

私立東海大學資訊工程學系研究所

碩士論文

指導教授：朱正忠 教授

實作一個針對多核心嵌入式系統以模板為基礎

之 SysML 需求塑模工具

An Implementation of Template-based SysML and
Requirement Modeling Tool for Multi-core Embedded
System

研究生：葛如恩

中華民國一百年十月二十四日

東海大學碩士學位論文考試審定書

東海大學資訊工程學系 研究所

研究生 葛如恩 所提之論文

實作一個針對多核心嵌入式系統以模板為基礎
之 SysML 需求塑模工具

經本委員會審查，符合碩士學位論文標準。

學位考試委員會

召集人

王豐堅

簽章

委員

盧志偉

張志宏

朱正忠

指導教授

朱正忠

簽章

中華民國 99 年 6 月 26 日

摘要

物件管理組織(OMG)提出模型驅動架構已有一段時間，使得學術及產業上的軟體開發多處利用模型來處理物件與流程的設計，導入此架構後的軟體系統能具備獨立的特性，將使用者的行為與軟體程式碼架構區分開來，於是能改善專案的複雜度管理，妥善處理日漸提高的軟體抽象化程度。另外，在軟體的生命週期中，需求階段裡主要是進行需求訪談及定義需求的工作，這些工作因為專案關係人並非具備軟體專業知識，造成透過自然語言的描述常模糊不清、不一致，使需求管理發生困難，同時，再加上隨時可能的變更，使得訂定規格所帶來的效益減低。

從以上兩方面，本研究期望藉由需求模型化後，能讓各個需求具備獨立的特性，也同時易於分析與統一格式，再透過實作之編輯器，建立需求輸入的介面，讓使用者能利用清單及按鈕等介面方便地撰寫完整的需求，並將資料整理成一份文件記錄。近年來，多核心嵌入式系統的發展迅速，在企業界及學術上都有許多的研究，是一重要的研究議題，因此，本研究也探討如何輔助開發多核心的軟硬體需求，利用預先定義的屬性欄位來處理各類需求，使得分類與辨識的工作更容易，同時，透過參考概要的輔助，使塑模圖形能附帶多核心嵌入式系統的屬性，以輔助後續開發。最後，以上的工作將使整個需求階段達成良好的輔助。

關鍵詞：需求塑模、模型驅動架構、嵌入式系統、多核心架構

Abstract

Object Management Group (OMG) has introduced Model-Driven Architecture for a while; this made many software adopted models to handle designs for objects and processes. These software systems then could have an independent feature, which separates the behavior of user and software codes architecture, and thus it can improve the complexity management of project and take good charge of the daily increasing software visualization level. Besides, in software life cycle, the main tasks in requirement phase are requirement interview and requirements definition. In these tasks, because stakeholders didn't have professional knowledge about computer software, the statements about requirements described by natural language are often ambiguous and inconsistent, and thus bring difficulty to requirement management. Moreover, with possible requirement changes at any time, the benefits from defining specifications lost.

To the two aspects stated above, this research was expected to make every requirements have independent features and make analysis easier by requirement modeling. Further, construct the interface for requirement inputs by this implementation, and then let users use lists and buttons easily describe complete requirements, and arrange the data to be a document record. Recently, multi-core embedded system trend was widely spread, and there are many research in both enterprise area and academic community which shows its importance. Therefore, this research also study about how to assist the development of HW/SW requirements of multi-core ES. We use pre-defined attributes to handle all different kinds of requirements in order to make the work of classification and recognition easier. Further, by the assistant from profile, we can add multi-core ES properties to modeling diagrams and then support following development. Finally, the work stated above would make requirement phase achieve good support.

Keywords: requirement modeling, model-driven architecture, embedded system,
multi-core architecture

目錄

摘要.....	1
Abstract	2
目錄.....	4
圖目錄.....	5
表目錄.....	6
第一章 導論.....	7
1.1 介紹.....	7
1.2 研究動機與目的.....	7
1.3 章節安排.....	7
第二章 相關研究.....	9
2.1 模板(Template).....	9
2.2 樣版(Pattern).....	10
2.3 系統塑模語言(Systems Modeling Language, SysML).....	11
2.4 模型驅動架構(Model-Driven Architecture, MDA).....	12
第三章 研究方法.....	14
3.1 系統架構.....	14
3.2 需求模型.....	14
3.3 需求擷取.....	15
3.4 需求模板與 SysML 圖形之間的對應.....	19
第四章 工具軟體的設計與實作.....	23
4.1 設計與實作之目的.....	23
4.2 需求擷取模組(Requirement elicitation module).....	25
4.3 SysML 圖形產生模組(SysML diagram generation module).....	30
4.3.1 di2 檔案產生器模組(di2 file generator module).....	33
4.3.2 uml 檔案產生器模組(uml file generator module).....	34
第五章 個案研究.....	39
5.1 輸入需求.....	39
5.2 建立需求階層.....	41
5.3 產生 SysML 圖形.....	42
5.4 需求模板之屬性資訊附加於 SysML 圖形中.....	43
第六章 結論與未來方向.....	48
參考文獻.....	49

圖目錄

圖 一、系統堆疊圖	14
圖 二、需求模型	15
圖 三、需求模板之 XML 檔案格式	16
圖 四、工具內部處理需求模板之<userReq>、<info>，及<ur_Attribute>標籤的屬性值的資料結構	19
圖 五、需求原形(requirement stereotypes)的虛擬語意(abstract syntax)	20
圖 六、Papyrus UML 系統之圖形編輯器中，編輯器下方將顯示 SysML 圖形附帶的 userReq 及 info 兩類屬性欄位	20
圖 七、系統架構圖	24
圖 八、需求擷取模組之類別圖	26
圖 九、需求擷取模組之存檔及轉換 SysML 圖形的循序圖	29
圖 十、SysML 圖形產生模組之類別圖	30
圖 十一、di2 檔案產生器模組之類別圖	33
圖 十二、uml 檔案產生器模組之類別圖	34
圖 十三、SysML 圖形產生模組之產生 uml 檔及 di2 檔的循序圖	38
圖 十四、SysML 圖形產生模組之產生 userReq 概要記錄及需求圖概要記錄的循序圖	38
圖 十五、DVR 架構圖	39
圖 十六、輸入需求及條列清單	41
圖 十七、產生基本 SysML 圖形的需求方塊	42
圖 十八、使用者修改後的 SysML 圖形	42
圖 十九、SysML 圖形附帶的需求模板各屬性可供修改	43

表目錄

表 一、量化需求範例	15
表 二、需求模板之 XML 檔案內各種標籤解釋	16
表 三、「Status Manager」需求資訊	21
表 四、SysML 圖形附帶的 userReq 及 info 兩類屬性欄位	22
表 五、VSS 子系統功能性需求範例	40
表 六、VSS 子系統非功能性需求範例	40
表 七、VSS 子系統效能需求範例	40
表 八、VSS 子系統其他類型需求範例	41
表 九、XML 檔案記錄(針對 Connection Server 及 Status Manager)	43
表 十、XML 檔案記錄(針對非功能性需求)	45
表 十一、XML 檔案記錄(針對效能需求及其它需求)	45

第一章 導論

1.1 介紹

本論文研究基於需求模板(Requirement Template)讓使用者撰寫需求的一套編輯器。此編輯器透過模板化的需求輸入、自訂的組織機制，將需求項目組織成 XML 檔案作為儲存之用，並可進一步產生基本的 SysML 需求圖方塊，並將需求文字記錄為 XML 檔案，目的是提供底層程式碼產生之依據。

1.2 研究動機與目的

過去本研究團隊提出一個基於 SysML 語言將使用者需求分析清楚，並產生配合軟體實際開發的系統需求規格的流程[1]，根據此研究指出，如何輔助需求的擷取並制定成軟體規格，並且與輔助後續開發上仍有值得研究之處。另外，學者[2]展示了使用 SysML 需求圖、使用案例圖和需求表格進行道路交通管理系統的定義和塑模，顯示出以圖形進行需求塑模時能額外表示需求間彼此相關聯的可利用性，從以上兩方面，給予了本論文研究方向，本論文即是基於 SysML 的需求塑模，透過圖形化的表示法，研究如何透過軟體工具協助定義軟體規格，以及輔助後續開發。

根據學者[3][4]所提，模板能提供輸入資料時更好的組織化，並且正規化所輸入的資料，避免資料缺漏或模糊等情況；而且以 XML 為格式的資料系統[5]可以達到資料分層分類應用的方便性；若分析需求規格文字中的重要項目，將可找出描述需求時的重要關鍵字，並提供下次撰寫需求的依據。根據以上的想法，本工具期望以模板作為資料輸入的基礎，作為基本機制，再搭配 XML 格式的文件記錄，用來將資料分層分類，並且提供使用者描述需求的重要關鍵字，讓使用者能更方便建立完整之需求，提高制定需求規格的效率。

1.3 章節安排

本論文第一章對研究做一基本介紹，說明動機與目的；第二章說明相關研

究，將目前其他學者的研究略為描述，作為參考之用；第三章說明系統和需求模型的抽象概念，及需求擷取方法；第四章說明工具系統的細部設計及實作的目的；第五章以一 DVR(Digital Video Recording)系統作為個案探討，第六章則為結論與未來方向。

第二章 相關研究

2.1 模板(Template)

模板在電腦科學的應用主要指一種規格化的資料處理技術，學者[6]研究一套利用一個 XML 文件所對應的 PDF 文件的字型集合型態重新合併該 XML 文件的邏輯結構的軟體工具，此工具使用了 XML 表示法作為一個模板來支援對於一個 PDF 文件的邏輯結構的插入工作，並因此產生了一個結構化/標籤化的 PDF 文件。加入邏輯結構這件工作從三個方面使 PDF 文件增加了價值：首先，可存取性提高，提供給視覺損傷的使用者的 PDF 畫面閱讀器運作得更好；第二，媒體選項改良，使用此結構作為一個指引而產生的重瀏覽 PDF 文件的能力使得一個 PDF 文件可以在手持設備上展示；第三，PDF 文件的可再用性從一個似 XML 結構樹的出現而大獲助益，並以致於能以閱讀的順序規範文字的擷取程序，例如：當與 XML 應用程式和資料庫銜接的過程。

此研究專為 Acrobat 系統建立了一個外掛程式，其目的是用來擷取出一個 XML 文件的邏輯結構並將此結構插入到一個現存未結構化的 PDF 文件。製作此外掛程式的目標是使用 XML 作為一個模板以將文件中的內容重新排序成為可閱讀的順序，並且能確定地插入明確的文字界限。藉由此外掛程式進行的工作以及一些對文件其它的改變使得這個外掛程式能夠將一份 PDF 文件轉換為一個標籤化的 PDF 文件。

當嵌入邏輯結構時的第一件工作是取得邏輯結構的來源文件，而此文件是儲存在一個被使用者選擇的外部檔案中。在此時外掛程式將從 XML 檔案中取出邏輯結構並且也暫存檔案中文字的資訊，同時，透過不同的層次來對應暫存的文字與 PDF 文件的文字內容。接著，當從 XML 文件暫存了其內容後，外掛程式必須立即確定一個閱讀順序，其基本的程序是將文件的內容分解成獨立的字母並隨後將這些字母組成依閱讀順序建立的段落。一旦此頁的內容被排序好了，就必須進行對應 XML 文件內容與 PDF 文件內容的工作，這是一項不容易的工作，因為即

使 PDF 的閱讀順序現在是已知的，在 XML 和 PDF 文件的版本中的實際內容順序可能是非常不同的。接著，為了控制文件內容在切割成獨立的字母時所造成的檔案增大，當此頁正被處理並且內容正被加入到結構樹的時候，作為組成邏輯內容的區塊的字母將被合併成更適當大小的區塊。最後，一旦對於單獨一頁的結構插入程序被完成了，即繼續重覆對於以下的頁面進行插入程序，直到所有內容均聯結到文件的結構樹，整個程序才全部完成。

2.2 樣版(Pattern)

近年學者[7]，利用需求樣版協助需求擷取，針對商品現成(off-the-shelf)軟體，從資訊專業諮詢師、客戶與供應商三者模擬需求方案訂定的過程，步驟包括：樣版探索、表單探索、子架構探索等。

在設計樣版(design pattern)的研究上，近年學者[8]透過延伸 UML 的循序圖提出一套表示樣版互動角色(interaction roles)和變異性(variants)的方法，此研究讓設計者能定義並能視覺化針對一個設計樣版所不同的樣版的角色以及互動群組中不同類型的互動，使用的主要元素包括互動(Interaction)、互動使用(InteractionUse)、生命線(LifeLine)和訊息(Message)等四種母類別。所提出的方法是針對循序圖利用 UML 參考概要(profile)進行延伸，並透過樣版互動(pattern interaction)的相關類別來附帶樣版資訊。循序圖因為擁有能透過互動使用(InteractionUse)的母類別來幫助一個大型的互動狀況分解成幾個小型的互動狀況的特性，所以被選用作為展示樣版互動性的圖形。其中，互動使用(InteractionUse)母類別也可被用來描述在其他循序圖中局部的互動狀況。也有學者[9]研究一套稱為 ePAD(eclipse plug-in for design Pattern Analysis and Detection)針對設計樣版分析與偵測的 Eclipse plug-in 工具，其可以透過對從原始碼擷取出來的資料模型進行的結構分析以及經由軟體系統的裝設與監控進行的行為分析來重建設計樣版物件(design pattern instance)，其中，學者提出了一套分析流程：首先利用原始碼擷取器從原始碼產生結構資訊和方法宣告與啟動資訊，接著進行模型分析與原

