

東海大學資訊工程研究所

碩士論文

指導教授：朱正忠 教授

MARTE 建模與需求追蹤性用於即時嵌入式系
統開發

Apply MARTE modeling and requirements
traceability to Real-time Embedded systems
development

研究生：黃景宜

中華民國一百零一年六月三十日

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 黃景宜 所提之論文

MARTE 建模與需求追蹤性用於即時嵌入式系統
開發

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

盧志偉

簽章

委員

楊朝棟

張志忠

陳瑞晃

指導教授

朱正忠

簽章

中華民國 101 年 6 月 29 日

摘要

隨著嵌入式系統的發展，嵌入式軟體工程的需求逐年增加，使用模型驅動開發可以加快專案開發速度，統一塑模語言為一種常用於模型驅動開發的核心建模語言，並針對嵌入式系統，已有 SysML 及 MARTE Profiles 等適用於嵌入式系統專案模型出現。

延續過往的 XUM 研究成果，讓軟體專案中的軟體模型以不同軟體工程時期分類，並拿出模型中的重要資訊，如 SysML 中的需求資訊或是 MARTE 可以設定資源及即時(real-time)等限制資訊，讓專案盡量接近現實中的嵌入式系統，更符合使用者需求。

我們從不同的模型拿取複雜的資訊簡化並物件化，比對需求圖中的需求條件與其他階層的限制條件，建立需求、限制條件間的關聯性，使得軟體專案中間資訊達成追蹤性，減少建模過程中的人為錯誤及疏失。

我們使用 Eclipse 為系統環境去實作系統，它可支援不同的軟體模型建模以及 XML 文件的讀取，並以其為環境建構一外掛工具，可搭配 Papyrus 模型工具。實作一個需求追蹤樹，可快速追蹤專案中的需求及限制資訊，從面板上可快速連接模型圖，方便使用者可在同一平台統整嵌入式系統專案。

關鍵詞：模型驅動、統一塑模語言、SysML、MARTE、追蹤性

Abstract

With the development of embedded system, an increased embedded Software Engineering demand year by year , and we can speed up project progress with Model-Driven Development(MDD). UML is a core language for MDD and the SysML and MARTE Profiles are using in embedded system models.

Based on the previous XUM research results, we assort models by software phase and capture impartment information of models, such as requirement information of Requirement Diagram or resource and real-time information of MARTE, We try to let model project to match real embedded system, let project to meet users needs.

We capture and simplified complex information from different model and create objects, than compared with requirement conditions of Requirement Diagram and constraints condition from other diagram, to create relation link between requirement and constraint, and achieve information traceability of software models, reduce human errors and mistakes in the modeling process.

We use Eclipse-based environment to implement the system, that can support software modeling and XML files, and we create a plug-in tools with the environment, and it also support Papyrus tool to create model. We implement a requirement traceability tree, let user can trace requirement and constraint information of model project, and it can connect model diagram quickly from panel, and user can easy to integrate the embedded system project on the same platform.

Keywords: Model-Driven, UML, SysML, MARTE, Traceability

目錄

摘要.....	II
Abstract.....	III
目錄.....	IV
第一章 導論	1
1.1 前言.....	1
1.2 研究動機	2
1.3 研究目的	4
1.4 章節安排	4
第二章 背景知識與相關研究	6
2.1 XML (extensible Markup Language).....	6
2.2 統一塑模語言(UML)	8
2.3 SysML	8
2.4 MARTE.....	11
2.5 XUM.....	13
2.6 VSL(Value Specification Language)	15
第三章 研究方法	17
3.1 延伸於 SysML 及 MARTE 的 XUM 整合模型.....	18
3.2 SysML 的需求及 MARTE 的限制條件抓取.....	20
3.3 資訊追蹤性及整合	24
第四章 案例研究與評估	30

4.1 專案模型	31
4.2 系統畫面	35
第五章 結論及未來展望	41
參考文獻	43

圖目錄

圖 一 經由 cmd 開啟 Sedna 資料庫，並啟動其中的 vmc 資料庫	7
圖 二 SDBAdmin 軟體表現 Sedna 資料庫	8
圖 三 SysML Diagram Types	9
圖 四 DVR 的需求圖	10
圖 五 papyrus 的 SysML 需求圖屬性欄	10
圖 六 Requirement Diagram 的 UML 檔	11
圖 七 MARTE 的架構	12
圖 八 MARTE Model	13
圖 九 子模型與 XUM 經 XUMM 轉換的關係	14
圖 十 XUM 整合模型的結構	15
圖 十一 XUM 資訊及鏈結架構	19
圖 十二 限制屬性定義	19
圖 十三 從專案中抓取資訊放入 XUM 中的流程	21
圖 十四 需求圖中的物件內的各種設定值	22
圖 十五 出現在不同圖中的限制條件	23
圖 十六 UML 檔中需拿取的重要資訊	23
圖 十七 需求圖的 UML 檔中的資訊	24
圖 十八 MARTE 類型圖中常見的限制資訊	24
圖 十九 XUM 中的限制鏈結結構	24
圖 二十 限制鏈結的建立及判別流程	25
圖 二十一 追蹤樹及其完整資訊	26
圖 二十二 活動圖的 MARTE 範例	27
圖 二十三 循序圖的 MARTE 範例	27
圖 二十四 關係鏈結與限制條件間的關聯性	28
圖 二十五 系統流程圖	31
圖 二十六 DVR 的需求圖	32
圖 二十七 DVR 的需求圖中的需求條件	33
圖 二十八 DVR 的案例圖	33
圖 二十九 DVR 的類別圖	34
圖 三十 DVR 的活動圖	35
圖 三十一 DVR 的循序圖	35
圖 三十二 主畫面示意圖	36
圖 三十三 主畫面列表	37
圖 三十四 需求追蹤樹及其資訊列表	38
圖 三十五 需求追蹤樹展開	39

表目錄

表 一 VSL 定義及範例列表.....	16
----------------------	----

第一章 導論

1.1 前言

軟體工程是一門不斷持續發展的學科，一開始軟體工程多用於在電腦上的軟體專案，嵌入式系統(embedded system)[1]因為限制於硬體如硬體體積、記憶體大小、開發平台等問題，在早期開發需要花上許多時間，開發效率不高；近年來因為硬體研發技術進步，硬體已向著體積愈小效能愈高的目標發展，能表現出來的效能及功能性不同以往，嵌入式系統已是現代人生活中更不可缺少的存在，例如智慧型家電、智慧型手機等，也因應複雜且龐大的嵌入式軟體需求，嵌入式系統專案的數量及規模增加。

因軟體專案開發時間長，且嵌入式系統專案更不同於傳統軟體系統，在不同設計之間的統合、細部分工等作業程式使得嵌入式系統開發與研究者無法以一個有協同性的開發，此外因為系統會因硬體、資源實際運行時間限制、不同平台而有所變化，而使得嵌入式系統開發困難、成本居高不下且品質浮動。

近年發展起來的模型驅動開發方式(Model-Driven Development, MDD)[2]，可加速軟體開發流程並減少成本，其中以統一塑模語言(Unified Modeling Language, UML)[3]為最主要的常用塑模語言。

物件管理組織(Object Management Group, OMG)基於系統及嵌入式需求定義了 SysML(systems Modeling Language)[4]及 MARTE(Modeling and Analysis of Real-Time and Embedded systems)[5]，是以 UML 為基礎進行擴充，可制作系統及需求模型等可應用在嵌入式系統中。

應用此二新類型的專案模型，可製作較為接近現實所需求的嵌入式軟體專案，對使用者來說實用性較高且實作時更利於減少硬體問題。

1.2 研究動機

因為嵌入式系統的開發困難，且嵌入式系統開發的相關工具並不充足，為了研發嵌入式專案的完整性，我們需要一些輔助工具，並為了減少過長的專案開發時間，軟體專案再利用也受到相當的重視。

軟體工程的生命週期包含了需求、設計、實作、測試維護等五階段，其中統一塑模語言(UML)多應用於需求、設計階段，而需求階段因為在開發前期，所得資訊較為抽象雜亂，需求階段的模型標準尚未完全，但是需求階段的資訊完整性與之後的開發階段息息相關，影響深遠。

開發過程中除了程式碼外也會產出需求文件、需求、設計模型等輔助專案進程，完整的需求文件及設計模型可以減少大量人為疏失及資訊錯誤，甚至可以自動化產生一些程式碼，並在測試維護期能降低

大量的時間成本，並可將現有的開發經驗再利用到相似的其他專案上。

在嵌入式專案中，除了傳統的建模資訊外，為了符合硬體及現實時間需求另外增加了硬體及即時性限制，因為硬體在實作時出現問題修正要花更多心力，因此在需求設計階段時能早期發現問題便能減少實作測試階段所花的成本，但增加的資訊量也增加了嵌入式系統的開發困難，因此如何在不同觀點的模型資訊中建立關聯性，並兼顧各種限制性資訊及需求條件，讓使用者利用需求追蹤性[6][7]找到相關的需求，方便整理相關的資訊成了我們關注的重點及實作的目標。

1.3 研究目的

我們為了同步配合軟體模型介面，可在建立關聯性鏈結及其追蹤性的同時可檢視專案模型進度，本論文在 Eclipse 環境建置一個外掛工具，將軟體專案各階段模型中的重要資訊取出，依不同軟體時期及類型資訊進行分類整理，利用 XML 資料庫保存專案中的 XML 資訊檔，配合 XQuery[8]語法將所需要的重要資訊從各模型資訊檔中抓出，放入 XUM 模型中進行整理，比對不同模型資訊中相同或相似物件的名稱，將不同圖型間의相同物件及限制資訊進行鏈結，並因嵌入式軟體有硬體及即時性上的需求，並將需求階段的限制需求資訊抓出來，與設計階段中 SysML、MARTE 模型中的限制型資訊等抓出並比對，以鏈結關係建立需求的追蹤性，使得嵌入式系統專案的資訊統合性更加完整，並在面版上表現出各物件完整資訊，可讓使用者在面版上直接修改限制資訊，減少查找模型圖型中不同需求限制資訊所需的時間。

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，此章說明與本論文相關的研究。

第三章 論文研究方法及理論。

第四章 系統實作及案例研究。

第五章 結論及未來展望。

第二章 背景知識與相關研究

本章節將介紹此論文中使用到的軟體工程及嵌入式系統所需的背景與知識。

2.1 XML (extensible Markup Language)

XML[9]是由 W3C 簡化了 SGML (Standard Generalized Markup Language)[10]而制訂出的標準，具有擴展性、可延伸性的特性，為一種資訊交換的格式；因 XML 呈現結構化的特性，故常用來描述文件的結構與內容，可適用於不同的程式語法及使用平台。

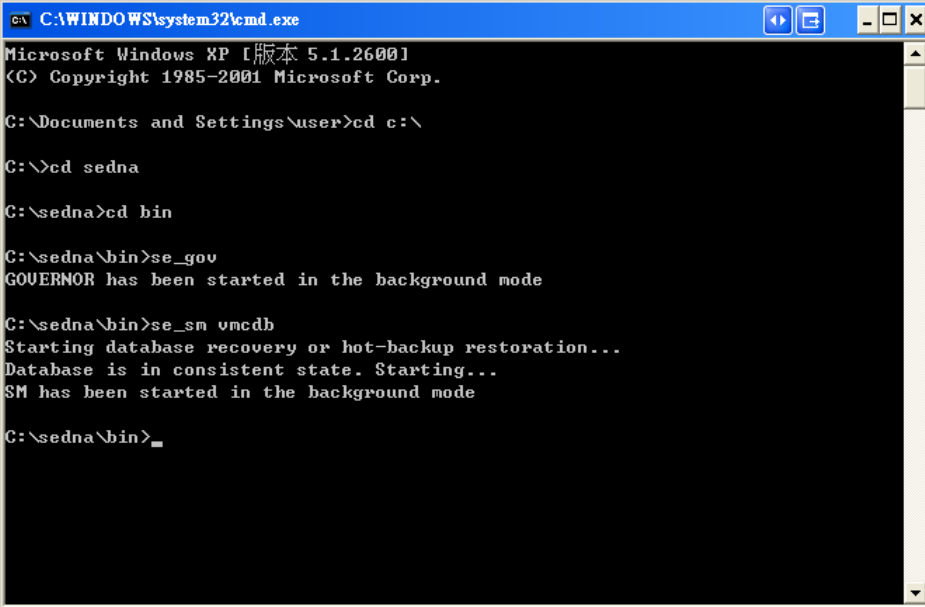
因為XML 本身就是一種Meta Language，可利用XML 定義各種 Model 的文件語言，在進行各種Model 轉換時可以利用XML 延伸出的XMI 來幫助檔的交換及存取。XMI 是XML Metadata Interchange 的縮寫，為UML交換資訊的格式，XMI 在1999 年3 月23 日正式被 W3C 所建議為正式標準。

XML 包含了 DTD 與 XSL:DTD(Document Type Definition)，是定義 XML 文件的規格(Schema)，如標籤(Tag)、從屬關係及限制條件等，XML 文件的驗證則是以 DTD 作為驗證條件(Criteria)。

我們所使用的 XML 資料庫 Sedna[11]一種 XML 資料庫，保存 XML 檔進入資料庫中，可對 XML 檔直接使用 XQuery 語法進行 XML

的檢索搜查、段落增減及刪存。

Sedna 的開啟如圖 一，需使用 cmd 進入 Sedna 資料夾，啟用開啟資料庫的指令，確認開啟後程式才能對指定資料庫進行連線，目前已有 SDBAdmin 等 Sedna 專用程式可直接觀看、檢視資料庫中的檔案並對資料庫下 Xquery 指令。



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [版本 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\user>cd c:\

