minOccurs="0"/>
    <xs:element name="變更說明及理由" type="xs:string" minOccurs="0"/>
    <xs:element name="建築概要面積" type="類別面積" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="層棟戶數" type="層棟戶數" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="停車輛數" type="停車輛數" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="LFT" type="xs:long"/>
  <xs:attribute name="Parent" type="xs:integer"/>

```

```
<xs:attribute name="Status" type="xs:integer"/>
</xs:complexType>
<xs:complexType name="樓層概要">
  <xs:sequence>
    <xs:element name="序號" type="xs:string" minOccurs="0" msdata:AutoIncrement="true"/>
    <xs:element name="棟別" type="xs:string" minOccurs="0"/>
    <xs:element name="層別" type="xs:string" minOccurs="0"/>
    <xs:element name="申請面積" type="xs:double" minOccurs="0"/>
    <xs:element name="露臺面積" type="xs:double" minOccurs="0"/>
    <xs:element name="陽臺面積" type="xs:double" minOccurs="0"/>
    <xs:element name="樓層高度" type="xs:double" minOccurs="0"/>
    <xs:element name="使用類組" type="xs:string" minOccurs="0"/>
    <xs:element name="範圍面積" type="xs:double" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="LFT" type="xs:long"/>
  <xs:attribute name="Parent" type="xs:integer"/>
  <xs:attribute name="Status" type="xs:integer"/>
</xs:complexType>
<xs:complexType name="地質概要">
  <xs:sequence>
    <xs:element name="序號" type="xs:string" minOccurs="0"/>
    <xs:element name="類別" type="xs:string" minOccurs="0"/>
    <xs:element name="內容" type="xs:string" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="LFT" type="xs:long"/>
  <xs:attribute name="Parent" type="xs:integer"/>
  <xs:attribute name="Status" type="xs:integer"/>
</xs:complexType>
<xs:complexType name="結構概要">
  <xs:sequence>
    <xs:element name="序號" type="xs:string" minOccurs="0"/>
    <xs:element name="類別" type="xs:string" minOccurs="0"/>
    <xs:element name="內容" type="xs:string" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="LFT" type="xs:long"/>
  <xs:attribute name="Parent" type="xs:integer"/>
  <xs:attribute name="Status" type="xs:integer"/>
</xs:complexType>
<xs:complexType name="土地標示及使用範圍">
  <xs:sequence>
    <xs:element name="序號" type="xs:string" minOccurs="0" msdata:AutoIncrement="true"/>
    <xs:element name="都市計劃區" type="xs:string" minOccurs="0"/>
    <xs:element name="分區類別" type="xs:string" minOccurs="0"/>
    <xs:element name="土地使用分區或編定用地" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

```

<xs:element name="地號" type="xs:string" minOccurs="0"/>
<xs:element name="面積" type="xs:double" minOccurs="0"/>
<xs:element name="同意使用面積" type="xs:double" minOccurs="0"/>
<xs:element name="土地所有權人" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="LFT" type="xs:long"/>
<xs:attribute name="Parent" type="xs:integer"/>
<xs:attribute name="Status" type="xs:integer"/>
</xs:complexType>
<xs:complexType name="使用道路情形">
  <xs:sequence>
    <xs:element name="使用道路" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="使用期限" type="xs:string" minOccurs="0"/>
    <xs:element name="使用理由" type="xs:string" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="LFT" type="xs:long"/>
  <xs:attribute name="Parent" type="xs:integer"/>
  <xs:attribute name="Status" type="xs:integer"/>
</xs:complexType>
<xs:complexType name="使用道路">
  <xs:sequence>
    <xs:element name="序號" type="xs:string" minOccurs="0" msdata:AutoIncrement="true"/>
    <xs:element name="路" type="xs:string" minOccurs="0"/>
    <xs:element name="巷" type="xs:string" minOccurs="0"/>
    <xs:element name="弄" type="xs:string" minOccurs="0"/>
    <xs:element name="號" type="xs:string" minOccurs="0"/>
    <xs:element name="長" type="xs:double" minOccurs="0"/>
    <xs:element name="寬" type="xs:double" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="雜項工作物概要">
  <xs:sequence>
    <xs:element name="序號" type="xs:string" minOccurs="0" msdata:AutoIncrement="true"/>
    <xs:element name="工作項目" type="xs:string" minOccurs="0"/>
    <xs:element name="長度" type="xs:double" minOccurs="0"/>
    <xs:element name="高度" type="xs:double" minOccurs="0"/>
    <xs:element name="面積" type="xs:double" minOccurs="0"/>
    <xs:element name="工程造价" type="xs:double" minOccurs="0"/>
    <xs:element name="其他" type="xs:string" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="LFT" type="xs:long"/>
  <xs:attribute name="Parent" type="xs:integer"/>
  <xs:attribute name="Status" type="xs:integer"/>