始碼分析，產生候選的設計樣版，最後再透過裝設、監控與確認重建出樣版物件。

在設計樣版的偵測(detection)方面，學者提出一套新方法[10]，稱為節點深度輸入表(DNIT, Depth-Node-Input-Table)，其可以擷取出在所研究的系統的有向圖之不同深度之中的設計樣版，此研究基於一套圖形對應的演算法：首先選擇一開始節點，根據此節點對有向圖的各節點給予不同的深度值，將深度值作為輸入值，接著按順序從 0, 1, 2..將不同輸入值所能到達的節點寫入表格中，即完成節點深度輸入表。接著須對各已知的設計樣版皆製作此表，利用這些表格，若要偵測某特定樣版是否出現在系統設計中，則可根據該樣版的表格以及系統設計的表格，從兩者中找出深度相同的節點，若能找到與樣版表格中相同的路徑，即確定此樣版存在系統設計中。透過此方法，偵測已知樣版的問題的複雜度能大為減低。

2.3 系統塑模語言(Systems Modeling Language, SysML)

OMG SysML[11]是一種描繪系統需求之語言，重用 UML[12]的部分圖形，如：使用案例圖、循序圖、狀態機圖等，並新增了需求圖(Requirement Diagram)及參數圖(Parametric Diagram)。SysML 也是一個系統工程塑模的標準，其有一系統工程流程，主要利用 SIMILAR 等六個字母代表的工作來進行，其中 I「尋找替代方案(investigate alternatives)」是重要的觀念，能幫助工程師避免落入唯一解(single solution)的思考侷限，而在需求階段就嘗試尋找替代方案(或者替代需求)，以便後續能根據各種評量的指標從多個方案中找出最適者。

近年學者有利用 SysML 在系統設計、架構與功能塑模以及嵌入式系統規格的驗證等方面的研究，其中，學者使用 SysML 來進行一個醫藥系統的整體設計[13]，指出 SysML 是一個具效益可利用的語言並且是一個針對系統中的系統的直觀的可行方案，雖然對於正確的應用中它需要類似 SYSMOD 的方法來支援，但顯示出 SysML 在系統設計方面的可應用性。另外，學者指出[14]區塊定義圖(block definition diagram)和內部區塊圖(internal block diagram)可被用在架構塑模中不同的方面：區塊定義圖描繪出系統中許多不同的類別的屬性和關係；內部區

塊圖則被用來定義系統內部的結構，例如系統中的連接(connection)和資料流(data flow)。系統工程師可多利用以上兩種圖形來進行架構的塑模以定義實體結構並捕捉元件間的互動；軟體工程師則可多使用活動圖及循序圖來進行功能的塑模並製作出適用的軟體。還有學者指出[15]透過對 SysML 的延伸，將可驗證那些以測試為基礎而必須在控制迴圈中透過外界系統來進行工作的系統，並且以一個蒸氣濕度計的模型來展示塑模延伸的使用情形，驗證的過程則是利用執行附帶測試案例的模擬工作來完成。

在參考概要的使用方面，有學者[16]利用製作 SysML 的參考概要(profile)來輔助分散事件模擬規劃(DEVS, Discrete event simulation specification)的模擬，其中透過結合針對元素模型(atomic model)和耦合模型(coupled model)的細部不同資料面向所描繪的 SysML 圖形，使得容易描述 DEVS 模型中結構與行為的屬性。SysML 顯然比其他標準更適用於 DEVS 的模型表示，因為其區塊圖提供了對 DEVS 模型分解最自然的表示方式，並且容易用來定義 DEVSML 模型，最終則可被現有的 DEVS 模擬器來執行，例如：DEVSJava 等。

2.4 模型驅動架構(Model-Driven Architecture, MDA)

OMG MDA[17]是一種支援 CORBA、JAVA、.NET、Web Service 的軟體開發架構，主要透過平台獨立模型(PIM)、平台特定模型(PSM)、轉換程式模式等輔助程式之開發，可將應用程式中的商業功能及行為從程式碼中獨立出來，使得新興的技術得以應用在商業軟體上。MDA 也是一種系統開發的方法論，它增加了模型在開發工作中的助益。

使用 MDA 的過程中，工程師先從計算獨立模型(CIM)開始對系統進行分析，接著建立平台獨立模型(PIM)，然後進行對應(mapping)的工作，其中模型型態的對應(model type mapping)主要是將利用 PIM 語言所定義的模型型態所產生的模型對應到以 PSM 語言所定義的型態來表示的模型；而此種對應也可包括定義出實際對應的規則(mapping rules)。

經過對應(mapping)以後，工程師透過模型轉換(model transformation)的工作產生一個平台特定模型(PSM)，它是一個對於平台獨立模型(PIM)所處理的同一系統加入了未來將如何使用平台的資訊後的模型。一個平台特定模型可以提供一些如何使用平台的資訊，而提供的資訊數量與它被使用的目的有關，也就是說，假如提供了所有製作一個系統所需的資訊，那麼它將成為一個系統的實作，相對地，假如它可以提供未來進一步的改良並且被轉換出一個可以直接實作的平台特定模型(PSM)，那麼它也可被作為平台獨立模型的角色。平台特定模型還可以提供各種不同的資訊，如程式碼，系統佈署描述等一些用來調校系統的資訊。

第三章 研究方法

3.1 系統架構

本工具之堆疊圖如下。在第四章有獨立兩節，說明需求擷取模組(Requirement elicitation module)及 SysML 圖形產生模組(SysML diagram generation module)的細部設計。

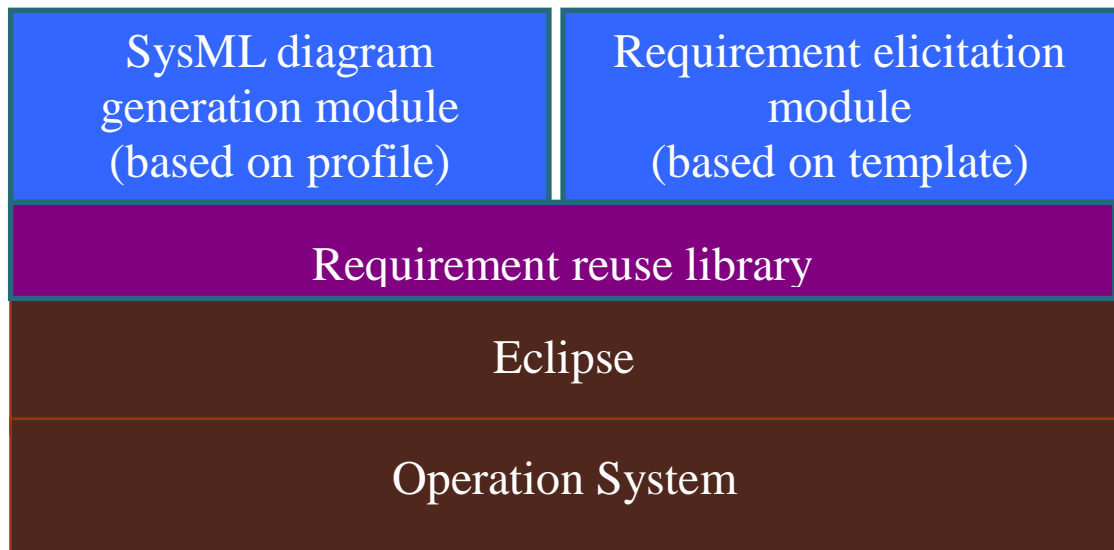


圖 一、系統堆疊圖

3.2 需求模型

在抽象觀點下我們將需求模型化，架構的考量如下：

- SysML 標準所提出的概念指出需求間應有包含(containment)的關係，即「父需求可以包含多個子需求」，因此，我們的模型也提供了需求間繼承(inheritance)以及包含的關連性。
- 為了提供紀錄 XML 格式資料時的操作方便，我們利用一群特定的參數、物件來處理字串資訊(xmlStrParams[]、xmlStrObjects[])。
- 本工具屬於利用平台獨立模型(PIM)的概念來進行塑模工作。

requirement
+id : string
-text : string
-parent : object
-child : object
-xmlStrParams[] : object
-xmlStrObjects[] : object

圖 二、需求模型

3.3 需求擷取

使用本工具時，第一個步驟是輸入所有可能的需求。在此，我們採用了「需求模板(Requirement Template)」作為各類需求資訊統一格式和檔案存取的依據，並且以獨立的屬性來記錄需求文字中的數值資訊，輔助使用者採用量化的方式來訂定各類需求，於是能更容易驗證需求規格的達成率，提高精確度，如下所示：

表 一、量化需求範例

回應時間	5 秒
傳輸速率	10 frame/s
客戶端數量	10 個

當使用者輸入需求資訊，工具即會以此格式做整理，利用需求模板的欄位值將資訊格式化，每一個欄位各自具有意義，此模板是一 XML 檔案，如下圖所示：

標籤/ 屬性名稱	資料參考範例
[-] [e] requirement_TemplatesData	
[-] [e] userReq	
[a] id	_W4K6QGxJEd6SRrRO6Ugvjg
[-] [e] ur_Element	
[a] type	function
[-] [e] ur_Attribute	
[a] name	internetConnection
[a] value	Connection Server最多可以同時處理5個客戶端連線
[a] valueType	String
[a] relationToDgt	小於或等於
[a] dgtValue	5
[a] dgtValueType	int
[a] unit	個
[-] [e] info	
[a] id	_W4K6QWxJEd6SRrRO6Ugvjg
[a] reqID	VSS-FNR-001
[a] reqKind	Functional
[a] reqName	Connection Server
[a] reqShortName	CS
[a] currentVersion	1.0.1
[a] previousVersion	1.0.0
[a] projectName	Digital Video Recording
[a] establishDate	2009.7.1
[a] establishUser	C.H.
[a] derivedFrom	VSS-FNR-004
[a] satisfyTo	VSS-FNR-003
[a] baseClass	_xGO08E6BE6Y04XoPnfHow

圖 三、需求模板之 XML 檔案格式

針對各 XML 標籤解釋如下：

表 二、需求模板之 XML 檔案內各種標籤解釋

標籤元素	屬性	必要性	定義
<requirement_TemplatesData>		必要(1)	需求模板標籤
<userReq>		必要(1 or *)	使用者需求標籤
	id	必要(1)	userReq 的識別碼
	parentReq	非必要(0 or 1)	父需求的識別碼
<ur_Element>		必要(1)	使用者需求元素標籤
	type	必要(1)	ur_Element 的類

					型
		<ur_Attribute>		必要(1)	使用者需求屬性標籤
			name	必要(1)	ur_Attribute 的名稱
			value	必要(1)	ur_Attribute 的資料內容
			valueType	必要(1)	資料內容之型態
			relationToDgt	非必要(0 or 1)	資料中數值資訊的邊界值關係
			dgtValue	非必要(0 or 1)	數值資訊的邊界值
			dgtValueType	非必要(0 or 1)	數值資訊的型態
			unit	非必要(0 or 1)	數值資訊的單位
			relatedAttrbt	非必要(0 or 1)	記錄其與 name 屬性所有可能之欄位值中最相關的一個
		<info>		必要(1 or *)	一個 userReq 所對應的資訊標籤
			id	必要(1)	電腦識別碼
			reqID	必要(1)	供使用者辨識的識別碼
			reqKind	必要(1)	所對應之 userReq 的需求分類
			reqName	非必要(0 or 1)	所對應之 userReq 的需求名稱
			reqShortName	非必要(0 or 1)	所對應之 userReq 的簡寫需求名稱
			currentVersion	必要(1)	所對應之 userReq 的目前版本編號
			previousVersion	非必要(0 or 1)	所對應之 userReq 的上一版本編號

		projectName	必要(1)	所對應之 userReq 的所屬專案名稱
		establishDate	必要(1)	所對應之 userReq 的建立日期
		establishUser	必要(1)	所對應之 userReq 的建立者
		derivedFrom	非必要(0 or 1)	所對應之 userReq 從什麼衍生而來
		satisfyTo	非必要(0 or 1)	所對應之 userReq 能夠符合或支援哪一需求
		baseClass	必要(1)	記錄採用了此 info 的物件其識別碼資訊

本工具針對模板之<userReq>、<info>標籤及需求屬性標籤<ur_Attribute>各種可能的屬性值，設計了 userReq 和 info 兩個資料結構，這兩個資料結構是繼承 UML 中的類別型態，其功能分述如下：

