

國立交通大學

土木工程學系
碩士論文

營建電子採購系統 XML 標準驗證系統
可行性探討

Feasibility Study of XML Standardization Validation for
Construction e-Procurement



研究生：王聖賢

指導教授：曾仁杰 博士

中華民國九十五年七月

營建電子採購系統 XML 驗證系統可行性探討

Feasibility Study of XML Standardization Validation for
Construction e-Procurement

研究生：王聖賢

Student : Sheng-Hsien Wang

指導教授：曾仁杰

Advisor : Ren-Jye Dzung

國立交通大學

土木工程學系

碩士論文

A Thesis

Submitted to Department of Civil Engineering

College of Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

In

Civil Engineering

July 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年七月

國立交通大學

論文口試委員會審定書

本校 土木工程 學系碩士班 王聖賢 君

所提論文：營建電子採購系統 XML 標準驗證系統可行性探討
Feasibility Study of XML Standardization
Validation for Construction e-Procurement

合於碩士資格標準、業經本委員會評審認可。

口試委員：曾仁杰 教授

曾仁杰

王維志 教授

王維志

黃世昌 教授

黃世昌

楊智斌 教授

楊智斌

楊亦東 教授

楊亦東

指導教授：曾仁杰 教授

曾仁杰

系主任：

洪良正

教授

中華民國 九十五年 七月 三日

營建電子採購系統 XML 驗證系統可行性探討

研究生：王聖賢

指導教授：曾仁杰 博士

國立交通大學土木工程學系（研究所）碩士班

摘要

營建署推廣營建電子化可說是不遺餘力，推動了許多相關計畫以求能達到營建電子化的效益；其中在 2000 年時，推動了為期四年的「營建產業資訊交換標準 aecXML 推廣計畫」，並於計畫第三年時制定完成營建產業常用的 12 項的 Schema 標準文件，目的是為了讓各家營建廠商在進行資訊交換時能有統一的標準。儘管如此，卻還是缺乏一套完善的營建產業 Schema 標準之驗證機制，來證明各家營建廠商所使用的營建 XML 表單是否遵循所制定的營建產業 Schema 標準。

本研究主要是擬定一套符合營建產業的驗證機制流程以及驗證系統，藉由營建署所制定的 12 項營建物料採購 Schema 標準，進而驗證廠商的電子採購系統所產出之 XML 文件是否合於營建產業 Schema 標準。根據 XML 文件中元素屬性的差異，透過驗證程式以及人為判斷的方式，將可能產生的錯誤歸類為九種類型，說明 XML 文件中產生錯誤的原因，促使能夠更快的發現文件錯誤之處。藉由這樣的驗證機制流程來檢驗廠商之 XML 文件，以求各家廠商交換的文件格式能夠統一。

關鍵詞：營建產業資訊交換標準、Schema、XML、驗證系統

Feasibility Study of XML Standardization Validation for Construction e-Procurement

Student : Sheng-Hsien Wang

Advisor : Ren-Jye Dzung

Department of Civil Engineering
National Chiao Tung University

Abstract

Although CPAMI(內政部營建署) constituted twelve schema in construction industry in 2004, but there is no mature validation procedures be applied. So, the main goal in this study is redact a better validation procedure and create a validation program to validate XML documents of e-Procurement. According the attribution of element, we can assort the errors to nine categories. Depend on the validation procedures and the validation system, we can find errors in XML documents. Use the validation procedures and validation program, we can make sure the XML documents would follow Schema standard, which constituted by CPAMI.

Keywords: Information exchanged standard in construction, Schema, XML, Validation system

誌謝

兩年，在這不算長也不算短的日子裡，終於拿到畢業證書了，的確得來不易。在這兩年間，有許多對我伸出援手的貴人，幫助我完成這一份巨著，必須要一個一個好好的感謝致意。首先，當然是要感謝論文的指導教授 曾仁杰老師，不辭辛勤的督促我的論文，讓我了解做論文的態度以及方法，增加了許多做事情的能力；另承蒙內審口試委員—王維志老師、黃世昌老師與黃玉霖老師；外審口試委員—楊智斌老師、楊亦東老師，在論文口試時給予諸多的指正與建議，使整個研究能更完整，在此致上深深的謝意。另外感謝在完成論文期間中給我許多指導以及在這兩年內幫助我完成許多事情的學長姊們：世旭學長、得榮學長、國勳學長、雅櫻學姐、珮茹學姊，沒有你們的協助，我想我這兩年一定會更加的水深火熱。

當然，還要感謝在這兩年一起奮鬥的同學們，讓我這兩年留下美好的回憶。首先是同組的姜辰、汎儀、煥雲、志仁，感謝你們時時的督促，才能讓我跟你們一起畢業；接下來是水田街的培浚、老皮、阿交、起輝，感謝你們在我疲累的回到家以後，還可以帶給我歡笑；接著是景翔、家豪、俊男、華偉、士評，感謝你們在這苦悶的研究日子裡陪我一起渡過。另外，也得要感謝各位學弟妹的幫助，讓我的研究生生活增色不少，尤其是名修、柏勳、鈺倫，真的感謝。最後，一定得要感謝陪伴我的女友---瑩潔，感謝妳在我這最忙碌的一年陪著我，聽我訴苦，並且在我低潮的時候給我安慰以及鼓勵，讓我有繼續往下走的動力。因此，我只想跟妳說：「有妳真好」。

最後，也是最重要的，當然是要深深地感謝隨時隨地支持著我的家人們：老爸、老媽、柏喬，因為有你們，我才能無後顧之憂的完成這兩年的研究，也才能順利的拿到碩士學位。沒有你們，也不可能有今天的我，因此我將此份喜悅以及榮耀獻給我生命中最重要的人們，並且永遠都以你們為榮。

目錄

摘要	I
Abstract	II
誌謝	III
目錄	IV
表目錄	VII
圖目錄	VIII
第 1 章 緒論	1
1.1 研究背景與動機	2
1.2 研究問題	2
1.3 研究目的	2
1.4 研究範圍與限制	2
1.5 研究方法	3
1.6 研究架構	3
1.7 研究流程	4
第 2 章 文獻回顧	6
2.1 營建物料採購資訊交換Schema標準	6
2.1.1 發展背景	6
2.1.2 營建產業 Schema 標準制訂項目	6
2.1.3 驗證示範	9
2.2 可延伸標記語言(XML)	9
2.2.1 XML簡介	10
2.2.2 XML文件架構	12
2.2.3 DTD 與 XML Schema	14
2.3 XML Schema介紹	16
2.3.1 XML Schema 特點	16
2.3.2 XML Schema 內容架構	17
2.4 文書及檔案管理電腦化作業規範驗證制度	18
2.5 小結	19
第 3 章 驗證機制功能與流程分析	20

3.1	前置準備階段	23
3.1.1	申請驗證	23
3.1.2	驗證準備	24
3.2	格式檢驗階段	24
3.3	系統驗證階段	27
3.3.1	文件內容之驗證	27
3.3.2	受驗系統呈現文件之驗證	43
3.3.3	驗證通過準則	45
3.4	簽發驗證報告與儲存資料階段	45
3.5	小結	46
第 4 章	系統說明與系統實作	48
4.1	系統說明	48
4.1.1	系統分析與功能	48
4.1.2	XML DOM 物件	49
4.1.3	Node 物件	50
4.1.4	ParseError 物件	52
4.2	實作成果	53
4.2.1	驗證系統介面展示	52
4.2.2	缺乏必要元素錯誤驗證	58
4.2.3	必要元素名稱錯誤驗證	60
4.2.4	非必要元素名稱錯誤驗證	62
4.2.5	必要及非必要元素資料型態錯誤驗證	64
4.2.6	未定義元素資料型態錯誤	66
4.2.7	必要元素及非必要元素出現次數錯誤驗證	68
4.2.8	未定義元素出現錯誤驗證	70
4.3	小結	72
第 5 章	結論與建議	73
5.1	結論	73
5.2	未來工作與建議	73
參考文獻		76
附錄一	營建物料採購Schema標準 1.0 版文件	80
附錄二	中華營建資訊標準協會驗證申請單填寫範本	130
附錄三	中華營建資訊標準協會驗證報告書填寫範本	131

附錄四	中華營建資訊標準協會驗證制度規範使用者手冊	134
附錄五	驗證系統程式碼	169
附錄六	論文口試委員建議及修訂	175



表目錄

表 2-1 營建產業物料採購資訊交換Schema標準總表	8
表 2-2 SGML、HTML和XML之間的差異比較.....	10
表 2-3 XML相關技術表.....	12
表 2-4 DTD與XML Schema比較表	15
表 3-1 錯誤類型列表.....	28
表 3-2 錯誤類型列表(實際驗證狀況).....	42
表 3-3 XML文件內容呈現範本.....	44
表 3-4 驗證機制流程整理表.....	47
表 4-1 Node物件之屬性值列表	51
表 4-2 Nodelist物件之屬性值列表	51
表 4-3 ParseError物件之屬性值列表.....	52



圖目錄

圖 1-1 研究流程圖.....	5
圖 2-1營建產業物料採購資訊交換Schema標準(草案)作業流程圖.....	7
圖 2-2 標準驗證制度範圍示意.....	9
圖 2-3 正確 XML 文件.....	13
圖 2-4 合法之 XML Schema 文件.....	17
圖 3-1 驗證機制流程圖.....	21
圖 3-2 受驗單位即驗證執行單位於各驗證階段執行之工作項目.....	22
圖 3-3 格式正確的XML文件(符合詢價單Schema1.0版).....	25
圖 3-4 必要元素未出現錯誤範例展示(以詢價單為例).....	30
圖 3-5 必要元素英文名稱錯誤範例展示(以詢價單為例).....	31
圖 3-6 非必要元素英文名稱錯誤範例展示(以詢價單為例).....	32
圖 3-7 必要元素資料型態錯誤範例展示(以詢價單為例).....	33
圖 3-8 非必要元素資料型態錯誤範例展示(以詢價單為例).....	34
圖 3-9 未定義元素資料型態錯誤範例展示(以詢價單為例).....	35
圖 3-10 必要元素出現次數錯誤範例展示(以詢價單為例).....	36
圖 3-11 非必要元素出現次數錯誤範例展示(以詢價單為例).....	37
圖 3-12 未定義元素出現次數錯誤範例展示(以詢價單為例).....	38
圖 3-13 人為判斷步驟圖.....	38
圖 3-14 測試之XML文件範例.....	38
圖 4-1 驗證系統IDEF圖.....	48
圖 4-2 XML DOM Tree.....	49
圖 4-3 XML與API關聯圖.....	50
圖 4-4 ParseError物件使用範例程式碼.....	52
圖 4-5驗證系統介面圖.....	53
圖 4-6受測試之XML文件(符合詢價單Schema).....	55
圖 4-7文件驗證成功訊息.....	57
圖 4-8包含<Project>元素之文件.....	58
圖 4-9缺少<Project>元素之文件.....	58
圖 4-10缺乏必要元素<Project>驗證訊息.....	59
圖 4-11必要元素名稱正確之文件.....	60
圖 4-12必要元素名稱錯誤之文件.....	60
圖 4-13必要元素<DeliveryLocation>名稱驗證訊息.....	61
圖 4-14非必要元素名稱正確之文件.....	62
圖 4-15非必要元素名稱錯誤之文件.....	62
圖 4-16非必要元素<AcceptTerm>名稱驗證訊息.....	63
圖 4-17元素資料型態正確之文件.....	64

圖 4-18 元素資料型態錯誤之文件	64
圖 4-19 <SheetDate> 以及 <InvitationDate> 資料型態驗證訊息	65
圖 4-20 未定義元素型態正確之文件	66
圖 4-21 未定義元素型態錯誤之文件	66
圖 4-22 未定義元素 <ProjectStartDate> 驗證訊息	67
圖 4-23 元素出現次數正確之文件	68
圖 4-24 元素出現次數錯誤之文件	68
圖 4-25 <QuotationDeadline> 及 <ProjectLocation> 出現次數驗證訊息	69
圖 4-26 未定義元素出現次數正確之文件	70
圖 4-27 未定義元素出現次數錯誤之文件	70
圖 4-28 未定義元素 <ProjectStartDate> 出現次數驗證訊息	71



第1章 緒論

1.1 研究背景與動機

營建產業自從導入電子化後，已由過往使用傳統紙張的模式演變為網路型態來傳遞資料，不但節省許多人工作業的時間，也同時避免重複作業。但是也因為導入電子化後，廠商的資料可以大量的交換，使得各家廠商的伺服器對資料的相容性以及資料交換格式變的格外重要；另一方面，營建署近幾年致力於補助廠商進行電子化供應鏈體系建置，所遇到的問題癥結為不同電子化供應鏈體系間廠商資訊交換的問題。由此看來，若不預先制訂資訊交換之標準，將可能造成廠商必須準備多樣資訊交換格式與內容，才足以滿足不同體系廠商進行採購電子化作業，形成不必要之浪費。

最近電子商務的崛起，使得大部分的交易行動可以在網路平台上進行，營建產業也是如此，所以網路平台的功能性與互動性日趨重要。HTML(HyperText Markup Language)即為統一網頁資訊在不同平台以相同方式顯示之語言，適合企業建立產品和服務項目之宣傳。但隨著時間的推進，HTML 已漸漸不足以符合功能強大的資訊交換平台，因此可擴充式標示語言(XML, eXtensible Markup Language)已變成新一代網路資訊格式。

因為 XML 文件寫法沒有既定的標準，是按照各家廠商的需求而制定，造成各家廠商的表單內容及格式都不相同，在資訊交換時會有資料格式不符及錯亂的情況產生。為避免這種情形發生，營建署在 2001 年實施了一個「營建產業資訊交換標準 aecXML 推廣計畫」，主要的工作內容是調查各家廠商表單的容及需求，然後利用 XML 撰寫一份標準文件供各家廠商參考，方便在制定營建產業表單有個依據；而這份 XML 文件即為 XML Schema，利用 XML Schema 來驗證廠商所交換的 XML 表單是否達到要求，以求資訊交換的正確性與一致性。

雖然 Schema 標準已制定完成，但是目前缺乏一套客觀的系統化方法，可以證明各家廠商所制訂出來的電子採購文件是否符合營建物料採購資訊交換 Schema 標準。換句話說，必須要有一套完整的驗證程序，才能來檢驗廠商之營建電子採購系統是否合乎於所制訂的 XML Schema 標準。因此本研究主要是為規劃及擬定一套完整的驗證流程和驗證系統，讓廠商可以按照所擬定的驗證流程來執行驗證的動作。

1.2 研究問題

由於營建電子採購 XML 文件的使用日趨頻繁，所以利用營建物料採購資訊交換 Schema 標準來制定 XML 文件也是必然的趨勢。儘管資訊交換廠商聲明所使用的 XML 文件是符合 Schema 標準，但是我們卻無法判斷是否屬實，是因為目前缺少一套較為客觀的系統化方法來檢驗廠商電子採購系統是否真的符合營建物料採購資訊交換 Schema 標準。因此，本研究問題有以下兩點：

1. 目前缺乏一套完整的營建資訊標準驗證流程來執行驗證的手續。
2. 缺少一個與本研究相符之營建資訊交換標準 XML 文件驗證系統。

1.3 研究目的

為了探討上述兩點的問題，因此本研究主要的目的如下兩點：

1. 建立驗證「營建物料資訊交換標準 1.0 版」Schema 之作業流程規範。
2. 建立驗證「營建物料資訊交換標準 1.0 版」Schema 之電子化系統。

期望藉由所擬定的營建物料採購資訊交換 Schema 標準驗證流程，中華營建資訊標準協會可參考本研究規劃之作業流程，規劃驗證機制；並提供標準之驗證系統，供未來實際應用之參考。

1.4 研究範圍與限制

本論文所研究的內容主要是在驗證流程的擬定以及驗證系統的功能分析與建置，有下列幾項的研究範圍限制：

1. 本研究著重於驗證機制流程及系統的建置，對於建立 Schema 表單的需求調查以及架構分析並不詳細討論。
2. 本研究所使用的 XML Schema 皆為營建署推行的「營建產業資訊交換標準 aecXML 推廣計畫」所擬定的 12 項 Schema 表單，根據這些 Schema 表單進行分析以及資料庫的建立，不額外再進行 Schema 表單的建立。
3. 文件元素名稱的資料主要以 12 項 Schema 表單的英文名稱為主，其他相關與

類似的英文名稱則不在探討的範圍內。

4. 本研究是利用 12 項營建產業 Schema 標準進行驗證的動作，但為了要簡便說明，只驗證其中一項 Schema 標準(詢價單)當做範例，其他沒談到之標準則依照所列舉的範例來執行驗證的程序，並非只是以詢價單 Schema 標準來驗證文件。

1.5 研究方法

本研究範疇大概可以分為二部份，詳細研究方法如下：

1. 擬訂一套完整的驗證流程

先行分析了解驗證機制所要檢驗的項目為何、定義出各階段主要的工作為何，並且參考行政院研考會所彙編的「文書及檔案管理電腦化作業規範驗證制度」，然後著手設計整個驗證機制的流程，從申請驗證開始，到最後的簽核驗證報表，完整的說明如何執行驗證手續。

2. 建置一套驗證系統

在系統建置方面，採用 VB6(Visual Basic 6.0)做為程式主要開發的工具，再透過 XML 相關的物件來建置整個的驗證系統。

1.6 研究架構

本研究架構分為五章，各章詳細內容如下：

第一章：緒論

敘述本研究之研究背景與動機、研究問題、研究目的、研究範圍與限制、研究方法以及研究流程等。

第二章：相關文獻回顧

本章節主要介紹 12 項營建物料採購資訊交換 Schema 標準、XML 概述以及 Schema 撰寫元素介紹，最後再大概說明行政院研考會所彙編的「文書及檔案管理電腦化作業規範驗證制度」。

第三章：驗證機制功能與流程分析

本章節主要是在擬定出完整的驗證機制流程，詳細分析受驗單位及驗證機構在各階段主要處理的項目；也將可能產生的錯誤類型做系統性的整理，並且說明實際上本研究可以做到的驗證程序。

第四章：系統說明與系統實作

分析驗證系統的流程，介紹建置系統所使用的技術理論以及XML 相關技術的應用；之後描述整個系統操作及運作方式，套入範例並且呈現整個系統的操作介面以及成果展示。

第五章：結論與建議

探討整個驗證流程以及驗證系統的結果與缺失，並提供未來研究發展的方向。

1.7 研究流程

本研究主要有兩大主題，一個是擬定一套完整的驗證流程，另一個是建置一套驗證系統來證明可行性。研究流程如圖 1-1 表示：

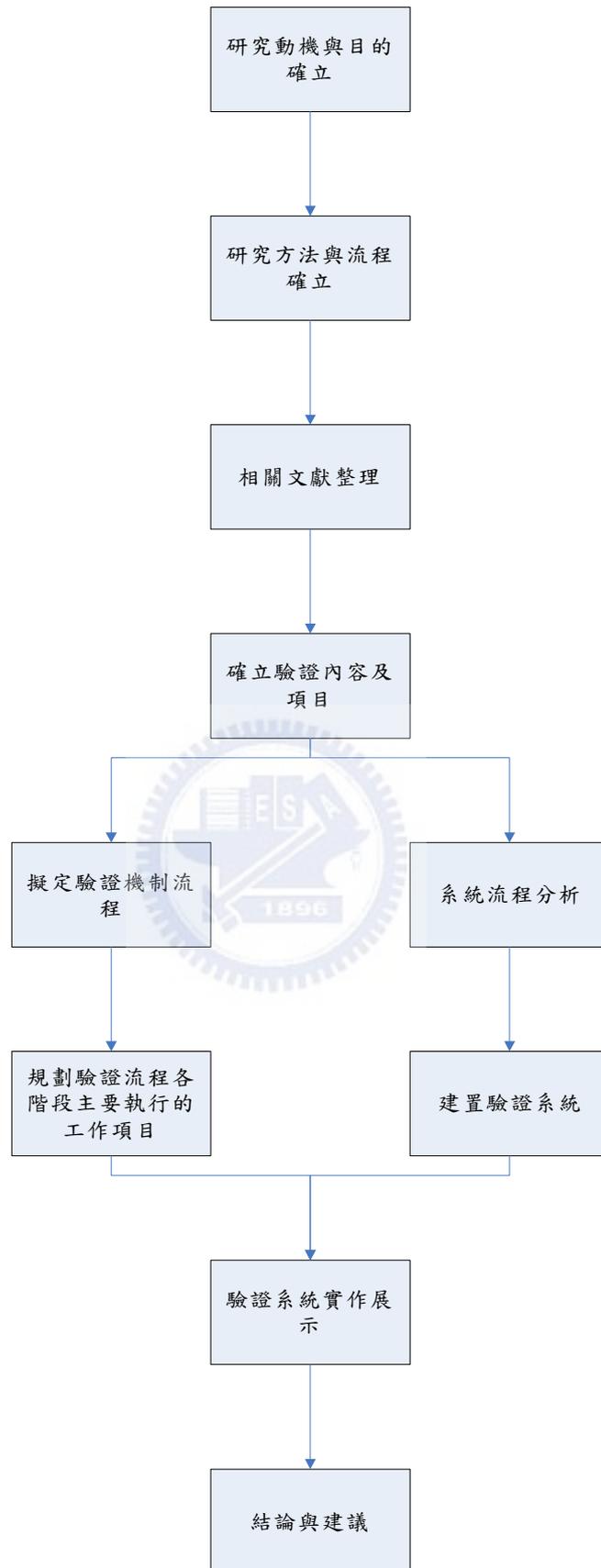


圖 1-1 研究流程圖

第2章 文獻回顧

2.1 營建物料採購資訊交換 Schema 標準介紹

2.1.1 發展背景

營建署於 2001 年推動「營建產業自動化與電子化計畫」，營建企業間電子化體系推廣目前已有 21 個供應鏈體系。由於各企業依其體系原有企業與作業特性發展其體系資訊交換平台，選擇資訊廠商合作之電子化採購交易系統平台，由於不同系統平台的資訊格式未能統一。