```

```
</xs:complexType>
<xs:complexType name="層棟戶數">
  <xs:sequence>
    <xs:element name="設計建築物高度" type="xs:double" minOccurs="0"/>
    <xs:element name="構造種類" type="xs:string" minOccurs="0"/>
    <xs:element name="簷高" type="xs:double" minOccurs="0"/>
    <xs:element name="幢" type="xs:string" minOccurs="0"/>
    <xs:element name="棟" type="xs:string" minOccurs="0"/>
    <xs:element name="地上層" type="xs:integer" minOccurs="0"/>
    <xs:element name="地下層" type="xs:integer" minOccurs="0"/>
    <xs:element name="戶" type="xs:integer" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="停車輛數">
  <xs:sequence>
    <xs:element name="總設計停車輛數" type="xs:integer" minOccurs="0"/>
    <xs:element name="法定輛數" type="xs:integer" minOccurs="0"/>
    <xs:element name="鼓勵輛數" type="xs:integer" minOccurs="0"/>
    <xs:element name="自行增設輛數" type="xs:integer" minOccurs="0"/>
    <xs:element name="室內停車輛數" type="xs:integer" minOccurs="0"/>
    <xs:element name="室外停車輛數" type="xs:integer" minOccurs="0"/>
    <xs:element name="法定機車輛數" type="xs:integer" minOccurs="0"/>
    <xs:element name="工廠裝卸位" type="xs:integer" minOccurs="0"/>
    <xs:element name="出入方式" type="xs:string" minOccurs="0"/>
    <xs:element name="停車方式" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="BIDRelation">
  <xs:sequence>
    <xs:element name="PBID" type="xs:string" minOccurs="0"/>
    <xs:element name="CBID" type="xs:string" minOccurs="0"/>
    <xs:element name="日期" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="歷程表">
  <xs:sequence>
    <xs:element name="日期" type="xs:string"/>
    <xs:element name="Event_Name" type="xs:string"/>
    <xs:element name="Value" type="xs:string"/>
    <xs:element name="執照日期" type="xs:string" minOccurs="0"/>
    <xs:element name="執照字" type="xs:string" minOccurs="0"/>
    <xs:element name="執照號" type="xs:string" minOccurs="0"/>
    <xs:element name="備註" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

```
</xs:complexType>
<xs:complexType name="作業文件">
  <xs:sequence>
    <xs:element name="類別" type="xs:string" minOccurs="0"/>
    <xs:element name="內容" type="xs:string" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="LFT" type="xs:long"/>
  <xs:attribute name="Parent" type="xs:integer"/>
  <xs:attribute name="Status" type="xs:integer"/>
</xs:complexType>
<xs:complexType name="類別面積">
  <xs:sequence>
    <xs:element name="類別" type="xs:string" minOccurs="0"/>
    <xs:element name="面積" type="xs:double" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="載重">
  <xs:sequence>
    <xs:element name="類別" type="xs:string" minOccurs="0"/>
    <xs:element name="大小" type="xs:double" minOccurs="0"/>
    <xs:element name="用途" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="所有權人">
  <xs:sequence>
    <xs:element name="序號" type="xs:string" minOccurs="0"/>
    <xs:element name="房屋所有權人姓名" type="xs:string" minOccurs="0"/>
    <xs:element name="房屋所有權人身分證統一編號" type="xs:string" minOccurs="0"/>
    <xs:element name="房屋所有權人住址" type="xs:string" minOccurs="0"/>
    <xs:element name="土地所有權人姓名" type="xs:string" minOccurs="0"/>
    <xs:element name="土地所有權人身分證統一編號" type="xs:string" minOccurs="0"/>
    <xs:element name="土地所有權人住址" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