C:\>cd sedna

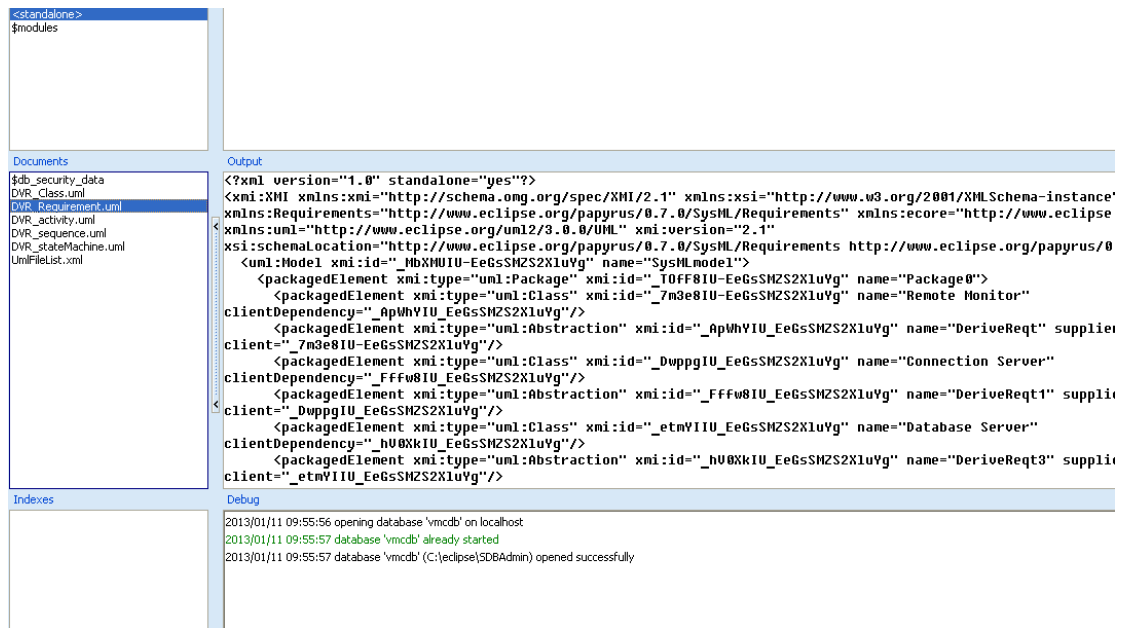
C:\sedna>cd bin

C:\sedna\bin>se_gov
GOVERNOR has been started in the background mode

C:\sedna\bin>se_sm vmcdb
Starting database recovery or hot-backup restoration...
Database is in consistent state. Starting...
SM has been started in the background mode

C:\sedna\bin>_
```

圖 一 經由 cmd 開啟 Sedna 資料庫，並啟動其中的 vmc 資料庫



圖二 SDBAdmin 軟體表現 Sedna 資料庫

2.2 統一塑模語言(UML)

統一塑模語言(Unified Modeling Language)[3]是一種核心建模語言，用圖型方式表達不同觀點的模型架構及其資訊，結合了物件導向觀點，多用於表達軟體系統的一種設計方式，由物件管理協會(Object Management Group, OMG)整合發表標準。

UML 集合了 Booch, OMT 和物件導向[12]的觀念，可自由擴展，整合為一通用的且易懂的塑模語言，經由不同的觀點表達出專案中的不同觀點。

2.3 SysML

SysML(Systems Modeling Language)[4][13][14]是 OMG 發表的一

種系統塑模語言，是為系統工程應用開發的標準塑模語言，是一般性圖型塑模語言用來分析、設計和驗證包括軟、硬體、資訊、人員、設備的系統。

如圖 三，SysML 延伸於 UML2.0，虛線方塊是與 UML2.0 上新增的模型，粗框的方塊則為從 UML2.0 修改的模型，能解決構造建模範圍廣泛的系統工程問題。

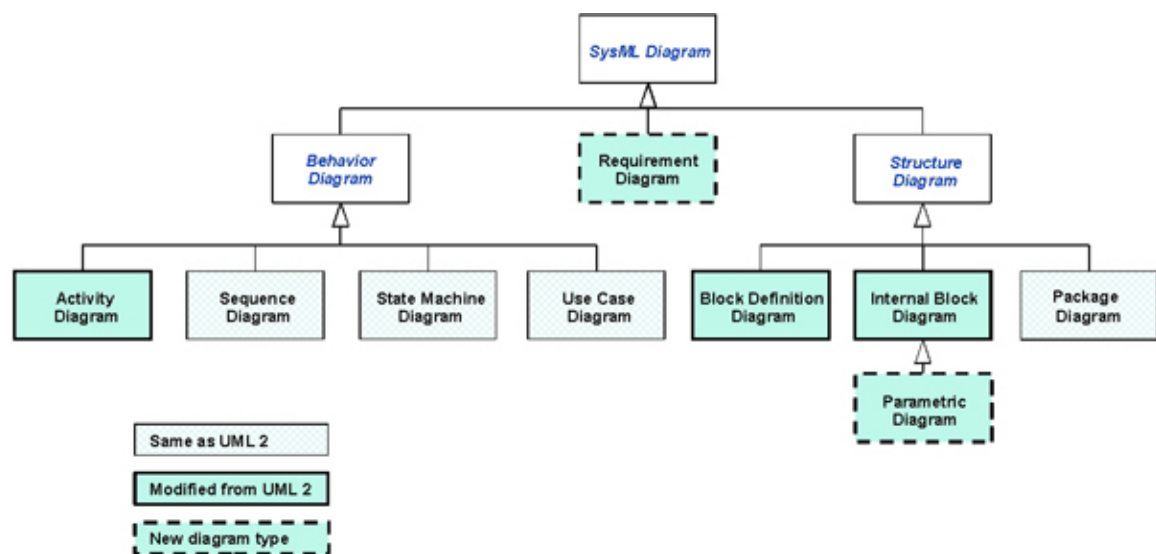


圖 三 SysML Diagram Types

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，新增圖類為參數圖和需求圖，目前已經有許多研究開始運用 SysML 進行系統塑模。如 Bock0 利用 SysML 以及 UML 2.0 來塑模系統需求，Johnson et.al.[12]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[15]。

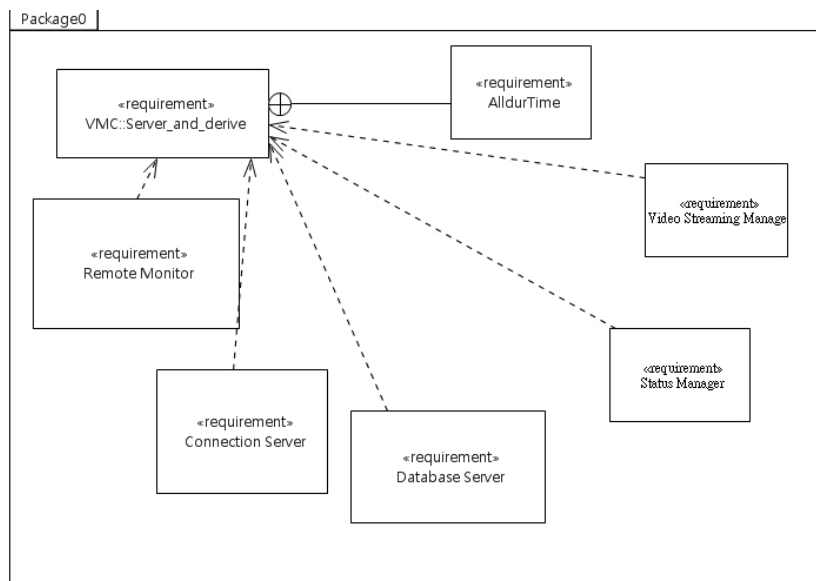


圖 四 DVR 的需求圖

圖 四為一監視系統(DVR)的需求圖，需求圖屬於 SysML Profile，定義了遠端監視器及所連接的伺服器及資料庫系統，另外還有整體執行時間的限制。此圖是使用 Eclipse 的 Papyrus 所繪製，需求圖的需求

設定等須寫在如

圖 五的表格框中，圖 五為 Video Streaming Manager 的物件屬性表，可設定設備 id、name 及 text，通常需求條件是寫在 text 欄中。

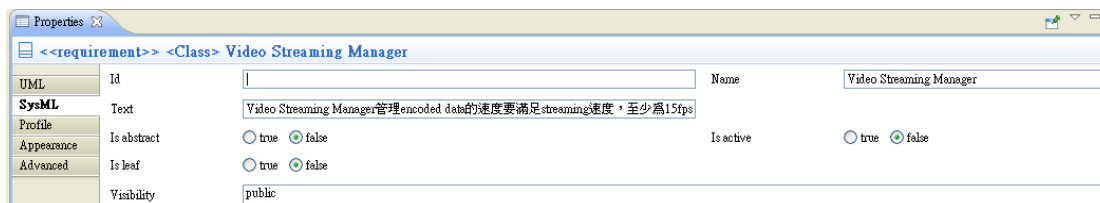


圖 五 papyrus 的 SysML 需求圖屬性欄

在 Papyrus 中 SysML 的資訊也是以 XML 格式表現，如圖 六，

此 uml 檔中包含 SysML 的 namespace 及物件名、Id、type 等重要資訊。