可藉由產業訂定資訊交換標準來有效解決，有了產業資訊交換標準，各軟體服務廠商仍可發展，唯須同時提供與標準相容之資訊儲存格式及交換機制。小型供應商便有可能以單一之軟體界面在各式各樣符合標準之採購平台進行報價；大型供應商也願意將其平台轉成與標準相容，以接受各式各樣符合標準之採購平台之詢價；採用不同採購平台之策略聯盟夥伴也可不受限於平台之相異，而進行重要採購資訊之交換與分享；中心廠商對於 ERP 及 SCM 軟體之選擇也變的更為靈活，也較容易整合。

2.1.2 營建產業 Schema 標準制訂項目

在研擬營建產業物料採購資訊交換 Schema 標準(草案)時，首先採取收集國內多家大型營造廠之採購作業手冊，針對其採購流程與採購作業中所使用之表單作分析，分析出營造廠商與供應商之間進行物料採購時所使用之標準為何。接著將表單內容參照 XML Schema 語法將表單內容，以 Schema 描述方式表達內容，以表格方式呈現標準。理由是以 Schema 語法呈現之內容，一般工程專業人員無法瞭解，審查過程耗費時間且有困難。透過以表格方式呈現，除工程人員可快速瞭解內容之外，資訊工程師透過表格亦可瞭解該如何透過 XML Schema 標準進行 Schema 之撰寫。(梁樾、陳春盛、曾仁杰，民 91 年)

圖 2-1 分別表示營造廠與供應商於採購、出貨及計價請款階段之一般流程，兩系列流程間透過箭線標註方式表明可能進行交換之資訊及流向。資訊項目名稱標註於箭線尾端代表尾端單位為主要之資訊填寫單位，標註於箭線中間者則代表雙方提供之資訊量相當。

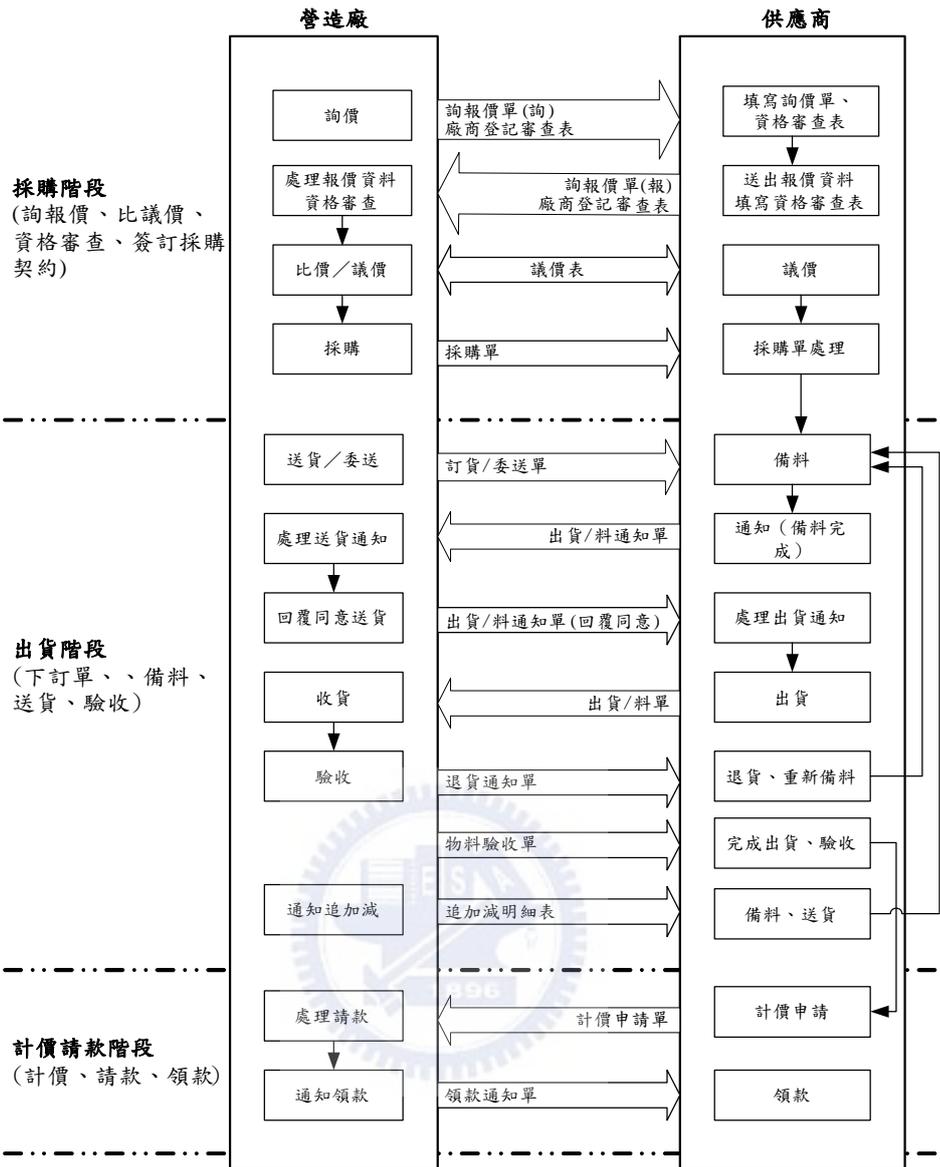


圖 2-1 營建產業物料採購資訊交換 Schema 標準(草案)作業流程圖
(梁樾、陳春盛、曾仁杰，民 91 年)

依據此流程所標註之資訊項目擬訂 12 個表單及其對應之 Schema，表 2-1 列出兩者之對應關係，並說明每種表單之使用時機。

表 2-1 營建產業物料採購資訊交換 Schema 標準總表
(梁樾、陳春盛、曾仁杰, 民 91 年)

表單名稱	表單傳送方式	表單使用說明
詢報價單	◎營造廠 to 供應商(詢價單) ◇供應商 to 營造廠(報價單)	詢價單、報價單整合成單一表單(Schema)。營造廠於詢價單填寫採購資訊傳給供應商，供應商填寫完畢(即為報價單)後傳回。
廠商登記審查表	◎營造廠 to 供應商(空的審查表) ◇供應商 to 營造廠(審查表)	營造廠提供此表予尚未登記之供應商，供其填寫傳回，以便營造廠建立供應商資料。
議價表	◎營造廠 to 供應商	營造廠於此表記錄供應商議價之結果。
採購單	◎營造廠 to 供應商	營造針對每一採購契約填寫採購採購資訊，傳送給物料供應商。
訂貨/委送單	◎營造廠 to 供應商	營造廠就採購單項目及及分批需求填寫此單，傳給供應商，予以通知備料。
出貨/料通知單	◇供應商 to 營造廠	供應商填寫此單，傳給營造廠，通知準備出貨，營造廠收到此單後，批覆意見給供應商，以告知供應商是否可以如其運送此批物料至交貨地點。
出貨/料單	◇供應商 to 營造廠	供應商填寫此單，於交貨時傳給營造廠簽收，以證明出貨/料。
計價申請單	◇供應商 to 營造廠 ◎營造廠 to 供應商(回覆申請意見)	供應商填寫此單，傳給營造廠，以便申請估驗計價，營造廠亦可利用此單回覆供應商申請結果。
退貨通知單	◎營造廠 to 供應商	營造廠驗退供應商物料時，發出此單已告知供應廠商物料遭受驗退並說明驗退原因。
領款通知單	◎營造廠 to 供應商	營造廠依正式計價單填寫此單，傳供應商以便通知供應商前來領款
物料驗收單	◎營造廠 to 供應商	物料進場時，營造廠依拒查驗結果填寫此單，傳給供應商，作為完成出料之證明
追加減明細表	◎營造廠 to 供應商	營造廠針對採購單彙整採購項目及數量修正之結果，傳給供應商備查

本研究之驗證機制所使用的 XML Schema 是參照此 12 項標準當做驗證依據；針對此 12 項 Schema 所制訂出的架構及內容，做一個文件檢核的動作，使文件在交換有個依據標準，也讓資訊在交換的過程中變的更有效率。

2.1.3 驗證示範

國內較具代表性之認證計畫為研考會推動之「文書及檔案管理電腦化作業規範驗證制度」及「公共工程 XML 資料交換標準」，由政府單位制訂文件格式標準，規定以 XML 標準為資料交換格式，再建立電子文件及軟體之認證，其認證之範圍如圖 2-2 所示。

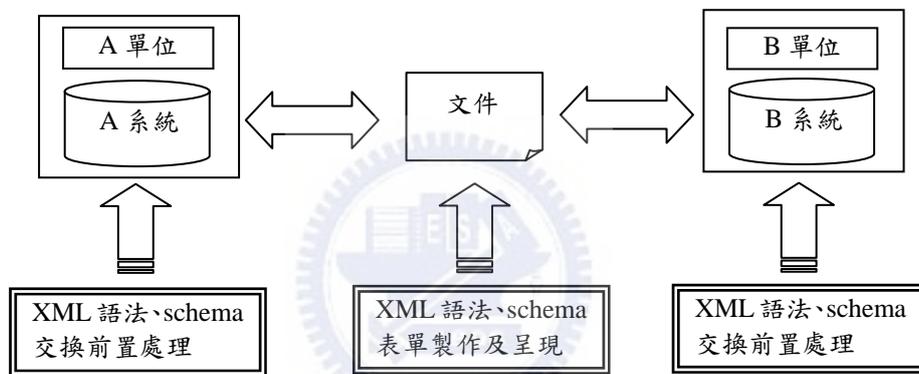


圖 2-2 標準驗證制度範圍示意
(梁樾、陳春盛、曾仁杰，民 94 年)

當 A 單位 B 單位經由本身的系統產生了一份欲交換之 XML 文件，準備進行資訊交流之前，必須要先使用驗證系統執行文件檢覈的動作，確認 XML 文件中的內容、語法及架構都符合營建產業 Schema 標準，才可以進行後續交換的動作。

本研究將針對 12 項 schema 標準草案進行資料交換標準應用軟體之驗證，並參考研考會及工程會之電子化文件認證制度，供廠商提出驗證申請，機構驗證並據以簽發驗證報告。

2.2 可延伸性標記語言(XML)

近年來網際網路盛行，其中又以 WEB 型態的網站居多；傳統的網頁是以

HTML 標示語言所撰寫而成的，再透過瀏覽器轉換成使用者看見的網頁。但隨著功能需求(例如：B2B)越來越大，單純的 HTML 已不能滿足現況，而且電腦也無法瞭解資料屬於哪一類型態及作何用途，因此 XML 也就隨之蘊育而生。

2.2.1 XML 簡介

XML 為 1986 年由國際標準組織(International Standard Organization, ISO)公佈的一個名為「標準通用標示語言」(Standard Generalize Markup Language, SGML)的精簡版。1998 年 2 月，美國 W3C(World wide Web Consortium, 全球資訊網協會)組織正式公佈 XML 的 Recommendation 1.0 版語法標準。XML 掌握了 SGML 其延展性、文件自我描述特性、以及其強大的文件結構化功能，但 XML 卻屏除了 SGML 過於龐大複雜以及不易普及化的缺點。字面上來看 XML 是一種標識語言，但嚴格來說它和 SGML 一樣是一種「元語言」(meta-language)。(Barillot, E. and Achard, F., 2000)換言之，XML 是一種用來定義其他語言的語法系統，這正是 XML 功能強大的主因。它可以促進各專業機構、不同產業界、學術界和特定應用領域發展各自標準的文件和訊息，以利資訊的交換、處理和相關衍生性資料加值服務。(Bryan, M., 2001)

XML 文件和訊息的主要特色在於它是結構以及資訊內容導向。結構化文件和訊息編碼方法的主要精神在於它可供其他電子資料傳遞、文件出版系統、電腦輔助設計或製造、資料庫管理等等。在處理重複和共享的資料時，能有效提升其效率和效能，節制資訊系統的開發建置和管理營運成本。(Miyazawa, T. et al., 2000)這種方法將資訊內容、結構和格式等不相同的文件要素予以區分。它保存了文件的資料和結構，可是卻不指出文件的呈現格式，如是格式的解析應在資料最後傳遞時，才依據用戶需求進行最佳化之處理。XML 技術本質上的優勢和特色，使商務資訊流電子化產生根本上的改變，並在應用上提供更多為的可能性。

將 XML、SGML、HTML 之間的差異及優缺點整理如表 2-2，可以更清楚的了解這三者之間用途及功能的不同。

表 2-2 SGML、HTML 和 XML 之間的差異比較(郭馨儀，民 89 年)

	SGML	HTML	XML
全名	Standard Generalized	Hyper Text Markup	eXtensible Markup

	Markup Language	Language	Language
中譯	標準通用標記語言	超文字標記語言	延伸標記語言
發展歷史	國際標準組織(ISO)於1986年公告的電子文件交換標準	源自SGML，1989年由歐洲核子物理研究中心的研究人員所創，為SGML的應用	源自SGML，1998年由W3C所發展
目的	規範文件製作及交換標準，與不同電腦系統交換	針對網頁頁面呈現方式	針對網頁內容，撰寫結構化文件語言
內容	<ol style="list-style-type: none"> 1. SGML宣告 2. DTD(Document Type Definition) 3. DI(Document Instance) 	<ol style="list-style-type: none"> 1. 文字 2. 圖形 3. 超連結 	<ol style="list-style-type: none"> 1. XML 2. XML Schema 3. XSL 4. XLL
特色	DTD(Document Type Definition)	網際網路上使用的超連結文件	內容與格式是分離的，不是綁死的
優點	<ol style="list-style-type: none"> 1. 各種資訊自行定義擴充標準 2. 維持資料格式一致 3. 可重複使用資源共享 4. 嚴謹正確的結構化文件 5. 不限電腦週邊語言的限制 6. 可隨所需格式輸出 	<ol style="list-style-type: none"> 1. 簡單易學 2. 支援各種方法製作HTML 3. HyperLink 4. 表單建立容易 5. 程式撰寫容易 6. 提供開放平台 	<ol style="list-style-type: none"> 1. 延伸SGML的優點 2. 改善HTML的缺點 3. 開發應用程式容易 4. 具延展性適用各個領域 5. 異質資料庫之間文件處理 6. 採用unicode
缺點	<ol style="list-style-type: none"> 1. 缺乏網頁瀏覽器的支援 2. 缺乏標準樣式支援 	<ol style="list-style-type: none"> 1. 結構障礙造成瀏覽障礙 2. 缺乏表達內容的支援 3. 格式固定限制資料交換 4. 缺乏一致性限制自動化 5. 搜尋不夠精確 6. 管理維護不易 	<ol style="list-style-type: none"> 1. 各界定義標籤不一致，難以通一執行 2. 各領域使用自行定義的標籤一產生相容性溝通的問題 3. 資料安全問題有議

1998年2月，XML建議標準提出，是目前通用的XML1.0版本標準，XML的規格便是由W3C XML工作群維護並負責增修根調整的工作。基本上SGML、HTML和XML不會因為誰的存在而使其他的標記語言消失。XML，主要運用在長期使用且須結構化的文件；HTML仍運用於快速出版的網頁，SGML仍無法被Internet接受，將繼續適用於複雜結構的運用。(郭馨儀，民89年)

表2-3為XML相關技術，包含了文件驗證及排版樣本等，而本研究只針對文件驗證的功能做探討。

表 2-3 XML 相關技術表(林錦雲，民 92 年)

XML 用途	相關技術	說明
文件驗證	DTD(Document Type Definition) XML Schema	檢查 XML 文件架構和標籤內容是否符合規則
顯示文件	CSS(Cascading Style Sheets) XSL/XSLT(eXtensible Stylesheet Language)	產生報表或依指定條件提取出 XML 文件的內容
資料交換	SOAP(Simple Object Access Protocol)	在商業應用上，XML 可以是分散式系統資料交換的
連結到其他資源	XPoint(XML Pointer Language) XLink(XML Linking Language)	支援連結到 XML、非 XML 文件，甚至是更複雜的連結

2.2.2 XML 文件架構

一份完整的 XML 文件內容必須包含標籤(Tag)以及文字(Text)，標籤的意義是為定義資料而成為一個元素，而文字則是代表標籤裡的內容。圖 2-3 為一份標準 XML 文件，並於其後以圖 2-3 說明文件中各個元件所代表的意義及內容寫法規則。(Phillips, L. A., 2001)

```
<?xml version = "1.0"?>
<!--booklist last update 2006-06-03-->
  <booklist>
    <book isbn = "72-80081-082">
      <title>The Gourmet Microwave</title>
      <publisher>New Moon</publisher>
      <price>9.95</price>
      <author>Bruce Lee</author>
      </other>
    </book>
  </booklist>
```

圖 2-3 正確 XML 文件

1. XML 宣告：

XML 宣告中定義文件所要遵循的版本，且此行必須以小寫的字母表示，同時也必須宣告所使用的編碼。此例中沒有宣告所使用的編碼，則表示用 UTF-8(英文)當編碼標準

2. 註解：

可隨意放置於文件中任何地方，並且須以下列之形式呈現：

```
<!--booklist last update 2006-06-03-->
```

3. 根元素：

XML 文件中最上層的元素，例如表 2-4 文件中的<booklist>即為根元素。一份文件中之根元素只能存在一個，如果超過即為不合法之文件。

4. 子元素：

一個元素底下可能會包含許多子元素，如表 2-4 之<book>即為<booklist>之子元素。

5. 空元素：

元素內沒有包含任何資料，如表 2-4 之<other>元素。

6. 屬性：

提供元素(element)有更多的參考資訊，加強描述元素，如表 2-4 之元素<book isbn = “72-80081-082”>，其中 isbn 即為 book 之屬性，屬性值為 72-80081-082。

上述六點即為 XML 文件包含的內容，除了註解及屬性之外，其他都必須出現在 XML 文件裏，才可稱為一份完整的 XML 文件；如要確定該 XML 文件是否合法，可以使用該份文件之 DTD(Document Type Defining)或是 XML Schema 執行驗證的動作。

2.2.3 DTD 與 XML Schema

文件需要驗證的理由有很多種，針對 B2B 或 B2C 電子商務而言，每份訂單都應該做正確性的檢查，來保障網站的安全。而驗證則可以帶來一個好處，將訂單規範為 XML 文件或是 XML 資料島時，接收端只需要簡單的幾道指令，剖析器就會執行訂單的基本格式確認，而不需要撰寫複雜的程式來做文件正確性的確認工作。驗證可以確認 XML 下列四點的正確性：(陳錦輝，民 90 年)

1. 確認元素(標籤)與屬性都是使用規範中的名稱以及內容。
2. 確認屬性是否可以屬於或應屬於某一個元素(標籤)。
3. 確認子元素(標籤)的順序性與個數是否合法。
4. 確認元素(標籤)與屬性的資料形態是否合法。(XML Schema 具有較多種的資料型態描述)。

為了保證 XML 文件符合某些事先定義的文法，需要先驗證 XML。首先，可以在文法內規範該類 XML 文件的結構，例如：某些元素「必須是」某一元素的子元素，或某些元素「可以是」另一元素的子元素。對於屬性而言，我們也可以規範哪些屬性是屬於哪些元素所有。對於資料型態方面，我們則可以規範元素與屬性的資料型態，甚至我們也可以在文法中規範哪些元素是空元素。

根據能力的不同，可以將目前所盛行驗證 Validated XML 的技術分為兩種：DTD 與 XML Schema。DTD 是最初用來驗證 XML 文件的技術，目的是定義 XML 文件中元素的架構、元素標籤以及屬性。由於 DTD 語法並沒有支援現代化物件導向的觀念，而且對於資料型態的資源過少，因此 W3C 開始著手開發替代 DTD 的技術。(Martin, D. et al., 2000)

XML Schema 原先為由微軟所提出的規格草案，歷經 1 年的審議過程，目前已經成為 W3C 的 Proposed Recommendation 標準。XML Schema 完全遵循了 XML 的基本語法(也就是說 XML Schema 也是一個 Well-Formed XML 文件)，不用像 DTD 要學習一套新的語法，而且 XML Schema 也改進了 DTD 的許多缺點，因此目前成為 XML 新一代的驗證技術。(Roy, J. and Ramanujan, A., 2000)

比較 DTD 與 XML Schema 彙整於表 2-4，就可以了解 XML Schema 之優勢。

表 2-4 DTD 與 XML Schema 比較表(林錦雲，民 92 年)

項目	DTD	XML Schema
與 XML 的關係	DTD 文件的撰寫方式與 XML 文件的架構不同，使用者必須多學另一種語法	XML Schema 本身就是一份 XML 文件，使用者並不用另外去學習另一種語法。同時，透過存取 XML 文件的程式介面，便可以修改或存取 XML Schema
支援的資料型態	只有支援 #PCDATA 單一資料型態	支援眾多的資料型態：布林、整數、浮點數、字串、日期等
文件元素	XML 文件只能套用單一的 DTD，同時 XML 文件中的元素應在 DTD 中完成定義	XML 文件內，可以使用 XML Schema 未定義的元素。且 XML Schema 可以指定不同 XML Schema 文件中的元素
文件架構	彈性太低	且支援名稱空間，有較佳的擴充性

2.3 XML Schema 介紹

2.3.1 XML Schema 特點

由上一節可知，XML Schema 比 DTD 更具有優越性，因此 XML Schema 已漸漸的取代以往的 DTD，變成目前制定各產業標準的主流。XML Schema 是 Microsoft 所提出的規格，目前已經成為 W3C 的正式規格；XML Schema 與 DTD 功能相同，都是在驗證 XML 文件，但是 XML Schema 比 DTD 更具有彈性，且規範的能力也比較強。(Roy, J. and Ramanujan, A., 2001)所以本研究所採用的營建產業資訊交換標準就是以 XML Schema 制定完成。以下就詳細介紹 XML Schema 之特色以及內容架構。

根據 XML Schema 相關資料，整理出 XML Schema 特點有以下八點：(陳錦輝，民 90 年)

1. XML Schema 即為一份 Well-Formed XML 文件，不需再重新學習別的語法。
2. XML Schema 對於資料型態的支援較多，包含字串、布林值、整點、浮點等等，而且未來新的物件所支援的資料形態會更多。
3. XML Schema 針對結構的定義，利用 minOccurs 以及 maxOccurs 來達到控制子元素的出現次數及目的，比 DTD 更有彈性。
4. XML Schema 使用名稱空間，因此具有可擴充性。
5. XML Schema 與 DTD 不同，所採用的是開放性架構，不要求所有的元素與屬性必須先宣告再使用。
6. XML Schema 可以針對個別元素指定不同的 XML Schema 作為驗證語法，最小的切割單位可以是一個標籤(元素)。
7. XML Schema 是一個 Well-Formed XML 文件，因此可以透過 DOM 介面存取細部元素與屬性，達到修改文法規則目的；除此之外，XML Schema Extension 還允許新增一些額外的資訊以輔助描述資料。

