

私立東海大學資訊工程研究所

碩士論文

指導教授：朱正忠 教授

嵌入式系統軟體於需求階段之支援工具

**A Supporting Tool for Embedded System Software in
Requirements Phase**

The logo of the National Central University Library is a circular seal. It features the university's name in English, "NATIONAL CENTRAL UNIVERSITY LIBRARY", around the perimeter. In the center, there are Chinese characters "中央圖書館" (National Central University Library) and a stylized emblem.

研究生：游昭榕

中華民國九十九年六月

摘要

在嵌入式系統開發初期，由於產品的開發時程(Time To Market)將會影響到市場的需求以及系統發展趨勢，在軟體開發生命週期中各階段之間的溝通對於整個發展時程影響很大，建置各階段完善的溝通機制，將能加速軟體的開發。而需求階段將軟體之規格限制、使用者需求等相關資訊規劃完成，並依據此資訊繪畫出相對應的 SysML 與 UML 圖形，此系統其餘開發人員時常會需要參照內容，軟體開發週期各階段與需求階段溝通將更為重要。因此提出了嵌入式系統軟體於需求階段之支援工具，建置一中央管理資料庫，讓開發人員在需求階段時所產出的相關資料(如需求樣版、SysML 與 UML 圖形等)利用友善的使用者介面將資訊上傳，並設計存取、下載與搜尋等使用者介面以利開發者使用，藉此整合資訊並加速開發，讓嵌入式系統軟體開發更加迅速與順利。

關鍵詞：嵌入式系統、使用者介面、資料庫、SysML。

Abstract

At the beginning of the embedded system development, product development time (Time to Market) will affect the market requirement and the trend of system development. In Software life cycle, the communication between every various phase has a great impact on the whole development process. Improve construct the communication mechanism between phases will be able to accelerate software development. The communication between every phases in software development cycle and requirement phase is very important. Developers design, development or testing refer to documents which produce by planning constraints on software specifications, user requirements and other relevant information at the requirement phase. This paper presents the supporting tool for embedded system software development in requirement phase. Build a centralized database, that developers can upload the information via friendly user interface (such as requirement patterns, SysML and UML graphics, etc.).Design the interface of new, modify, delete and search, then developers can facilitate use the user interface, to integrate information and let the-embedded systems development process more quickly and smoothly.

Keywords: Embedded System, UI-Driven, Design Pattern, Database, SysML

目錄

摘要.....	I
Abstract.....	II
圖目錄.....	V
表目錄.....	VII
第 1 章 導論.....	1
1.1 前言.....	1
1.2 研究動機.....	2
1.3 研究目的.....	3
1.4 章節安排.....	4
第 2 章 背景知識與相關研究.....	5
2.1 軟體工程.....	5
2.1.1 簡介.....	5
2.1.2 軟體生命週期.....	7
2.1.3 物件導向.....	7
2.2 UML 與 SysML.....	9
2.3 使用者介面.....	10
2.4 延伸標記語言.....	11
2.5 XQuery.....	12
2.6 XML 資料庫.....	13
2.6.1 關聯式資料庫管理系統.....	13
2.6.2 原生式 XML 資料庫.....	13
2.7 VERTAF/Multi-Core	14
第 3 章 研究方法.....	16
3.1 軟體生命週期需求階段產出分析.....	16
3.2 系統概況.....	18
3.2.1 系統架構.....	18
3.2.2 使用者流程與情節分析.....	19
3.3 資料庫設計.....	24
第 4 章 案例研究.....	30
4.1 案例背景.....	30
4.2 案例介紹.....	31
4.2.1 登入畫面.....	31
4.2.2 建立新專案.....	32
4.2.3 建立新圖形資訊.....	34
4.2.4 修改與查詢專案敘述.....	37
4.2.5 修改圖形資訊.....	39

4.2.6	搜尋圖形資訊.....	39
4.2.7	刪除圖形資訊.....	42
第 5 章	結論與未來工作.....	45
參考文獻	46
附錄	50

圖目錄

圖 1、軟體專案開發成功率.....	5
圖 2、軟體工程定義.....	6
圖 3、軟體發展生命週期 (SDLC).....	7
圖 4、XML 與 XQuery 發展的過程概略.....	12
圖 5、XML 與關聯式資料庫間的轉換.....	13
圖 6、VMC 各子計畫之關係圖	15
圖 7、需求圖擴充原型.....	17
圖 8、SysML Diagram Types	17
圖 9、系統架構圖.....	19
圖 10、Use Case Diagram.....	20
圖 11、IM 系統操作流程圖	24
圖 12、XML Tag 標籤階層關係.....	27
圖 13、XML 標籤階層示意範例.....	28
圖 14、利用 XQuery 語法於資料庫中搜尋 XML 文件.....	28
圖 15、以日期排序的條件式.....	29
圖 16、以檔案名稱排序的條件式.....	29
圖 17、DVR 系統架構圖	31
圖 18、登入畫面.....	32
圖 19、VMC 子計畫二於本論文的部份	32
圖 20、登入畫面選擇 New Project.....	33
圖 21、建立新專案.....	33
圖 22、輸入專案名稱與描述.....	33
圖 23、登入畫面選擇 Open Exist Project	34
圖 24、於 IM Tool 視窗選取專案與開發階段	35
圖 25、新增圖形資訊表格.....	35
圖 26、選取圖形類別.....	36
圖 27、填寫新增圖形資訊.....	36
圖 28、於列表顯示新增的圖形資訊.....	37
圖 29、修改專案敘述.....	38
圖 30、進入修改專案描述視窗.....	38
圖 31、修改圖形資訊後的列表.....	39
圖 32、選取專案與開發階段即可顯示相關圖形.....	40
圖 33、以檔案名稱排列表單.....	41
圖 34、顯示 XML 資訊以利使用者查看.....	41
圖 35、顯示 SysML 圖形.....	42

圖 36、選取欲刪除圖形並刪除.....	43
圖 37、確認是否刪除.....	43
圖 38、刪除後的 IM Tool 視窗	44

表目錄

表 1、建立新專案之 Use Case Description.....	20
表 2、搜尋之 Use Case Description.....	21
表 3、新增之 Use Case Description.....	21
表 4、修改之 Use Case Description.....	22
表 5、刪除之 Use Case Description.....	23
表 6、XML 標籤列表.....	25

第1章 導論

1.1 前言

軟體工程之相關議題，從軟體生命週期的各個階段衍生出來，包括需求分析、系統設計，到軟體開發、驗證、測試，直到最後的維護。由於軟體在本質上具有高度的彈性，當不同公司與不同團隊採用各別的方法、Models 及 Data Schema 描述方式，便會造成進行軟體開發(Development)、整合(Integration)、再利用(Reuse)時的極大困難，最後導致軟體整體開發時的耗費人工、低效率、高價格等不利現象不斷出現。在嵌入式系統軟體方面，除了以上一般軟體開發時可能出現的問題，另外還有其他特性在需求階段時即需納入考慮。在軟體生命週期中，需求階段為所有開發的開端，開發者於需求階段集合使用者的限制與需求，將其資訊轉換為 UML 與 SysML 的圖形以供其他階段開發者使用，故軟體生命週期各階段與需求階段的資訊流通是很頻繁的。

嵌入式系統為一種電腦軟體與硬體的綜合體，對於軟硬體間有「量身定做」的原則，其於某一種特殊用途上，針對這項用途開發出獨特性的整合系統，而根據統計，大部份的嵌入式系統應用在通訊、工業、醫療、軍工航太與其他(消費性電子、車用)等領域。

嵌入式軟體的開發面對愈來愈多樣化的應用，透過新一代作業系統推出，也可能會有機會創造出不一樣產業應用型態，在此趨勢下，開發者必須完全掌握嵌入式系統整合能力與掌握應用導向，此項挑戰比過去大的多。開發嵌入式系統軟體的所面臨的挑戰增加，嵌入式系統開發時需考慮的因素比開發一般軟體來的多，根據 Tech Insights 的嵌入式系統設計報告(Embedded Systems Design reports)所進行的「2008 年嵌入式市場研究報告」(Embedded designers on tighter schedules, juggling multiple projects in 2008)指出，嵌入式系統開發人員在 2008 年面臨了許多新專案開發的壓力，因此 50% 以上的開發專案沒有如期完成，由此可見，嵌入式系統軟體開發在時程上有很大的改進空間。

1.2 研究動機

嵌入式系統軟體開發的複雜度日益增加，當不同公司與不同團隊採用各別的方法、Models 及 Data Schema 描述方式，便會造成進行軟體開發(Development)、整合(Integration)、再利用(Reuse)時的極大困難，而嵌入式系統軟體開發時需考慮比一般軟體更多的特性與限制，困難度又大大增加，導致在開發時程上往往無法如期完成。因此在開發生命週期中各階段之間的溝通議題非常的重要，各階段間資訊傳遞的便利性與速度為開發者所關注，而各階段與需求階段的傳遞尤為重要，因為需求階段為生命週期的起頭，結合了所有使用者的需求與限制。如果能減少資訊傳遞的時間將能降低軟體開發時程延宕的可能性，而減少傳遞時間的方法不只是增加傳遞速度，建立讓開發者能方便操作的工具也是方法之一。

當我們將開發中需求階段傳輸的資料格式定義為 XML 檔案，以能提高整合性，各開發者將其完成或半完成的檔案以 XML 的格式儲存至統一的資料庫供下一階段開發者使用，甚至在任何時間，某一開發者欲參考需求階段的資訊，也可至統一資料庫拿取所需要的資料。因此，各開發者與資料庫之間的存取溝通將影響著軟體開發的流暢度。我們開發需求階段所需要的使用者介面，提供開發者更便利的方法取得所需要的資訊。

1.3 研究目的

對於一個嵌入式系統軟體的開發通常需要團隊合作(Team work)，面對不同開發人員與不同開發環境，資訊整合與一致性非常重要。針對嵌入式系統軟體在開發時碰到許多困難與挑戰導致開發時程經常無法如期完成，需求階段與其他各階段的溝通速度與便利性占了很大的因素，為此設計了於需求階段的一個有善的使用者介面為支援工具以利開發者使用。

建立一個有善的使用者介面讓開發者能的尋找欲參考的相關資訊，例如其餘階段開發者想查看 Use Case Diagram，即可透過此介面選擇需求階段進入欲搜尋的專案名稱裡進行檔案搜尋，搜尋完成後將相關資訊顯示給此開發者以利其參考，因此可減少不同開發者溝通的問題與等待傳輸的時間。藉由此方法提升各階段間溝通的效率與正確性，減少因溝通不良所浪費的時間成本與人力成本，提高嵌入式系統軟體開發如期完成的可能性。

1.4 章節安排

論文的章節安排分述如下：

第二章 背景知識與相關研究，此章節說明與本論文相關的研究，並介紹使用者介面之觀念，以及物件導向的基本特性及其相關的應用。

第三章 此章說明本方法提出系統流程、使用者操作流程分析與資料庫設計。

第四章 案例研究，本章說明利用本論文設計使用者介面之應用並實際操作。

第五章 結論及未來方向，此章將本論文之研究成果做總結，並提出未來研究之方向。

第2章 背景知識與相關研究

2.1 軟體工程(Software Engineering)

2.1.1 簡介

隨著電腦科技的快速發展，軟體與硬體產業愈來愈發達。目前全世界幾乎所有的國家都必須依賴複雜的電化系統來處理所有的事務，並且也有愈來愈多的商品結合了電腦的軟硬體技術。隨著科技的進步，人們對於電腦科技的需求愈來愈多。一套完整的資訊系統，軟體花費的成本比硬體高很多。目前軟體系統的需求量愈來愈大，複雜度與異質性也愈來愈高，若開發軟體並無使用標準化的開發方法或是開發統程不夠完善，則會導致開發時間延遲、成本提高、執行效率不佳與難以維護等問題。

根據美國一間著名的市場研究機構 Standish Group 對軟體專案團隊進行大規模的調查，在 2006 年的報告中(如圖 1)，只有 35%的軟體專案能符合預算且準時交付軟體，有 19%的專案則因為程式偏離需求或是無法解決問題，導致在專案完成之前就被取消，佔最大比例(46%)的情況為軟體雖順利交付，但卻超出預算或時間或是只完成某部份的功能與特性。雖然 Standish Group 所調查的單位是美國的軟體開發組織，但目前台灣的軟體開發也是相同的情況，延遲交貨、品質不良或不符合客戶需求等。

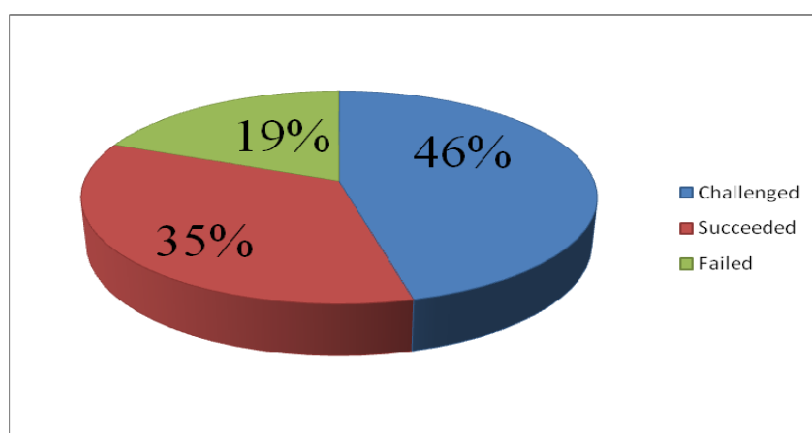


圖 1、軟體專案開發成功率

為了解決上述軟體系統開發所遭遇的困難，北大西洋公約組織(North Atlantic Treaty Organization, NATO)招集了將近 50 名計算機科學家、程式設計師與工業界巨頭於 1968 年在德國與辦了首次的軟體工程學術會議。提出了以「軟體工程」來規範軟體開發時所需的專業知識，同時也建議了軟體開發應該是以類似工程的活動來進行。

軟體工程主要包含了軟體專案管理與軟體開發技術這兩方面的內容(如圖 2)而這二方面的專業領域則來自於管理科學與資訊科學[1]。

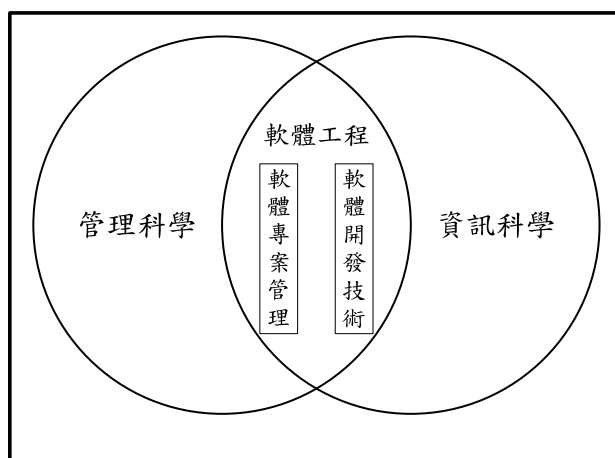


圖 2、軟體工程定義

在 1993 年美國 IEEE 的計算機協會和 ACM 組成了軟工程協調委員會 (Software Engineering Constituent Committee, SWECC)，並在 2004 年 6 月完成了軟體工程知識(Software Engineering Body of Knowledge, SWEBOK)2004 版全文。SWEBOK 完整的描述出實踐軟體工程所需具備的核心知識如下：