```
<?xml version="1.0" encoding="UTF-8"?>
<xmi:XMI xmi:version="2.1" xmlns:xmi="http://schema.omg.org/spec/XMI/2.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SysML="http://www.omg.org/spec/sysml/2.1" xmi:id="SysMLmodel">
  <uml:Model xmi:id="_MbXMUIU-EeGsSMZS2XluYg" name="SysMLmodel">
    <packagedElement xmi:type="uml:Package" xmi:id="_TOff8IU-EeGsSMZS2XluYg" name="Package0">
      <packagedElement xmi:type="uml:Class" xmi:id="_7m3e8IU-EeGsSMZS2XluYg" name="Remote Monitor"
        clientDependency="_ApWhYIU_EeGsSMZS2XluYg"/>
      <packagedElement xmi:type="uml:Abstraction" xmi:id="_ApWhYIU_EeGsSMZS2XluYg" name="DeriveReq1"
        supplier="_gR4q0IU-EeGsSMZS2XluYg" client="_7m3e8IU-EeGsSMZS2XluYg"/>
      <packagedElement xmi:type="uml:Class" xmi:id="_DwppgIU_EeGsSMZS2XluYg" name="Connection Server"
        clientDependency="_Fffw8IU_EeGsSMZS2XluYg"/>
      <packagedElement xmi:type="uml:Abstraction" xmi:id="_Fffw8IU_EeGsSMZS2XluYg" name="DeriveReq1"
        supplier="_gR4q0IU-EeGsSMZS2XluYg" client="_DwppgIU_EeGsSMZS2XluYg"/>
      <packagedElement xmi:type="uml:Class" xmi:id="_etmYIIU_EeGsSMZS2XluYg" name="Database Server"
        clientDependency="_hV0XkiU_EeGsSMZS2XluYg"/>
      <packagedElement xmi:type="uml:Abstraction" xmi:id="_hV0XkiU_EeGsSMZS2XluYg" name="DeriveReq3"
        supplier="_gR4q0IU-EeGsSMZS2XluYg" client="_etmYIIU_EeGsSMZS2XluYg"/>
      <packagedElement xmi:type="uml:Class" xmi:id="_ee1G4IVA-EeGsSMZS2XluYg" name="AllExecTime">
      <nestedClassifier xmi:type="uml:Class" xmi:id="_gR4q0IU-EeGsSMZS2XluYg" name="VMC::Server_and_derive">
      <nestedClassifier xmi:type="uml:Class" xmi:id="_PDZBsIVCEeGsSMZS2XluYg" name="AllExceTime"/>
    </packagedElement>
  </uml:Model>
</xmi:XMI>
```

圖六 Requirement Diagram 的 UML 檔

2.4 MARTE

MARTE(Modeling and Analysis of Real-Time and Embedded systems) [5][16]是 UML2.0 的擴展文件，多應用在即時系統和嵌入式系統上的模型驅動開發上，MARTE 由 OMG 定義，為即時系統和嵌入式系統模型的基本概念，應用在硬體和軟體模型的塑模，支援 UML2 模型的量化分析，包含了可排程性和效能分析[17]。

如圖七，MARTE 對於即時系統和嵌入式系統定義了基本概念，支援純應用程式塑模規範(例如功能性和時間的塑模)、建模及分析三個部分，在規範、設計、驗證、程式碼產出等的開發工具之間提供了互通性。

近年有不少學者對 MARTE 進行塑模及分析研究，MARTE 所提供的 Non-Functional Properties 進行可靠度的分析[18]，Espinoza 等人則點出同時使用 SysML 與 MARTE 進行塑模時會產生衝突與挑戰的地方[19]。

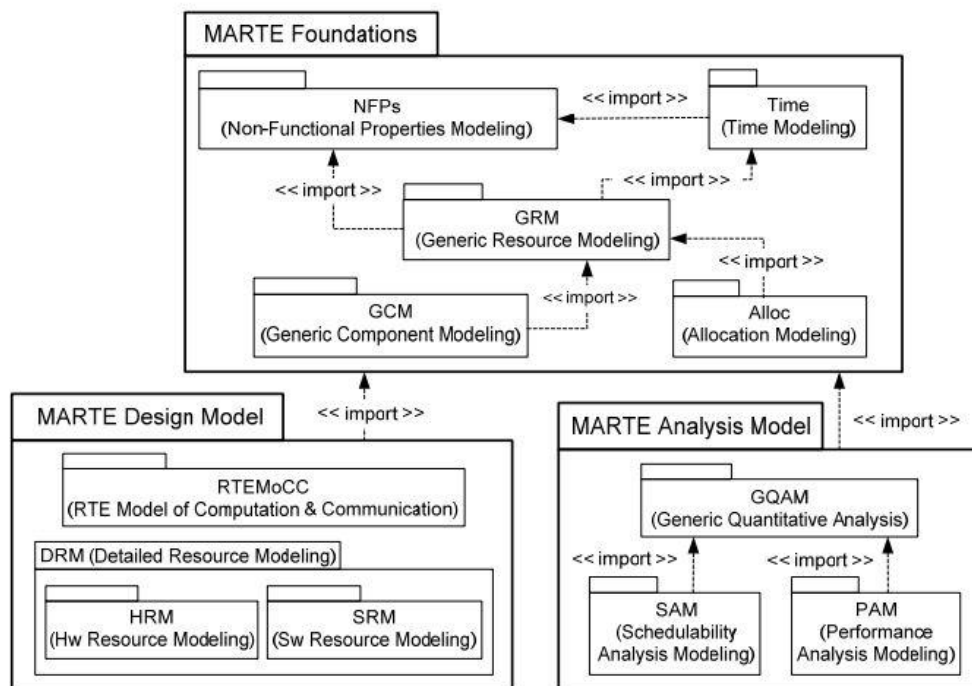


圖 七 MARTE 的架構

舉例 MARTE 模型如圖 八類似，MARTE 可延伸 UML 圖型，於 constraint 區塊標注上各種硬體或即時限制等資訊，可直接讓使用者表達出嵌入式系統上的軟硬體限制，MARTE 的 constraint 資訊大都是使用 VSL 語法去描述資訊內容。

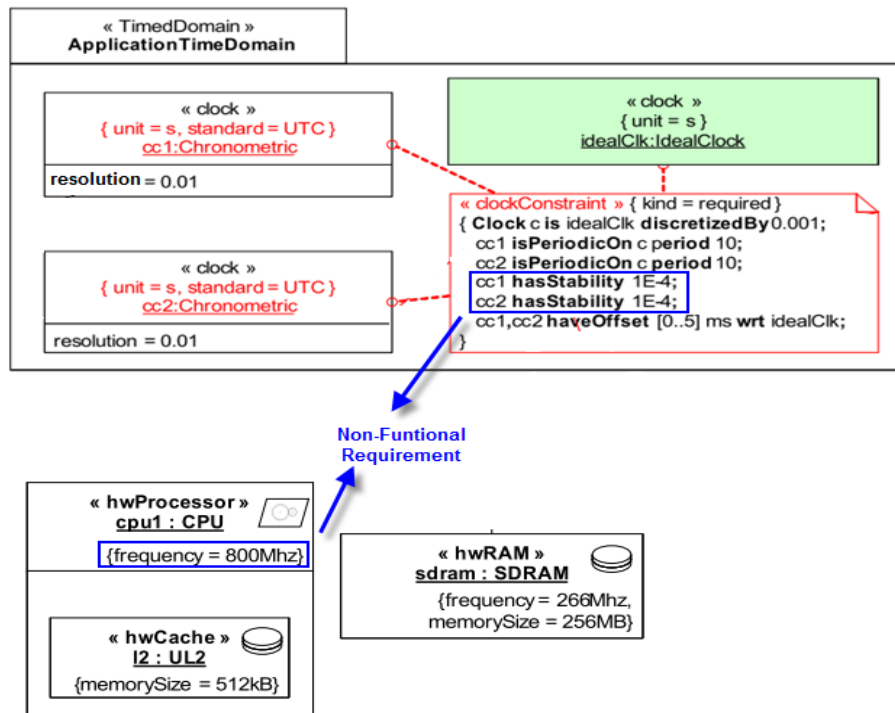
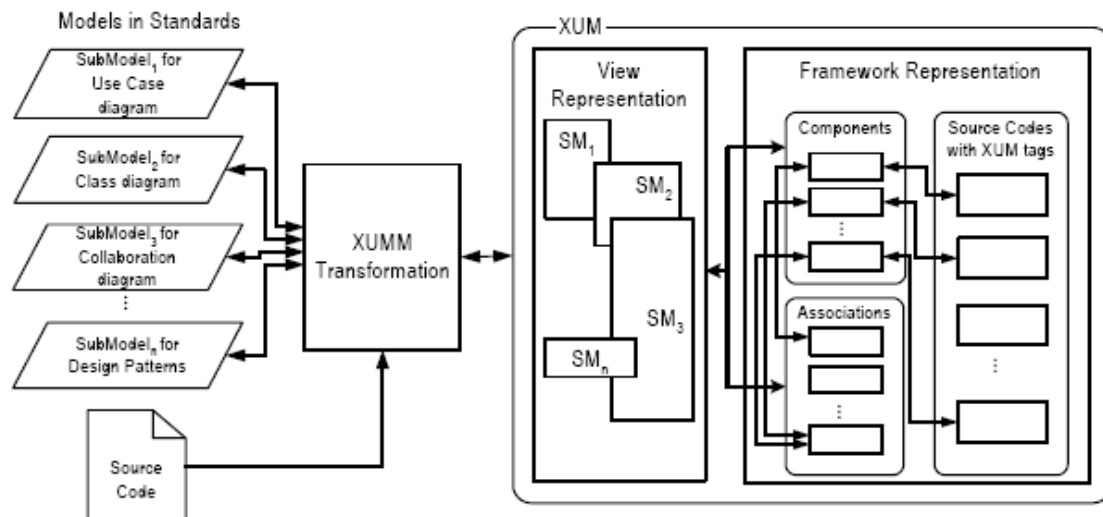


圖 八 MARTE Model

2.5 XUM

XUM(XML-Base Unified Model)[20][21]為一基於 XML 格式的資訊整合模型，經由 XUMM(XML-Based Unified Meta-Model)來定義 XUM 中的資料型態，並轉化整理 UML 中的重要資訊為 XUM 格式，如圖 九。

XUM可以有效地將不同的標準整合在一起，將相關的方法表示成元件(Components)、關聯(Associations)以及整合鏈結(Integration Links)，已可證明軟體設計與軟體實作階段是能夠串接而達到一致性 [22][23]。



圖九 子模型與 XUM 經 XUMM 轉換的關係

XUM的結構中包含了三種結構，如圖十，分成不同專案階段，不同圖型的圖型資訊，相關鏈結關係，此圖分成四種專案階段，需求階段、設計階段、實作階段、測試階段，其下包含不同的圖型，圖型中含有不同資訊，經由抽象鏈結(Abstraction Link)連結需求階段及設計階段等不同階段的相關物件及關係，由整合鏈結(Integration Links)連結同階段間的相關物件及關係。

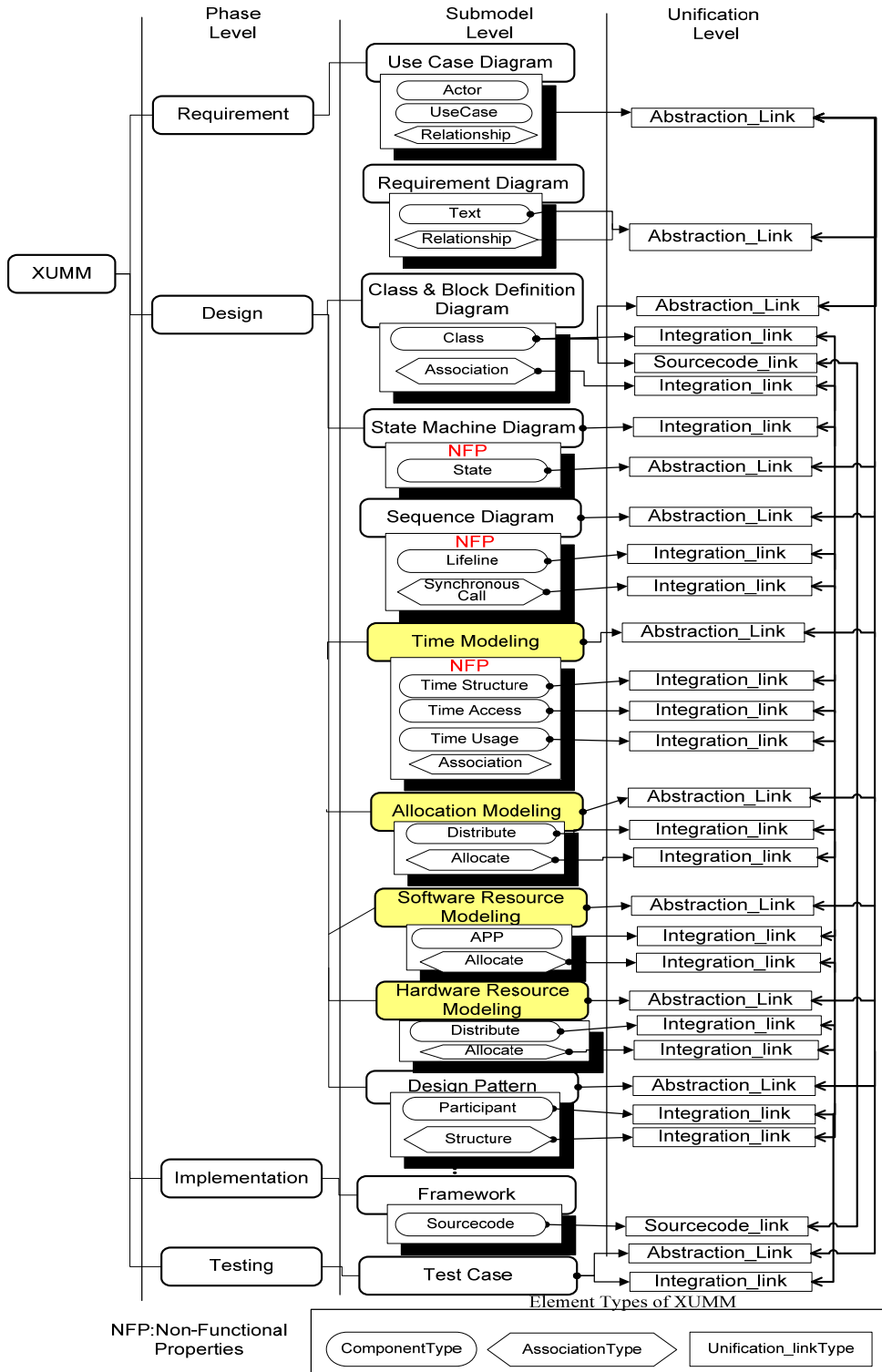


圖 十 XUM 整合模型的結構

2.6 VSL(Value Specification Language)

當在軟體模型需要輸入限制及需求資訊等，最大的問題是如何規格化資訊，在 SysML 及 MARTE 中主要使用 VSL(Value Specification Language)語法去描述限制條件及其定義，如表 一，VSL 除了可以表現單純的數值外，也可表現描述式及判斷選擇等，具有多變化及多應用度的優點。

表 一 VSL 定義及範例列表

數值類型	範例
Real Number	1.5E-10
Date/Time	#12/05/13 12:00:00#
Collection	{1,3,10,55} {{2,5},{1,8}}
Tuple and choice	(value=10, unit=s)
Interval	[1..255] [\$A1...\$A10]
Variable declaration	io\$var var
Arithmetic Operation	+(10,var1) 10+var
Conditional Expression	((var<10)?(10^2):10)

第三章 研究方法

嵌入式軟體工程需要的除了基礎的模型物件，還包含許多硬體需求及實際的限制，在不同的模型中可表現出同物件的不同觀點，但需求及限制不會因為觀點的改變而變化，反而可能因為建模者的疏忽及錯誤，遺失或弄錯限制條件，造成專案完結後的成果出現問題。

延續 XUM 研究，我們可以知道同專案中的不同模型間的一致性十分重要，而建立了鏈結關係使其存在追蹤性時，當不一致的模型出現，可以查尋是否中間是出現問題，多為人為疏失，結果會造成運作上的錯誤，在嵌入式系統的實作上可能造成嚴重的問題。

經由程式抓出各模型中的重要資訊，我們可在建立鏈結連接不同的模型資訊，在 XUM 中建立不同模型間的平行與垂直關係，但是在嵌入式專案中，基於 SysML 及 MARTE 於硬體及即時性的延伸應用，重要的資訊不只有單純的物件間的關聯，還有物件間的需求及各種資源、即時限制(Constraint)資訊，因為相關的限制會同時存在於不同的模型間，因此我們追蹤不同階層中的需求及限制資訊，找出可能存在不一致資訊或資訊缺失的物件，將此資訊抓出並以物件追蹤樹的方式表現出來，可快速讓使用者發現錯誤。如何抓取限制資訊並將其統整進入 XUM 模型中為我們研究的主要議題。

3.1 延伸於 SysML 及 MARTE 的 XUM 整合模型

在軟體專案中，不同時期的開發階段會需要使用不同類型的模型呈現，因為同為一專案中的模型，不同模型間很可能會出現相關聯的物件或是相同的物件，差別在於不同圖型中有不同的表達法及應用方式。

在嵌入式系統專案中，需求階段可應用的模型為案例圖(Use Case Diagram)及 SysML 的需求模型(Requirement Diagram)。

設計階段使用的為區塊定義圖(Block Definition Diagram)、狀態機圖(State Machine Diagram)、循序圖(Sequence diagram)及 MARTE 新增配置圖(Allocation Diagram)、軟體資源建模(Software Resource Modeling, SRM)、硬體資源建模(Hardware Resource Modeling, HRM)以協助嵌入式系統的設計，並有 Time modeling 以表達在多核心環境下的平行。

我們使用 XUM 模型去保存不同的模型資訊和關係鏈結資訊，關係鏈結(Unification Links)包含抽象鏈結和整合鏈結，抽象鏈結(Abstraction Link)是串連需求到計設的整合鏈結(Integration Links)，整合鏈結為不同觀點中的相關物件，另有程式鏈結則為設計階段的模型圖與實作階段的程式鏈結(Source Code Link)。

針對於嵌入式 SysML 及 MARTE Profile 所新增的圖型資訊，我們於關係鏈結中新增一種限制鏈結(Constraint Link)，此鏈結連結需求

階段中的需求圖中使用者設定的需求條件的物件及與設計階段與有相關限制條件的物件，使其可以快速向下追蹤到其相關的限制資訊。

並於 XUM 模型中，新增了對應於 SysML 及 MART 圖型及其限制鏈結的結構，格式如圖 十一、圖 十二。

```
<xsd:element name="XUM">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="1" minOccurs="1" name="requirement" type="requirementPhase">
      </xsd:element>
      <xsd:element maxOccurs="1" minOccurs="1" name="design" type="designPhase">
      </xsd:element>
      <xsd:element maxOccurs="1" minOccurs="1" name="implementation" type="implementPhase"/>
      <xsd:element maxOccurs="1" minOccurs="1" name="unification_link" type="unificationlinkType">
      </xsd:element>
      <xsd:element maxOccurs="1" minOccurs="1" name="data_dictionary" type="wordType">
      </xsd:element>
      <xsd:element name="database" type="dbType"/>
      <xsd:element name="code_path" type="xsd:string"/>
      <xsd:element name="constraint_link" type="constraintlinkType"/>
      <xsd:element name="constraint_integrateLink" type="constraintintegrateType"/>
    </xsd:sequence>
    <xsd:attribute name="project_name" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>
```

圖 十一 XUM 資訊及鏈結架構