8. XML Schema 與 DTD 相同，只能驗證文件結構與資料型態，而不能驗證文件的語義。

2.3.2 XML Schema 內容架構

圖 2-4 為一份合法之 XML Schema，以此範例說明 XML Schema 內容元素：

```
01 <?xml version="1.0" encoding="UTF-8"?>
02 <!-- edited with XMLSPY v2004 rel. 4 U (http://www.xmlspy.com) by orion (ORiON) -->
03 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >
04   <xs:element name="RequestForQuotation">
05     <xs:annotation>
06       <xs:documentation>Root Element</xs:documentation>
07     </xs:annotation>
08     <xs:complexType>
09       <xs:sequence>
10         <xs:element name="SheetTitle" type="SheetTitle" minOccurs="0"/>
11         <xs:element name="InvitationDate" type="xs:date"/>
12         <xs:element name="QuotationDeadline" type="xs:date"/>
13         <xs:element name="InvitationVersion" type="xs:string" minOccurs="0"/>
14         <xs:element name="InvitedCom" minOccurs="0">
15           <xs:complexType>
16             <xs:sequence>
17               <xs:element name="ComName" type="xs:string" minOccurs="0"/>
18               <xs:element name="ComTaxNo" type="xs:string" minOccurs="0"/>
19             </xs:sequence>
20           </xs:complexType>
21         </xs:element>
22       </xs:sequence>
23     </xs:complexType>
24 </xs:element>
25 </xs:schema>
```

圖 2-4 合法之 XML Schema 文件

各元素標籤功能如下說明：

1. 首先要先宣告使用 XML 的版本以及編碼，如同文件中的第 01 行
2. XML Schema 與 XML 相同，可以於文件中使用註解，如同文件中第 02

行。

3. XML Schema 也為一份標準的 XML 文件，必須要有根標籤將文件內容涵蓋於裡面，而<xs:schema>與</xs:schema>即為唯一根標籤，如同文件中第 03 行到第 24 行。
4. 文件第 03 行中，有宣告所使用的 namespace，該 namespace 是指定 XML Schema 標示語言所使用的名稱空間，所以從第 03 行到第 24 行所使用的元素名稱都是來自於該 namespace，如 schema、annotation、element、sequence、complextypе。
5. 第 04 行到第 08 行是用來宣告根元素 RequestForQuotation，第 05 行到第 07 行為註解，不會對 XML 文件產生作用，只說明 RequestForQuotation 為此份文件之根元素；而第 09 行說明 XML 文件元素出現的順序必須與 XML Schema 所規定的順序相同。
6. 第 10 行到第 18 行為設定元素 SheetTitle、InvitationDate、QuotationDeadline、InvitationVersion、InvitedCom 以及 InvitedCom 底下之子元素 ComName、ComTaxNo 之名稱(name)、資料型態(type)及出現次數(minOccur，如沒設定即為必出現一次)。

2.4 文書及檔案管理電腦化作業規範

行政院秘書處於八十三年頒布「公文處理現代化推動方案」，同年，行政院研究發展考核委員會（以下簡稱行政院研考會）即與行政院秘書處、台灣省政府秘書處及台灣省政府研考會共同彙編「文書及檔案管理電腦化作業規範」，並函頒實施。為加速推動公文處理電子化、促使各級政府機關採用符合規範之文書及檔案管理電腦化作業相關軟體、便於機關間之公文電子交換及有助於委外作業系統驗收規格之建立，行政院研考會同時也頒布「文書及檔案管理電腦化作業規範驗證制度」以為配合。如今事隔五年，電腦與網路科技已有很大的進步，網際網路之應用也蔚為風氣，而行政院也在八十七年三月修正頒布「文書處理檔案管理手冊」，將公文格式大幅簡化。行政院研考會遂於八十八年再度修正「文書及檔案管理電腦化作業規範」及「文書及檔案管理電腦化作業規範驗證制度」以為因應。

電子公文產品的驗證服務替供應商、使用者及電子公文交換的相關單位創造三贏局面。供應商的電子公文產品通過驗證表示能產生符合規範的電子公文，使用者可放心採用，從而擴展商機。使用者採購電子公文產品時，要求供應商提供驗證合格證明即可，省去部份開立規格的麻煩。如果參與電子公文交換的產品都通過了驗證，則單位間電子公文的互通性自然形成。

文書及檔案管理電腦化作業規範(八十八年修訂版)頒布後，資訊業者可據以開發下列任何一類電子公文相關產品：

1. XML 語法剖析軟體
2. 電子公文製作及呈現軟體
3. 電子公文交換前置處理軟體
4. 電子公文稽催管制軟體
5. 電子公文檔案管理軟體
6. 以上各類軟體功能任意組合而成之軟體

本驗證制度之重心在確保電子公文交換時之互通性，故所有驗證皆以此為前提。稽催管制與檔案管理係機關單位內部作業，故此類軟體並不做為驗證對象。至於由若干軟體組合而成之電子公文產品，則僅就其與 XML 語法剖析、電子公文製作、呈現及交換前置處理相關之功能驗證，產品之執行效能 (performance) 及各項功能間之整合程度不在驗證範圍之內。本驗證將著重驗證基本核心功能(如 XML 語法剖析、製作、呈現、交換)，此與機關採購產品之驗收不同(後者涉及業務面如收文、發文流程是否符合所定需求)。

2.5 小結

本研究主要是參考行政院發展考核委員會所彙編的「文書及檔案管理電腦化作業規範驗證制度」來擬定驗證機制流程；再以 12 項之營建物料採購資訊交換 Schema 標準當作依據，配合著 XML 相關技術來建置驗證系統，探討整套驗證流程的可行性。

第3章 驗證機制功能與流程分析

本研究是統計營建業常採用 12 種電子化表單內容交換時可能會遇到的問題以及電子採購系統的功能分析之後，進而規劃出完整的驗證流程。內容包含由申請驗證開始，最後到將驗證結果儲存於資料庫中的一切程序，透過本研究建置的驗證系統，使廠商之電子採購系統所產出的 XML 電子採購文件能符合營建產業 Schema 標準。驗證流程規劃主要分為四個階段，分別為：前置作業階段、格式檢驗階段、系統驗證階段、簽證及儲存階段，各個階段都有其主要功能，而且必須完成前一階段的事項才能繼續下一階段的工作。

完整之驗證流程如圖 3-1 所示，而圖 3-2 則是將圖 3-1 之流程圖分成受驗單位及驗證執行單位兩個個體，分別清楚的呈現出雙方單位於各個驗證階段所要執行的工作項目。

在受驗單位部份，申請驗證並填表繳費完成後，可以於指定地點安裝受驗系統。當驗證開始時，由受驗系統產出受驗文件，並由驗證執行單位驗證文件內容架構；當文件內容驗證完畢後，受驗系統須接收驗證執行單位所產出的測試文件，讀取此文件後並將其呈現成階層性的表格樣式。當整個驗證程序完畢後，受驗單位會收取到一份驗證報告書，受驗單位必須要留存供日後查驗之用。

在驗證執行單位部分，先行審核申請驗證單位的資格，於審核通過後安排驗證的時間與地點以執行驗證的工作。當驗證開始時，驗證執行單位接受到受驗單位的受驗文件之後，先行驗證該文件是否符合 XML 文件規則，再行比對文件之內容是否與 Schema 標準規範的相同。當文件內容驗證完畢後，由執行驗證單位產出測試文件，交由受驗單位做讀取呈現的動作。當整個驗證程序完畢後，由驗證執行單位記錄並統計驗證過程中所產生的錯誤種類及原因，並將其撰寫於驗證報告書中，最後由雙方簽核並留存以供日後查核。

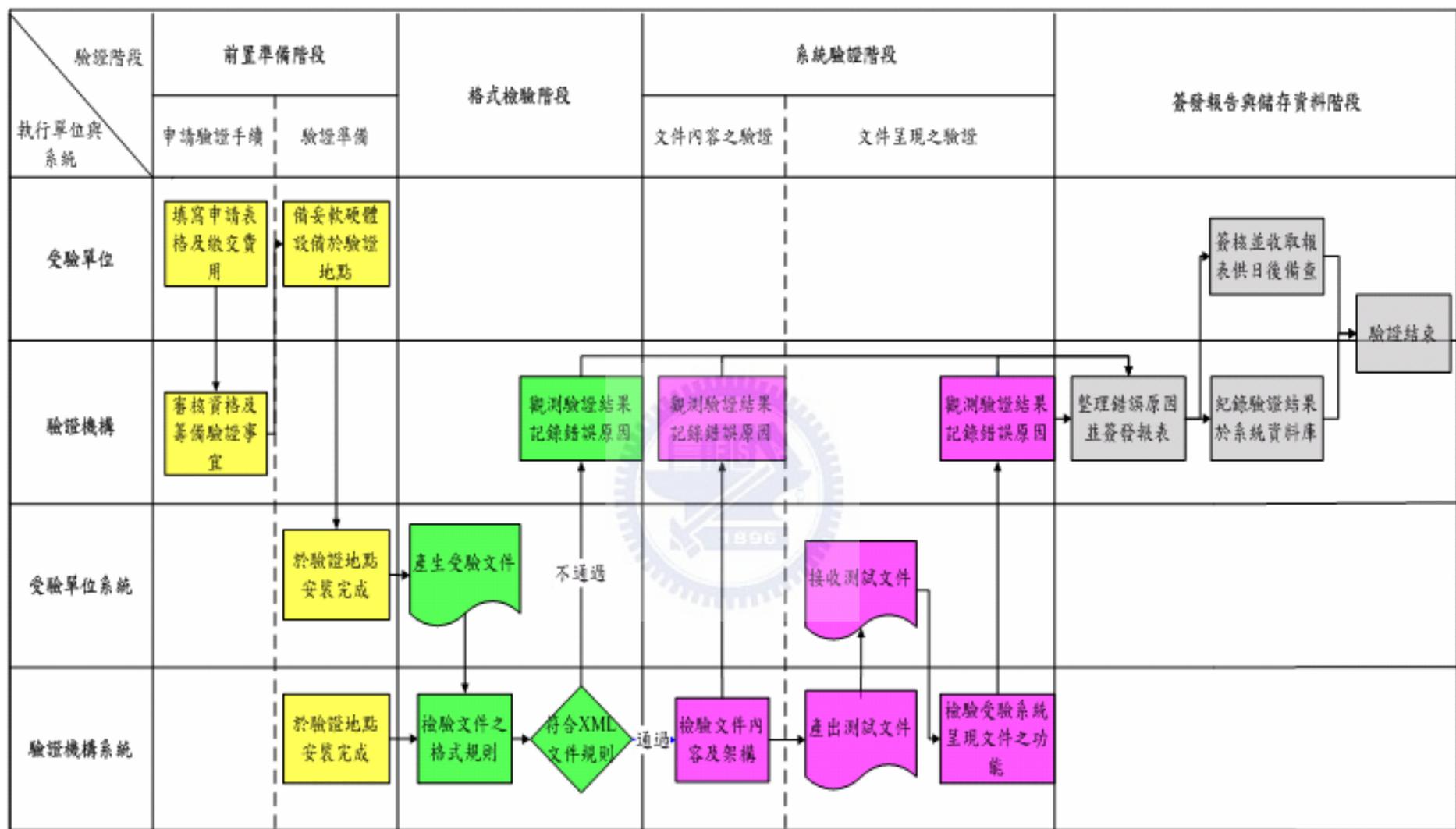


圖 3-1 驗證機制流程圖

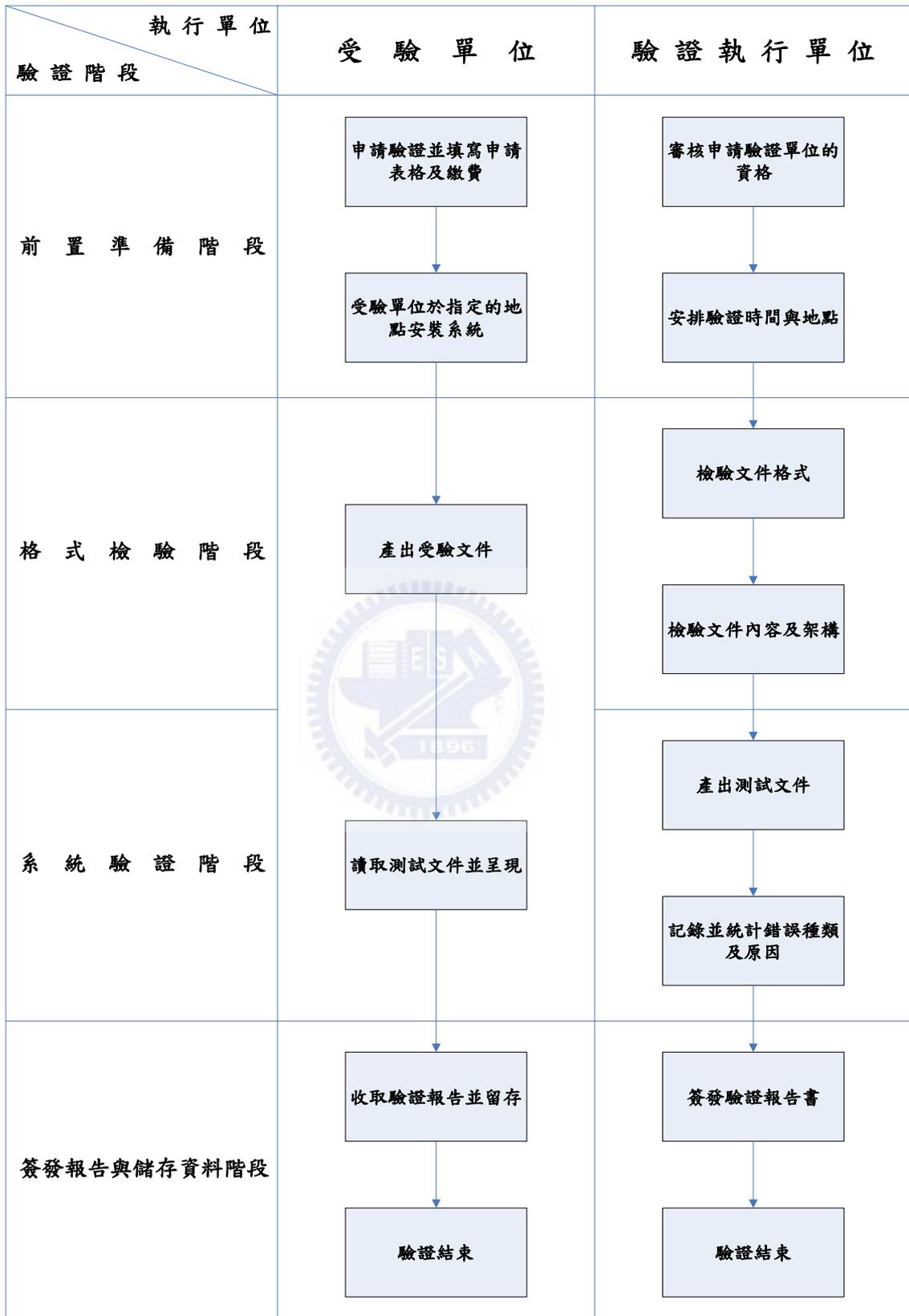


圖 3-2 受驗單位及驗證執行單位於各驗證階段執行之工作項目

3.1 前置準備階段

驗證程序應把握公平、公正、公開的原則，所有的驗證動作(系統驗證、簽發報表等等)皆應於受驗單位代表與驗證執行單位代表雙方都在場的情況下進行，避免驗證結果之爭議。但是在執行驗證的動作前，受驗單位必須先完成申請驗證的動作(繳交文件、費用、安排驗證時間及地點)，才得以進行後續的動作。所以前置準備階段主要目的為完成驗證之前的必要手續以及安裝受驗系統。

3.1.1 申請驗證

申請驗證手續主要是由受驗單位填寫申請文件及繳交驗證費用之後，再由驗證執行單位審核並安排驗證的時間與地點，詳細的操作手續如下列兩點：

1. 填寫申請表格並繳交費用

- (1) 先與驗證機構領取驗證申請書(詳見附錄一)，並填寫完畢。
- (2) 驗證申請書之內容須包含申請驗證單位、申請單位代表、聯絡電話、聯絡地址、受驗系統名稱以及所驗證的表單種類等等。
- (3) 申請驗證單位將申請文件直接送繳至驗證執行單位，並繳交驗證申請費，驗證執行單位開立發票給受驗單位，而費用由驗證機構制定。

2. 審核資格與籌備驗證事宜

- (1) 驗證執行單位接收到受驗單位的申請文件後，於三天內審查完畢，如符合規定事項，則通知受驗單位前來驗證機構安排驗證事宜。
- (2) 如果審查不通過，通知受驗單位於五天內補正文件，逾期則以放棄論。
- (3) 受驗單位應於指定日期前往驗證機構安排驗證時間、地點以及受驗單位所需之軟硬體設備。如受驗單位無法於約定日期前來，則須先與驗證單位通知，並可只以一次為限，否則須重新提出申請。
- (4) 安排完成驗證時間與地點後，受驗單位繳交驗證執行費用給驗證執行單位，並由驗證單位開立發票給受驗單位。

(5) 費用說明：

- 驗證執行費用採按項計費。
- 通過驗證的文件，如版本不同(如增刪元素等)或因協會之 Schema1.0 版標準修改而再次申請驗證者，則驗證執行單位得斟酌減少驗證費用。

3.1.2 驗證準備

當申請的手續完成後，就該進行驗證準備的動作，而驗證準備的項目有：

1. 受驗證單位應於約定時間至指定地點安裝所需的軟硬體設備以及系統。
2. 指定地點原則是以驗證機構指定為主。
3. 驗證場所如果是在驗證機構裡，受驗單位應於驗證日期前到驗證機構將系統以及軟硬體設施安裝妥當。
4. 受驗單位應於指定日期前將系統安裝妥當，安裝期間受驗單位可以修改系統，安裝完成得與驗證機構雙方同意下加以封裝，交給驗證單位存證。

3.2 格式驗證階段

此階段可以算是執行驗證手續的先行作業，主要的工作是先檢驗受驗單位的系統所產出的電子採購文件是否合乎 XML 文件的規則，也就是文件必須為格式正確 (Well-Formed)，否則就不是合法的 XML 文件，無法進行後續的驗證動作。

此階段的運作程序為受驗單位匯出欲驗證的文件至驗證機構的驗證系統中，由驗證系統檢驗文件是否符合 XML 文件之規則，驗證成功後再行進行後續動作，如果失敗則簽發驗證結果報表後結束驗證。

符合 Well-Formed 的文件如圖 3-3，並說明 XML 文件標準格式規則之內容為何。

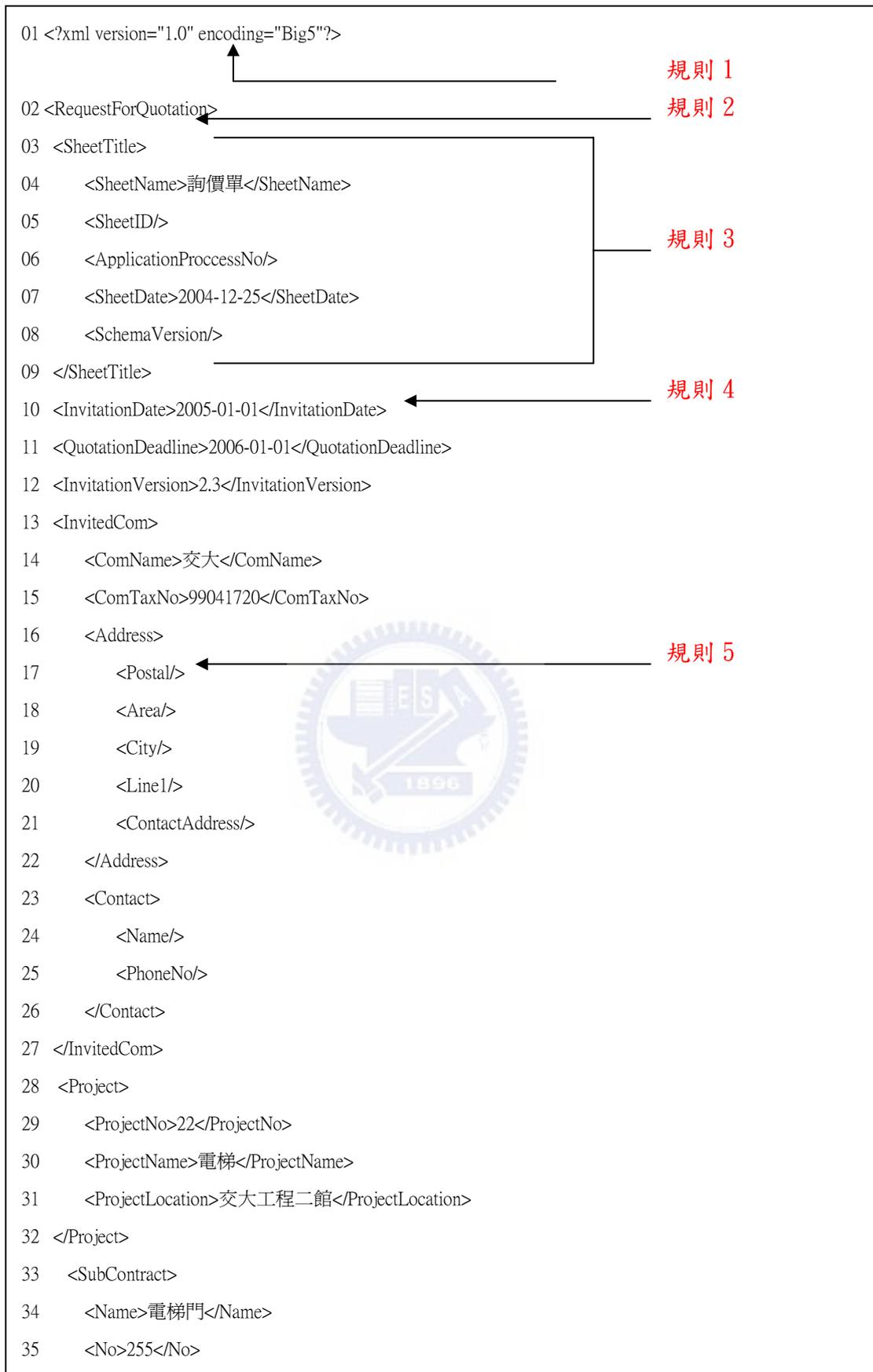


圖 3-3 格式正確的 XML 文件(符合詢價單 Schema1.0 版)