- 軟體需求(Software Requirement)
- 軟體設計(Software Design)
- 軟體建構(Software Construction)
- 軟體測試(Software Test)
- 軟體維護與更新(Software Maintenance)
- 軟體配置管理(Software Configuration Management, SCM)
- 軟體工程管理(Software Engineering Management)
- 軟體開發流程(Software Development Process)
- 軟體工程工具與方法(Computer-Aided Software Engineering, CASE)
- 軟體品質(Software Quality)

2.1.2 軟體生命週期(Software Life Cycle)

軟體開發工程師必須組合出一個包含過程、方法及工具層次的開發策略。這樣的策略經常被稱為軟體發展生命週期模型(Software Development Life Cycle Model, SDLC) [2] [3]，如圖 3。

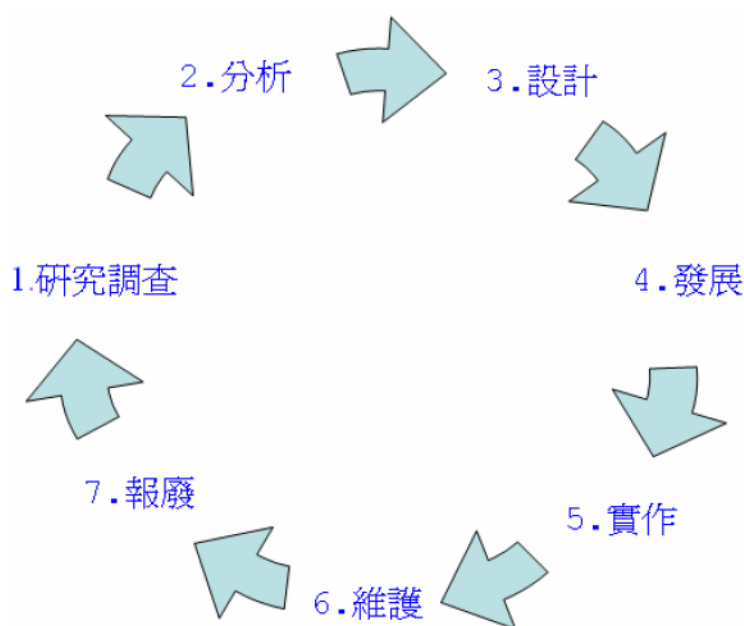


圖 3、軟體發展生命週期 (SDLC)

2.1.3 物件導向(Object-oriented, OO)

物件導向是軟體工程領域的一項重要的技術，物件導向的觀點在於將現實世界人、事、物等許多概念的靜態資料及動態行為表示成為物件，而這些概念之間的互動也會對應至物件之間的交互關係[1]。一個物件往往被定義成為一個角色，擁有各自的特徵與行為，而所謂的商業模型(Business Model)就是將某個領域的資訊定義成個別的物件。利用物件導向技術可以使軟體的開發更快、成本更低、及得到較高的品質。物件導向分析與設計(OOA/D)基礎於物件導向的概念成為了軟體開發方法的主流。現代的軟體開發觀點，許多都是透過物件導向的觀念來進行分析、設計。而所謂的物件導向應該具備下列的特性[4][5]：

1. 物件(Object) 以及訊息(Message)
2. 繼承(Inheritance)

3. 封裝(Encapsulation)
4. 動態連結(Dynamic Binding)

利用物件導向所建製的軟體系統，它們的主要構成元素，就是物件以及類別。Coad 與 Yourdon 在 1990 年時，將物件做以下的定義：「物件—問題領域中某些事物的抽象化，可以反應保留系統資訊的能力、與系統互動的能力以及兩者兼顧。」類別，則是針對某一組共通的物件所做的一種描述。每個物件都應具有以下三點特性：

1. 識別名稱(Identity)：用以和其他的物件作區分
2. 狀態(State)：描述物件特性的屬性資料
3. 行為(Behavior)：提供給外界使用的服務或操作等三部份

而在一個使用物件導向的軟體開發過程中，主要可以分為以下三個階段：

1. 物件導向分析(Object-Oriented Analysis, OOA)：OOA 主要是在需求階段，從需求文件的字彙裡，透過物件以及類別來對問題領域(Problem Domain)進行塑模(Model)。
2. 物件導向設計(Object-Oriented Design, OOD)：OOD 的目的是為了讓開發者能夠了解系統的架構。在 OOD 的過程中，會導致物件導向分解(Object-Oriented Decomposition)。而 OOD 主要是使用邏輯(Logical)以及實體(Physical)的標記法來表示系統的靜態(Static)及動態(Dynamic)的觀點，並且會將系統之間的物件以及彼此之間的關係描述出來。
3. 物件導向程式設計(Object-Oriented Programming, OOP)：OOP 是指在軟體發展過程中，將 OOA、OOD 的思想進行表達和實現。

在物件導向模型的研究中，最令人矚目的其中一項，就是如何把物件導向模型與模糊邏輯結合。在李允中教授的相關研究中[9][10][11]，提出了如何使用模糊邏輯來制定(Formulate)、獲得(Capture)並且分析(Analyze)一些不精確的需求規格文件(Imprecise Requirement)，並且利用 XML 來標記 Fuzzy 物件，提供轉換規

則，分別將模糊物件模型(Fuzzy Object Model)以及模糊物件規格書(Fuzzy Object Specification)轉為 XML Schema 以及 XML 檔。

2.2 UML 與 SysML

Unified Modeling Language (UML) [15]是用來將一個軟體系統描述成視覺化軟體模型的標準表示法。UML 提供了許多既定的圖示用來描述軟體開發過程不同階段的抽象概念，例如使用案例圖(Use case diagram)、類別圖(Class diagram)、溝通圖(Communication diagram)等等。利用這些圖示法可以讓使用者很容易的瞭解軟體的分析與設計過程。也因此 UML 已成為物件導向軟體開發的一項重要的技術。但 UML 的使用往往過於彈性而缺乏嚴謹，導致需求分析與系統設計的模型容易產生不完整及不一致性[16]。然而不只 UML，包含所有現有的模型建置技術(modeling techniques)都必須要有一個正規化的方式以達成軟體開發與維護過程中軟體模型的高度整合性。

OMG SysML 是 SysML(Systems Modeling Language)是一種新的系統塑模語言，是為系統工程應用開發的標準塑模語言，SysML 是一個一般性圖形塑模語言用來分析、設計和驗證包括軟、硬體、資訊、人員以及設備之複雜系統。尤其是，SysML 提供了圖形化的表示方式來塑模系統需求、行為、結構和參數，使它可以用來與其他工程分析模型結合。SysML 沿用了部分 UML 2.0 的子集，並增加了一些拓展。SysML 技術的設計是為了提供簡單但功能強大的構造建模範圍廣泛的系統工程問題。

SysML 包含了的九種基本圖形：類別圖、裝配圖、使用者案例圖、需求圖、參數圖、順序圖、活動圖、狀態圖和時間圖，雖然 SysML 是 OMG 2007 年 9 月才正式 release 的標準，但目前已經有許多研究開始運用 SysML 進行系統塑模，如 Bock[17]利用 SysML 以及 UML 2.0 來塑模系統需求，Johnson et.al. [18]利用 SysML 來描述連續系統的動態行為。也有部分研究利用 SysML 來作為 SoC 設計的塑模方法，如 Vanderperren 與 Dehaene[48]。市面上也有相關的軟體應用 SysML 與 MDA 來輔助嵌入式系統開發，如 Rhapsody，或是輔助需求開發，如 ARTiSAN Studio[19]，然而相關的開放源碼軟體仍不多見。

2.3 使用者介面(User Interface, UI)

使用者介面是系統和使用者之間進行互動和資訊交換的媒介，它實作資訊的內部形式與人類可以接受形式之間的轉換[1]。

使用者介面是介於使用者與硬體而設計彼此之間互動溝通相關軟體，目的在使得使用者能夠方便有效率地去操作硬體以達成雙向之互動，完成所希望藉助硬體完成之工作，使用者介面定義廣泛，包含了人機互動與圖形使用者介面，凡參與人類與機械的資訊交流的領域都存在著使用者介面[12]。

使用者和系統之間一般用面向問題的受限自然語言進行互動。目前有系統開始利用多媒體技術開發新一代的使用者介面。以下為組成要素[13][14]：

- (1) 基本要素：必須有人類感官與作用體所產生輸出、輸入與運作內容三方互動設計。
- (2) 使用者介面使得使用者與系統之間雙向的資訊傳遞成為可能：
 - i. 輸入：允許使用者操作執行一套系統。
 - ii. 輸出：允許系統向使用者傳達操作效果。
- (3) 人性要素 (Human Factor)：人類感官知覺、心理情緒、認知、學習、記憶、反應、以及處理資訊的模式、個別背景之差異等等每一項都和用戶介面有密切的關係，直接或間接地影響使用者介面的效能。
- (4) 輔助與說明：適當的說明檔案，不管是線上的或是傳統的檔案。讓使用者明瞭系統究竟可以做與如何來達成工作。並讓使用者能夠花最少的時間在介面的熟悉上，而將時間投注於完成其預定之工作項目。最簡單的輔助說明就是圖形或符號。

2.4 延伸標記語言(eXtensible Markup Language, XML)

目前網站上最流行的 HTML 語言提供了在 Internet 環境下可展示資料交換的功能，但是由於其固定標籤(tag)的設計，只提供了在網路作業中資訊展現的功能，而無法提供說明資訊內容的能力，而限制了資料在 Internet 網站上被再運用及應用系統方面的發展。基於此需求，W3C (World Web Consortium)於 1998 年初公佈了 XML (eXtensible Markup Language) 的標準，提供了一個可自訂標示名稱功能的標記語言(Markup Language)，使用者可以很容易的運用 XML 在網路上自行製作所需要的文件內容及格式，並廣泛的發展不同層次的應用[6]。XML 不但具有標記語言的特性外，也可以視為是中介語言(meta-language)，因為它可以讓建置者自行定義語法規則，亦即透過定義的延伸，指定標籤的語法，而不是端視標籤本身的定義[7]。

此外，因為具有 meta-language 的特性，XML 也被常用於描述各種 Domain，如應用在數學公式上、化學方程式上以及其他的專業上，也都有倡議之中的標準格式。在美國一個宣傳和促進在網際網路上面做電子交易的 CommerceNet 組織，就召集了一些業界的成員來制訂電子交易專用的 DTD，以支援 XML 的 EDI 文件格式[8]。XML 具有擴展性 (Extensibility)、結構性 (Structure)、描述性 (Description)、確認性 (Validation) 等特性，其最初設計的目的便是希望成為全球資訊網上能具有描述性且又是可以交換結構性資料的標準格式。因此，XML 可讓整個產業、學術界、專業機構發展個別的文件格式定義，以標準化的方式呈現文件內的資訊[9]。

XML 的主要優點之一，就是它能為一個文件建立起結構。每一個 XML 文件都包含了邏輯結構和實體結構，邏輯結構在敘述以何種順序包含那一些元素；而實體結構則為文件中的真實資料[10]。XML 的資料架構可讓使用者依不同的方式來檢視資料、按不同的法則排序、或瀏覽時固定顯示些資訊讓文件使用更有效率如點選某個程式軟體的聯結、便只秀出符合該使用者視窗版本的資訊。此外，XML 也可應用在 Intranet 或 Extranet 像是以資料庫為主的網站、依某些格式顯示大量的資料、XML 有很好的解決方案。而以 Extranet 的觀點來看、企業可透過 XML 瀏覽器將資訊傳達給使用者、整個產業更可依循某特定的 XML 標

準、將資訊完整呈現。在存取異質資料庫方面 XML 也優於 HTML，資料可以一致的方式呈現[11]。

2.5 XQuery

為了制定 XML 的查詢方式，因此 W3C 於 2000 年的 1 月首先提出了 XML Query Requirements，討論對 XML 資料做查詢時的需求[27][28]。於 2001 年 6 月 W3C 以 Jonathan Robie (Software AG)、Don Chamberlin (IBM Almaden Research Center) 及 Daniela Florescu (INRIA) 三人共同開發的 Quilt 語言為基礎定義 XQuery，此後 XQuery 便為 XML 查詢語言的標準，目前最新版本已於 2006 年 6 月 8 日出版，發展的過程概略如圖 4 所示。

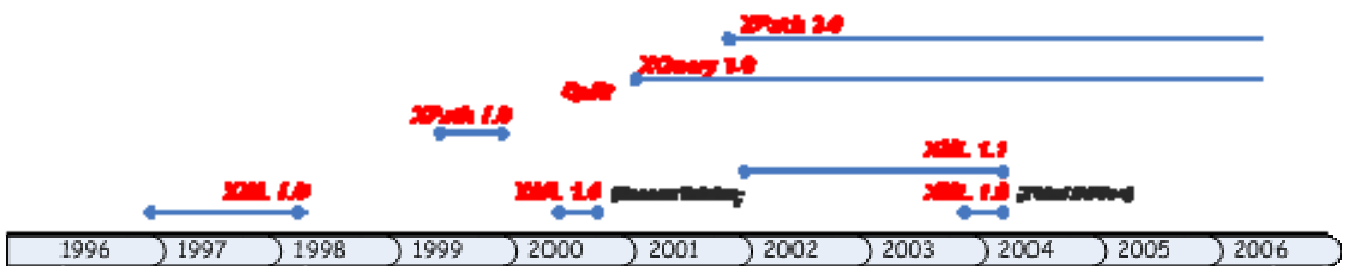


圖 4、XML 與 XQuery 發展的過程概略

XQuery 是針對各類 XML 資料所設計，不論其儲存為文件或是在關聯式資料庫中。XQuery 其資料模型(data model)視一個 XML 文件為一棵標記樹(label tree)，並考慮標記樹中每個節點及所有值之順序關係。其特性是路徑表示法的使用，使用者可以利用路徑表示法在 XML 資料中尋找任意長度路徑的資料[29]。因此透過 XQuery 的執行，便可取得 XML 文件中符合路徑表示法的資料序列，其順序是依照文件中各個元素之間的順序關係而定[30][31]。另外使用者也可以從資料中取得關於 XML 文件 schema 的資訊。

2.6 XML 資料庫(Database)

2.6.1 關聯式資料庫管理系統(Relationship Manager Data

Base System, RDBMS)

關聯式資料庫透過轉換機制支援以 XML 為輸出入格式[20][21]，因為轉換的負擔使效能下降導致系統開發時間拉長，管理成本也增加。而各關聯式資料庫供應商的 XML 轉換機制是各自建立。其優點為可透過 SQL 指令查詢，不必額外訓練學習。但缺點為關聯式資料庫搜尋能力受限於 XML 與 SQL 指令的對映限制且擴充性差，關連式資料庫無法處理複雜的 XML 階層/巢狀結構[22][23]。