- userReq：從功能性需求、非功能性需求等方面列出嵌入式系統可能的需求屬性(特性)，並且針對多核心設計了 cpuNum、cpuCore 兩個屬性，用來輔助使用者撰寫 CPU 個數或核心數方面的要求。
- info：根據需求規格書上常見的各種編號、名稱等資訊，再加上 SysML 語言的需求圖(Requirement Diagram)所設計的需求之間的關連性(relationship)，用來做為需求規格的基本資訊欄位。

userReq 和 info 的高階概念圖如下：

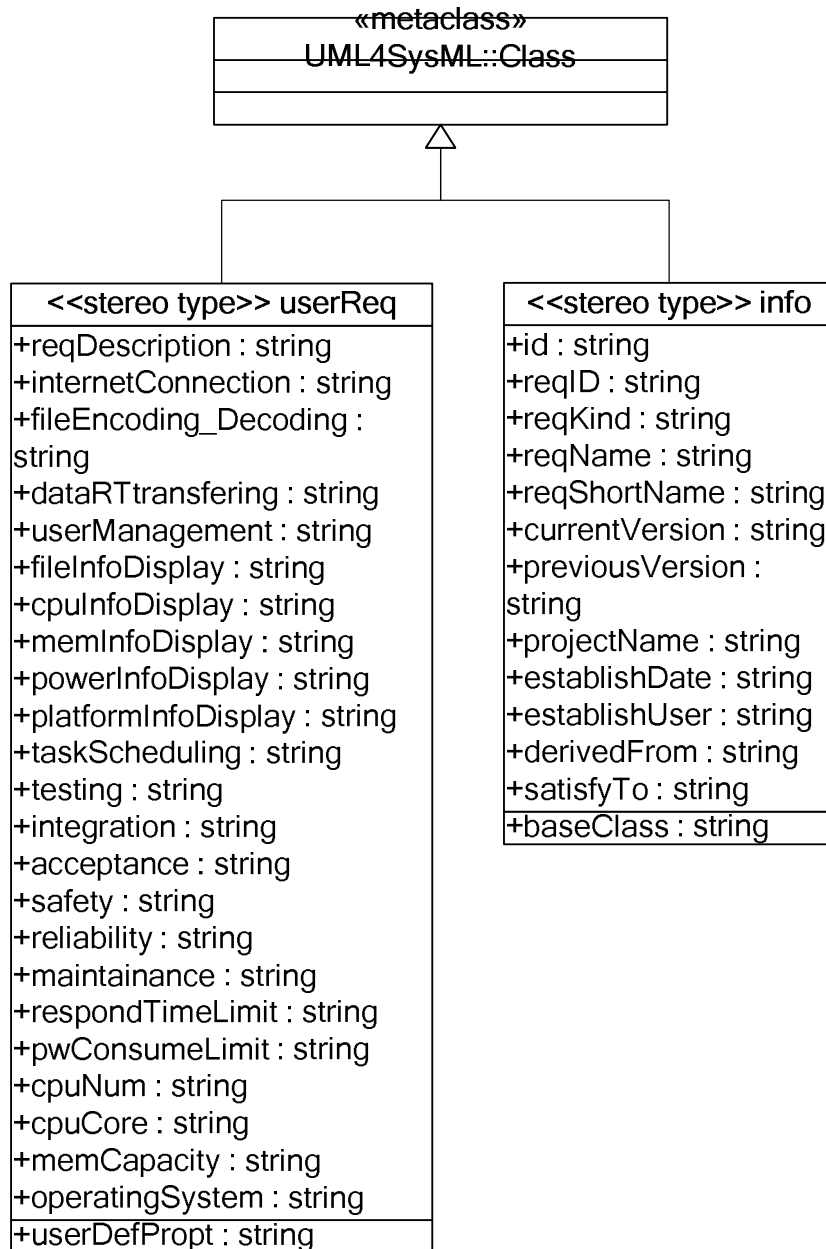


圖 四、工具內部處理需求模板之<userReq>、<info>，及<ur_Attribute>標籤的屬性值的資料結構

3.4 需求模板與 SysML 圖形之間的對應

我們利用 Papyrus UML 系統做為使用者閱讀 SysML 圖形及後續進一步編輯圖形的平台，因此，我們提供 Papyrus UML 系統辨識需求模板各 XML 標籤的參考概要(profile)。這個參考概要與上一節所示的資料結構完全相同，目的是讓 SysML 圖形能附帶需求模板所有的屬性欄位，同時讓使用者未來可以修改編輯。

在整體的語法概念上，我們使用了 UML 語言的延伸結構中的 stereotype 來建立，如下圖所示：

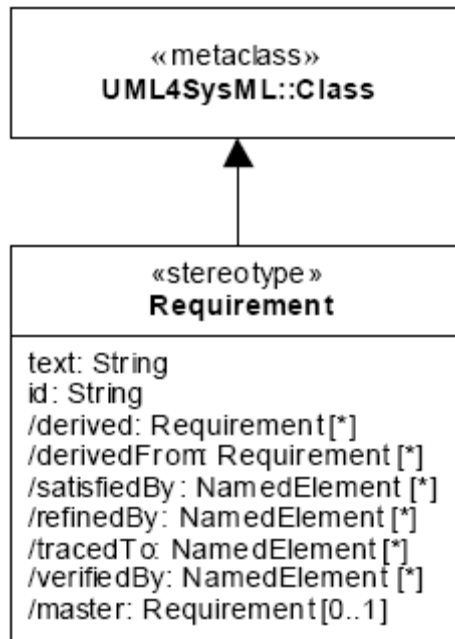


圖 五、需求原形(requirement stereotypes)的虛擬語意(abstract syntax)以「Status Manager」功能性需求為例，使用 Papyrus UML 系統閱讀其 SysML 圖形時，編輯器下方將顯示 SysML 圖形附帶的資訊，如下圖所示：

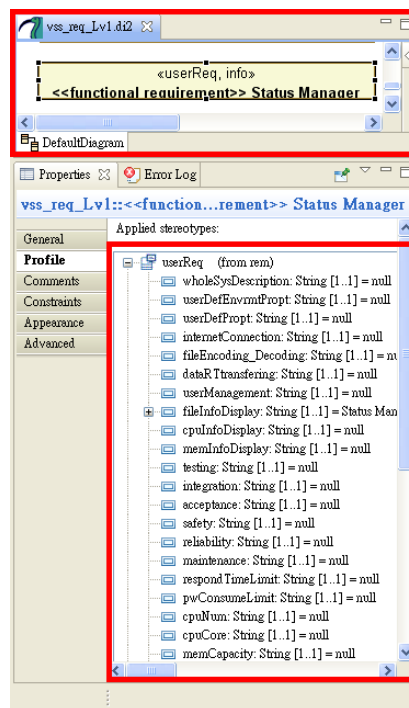


圖 六、Papyrus UML 系統之圖形編輯器中，編輯器下方將顯示 SysML 圖形附帶的 userReq 及 info 兩類屬性欄位

其透過需求模板記錄的需求資訊如下表，針對單一一個需求模板的 XML 屬性(如：fileInfoDisplay、ReqID、ReqKind...等)，記錄一個資料值：

表 三、「Status Manager」需求資訊

需求模板屬性欄位		需求資訊資料值
分類	屬性	
userReq	fileInfoDisplay	Status Manager 要能夠提供實體資料庫伺服器上現有的檔案清單及數位攝影機清單。
Info	ReqID	VSS- FNR-003
	ReqKind	Functional
	reqName	Status Manager
	reqShortName	SM
	currentVersion	1.0.0
	projectName	Digital Video Recording
	establishDate	2010.7.29
	establishUser	C.H.

下表則顯示「Status Manager」所擁有的需求模板屬性可以對應到 SysML 圖形附帶的資訊中 userReq 及 info(以藍色方框框起來者)兩類欄位。

表 四、SysML 圖形附帶的 userReq 及 info 兩類屬性欄位

需求模板屬性欄位		SysML 圖形附帶的資訊
分類	屬性	
userReq	fileInfoDisplay	<div style="border: 1px solid blue; padding: 2px;">userReq (from rem)</div> <ul style="list-style-type: none"> wholeSysDescription: String [1..1] = null userDefEnvrmtPropt: String [1..1] = null userDefPropt: String [1..1] = null internetConnection: String [1..1] = null fileEncoding_Decoding: String [1..1] = null dataRttransferring: String [1..1] = null userManagement: String [1..1] = null fileInfoDisplay: String [1..1] = Status Manager要能夠提供 cpuInfoDisplay: String [1..1] = null memInfoDisplay: String [1..1] = null testing: String [1..1] = null integration: String [1..1] = null acceptance: String [1..1] = null safety: String [1..1] = null reliability: String [1..1] = null maintenance: String [1..1] = null respondTimeLimit: String [1..1] = null pwConsumeLimit: String [1..1] = null cpuNum: String [1..1] = null cpuCore: String [1..1] = null memCapacity: String [1..1] = null operatingSystem: String [1..1] = null dataTransferAmountLimit: String [1..1] = null <div style="border: 1px solid blue; padding: 2px; margin-top: 10px;">info (from rem)</div> <ul style="list-style-type: none"> id: String [1..1] = _P8bZmWxCed4eUiqGJhjGiV reqID: String [1..1] = VSS-FNR-003 reqKind: String [1..1] = Functional reqName: String [1..1] = Status Manager reqShortName: String [1..1] = SM currentVersion: String [1..1] = 1.0.0 previousVersion: String [1..1] = null projectName: String [1..1] = Digital Video Recording establishDate: String [1..1] = 2009.7.1 establishUser: String [1..1] = C.H. derivedFrom: String [1..1] = null satisfyTo: String [1..1] = null baseClass: String [1..1] = _OOPZzTknEd99VDWRj0k2QL
Info	ReqID	
	ReqKind	
	reqName	
	reqShortName	
	currentVersion	
	projectName	
	establishDate	
	establishUser	

第四章 工具軟體的設計與實作

4.1 設計與實作之目的

在需求擷取的過程中，為了避免需求定義模糊不清，同時輔助開發者更容易設計多核心嵌入式系統的效能及耗電需求。透過建立符合 SysML 標準之需求圖形，銜接後續的開發。本工具的實作目標如下：

- **提供清單式的需求擷取與需求分類編輯**

透過統一的按鈕及清單控制介面，使用者可利用清單將需求條列下來，並且可以針對多核心嵌入式系統的屬性編輯需求項目的內容，使得更容易在需求階段就進行多核心嵌入式系統的輔助開發。透過自動編號的功能，可將需求分類，讓使用者更容易閱讀與修改。

- **建立物件式需求雛形**

在工具內部建立一套物件式的需求雛形，內容記錄包括需求名稱、編號、上層需求、文字資料、數值資料、XML 記錄資料和各類需求暫存統計資料等。此雛形可與清單介面配合，統一資訊輸入的來源，並且內部對需求資訊做監控，可即時檢查出可能的錯誤。

- **提供參考概要(profile)的輔助機制**

工具預先將參考概要的完整 XML 檔案的字串記錄在程式中，接著再將其輸出成檔案，使得未來建立的 SysML 圖檔可與此概要配合，讓 Papyrus UML 系統可辨識附帶在 SysML 圖檔上的需求屬性資訊；另一方面，透過分析圖檔的概要資訊在 Papyrus UML 系統中的記錄方法，進一步建立工具需求屬性資訊的撰寫機制。

- **建立專屬的 SysML 圖檔**

工具預先將 SysML 圖檔的架構 XML 字串記錄在程式中，當使用者欲轉換圖形時，可將重要圖形項目寫入 XML 字串架構中，並輸出成為獨立的檔案。圖檔包括副檔名為 .uml 及 .di2 的兩種檔案，其中，.uml 檔記錄了各類

參考概要預設值、XMI 資訊以及資源的路徑值；.di2 檔案則記錄圖形座標、線條、字型等資訊。透過圖檔獨立的特性，將提高本工具的可攜性，並容易建立特殊的圖形關聯。