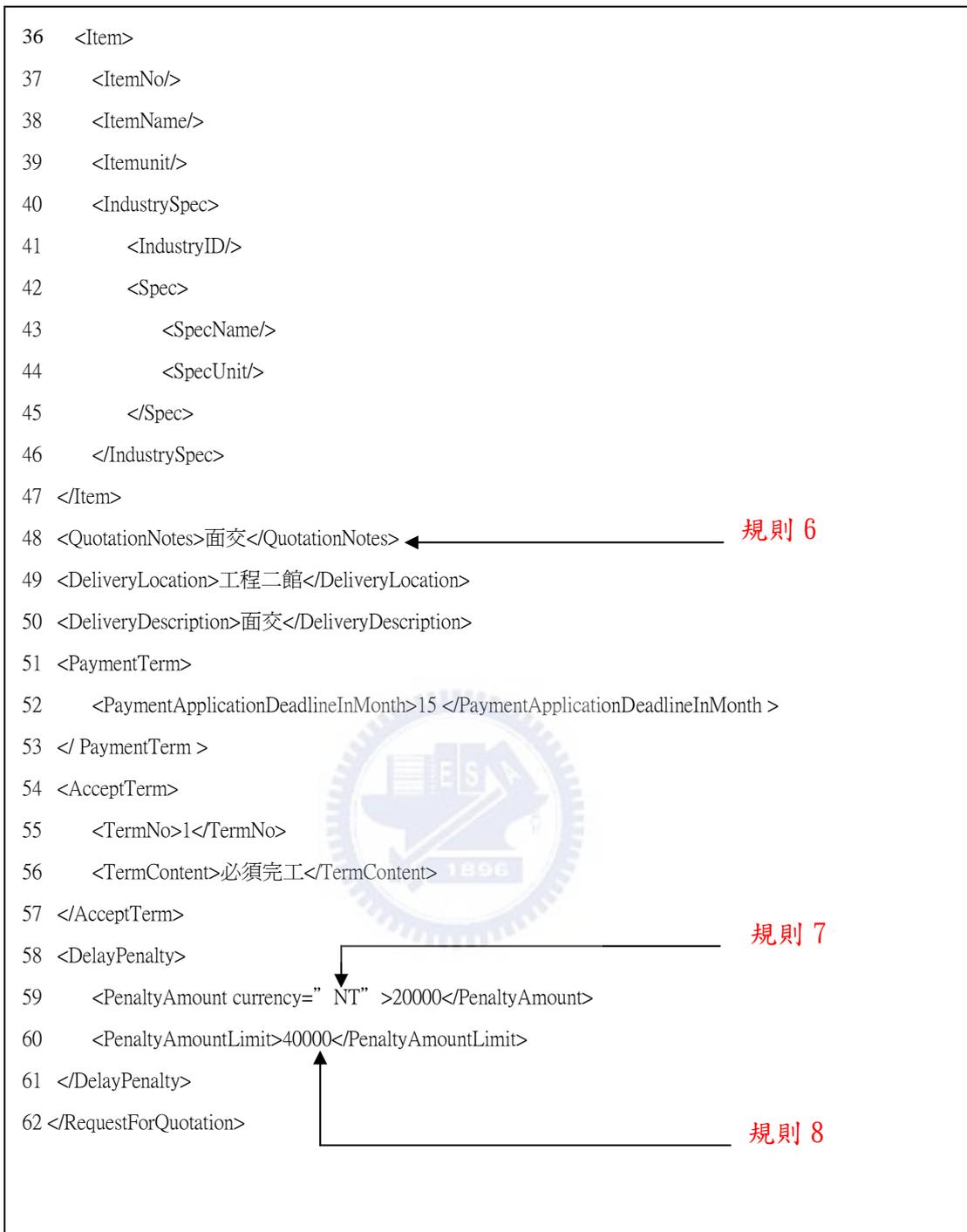


圖 3-3 續 格式正確的 XML 文件(符合詢價單 Schema1.0 版)

一份合法之 XML 文件需符合下列八項規則：

規則一：XML 文件的宣告得以小寫之「xml」當開頭，且必須設定 version 的屬性值，如第 01 行。

規則二：必須要有一個根節點，而且也只能擁有一個根節點，如第 02 行，根元素為<RequestForQuotation>。

規則三：所有的標籤元素必須要以樹狀的方式排列呈現，如第 03~08 行。

規則四：除了空標籤之外，所有的標籤都必須成對的出現，也就是說必須要有開始標籤與結束標籤，如第 10 行。

規則五：空標籤的標籤結尾必須含有「/」，如第 17 行。

規則六：標籤的名稱以及屬性值都必須合於規則，而且大小寫不能混淆，否則視為不同標籤，如第 48 行，不能寫成<QuotationNotes>面交</quotationNotes>。

規則七：指定的屬性值前後都必須備「”」或者是「'」符號包圍住，如第 59 行，NT 為元素<PenaltyAmount>之屬性值。

規則八：特殊字元必須依照規定撰寫，如第 60 行。（“<”與”&”不得出現在內文）

3.3 系統驗證階段

此階段為本驗證機制最關鍵的部份，主要分為二部分進行驗證：第一部分為驗證系統是否有產生符合 Schema 文件的能力，即驗證電子採購文件之內容是否符合 Schema1.0 版的標準，也是主要驗證的部分；第二部分為檢驗系統是否能夠讀取符合 Schema 的文件，並將文件內的資料作階層性的呈現。以下就詳細說明此兩部分的驗證內容。

3.3.1 文件內容之驗證

當系統執行驗證 XML 文件的動作時，主要是針對元素的類型(必要的元素、非必要的元素、未定義的元素)以及資料內容進行分析，而每種類型都有正確的規則，如不符合規則即會產生各種形式的錯誤，而此三條規則分別為：

1. 必要的元素(即為 Schema 中規定最少出現的次數為 1 者)一定要出現，且出現的英文名稱一定要與 Schema 中規定的相同，資料型態與出現次數也必須符合 Schema 標準的要求。
2. 非必要元素(即為 Schema 中規定最少出現的次數可為 0 者)出現時，其英文名稱必須要與 Schema 的規定相同，內容的資料型態以及出現次數也必須符合 Schema 標準的要求。
3. 當未定義的元素出現時，對其英文名稱並沒有強制的要求，但是元素內容的資

料型態以及出現次數也必須符合常理。例如文件中之<SheetDate>或<InvitationDate>等有關於「日期」(Date)的元素，最多應只能出現一次，如出現一次以上則不合理。

此三項原則為系統驗證的參考依據，根據上面的三條規則，我們可以把錯誤類型大致分為四大類：必要元素未出現、元素英文名稱錯誤、元素內容資料型態錯誤以及元素出現次數錯誤。再依據元素的屬性，更細分成九種錯誤型態：在必要元素的部份，會出現四種型態的錯誤：缺乏必要元素、必要元素英文名稱錯誤、必要元素內容資料型態錯誤以及必要元素出現次數錯誤。非必要元素部份，會有三種型態的錯誤：非必要元素英文名稱錯誤、非必要元素內容資料型態錯誤以及非必要元素出現次數錯誤。至於非定義元素則會有兩種型態的錯誤：未定義元素內容資料型態錯誤以及未定義元素出現次數錯誤。雖然某些錯誤的原因相同，但因為是元素屬性的不同而有所分別。

在理想的狀況下，驗證程式可以將這九種的錯誤型態驗證出來，但是礙於現實的結果，有些錯誤型態是本研究系統難以達成的。把錯誤類型整理如表 3-1，有關於難以達成理想狀況的原因，放置於其後探討。

表 3-1 錯誤類型列表

錯誤類型	錯誤名稱	元素型態	錯誤說明
必要元素未出現	第一型 缺乏必要元素	必要元素	Schema 中所要求的必要元素未在文件表單中
元素英文名稱錯誤	第二型 必要元素英文名稱錯誤	必要元素	文件中出現的元素英文名稱寫法與 Schema 規定的不符
	第三型 非必要元素英文名稱錯誤	非必要元素	

元素內容資料型態錯誤	第四型 必要元素內容資料型態錯誤	必要元素	元素的內容資料型態與 Schema 所規定的要求不符合
	第五型 非必要元素內容資料型態錯誤	非必要元素	
	第六型 未定義元素內容資料型態錯誤	未定義元素	
元素出現次數錯誤	第七型 必要元素出現次數錯誤	必要元素	元素出現的次數不在 Schema 規定的範圍內
	第八型 非必要元素出現次數錯誤	非必要元素	
	第九型 未定義元素出現次數錯誤	未定義元素	

針對各型的錯誤分別舉例，並說明產生錯誤的原因於其後。圖例的呈現方式皆為上部份是 XML Schema 部分文件，下部份為 XML 部分文件，經由系統比對之後產生各型錯誤。

實例 1：必要元素未出現

此型錯誤只會出現在必要元素中，當文件中未含有 Schema 規定最小出現次數為 1 的元素，即會產生此型錯誤。以詢價單為例，系統所產出的文件中如少了詢價日期此欄訊息，系統則會判別此為必要元素未出現的錯誤。

在 XML Schema 標準文件(A)中，定義一必要元素為「QuotationDeadline」，受驗文件(B)中卻沒有出現此元素。經由系統比對之後，發現文件中缺少了「QuotationDeadline」元素，造成錯誤。如圖 3-4 所示：

Schema 標準文件(A)

```
.....  
<xs:element name="InvitationDate" type="xs:date"/>  
<xs:element name="QuotationDeadline" type="xs:date"/>  
<xs:element name="InvitationVersion" type="xs:string"  
minOccurs="0"/>  
<xs:element name="InvitedCom" minOccurs="0"/>  
.....
```

受驗文件(B)

```
.....  
<InvitationDate>2005-01-01</InvitationDate>  
缺少必要元素 QuotationDeadline  
<InvitationVersion>2.3</InvitationVersion>  
<InvitedCom>  
.....  
.....
```

比對

驗證錯誤

圖 3-4 必要元素未出現錯誤範例展示(以詢價單為例)

實例二：必要元素英文名稱錯誤

當文件中元素名稱並非與 Schema 所制定的名稱相同，但兩元素的文字意義是相似的，即歸類為此類錯誤。系統在抓取文件資料時，主要的根據是依照元素名稱而擷取元素資訊，不正確的元素名稱會造成系統無法執行動作，所以要驗證文件元素名稱以避免發生此種狀況。

在 XML Schema 文件(C)中定義一必要元素為「InvitationDate」，但在 XML 受驗文件(D)卻出現一元素為「InvitedDate」。經由比對之後，發現受驗文件中元素「InvitedDate」與 Schema 文件中所定義元素「InvitationDate」雖然名稱不同，但其字義是相同，所以歸類此型錯誤。如圖 3-5 所示：

Schema 標準文件(C)

```
.....  
<xs:element name="InvitationDate" type="xs:date"/>  
<xs:element name="QuotationDeadline" type="xs:date"/>  
<xs:element name="InvitationVersion" type="xs:string"  
minOccurs="0"/>  
<xs:element name="InvitedCom" minOccurs="0"/>  
.....
```

受驗文件(D)

```
.....  
<InvitedDate>2005-01-01</InvitedDate>  
<QuotationDeadline>2006-01-01</QuotationDeadline>  
<InvitationVersion>2.3</InvitationVersion>  
<InvitedCom>  
.....  
.....
```

比對

驗證錯誤

圖 3-5 必要元素英文名稱錯誤範例展示(以詢價單為例)

實例三：非必要元素名稱錯誤

在 XML Schema 標準文件(E)中，定義一非必要元素「InvitationVersion」，但在 XML 受驗文件(F)中卻出現一元素「InvitedVersion」。經由比對之後，發現受驗文件中元素「InvitedVersion」與 Schema 標準文件中所定義元素「InvitationVersion」雖然名稱不同，但其字義是相同，所以歸類此型錯誤。如圖 3-6 所示：

Schema 標準文件(E)

```
.....  
<xs:element name="InvitationDate" type="xs:date"/>  
<xs:element name="QuotationDeadline" type="xs:date"/>  
<xs:element name="InvitationVersion" type="xs:string"  
minOccurs="0"/>  
<xs:element name="InvitedCom" minOccurs="0"/>  
.....
```

受驗文件(F)

```
.....  
<InvitationDate>2005-01-01</InvitationDate>  
<QuotationDeadline>2006-01-01</QuotationDeadline>  
<InvitedVersion>2.3</InvitedVersion>  
<InvitedCom>  
.....  
.....
```

比對

驗證錯誤

圖 3-6 非必要元素英文名稱錯誤範例展示(以詢價單為例)

實例四：必要元素資料型態錯誤

雖然資料表示的形式有很多種，但是元素內容的資料型態在 Schema 標準文件中都已有規定，必須遵守；如與 Schema 規定不相同者，不能通過驗證。

在 XML Schema 標準文件(G)中定義一必要元素「QuotationDeadline」，其元素內容資料型態定義為「Date」，應以「2006-01-01」之格式表示；但在 XML 受驗文件(H)中，元素「QuotationDeadline」之資料型態卻是以 2006/01/01 表示。經由系統比對之後，發現受驗文件中的元素「QuotationDeadline」資料型態非以「Date」形態表示，造成驗證錯誤。如圖 3-7 所示。

Schema 標準文件(G)

```
.....  
<xs:element name="InvitationDate" type="xs:date"/>  
<xs:element name="QuotationDeadline" type="xs:date"/>  
<xs:element name="InvitationVersion" type="xs:string"  
minOccurs="0"/>  
<xs:element name="InvitedCom" minOccurs="0"/>  
.....
```

受驗文件(H)

```
.....  
<InvitationDate>2005-01-01</InvitationDate>  
<QuotationDeadline>2006/01/01</QuotationDeadline>  
<InvitationVersion>2.3</InvitationVersion>  
<InvitedCom>  
.....  
.....
```

比對

驗證錯誤

圖 3-7 必要元素資料型態錯誤範例展示(以詢價單為例)

實例五：非必要元素資料型態錯誤

在 Schema 標準文件 (I) 中，定義一非必要元素「PaymentApplicationDeadlineInMonth」，其資料型態為「positiveInteger」，應以「15」的方式表示；但在 XML 受驗文件(J)中，其元素之資料型態卻以「十五」表示。經由系統比對之後，發現受驗文件中的元素「PaymentApplicationDeadlineInMonth」資料型態非以「positiveInteger」形態表示，造成驗證錯誤。如圖 3-8 所示。

Schema 標準文件(I)

```
.....  
<xs:element name="Retention" type="Payment" minOccurs="0"/>  
<xs:element name="DownPayment" type="Payment"  
minOccurs="0"/>  
<xs:element name="PaymentApplicationDeadlineInMonth"  
type="xs:positiveInteger" minOccurs="0"/>  
<xs:element name="EndPayment" type="Payment"  
minOccurs="0"/>  
.....  
.....
```

受驗文件(J)

```
.....  
<PaymentTerm>  
  <PaymentApplicationDeadlineInMonth>十五  
  </PaymentApplicationDeadlineInMonth >  
</ PaymentTerm >  
.....  
.....
```

比對

驗證錯誤

圖 3-8 非必要元素資料型態錯誤範例展示(以詢價單為例)

實例六：未定義元素資料型態錯誤

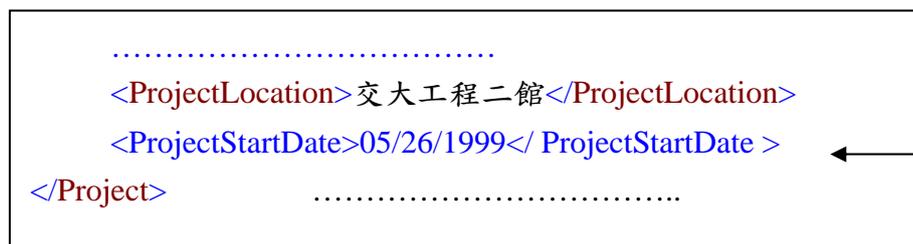
在此型的錯誤中，雖然對未定義的元素資料型態無規定，但是表現的方式須符合常態，而且如果之前有出現過類似的資料，就必須根據之前所表示的型態來呈現，不可有前後不一的情形。舉例來說，假設廠商欲交換的詢價單中多了一元素為「ProjectStartDate」(開工日期)，雖然 Schema 未對「ProjectStartDate」有詳細定義，但是根據之前之元素「InvitationDate」(詢價日期)，其資料型態為「Date」，因此「ProjectStartDate」的資料型態也要為「Date」之型態，如不符合則不通過驗證。

在 XML Schema 標準文件(K)中未定義元素「ProjectStartDate」，但在 XML 受驗文件(L)中卻有出現此元素，且資料型態以「05/26/1999」方式出現。經由比對之後，發現該元素之資料型態也應以「Date」型態出現，因而造成驗證錯誤，如圖 3-9 所示。

Schema 標準文件(K)



受驗文件(L)



比對

驗證錯誤

圖 3-9 未定義元素資料型態錯誤範例展示(以詢價單為例)

實例七：必要元素出現次數錯誤

在 XML Schema 標準文件(M)中，定義一必要元素「InvitationDate」，其出現次數只能出現一次；但在 XML 受驗文件(N)中，元素「InvitationDate」卻出現了兩次。經由系統比對之後，發現受驗文件中元素「InvitationDate」出現次數與 Schema 標準文件所定義的不符，因而造成驗證錯誤，如圖 3-10 所示。

Schema 標準文件(M)

```
.....  
<xs:element name="InvitationDate" type="xs:date"/>  
<xs:element name="QuotationDeadline" type="xs:date"/>  
<xs:element name="InvitationVersion" type="xs:string"  
minOccurs="0"/>  
<xs:element name="InvitedCom" minOccurs="0"/>  
.....
```

受驗文件(N)

```
.....  
<InvitationDate>2005-01-01</InvitationDate>  
<InvitationDate>2005-02-03</InvitationDate>  
<QuotationDeadline>2006-01-01</QuotationDeadline>  
<InvitationVersion>2.3</InvitationVersion>  
<InvitedCom>  
.....
```

比對

驗證錯誤

圖 3-10 必要元素出現次數錯誤範例展示(以詢價單為例)

實例八：非必要元素出現次數錯誤

在 XML Schema 標準文件(O)中，定義一非必要元素「SheetDate」，其出現次數只能出現一次；但在 XML 受驗文件(P)中，元素「SheetDate」卻出現了兩次。經由系統比對之後，發現受驗文件中元素「SheetDate」出現次數與 Schema 標準文件所定義的不符，造成錯誤，如圖 3-11 所示。

Schema 標準文件(O)

```
.....  
<xs:element name="SheetName" type="xs:string"  
minOccurs="0"/>  
<xs:element name="SheetID" type="xs:string" minOccurs="0"/>  
<xs:element name="ApplicationProccesNo" type="xs:string"  
minOccurs="0"/>  
<xs:element name="SheetDate" type="xs:date" minOccurs="0"/>  
<xs:element name="SchemaVersion" type="xs:string"  
minOccurs="0"/>
```

受驗文件(P)

```
.....  
<SheetTitle>  
  <SheetName>詢價單</SheetName>  
  <SheetID/>  
  <ApplicationProccesNo/>  
  <SheetDate>2004-12-25</SheetDate>  
  <SheetDate>2004-12-30</SheetDate>  
  <SchemaVersion/>
```

驗證錯誤

比對

圖 3-11 非必要元素出現次數錯誤範例展示(以詢價單為例)

實例九：未定義元素出現次數錯誤

對於此類型之錯誤，雖然 Schema 標準文件中未對元素有詳細定義，但是其出現次數必須按照常規。舉例來說，如 XML 受驗文件(R)中出現元素「ProjectStartDate」，儘管 Schema 標準文件(Q)未對元素「ProjectStartDate」定義，但是如果有此元素資訊出現的話，一般而言最多次數只能出現一次，如出現一次以上即為錯誤，如圖 3-12 所示。

Schema 標準文件(Q)

```
.....  
<xs:element name="ProjectName" type="xs:string"/>  
<xs:element name="ProjectLocation" type="xs:string"  
  minOccurs="0"/>  
  未定義元素「ProjectStartDate」  
</xs:sequence>  
</xs:complexType>  
.....
```

受驗文件(R)

```
.....  
<ProjectName>電梯</ProjectName>  
<ProjectLocation>交大工程二館</ProjectLocation>  
<ProjectStartDate>05/26/1999</ ProjectStartDate >  
<ProjectStartDate>06/26/1999</ ProjectStartDate >  
</Project>  
.....
```

驗證錯誤

比對

圖 3-12 未定義元素出現次數錯誤範例展示(以詢價單為例)

以上所舉例說明的九種錯誤類型，是為本研究理想的成果，也就是期望驗證系統能達到的功能。但是實際上，要達成這樣的目標是有相當的困難度，因為所牽涉到的學術領域與本研究的性質差異性太大，因此有些錯誤類型之驗證無法在本研究達成。不能直接由系統驗證的錯誤類型主要有四種類型：必要元素英文名稱錯誤、非必要元素英文名稱錯誤、未定義元素資料型態錯誤以及未定義元素出現次數的錯誤；至於難以使用系統直接驗證的原因，主要有兩點：

1. 歸類定義問題

如何才算是必要元素未出現？如何才算式元素英文名稱錯誤？在定義這兩個問題，本身就很難界定。舉例來說，如果 Schema 元素中有一必要元素為「Send Address」，而在 XML 文件中則出現「Delivery Location」，雖然英文名稱大不相同，但是中文名稱都可以譯為「送貨地點」，那麼系統會將「Delivery Location」當做「Send Address」的錯誤名稱，亦或是將「Delivery Location」視為新元素而將元素「Send Address」判斷為未出現，對系統而言是很難做區分的；如以人為判斷是很容易可以達成，但以系統來做判別則是需要花很大的功夫。因此，可以為這兩種形式做個定義：如果意義相同的兩種不同字串，則將視為名稱不同；如果兩種詞彙意義不同，則歸類為不同元素，也就是會產生必要元素未出現的錯誤。如此一來，用人為的判斷就可以很清楚的界定兩種錯誤的型態，但如要用系統作判斷，則必須配合著人工智慧的相關技術才可以鑑定出來。