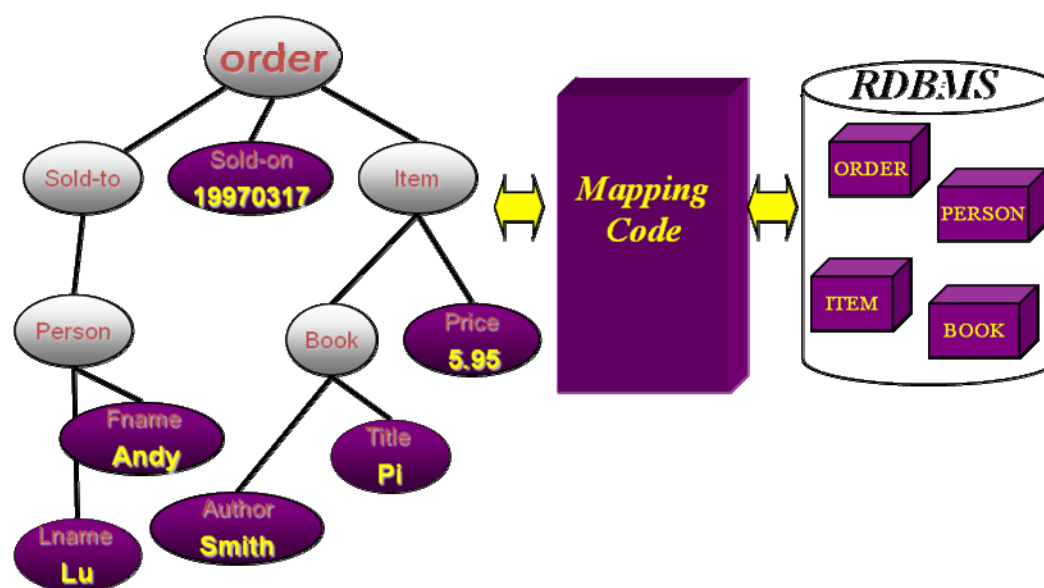


圖 5、XML 與關聯式資料庫間的轉換

2.6.2 原生式 XML 資料庫(Native XML DataBase)

近年來興起一種專為儲存 XML 文件的原生型 XML 資料庫。它是以 XML 文件為基本儲存單位，儲存時不需將文件拆解而儲存於多個表格，或預先設定儲存欄位的資料型態。因此，可以容易的存取整份 XML 文件，並且不會改變其原本的資料結構與內容[24][25]。除上述優點外，並提供 XQuery 與 XPath 等操作方式、

全文檢索(full-text searching)、自動索引(automatic indexing)、Java 程式語言整合(Java binding)等機制，可容易的進行程式的開發與應用。由於數位典藏領域之後設資料相當複雜，且資料需求經常改變，因此，以整份 XML 文件為一個基本的儲存單位，對於更改後資料規格的架構及內容有較高的彈性。而 XML 資料庫系統對 XML 文件的處理機制較完整，所以我們以原生型 XML 資料庫系統儲存 XML 文件[26]。

Sedna 是一套原生式 XML 資料庫，具有自動索引(Automatic Indexing)、全文搜尋(Full-Text Search)及支援 XQuery 查詢等功能。Sedna 儲存 XML 以 XML 文件為最小的儲存單位，使用者不需預先匯入 XML 規範就可以存入 Sedna。Sedna 支援 Java、C 及 C++，本論文中以 Java 實現並可使用一些已經開發好的 Java API，如 XML:DB 及 XQJ。開發人員可使用 XML:DB 及 XQJ 提供的方法從 Sedna 取得特定的 XML 文件及其內容，並對資料進行查詢、新增、修改或刪除等動作。

2.7 VERTAF/Multi-Core (VMC)

VERTAF/Multi-Core (VMC) 為一針對多核心嵌入式架構之整合性軟體開發環境，開發者可藉由本環境以 SysML 描述其系統需求，以 UML 標準符號塑模其系統設計，並將該設計套用樣式結構自動轉換為更高品質的設計，透過優良的設計模型自動產生程式碼，最後並可對程式碼加以測試[32]。因此，本系統具下列優點：(1) 支援多核架構嵌入式軟體之設計與驗證；(2) 提供一個整合性的開發環境；(3) 因本環境乃基於 UML 標準所建構，所以易與目前既有之 UML 開發工具整合；(4) 支援既有之多核架構開發函式庫，如：OpenMP 以及 Intel Threading Building Blocks。

此整合計畫一共區分為七個子計畫，含括系統開發生命週期之需求、設計、實作、測試與維護階段，各子計畫間的關係如圖 6 所示，總計畫包含七項子計畫：子計畫一以 SysML 為基礎之多核心嵌入式系統需求塑模環境，子計畫二支援多核心嵌入式軟體設計之多重觀點整合模型與可再用元件庫，子計畫三多核心嵌入式軟體設計之設計支援系統，子計畫四多核心嵌入式軟體之合成與程式碼生成，子計畫五多核心嵌入式軟體設計工具系統之架構支援，子計畫六多核心嵌入

式軟體設計工具系統之平行程式優化支援，以及子計畫七多核心嵌入式軟體設計之測試支援系統。

本論文是由朱正忠教授所領導的 Integration and Maintenance 階段「VMC_MUM」的 Modeling Traceability & Integration Management 及 Unified Model Description 兩個部份，其研究方法及研究案例會在第三章與第四章中詳述之。

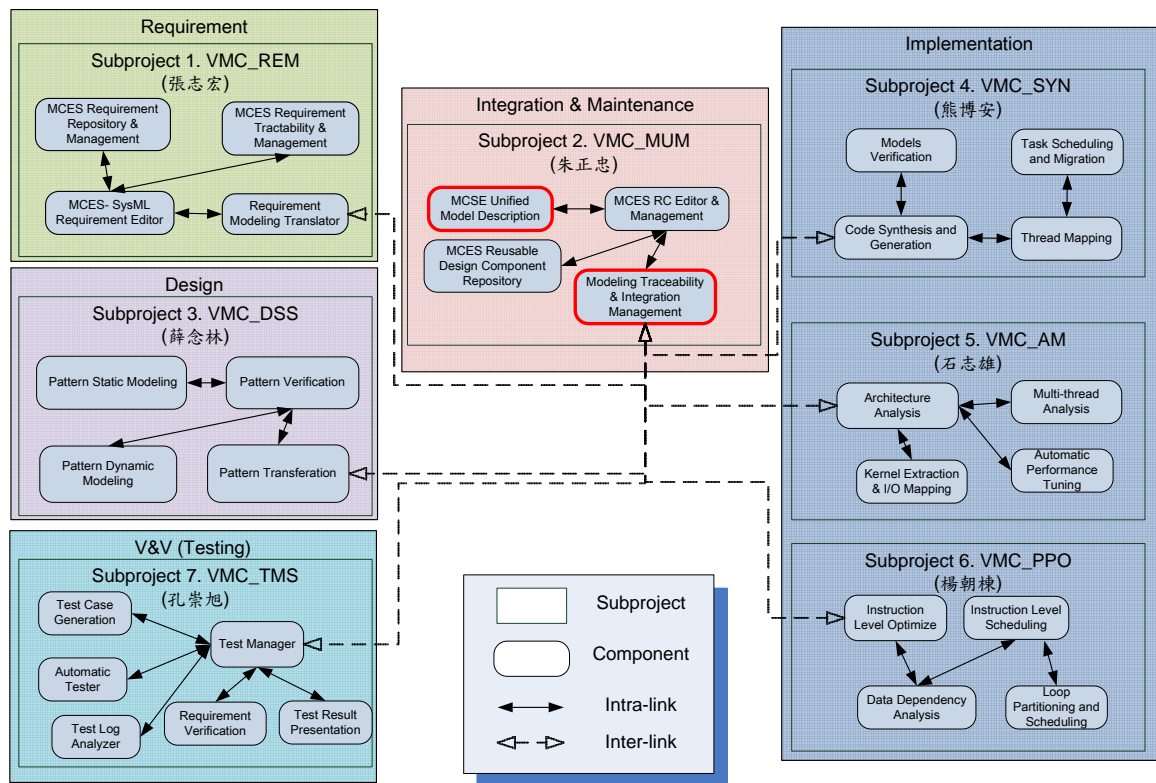


圖 6、VMC 各子計畫之關係圖

第3章 研究方法

本論文中為開發嵌入式系統軟體於需求階段的支援工具，為了完成此系統我們必須先了解在嵌入式系統軟體於需求階段時產出的檔案與其他使用者想要得到什麼資訊，藉由以上資訊我們將可進行下一步，分析系統架構與與流程並設計系統的使用者介面(User Interface)與分析操作流程，最後針對所接收到的資訊分析設計資料庫。

3.1 軟體生命週期需求階段產出分析

在需求階段開發者以 SysML 為基礎的模型驅動(Model-Driven)需求塑模流程，開發者利用 Profile 來制定一般化的需求樣版，讓 SysML 的圖形附帶有需求的相關資訊，系統使用者定義為專案經理、系統分析師等軟體開發相關人員，目的在於幫助使用者建立完整的需求，以及導引使用者輸入正確的需求文件內容，並且將加入需求量化(Requirement Quantization)的概念，量化的單位能減少使用者在建立需求的時候，出現需求描述不精確的情形，也降低需求塑模階段因為客戶所提供的需求規格模糊不清而造成開發成本與時間的浪費，需求階段最終目的是希望能讓使用者在操作的圖形化需求塑模工具將需求圖繪製完成之後，所產出 XML 檔案內容中就能包含撰寫系統需求規格書(System Requirement Specification Document) 所需要的部份需求資訊。

SysML 的需求圖有助於組織使用者的需求，並且能夠有效地表現出不同需求間的關係。需求圖可以顯現出需求與需求之間的階層(hierarchy)，讓系統設計師清楚地看出需求階層關係與系統架構；還有數種表示方法來找出需求間的關係，包括 derive (衍生)、copy(複製)、satisfy(滿足)、verify(驗證)、refine(提煉)、trace(追蹤)等，通常應用於比較大型且複雜系統方面。SysML 可以應用表示各種領域的模型元素，靠著需求圖與視覺化的關係。事實上，SysML 的需求圖為 UML 類別圖(class diagram)的擴充原型(stereotype)，圖 7 為 SysML 圖形的形態，說明

SysML 以 UML 2.0 為基礎做延伸發展，包含 UML 2.0 的圖形應用，並且透過修改或新增圖形定義來擴展 SysML 的功能面。因此在需求階段的開發者將會產出 SysML 的圖形以供其他階段開發者參考，其圖形為 Actively Diagram, Sequence Diagram, State Machine Diagram, Block Definition Diagram, Internal Block Diagram, Package Diagram, Requirement Diagram, Parametric Diagram, Use Case Diagram 共九種圖形(圖 8)。

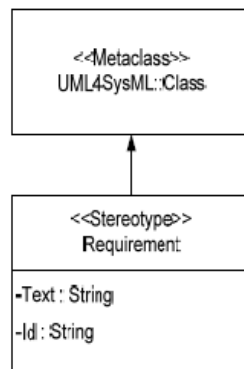


圖 7、需求圖擴充原型

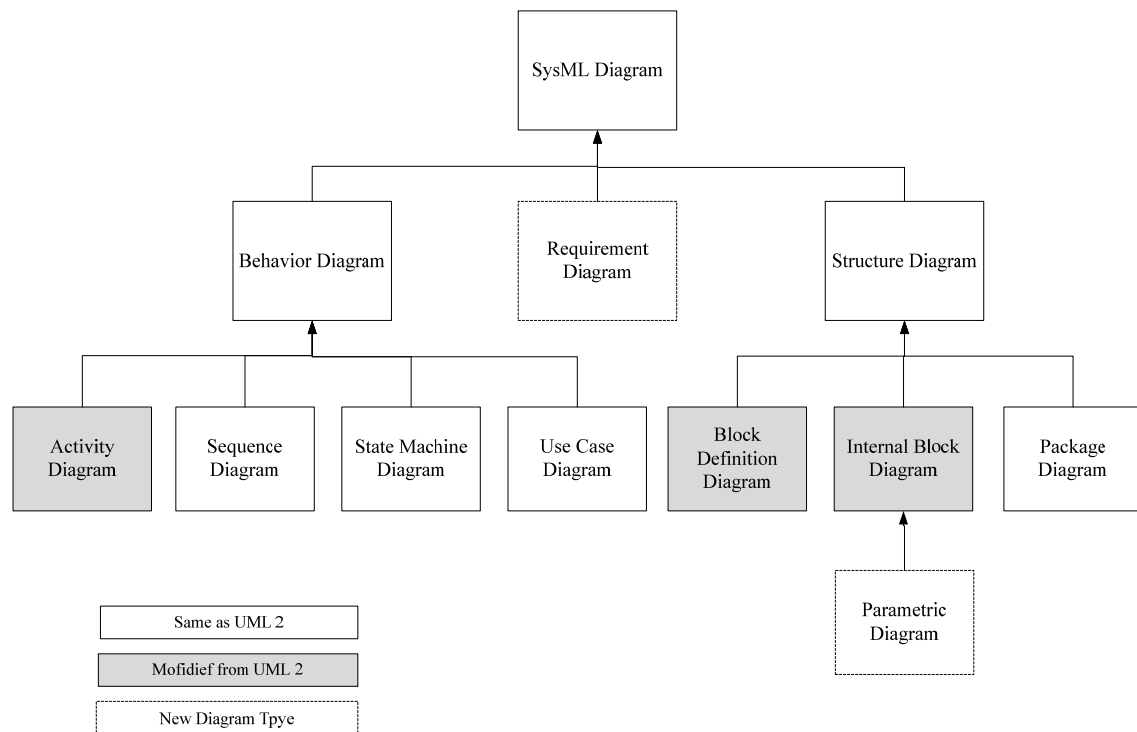


圖 8、SysML Diagram Types

對於嵌入式系統軟體之需求階段有一個很重要的地方為需求描述文件應當對於硬體需求規格加強描述，並建立相關資訊以便在後續階段引導程式設計師引入適當函式庫或類別庫，符合多核架構系統設計需求，以便其他階段及早確認各階段開發與設計是否有符合硬體環境及使用者需求。因此需求階段與其他各階段的溝通將非常重要，在本論文中，開發嵌入式系統軟體於需求階段的支援工具，以提高與需求階段的溝通速度並銜接平順增加使用者便利性。

3.2 系統概況

在本論文中，我們提出一個嵌入式系統軟體於需求階段的支援工具，並做了兩個部份進行分析與設計：第一：我們對於此系統設計與分析後的結果提出一個架構圖，並針對此架構圖說明系統運作狀況。