本工具的架構圖如下：

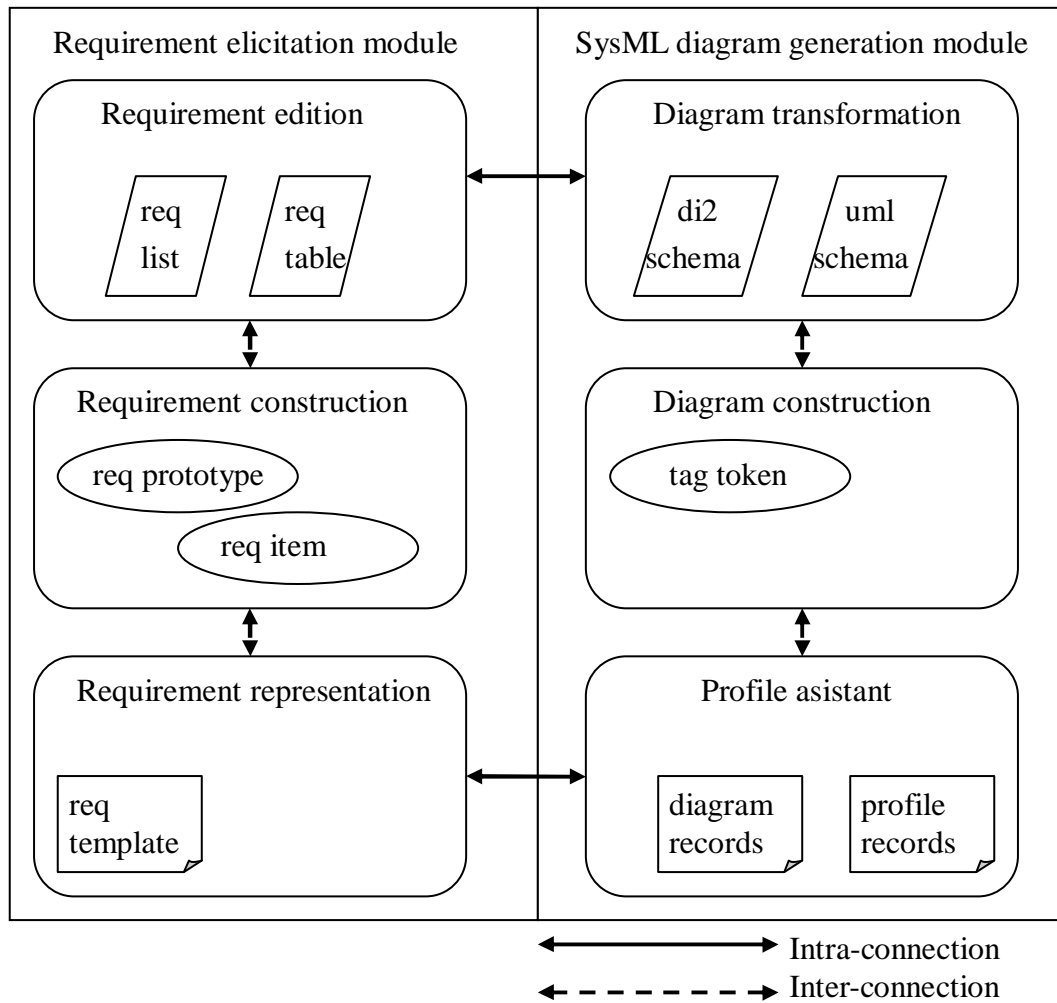


圖 七、系統架構圖

本工具主要可分為兩大模組，包括需求擷取模組(Requirement elicitation module)及 SysML 圖形產生模組(SysML diagram generation module)。需求擷取模組的功能是提供使用者介面，讓使用者可以輸入需求資訊，並且利用模板來做格式化的處理，此模組可分為三部份，第一，需求編輯(requirement edition)指工具提供的編輯功能，主要透過清單(req list)和表格(req Table)等元件來運作。第二，需求建構(requirement construction)指需求的建構機制，包括程式中利用物件式的需

求雛形(req prototype)做為基本建構單元，將資訊、結構、運作方法完整地在一個雛形中定義清楚，往後的工作僅需針對雛形進行操作，即可完成需求資訊的編輯與管理。第三，需求表示(requirement representation)指系統底層的資料表示方法，主要利用需求模板(req template)做為記錄需求的格式，並且可以做為模組間資訊交換之用。

SysML 圖形產生模組主要負責將需求資訊轉換成 SysML 圖形，可分為三部份，第一，圖形轉換(diagram transformation)指需求資訊與圖形資訊的對應是透過比對檔案的綱要(schema)來進行，必須分析綱要的結構，將對應的資料值寫入標籤(tag)中，使得資訊能被保存而沒有遺失地轉換過去。第二，圖形建構(diagram construction)指 SysML 圖形的建構機制，程式中主要利用標籤單元(tag token)來處理各種的資料值，除了做為基本建構單元外，也提供 XML 字串的處理機制，包括分類及輸出等功能。第三，概要輔助(profile assistant)指透過建立參考概要，SysML 圖形將能附帶需求模板的屬性值，使得圖形能延伸記錄多核心嵌入式系統的屬性，以提供後續開發使用。模組在此部份必需同時處理圖形記錄(diagram records)與概要記錄(profile records)，兩者必須區分開來，但都以 XML 為格式來記錄。

4.2 需求擷取模組(Requirement elicitation module)

負責接收使用者輸入之需求資訊。主要利用 RequirementEditor 建立使用者介面，提供清單與按鈕來編輯需求，在設計架構上則利用 ReqItemPrototype 作為需求資料單元的雛形，用以儲存單一個需求的文字描述、編號等資訊，另外，也處理了由需求文字轉換為 SysML 圖形或 XML 檔案的銜接工作。從靜態結構來看，其類別圖如下：

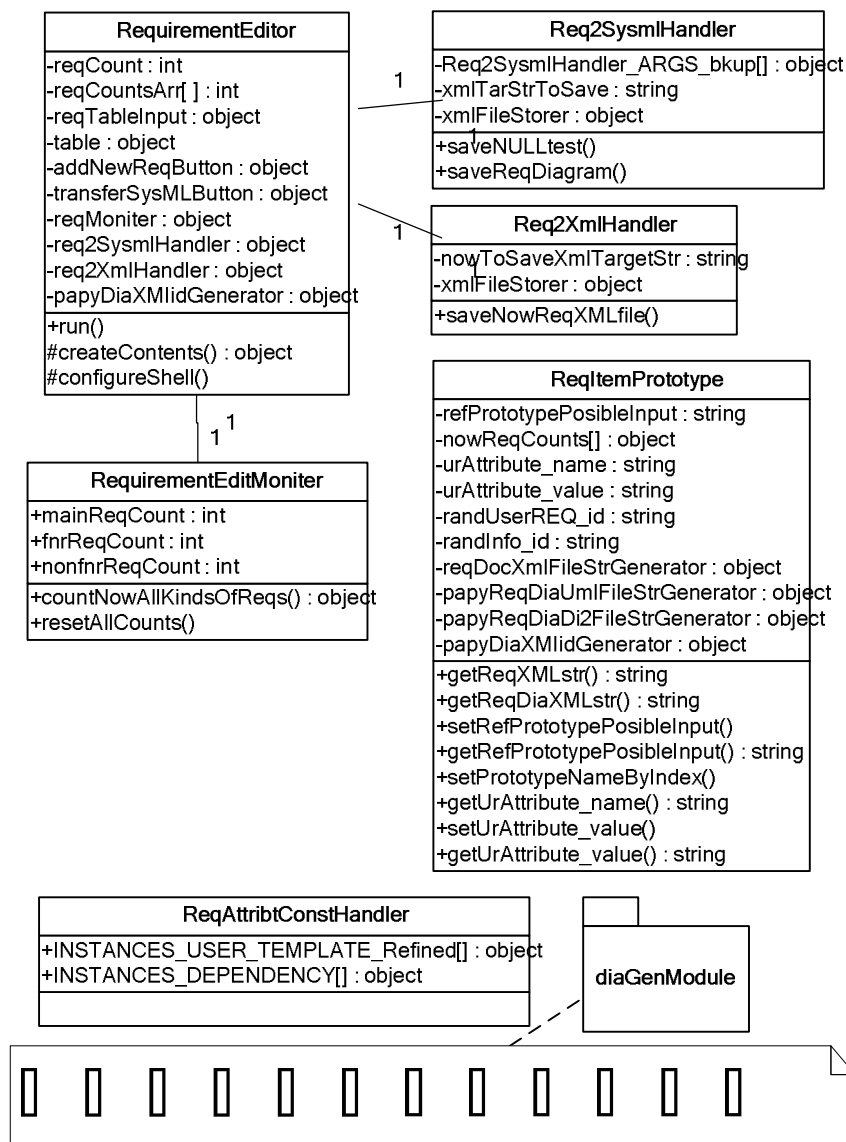


圖 八、需求擷取模組之類別圖

- RequirementEditor :
 - 作為擷取需求之界面，主要包括表格(table)、按鈕(addNewReqButton、transferSysMLButton)。
 - 負責暫存需求資訊；提供需求清單給使用者看。
 - 儲存目前各類需求之數量(reqCount、reqCountsArr)，以便提供流水編號的依據。
 - 利用表格內容之預設輸入結構(reqTableInput 物件，型態為 ReqItemPrototype 物件串列)建立輸入資料的機制。

- 提供按鈕給使用者啟動編輯器各種子功能。
 - 利用一個監控者(reqMonitor 物件，實作 RequirementEditMonitor 類別)來監控編輯器內暫存的資料。
 - 實作 Req2SysmlHandler 類別(req2SysmlHandler 物件，將需求轉換為 SysML 圖形的處理者)，負責呼叫存檔器(xmlFileStorer 物件)儲存 SysML 圖形檔。
 - 實作 Req2XmlHandler 類別(req2SysmlHandler 物件，將需求紀錄為 XML 檔案的處理者)，負責呼叫存檔器(xmlFileStorer 物件)儲存 XML 檔案。
 - 實作 PapyDiaXMlidGenerator 類別(papyDiaXMlidGenerator 物件，Papyrus 系統內圖形之 XMI 識別碼產生者)，負責產生 XMI 識別碼以便與 Papyrus 系統相容。
- Req2SysmlHandler：
 - 將需求記錄為 SysML 圖形檔案之處理者。
 - 是負責轉換 SysML 圖形的主要物件。
 - 備份一份初始參數(Req2SysmlHandler_ARGS_bkup[]，型態為字串陣列)。
 - 負責呼叫存檔器(xmlFileStorer)將 SysML 圖形的 XML 字串存檔。
 - Req2XmlHandler：
 - 將需求記錄為 XML 檔案之處理者。
 - 是負責將需求紀錄為 XML 檔案的主要物件。
 - 負責整理、排列即將記錄的 XML 字串，必須確認 XML 字串的順序正確。
 - 負責呼叫存檔器(xmlFileStorer)將 XML 字串存檔。
 - RequirementEditMonitor：
 - 需求編輯器監控者。
 - 負責統計各類需求之數量。

- ReqItemPrototype :
 - 需求項目基礎型態。
 - 定義一個需求所必要的限制、預設條件等，以及此需求的文字描述 (refPrototypePossibleInput)，是記錄需求資訊的基本單位。
 - 儲存直到最新一個新增的需求為止的各類需求數量(nowReqCounts[]，型態為整數陣列)。
 - 暫存需求文件及需求圖 XML 檔案的各類標籤資訊(urAttribute_name、urAttribute_value 等)，以便未來儲存為 XML 格式的檔案。
 - 實作 REMreqDocXmlFileStrGenerator 類別(reqDocXmlFileStrGenerator 物件，REM 系統需求文件 XML 檔案之 XML 字串產生者)，負責產生 REM 系統需求文件 XML 檔案的標籤資訊。
 - 實作 PapyReqDiaUmlFileStrGenerator 類別 (papyReqDiaUmlFileStrGenerator 物件，Papyrus 系統內需求圖 uml 檔案之字串產生者)，負責產生 Papyrus 系統需求圖 uml 檔案的標籤資訊。
 - 實作 PapyReqDiaDi2FileStrGenerator 類別 (papyReqDiaDi2FileStrGenerator 物件，Papyrus 系統內需求圖 di2 檔案之字串產生者)，負責產生 Papyrus 系統需求圖 di2 檔案的標籤資訊。
 - 實作 PapyDiaXMIidGenerator 類別(papyDiaXMIidGenerator 物件，Papyrus 系統內圖形之 XMI 識別碼產生者)，負責產生 XMI 識別碼以便與 Papyrus 系統相容。
 - 依照需求文件 XML 檔案之標籤順序，串接所有標籤及屬性後，回傳整段 XML 字串(getReqXMLstr 方法)。
 - 依照需求圖 XML 檔案之標籤順序，串接所有標籤及屬性後，回傳整段 XML 字串(getReqDiaXMLstr 方法)。
 - 提供存取暫存之各類標籤資訊。
- ReqAttribtConstHandler :

- 需求屬性常數處理者。
- 儲存重要且被多個類別共用的屬性字串、常數字串，以便統一處理。
- diaGenModule：
 - 圖形產生模組。
 - 負責處理所有圖形的資訊、以及在螢幕上繪製圖形等工作。

從動態結構來看，以下為需求擷取模組之存檔及轉換 SysML 圖形的循序圖：在存檔部份，先由 RequirementEditor 呼叫 Req2XmlHandler，再呼叫 XmlFileStorer 進行存檔，存檔之後回傳儲存的路徑及檔名；在轉換 SysML 圖形部份，先由 RequirementEditor 呼叫 Req2SysmlHandler，再呼叫 XmlFileStorer 進行存檔，存檔之後回傳儲存的路徑及檔名，並且檔名是隨機產生的字串。