2. 缺少驗證準則

當 XML 文件出現 XML Schema 未定義的元素時，因為系統沒有可以參考的驗證標準，最多只可以呈現哪些元素是 XML Schema 中未詳細定義，但是無法針對未定義元素的元素英文名稱、資料型態以及出現次數做出正確或錯誤的判別。

基於以上的理由，本系統無法直接對於必要元素英文名稱錯誤、非必要元素英文名稱錯誤、未定義元素資料型態錯誤以及未定義元素出現次數錯誤四種錯誤做驗證的動作。因此採用的方式為系統檢驗以及人為判斷的動作整合，共同完成驗證的手續。

需要使用人為判別驗證的錯誤，主要為必要元素英文名稱錯誤、非必要元素英文名稱錯誤、未定義元素資料型態錯誤以及未定義元素出現次數錯誤，佔了九種錯誤裡面的四種，人工介入比例大約為 45%。人為判斷步驟為圖 3-13，並說明如後：

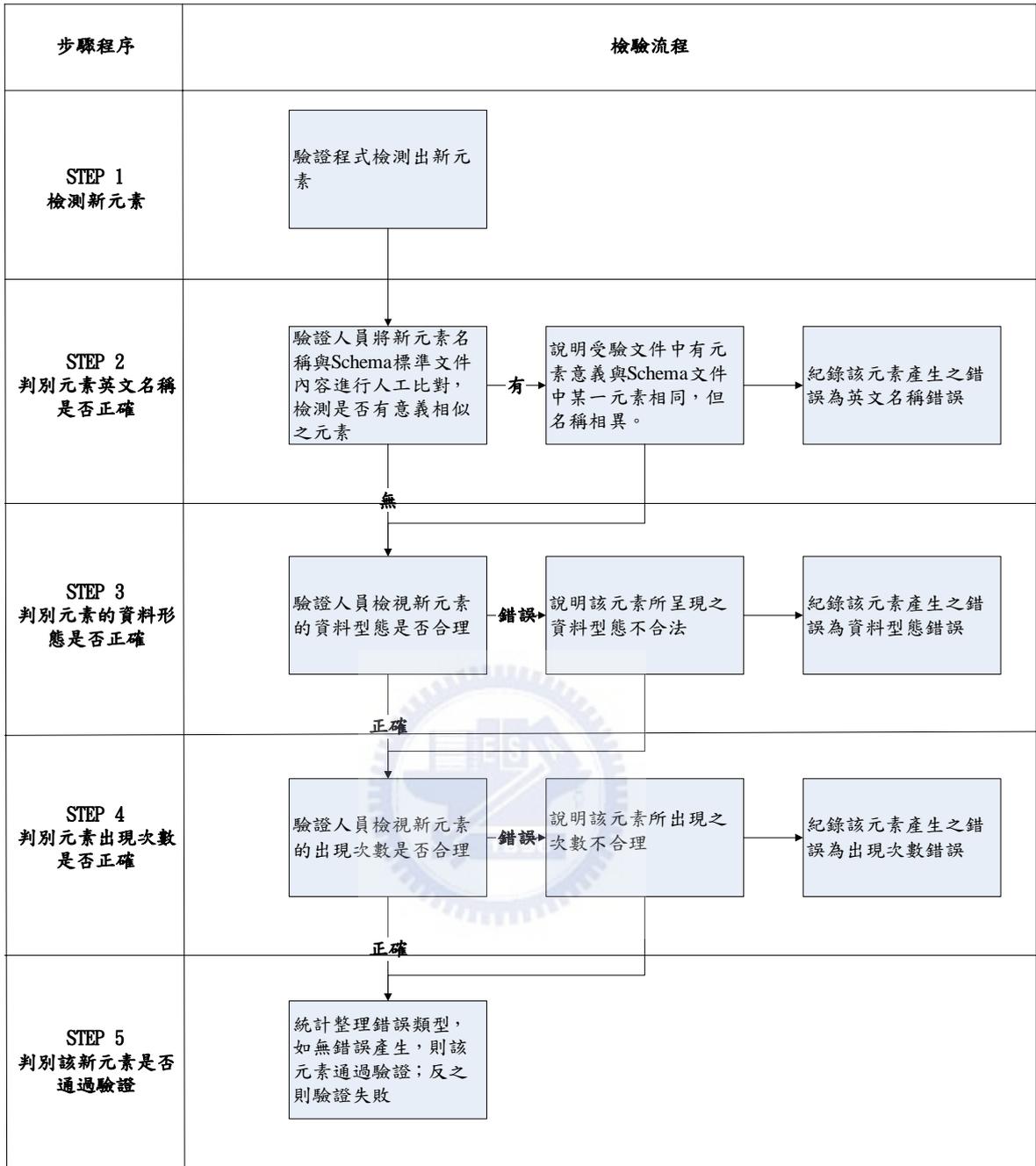


圖 3-13 人為判斷步驟圖

STEP 1：檢測新元素

當受驗文件中出現了 Schema 標準文件未定義的元素名稱，驗證程式會將該元素名稱以新元素的形式檢測出來，之後在進行後續的動作。

STEP 2：判別元素英文名稱是否正確

當該元素檢測出來後，驗證人員先行對照 Schema 標準文件中是否有與該元素

字義相同但名稱不同的元素。如果對照過後發現有這樣的元素名稱，則可以說明所檢測出來的新元素應為「元素英文名稱錯誤」，並且記錄該錯誤原因；如果於 Schema 標準文件沒有發現類似之元素，則是通過了「元素英文名稱」之驗證。

例如檢測出一元素名為「Delivery Location」，經由與詢價單 Schema 標準文件對照之後，發現有一元素「Send Address」與其意思相同，但名稱不同，因此將其判斷為「元素英文名稱錯誤」。

如系統檢測出一元素為「ProjectStartDate」，與詢價單 Schema 標準文件對照過後，並沒有發現與該元素意思相同的元素，說明了元素「ProjectStartDate」為一新元素，並沒有「元素英文名稱錯誤」的情況產生。

STEP 3：判別元素的資料形態是否正確

當「元素英文名稱」項目驗證完畢後，如有產生「元素英文名稱錯誤」的元素，則該元素的資料型態應以 Schema 標準文件中正確元素名稱的定義為主；如果沒有產生「元素英文名稱錯誤」的元素，代表則為新元素，其資料型態應以之前所定義的類似元素資料型態表示。

舉例來說，在 XML Schema 標準文件中定義一必要元素為「InvitationDate」，但在 XML 受驗文件卻是出現元素名為「InvitedDate」，說明了元素「InvitedDate」為「InvitationDate」的「元素英文名稱錯誤」，因此元素「InvitedDate」之資料型態應與「InvitationDate」定義的相同，須以「Date」的資料型態呈現，否則不通過驗證。

但如有一元素名為「ProjectStartDate」，在 Schema 標準文件中找不到與其相同意思的元素，說明了「ProjectStartDate」為新元素，因此其資料型態應該以之前出現過有關「日期」的元素名稱的資料型態為主，如「InvitationDate」，必須以「Date」的方式呈現，否則不通過驗證。

STEP 4：判別元素出現次數是否正確

判斷的準則如 STEP 3，也是以該元素是否有產生「元素英文名稱錯誤」的情況。如果有產生錯誤，則是以 Schema 標準文件中正確名稱的元素定義之出現次數為主；如果沒有錯誤，則該元素的出現次數應符合一般常理。

例如「InvitedDate」為「InvitationDate」的「元素英文名稱錯誤」，所以元素「InvitedDate」的出現次數應與詢價單 Schema 標準文件中的元素「InvitationDate」所定義的相同。但如果是「ProjectStartDate」，因為詢價單 Schema 文件中沒有定義，所以其出現次數應符合常理，以一次為主，如不符合則不通過驗證。

STEP 5：判別該新元素是否通過驗證

當驗證程式檢測出來之元素經過 STEP 2~STEP 4 的判斷之後，如有任一步驟產生錯誤的話，則該元素並不能通過驗證；如果每一步驟都通過的話，則說明了該元素為通過驗證之新元素。

將錯誤列表 3-1 修正成實際驗證狀況，加上執行驗證的方法並改成表 3-2

表 3-2 錯誤類型列表(實際驗證狀況)

錯誤類型	錯誤名稱	元素型態	錯誤說明	執行驗證方法
必要元素未出現	第一型 缺乏必要元素	必要元素	Schema 中所要求的必要元素未在文件表單中	直接由系統檢測
元素英文名稱錯誤	第二型 必要元素英文名稱錯誤	必要元素	文件中出現的元素英文名稱寫法與 Schema 規定的不符	先由系統檢測出新元素，再經由人為判斷。判斷之後，如元素意義與 Schema 定義雷同但英文名稱不同，則歸類為此類錯誤
	第三型 非必要元素英文名稱錯誤	非必要元素		
資料型態錯誤	第四型 必要元素資料型態錯誤	必要元素	元素的內容資料型態與 Schema 所規定的要求不符合	直接由系統檢測
	第五型 非必要元素資料型態錯誤	非必要元素		直接由系統檢測

	第六型 未定義元 素資料形 態錯誤	未定義元 素		先由系統檢測出新 元素，再經由人為 判斷。判斷之後， 如為新元素，則檢 測該元素的資料形 態是否合於規範
元素出現 次數錯誤	第七型 必要元素 出現次數 錯誤	必要元素	元素出現的次數不 在 Schema 規定的 範圍內	直接由系統檢測
	第八型 非必要元 素出現次 數錯誤	非必要元 素		直接由系統檢測
	第九型 未定義元 素出現次 數錯誤	未定義元 素		先由系統檢測出新 元素，再經由人為 判斷。判斷之後， 如為新元素，則檢 測該元素的出現次 數是否合於規範

3.3.2 受驗系統呈現文件之驗證

當受驗系統接受到一份文件時，必須能夠將文件讀取呈現。所以在執行此項驗證時，由驗證單位產出一份測試文件，交由受驗系統做讀取呈現的動作；受驗單位除了需將元素順序、元素名稱及內容都正確呈現之外，還必須能夠呈現元素總數、元素出現次數之相關訊息。系統呈現後，交由驗證單位代表檢驗最後呈現的結果，評估是否通過驗證。

當受驗系統接收到一份如圖 3-14 之 XML 文件後，應將文件之內容作如表 3-3 或類似的方式呈現，才算是可以通過驗證手續。

```

<RequestForQuotation>
  <SheetTitle>
    <SheetName>詢價單</SheetName>
    <SheetDate>2005-09-21</SheetDate>
  </SheetTitle>
  <InvitationDate>2005-09-21</InvitationDate>
  <QuotationDeadline>2005-11-25</QuotationDeadline>
  <Project>
    <ProjectName>工程二館電梯工程</ProjectName>
    <ProjectLocation>交通大學</ProjectLocation>
  </Project>
  <Item>
    <ItemName>10號鋼筋</ItemName>
    <IndustrySpec>
      <SpecName>CNS</SpecName>
      <SpecUnit>號</SpecUnit>
    </IndustrySpec>
  </Item>
  <DeliveryLocation>交通大學工程二館</DeliveryLocation>
</RequestForQuotation>

```

圖 3-14 測試之 XML 文件範例

根據圖 3-14 的測試文件內容，將其內容資料轉換成表 3-3 的方式呈現出來。

表 3-3 XML 文件內容呈現範本

根元素名稱	第一層元素名稱	第二層元素名稱	第三層元素名稱	元素資料內容	元素出現次數
RequestForQuotation					1
	SheetTitle				1
		SheetName		詢價單	1
		SheetDate		2005-09-21	1
	InvitationDate			2005-09-21	1

	QuotationDeadline			2005-11-25	1
	Project				1
		ProjectName		工程二館電梯工程	1
		ProjectLocation		交通大學	1
	Item				1
		ItemName		10 號鋼筋	1
		IndustrySpec			1
			SpecName	CNS	1
			SpecUnit	號	1
	DeliveryLocation			交通大學工程二館	1

3.3.3 驗證通過準則

每一測試系統都需通過各類錯誤的檢測，全部通過才算驗證合格，如有任一項目未通過驗證，即簽發「未通過」驗證報告。

3.4 簽發驗證報告與儲存資料階段

此階段分為兩個步驟，簽發驗證結果報告以及儲存驗證結果於驗證機構的資料庫。當受驗單位的系統驗證完畢之後，由驗證機構簽發一份驗證結果報告書(詳見附錄二)給受驗單位，報告書中詳細記載驗證時間，通過驗證的等級以及驗證錯誤的原因等等相關資料，一式兩份，分別由受驗單位及驗證機構各自收存，作為以後備查之用。

當驗證報告已簽發完畢後，驗證機構人員會將這次驗證的結果記錄在驗證機構的資料庫中，內容包含廠商名稱、驗證日期、驗證系統版本、驗證通過文件、文件產生之錯誤類型等等，以供日後審查之用。

3.5 小結

本節主要分析驗證機制的運作流程，之後根據驗證機制的需求進而建置驗證主系統，以完成整套的運作流程。以下將驗證機制流程依照各階段分類，並且再加入各階段作業人員、動作概述加以整理如表 3-4：



表 3-4 驗證機制流程整理表

階段 說明項目	第一階段 前置準備階段		第二階段 格式驗證階段	第三階段 系統驗證階段		第四階段 報告與儲存階段	
	流程動作	申請驗證手續	驗證準備	檢驗欲受驗 XML 文件之格式	驗證文件元素名稱、資料型態、以及資料出現次數	驗證受驗系統是否能夠近似呈現接收到之文件	簽發驗證結果報告
作業人員與系統	受驗單位代表	受驗單位代表 受驗單位系統 驗證機構代表 驗證機構系統	受驗單位系統 驗證機構代表 驗證單位系統	受驗單位系統 驗證機構代表 驗證機構系統	受驗單位系統 驗證機構代表 驗證機構系統	受驗單位代表 驗證機構代表	驗證機構代表
動作概述	領取驗證申請單，填寫完畢後隨著其他必要文件一起繳回驗證機構	將執行驗證實所需要之軟硬體設備安裝完畢	由受驗系統產出欲受驗之 XML 文件，交由驗證系統驗證文件的格式是否符合 XML 標準規則	檢驗 XML 文件中元素的出現與否、資料型態、出現次數等必要資訊	由驗證系統產出一份測試文件至受驗系統，檢驗受驗系統能否將元素名稱、元素順序、資料型態及出現次數正確呈現	將驗證的結果以及錯誤訊息紀錄於結果報表中，雙方各執一份留存	將受驗證的相關訊息紀錄於驗證機構之資料庫中，便於日後查詢



第4章 系統說明與系統實作

本研究所提出的 XML 文件驗證程式，是利用 XML 技術中的文件物件模型 (Document Object Model, DOM)物件、錯誤剖析(ParseError)物件以及節點(Node)物件等等相關物件，再加上以 Visual Basic 6.0 為程式撰寫介面共同開發。當系統接受到一份格式正確(Well-Formed)之 XML 文件時，先將其轉換成樹狀(DOM Tree)的結構，之後再根據 DOM Tree 的節點做驗證分析，顯示各 Node 的驗證訊息，就可得知文件是否通過驗證以及錯誤的類別。

4.1 系統說明

4.1.1 系統分析與功能

圖 4-1 為本驗證系統之 IDEF0 圖，輸入的物件為營建電子採購 XML 文件，系統控制物件則為營建電子採購 Schema 標準，而控制人員則為執行驗證的人員，最後產出的資訊為驗證訊息。

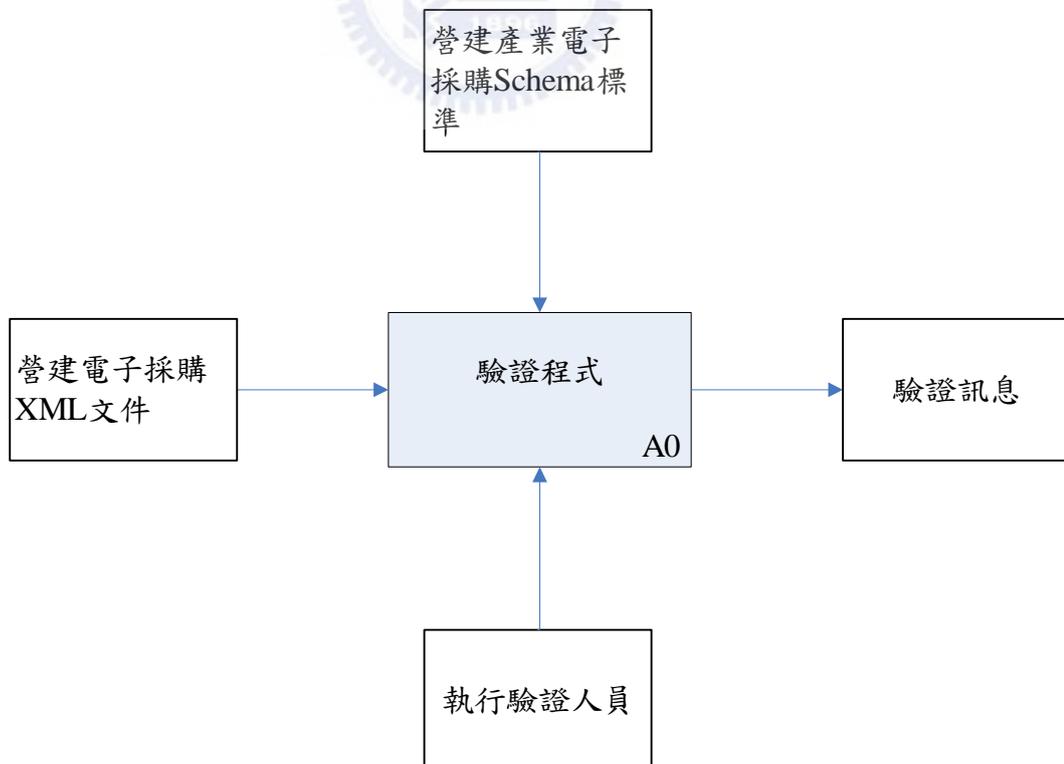


圖 4-1 驗證系統 IDEF 圖

本系統最主要的功能，是以所制定營建物料採購資訊交換 Schema 標準為規範，透過 XML 技術的 DOM、Node 以及 ParseError 等物件，檢驗 XML 文件之內容與格式是否與 Schema 標準相符

4.1.2 XML DOM 物件

文件物件模型(Document Object Model, DOM)是一種以程式設計方式進行讀取、修改及操作 XML 文件。DOM 物件在對文件進行解讀時，會以階層結構的方式存於記憶體中，通常稱此結構為 XML DOM Tree，如圖 4-2 所示。圖中 Document 為文件之根元素，其底下之子元素則有 Document Element 以及 Element，Element 底下之子元素則可為 Attr(屬性)、Text(文字)或 Element(元素)。MSXML(Microsoft XML Core Service, XML 軟體套裝物件)就是利用 XML DOM 元件，將 XML 切割成一個個的節點，然後再針對各節點的標籤做 Well-Formed 的檢測；如 XML 文件擁有所參照的 Schema 之文件，會依據 Schema 的宣告與 XML DOM Tree 進行比對的動作，檢查是否符合所宣告的 Schema。

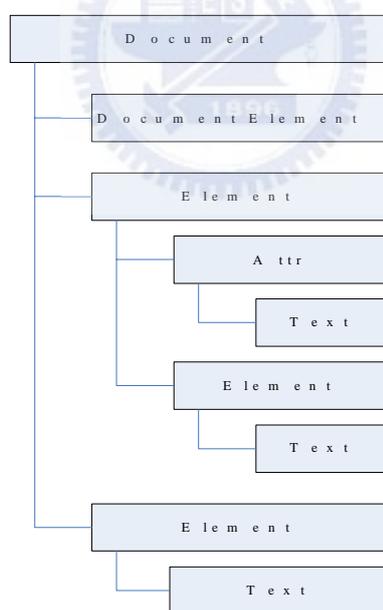


圖 4-2 XML DOM Tree

圖 4-3 表示的為當使用 MSXML 做剖析器時，會先將 XML 文件轉換成 XML DOM Tree，由應用程式讀取 XML DOM Tree 中的資料，再透過應用程式介面(Application Program Interface, API)對 XML 文件進行修改的動作，也是於 XML DOM Tree 中的節點執行編輯的動作。

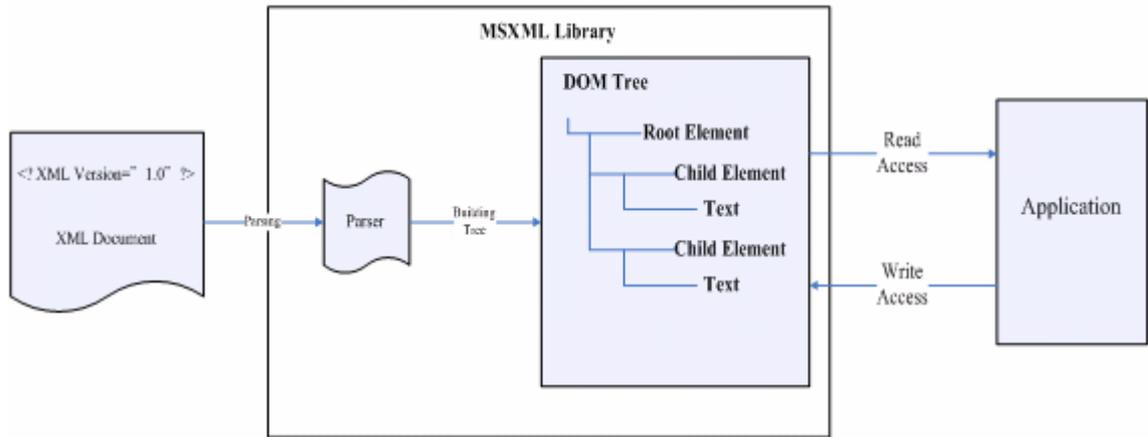


圖 4-3 XML 與 API 關聯圖(MSXML4.0)