第二為分析使用者要求的服務為何，亦為他們對系統的需求，深入清楚了解需求後便能制定使用者對於系統的操作流程，利用使用案例圖(Use Case Diagram)、使用案例描述(Use Case Description)與操作流程圖來說明使用者的動作。

3.2.1 系統架構

在此流程中說明了系統架構，流程圖分成三個層面(如圖 9)，首先為使用者介面(User Interface)，我們依據需求階段的產出檔案與流程分析出使用者需求來設計有一致性(Consistency)與可用性(Usability)的介面，減少使用者對於操作介面的學習時間與能確保滿足使用者的所有需求。在第一層之中為使用者介面所提供的系統應用功能，本系統提供 Add、Modify、Delete 與 Search 之功能，第二層為將透過使用者介面收集到的資訊進行分類(Classify)，包含專案名稱、檔案名稱等，可依資料格式與類型分成 Project Document, File Document 與 SysML Document 三部份，各部份將儲存不同的資訊：

1. Project Document：由使用者介面中接收到有關專案的資訊皆會儲存至此部

- 份，如專案名稱、階段選擇與專案管理者等資訊。
2. File Document：由使用者上傳的檔案類型與名稱即儲存於此部份。
 3. SysML Document：由使用者上傳的 XML 檔案內容與相關資訊即儲存於此部份。

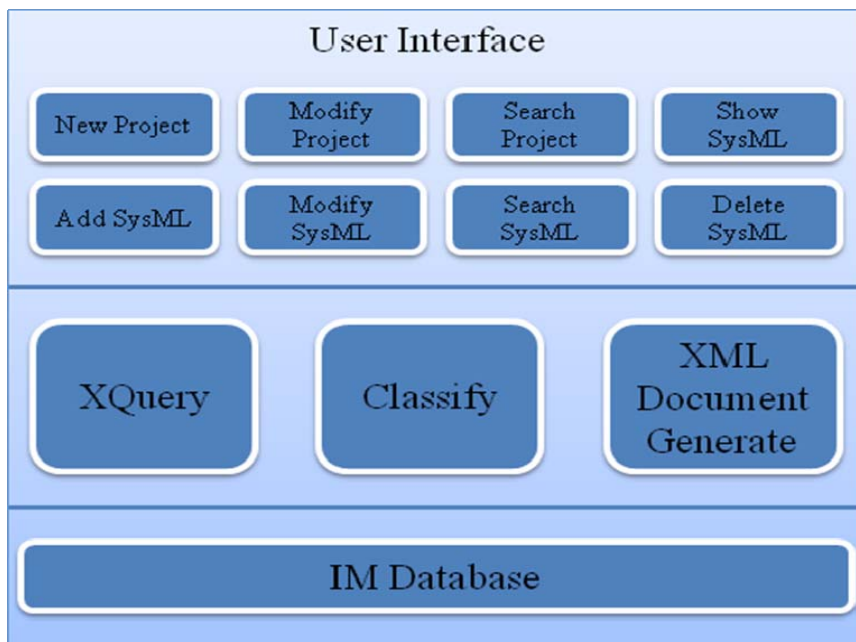


圖 9、系統架構圖

資料分類為 Project Document, File Document 與 SysML Document 三部份後，依照本論文所設計之 XML Tag(如表 6)進行儲存，最後將資料整合為一個 XML 檔案儲存至系統架構中第三層的 IM 資料庫。第二部份為透過使用者介面，利用搜尋(Search)直接進入資料庫取得 XML 檔案。資料庫之設計內容於 3.3 會詳細說明。

3.2.2 使用者流程與情節分析

不管是一般軟體或嵌入式軟體，軟體開發週期依舊為需求、設計、實作、測試與維護。在開發任何系統前都必須先清楚了解使用者的需求，此階段常常被使用者用來評估系統是否滿意的重要因素之一。而流程分析的主要目的在於釐清使用者所要求的服務、系統對於使用者應有什麼樣的回應，藉此了解使用者的需求

讓系統的設計能更完整。Use Case Diagram 能清楚描述出使用者(user)對系統的需求，本研究根據需求階段所產出的資訊與檔案與使用者對系統的需求分析出 Use Case Diagram(如圖 10)，主要需求為 New Project 與 List，其中 List 又包含了 Search, Add, Modify 與 Delete 四項。

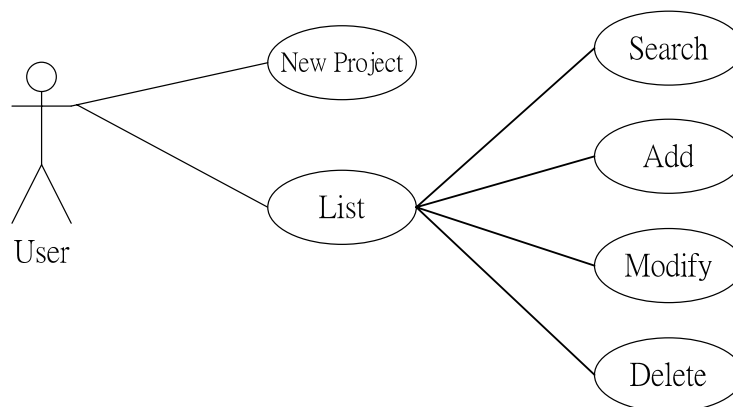


圖 10、Use Case Diagram

Use Case Diagram 雖然可以很明確的定義系統要做的事，但卻無法看到系統如何完成 Use Case 的細節以及使用者與系統之間的互動過程。因此，在流程分析中，還必須要撰寫使用者案例描述(Use Case Description)，詳細的描述使用者與系統間的行為，本研究分析了各階段開發者對於需求階段產出資訊的需求與系統對於使用者的需求所產生的回應描述大約例舉至下表 1 至表 5。

表 1、建立新專案之 Use Case Description

Name ：New Project	
Actors ：使用者	
Goals ：建立新專案	
Actor actions	System responses
1. 使用者按下 "New Project" 要求新增專案。 3. 使用者填入新專案名稱與專案描述等。	2. 系統彈跳 "New Project" 視窗。 4. 系統將資料儲存於資料庫。

表 2、搜尋之 Use Case Description

Name ：搜尋	
Actors ：使用者	
Goals ：搜尋XML檔案並展示	
Actor actions	System responses
<p>1.使用者按下”Open Exist Project”要求。</p> <p>3.使用者於下拉式選單選擇階段與專案名稱。</p> <p>5.使用者點選列表中的圖形。</p> <p>7.使用者可於XML檔案中搜尋資料。</p>	<p>2.系統彈跳”IM Tool”視窗。</p> <p>4.系統依使用者所選擇之階段與專案名稱至資料庫搜尋所有圖形並將圖形列表列出。</p> <p>6.系統將使用者所點選的圖形XML檔案於視窗中顯示。</p>

表 3、新增之 Use Case Description

Name ：新增。	
Actors ：使用者	
Goals ：新增新的檔案	
Actor actions	System responses
<p>1.使用者按下” Open Exist Project”要求。</p> <p>3.使用者於下拉式選單選擇階段與專案名稱。</p> <p>5.使用者點選列表中空白</p>	<p>2.系統彈跳”IM Tool”視窗。</p> <p>4.系統依使用者所選擇之階段與專案名稱至資料庫搜尋所有圖形並將圖形列表列出。</p>

<p>處連點兩下。</p> <p>7. 使用者於 "Add or Modify" 視窗填寫圖形資訊並上傳圖形xml檔案。</p>	<p>6. 系統單跳出 "Add or Modify" 視窗。</p> <p>8. 系統將資訊增加至資料庫並顯示於圖形列表。</p>
--	--

表 4、修改之 Use Case Description

Name ：修改	
Actors ：使用者	
Goals ：修改檔案說明或內容	
Actor actions	System responses
<p>1. 使用者按下 "Open" 要求。</p> <p>3. 使用者於下拉式選單選擇階段與專案名稱。</p> <p>5. 使用者點選列表中圖形名稱處連點兩下。</p> <p>7. 使用者於 "Add or Modify" 視窗修改圖形資訊或上傳圖形xml檔案。</p>	<p>2. 系統彈跳 "IM Tool" 視窗。</p> <p>4. 系統依使用者所選擇之階段與專案名稱至資料庫搜尋所有圖形並將圖形列表列出。</p> <p>6. 系統單跳出 "Add or Modify" 視窗。</p> <p>8. 系統將更新資訊增加至資料庫並顯示於圖形列表。</p>

表 5、刪除之 Use Case Description

Name ：刪除	
Actors ：使用者	
Goals ：刪除檔案	
Actor actions	System responses
1.使用者按下” Open Exist Project”要求。	2.系統彈跳”IM Tool”視窗。
3.使用者於下拉式選單選擇階段與專案名稱。	4.系統依使用者所選擇之階段與專案名稱至資料庫搜尋所有圖形並將圖形列表列出。
5.使用者點選列表中的圖形。	6.系統將使用者所點選的圖形XML檔案於視窗中顯示。
7.使用者按下”Delete”鈕。	8.系統將資料於資料庫中刪除。

根據以上 Use Case Diagram 與 Use Case Description 的分析，我們將使用者對系統的操作流程統整為操作流程图，如圖 11 左上起始點進入系統選擇”New Project”即可建立新專案或點選”Open Exist Project”的功能會彈跳出列表視窗，並且能對資料庫內的資訊進行 Search, Add, Modify 與 Delete 四項動作，使用者介面會引導使用者填寫與修改資訊以達到使用者的需求，最後離開系統。

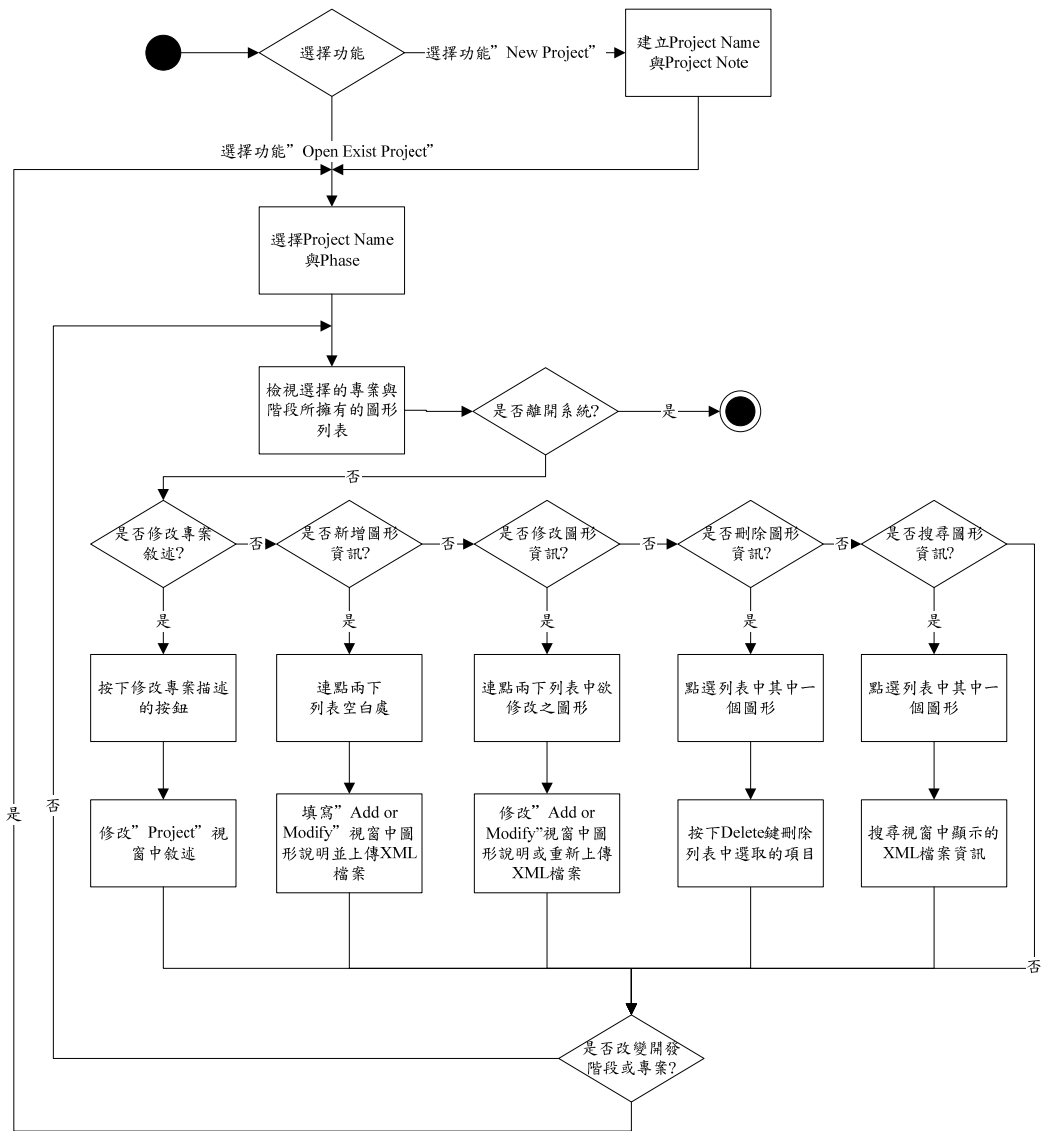


圖 11、IM 系統操作流程圖

3.3 資料庫設計

對一個資訊系統而言資料庫是不可或缺的部份，面對於不同型態的資訊選擇的資料庫也不同。對於儲存 XML 的資料庫大至上可分成關聯式資料庫管理系統 (Relationship Manager Data Base System, RDBMS) 與原生式 XML 資料庫 (Native XML Database)，關聯式資料庫透過轉換機制支援以 XML 為輸出入格式，但因為轉換的負擔使效能下降導致系統開發時間拉長，管理成本也增加。原生型 XML 資料庫以 XML 文件為基本儲存單位，儲存時不需將文件拆解而儲存於多個表格，或預先設定儲存欄位的資料型態，因此可以容易的存取整份 XML 文件，並

且不會改變其原本的資料結構與內容。根據以上的分析，在此論文中選擇了原生式資料庫-Sedna 來做為此系統的資料庫，。

藉由使用者透過本論文中設計的使用者介面輸入在需求階段所產出以XML格式呈現的SysML九種圖形以及其說明與相關資訊，我們接收到許多XML檔案與資訊，為了整合所有資訊並以統一的方式儲存，本論文制定了XML標籤(Tag)將接收到的資訊進行統一儲存。

表 6、XML 標籤列表

Part	XML Element	XML Attribute	XML Tag
Project Document	Null	Project name	<project_name pname="project name">
	Null	Phase	<dev_phase phase="phase">
	Project Manager	Null	<project_manager>
	Number	Null	<number>
	Start Date	Null	<start_date>
	Finish Date	Null	<finish_date>
	Project note	Null	<project_note>
File Document	Null	File type	<file_type type=" file type ">
	Null	File name	<file_name fname="file name">
SysML	UML address	Null	<uml_address>
	Di2 address	Null	<di2_address>