```
<xsd:complexType name="constraintlinkType">
  <xsd:attribute name="date" type="xsd:string"/>
  <xsd:attribute name="to_obj" type="xsd:string"/>
  <xsd:attribute name="from_obj" type="xsd:string"/>
  <xsd:attribute name="value" type="xsd:string"/>
  <xsd:attribute name="classType" type="xsd:string"/>
  <xsd:attribute name="obj_Name" type="xsd:string"/>
  <xsd:attribute name="from_file" type="xsd:string"/>
  <xsd:attribute ref="id"/>
</xsd:complexType>
```

圖 十二 限制屬性定義

圖 十一為 XUM 中的所需的資訊及鏈結結構，unification_link 為整合鏈結，constraint_link 為一限制結構。

圖 十二中 constraintlinkType 定義了 constraint_link 的細部定義，

內含限制結構的唯一 id，時間、限制條件出現的檔案名、與限制條件有鏈結的物件 id、限制條件的數值。

3.2 SysML 的需求及 MARTE 的限制條件抓取

軟體專案的流程有五個階層：需求階段、設計階段、實作階段、測試階段、維護階段，在不同階段會產出所需的模型及文件，而軟體模型的建立也依需求有先後關係，一般在需求階段時會制作需求規格書(SRS)及專案計畫書(PEP)，內含需求模型及 USE CASE 等模型，而在嵌入式專案中，格外注重硬體條件及成果效能，而各種文字需求規格，會在需求模型(Requirement Diagram)以參數數值表達想要的各種限制條件，如即時限制、資源配置、硬體設備等限制。

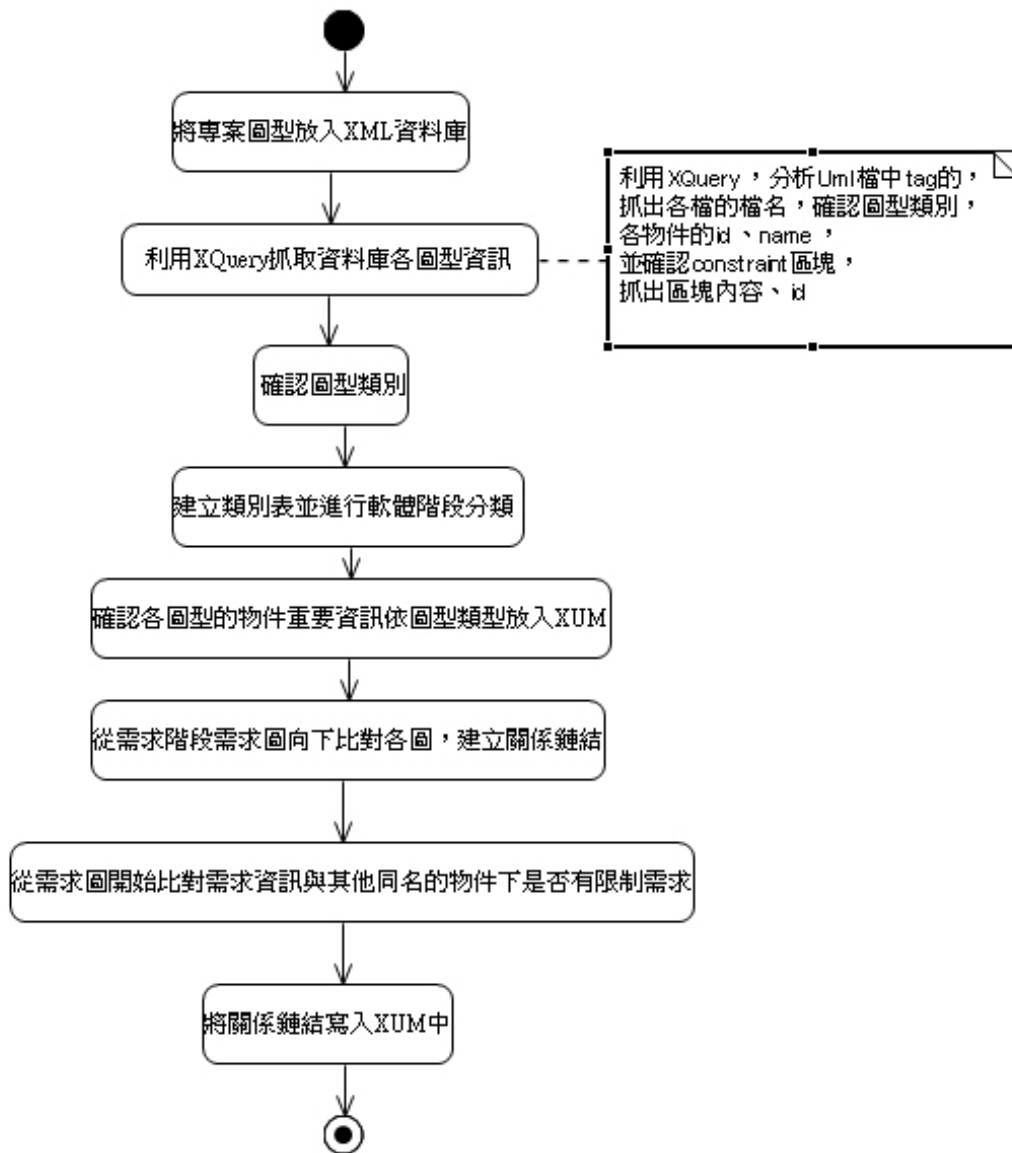


圖 十三 從專案中抓取資訊放入 XUM 中的流程

在專案中需求圖是在需求時期就會定義好專案的設備或需求、限制等，抓取資訊流程如圖 十三，並經由判別 uml 檔中 Tag 中，例如從 packagedElement 中抓取 xmi:type、xmi:id 跟 name，其中 xmi 開頭的 tag 為 SysML 的 uml 檔中所定義的 tag，需加載 xmi namespace 才可進行 xquery 搜取資料，xmi:id 可依 Requirements:Requirement 中的 base_Class 找到相關聯性的需求設定，如圖 十四。

```

<Requirements:Requirement xmi:id="_gS1GAIU-EeGsSMZS2XluYg" base_Class="_gR4q0IU-EeGsSMZS2XluYg"/>
<Requirements:Requirement xmi:id="_7m_awiU-EeGsSMZS2XluYg" id="DVR1.1" base_Class="_7m3e8IU-EeGsSMZS2XluYg"/>
<Requirements:DeriveReq xmi:id="_ApfEQIU_EeGsSMZS2XluYg" base_Abstraction="_ApWhYIU_EeGsSMZS2XluYg"/>

<Requirements:Requirement xmi:id="_DwvJEIU_EeGsSMZS2XluYg"
    text="OS=&quot;Linux/Ubuntu&quot;;memory=&quot;2GB&quot;;HD=&quot;120GB&quot;;
    text=&quot;Max_execTime>100ms and Min_excTime&lt;500ms&quot;" id="DVR1.2" base_Class="_DwppgIU_EeGsSMZS2XluYg"/>

<Requirements:DeriveReq xmi:id="_Ffl3kiU_EeGsSMZS2XluYg" base_Abstraction="_Fffw8IU_EeGsSMZS2XluYg"/>
<Requirements:Requirement xmi:id="_ettF0IU_EeGsSMZS2XluYg"
    text="OS=&quot;Linux&quot;;Type=&quot;XML Database&quot;;HD=&quot;2TB&quot;"
    id="DVR1.3" base_Class="_etmYIIU_EeGsSMZS2XluYg"/>
<Requirements:DeriveReq xmi:id="_hV53IIU_EeGsSMZS2XluYg" base_Abstraction="_hV0XkiU_EeGsSMZS2XluYg"/>
<Requirements:Requirement xmi:id="_ee6mcIVAEeGsSMZS2XluYg"
    text="" id="DVR1.4" base_Class="_ee1G4IVAEeGsSMZS2XluYg"/>
<Requirements:Requirement xmi:id="_PDgWcIVCEeGsSMZS2XluYg"
    text="text=&quot;durTime&lt;5s and durTime>500ms&quot;" id="DVR1.4" base_Class="_PDZBsIVCEeGsSMZS2XluYg"/>
</xmi:XMI>

```

圖 十四 需求圖中的物件內的各種設定值

我們將模型中的每個部件看作一個物件，以物件為基礎單位進行關係鏈結，在 MARTE 中，物件與物件間常設有資源(Resource)及即時性(Real-time)的軟硬體限制，如圖 十五，(a)及(b)為同一專案中的不同的圖，圈出來的為限制條件，限制條件可以存在於物件或物件間，限制存在於不同觀點的模型間，在嵌入式系統中，限制條件影響到實際狀況下的運作，必須要將這些限制資訊從模型資訊中抓取出來。

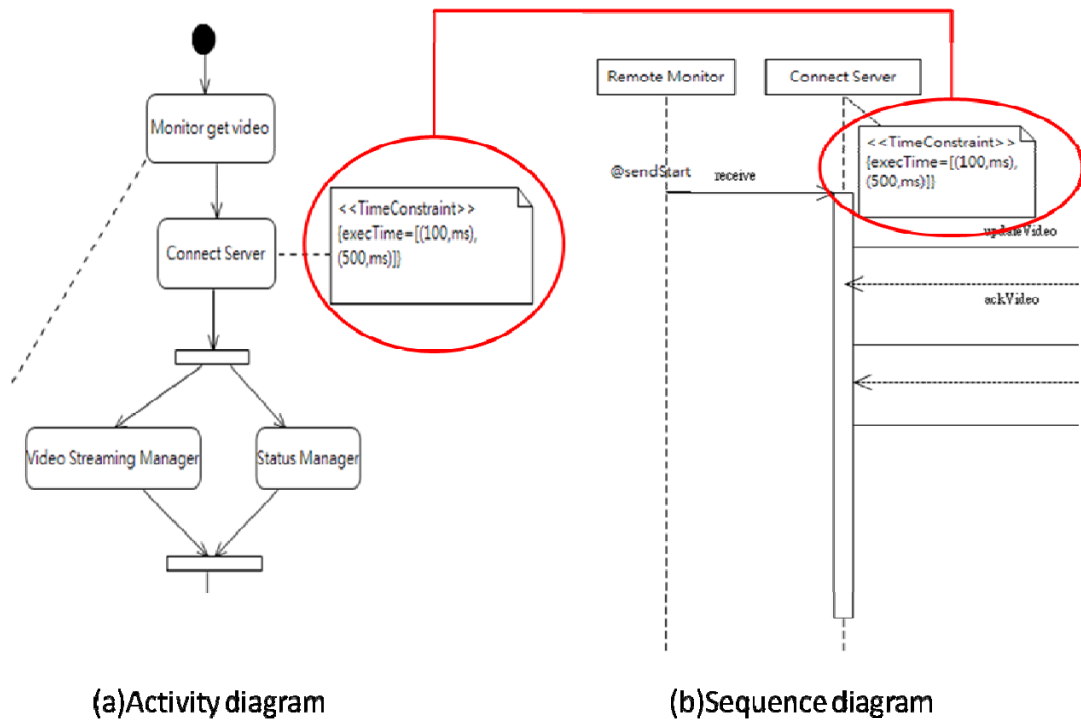


圖 十五 出現在不同圖中的限制條件

經由 Pyparus 建模工具所建之專案中，UML、SysML 及 MARTE 的模型圖的資訊皆保存於同模型名的 uml 檔中，經由讀取此 XML 格式檔，我們可依標籤名(tagName)確認物件類型，拿取出如物件類別、物件名、id 等重要資訊，並找到重要需求及限制條件的區塊，如圖 十六，擷取出所需的重要資訊及區塊與物件間的連接關係。

```

type                                     id
<node xmi:type="uml:OpaqueAction" xmi:id="_XbczgGmQEeGEzvhrHCXSBA"
  name="Connection Server" outgoing="_lfQFcGmQEeGEzvhrHCXSBA"
  incoming="_gdgp4GmQEeGEzvhrHCXSBA"/>
name

```

圖 十六 UML 檔中需拿取的重要資訊

在 UML 中保存大部分資訊的為 uml 檔，uml 檔為 XML 格式的檔

案。我們可以抓取物件，並同時抓取相應的需求資訊，如圖 十七，將需求資訊及物件間建立起關係鏈結。

```
<Requirements:Requirement xmi:id="_DwvJEIU_EeGsSMZS2XluYg" text="OS=&quot;Linux/Ubuntu&quot;;memory=&quot;2GB&quot;;text=&quot;Max_execTime>100ms and Min_excTime&lt;500ms&quot;;" id="DVR1.2" base_Class="_DwppgIU_EeGsSMZS2XluYg"/>
```

圖 十七 需求圖的 UML 檔中的資訊

在 MARTE 中，以圖 十八為例，先找到與限制條件區塊相連接的物件，將限制條件內容及連相連關係抓取出來，並將拿取到的資訊依其類別及類型放入 XUM 模型中，建立整合鏈結及限制屬性。

```
<ownedComment xmi:id="_5Uy3EGmREeGEzvhRHCKXSBA" annotatedElement="_XbczgGmQeGEzvhRHCKXSBA_2VLgwGmQeGEzvhRHCKXSBA">
  <body>&lt;&lt;TimeConstraint>>&#xD;
  (execTime=[(100,ms),(500,ms)])&#xD;
&lt;/body>
&lt;/ownedComment>
```

圖 十八 MARTE 類型圖中常見的限制資訊

經過程式將物件中需求、限制資訊進行比對判別後，圖 十九為 XUM 中建好的限制鏈結架構，含有鏈結唯一 id 及鏈結資訊判定狀態，及需求、限制資訊內容等重要資訊。