## 附錄 F：系統相關軟體量測值

Package Name	Class Name	方法	屬性	事件	LOC	
DataClass	absbaseinfo.vb	3	8	0	77	
	absbinding.vb	37	32	8	1086	
	absKVBinding.vb	7	6	0	111	
	BaseInfo 承造人.vb	17	17	0	89	
	BaseInfo 建築師.vb	11	3	0	69	
	BaseInfo 簽證技師.vb	14	3	0	81	
	CaseFactory.vb	4	8	0	130	
	HistoryBinding.vb	9	9	0	121	
	KV 地質概要 Binding.vb	7	7	0	10	
	KV 作業文件 Binding.vb	7	7	0	110	
	KV 結構概要 Binding.vb	3	4	0	10	
	用地概要 Binding.vb	11	11	0	147	
	地質概要 Binding.vb	3	4	0	51	
	使用共同壁協定 Binding.vb	6	7	0	90	
	使用道路情形 Binding.vb	4	5	0	64	
	承造人 Binding.vb	17	17	0	225	
	建築概要明細 Binding.vb	17	17	0	236	
	起造人_幢棟層戶 Binding.vb	5	4	0	57	
	起造人 Binding.vb	12	12	0	162	
	基本資料 Binding.vb	19	2	0	76	
	基地概要 Binding.vb	22	19	0	275	
	設計人 Binding.vb	12	12	0	160	
	結構概要 Binding.vb	3	4	0	51	
	監造人 Binding.vb	12	12	0	159	
	幢棟層戶 Binding.vb	9	10	0	130	
	樓層概要 Binding.vb	11	11	0	153	
	雜項工作物概要 Binding.vb	9	9	0	119	
	簽證技師 Binding.vb	18	20	0	159	
	DataTools	DataEditor	7	17	0	274
		DataMemberValueEditor	2	2	0	
DataMover		8	9	0	159	
DataNav		5	14	1	211	
DataSimpleEditor		10	14	3	249	
DataSrcValueEditor		10	15	3		

	DataTool	19	25	3	439
	FullDataTools	31	51	7	985
	mdlFunctions	1	0	0	25
LFT	IDGen	2	2	0	82
	LFT	23	20	0	750
	XFileOp	5	0	0	211
PanelForm	frmPanelBase	5	5	0	98
	frmPnlUpload	4	12	0	142
	frmPnl 用地概要	4	12	0	370
	frmPnl 共同壁協議	4	13	0	328
	frmPnl 地質概要	4	19	0	872
	frmPnl 技師查核項目	4	19	0	1100
	frmPnl 使用道路申請	6	22	0	236
	frmPnl 承造人	4	39	0	614
	frmPnl 拆除物概要	7	52	0	242
	frmPnl 拆除物樓層概要	3	80	0	312
	frmPnl 建築師查核項目	4	131	0	1636
	frmPnl 建築概要明細	4	42	0	627
	frmPnl 起造人	17	62	0	1141
	frmPnl 基本資料	10	119	0	1956
	frmPnl 基地概要	5	52	0	745
	frmPnl 執照更正	3	5	0	101
	frmPnl 執照補發	3	25	0	369
	frmPnl 設計人	4	27	0	443
	frmPnl 報表預覽	17	13	0	693
	frmPnl 結構概要	7	93	0	1414
	frmPnl 監造人	4	27	0	442
	frmPnl 監督人	4	22	0	387
	frmPnl 增建樓層概要	4	63	0	892
	frmPnl 樓層概要	4	27	0	434
	frmPnl 檢討項目簽證	4	39	0	516
	frmPnl 雜項工作物概要	4	25	0	425
	frmPnl 簽證技師	4	43	0	680
	frmPnl 變更使用執照_樓層概要	4	32	0	506
	frmPnl 變更承造人	4	70	0	1042
	frmPnl 變更起造人	24	125	0	2037
	frmPnl 變更設計用地概要	5	37	0	594
	frmPnl 變更設計建築概要	3	80	0	1134

## 建築管理請照網路作業系統之研究 (II)

	frmPnl 變更設計基地概要	4	98	0	1325
	frmPnl 變更設計樓層概要	3	51	0	776
	frmPnl 變更設計雜項工作物概要	4	46	0	768
	frmPnl 變更監造人	4	50	0	765
	Form2	3	1	0	39
popform	eEditMode	0	1	0	
	ePopFormType	0	1	0	
	frmPopBase	14	14	1	270
	frmPop 土地所有權人	8	22	0	364
	frmPop 共同壁所有權人	9	14	0	242
	frmPop 使用道路	8	16	0	307
	frmPop 其他基地類別	8	6	0	146
	frmPop 拆除層棟戶數	8	16	0	307
	frmPop 建築概要面積	8	6	0	146
	frmPop 停車輛數	8	24	0	448
	frmPop 層棟戶數	8	22	0	389
	mdlFunctions	1	0	0	26
	PopFormFactory	6	2	0	100
	XmlNodeDetailEditor	10	24	0	401
SimpleControls	Address	12	32	1	607
	Area	7	33	1	430
	DataGroupControl	11	19	2	371
	DateTextBox	9	12	0	221
	eSeparateLineType	0	1	0	
	frm 使用分區	4	17	0	252
	GNum	10	20	1	368
	HideTab	14	18	0	269
	ICurrValue	1	0	0	3
	ImageLabel	4	4	0	88
	LandType	4	10	0	199
	LandTypeData	2	9	0	
	PopupTextBox	5	5	0	88
	score	11	6	1	173
	SeparateLine	18	6	2	71
	_Structure	7	9	1	142
	UseTypes	6	15	1	252
	XmlTreeList	9	6	0	123
	XMLTreeView	20	24	2	503