Part	XML Element	XML Attribute	XML Tag
Document	Version	Null	<version>
	SysML uml	Null	<sysml_uml>
	SysML di2	Null	<sysml_di2>
	SysML note	Null	<sysml_note>
	Date	Null	<date>

每個XML標籤會對應使用者介面的欄位，當使用者依照介面的設計輸入或選擇資訊時，系統會將資訊對應XML標籤儲存。以下說明各XML標籤的定義：

<project_name fname="project name">：儲存使用者輸入或選擇的專案名稱。

<dev_phase phase="phase">：儲存使用者所選擇的階段。

<project_manager>：儲存專案管理者名稱。

<number>：儲存專案參與人數。

<start_date>：儲存專案執行啟始日期。

<finish_date>：儲存專案執行預計完成日期。

<project_note>：儲存使用者輸入的專案描述。

<file_type type="file type">：儲存使用者入之SysML圖形類別。

<file_name fname="file name">：儲存使用者輸入的以XML描述的SysML圖形檔案名稱。

<uml_address>：儲存使用者選取的以XML描述的SysML圖形檔案位址。

<di2_address>：儲存使用者選取的以XML描述的SysML圖形檔案位址。

<version>：儲存使用者輸入的以XML描述的SysML圖形檔案版本。

<sysml_uml>：儲存以XML格式描述的SysML圖形資訊。

<sysml_di2>：儲存以XML格式描述的SysML圖形格式。

<sysml_note>：儲存使用者輸入的SysML圖形描述。

<date>：儲存檔案上傳時間，以利版本控制。

圖 12為XML標籤分層概念圖，一個專案即會產生一顆分層概念樹，最上層為Project Document儲存案專名稱與開發階段，中間層則儲存File Document，其內容為檔案圖形類別與檔案名稱，最下層為SysML Document儲存XML檔案中各項相關資訊，如：位址、版本等。圖 13為依照XML標籤分層概念的示意範例，並定義其schema如附錄一。

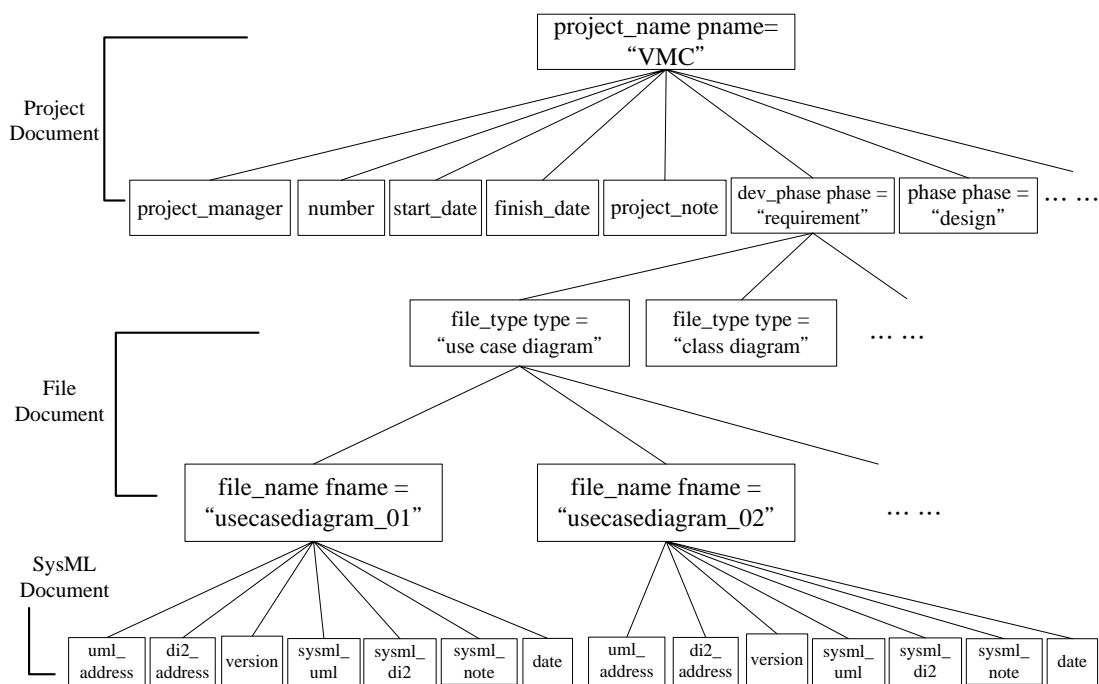


圖 12、XML Tag 標籤階層關係

```

<project_name pname=" DVR " >
  <project_manager>Chao-Jung Yu</project_manager>
  <number>20</number>
  <start_date>2010/06/21</start_date>
  <finish_date>2011/06/21</finish_date>
  <project_note>此為DVR專案中的描述 </project_note>
  <dev_phase phase=" requirement " >
    <file_type type=" Use Case Diagram " >
      <file_name fname=" usecasediagram_01 " >
        <uml_address>D:\ Workspace\ usecasediagram_01.uml </uml_address>
        <di2_address> D:\ Workspace\ usecasediagram_01.di2</di2_address>
        <version> 1.0 </version>
        <sysml_uml>
          <?xml version="1.0" encoding="UTF-8"?>
          <uml:Model xmi:version="2.1" xmlns:xmi="http://schema.omg.org/spec/XMI/2.1"
          <packageImport xmi:id="_6iX18MlLEd6ahKv0LzAHGQ">
            <importedPackage xmi:type="uml:Model" href="pathmap://UML_LIBRARIES/UM"
          </packageImport>
          <packagedElement xmi:type="uml:Collaboration" xmi:id=" EyBggMlMEd6ahKv0LzA"
          <ownedBehavior xmi:type="uml:Interaction" xmi:id=" EyKqcm lMEd6ahKv0LzAHGQ"
          <lifeline xmi:id=" _vev6sM lMEd6ahKv0LzAHGQ" name="DVR Status Info" coveredB
        </sysml_uml>
        <sysml_di2>
          <di2:Diagram xmi:version="2.0" isVisible="true" fontFamily="Arial" lineSty
          <contained xsi:type="di2:GraphNode" isVisible="true" fontFamily="Arial"
          <property key="STEREOTYPE_DISPLAY_LOCATION" value="Comment"/>
          <property key="QUALIFIED_NAME_DEPTH" value="0"/>
          <contained xsi:type="di2:GraphNode" isVisible="true" fontFamily="Arial" li
          <property key="STEREOTYPE_DISPLAY_LOCATION" value="Comment"/>
          <property key="QUALIFIED_NAME_DEPTH" value="0"/>
          <contained xsi:type="di2:GraphNode" isVisible="true" fontFamily="Arial" li
        </sysml_di2>
        <sysml_note>此為使用者案例圖 </sysml_note>
        <date>2010/06/08</date>
      </file_name>
    </file_type>
  </dev_phase>
</project_name>

```

圖 13、XML 標籤階層示意範例

雖然本論文為需求階段的支援工具，但設計了選擇階段的標籤(<phase>)，因未來可將此支援工具擴展至軟體生命週期的所有階段，故增加此功能。當資訊對應XML標籤儲存完畢後，需求階段開發者或是其他各階段開發者欲查詢圖形資訊，即可進入本系統選擇階段與專案名稱，系統將會透過XQuery進入資料庫中依照使用者的選擇對應XML標籤搜尋出所有圖形與其相關內容並顯示出來。藉由此方法不儘可以統一需求階段資訊並可減少各開發者的搜尋時間，以下圖為使用XQuery語法搜尋於VMC.xml檔案中的VMC專案，於需求階段的檔名為usecasediagram_01的使用者案例圖，將其XML檔案顯示於使用者介面。

```

doc("VMC.xml")/project_name[@pname='VMC']/dev_phase[@phase='requirement']/file_type[@type='Use Case Diagram']/file_name[@fname='usecasediagram_01']/sysml_uml

```

圖 14、利用 XQuery 語法於資料庫中搜尋 XML 文件

在進行搜尋圖形資訊時，圖形表單的排列順序將會影響使用者搜尋時間，故可加入某些條件來加速搜尋，例如以日期排序，點選使用者介面表單上的"date"，我們便在XQuery語法中輸入如圖 15的條件式即可以日期排序，藉由此方法使用

者可輕易尋找出最新修改後的圖形。亦或是使用檔案名稱進行排序，條件式如圖 16，便可將相同圖形排列在一起方便使用者尋找。

```
order by $dev_phase/file_type/file_name/date
```

圖 15、以日期排序的條件式

```
order by $dev_phase/file_type/file_name
```

圖 16、以檔案名稱排序的條件式

第4章 案例研究

4.1 案例背景

在本節中，我們將針對 DVR(Digital Video Recording)系統中的 Video Stream Server(VSS)進行需求塑模的流程作為本篇研究的研究案例，來說明基於 SysML 的需求塑模流程，DVR(Digital video recording)是一個即時錄影的多媒體系統，可以提供遠端使用者即時觀看或者是收看已錄影的存檔影片，是由多部攝影機與資料庫結合建構而成的系統，DVR 系統主要分為三大部份，包括：Remote Monitor Client (RMC)、 Video Streaming Server (VSS)及 Parallel Video Encoding (PVE)。

此案例中我們將會接收開發者於需求階段所產生的九種 SysML 圖形，其圖形為 Activity Diagram, Sequence Diagram, State Machine Diagram, Block Definition Diagram, Internal Block Diagram, Package Diagram, Requirement Diagram, Parametric Diagram, Use Case Diagram，並讓開發者能依循使用者界面的指示輸入圖形相關資訊並將所有資訊儲存於資料庫中，專案其他階段開發者皆能透過使用者介面進入此系統進行搜尋。圖 17 為 DVR 系統架構圖。

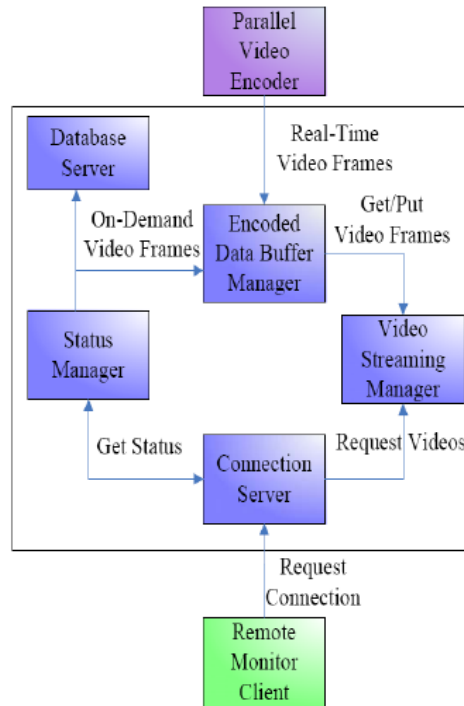


圖 17、DVR 系統架構圖

4.2 案例介紹

於本節中，我們將實作上一章的研究方法，並實際操作後截取圖形進行說明。以下將說明建立新專案、建立新圖形資訊、修改圖形資訊、修改專案敘述、搜尋圖形資訊與刪除圖形資訊六項功能。

4.2.1 登入畫面

在進入 Integration Management Tool(IM Tool)一開始必須先登入系統，輸入使用者姓名、連絡信箱與密碼後選擇系統功能圖 18。本系統為一針對多核心嵌入式架構之整合性軟體開發環境之子計畫二_支援多核心嵌入式軟體設計之多重觀點整合模型與可再用元件庫的部份，如 2.7 節所示，其系統分為 RCR Editor 與 IM Tool 兩部份，本論文系統集中於 IM Tool 的部份，使用者輸入基本資料並選擇欲執行的動作 New Project 或 Open Exist Project 即可進入系統。

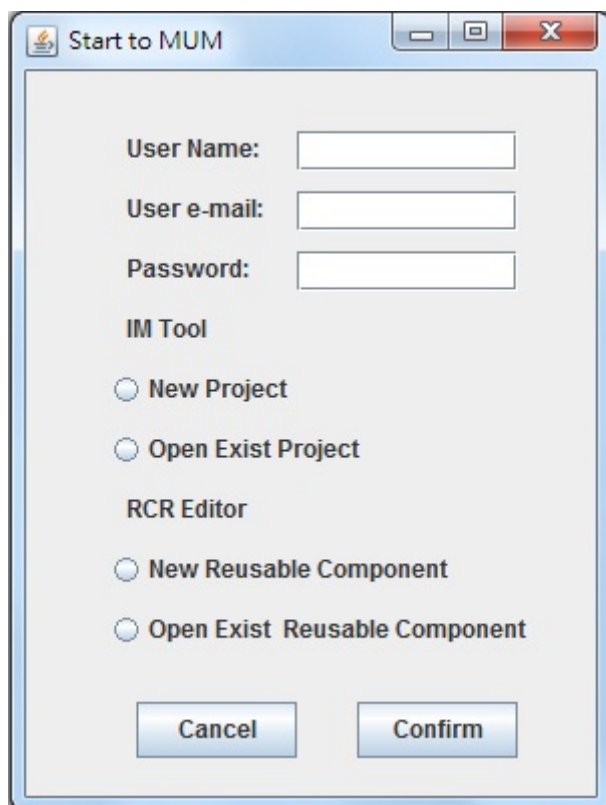


圖 18、登入畫面

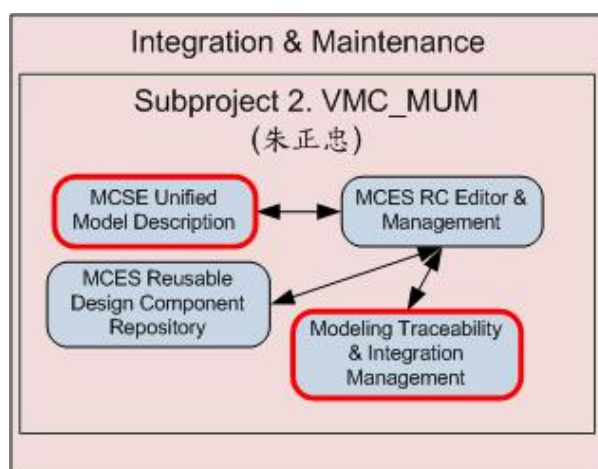


圖 19、VMC 子計畫二於本論文的部份

4.2.2 建立新專案

如欲開立一個新的專案需於登入畫面選擇 New Project(如圖 20)，系統會彈跳出一 New Project 視窗(如圖 21)，使用者輸入專案名稱與專案敘述後按下”OK”完成新增(如圖 22)。