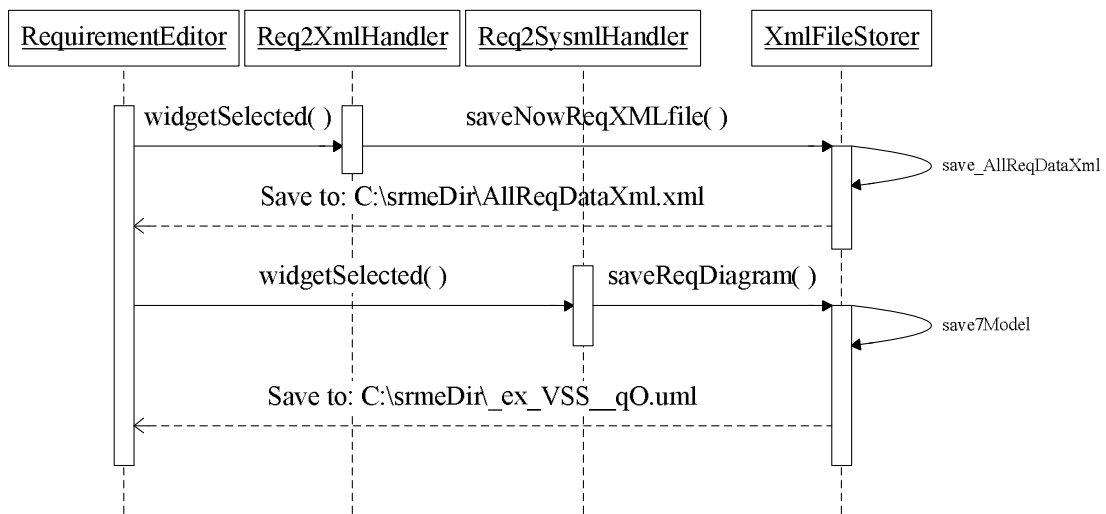


圖 九、需求擷取模組之存檔及轉換 SysML 圖形的循序圖

4.3 SysML 圖形產生模組(SysML diagram generation module)

負責接收需求擷取模組之需求資訊，並將資訊記錄在圖形中。主要利用 REMprfTagStrGenerator、PapySysmlPrfTagStrGenerator 輔助產生各種概要(profile)的字串資料，在細部設計上利用 PapyDiaXMLidGenerator 產生隨機的 XMI 辨識碼，以便與 Papyrus 系統整合，另外，進一步設計了 di2Generator 及 umlGenerator 兩個子模組，專門處理一個 SysML 圖檔所包括的 di2 檔案、uml 檔案的產生工作。從靜態結構來看，其類別圖如下：

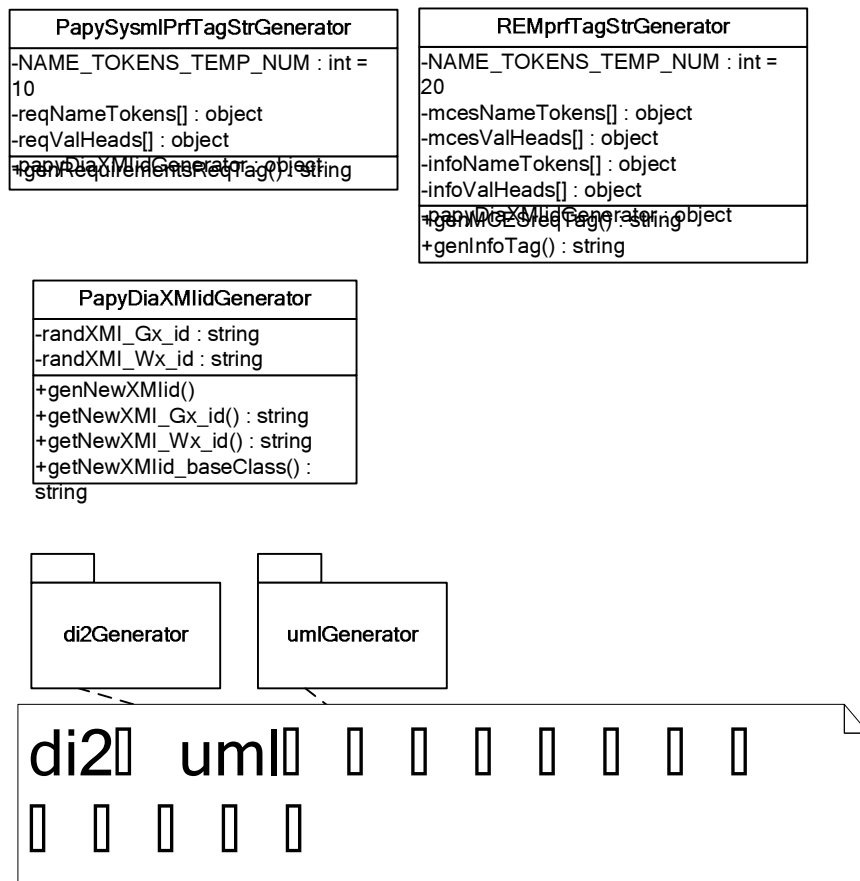


圖 十、SysML 圖形產生模組之類別圖

- PapySysmlPrfTagStrGenerator
 - Papyrus 系統的 SysML 概要所定義的標籤資訊之字串產生者。
 - 是負責產生 Papyrus 系統的 SysML 概要所定義的標籤資訊的主要物件。
 - 儲存 SysML 概要定義的標籤字串單元之數量限制

(NAME_TOKENS_TEMP_NUM=10，暫定 10 個)。

- 儲存需求圖各種標籤字串單元的名稱(reqNameTokens[])，型態為字串陣列)。
 - 儲存需求圖各種標籤字串單元的名稱寫入值(reqValHeads[])，型態為字串陣列)。
 - 實作 PapyDiaXMIidGenerator 類別(papyDiaXMIidGenerator 物件，Papyrus 系統內圖形之 XMI 識別碼產生者)，負責產生 XMI 識別碼以便與 Papyrus 系統相容。
 - 依照需求圖各種標籤內容資訊順序，串接底下所有標籤及屬性後，回傳整段 XML 字串(genRequirementsReqTag 方法)。
- REMprfTagStrGenerator
 - REM 概要所定義的標籤資訊之字串產生者。
 - 是負責產生 REM 概要所定義的標籤資訊的主要物件。
 - 儲存 REM 概要定義的標籤字串單元之數量限制
(NAME_TOKENS_TEMP_NUM=20，暫定 20 個)。
 - 儲存需求圖 MCESreq 標籤字串單元的名稱(mcesNameTokens[])，型態為字串陣列)。
 - 儲存需求圖 MCESreq 標籤字串單元的名稱寫入值(mcesValHeads[])，型態為字串陣列)。
 - 儲存需求圖 info 標籤字串單元的名稱(infoNameTokens[])，型態為字串陣列)。
 - 儲存需求圖 info 標籤字串單元的名稱寫入值(infoValHeads[])，型態為字串陣列)。
 - 實作 PapyDiaXMIidGenerator 類別(papyDiaXMIidGenerator 物件，Papyrus 系統內圖形之 XMI 識別碼產生者)，負責產生 XMI 識別碼以便與 Papyrus 系統相容。

- 依照需求圖 MCEsreq 標籤內容資訊順序，串接底下所有標籤及屬性後，回傳整段 XML 字串(genMCEsreqTag 方法)。
- 依照需求圖 info 標籤內容資訊順序，串接底下所有標籤及屬性後，回傳整段 XML 字串(genInfoTag 方法)。
- PapyDiaXMIidGenerator
 - Papyrus 系統內圖形之 XMI 識別碼產生者。
 - 負責產生 Papyrus 系統內圖形之 XMI 識別碼，以便本工具產生的圖形能與 Papyrus 系統相容。
 - 儲存隨機的 XMI 識別碼(randXMI_Gx_id、randXMI_Wx_id。有兩種類型，即字串內部含有"Gx"字串或是"Wx"字串兩種)。
 - 產生隨機的 XMI 識別碼(genNewXMIid 方法)。
 - 取得隨機的 XMI 識別碼(getNewXMI_Gx_id、getNewXMI_Wx_id 方法)。
 - 取得隨機的 XMI 識別碼(getNewXMIid_baseClass 方法)，提供識別單一一個物件。
- di2Generator
 - di2 檔案產生器模組。
 - 負責產生 Papyrus 系統 di2 檔案及相關機制。
- umlGenerator
 - uml 檔案產生器模組。
 - 負責產生 Papyrus 系統 uml 檔案及相關機制。

4.3.1 di2 檔案產生器模組(di2 file generator module)

接收需求擷取模組之需求資訊後，以符合 Papyrus 系統 di2 檔案的綱要 (schema) 的編寫方式，負責將各種 XML 標籤與屬性的欄位值記錄為 XML 格式的文字檔案。在細部設計上，主要必須依照標籤內容資訊順序，串接底下所有標籤及屬性後，回傳整段 XML 字串。從靜態結構來看，其類別圖如下：

PapyReqDiaDi2FileStrGenerator
-NAME_TOKENS_TEMP_NUM : int = 50
-XML_TAG_VERSION_DEFT_VAL : string = 1.0
-nameTokens[] : object
-valHeads[] : object
-papyDiaXMLidGenerator : object

圖 十一、di2 檔案產生器模組之類別圖

- PapyReqDiaDi2FileStrGenerator
 - Papyrus 系統內需求圖 di2 檔案之字串產生者。
 - 負責產生需求圖 di2 檔案所定義的標籤資訊。
 - 儲存 Papyrus 系統內需求圖 di2 檔案的標籤字串單元之數量限制 (NAME_TOKENS_TEMP_NUM =50，暫定 50 個)。
 - 儲存 xml 標籤參數預設值(例如：
XML_TAG_VERSION_DEFT_VAL="1.0"，XML 版本預設為 1.0 版)。
 - 儲存需求圖 di2 檔案各種標籤字串單元的名稱(nameTokens[])，型態為字串陣列)。
 - 儲存需求圖 di2 檔案各種標籤字串單元的名稱寫入值(valHeads[])，型態

為字串陣列)。

- 實作 PapyDiaXMlidGenerator 類別(papyDiaXMlidGenerator 物件，Papyrus 系統內圖形之 XMI 識別碼產生者)，負責產生 XMI 識別碼以便與 Papyrus 系統相容。

4.3.2 uml 檔案產生器模組(uml file generator module)

此模組在系統中的角色與工作，與 4.3.1 節「di2 檔案產生器模組」相似。亦即：以符合 Papyrus 系統 uml 檔案的綱要(schema)的編寫方式，負責將各種 XML 標籤與屬性的欄位值記錄為 XML 格式的文字檔案；必須依照標籤內容資訊順序，串接底下所有標籤及屬性後，回傳整段 XML 字串。從靜態結構來看，其類別圖如下：

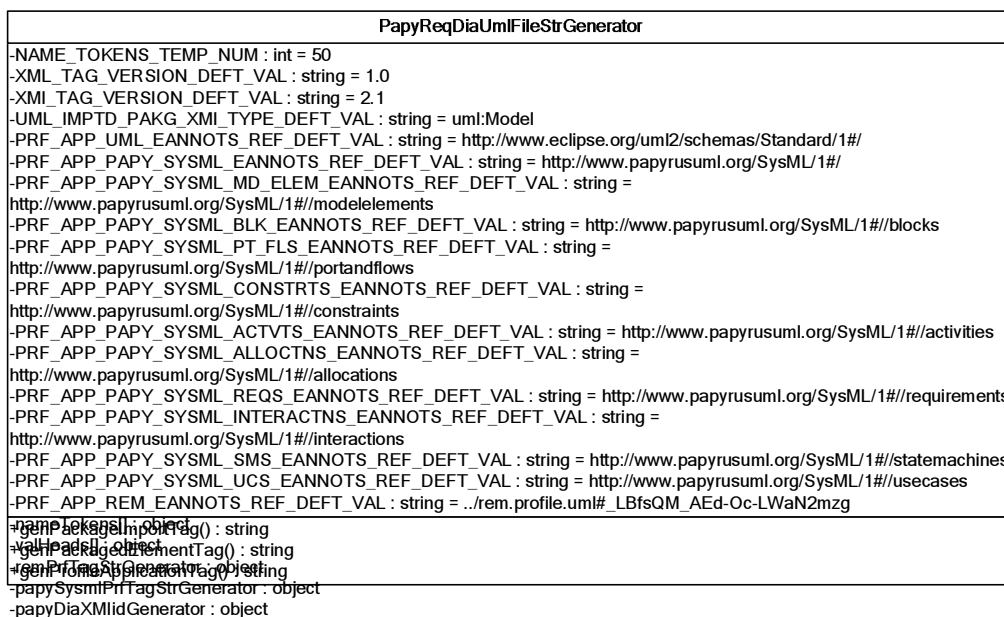


圖 十二、uml 檔案產生器模組之類別圖

- PapyReqDiaUmlFileStrGenerator
 - Papyrus 系統內需求圖 uml 檔案之字串產生者。
 - 是負責產生需求圖 uml 檔案所定義的標籤資訊的主要物件。
 - 儲存 Papyrus 系統內需求圖 uml 檔案的標籤字串單元之數量限制 (NAME_TOKENS_TEMP_NUM =50，暫定 50 個)。