	地址	4	39	0	297
	地號	4	18	0	151
ToolClass	AutoTabClass	9	2	0	140
	cFTP	22	15	3	549
	clsCryptStream	7	5	0	
	clsEnCode	5	2	0	
	clsZLib	1	0	0	57
	DownloadProcess	3	2	1	68
	Env	17	55	0	481
	frmMessage	4	5	0	100
	frmNewUser	8	16	0	248
	LogFileOp	2	4	0	27
	mdiGlobal	13	5	0	353
	mEnv	1	0	0	481
	NameValues	6	11	0	159
	SysArgs	4	12	0	99
	UserInfo	14	17	0	411
建築管理請照網 路作業系統	clsMain	11	7	0	178
	frmGetUID	6	12	247	247
	frmlogin	14	13	0	224
	frmMainForm	8	8	0	264
	frmPassword	4	11	0	170
	frmUserManage	11	12	0	254
	frm 使用分類	30	37	0	
	frm 版本	4	7	0	138
	frm 建築構造種類	57	48	13	58
	frm 基本資料編修	13	112	0	1695
	frm 符號表	3	10	0	164
	frm 縣市設定	5	12	0	228
	myMenu	4	3	0	83
	myMenuItem	8	14	0	254
	Module1	3	6	0	78
	QueryBuilder	11	17	0	217
	tw	10	0	0	
	案件列表 1	30	83	0	1812
執照申請 2	frm 執照申請 Main	9	11	0	426

報表分類	報表名稱	LOC
共同檔案	A11-2 設計人名冊一	102
	A11-2 設計人名冊二	104
	A11-3 設計人名冊	41
	A11-4 建築物概要表	77
	A11-5 委託書	45
	A11-5 委託書_變更	45
	A11-6 建築物增建概要表	118
	A12-12 地號表	44
	A12-4 土地使用權同意書	131
	A12-5 使用共同壁協定書	100
	A13-1 建造執照及雜項執照變更設計審查表	28
	A13-2 建照執照及雜項執照規定項目審查表	59
	A13-3 使用道路申請表	68
	A21-4 雜項工作物概要表	96
	B11-3 承造人名冊	58
	B11-4 監造人名冊	41
	建造執照及雜項執照建築師簽證表	33
	建造執照及雜項執照建築師簽證表_變更	33
	建築物結構與設備專業技師簽證報告	96
	使用執照	C11-1 使用執照申請表
C11-2 起造人名冊三		105
C11-2 起造人名冊四		107
C11-3 建築物概要明細表		451
C11-4 雜項工作物概要明細表		309
C12-1 使用執照審查表		30
拆除執照		D11-1 拆除執照申請書
	D11-2 申請人明冊	38
	D11-3 拆除物概要表	47
	D13-1 拆除執照審查表	25
建造執照	A11-1 建造執照申請書	420
	A11-7 建築物概要明細表	437
開工及施工勘驗	B11-1 建築工程開工申報書	168

	B11-2 建築工程開工展期申報書	99
	B12-1 建照執照變更起照人承造人監造人審查表	39
	B13-1 變更起造人申報書	196
	B13-2 變更起造人名冊一	116
	B13-2 變更起造人名冊二	97
	B13-3 變更承造人申報書	215
	B13-4 變更承照人名冊	107
	B13-5 變更監造人申報書	198
	B13-6 變更監造人名冊	73
	B14-1 建築工程堪驗申報書	169
	B21-1 建築工程竣工展期申請書	184
舊表格	舊表-建造執造申請書	468
雜項執照	A21-1 雜項執照申請書	379
	A21-5 雜項工作物概要明細表	304
	A23-1 山坡地雜項執照審查表	28
變更使用執照	C21-1 變更使用執照申請書	175
	C21-2 變更使用執照概要表	66
	C21-3 變更使用執照檢討項目簽證表	88
	C21-4 申請人名冊	66
	C22-1 變更使用執照審查表	58
	C22-2 變更使用執照竣工堪驗審查表	30
變更設計	A13-5 建造執照及雜項執照規定項目審查表	170
	A31-1 第次變更設計申請書	450
	A31-4 建築物變更設計概要表	119
	A31-5 雜項工作物變更設計概要表	170
	A32-2 變更設計地號表	79
	9	58
		8251