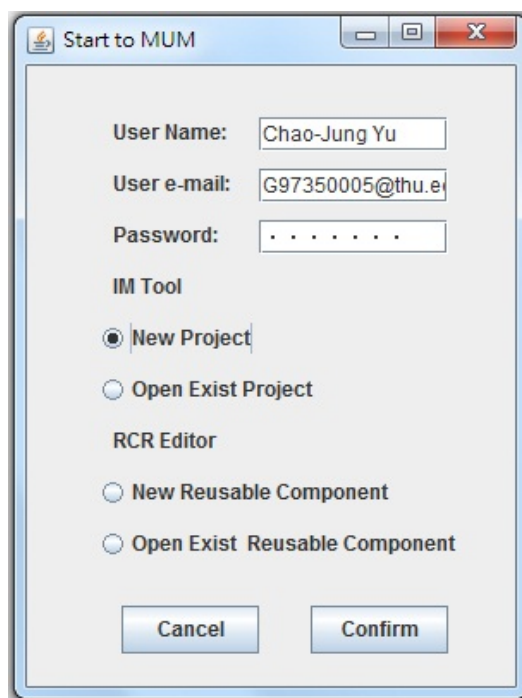


圖 20、登入畫面選擇 New Project

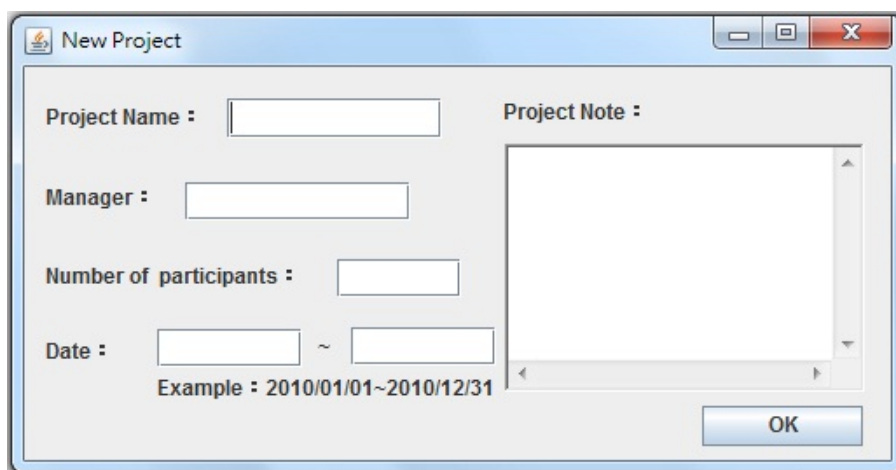


圖 21、建立新專案

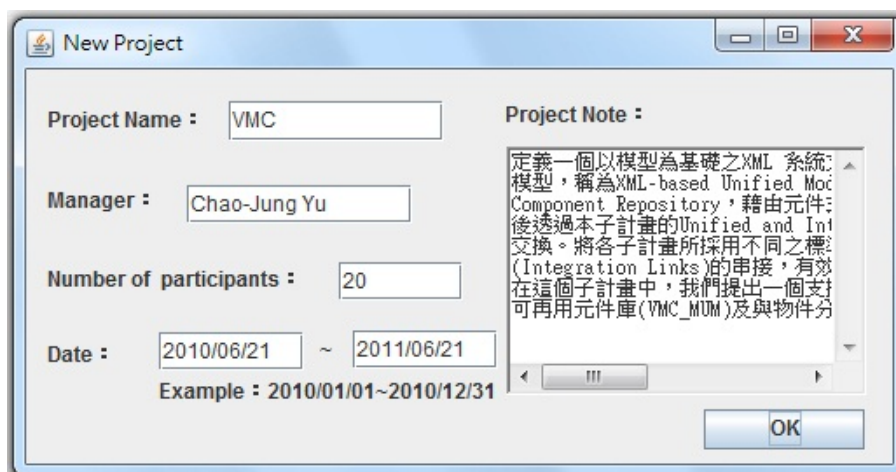


圖 22、輸入專案名稱與描述

4.2.3 建立新圖形資訊

使用者於啟始登入畫面點選 Open Exist Project 進入 IM Tool(如圖 23)，於視窗上方選取正確的專案名稱與開發階段(如圖 24)，並點選下方列表空白處即可進入新增畫面(Add or Modify)(如圖 25)。於 Add or Modify 視窗上填入圖形相關資訊，如圖形類別(如圖 26)、圖形名稱、版本與備註等，並選取檔案上傳位置(如圖 27)。輸入完畢案下”Finish”按鍵即可完成新增，IM Tool 視窗列表上則增加了剛剛輸入的圖形資訊(如圖 28)。

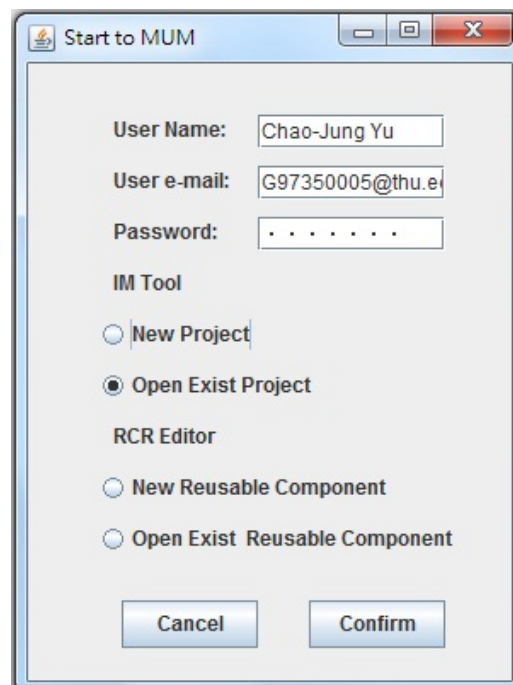


圖 23、登入畫面選擇 Open Exist Project

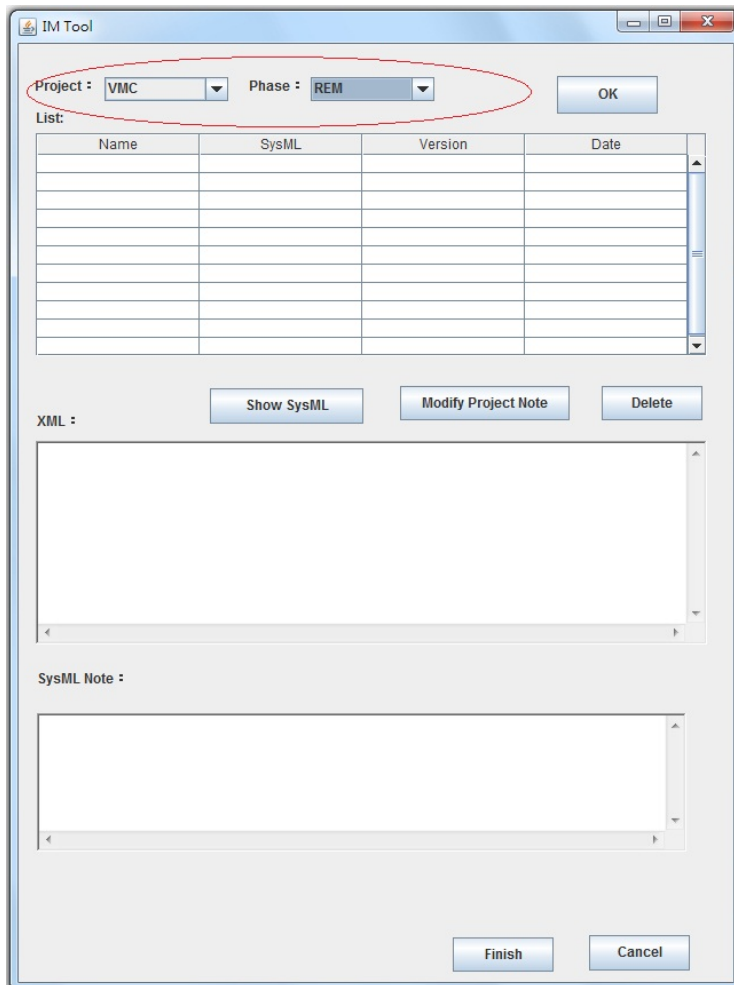


圖 24、於 IM Tool 視窗選取專案與開發階段

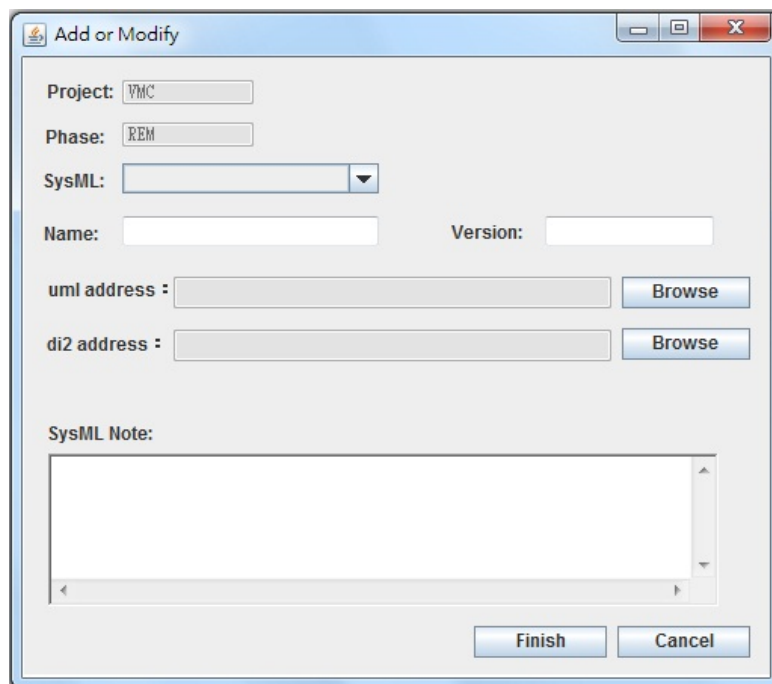


圖 25、新增圖形資訊表格

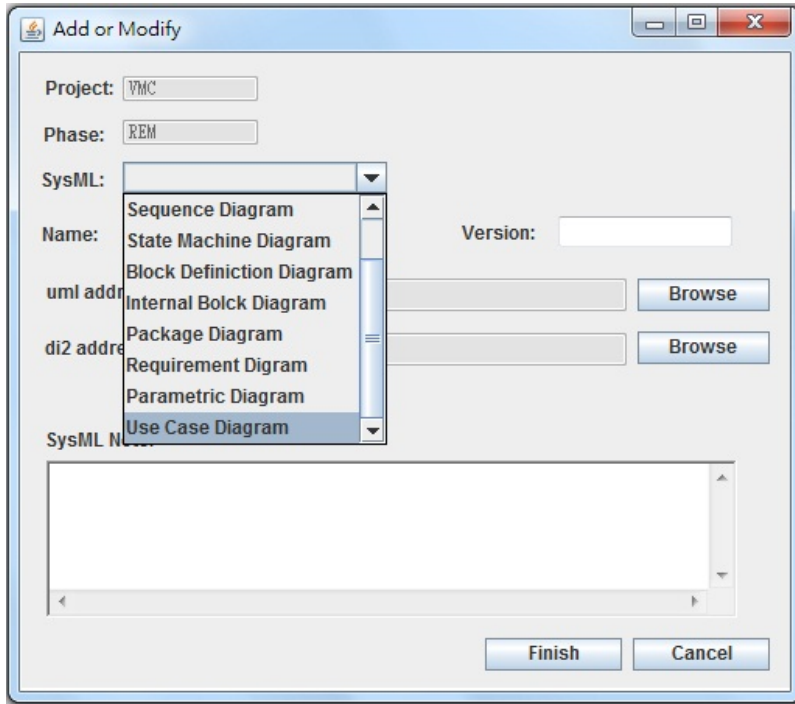


圖 26、選取圖形類別

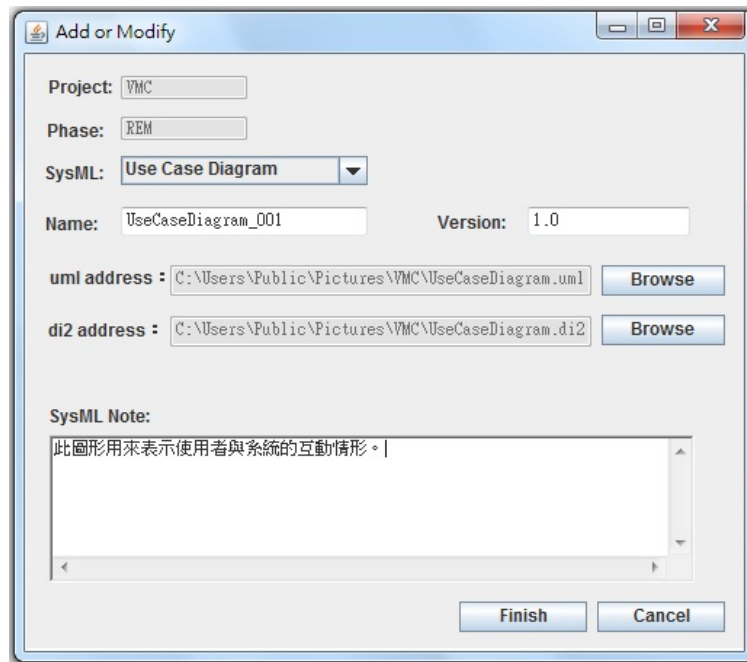


圖 27、填寫新增圖形資訊

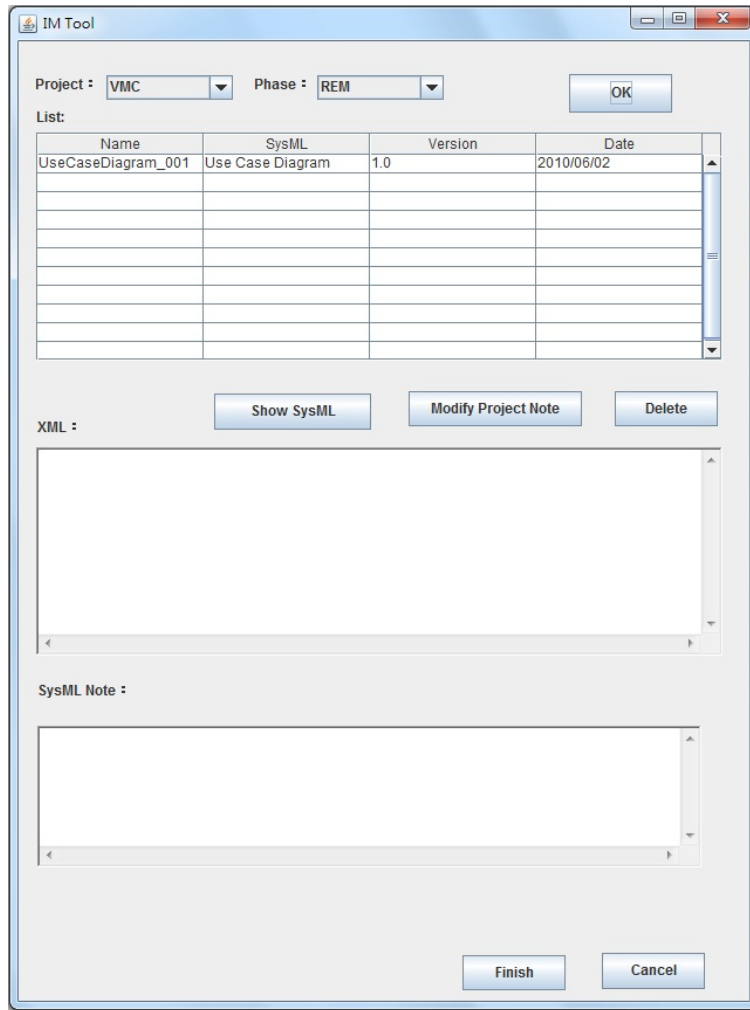


圖 28、於列表顯示新增的圖形資訊

4.2.4 修改與查詢專案敘述

IM Tool 視窗中於下拉式選單中選取欲修改或查詢的專案敘述並點選”Modify Project Note”按鍵(如圖 29)即可進入此專案敘述的修改視窗(如圖 30)。