- 儲存 xml 標籤參數預設值(例如：
XML_TAG_VERSION_DEFT_VAL="1.0"，XML 版本預設為 1.0 版)。
- 儲存 XMI 標籤參數預設值(例如：
XMI_TAG_VERSION_DEFT_VAL="2.1"，XMI 版本預設為 2.1 版)。
- 儲存 UML 匯入套件預設值(例如：
UML_IMPTD_PAKG_XMI_TYPE_DEFT_VAL="uml:Model"，圖形預設
為 uml 套件的 Model 元素)。
- 儲存 UML 概要參照預設值(例如：
PRF_APP_UML_EANNOTS_REF_DEFT_VAL="http://www.eclipse.org/u
ml2/schemas/Standard/1#/"，預設 eAnnotations 標籤之網址參照)。
- 儲存 Papyrus uml4sysml 概要參照預設值(例如：
PRF_APP_PAPY_SYSML_EANNOTS_REF_DEFT_VAL="http://www.pa
pyrusuml.org/SysML/1#/"，意義同上)。
- 儲存 Papyrus uml4sysml modelelements 概要參照預設值(例如：
PRF_APP_PAPY_SYSML_MD_ELEM_EANNOTS_REF_DEFT_VAL="http://www.papyrusuml.org/SysML/1#//modelelements"，意義同上)。
- 儲存 Papyrus uml4sysml blocks 概要參照預設值(例如：
PRF_APP_PAPY_SYSML_BLK_EANNOTS_REF_DEFT_VAL="http://w
ww.papyrusuml.org/SysML/1#//blocks"，意義同上)。
- 儲存 Papyrus uml4sysml portandflows 概要參照預設值(例如：
PRF_APP_PAPY_SYSML_PT_FLS_EANNOTS_REF_DEFT_VAL="http:
//www.papyrusuml.org/SysML/1#//portandflows"，意義同上)。
- 儲存 Papyrus uml4sysml constraints 概要參照預設值(例如：
PRF_APP_PAPY_SYSML_CONSTRTS_EANNOTS_REF_DEFT_VAL="http://www.papyrusuml.org/SysML/1#//constraints"，意義同上)。
- 儲存 Papyrus uml4sysml activities 概要參照預設值(例如：

PRF_APP_PAPY_SYSML_ACTVTS_EANNOTS_REF_DEFT_VAL="http://www.papyrusuml.org/SysML/1#/activities"，意義同上)。

- 儲存 Papyrus uml4sysml allocations 概要參照預設值(例如：

PRF_APP_PAPY_SYSML_ALLOCTNS_EANNOTS_REF_DEFT_VAL="http://www.papyrusuml.org/SysML/1#/allocations"，意義同上)。

- 儲存 Papyrus uml4sysml requirements 概要參照預設值(例如：

PRF_APP_PAPY_SYSML_REQS_EANNOTS_REF_DEFT_VAL="http://www.papyrusuml.org/SysML/1#/requirements"，意義同上)。

- 儲存 Papyrus uml4sysml interactions 概要參照預設值(例如：

PRF_APP_PAPY_SYSML_INTERACTNS_EANNOTS_REF_DEFT_VAL="http://www.papyrusuml.org/SysML/1#/interactions"，同上)。

- 儲存 Papyrus uml4sysml statemachines 概要參照預設值(例如：

PRF_APP_PAPY_SYSML_SMS_EANNOTS_REF_DEFT_VAL="http://www.papyrusuml.org/SysML/1#/statemachines"，意義同上)。

- 儲存 Papyrus uml4sysml usecases 概要參照預設值(例如：

PRF_APP_PAPY_SYSML_UCS_EANNOTS_REF_DEFT_VAL="http://www.papyrusuml.org/SysML/1#/usecases"，意義同上)。

- 儲存 REM 概要參照預設值(例如：

PRF_APP_REM_EANNOTS_REF_DEFT_VAL="../rem.profile.uml#_LBfsQM_AEd-Oc-LWaN2mzg"，意義同上)。

- 儲存需求圖 uml 檔案各種標籤字串單元的名稱(nameTokens[])，型態為字串陣列)。

- 儲存需求圖 uml 檔案各種標籤字串單元的名稱寫入值(valHeads[])，型態為字串陣列)。

- 實作 REMprfTagStrGenerator 類別(remPrfTagStrGenerator 物件，是負責產生 REM 概要所定義的標籤資訊的主要物件)，負責產生 REM 概要所

定義的標籤資訊。

- 實作 PapySysmlPrfTagStrGenerator 類別(papySysmlPrfTagStrGenerator 物件，Papyrus 系統的 Sysml 概要所定義的標籤資訊之字串產生者)，負責產生 Sysml 概要所定義的標籤資訊。
- 實作 PapyDiaXMlidGenerator 類別(papyDiaXMlidGenerator 物件，Papyrus 系統內圖形之 XMI 識別碼產生者)，負責產生 XMI 識別碼以便與 Papyrus 系統相容。
- 依照需求圖 uml 檔案之 packageImport 標籤內容資訊順序，串接底下所有標籤及屬性後，回傳整段 XML 字串(genPackageImportTag 方法)。
- 依照需求圖 uml 檔案之 packagedElement 標籤內容資訊順序，串接底下所有標籤及屬性後，回傳整段 XML 字串(genPackagedElementTag 方法)。
- 依照需求圖 uml 檔案之 profileApplication 標籤內容資訊順序，串接底下所有標籤及屬性後，回傳整段 XML 字串(genProfileApplicationTag 方法)。

從動態結構來看，SysML 圖形產生模組之產生 uml 檔及 di2 檔的循序圖如下：在產生 uml 檔的部份，ReqItemPrototype 呼叫 PapyReqDiaUmlFileStrGenerator 然後接著產生 uml 檔最外層的 uml 模型標籤，最後回傳存檔完成的訊息；在產生 di2 檔的部份，則由 ReqItemPrototype 呼叫 PapyReqDiaDi2FileStrGenerator 然後產生 di2 檔最外層的 di2 圖形標籤，最後回傳存檔完成的訊息。

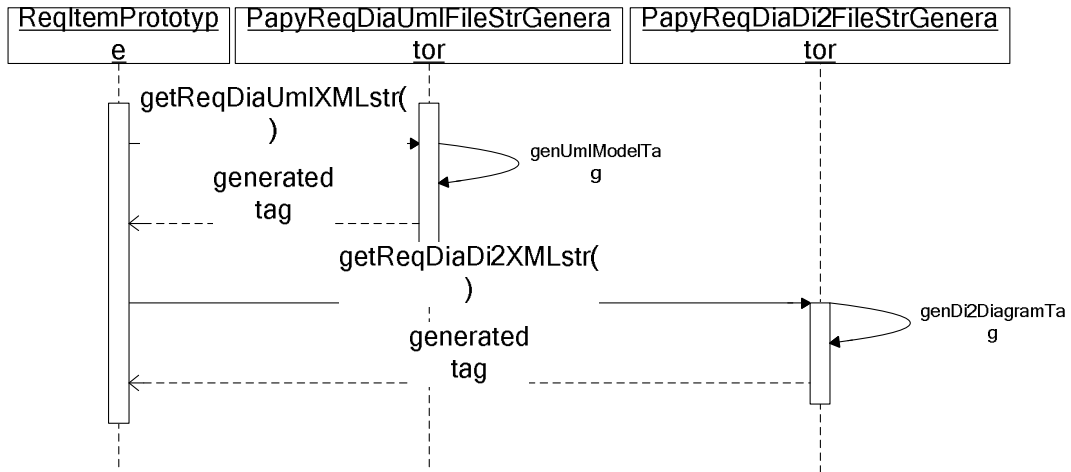


圖 十三、SysML 圖形產生模組之產生 uml 檔及 di2 檔的循序圖

從動態結構來看，SysML 圖形產生模組之產生 userReq 概要記錄及需求圖概要記錄的循序圖如下：在產生 userReq 概要記錄部份，由 ReqItemPrototype 呼叫 REMprfTagStrGenerator 然後產生 userReq 標籤的資訊，並將其記錄起來；在產生需求圖概要記錄部份，由 ReqItemPrototype 呼叫 PapySysmlPrfTagStrGenerator 然後產生 RequirementsReq 標籤的資訊，最後將其記錄起來。

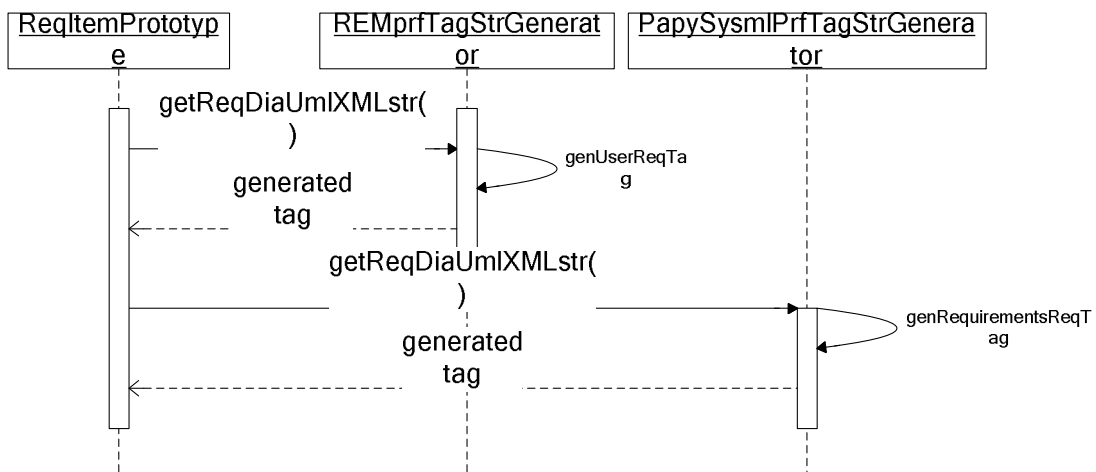


圖 十四、SysML 圖形產生模組之產生 userReq 概要記錄及需求圖概要記錄的循序圖

第五章 個案研究

本研究利用 DVR (Digital Video Recording) 系統作為案例，說明工具實際的使用狀況。DVR 系統是基於嵌入式 ARM 處理器，利用數個攝影機拍攝影像，用多核心處理器平行接收及做存放的處理，可即時傳給遠端使用者觀看。系統架構圖如下：

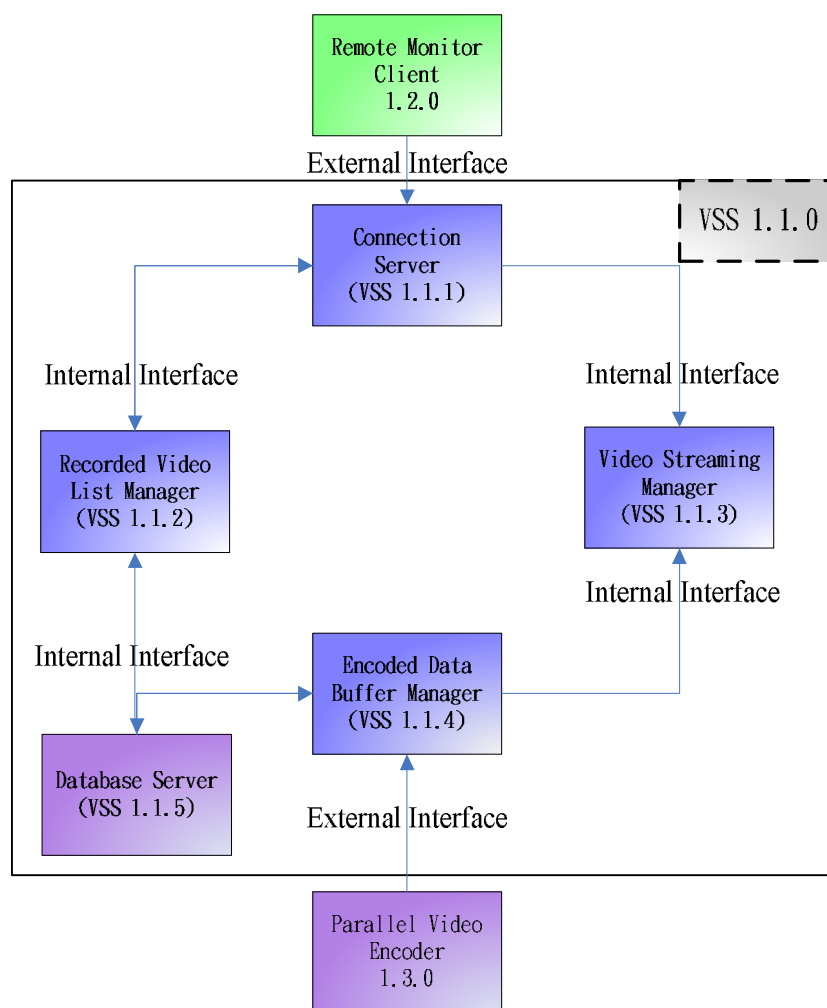


圖 十五、DVR 架構圖

5.1 輸入需求

以下針對 Video Streaming Server(VSS)子系統來討論。VSS 提供三種服務：

1. 提供遠端連線服務。
2. 提供遠端線上播放即時影像或播放存檔影像服務。
3. 提供 PVE 儲存加密後的訊框的服務。

Video Streaming Server 主要的功能是讓使用者透過伺服器提供的連線功能

來進行工作，當使用者與伺服器建立連線之後，系統立即從 Data Encoding Server 取得數位攝影機所錄製的影像畫面以及相關資訊，VSS 包含了五個元件，分別是 Connection Server (CS)、Recorded Video List Manager (RVLM)、Video Streaming Manager (VSM)、Encoded Data Buffer Manager (EDBM)以及 Database Server (DS)，用以上的五個元件來當作實現工作平行的功能。