```
<constraint_link id="_7m_awiU-EeGsSMZS2XluYg%_wt-4IHSbEeGtVv5WjWtqbQ" obj_Name="Remote Monitor"
classType="N" value="" from_obj="DVR_Class.di" date="2013/1/10 上午 01:31:36"/>
<constraint_link id="_7m_awiU-EeGsSMZS2XluYg%_URu8wGggEeGcTd2jknPBrg" obj_Name="Remote Monitor"
classType="Y" value="" from_obj="DVR_sequence.di" date="2013/1/10 上午 01:31:36"/>
<constraint_link id="_7m_awiU-EeGsSMZS2XluYg%_QBNPggmQeGEzvhRHCKXSBA" obj_Name="Remote Monitor"
classType="Y" value="" from_obj="DVR_activity.di" date="2013/1/10 上午 01:31:36"/>
<constraint_link id="_DwvJEIU_EeGsSMZS2XluYg%_rGd2AHSbEeGtVv5WjWtqbQ" obj_Name="Connection Server" |
classType="N" value="OS=&quot;Linux/Ubuntu&quot;;memory=&quot;2GB&quot;;HD=&quot;120GB&quot;;durTime=[(100,ms),(500,ms)]"
from_obj="DVR_Class.di" date="2013/1/10 上午 01:31:36"/>
<constraint_link id="_DwvJEIU_EeGsSMZS2XluYg%_VLbk4GggEeGcTd2jknPBrg" obj_Name="Connection Server"
classType="N" value="OS=&quot;Linux/Ubuntu&quot;;memory=&quot;2GB&quot;;HD=&quot;120GB&quot;;durTime=[(100,ms),(500,ms)]"
from_obj="DVR_sequence.di" date="2013/1/10 上午 01:31:36"/>
<constraint_link id="_DwvJEIU_EeGsSMZS2XluYg%_XbczgGmQeGEzvhRHCKXSBA" obj_Name="Connection Server"
classType="Y" value="OS=&quot;Linux/Ubuntu&quot;;memory=&quot;2GB&quot;;HD=&quot;120GB&quot;;durTime=[(100,ms),(500,ms)]"
from_obj="DVR_activity.di" date="2013/1/10 上午 01:31:36"/>
```

圖 十九 XUM 中的限制鏈結結構

3.3 資訊追蹤性及整合

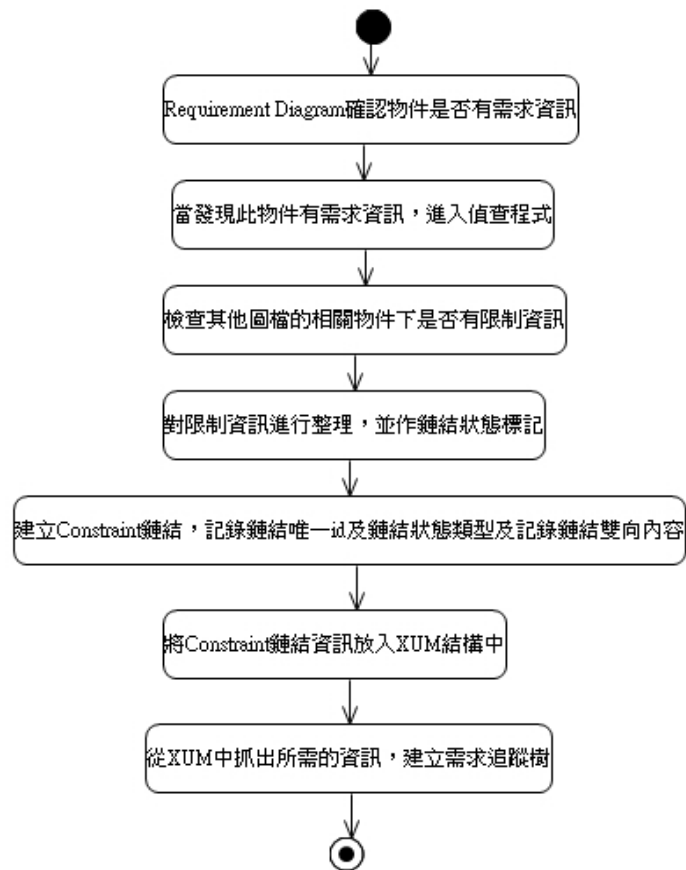


圖 二十 限制鏈結的建立及判別流程

圖 二十為需求及限制資訊抓取後並建立限制鏈結(Constraint Link)的流程，經由需求圖中的需求資訊及其他圖型中的物件下的限制資訊的判別，只要需求圖中有需求資訊的物件，找到與其有關的其他圖型的物件，依其下是否有限制資訊的存在，標記鏈結狀態，利於在追蹤樹中讓使用者觀察是否有限制資訊的缺失。

我們經由關係鏈結，可以連接起不同觀點中的相同物件，為表達在軟體專案模型中的追蹤性，我們將鏈結串接完成並實作成一物件追蹤樹，在資訊經過新增修改時，可讓使用者可快速追蹤到相關的圖

型，方便使用者比較需求期及設計期的限制資訊，易於正確的資訊，
如圖 二十一。

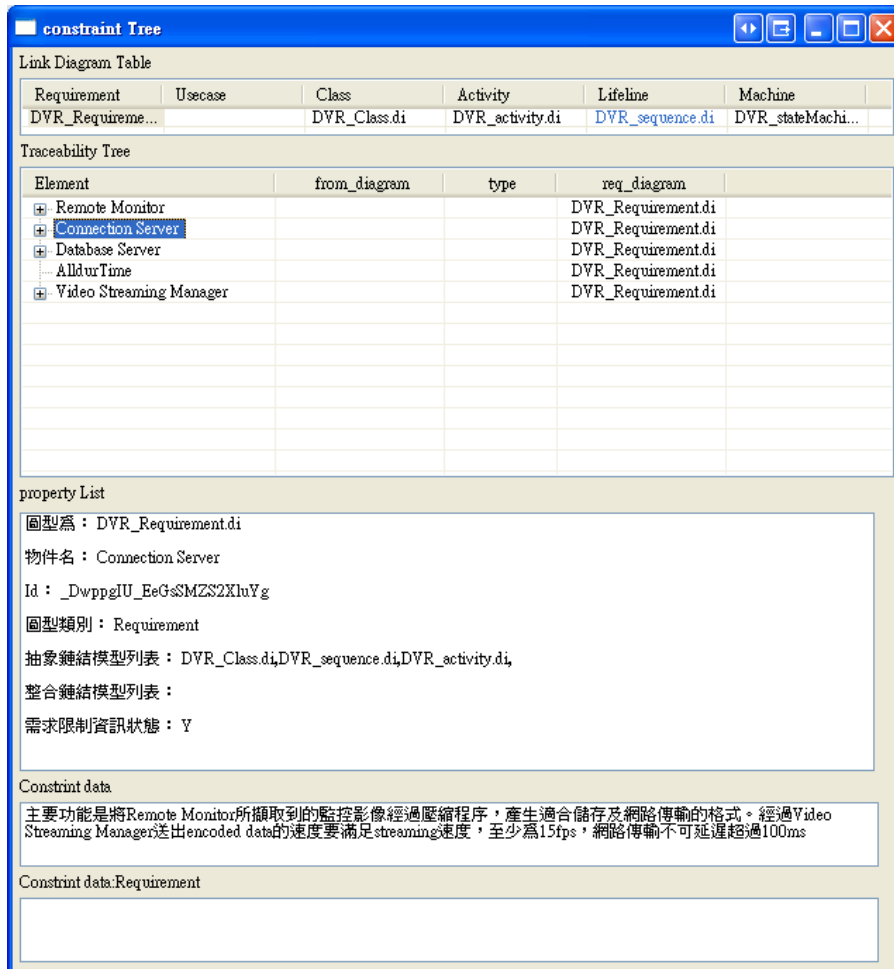


圖 二十一 追蹤樹及其完整資訊

在此物件追蹤樹中，以樹狀結構表現出需求模型與其他模型間的追蹤關係，根節結為需求模型的物件，其子樹為其他模型與其抽象鏈結的物件，子樹的平行子節點表現出整合鏈結。

對於限制資訊，經由整合鏈結除了能夠確保物件名相同外，在需求圖中可找到其下的限制資訊。

相關的限制資訊會出現在之後階段的圖型中，特別是在有相關性

的物件或同類型的物件下，若確認為是在不同階層、不同模型中的相關聯物件，基於同一資訊用在於相同的物件間，如圖 二十二及圖 二十三，因此找到了相同的物件即可抓取不同觀點中的需求及限制。

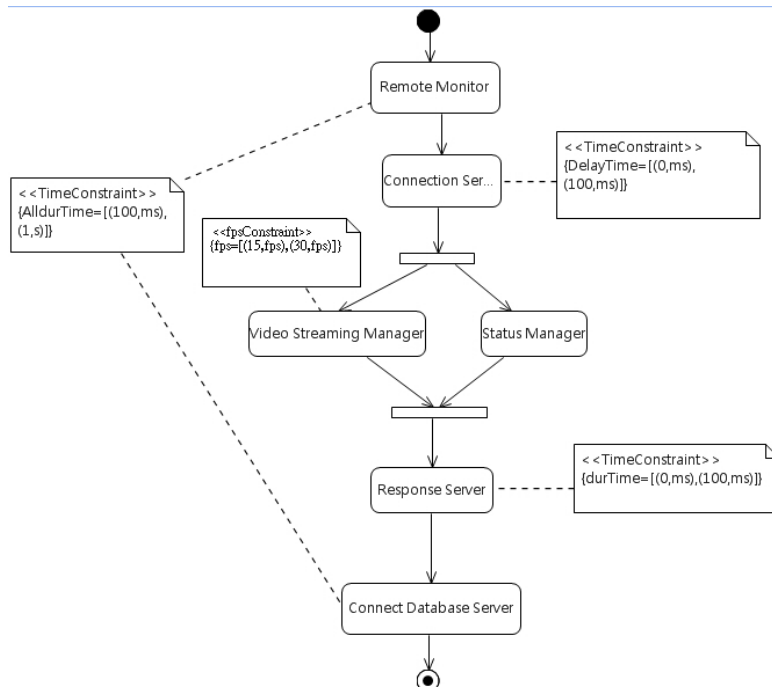


圖 二十二 活動圖的 MARTE 範例

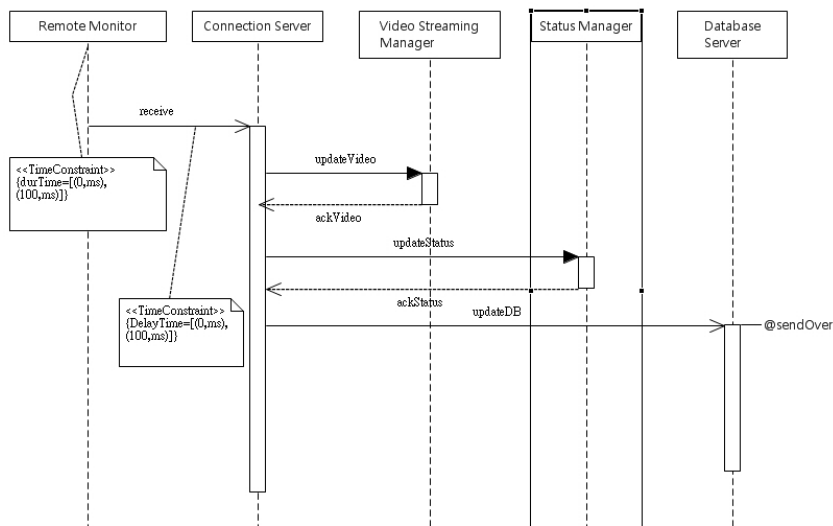


圖 二十三 循序圖的 MARTE 範例

如圖 二十四，圖(a)為需求圖，其中 Connect Server 的屬性 text

項中定義了限制條件，與圖(b)為同專案的循序圖，其中的 Connect Server 存在有整合鏈結，鏈結一個限制條件，圖(c) 為同專案的活動圖，其中的 Connect Server 與圖(b)中的 Connect Server 和圖(a)中的 Connect Server 也存在整合鏈結，並鏈結一限制條件，因為圖(a)中 Connect Server 與圖(b)中的 Connect Server 和圖(c)中的 Connect Server 皆以整合鏈結連接，其限制條件可能相同或相似。

若是使用者之後在需求圖中修改了需求的此項的限制條件，其他圖的限制條件也應與其一同改變，避免因為人為疏失。