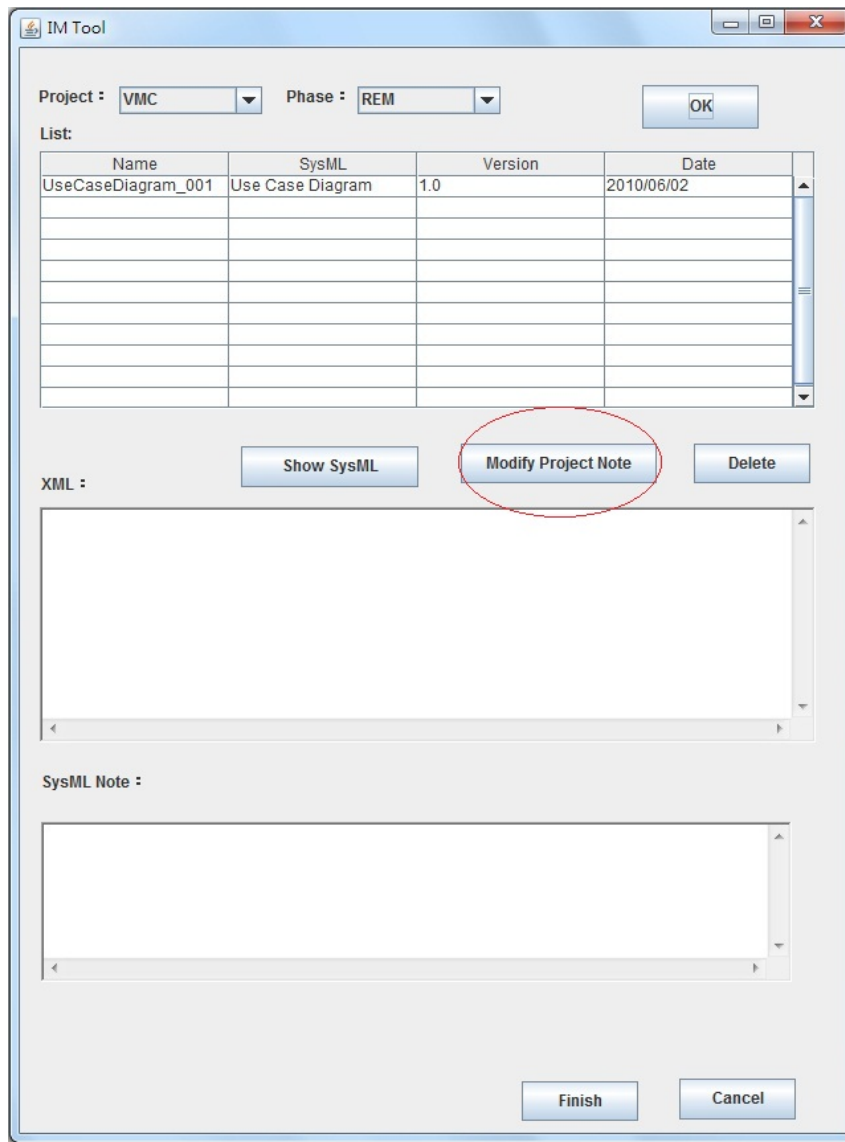


圖 29、修改專案敘述



圖 30、進入修改專案描述視窗

4.2.5 修改圖形資訊

於 IM Tool 視窗中列表上點選欲修改的圖形，系統即彈跳出 Add or Modify 視窗讓使用者進行修改或是重新上傳，完成後系統將最新資訊顯示至列表上，如圖 31 原本使用者案例圖版本為 1.0 經過修改後為版本 1.5，日期的部份也有做更新。

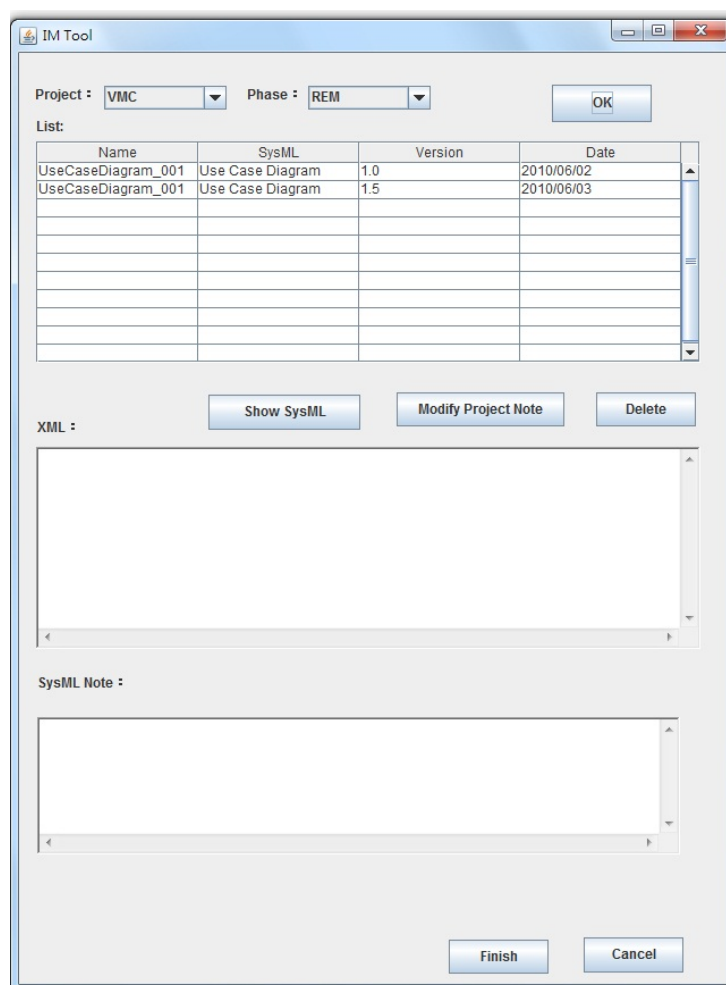


圖 31、修改圖形資訊後的列表

4.2.6 搜尋圖形資訊

選取專案名稱與開發階段，列表中會顯示此專案與此開發階段所擁有的圖形與其資訊(如圖 32)，並可依名稱、圖形或時間來進行排序搜尋，圖 33 以檔案名

稱將表單排序。選取欲了解的圖形，IM Tool 視窗下半部即出現 XML 文件供使用者查詢與了解(如圖 34)。如果使用者想要查看其圖形，可點選”Show SysML”按鍵即可顯示完整圖形(如圖 35)。

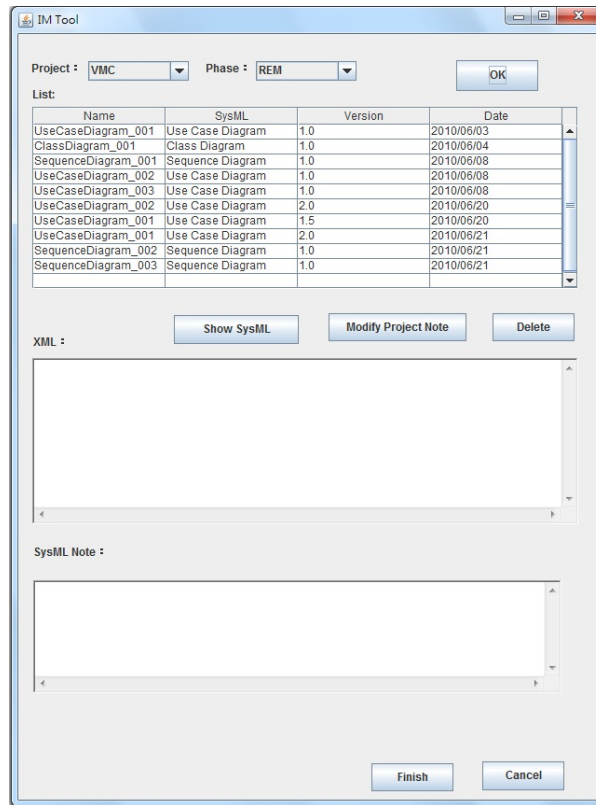


圖 32、選取專案與開發階段即可顯示相關圖形

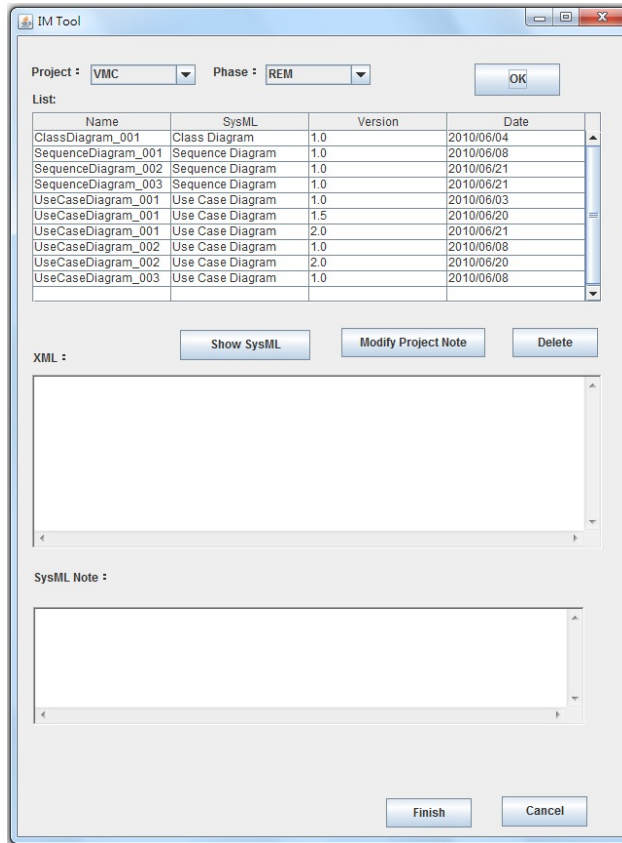


圖 33、以檔案名稱排列表單

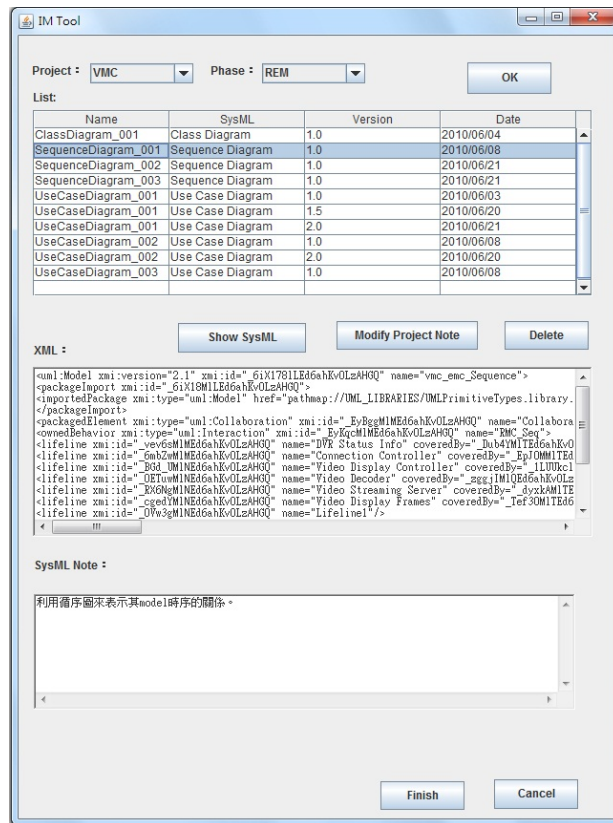


圖 34、顯示 XML 資訊以利使用者查看

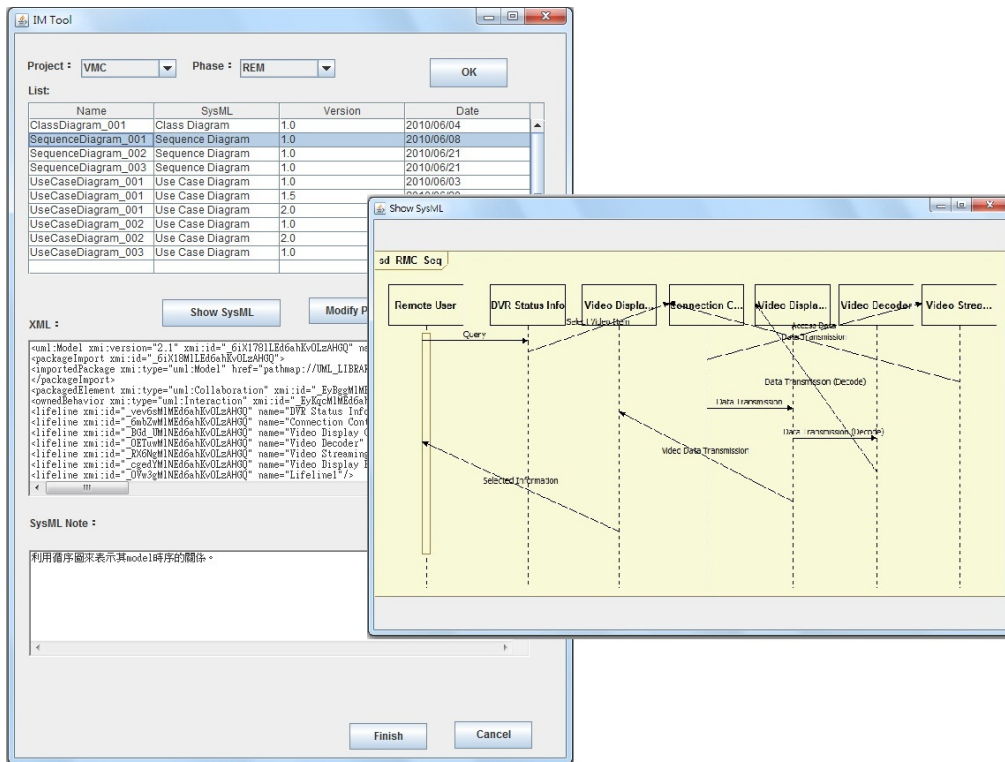


圖 35、顯示 SysML 圖形

4.2.7 刪除圖形資訊

選取專案名稱與開發階段，並於列表中點選欲刪除的圖形(如圖 36)，按下”Delete”鍵，系統會跳出一個警告視窗以防止誤刪(如圖 37)，使用者再次確認後即可刪除(如圖 38)。