預設嵌入式目標板有 2 顆處理器、各有 4 個核心，記憶體為 1GB，電力消耗為 800watt，作業系統為 Linux 平台。

VSS 之功能性需求如下：

表 五、VSS 子系統功能性需求範例

編號	描述
VSS-FNR-001	Connection Server 最多可以同時處理 5 個客戶端連線。
VSS-FNR-002	Connection Server 要能夠依照客戶端的要求去呼叫相對應的元件來提供服務。
VSS-FNR-003	Status Manager 要能夠提供實體資料庫伺服器上現有的檔案清單及數位攝影機清單。
VSS-FNR-004	Database Server 可以將加密後的影像存成完整的檔案。
VSS-FNR-005	Video Streaming Manager 可以提供儲存的影像檔案給客戶端下載。
VSS-FNR-006	Video Streaming Manager 可以同時進行 2 個以上的 80frame/s 的即時串流傳輸。
VSS-FNR-007	Encoded Data Buffer Manager 管理暫存的即時影像並分別將暫存的資料提供做即時傳輸，以及將資料送給 Database Server 以建立和儲存完整的影像檔案。

系統非功能性需求如下：

表 六、VSS 子系統非功能性需求範例

編號	描述
VSS-NONFNR-001	若遇當機狀況，必須 5 秒內重新啟動至待機狀態。

系統在效能方面之需求為：

表 七、VSS 子系統效能需求範例

編號	描述
VSS-HIGH_PRPMC-001	Video Streaming Server 必須要在 5 秒內建立客戶端連線。

VSS-HIGH_PRPMC-002	Video Streaming Server 必須維持每一個串流的傳輸都在 50bps 以上。
--------------------	---

系統之其他需求為：

表 八、VSS 子系統其他類型需求範例

編號	描述
VSS-OTHER-001	在目標板之外，另設置一個實體資料庫伺服器，負責分類儲存加密後的影像檔案，並提供查詢功能。

輸入上列需求後，編輯器畫面如下圖所示：

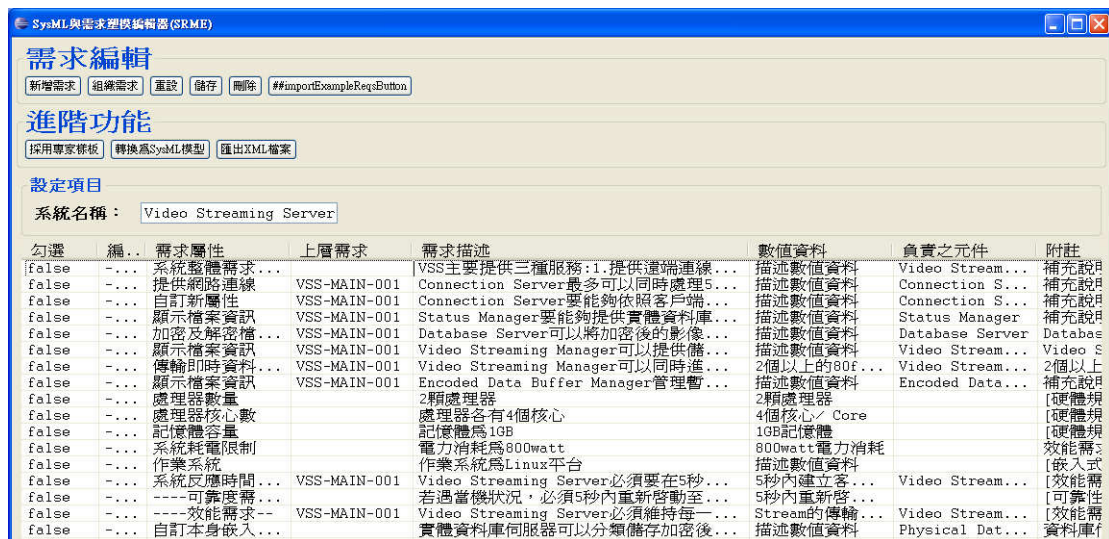


圖 十六、輸入需求及條列清單

5.2 建立需求階層

透過需求編輯表格中的「上層需求」欄位，提供使用者輸入需求之間的關連性。

5.3 產生 SysML 圖形

工具可產生基本 SysML 圖形的需求方塊，如下所示：

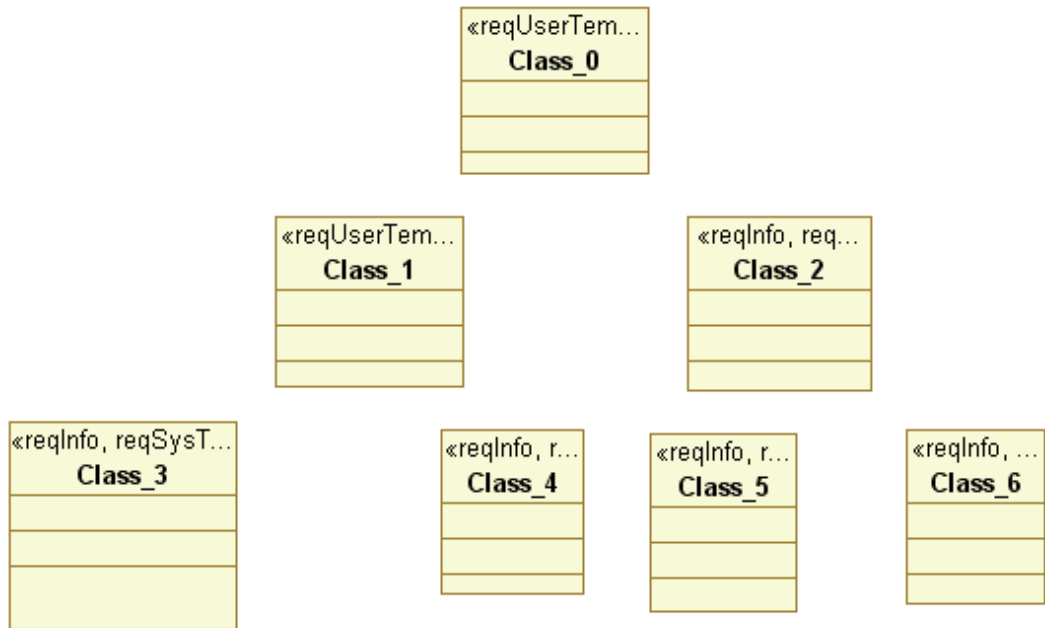


圖 十七、產生基本 SysML 圖形的需求方塊

之後，再由使用者進一步修改，可得到以下的 SysML 需求圖：

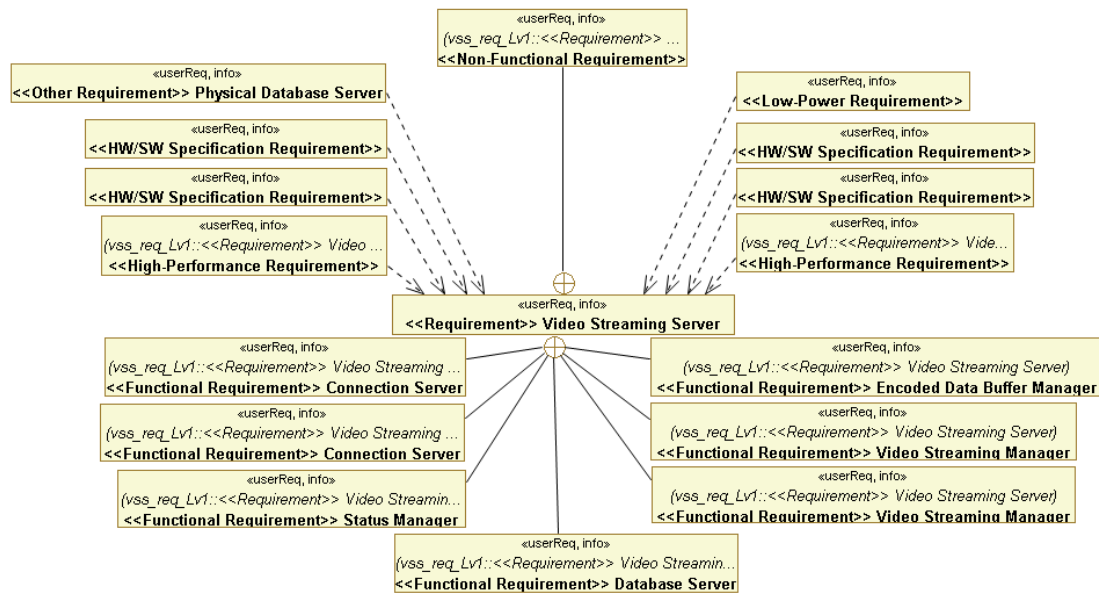


圖 十八、使用者修改後的 SysML 圖形

在 Papyrus UML 系統上點選某一需求後(以 Status Manager 為例)，即可於下方之樹狀列表進一步修改編輯。如下圖所示：

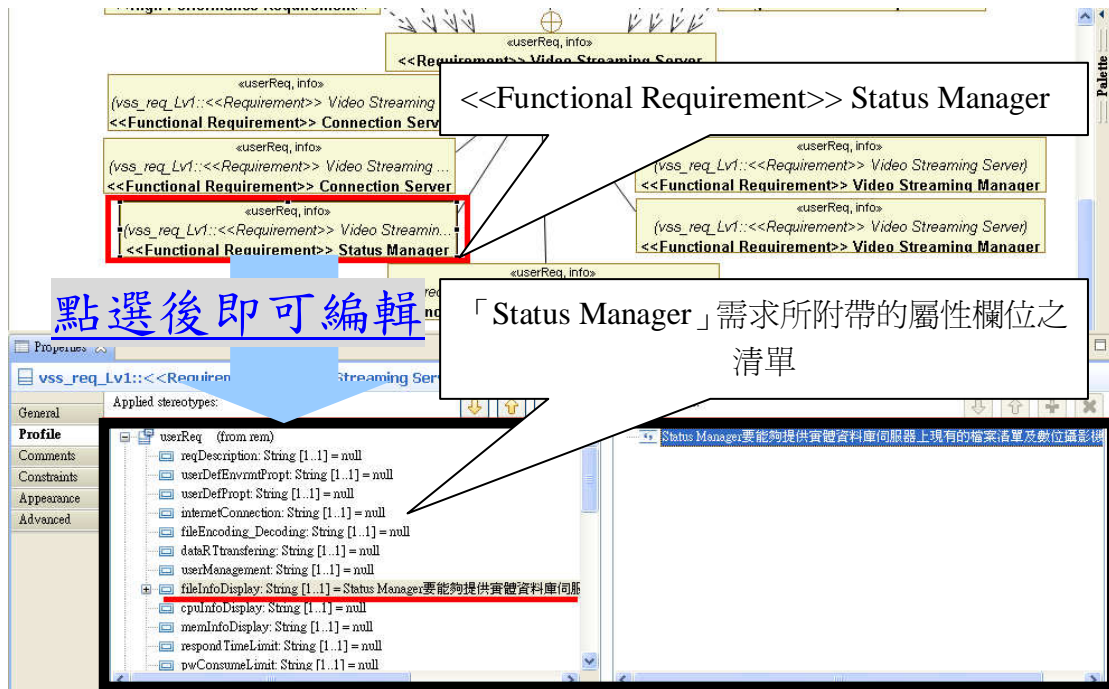


圖 十九、SysML 圖形附帶的需求模板各屬性可供修改

5.4 需求模板之屬性資訊附加於 SysML 圖形中

本工具在 Papyrus UML 平台的參考概要(profile)機制輔助下，將需求模板之屬性資訊附加於 SysML 圖型中，並利用以 XML 格式記錄這些資訊，以下為各類需求之檔案記錄：

表 九、XML 檔案記錄(針對 Connection Server 及 Status Manager)