並可在有相同的整合鏈結的項目與使用者進行確認，確定是否為同一限制條件，當其中一圖中的限制條件不小心輸入不同的數值時，進行資訊追溯偵查，告之使用者在輸入時出現人為疏失，將需求資訊錯誤的風險減到最低。

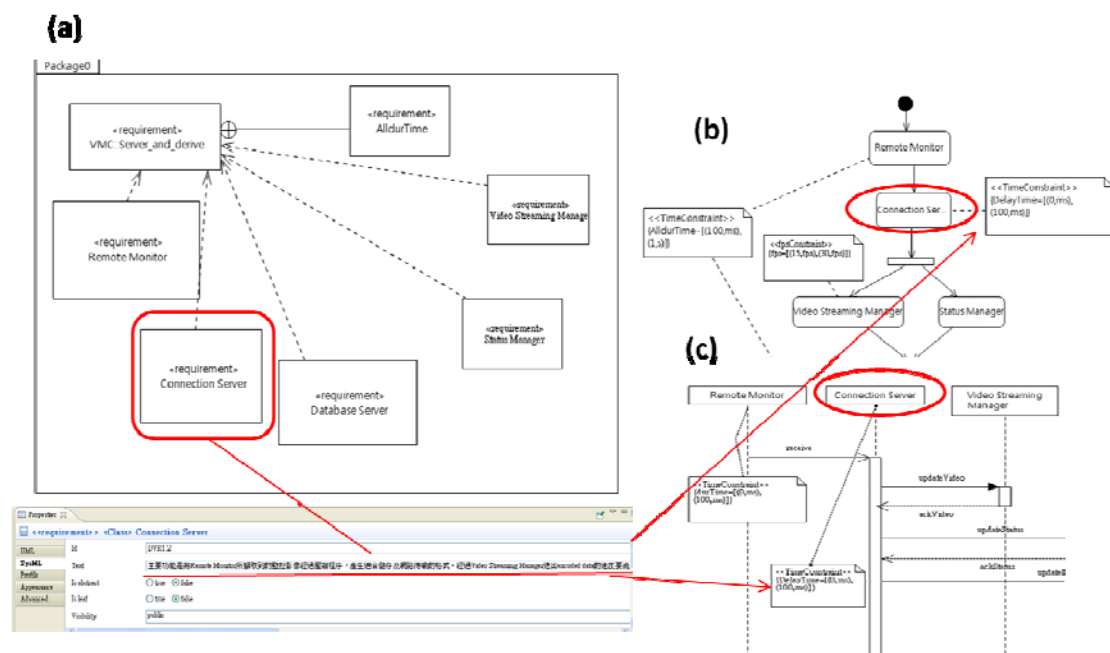


圖 二十四 關係鏈結與限制條件間的關聯性

經由 XUM 中整合的需求、限制條件，我們可以進行物件的需求、限制追蹤及整合，關連起不同觀點的相同物件及與其相連的資訊，可找到不同模型中的相同資訊位置，並可自動追蹤同一資訊在不同觀點的相關模型，減少不同觀點之間的資訊出現不同導致問題造成的結果錯誤。

並且我們使用 XML 資料庫作為保存資料的資料庫，XML 資料庫可將 XML 檔整個放入資料庫中建檔，透過 XQuery 語法，可以對資料庫進行搜尋、增減，XML 資料庫不只取得資訊便利且善於保存檔案，使用 XML 資料庫可減少多更改 XML 檔所需花費的時間，並且經由良好的資料庫索引定義，我們可以將專案完整保存並物件化，利於之後有相似專案可快速的再利用，減少專案開發時間。

第四章 案例研究與評估

在本章將以一嵌入式系統專案例子，在 Eclipse 環境上進行案例研究，此案例為一監視攝影系統(DVR)，此系統有主要有三個部分，遠端監視器(Remote Monitor)、連接的伺服器(Connect Server)、保存過往視訊的資料庫(Database Server)。

圖 二十五為我們的系統流程圖，我們先將專案的資料夾匯入系統後，在專案列表中新增一個新的 XUM 檔，並按下鈕使系統檢查此資料夾中的所有模型檔並建立 List 表，此時需向使用者詢問資料庫的相關資訊，並與 XML 資料庫進行連接，將資料夾中的模型的 uml 檔都匯入 XML 資料庫。

利用 Xquery 對 XML 資料庫，將不同模型圖中所需的資料如檔名、物件名、Id、需求、限制等資訊進行選擇性重點資訊抓取，並將資訊依類型類別等置入 XUM 檔相對位置中，並拿取 XUM 中的資訊進行比對並建立關係鏈結，同時比對需求及限制資訊，建立追蹤樹列表。

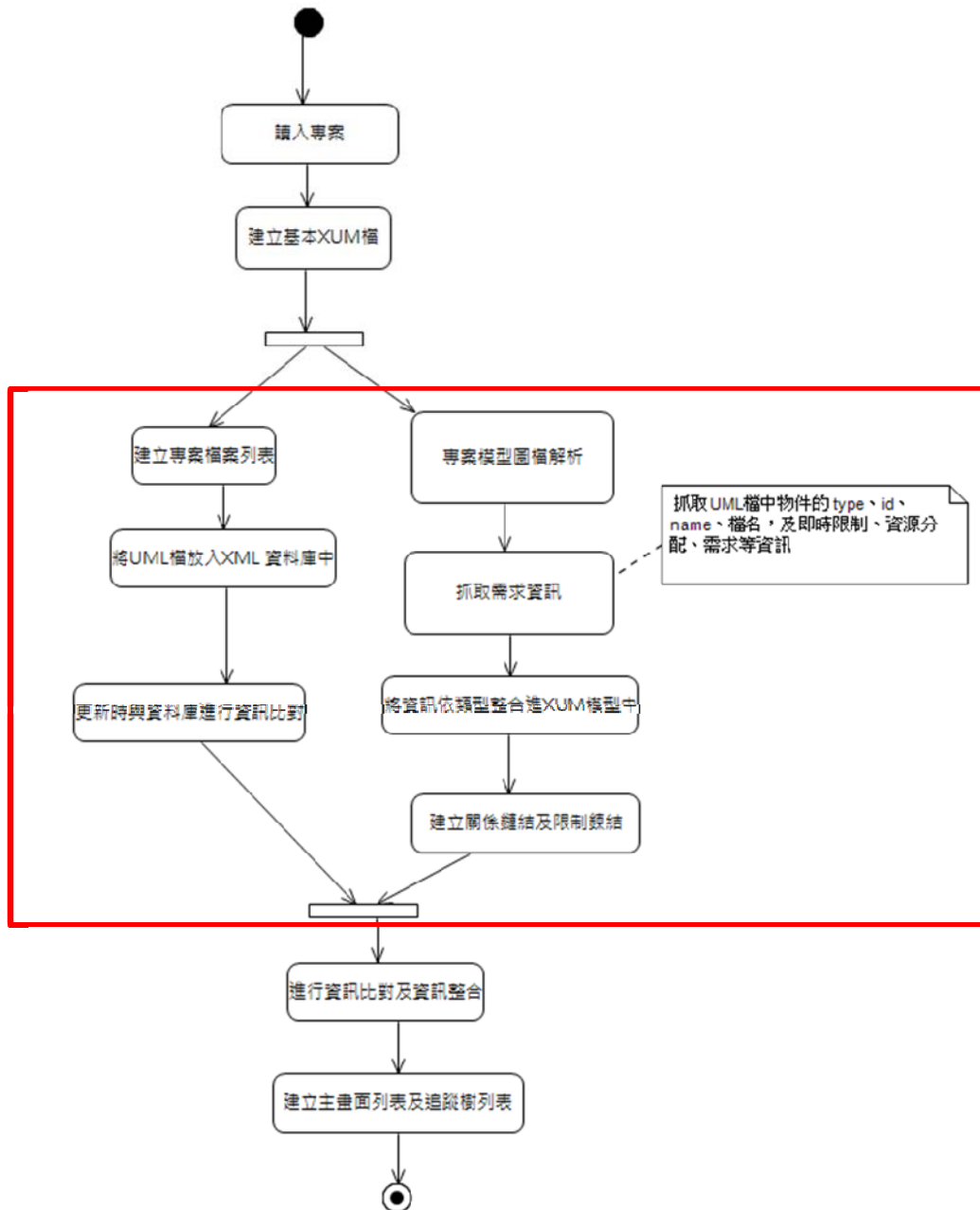


圖 二十五 系統流程圖

4.1 專案模型

此專案模型皆使用 Eclipse 的 papyrus 套件去繪製，從需求階段的需求圖(Requirement Diagram) 及案例圖(Use Case)到設計階段的類別

圖(Class Diagram) 、活動圖(Activity Diagram) 、循序圖(Sequence Diagram)等不同類型的模型圖，圖 二十六為 DVR 的需求圖，表現了五個主物件，遠端監視器(Remote Monitor)、伺服器(Server)、資料庫(Database Server)、視訊流管理(Video Streaming Manager)、狀態管理(Status Manager)，在其下的 text 細則描寫了嵌入式專案系統需求的硬體及設備及系統描寫等需求資訊，以此案例來看，有即時限制如在連接的伺服器上為了避免伺服器處理時間過長或是機器出問題而造成影片延遲等字串。

以圖 二十七為 DVR 需求圖中的 Connection Server 物件的需求條件為例，描寫了連接了伺服器後要經過什麼處理及限制影片幀率不可小於 15fps，和網路傳輸不可延遲超過 100ms 等資訊。

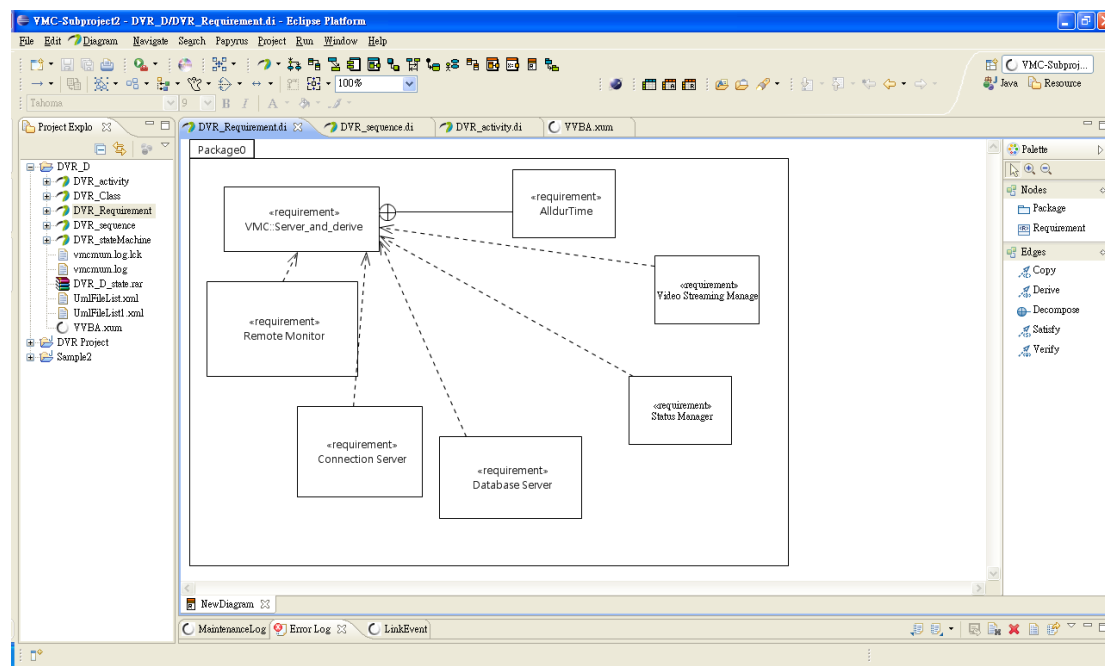


圖 二十六 DVR 的需求圖

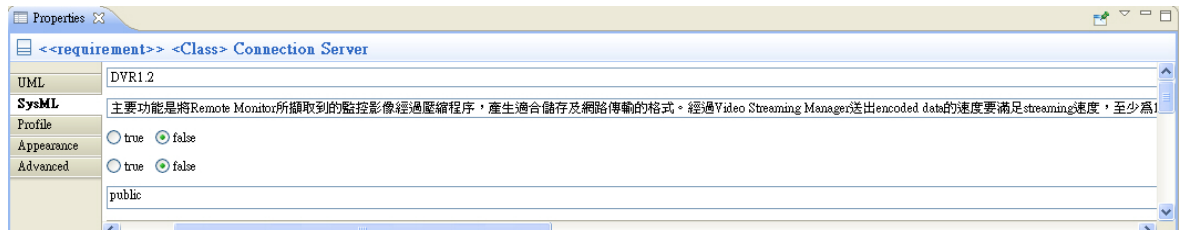


圖 二十七 DVR 的需求圖中的需求條件

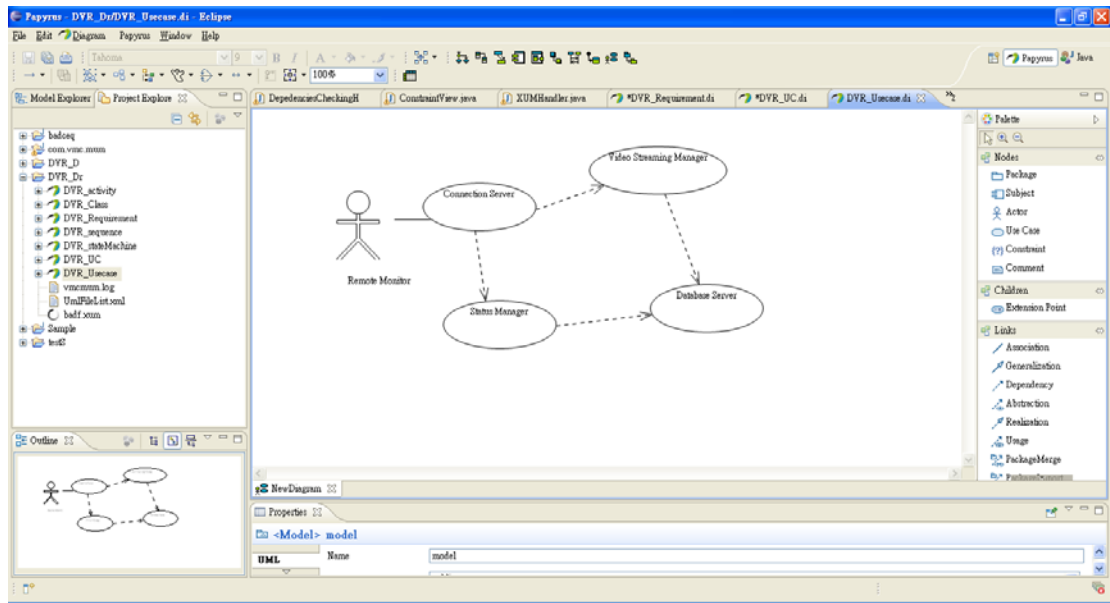


圖 二十八 DVR 的案例圖

圖 二十八，為 DVR 的案例圖，表現了遠端監視器與各子系統的功能性關係，如圖 二十九為 DVR 的類別圖，表現了各類別間的關係。

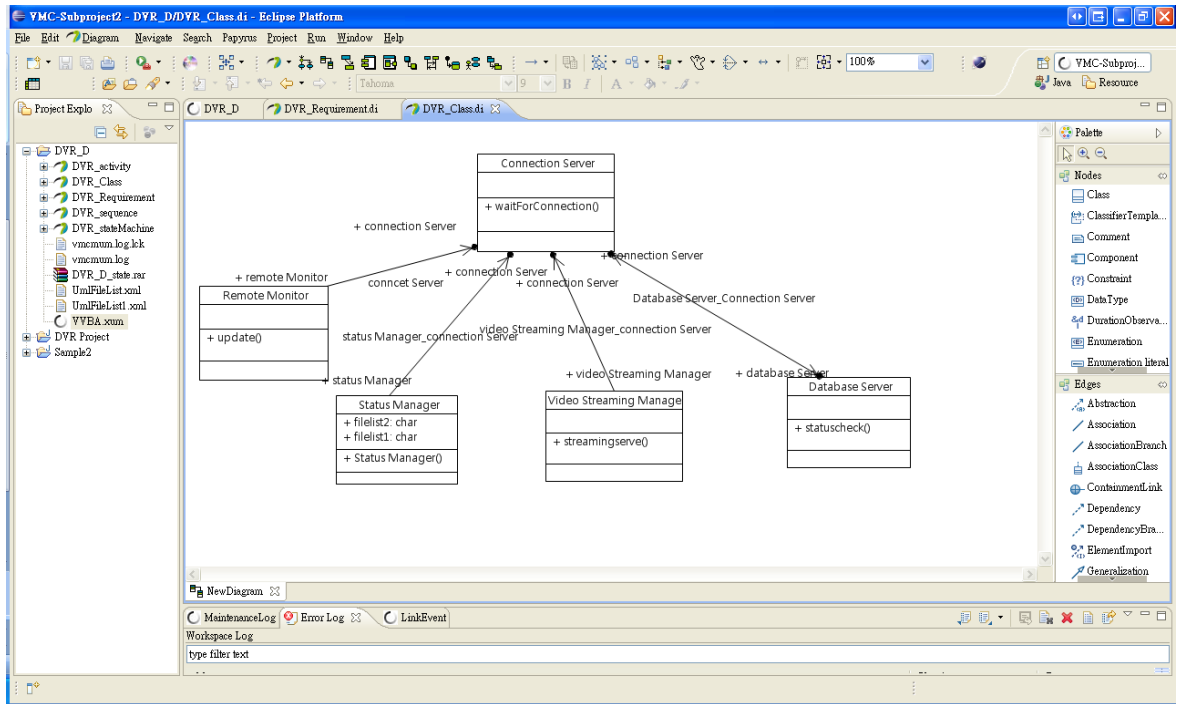


圖 二十九 DVR 的類別圖

圖 三十為 DVR 的活動圖，表現了此系統的行為流程，此監視系統先經由攝影機接受影片，之後連接伺服器進行視訊流處理和狀態管理，之後再連接伺服器將影片資訊等保存入資料庫伺服器中，其中限制了整體流程中的執行時間限制及連結伺服器的限制時間等資訊，圖 三十一為系統的循序圖表現了系統部件間的互動流程，也同樣存在有時限限制等。

從這些模型圖中，從需求階段的案例圖及需求圖等到設計階段的類別圖、活動圖、循序圖等圖，可以表現出整個監視器系統的需求及構成和運作等流程，並利用我們所作的輔助系統可以方便的觀察並切換各種系統專案模型圖。

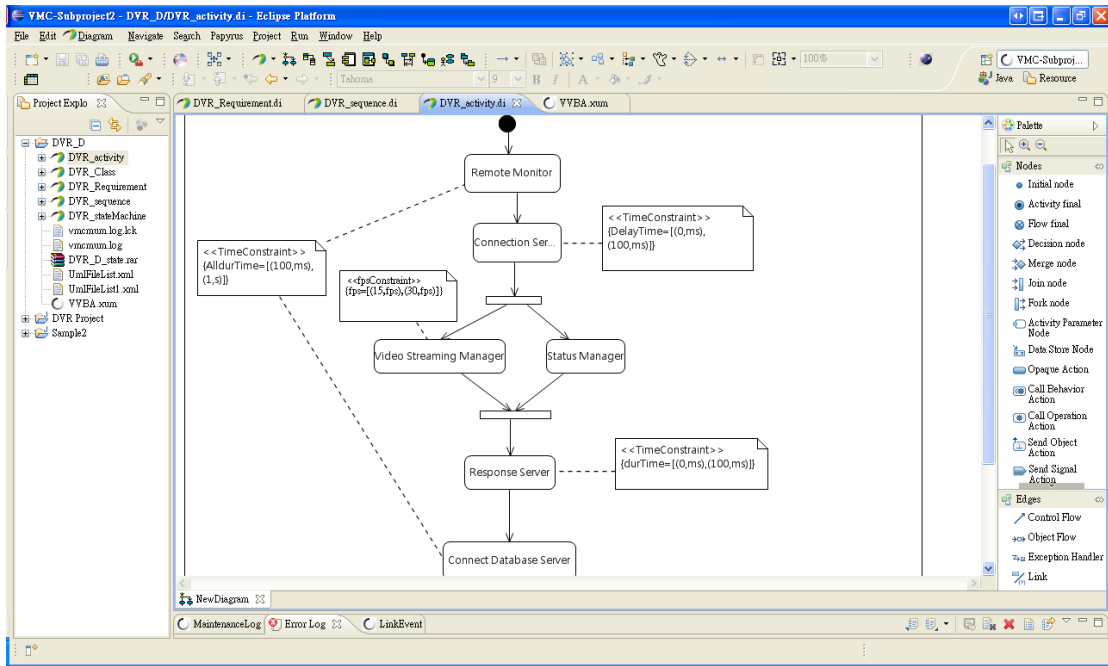


圖 三十 DVR 的活動圖

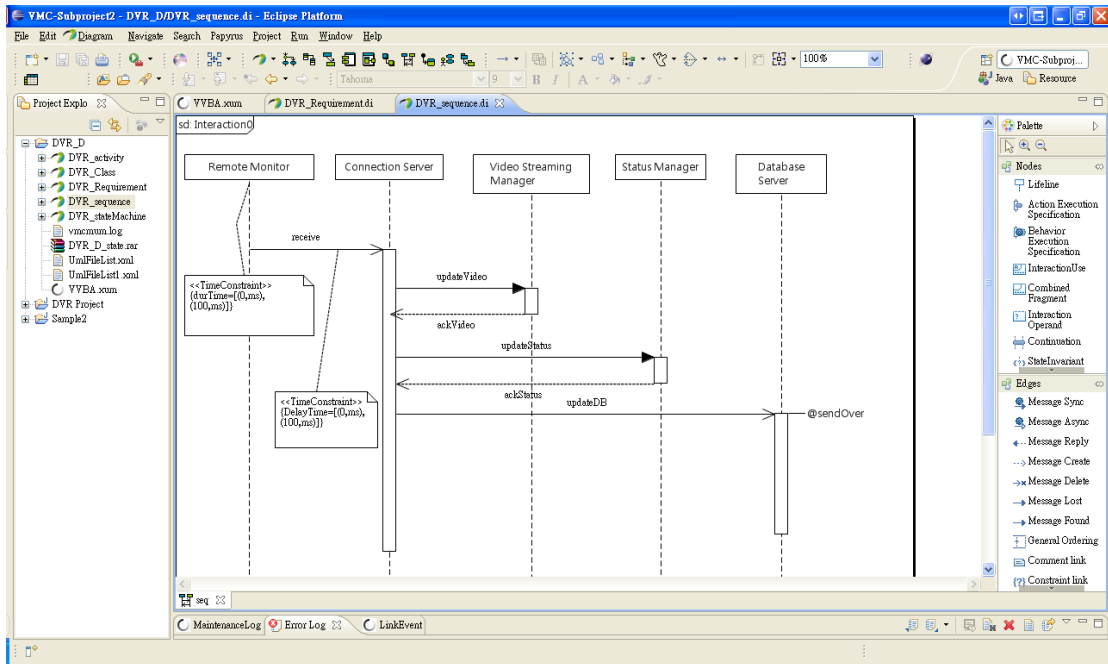


圖 三十一 DVR 的循序圖

4.2 系統畫面

系統畫面分成兩種，一種是元件列表，一種是需求樹追蹤表，如

圖 三十二，經由主畫面上的 Traceability Tree 按鈕可進入需求樹追蹤表。

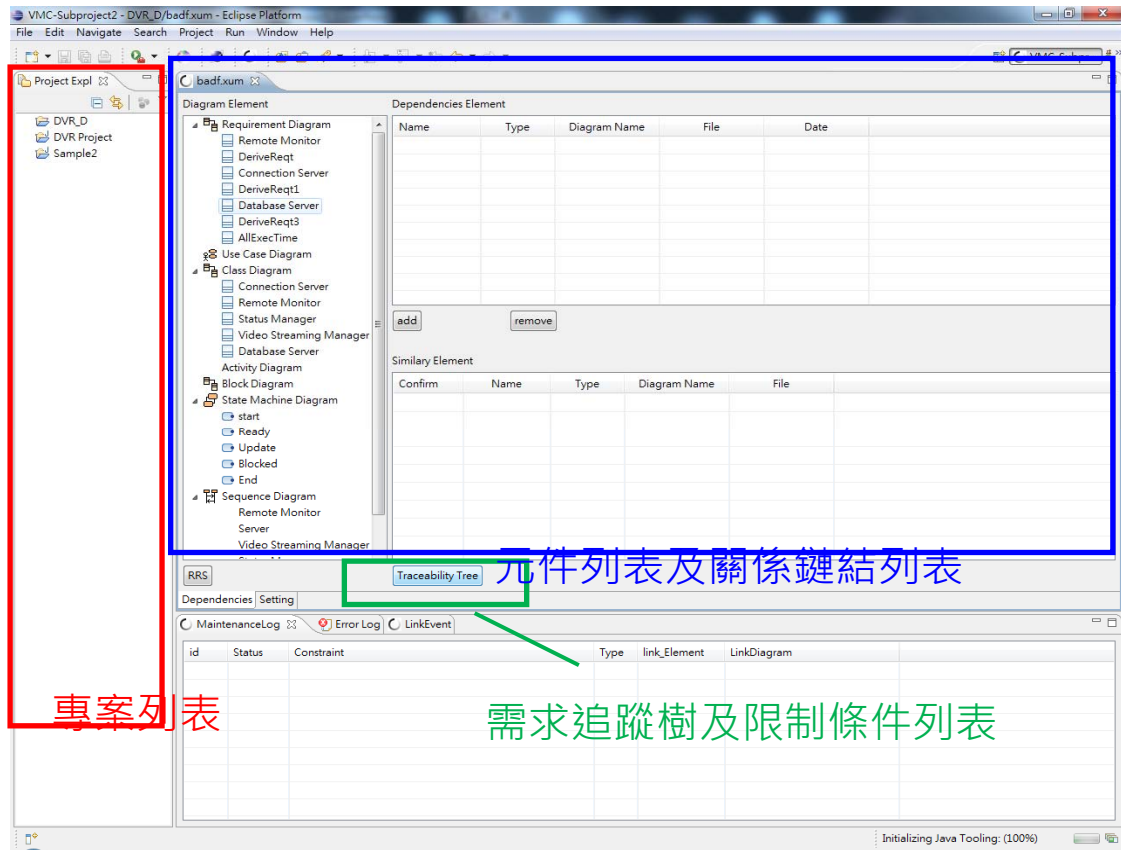


圖 三十二 主畫面示意圖

在主畫面上，我們從專案中的各模型圖，如上一節中的各種階層模型圖中的 UML 檔案內抓取需求的重要資訊，如檔名、來源、物件名、物件 id、類別等，將各類型的模型圖中的相似物件經過字串比對，建立跨階層的抽象鏈結及同階層的整合鏈結，經由元件列表可以看出不同階層及相關聯的物件資訊，如圖 三十三，我們補全嵌入式系統所需的 UML 及 SysML 的軟體模型類別，在主畫面可直接列出，方便使用者觀看。

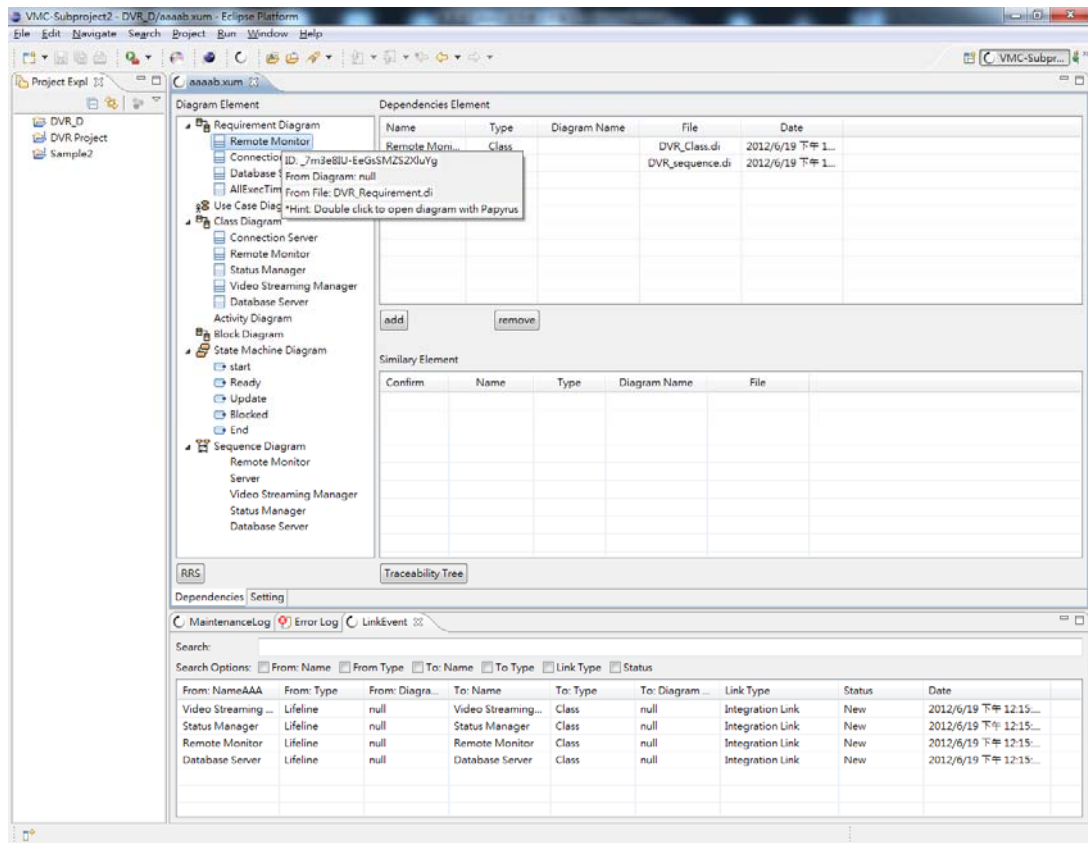