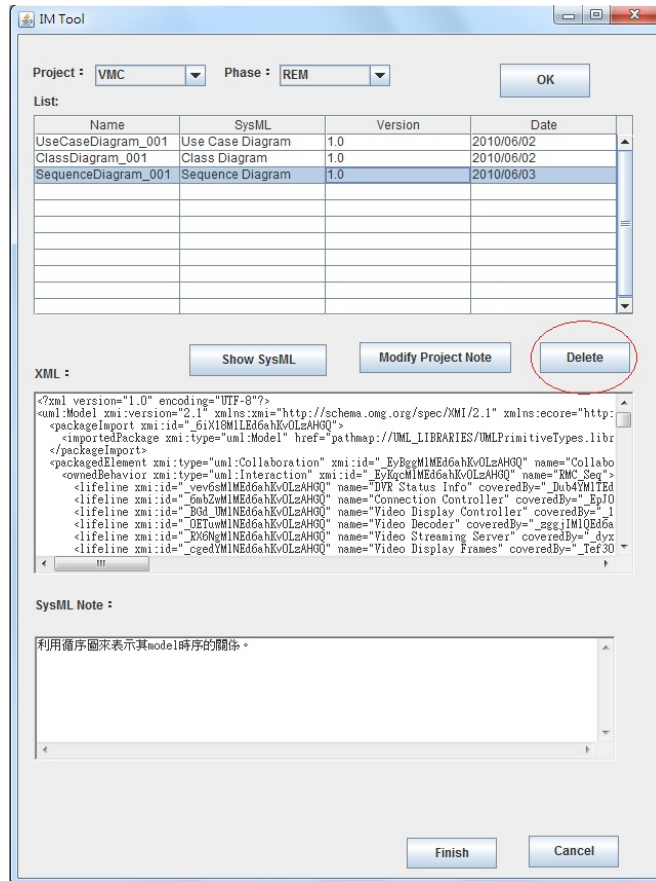


圖 36、選取欲刪除圖形並刪除

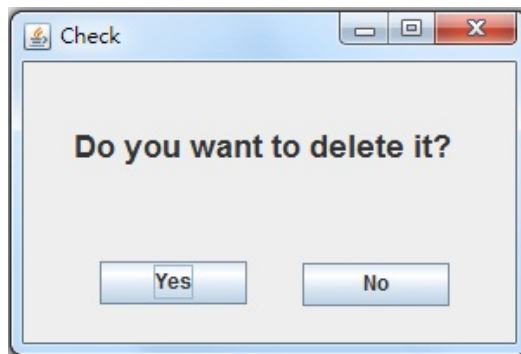


圖 37、確認是否刪除

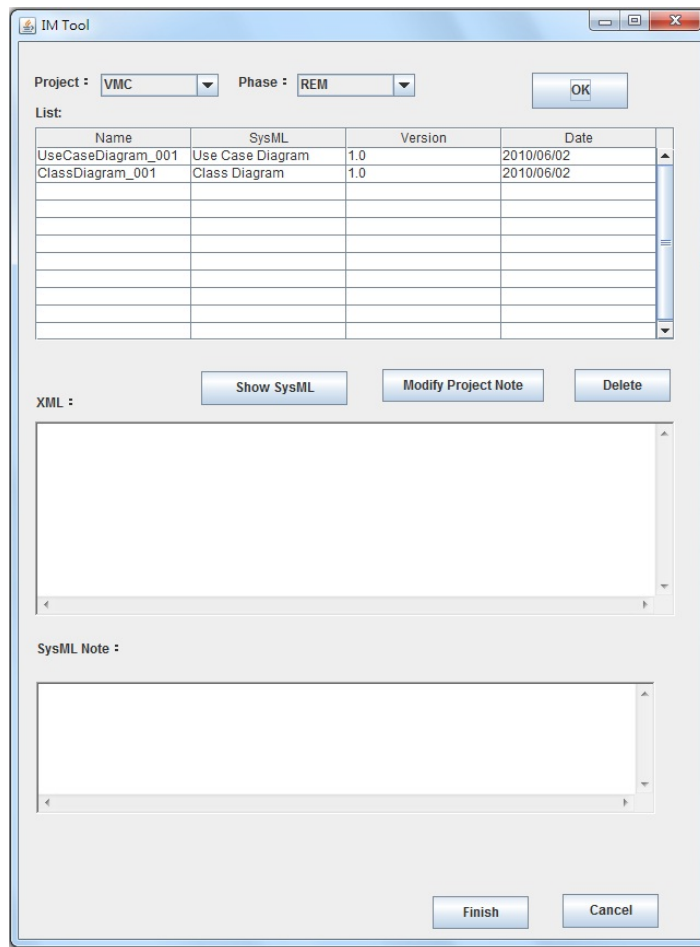


圖 38、刪除後的 IM Tool 視窗

第5章 結論與未來工作

需求階段是軟體生命之首，所有使用者的需求資訊皆由此階段分析了解，故需求階段和其他階段的溝通是相當頻繁的，其間溝通的時間與流暢度將影響專案的時程。嵌入式系統軟體需求階段不止要分析使用者需求資訊，硬體限制也是很重要的資訊，故對於其各階段的資訊溝通更影響了他們的嵌入式系統軟體的專案時程。

於本研究中，我們詳細分析嵌入式系統軟體需求階段的產出與其他階段之關係，並了解到需求階段為軟體生命週期之首，是了解使用者需求與分析硬體限制之處，系統的設計皆需以此為基礎，故各階段與需求階段的溝通品質相當重要。我們開發了一個於需求階段的支援工具，首先我們分析需求階段的產出與其他階段之關係，進而針對系統設計系統流程、並分析了解使用者需求與操作流程，建立一資料庫設計統一 XML 標籤以利整合，運用 XQuery 對 XML 檔案進行整合搜尋，利用此支援工具提升各階段與需求階段溝通速度並減少時間成本。未來我們可以將此工具擴展至各階段，而不是只有與需求階段的溝通，分析各階段間資訊的流程與各資訊間的關係，進而發展一套能完整支援嵌入式系統軟體生命週期的開發支援工具。

參考文獻

- [1] I. Sommerville 原著;鍾俊仁, 姜子龍, 吳正宇譯(2004)。<<軟體工程>>。台灣培生教育出版股份有限公司。
- [2] T. Kroeger and N. Davidson., “A Perspective-Based Model of Quality for Software Engineering Processes,” Conference on Software Engineering ASWEC '09, 2009, pp. 152-161.
- [3] 國立臺灣大學計算機及資訊網路中心, ”軟體發展生命週期”。
http://www.cc.ntu.edu.tw/chinese/epaper/0002/20070920_2011.htm (2010.05.25 瀏覽)
- [4] X. Dong and, M. W. Godfrey, “System-level Usage Dependency Analysis of Object-Oriented Systems,” ICSM 2007. IEEE International Conference on Software Maintenance, 2007, pp.375-384.
- [5] C.W. Lu, C.C. Chu., C.H. Chang and C.H. Wang, “A Model-based Object-oriented Approach to Requirement Engineering (MORE),” COMPSAC Conference on Computer Software and Applications, 2007, vol. 1, pp. 153-156.
- [6] R. J. Glushko, J. M. Tenenbaum and B. Meltzer, “An XML Framework for Agent-based E-commerce,” Communications of the ACM, 1999, March, 42(3): 106-114.
- [7] P. Ciancarini, F. Vitali and C. Mascolo. “Managing Complex Documents Over the WWW: A Case Study for XML,” IEEE Transactions on Knowledge and Data Engineering, 1999, July/August, 11(4):629-638.
- [8] M. G. Wales, "WIDL: Interface Definition for the Web," IEEE Internet Computing, 1999,3(1): 55-59.
- [9] J. Rosenberg, J. Lennox and H. Schulzrinne, “Programming Internet Telephony Services,”IEEE Internet Computing, 1999, MAY/JUNE , 3(3): 63-72.
- [10] A. V. Royappa, "Implementing catalog clearinghouses with XML and XSL, " the Proceedings of the 1999 ACM symposium on Applied computing, 1999, pp. 616-621.

- [11] C.C. Chu, M.H. Wu, M.S. Chen, C.W. Lu, and C.H. Chang, "Data Format Interchange Using XML," Proceedings of the 10th Workshop on Object-Oriented Technology and Applications, 1999, October 7, pp.142-147.
- [12] T.N.Q. Tram, C.G.T.Tai and D.T.B.Thuy, "User interface design pattern Management System Support for Building Information System," 2006 1st International Conference on Digital Information Management, 2006, pp. 96-101.
- [13] W.T. Tsai, Q. Huang, J. Elston and Y. Chen, " Service-Oriented User Interface Modeling and Composition," ICEBE '08, IEEE International Conference on e-Business Engineering, 2008, pp. 21-28.
- [14] J.C. Quiroz, S.J. Louis, A. Shankar and S.M. Dascalu, "Interactive Genetic Algorithms for User Interface Design," CEC 2007 IEEE Congress on Evolutionary Computation, 2007, pp. 1366-1373.
- [15] OMG, "Recently Adopted OMG Modeling Specifications".
http://www.omg.org/technology/documents/recent/omg_modeling.htm
 (2010.05.24 瀏覽)
- [16] R.H. Bourdeau and B.H.C. Cheng, "A formal semantics for object model diagrams, " IEEE Transactions on Software Engineering, 1995. pp. 799-821.
- [17] C. Bock, "SysML and UML 2 Support for Activity Modeling, " *Systems Engineering*, 2006, pp. 160-186.
- [18] T. Johnson, J. Jobe, C. Paredis and R. Burkhart "Modeling Continuous System Dynamics in SysML" *Proceedings of the IMECE 2007*, Nov ' 2007.
- [19] Artisan, "Artisan". <http://www.artisansw.com/products/> (2010.05.25 瀏覽)
- [20] M.E. El-Sharkawi and N.A. El-Hadi El Tazi, "LNV: relational database storage structure for XML documents," The 3rd ACS/IEEE International Conference on Computer Systems and Applications, 2005.
- [21] P.L. Suei, J. Wu, Y.F. Lu, D.N. Lee, S.C. Chou and C.Y. Lin, "A Novel Query Preprocessing Technique for Efficient Access to XML-Relational Databases," 2009 First International Workshop on Database Technology and Applications, 2009, pp. 565-569.
- [22] X. Yuan, C.Y. Wang, H. Ning, X. Lian and J.X. Shi, "A Native XML Storage

- Scheme seamlessly Integrated with Relational Databases," 2009 1st International Conference on Information Science and Engineering (ICISE), 2009, pp.742-745.
- [23] Y.W. Xie, C.Y. Wang, Z.Y. Cao and Y. Chen, "Research on Store XML Data in Relational Database Based on XML Schema," 2007 NPC Workshops IFIP International Conference on Network and Parallel Computing Workshops, 2007, pp. 1001-1005.
- [24] C.Y. Wang, X. Yuan, S. Yu, H. Ning and H.B. Zhang, "A Storage Scheme of Native XML Database Supporting Efficient Updates," 2009 First International Workshop on Database Technology and Applications, 2009, pp. 522-525.
- [25] H. Wang, X.Q. Wang and W. Zeng, "A Research on Automaticity Optimization of KeyX Index in Native XML Database," 2008 International Conference on Computer Science and Software Engineering, 2008, Vol. 4, pp. 700-703.
- [26] H. Liu, J. Zhao, L.Y. Wang, Guangxian Lv and Yafei Ji, "Native XML database technology of digital distribution network," SUPERGEN '09 International Conference on Sustainable Power Generation and Supply, 2009, pp. 1-5.
- [27] W3C, "W3C XML Query (XQuery)".<http://www.w3.org/XML/Query/> (2010.05.24 瀏覽)
- [28] 資訊櫥窗, "淺談 SQL Server 2005 Xquery 新功能".
<http://nft01.nuk.edu.tw/lib/e-news/20060801/3-2.htm> (2010.05.24 瀏覽)
- [29] R. Porkodi, V. Bhuvaneswari, R. Rajesh and T. Amudha,"An Improved Association Rule Mining Technique for Xml Data Using Xquery and Apriori Algorithm," Conference on IACC 2009 IEEE International Advance Computing, 2009, pp. 1510-1514.
- [30] R. Jonathan, " XML Processing and Data Integration with XQuery," IEEE Internet Computing, 2007, vol.11, Is. 4, pp. 62-67.
- [31] P. Reveliotis and M. Carey, " Your Enterprise on XQuery and XML Schema: XML-based Data and Metadata Integration," 22nd International Conference on Data Engineering Workshops, Proceedings, 2006, pp. 80-80.
- [32] C.S. Lin, P.A. Hsiung, S.W. Lin, Y.R. Chen, C.H Lu, S.Y. Tong, W.T. Su, C.C. Chu, C.H. Shih, N.-L. Hsueh, C.H. Chan and C.S. Koong, "VERTAF/Multi-Core: A SysML-based Application Framework for Multi-Core Embedded Software

Development," Journal of the Chinese Institute of Engineers, 2009, 32(7):985-992 (SCI).

附錄

1. 附錄一

```
<?xml version="1.0" encoding="UTF-8"?>
<!--W3C Schema generated by XMLSpy v2010 rel. 3 sp1
(http://www.altova.com)-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="version">
    <xs:simpleType>
      <xs:restriction base="xs:decimal">
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="uml_address">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="sysml_uml" type="xs:string"/>
      <xs:element name="sysml_note">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="sysml_di2" type="xs:string"/>
        <xs:element name="start_date">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="project_note">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
          </xs:simpleType>
        </xs:element>
      </xs:simpleType>
    </xs:element>
  </xs:schema>
```

```

</xs:element>
<xs:element name="project_name">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="project_manager"/>
      <xs:element ref="number"/>
      <xs:element ref="start_date"/>
      <xs:element ref="finish_date"/>
      <xs:element ref="project_note"/>
      <xs:element ref="phase"/>
    </xs:sequence>
    <xs:attribute name="pname" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>
<xs:element name="project_manager">
  <xs:simpleType>
    <xs:restriction base="xs:string">
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="dev_phase">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="file_type"/>
    </xs:sequence>
    <xs:attribute name="phase" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>

```

```

<xs:element name="number">
  <xs:simpleType>
    <xs:restriction base="xs:byte">
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
<xs:element name="finish_date">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
<xs:element name="file_type">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="file_name"/>
    </xs:sequence>
    <xs:attribute name="type" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>
<xs:element name="file_name">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="uml_address"/>
      <xs:element ref="di2_address"/>
      <xs:element ref="version"/>
      <xs:element ref="sysml_uml"/>
      <xs:element ref="sysml_di2"/>
      <xs:element ref="sysml_note"/>
      <xs:element ref="date"/>
    </xs:sequence>
    <xs:attribute name="fname" use="required">
      <xs:simpleType>

```

```
        <xs:restriction base="xs:string">
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
</xs:element>
<xs:element name="di2_address">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
<xs:element name="date">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
</xs:schema>
```