如在 VB6 中要宣告使用 XML DOM 時，首先必須要參照(Reference)到 MSXML5.0 版本的物件，才可以使用其中之 DOM 物件。其宣告的語法如下：

```
Dim XMLDoc As DOMDocument50
```

此段語法為定義一參數 XMLDoc 為 XML DOM 5.0 版本的文件；系統本身會按照此宣告的方式將 XML 文件轉換成 DOM 結構，之後再陸續執行相關操作 XML 內容的動作。

4.1.3 Node 物件

當 XML DOM Tree 產生以後，XML 文件裡的所有資料全都被放入於此樹之某些節點中，於是就可以透過 Node 物件查詢出我們所需要的相關訊息。

表 4-1 及表 4-2 為在本研究開發的驗證程式裡所用到的相關節點(Node)物件以及節點列表(NodeList)物件，透過這些相關物件可以將 XML DOM Tree 中的元素節點做驗證的動作。

表 4-1 Node 物件之屬性值列表

屬性名稱	說明
childNodes	傳回該節點所有的子節點所形成的節點列表物件，並可以利用索引的方式取出各個子節點。
selectNodes	傳回符合條件的所有元素節點所形成的節點列表物件
Nodename	傳回所有符合規則的屬性名稱，元素名稱等等的節點名稱。

表 4-2 Nodelist 物件之屬性值列表

屬性名稱	說明
Length	傳回該元素節點列表有幾個元素節點
Item(i)	傳回該元素節點列表物件內的第 i 個元素
Nextnode	傳回下一個元素

Node 物件在 VB6 環境下的宣告之語法如下所示：

```
01 Dim xmlDoc As XmlDocument50
02 Dim currNode As IXMLDOMNode
```

此段語法除了第一行定義 xmlDoc 為 XML DOM 5.0 版本之外，另外第二行還定義了 currNode 為 XML DOM 中的 Node 物件。宣告之後就可以使用 Node 物件中的屬性值，使用方式為直接於 currNode 之後加上屬性名稱即可，如 currNode.NodeName。而 Nodelist 物件宣告如同 Node 物件，如下所示：

```
01 Dim xmlDoc As XmlDocument50
02 Dim objNodeList As IXMLDOMNodeList
```

第一行的語法是定義參數 xmlDoc 為 XML DOM5.0 版本，而第二行則是宣告參數 objNodeList 可以使用 Node 物件以及 NodeList 物件屬性。透過 Node 物件、Nodelist 物件以及 VB6 之迴圈結構找出文件中全部的節點，之後再針對各個節點執行驗證的動

作，就可以知道產生錯誤的地方是位於文件中的哪些節點處以及錯誤的類型。因此這兩個物件是驗證程式最核心的部份。

4.1.4 ParseError 物件

當 XML 文件在執行驗證的動作時，可以透過 ParseError 物件瞭解文件驗證成功與否或是發生錯誤的原因。相同的，ParseError 物件也是擁有許多屬性值，如表 4-3，運用這些屬性值來幫助驗證程式對於文件錯誤的地方以及原因作說明，讓執行驗證的人員可以根據所產出的錯誤原因訊息歸類錯誤的類型，以利於整體驗證流程的執行。

表 4-3 ParseError 物件之屬性值列表

屬性名稱	說明
errorCode	傳回錯誤代碼
line	傳回文件中發生錯誤的列次
reason	傳回文件發生錯誤的原因

圖 4-4 為一段本驗證程式的程式碼，說明如何於 VB6 中使用 ParseError 物件之屬性值：

```

01 Dim oError As IXMLDOMParseError
02     Set oError = oXMLDoc.Validate
03     If oError.errorCode <> 0 Then
04         strout = strout + vbTab _
05             + "XMLDoc is not valid because"
06             + vbNewLine + oError.reason +
07             vbNewLine
08     Else
09         strout = strout _
10             + vbTab + "文件驗證通過" +
11             vbNewLine + vbNewLine
12     End If

```

圖 4-4 ParseError 物件使用範例程式碼

本程式碼是先宣告 oError 變數為 ParseError 物件，接著使用迴圈結構，當

oError.errorCode 不等於 0 時(亦即代表有錯誤產生)，程式本身會利用 oError.reason 顯示出錯誤的原因，證明此份文件含有錯誤；如果 oError.errorCode 等於 0 時，程式就會顯示出”文件通過驗證”的訊息給執行驗證人員知道，說明此份文件是合格的。驗證人員可以根據產出的訊息，再加上一些比對的工作，就可以判斷出文件中出現何種錯誤類型。

4.2 實作成果

4.2.1 驗證系統介面展示

驗證系統是使用 Visual Basic 6.0 撰寫程式操作的介面，並且加上 XML 相關物件共同開發。系統介面如圖 4-5，供執行驗證人員操作之用。



圖 4-5 驗證程式介面圖

程式介面各部份說明：

- (1) 欲驗證之文件檔(XML)：此行包含兩個物件，一個為空白的文字方塊，一個則為選擇檔案的控制鈕。執行驗證人員可以透過控制鈕選取欲驗證的 XML 文件，而被選擇的檔案名稱會顯示於空白的文字方塊中；亦或是直接於文字方塊中輸入欲驗證的檔案名稱，也相同於選擇的功能。
- (2) 欲驗證之標準文件檔(XSD)：此行也同樣包含了空白文字方塊以及檔案選取鈕兩個物件，執行驗證人員透過檔案選取鈕選擇欲參考的文件檔，亦即為 Schema 文件。
- (3) 欲驗證之元素節點名稱：當欲驗證之 XML 文件以及 Schema 文件都選取完畢後，就可以檢驗整份文件的元素節點。如有需要檢驗某一特定元素時，可以直接在空白的文字方塊中輸入該元素的名稱(如不輸入，預設值為文件根元素節點)，就可以得到該元素及其底下之子節點驗證之後的訊息。
- (4) 驗證結果訊息：當上面三部份都已經輸入完畢之後，就可以按下「執行驗證」的控制鈕，系統會於空白文字方塊中顯示執行驗證後的訊息。所顯示的訊息有三部份，第一部分為整份文件驗證訊息，第二部份為根元素節點(父節點)驗證訊息，第三部份則為其下的子節點驗證訊息。執行驗證人員可以根據顯示的訊息來判別錯誤的類型，了解產生錯誤之處，以便於驗證報告的簽發。

圖 4-6 為一份符合詢價單 Schema 之 XML 範例文件，其中沒有新元素的增加以及其他錯誤產生；將圖 4-6 文件利用驗證程式執行比對偵錯的動作，可得到驗證成功之訊息，如圖 4-7。

```

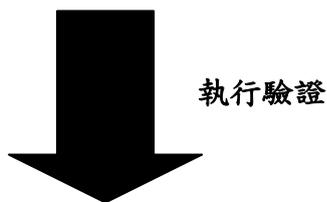
<RequestForQuotation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SheetTitle>
    <SheetName>詢價單</SheetName>
    <SheetDate>2005-09-21</SheetDate>
  </SheetTitle>
  <InvitationDate>2005-09-21</InvitationDate>
  <QuotationDeadline>2005-11-25</QuotationDeadline>
  <InvitedCom>
    <Contact>
      <Name>伊炎涼</Name>
      <MobileNo>0988888888</MobileNo>
    </Contact>
    <ComName>太潤營造股份有限公司</ComName>
  </InvitedCom>
  <Project>
    <ProjectName>工程二館電梯工程</ProjectName>
    <ProjectLocation>交通大學</ProjectLocation>
  </Project>
  <SubContract>
    <Name>鋼筋採購</Name>
  </SubContract>
  <Item>
    <ItemName>10號鋼筋</ItemName>
    <IndustrySpec>
      <IndustryID/>
      <Spec>
        <SpecName>CNS</SpecName>
        <SpecUnit>號</SpecUnit>
      </Spec>
    </IndustrySpec>
  </Item>
  <DeliveryLocation>交通大學工程二館</DeliveryLocation>
  <PaymentTerm>
    <PaymentOnDelivery>
      <PaymentName>鋼筋交貨款</PaymentName>
      <PaymentType>

```

圖 4-6 受測試之 XML 文件(符合詢價單 Schema)

```
<Transferring>
  <BankNo>700</BankNo>
  <BankAccount>05264786521348</BankAccount>
</Transferring>
</PaymentType>
</PaymentOnDelivery>
<DownPayment>
  <PaymentName>鋼筋訂金</PaymentName>
  <PaymentType>
    <Cash>
      <Amount>3500000</Amount>
    </Cash>
  </PaymentType>
</DownPayment>
</PaymentTerm>
<AcceptTerm>
  <TermNo>1</TermNo>
  <TermContent>號數及數量要正確</TermContent>
</AcceptTerm>
</RequestForQuotation>
```

圖 4-6 續 受測試之 XML 文件(符合詢價單 Schema)



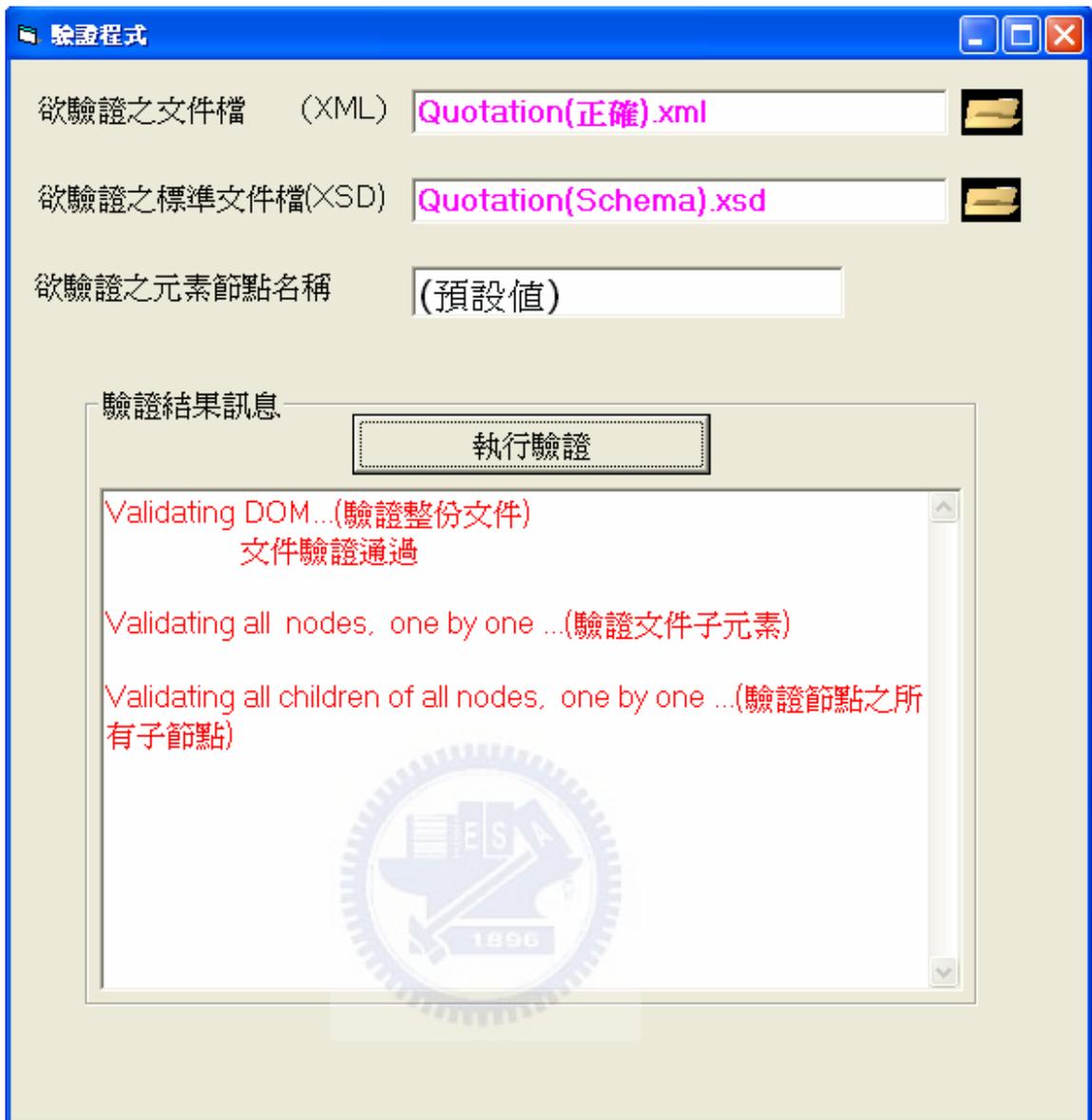


圖 4-7 文件驗證成功訊息

4.2.2 缺乏必要元素錯誤驗證

圖 4-8 為一份符合 Schema 標準之部分 XML 文件，而圖 4-9 則是將必要元素 <Project>刪除的部份 XML 文件。將圖 4-9 文件利用驗證程式與詢價單 Schema 文件進行比對偵錯的動作，並於文字方塊中顯示驗證錯誤訊息，如圖 4-10 所示。

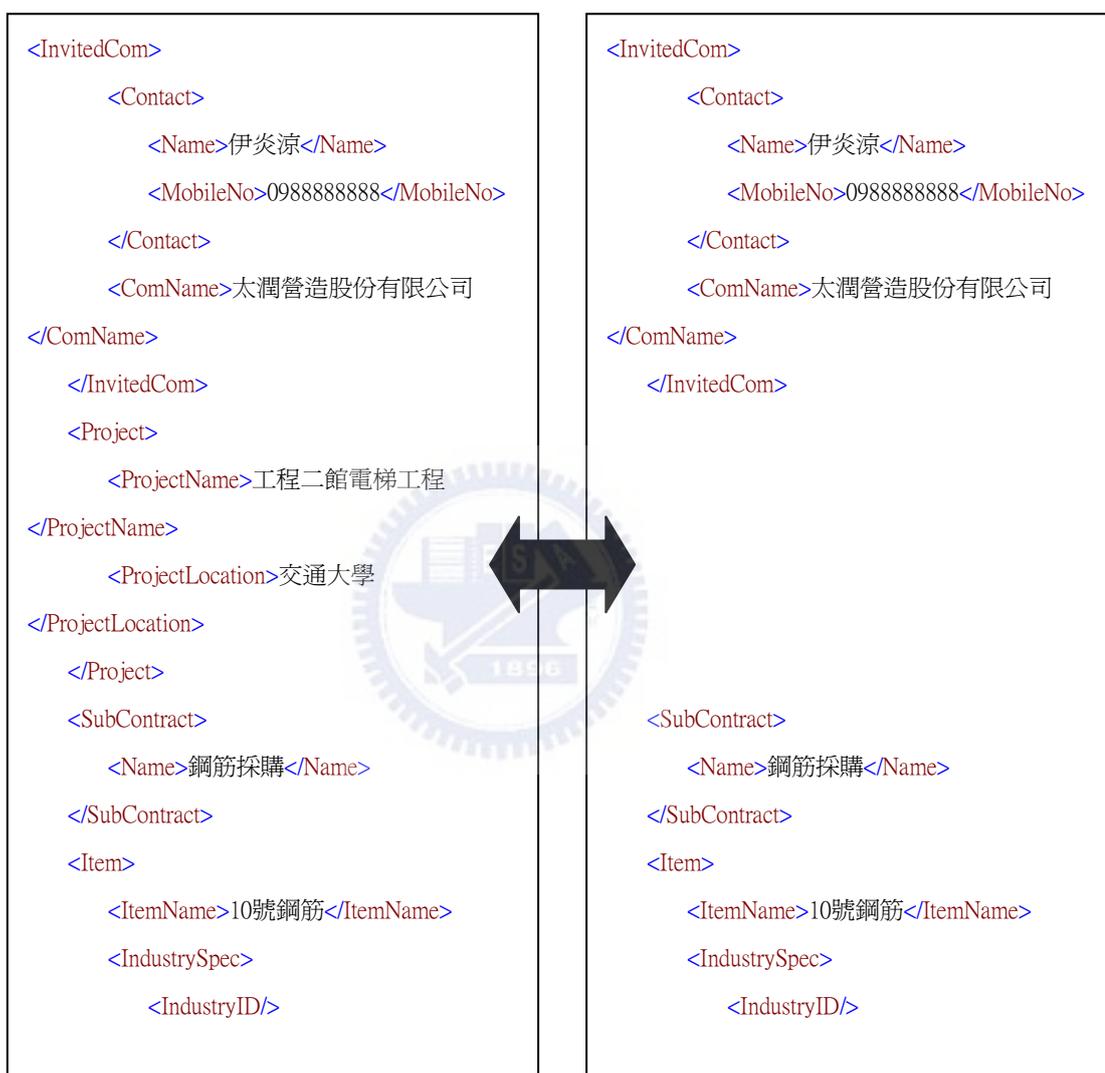


圖 4-8 包含<Project>元素之文件

圖 4-9 缺少<Project>元素之文件



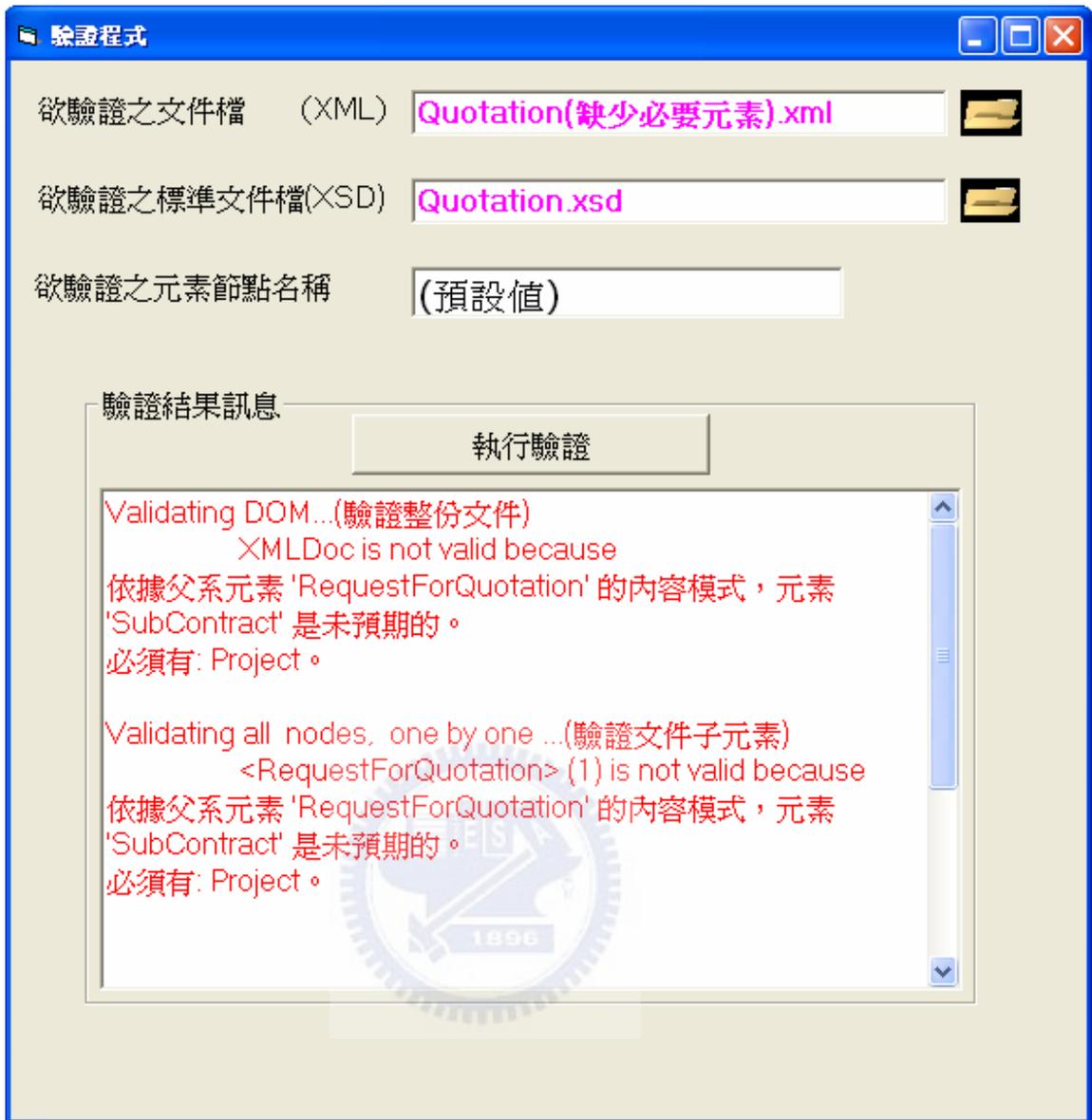


圖 4-10 缺乏必要元素<Project>驗證訊息

驗證結果訊息說明：

系統判別結果：此訊息表示根據詢價單 Schema 的定義宣告，表 4-7 文件的內容是有缺少必要元素，而缺少的必要元素則是為<Project>，因此產生驗證錯誤訊息。

人工判別結果：根據驗證訊息，將此錯誤歸類於缺乏必要元素之錯誤。

4.2.3 必要元素名稱錯誤驗證

圖 4-11 為一份符合詢價單 Schema 標準的部份 XML 文件，圖 4-12 則是將圖 4-11 文件中的必要元素<DeliveryLocation>換寫成<SendAddress>，接著再利用驗證程式執行驗證偵錯的動作。系統會先偵察出缺少的必要元素< DeliveryLocation >以及增加的未定義元素<SendAddress>，如圖 4-13 所示；再由驗證人員檢視兩者是否意義是否雷同；以此例來說兩者元素名稱的意義是相同的，因此我們將此類錯誤歸類於必要元素名稱錯誤。

```
<Spec>
    <SpecName>CNS</SpecName>
    <SpecUnit>號</SpecUnit>
</Spec>
</IndustrySpec>
</Item>
<DeliveryLocation>交通大學工程二館
</DeliveryLocation>
<PaymentTerm>
    <PaymentOnDelivery>
        <PaymentName>鋼筋交貨款
    </PaymentName>
    <PaymentType>
        <Transferring>
            <BankNo>700</BankNo>
        </Transferring>
        <BankAccount>05264786521348</BankAccount>
    </PaymentType>
    </PaymentOnDelivery>
</PaymentTerm>
```