圖 三十三 主畫面列表

在追蹤性上，經由主畫面上的 Traceability Tree 按鈕可進入需求樹追蹤表，如圖 三十四，追蹤表以樹狀結構表現出需求物件與其相關鍵結物件，並顯示出其細節資訊及限制資訊，是為了方便使用者能同時觀察到需求相聯性並追蹤到相關聯性的模型，使用者單點擊節點物件可顯示其物件及限制的資訊，雙擊節點物件可進入相關聯的軟體模型圖型中。

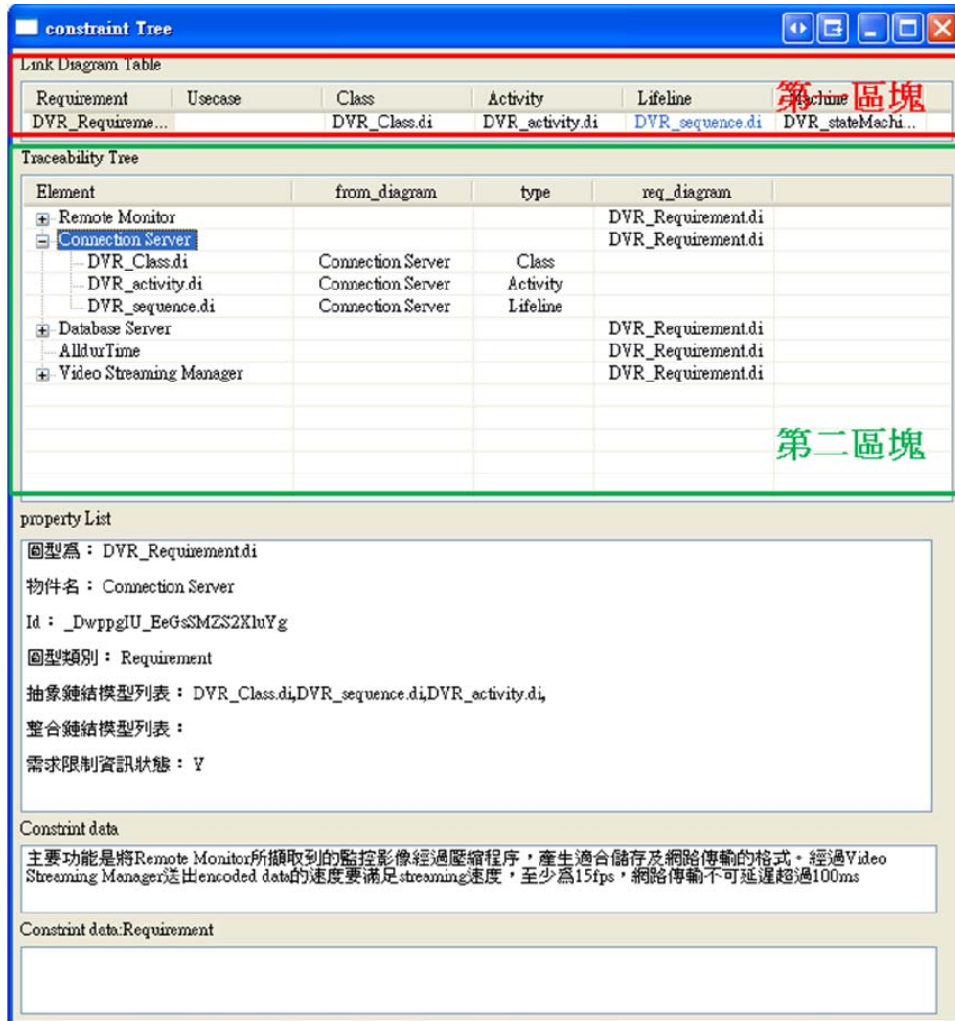


圖 三十四 需求追蹤樹及其資訊列表

此面版第一區塊為各類型圖型表格，依圖型類別分類，點擊列表中的物件可直接打開此圖檔，方便使用者立刻進入想進入的圖檔。

此面版第二區塊 Traceability Tree 列表為一樹狀結構，在根節點的物件為需求物件，其子樹結構為與其有抽象鏈結的其他軟體模型物件，子樹中的平行的節點物件表現相關的出整合鍵結。

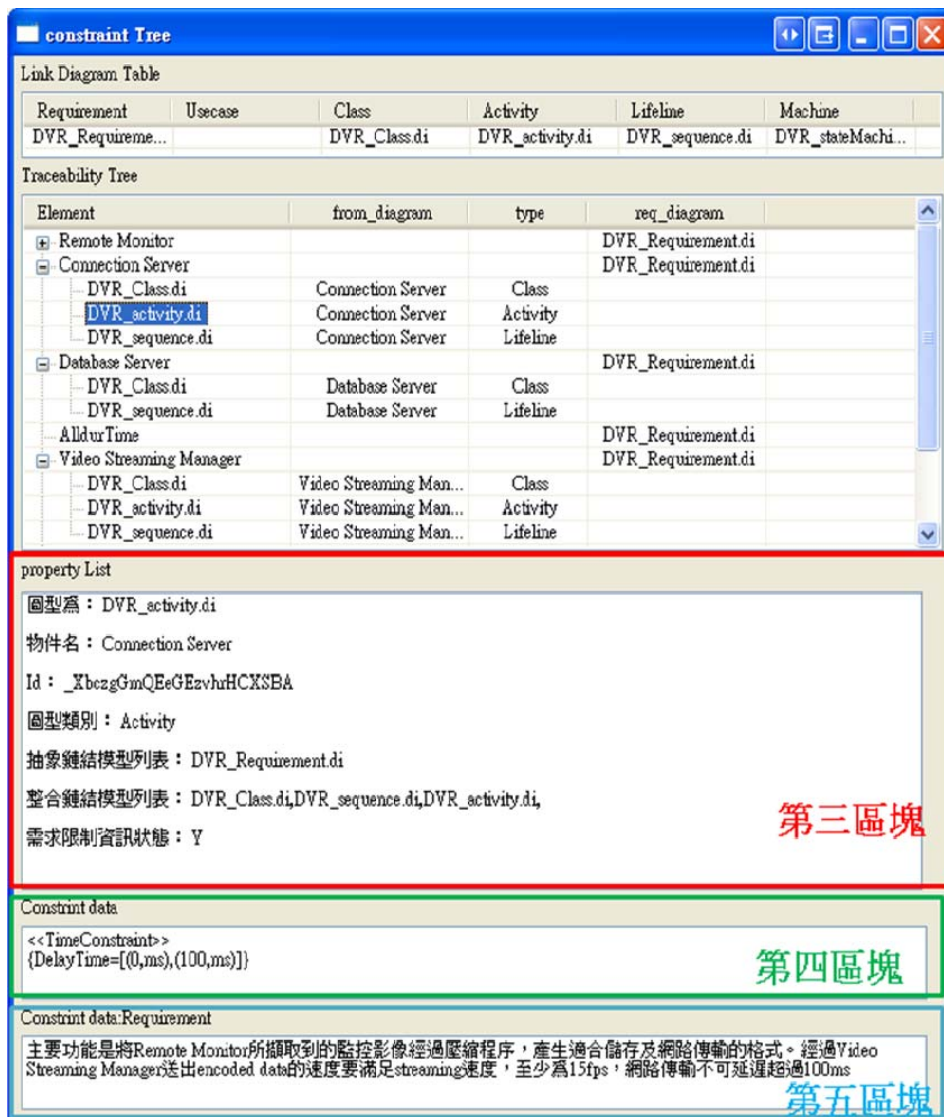


圖 三十五 需求追蹤樹展開

如圖 三十五，第三區塊 Property List 是為了表現需求物件及相關物件的完整資訊，從需求物件的名稱、uml 檔中的唯一 id、圖型類別、來源檔，抽象鏈結模型列表為與其相關聯的抽象鏈結圖型、整合鏈結模型列表為與其相關聯的整合鏈結圖型，需求限制資訊狀態表示為 Y 則代表限制鏈結兩方皆有限制資訊，若為 N 則代表需求圖中的物件含有需求資訊，但被選中的圖型物件卻缺少，可能是因為人為疏

失。

第四區塊顯示了此物件中下的需求、限制資訊及 MARTE 圖型中的限制資訊，此區塊除了從 uml 檔中抓取的需求、限制資訊外，因為 uml 檔中會對當初使用者輸入的資訊中的某些特殊字元，例如 { “,<,>,,…} 等進行字元替換，因從 uml 檔中直接抓出的資訊會出現人眼無法辨識的亂碼，需經過程式轉譯後才以使用者可以理解的資訊內容表現在區塊上。

第五區塊顯示需求圖中相關物件中的需求資訊，方便使用者對照需求圖中及被選中的圖型物件下的限制資訊，可以立刻知道需求資訊的內容。

此面版除了可顯示資訊外，針對限制資訊比對後出現不同的狀態，使用可直接在版面上進行檢視修改，並於確認修改完成後按下 ok 按鈕確認修改，會再將資訊中的特殊字元進行轉換後修改目標資料夾中的 uml 檔、資料庫中的 uml 檔及 XUM 檔，並再次進行限制資訊的比對。

使用者可利用此面版直接檢視專案中的軟體模型裡的需求、限制等，並經由追蹤性直接修改所需的資訊，不需要重覆查找專案模型圖，在模型圖中花時間修改及比對，可減少使用者在軟體流程中的時間花費。

第五章 結論及未來展望

隨著軟體專案及模型語言的進化，對軟體模型驅動模型研究愈加成熟，並且隨著智慧型行動裝置及智慧型機器的流行，嵌入式系統的專案的軟體工程研究也十分熱門，而我們用以模型驅動的方法建立嵌入式系統專案，希望能加快嵌入式系統專案的進程。

延續 XUM 的研究，我們將嵌入式系統專案的模型在需求及現實實行限制上進行完整的資訊整合，除了整合鏈結關係之外，我們針對可用於表示嵌入式模型的 SysML 及 MARTE 模型新增的資訊，新增了限制結構，在不同物件及不同模型圖及層級間建立鏈結，並將 SysML 及 MARTE 新增的模型加入 XUMM 架構中，整合資訊進入 XUM。

需求資訊及限制資訊間的追蹤性可以將需求圖與其他階層的模型圖中的相關物件建立物件鏈結及需求、限制鏈結追蹤，當使用者不慎輸入錯誤的資訊，經由其追蹤性關聯，可讓使用者可以提早發現問題，減少人工檢查的時間及人為錯誤。

因需求資訊雖無法對自然語言進行解析，限制資訊等已設定為規格化的 VSL 語言，以便於追蹤性的資訊比對。

我們在 Eclipse 環境上，搭配 papyrus 套件所提供的模型專案，使用 Eclipse 外掛的方式建立系統，使得使用者可以同時更動觀察模型

間的追蹤性及直接打開想修改的圖型中改動限制資訊。

在未來的工作，希望能擴充測試階段的模型及實作自動化測試，讓此系統能完整應用於專案全流程，並更加的方便自動化。

參考文獻

- [1] U. Rasthofer, "Component-based software engineering for distributed embedded real-time systems", *Software, IEE Proceedings*, vol. 148, pp. 99-103, 2001.
- [2] A. W. Brown, S. Iyengar and S. Johnston, "A Rational approach to model-driven development", *IBM Systems Journal*, vol. 45, pp. 463-480, 2006.
- [3] Object Management Group. "OMG unified modeling language specification. Version 2.3", *Retrieved Nov 15, 2010* from <http://www.omg.org/spec/UML/2.3/>
- [4] Object Management Group. "OMG Systems Modeling Language", from <http://www.omg.sysml.org/>
- [5] Object Management Group. "The UML Profile for MARTE: Modeling and Analysis of Real-Time and Embedded Systems", from <http://www.omg.marte.org/>