編號	需求名稱
VSS-MAIN-001	Video Streaming Server
主要需求資訊	
VSS 主要提供三種服務：1. 提供遠端連線服務。2. 提供遠端線上播放即時影像或播放存檔影像服務。3. 提供PVE儲存加密後的訊框的服務。	
XML 記錄	
<pre> <rem:userReq xmi:id="_d12eUJpmEd-136ODb1LRBg" reqDescription="VSS 主要提供三種服務：1.提供遠端連線服務。2.提供遠端線上播放即時影像或播放存檔影像服務。3.提供 PVE 儲存 encoded frame 的服務。" base_Class="_vtVSSJplEd-136ODb1LRBg"/> <rem:info xmi:id="_d12eUZpmEd-136ODb1LRBg" id="_g1HbNWxGEd0HIHS9HDQW4U" reqID="VSS-MAIN-001" reqKind="Main" reqName="Video Streaming Server" reqShortName="VSS" currentVersion="1.0.0" projectName="Digital Video Recording" </pre>	

establishDate="2010.7.29" establishUser="C.H." baseClass="_s3F4NOSyEd000KNA6vK7ca" base_Class="_vtVSsJplEd-1360Db1LRBg" />	
編號	需求名稱
VSS-FNR-001、 VSS-FNR-002	Connection Server
主要需求資訊	
<ul style="list-style-type: none"> ● Connection Server最多可以同時處理5個客戶端連線。 ● Connection Server要能夠依照客戶端的要求去呼叫相對應的元件來提供服務。 	
XML記錄	
<pre> <rem:userReq xmi:id="_78qZgJppEd-1360Db1LRBg" internetConnection="Connection Server最多可以同時處理5個客戶端 連線。" base_Class="_j5hwoJppEd-1360Db1LRBg" /> <rem:info xmi:id="_78qZgZppEd-1360Db1LRBg" id="_zUrdJWxYEd8oYHDvPG8eDB" reqID="VSS-FNR-001" reqKind="Functional" reqName="Connection Server" reqShortName="CS" currentVersion="1.0.0" projectName="Digital Video Recording" establishDate="2010.7.29" establishUser="C.H." baseClass="_BlaasJskEd3uFnPNXSJRBo" base_Class="_j5hwoJppEd-1360Db1LRBg" /> <rem:userReq xmi:id="_hc4yQJptEd-1360Db1LRBg" userDefPropt="Connection Server要能夠依照客戶端的要求去呼叫相對 應的元件來提供服務。" base_Class="_UpxsIJptEd-1360Db1LRBg" /> <rem:info xmi:id="_hc4yQZptEd-1360Db1LRBg" id="_X0XCWxPEd1TK4WaVvDRhU" reqID="VSS-FNR-002" reqKind="Functional" reqName="Connection Server" reqShortName="CS" currentVersion="1.0.0" projectName="Digital Video Recording" establishDate="2010.7.29" establishUser="C.H." baseClass="_obI8VUx3Ed5gFbXv2CM11J" base_Class="_UpxsIJptEd-1360Db1LRBg" /> </pre>	
編號	需求名稱
VSS-FNR-003	Status Manager
主要需求資訊	
<ul style="list-style-type: none"> ● Status Manager要能夠提供實體資料庫伺服器上現有的檔案清單及數位攝影機清單。 	

XML記錄
<pre> <rem:userReq xmi:id="_q--XEJpuEd-136ODb1LRBg" fileInfoDisplay="Status Manager要能夠提供實體資料庫伺服器上現有的檔案清單及數位攝影機清單。" base_Class="_kc2G0JpuEd-136ODb1LRBg" /> <rem:info xmi:id="_q--XEZpuEd-136ODb1LRBg" id="_CWHJqWxAEd0DJTBCSsz42Ws" reqID="VSS-FNR-003" reqKind="Functional" reqName="Status Manager" reqShortName="SM" currentVersion="1.0.0" projectName="Digital Video Recording" establishDate="2010.7.29" establishUser="C.H." baseClass="_XNzqOUTnEd08WwUCm3e8wv" base_Class="_kc2G0JpuEd-136ODb1LRBg" /> </pre>

以下針對非功能性需求來展示 XML 檔案。

表 十、XML 檔案記錄(針對非功能性需求)

編號	需求名稱
VSS-NONFNR-001	無
主要需求資訊	
<ul style="list-style-type: none"> ● 若遇當機狀況，必須5秒內重新啟動至待機狀態。 	
XML記錄	
<pre> <rem:userReq xmi:id="_Kg3rYJrOEd-136ODb1LRBg" reliability="若遇當機狀況，必須5秒內重新啟動至待機狀態。" base_Class="_E0_GoJrOEd-136ODb1LRBg" /> <rem:info xmi:id="_Kg3rYZrOEd-136ODb1LRBg" id="_XVcBMWxXEd2LNwvav9Y3Yt" reqID="VSS-NONFNR-001" reqKind="Non-Functional" currentVersion="1.0.0" projectName="Digital Video Recording" establishDate="2010.7.29" establishUser="C.H." baseClass="_BeRIwKfIEdlxRWiqyETwIT" base_Class="_E0_GoJrOEd-136ODb1LRBg" /> </pre>	

以下針對效能需求及其它需求來展示 XML 檔案。

表 十一、XML 檔案記錄(針對效能需求及其它需求)

編號	需求名稱
VSS-HIGH_PRPMC-001	無
主要需求資訊	
<ul style="list-style-type: none"> ● Video Streaming Server必須要在5秒內建立客戶端連線。 	

XML記錄	
<pre><rem:userReq xmi:id="_LahbwJrEEed-136ODb1LRBg" respondTimeLimit="Video Streaming Server必須要在5秒內建立客戶 端連線。" base_Class="_HjKvsJrEEed-136ODb1LRBg" /> <rem:info xmi:id="_LahbwZrEEed-136ODb1LRBg" id="_cWPk7WxBEd6500KnWneRCF" reqID="VSS-HIGH_PRPMC-001" reqKind="High-Performance" currentVersion="1.0.0" projectName="Digital Video Recording" establishDate="2010.7.29" establishUser="C.H." baseClass="_E9ejBTa0Ed38TRCkCguQ3P" base_Class="_HjKvsJrEEed-136ODb1LRBg" /></pre>	
編號	需求名稱
VSS-HIGH_PRPMC-002	無
主要需求資訊	
<ul style="list-style-type: none"> ● Video Streaming Server必須維持每一個串流的傳輸都在50bps以上。 	
XML記錄	
<pre><rem:userReq xmi:id="_0KrbkJrEEed-136ODb1LRBg" dataRTtransferring="Video Streaming Server必須維持每一個串流的 傳輸都在50bps以上。" base_Class="_weMiwJrEEed-136ODb1LRBg" /> <rem:info xmi:id="_0KrbkZrEEed-136ODb1LRBg" id="_merdnWxWEd3DA8vDgS7aGl" reqID="VSS-HIGH_PRPMC-002" reqKind="High-Performance" currentVersion="1.0.0" projectName="Digital Video Recording" establishDate="2010.7.29" establishUser="C.H." baseClass="_rlNhiJ3YEd8WQCsQPEXvle" base_Class="_weMiwJrEEed-136ODb1LRBg" /></pre>	
編號	需求名稱
VSS-OTHER-001	無
主要需求資訊	
<ul style="list-style-type: none"> ● 實體資料庫伺服器可以分類儲存加密後的影像檔案，並提供查詢功能。 	
XML記錄	
<pre><rem:userReq xmi:id="_m3CIUJrREd-136ODb1LRBg" userDefEnvrmtPropt="實體資料庫伺服器可以分類儲存加密後的影像檔 案，並提供查詢功能。" base_Class="_k2SYkJrREd-136ODb1LRBg" /> <rem:info xmi:id="_m3CIUZrREd-136ODb1LRBg" id="_-zFakWxKEd6bSDl45MiaJs" reqID="VSS-OTHER-001" reqKind="Other" reqName="Physical Database Server" reqShortName="PDS" currentVersion="1.0.0"</pre>	


```
previousVersion="NULL" projectName="Digital Video  
Recording" establishDate="2010.7.29" establishUser="C.H."  
baseClass="_vFzO2FcgEd3lL5huUFwBbg"  
base_Class="_k2SYkJrREd-1360Db1LRBg" />
```

第六章 結論與未來方向

本研究實作一套基於需求模板(Requirement Template)讓使用者撰寫需求的編輯器。針對多核心嵌入式系統，輔助使用者定義需求，並產生基本架構的 SysML 圖形，於是更容易銜接底層程式碼的產生。

此編輯器利用內部建立的程式機制以及與使用者溝通的介面來完成。首先，在機制方面，透過編輯器中建立的物件式需求雛形，可以方便獨立處理每個需求，將需求編號、屬性、文字內容、數值內容以及 XML 記錄資料的操作與特性都完整地定義下來，使得資料輸入與操作的方法均統一，更容易找出錯誤。另外，透過參考概要(profile)的輔助機制，使得 Papyrus UML 系統能辨識附帶在 SysML 圖檔上的需求屬性資訊，這些資訊是基於需求模板而產生，能夠針對多核心嵌入式系統的屬性來編寫，於是能輔助多核心嵌入式系統的開發。

在介面方面，主要透過清單來控制與整理，讓使用者容易條列需求並檢視，而功能均透過按鈕來觸發，使得操作也十分容易。最後，透過轉換 SysML 圖形，使得文字的需求變成圖形化的表示方式，可以提高可閱讀性，並且能定義需求之間的關聯，增加整體需求的整合程度。

未來的研究首先朝向加入資料庫輔助需求規格之即時分析，透過搜集過去開發的系統資料，經過特殊的關聯整理之後，即可針對單一需求對各個規格項目附註預期效益，使得進行開發之任何時刻，均能夠檢視目前所選用之軟硬體規格與既有設計搭配之後所能達成之效益。其次，繼續延伸學者在 XUM[18]研究的經驗，為了達到更好的需求追蹤與管理，利用建立需求溯源性矩陣之資料結構在工具中，使得需求之間能雙向追溯，讓後續維護的工作更容易。

參考文獻

- [1] 朱正忠、張志宏、陳有韋，「以SysML為基礎的系統需求研究方法」，東海大學資訊工程研究所論文，2009。
- [2] Michel dos Santos Soares; Jos Vrancken; "Model-Driven User Requirements Specification using SysML", Journal of Software, 2008, Vol.3, No.6.
- [3] 黃有評、賴史詠，「以模板為基礎之車牌辨識系統設計與應用」，大同大學資訊工程研究所論文，2003。
- [4] A. Aurum; C. Wohlin; "Requirements Engineering: Setting the Context", Engineering and Managing Software Requirements, Springer-Verlag, 2005, pp. 1-15.
- [5] Nassis, V.; Rajugan, R.; Dillon, T.S.; Rahayu, W.; "A requirement engineering approach for designing XML-view driven, XML document warehouses", 29th Annual International Computer Software and Applications Conference, 2005, Vol.2, pp.388-395.
- [6] Matthew R. B. Hardy; David F. Brailsford; Peter L. Thomas; "Creating structured PDF files using XML templates", Proceedings of the 2004 ACM symposium on Document engineering, 2004, pp.99-108.
- [7] Renault, S.; Mendez-Bonilla, O.; Franch, X.; Quer, C.; "PABRE: Pattern-based Requirements Elicitation", Third International Conference on Research Challenges in Information Science, 2009, pp. 81-92.
- [8] Keen Ngee Loo, Sai Peck Lee, "Representing design pattern interaction roles and variants", Second International Conference on Computer Engineering and Technology, 2010, Vol.6, pp. 470-474.
- [9] De Lucia, A.; Deufemia, V.; Gravino, C.; Risi, M.; "An Eclipse plug-in for the detection of design pattern instances through static and dynamic analysis", IEEE

- International Conference on Software Maintenance, 2010, pp. 1-6.
- [10] Pandel, A.; Gupta, M.; Tripathi, A.K.; "DNIT — A new approach for design pattern detection", International Conference on Computer and Communication Technology, 2010, pp. 545-550.
- [11] OMG. Systems Modeling Language (SysML). <http://www.omg.sysml.org/>
- [12] OMG. The Unified Modeling Language (UML). <http://www.uml.org>
- [13] Costa, T.; Sampaio, A.; Alves, G.; "Using SysML in Systems Design", International Conference on Information Management, Innovation Management and Industrial Engineering, 2009, Vol.4, pp. 615-618.
- [14] Yue Guo; Chakrapani Rao, A.; Jones, R.P.; "Architectural and Functional Modelling of an Automotive Driver Information System Using SysML", IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, 2008, pp. 552-557.
- [15] Kawahara, R.; Nakamura, H.; Dotan, D.; Kirshin, A.; Sakairi, T.; Hirose, S.; Ono, K.; Ishikawa, H.; "Verification of embedded system's specification using collaborative simulation of SysML and simulink models", International Conference on Model-Based Systems Engineering, 2009, pp. 21-28.
- [16] Nikolaidou, M.; Dalakas, V.; Mitsi, L.; Kapos, G.-D.; Anagnostopoulos, D.; "A SysML Profile for Classical DEVS Simulators", The Third International Conference on Software Engineering Advances, 2008, pp. 445-450.
- [17] OMG, Model-Driven Architecture(MDA), <http://www.omg.org/mda/>
- [18] C.W. Lu; W.C. Chu; C.H. Chang; W.D. Lian; D. L. Yang; "Integrating Divers Paradigms in Evolution and Maintenance by an XML-based Unified Model", Journal of Software Maintenance and Evolution, 2003, Vol.15, Issue.3, pp. 111-144.