圖 4-11 必要元素名稱正確之文件

```
<Spec>
    <SpecName>CNS</SpecName>
    <SpecUnit>號</SpecUnit>
</Spec>
</IndustrySpec>
</Item>
<SendAddress>交通大學工程二館
</SendAddress>
<PaymentTerm>
    <PaymentOnDelivery>
        <PaymentName>鋼筋交貨款
    </PaymentName>
    <PaymentType>
        <Transferring>
            <BankNo>700</BankNo>
        </Transferring>
        <BankAccount>05264786521348</BankAccount>
    </PaymentType>
    </PaymentOnDelivery>
</PaymentTerm>
```

圖 4-12 必要元素名稱錯誤之文件



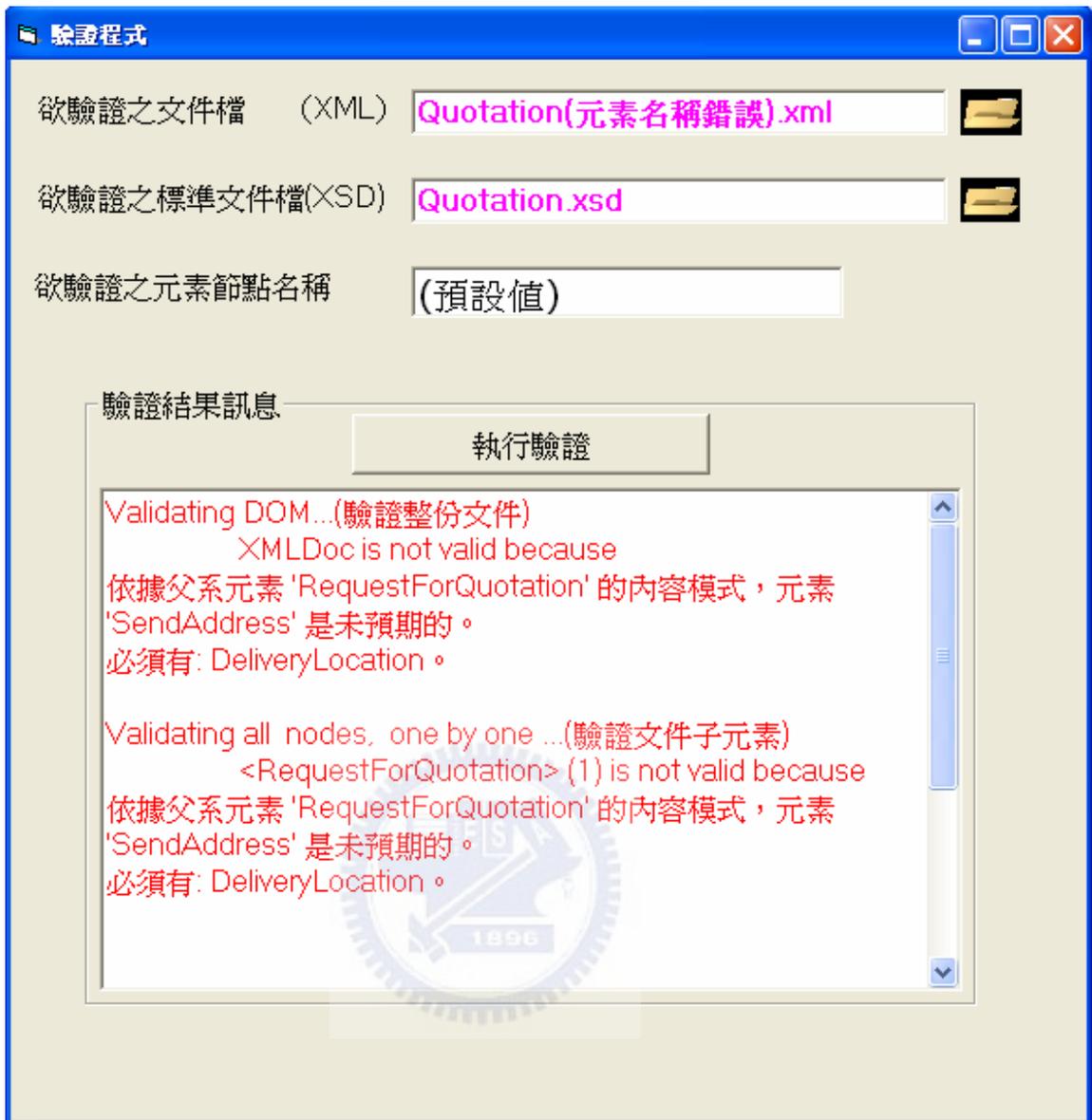


圖 4-13 必要元素<DeliveryLocation>名稱驗證訊息

驗證結果訊息說明：

系統判別結果：此訊息表示驗證系統找出文件中含有一個名為<SendAddress>的新元素，而且也缺少了一個名為<DeliveryLocation>的必要元素。

人工判別結果：驗證人員依據此訊息，判斷此兩元素雖然名稱不同，但其意義相似，因此判別此狀況為必要元素名稱錯誤。

4.2.4 非必要元素名稱錯誤驗證

圖 4-14 為一份符合詢價單 Schema 標準之部分 XML 文件，而表 4-15 則是表 4-14 文件中的非必要元素<AcceptTerm>改寫成<RecieveTerm>，之後再將此份文件利用驗證程式執行偵錯的動作。相較於驗證必要元素名稱錯誤的結果，差異處在於系統只會將元素< RecieveTerm >判別為 Schema 未定義的新元素，並於驗證訊息欄顯示相關訊息，但並不會呈現元素<AcceptTerm>未出現的訊息，如圖 4-16 所示。原因在於元素<AcceptTerm>出現與否都不會影響整份文件的驗證結果，所以才會有這樣的現象產生。判別此型錯誤的準則與必要元素名稱錯誤相同，相同意義但不同元素名稱才能歸類於此類錯誤。

```
<DownPayment>
  <PaymentName>鋼筋訂金
</PaymentName>
  <PaymentType>
    <Cash>
      <Amount>3500000</Amount>
    </Cash>
  </PaymentType>
</DownPayment>
</PaymentTerm>
<AcceptTerm>
  <TermNo>1</TermNo>
  <TermContent>號數及數量要正確
</TermContent>
</AcceptTerm>
</RequestForQuotation>
```

圖 4-14 非必要元素名稱正確之文件

```
<DownPayment>
  <PaymentName>鋼筋訂金
</PaymentName>
  <PaymentType>
    <Cash>
      <Amount>3500000</Amount>
    </Cash>
  </PaymentType>
</DownPayment>
</PaymentTerm>
<RecieveTerm>
  <TermNo>1</TermNo>
  <TermContent>號數及數量要正確
</TermContent>
</AcceptTerm>
</RequestForQuotation>
```

圖 4-15 非必要元素名稱錯誤之文件



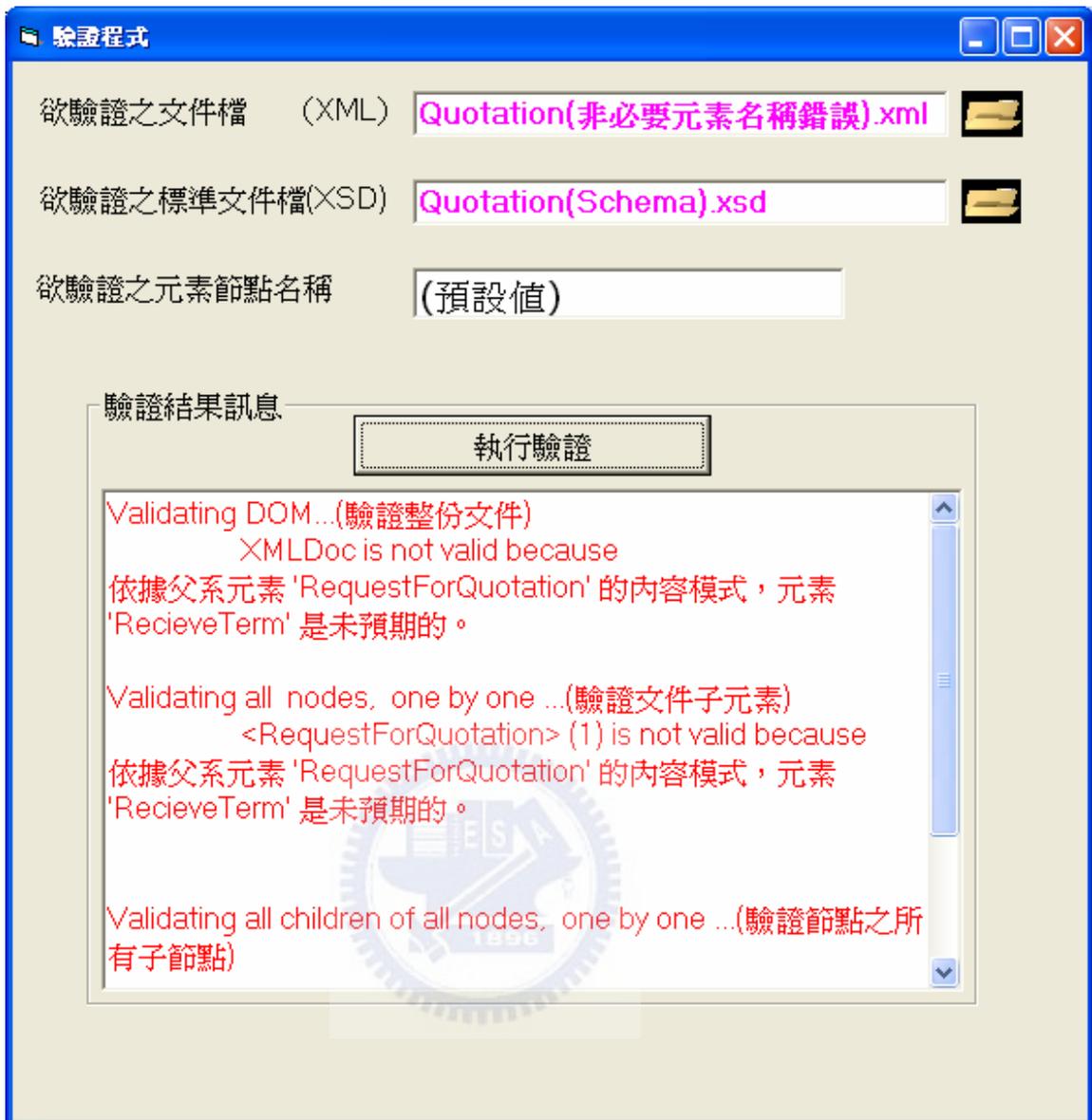


圖 4-16 非必要元素<AcceptTerm>名稱驗證訊息

驗證結果訊息說明：

系統判別結果：此訊息表示元素<RecieveTerm>並未於詢價單 Schema 中定義，也就表示元素< RecieveTerm >為一新元素。

人工判別結果：驗證人員依據此訊息，先行檢查詢價單 Schema 中所定義的非必要元素裡是否有與元素< RecieveTerm >意義雷同但名稱不同的元素，結果發現有一非必要元素<AcceptTerm>符合此條件。因此，將此驗證結果歸類於非必要元素名稱錯誤。

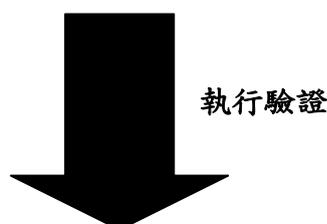
4.2.5 必要及非必要元素資料型態錯誤驗證

圖 4-17 為一份符合詢價單 Schema 標準的部份 XML 文件，圖 4-18 則為將圖 4-17 文件中之必要元素<InvitationDate>以及非必要元素<SheetDate>中的資料型態改寫成 XML 不接受的日期表現形式，再由驗證程式執行偵錯的動作。<InvitationDate>以及<SheetDate>中的資料型態應以”西元年份-月份-日期”的方式表現，如非以此格式填寫，系統則會顯示資料格式錯誤的訊息，如圖 4-19 所示。



圖 4-17 元素資料型態正確之文件

圖 4-18 元素資料型態錯誤之文件



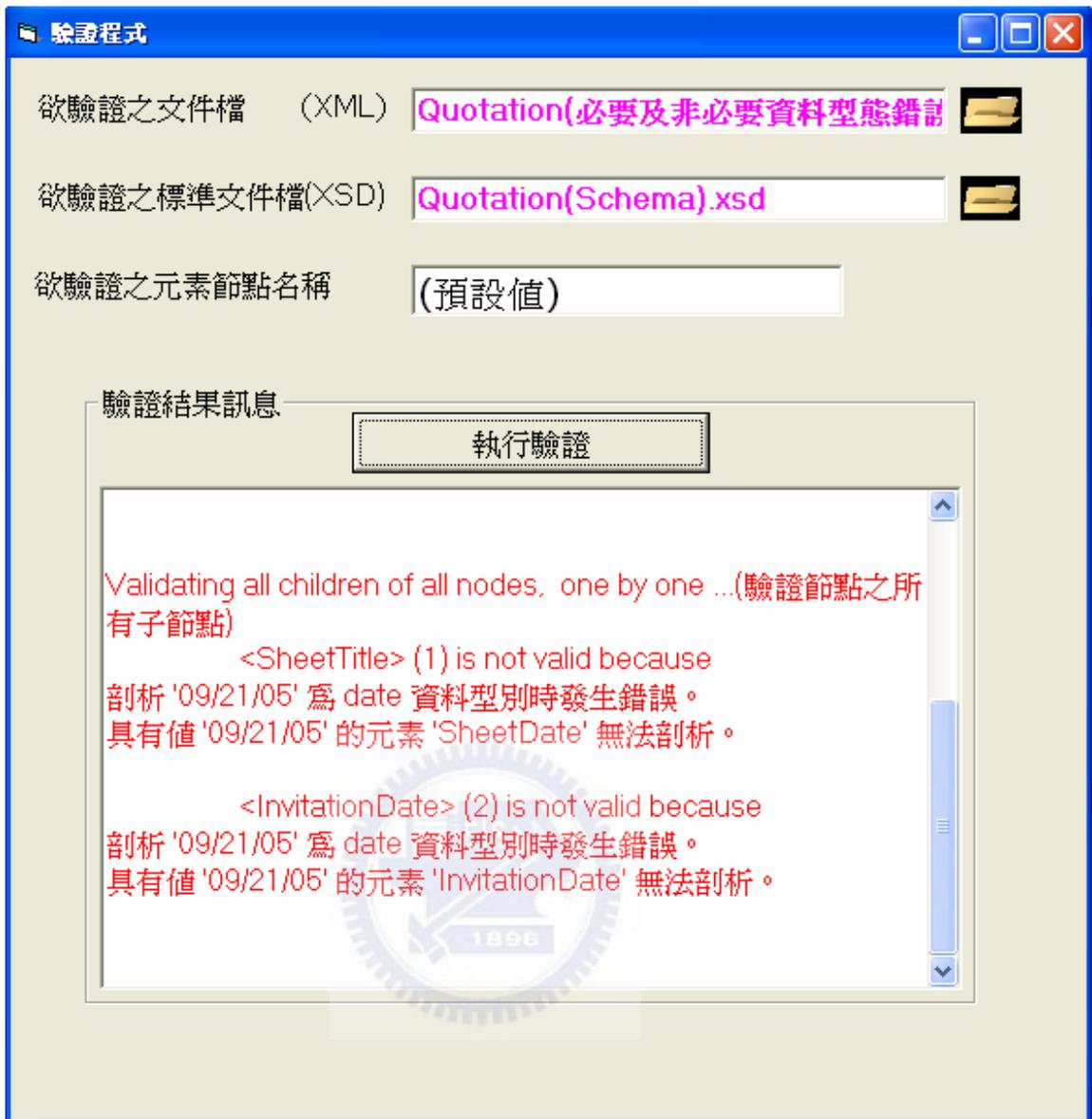


圖 4-19 <SheetDate>以及<InvitationDate>資料型態驗證訊息

驗證結果訊息說明：

系統判別結果：此訊息表示系統在剖析非必要元素<SheetDate>以及必要元素<InvitationDate>之資料型態時，偵測到兩元素的資料內容是以”月/日/年”的方式表現，而非以詢價單 Schema 規定的”年-月-日”的 Date 型態呈現。

人工判別結果：依據驗證訊息，將此驗證結果歸類於必要元素及非必要元素資料型態錯誤。

4.2.6 未定義元素資料型態錯誤

圖 4-20 為一份符合詢價單 Schema 標準之部分 XML 文件，圖 4-21 則為將圖 4-20 文件內加入一未定義元素<ProjectStartDate>，再由驗證系統執行偵錯的動作。雖然元素<ProjectStartDate>沒有在詢價單 Schema 中定義，但是任何描述日期之元素所屬的資料型態應為”Date”之型態，才符合 Schema 定義的規則。如非以此型態出現，則歸類於此型式的錯誤。

```
<Contact>
  <Name>伊炎涼</Name>
  <MobileNo>0988888888</MobileNo>
</Contact>
<ComName>太潤營造股份有限公司
</ComName>
<InvitedCom>
<Project>
  <ProjectName>工程二館電梯工程
</ProjectName>
  <ProjectLocation>交通大學
</ProjectLocation>
</Project>
<SubContract>
  <Name>鋼筋採購</Name>
</SubContract>
<Item>
  <ItemName>10號鋼筋</ItemName>
  <IndustrySpec>
```

圖 4-20 未定義元素型態正確之文件

```
<Contact>
  <Name>伊炎涼</Name>
  <MobileNo>0988888888</MobileNo>
</Contact>
<ComName>太潤營造股份有限公司
</ComName>
<InvitedCom>
<Project>
  <ProjectStartDate>03/21/06
</ProjectStartDate>
  <ProjectName>工程二館電梯工程
</ProjectName>
  <ProjectLocation>交通大學
</ProjectLocation>
</Project>
<SubContract>
  <Name>鋼筋採購</Name>
</SubContract>
<Item>
  <ItemName>10號鋼筋</ItemName>
  <IndustrySpec>
```

圖 4-21 未定義元素型態錯誤之文件



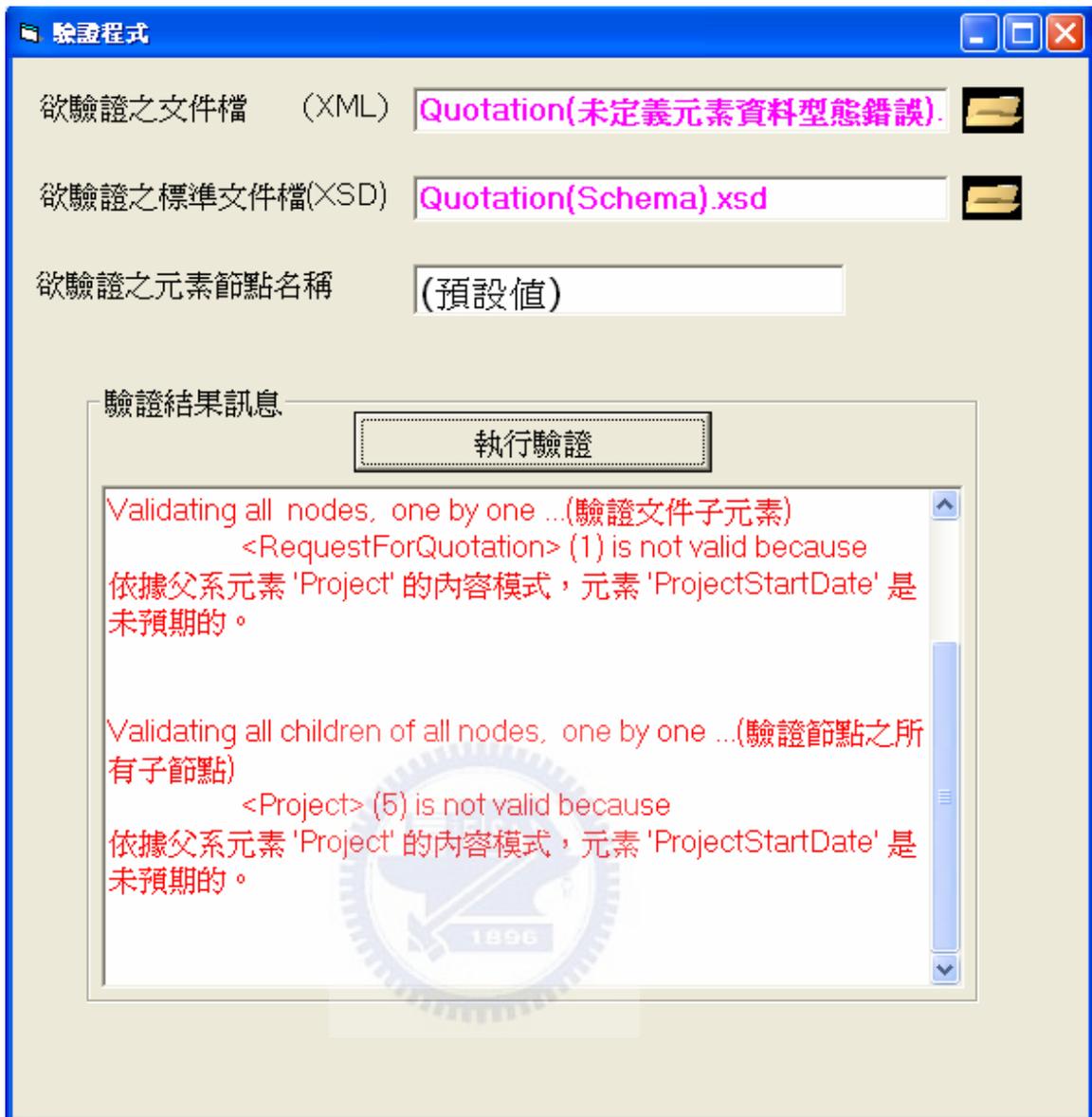


圖 4-22 未定義元素<ProjectStartDate>驗證訊息

驗證結果訊息說明：

系統判別結果：系統先行檢測出有一元素<ProjectStartDate>是未在詢價單 Schema 中有詳細定義，再經由驗證人員檢驗詢價單 Schema 內容之後，確定此元素是為一新元素。

人工判別結果：執行驗證人員檢查該元素所包含的資料型態，發現元素的內容呈現方式並非以”Date”的型態呈現，因此將此驗證結果歸類於未定義元素資料型態錯誤。

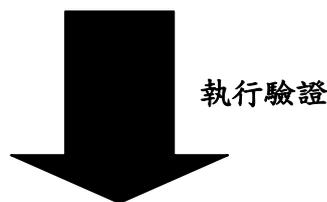
4.2.7 必要元素及非必要元素出現次數錯誤驗證

圖 4-23 為一份符合詢價單 Schema 標準之部分 XML 文件，圖 4-24 為將圖 4-23 文件中的必要元素<QuotationDeadline>以及非必要元素<ProjectLocation>的出現次數都各改為兩次，再由驗證程式執行偵錯的動作，結果如圖 4-25 所示。對於詢價單來說，此兩者元素最多只可以出現一次，出現一次以上就已屬不合理，如有此狀況產生都是屬於出現次數的錯誤。



圖 4-23 元素出現次數正確之文件

圖 4-24 元素出現次數錯誤之文件



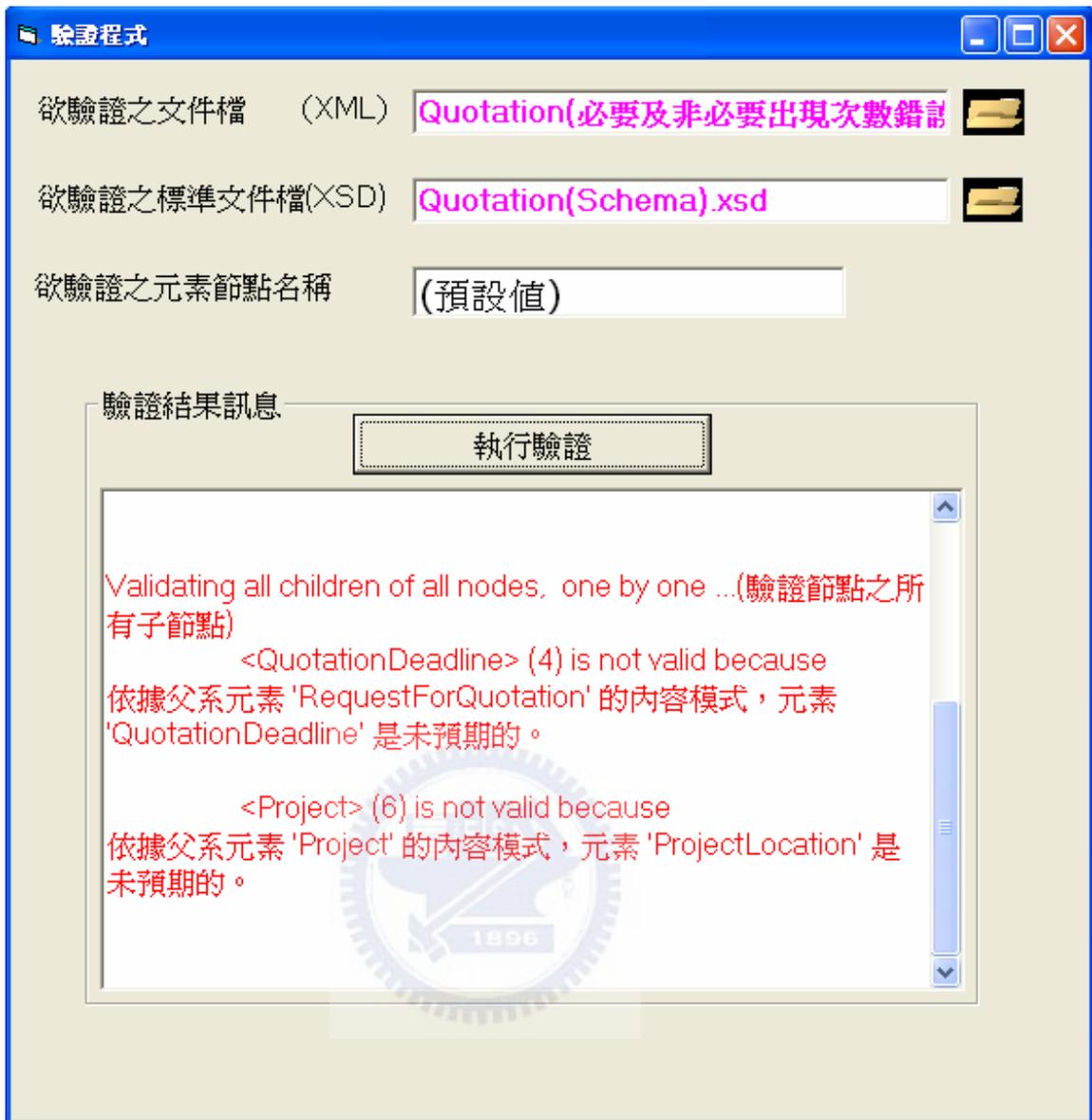


圖 4-25 <QuotationDeadline>及<ProjectLocation>出現次數驗證訊息

驗證結果訊息說明：

系統判別結果：系統檢測出<QuotationDeadline>及<ProjectLocation>為一未定義之元素，判別為新元素。

人工判別結果：執行驗證人員發現詢價單 Schema 中對此兩元素有詳細的定義，因此此訊息代表必要元素< QuotationDeadline >以及非必要元素<ProjectLocation>出現的次數超過詢價單 Schema 中規定的最大次數，造成此驗證訊息的產生。因此將此型的錯誤歸類於必要元素或非必要元素出現次數的錯誤。

4.2.8 未定義元素出現錯誤驗證

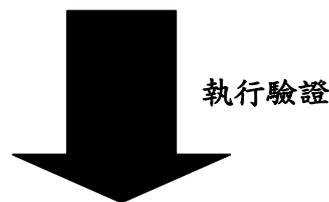
圖 4-26 為一份符合詢價單 Schema 標準的部份 XML 文件，圖 4-27 則將圖 4-26 文件加入一未定義元素<ProjectStartDate>兩次，再利用驗證程式執行偵錯的動作，如圖 4-28 所示。雖然詢價單 Schema 中對於<ProjectStartDate>未有詳細定義，但如果有出現於文件中，最多也只能出現一次，出現一次以上即為錯誤。

```
<ComName>太潤營造股份有限公司
</ComName>
</InvitedCom>
<Project>
</Project>
<ProjectName>工程二館電梯工程
</ProjectName>
<ProjectLocation>交通大學
</ProjectLocation>
</Project>
<SubContract>
</ProjectLocation>>
</Project>
<SubContract>
<Name>鋼筋採購</Name>
</SubContract>
<Item>
<ItemName>10號鋼筋</ItemName>
```

圖 4-26 未定義元素出現次數正確之文件

```
<ComName>太潤營造股份有限公司
</ComName>
</InvitedCom>
<Project>
<ProjectStartDate>2006-01-01
</ProjectStartDate>
<ProjectStartDate>2006-03-01
</ProjectStartDate>
<ProjectName>工程二館電梯工程
</ProjectName>
<ProjectLocation>交通大學
</ProjectLocation>
</Project>
<SubContract>
</ProjectLocation>>
</Project>
<SubContract>
</ProjectLocation>>
</Project>
<SubContract>
<Name>鋼筋採購</Name>
</SubContract>
```

圖 4-27 未定義元素出現次數錯誤之文件



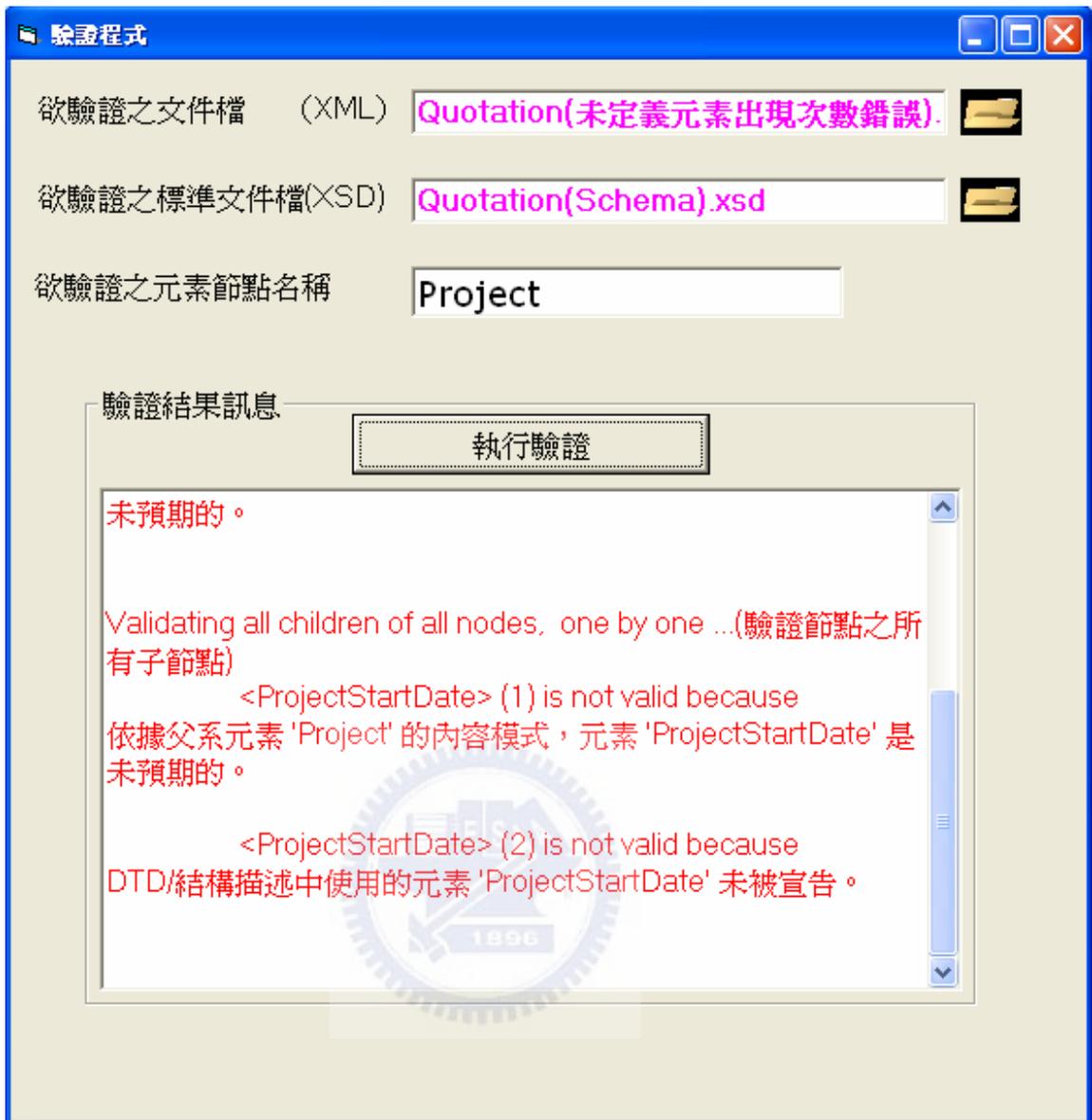


圖 4-28 未定義元素<ProjectStartDate>出現次數驗證訊息

驗證結果訊息說明：

系統判別部份：系統檢測出元素<ProjectStartDate>在文件出現兩次，且此元素也未於 Schema 中定義，判別為新元素。

人為判別部份：驗證人員根據此訊息檢驗詢價單 Schema 後，發現此元素的確為一新元素。執行驗證人員判斷此元素所代表的意思是”專案開工日期”，所以最多也只能出現一次，但在文件中出現了兩次，而造成了驗證錯誤。因此將此型的錯誤都歸類於未定義元素出現次數的錯誤。

4.3 小結

雖然驗證程式本身會自動產出驗證訊息，但還是需要透過專業的人員解讀之後才能確定是何種錯誤的產生，因為系統只會對於元素缺少、新元素加入以及資料型態錯誤做判別的動作，至於其他型態的錯誤都是需要執行驗證的人員對照 Schema 標準的內容才可以下定論。如果能將人工智慧的技術導入驗證系統中，讓系統自動化的偵測出文件的各種錯誤，肯定會讓驗證機制更加優越。



第5章 結論與建議

5.1 結論

本論文是由申請驗證開始，到最後的簽發驗證報告結束，擬定出一套完整的驗證機制流程；並且也根據所要驗證的項目，開發出了一個相關的 XML 文件驗證程式，以利於找出 XML 文件不合格之處。藉著本研究所建置之驗證流程以及驗證程式，可以達到下列幾項效益：

- (1) 完成一套完整的 XML 文件驗證流程，確保營建廠商電子採購系統間資料交換的合理與合法性。
- (2) 整理可能產生錯誤的類型，以求受驗證單位可以針對文件錯誤之處加以改進。
- (3) 取代人工作業，利用驗證程式節省 XML 文件驗證的時間。
- (4) 透過驗證程式的運用，可以直接了解文件中各節點的詳細資料，並且可以根據系統驗證的訊息，明確指出文件錯誤發生的節點，進而了解產生的錯誤屬於何種類型。
- (5) 本研究所開發的驗證系統，不只適用於營建電子採購 XML 文件，只要擁有其標準 Schema 的 XML 文件都可以運用此程式來輔助文件驗證的工作。

5.2 未來工作與建議

本研究所提出的 XML 文件驗證機制，主要是希望把各家廠商不同格式及欄位資料的營建採購電子 XML 文件，透過 12 項營建資訊交換 Schema 標準達到文件格式統一化的目的；再藉著文件驗證系統來幫助檢測文件內容的正確性，以求達到文件標準化的目標。對於未來的工作及建議，主要分為驗證流程與驗證程式兩大部分，以下就各部分詳細闡述：

驗證流程部分：

- (1) 制定出更為嚴謹的驗證機制

本研究所擬定之驗證流程重點是於文件驗證的部份，對於其他的部份如申

請驗證審核資格手續或是驗證費用等等其他相關驗證流程及手續並沒有很詳細的說明，僅僅根據相關文獻建構出大概的驗證流程雛形，提供給想要執行文件驗證的單位一個參考。所以未來可以針對本研究所規劃的驗證機制不合理及執行上不完善之處，加以修改成更符合於現況的機制，並且更為詳細的定義出各個階段所要執行的手續以及內容，給未來要執行驗證文件相關工作的單位能有個更加有說服力的流程機制。

(2) 規劃出一套可經由網路執行 XML 文件驗證工作的驗證機制

本研究所規劃的驗證機制是以現場的方式進行，未來可以根據此驗證機制修改成網路驗證機制模式，使整套文件驗證的工作透過網路完成。規劃網路化驗證機制，要特別注意的就是安全性及公平性的問題，安全性就是要避免驗證系統受到不合法檔案的干擾，例如病毒以及其他人為故意的破壞；至於公平性，也就是要避免作弊的情形發生，讓整個驗證工作更具威信力。發展網路驗證機制是很具有前瞻性的，不但使得文件驗證的工作更為簡單方便，也可以透過網路方式令文件驗證的概念推廣到更多的領域及地方，很值得具有網路相關技術人士對這部份做深入的探討及研究。

驗證程式部分：

(1) 持續擴大驗證系統之功能，使驗證系統能趨於完美

本研究所開發的驗證程式主要的功能只有對於 XML 文件做偵錯的動作，而且有些需與人工智慧等相關技術的功能也無法與本驗證系統整合，以致於無法避免要使用到人工的方式完成某些工作，讓整套驗證系統執行期中稍有麻煩之處。因此未來研究方向是可以開發如何將人工智慧技術應用於驗證系統中，開發出一套真正完全自動化的系統，不必經由人工作業就可以達到驗證文件的目標，使其執行過程更為方便與省時。

(2) 驗證系統網路化

目前的驗證程式為單機版的系統，必須雙方都於驗證地點才能執行驗證的動作；但由於電腦網路化的盛行，許多工作都可以經由網路的傳遞來完成，不但縮短了許多前置作業的時間，也讓工作執行效率提高很多。儘管將程式植基於網路還有許多安全性以及技術性的問題等待克服，但網路化

必然是未來的趨勢，因此驗證程式未來發展的方向可以從這部分著手；例如在安全性的部分，可以發展更新更完善的資料安全加密解密程式，有效的防止惡意攻擊的手段；而在功能性部分，即可以發展一個驗證的網路平台，供廠商登入後直接執行驗證的工作，不但省下許多時間，也提高了驗證執行效率。

(3) 持續更新系統所使用的物件，如 DOM 物件、ParseError 物件等等物件

在程式語言的世界裡，不管是系統物件或者是程式開發的環境，其功能及效用強化的速度是非常迅速的；一直到現在，XML 的標準還在不定時的更新，W3C 也在持續更新許多協定，想必某些目前無法達到的成果，透過未來物件的更新以及新的程式開發環境的出現，可能就可以完成所預期的目標。未來可以多著重於關切物件的發展，版本越新的物件能達到的功能也就越多，而驗證程式可以根據最新發展的物件加以修正，令驗證系統能更臻至完美。



參考文獻

中文文獻

行政院研考會，「文書及檔案管理電腦化作業規範驗證制度」，行政院研考會，民國 89 年 3 月。

李崑堂，「SQL Server 2000 與 XML 應用技術」，學貫行銷股份有限公司，民國 91 年。

吳逸賢、吳目誠，「精采 Visual Basic 6.0 學習範本」，網奕科技，民國 93 年。

何淑媛，「XML 與軟體代理人於電子資料交換之應用研究」，逢甲大學資訊工程學系碩士班，碩士論文，民國 91 年 1 月。

杜仕斌，「Visual Basic 6 範例教本」，學貫行銷股份有限公司，民國 90 年。

吳笠禎，「XML 技術應用於建築管理資訊交換與共享之研究」，國立台灣大學土木工程研究所，碩士論文，民國 91 年 6 月。

吳權威、梁仁楷，「Visual Basic 6.0 入門與應用」，基峯資訊股份有限公司，民國 92 年。

吳明勳，「使用 XML 製作之通用性資料交換模型」，東海大學資訊科學系，碩士論文，民國 89 年 6 月

林錦雲，「利用 XML 驗證之網頁安全防護機制」，國立暨南國際大學資訊管理研究所，碩士論文，民國 92 年 7 月。

梁樾、陳春盛、曾仁杰，「營建產業資訊交換標準 aecXML 推廣計畫(I)」，內政部營建署委託研究計畫，民國 91 年 12 月。

梁樾、陳春盛、曾仁杰，「營建產業資訊交換標準 aecXML 推廣計畫(II)」，內政部營建署委託研究計畫，民國 92 年 12 月。

梁樾、陳春盛、曾仁杰，「營建產業資訊交換標準 aecXML 推廣計畫(III)」，內政部營建署委託研究計畫，民國 93 年 12 月。

梁樾、陳春盛、曾仁杰，「營建產業資訊交換標準 aecXML 推廣計畫(IV)」，內政部營建署委託研究計畫，民國 94 年 12 月。

陳錦輝，「XML 與 ASP 網路實作大全」，金禾資訊股份有限公司，民國 90 年。

郭馨儀，「整合企業應用系統研究-以 XML 為資料交換之基礎」，中華大學工業工程與管理研究所，碩士論文，民國 89 年。

陳佳筠，「XML 在地理資料共享與交換之應用—以人口資料為例」，國立台灣師範大學地理研究所，碩士論文，民國 92 年。

勞虎，「無廢話 XML」，兩隻老虎工作室，民國 88 年。

趙敏宏，「以德菲法制訂 XML 文件之 schema 程序」，國立成功大學資訊管理研究所，碩士論文，民國 92 年 6 月。

謝榮洲，「植基於網路化之 HL7XML 電子病歷系統」，國立成功大學工程研究所，碩士論文，民國 91 年 6 月。

戴慧明，「行政機關公文電子交換關鍵成功因素與效益之研究」，銘傳大學資訊管理學系碩士在職專班，碩士論文，民國 93 年 6 月。

英文文獻

Barillot, E. and Achard, F., "XML: a lingua franca for science," Trends in Biotechnology, Vol. 18, pp. 615-617, 2000

Bryan, M., "An Introduction to the Extensible Markup Language(XML)," Bulletin of the American Society for Information Science, Vol. 25, No. 1, pp. 11-14, 1998

Martin, D., Bribeck, M., Kay, M., Loesgen, B., Pinnock, J., Livingstone, S., Williams, K., Anderson, R., Mohr, S., Baliles, D., Peat, B., Ozu, N., "Professional XML", Wrox Press Ltd, January 2000

Miyazawa, T., Kushida, T., "Advanced Internet XML/EDI Model Base on Secure XML Document," Seventh International Conference on Parallel and Distributed Systems : Workshops, 2000

Phillips, L. A., "Special Edition Using XML," 第三波資訊股份有限公司, 2001

Roy, J., Ramanujan, A., "XML : Data's Universal Language," IT Professional, Vol. 23,

May/June 2000

Roy, J., Ramanujan, A., “XML Schema Language: Taking XML to the Next Level,” IT Professional, Vol. 3, March/April 2001

網路文獻

IAI-aecXML網站，<http://www.iai-na.org/aecxml/mission.php>

Microsoft MSDN online Web Workshop : “Introduction to schema” ,
<http://www.microsoft.com/workshop/>

W3C , “Decryption Transform for XML Signature ” ,
<http://www.w3c.org/TR/xmlenc-decrypt/> , June 2001

W3C , “Document Object Model (DOM) Activity Statement” ,
<http://www.w3.org/DOM/Activity> , October 2001

W3C , “Document Object Model (DOM) Level 1 Specification” ,
<http://www.w3.org/TR/REC-DOM-Level-1/> , October 1998

W3C , ”Markup Validation Service v0.7.2” , <http://validator.w3.org/>

W3C Working Group , ”Extensible Markup Language (XML) 1.0 Recommendation” ,
<http://www.w3.org/TR/1998/REC-xml-19980210.html>

W3C , ”XML Schema” , <http://www.w3.org/XML/Schema>

W3C Candidate Recommendation , World Wide Web Consortium XML Schema Part 1 :
Structures , <http://www.w3.org>

W3C Candidate Recommendation , World Wide Web Consortium *XML Schema Part 2 :
Data types* , <http://www.w3.org>

XML台灣資訊網，<http://www.xml.org.tw/Default.asp>

中華營建資訊標準協會網站，<http://www.aecxml.idv.st/>

中華民國電子商務/資料交換標準委員會網站，<http://www.twtec.org.tw/>

行政院公共工程委員會，計畫網站：

<http://pcstd.pcc.gov.tw/XMLPlan2/portal/home/index.jsp>