- [6] A. Egyed and P. Grunbacher, "Identifying requirements conflicts and cooperation: how quality attributes and automated traceability can help", *Software, IEEE*, vol. 21, pp. 50-58, 2004.
- [7] O. Gotel and S. Morris, "Requirements Tracery", *Software, IEEE*, vol. 28, pp. 92-94, 2011.
- [8] D. Chamberlin, "XQuery: An XML query language", *IBM Systems Journal*, vol. 41, pp. 597-615, 2002.
- [9] Object Management Group. "CORBA®, XML And XMI® Resource Page", from <http://www.omg.org/technology/xml/>
- [10] Object Management Group. "XML Metadata Interchange(OMG XMI)", from http://www.jeckle.de/files/xmi_mdc.pdf
- [11] Sedna. "Sedna: Native XML Database System", from <http://www.sedna.org/>
- [12] R.E. Johnson and B. Foote, "Designing reusable class" *Journal of Object-Oriented Programming*, pp 22-35, 2005
- [13] J. Holt and S. Perry, "SysML: describing the system [Notebook: modeling languages] ", *Information Professional*, vol. 3, pp. 35-37, 2006.
- [14] F. Valles-Barajas, "A SysML-Based Methodology for Manufacturing Machinery Modeling and Design", *Latin America Transactions, IEEE*, vol. 8, pp. 259-268, June 2010.
- [15] Y. Vanderperren and W. Dehaene, "UML 2 and SysML: an approach to deal with complexity in SoC/NoC design", *Design, Automation and*

- Test in Europe*, vol. 2, pp. 716-717, 2005.
- [16] K. Triantafyllidis and E. Bondarev and P.H.N. de With, "Low-Level Profiling and MARTE-Compatible Modeling of Software Components for Real-Time Systems", *Software Engineering and Advanced Applications (SEAA)*, pp. 216-223, 2012.
- [17] H. Yu, T. J. P., L. Besnard, T. Gautier, F. Mallet, C. André and R. de Simone, "Polychronous Analysis of Timing Constraints in UML MARTE", *Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW)*, pp. 145-151, 2010.
- [18] F. Mallet and R. Simone, "MARTE: A Profile for RT/E Systems Modeling, Analysis and Simulation? ", *First International Conference on Simulation Tools and Techniques for Communications*, pp. 1-8, 2008
- [19] H. Espinoza, D. Cancila , B. Selic , and S. Gérard, "Challenges in Combining SysML and MARTE for Model-Based Design of Embedded Systems", *Model Driven Architecture - Foundations and Applications Lecture Notes in Computer Science* ,vol. 5562 pp. 98-113,2009
- [20] Y.H. Tsai and C.W. Chu, "Enhancing Requirements Gathering Practices With an XML-based Unified Model by Use Case Modeling", *2003 Symposium on Application and Development of Information Technologies*, 2003.
- [21] C.W. Chu and C.H. Wang, "An XML-based Meta-model for Process and Agent-based Integrated Software Evolution environment (PRAISE) ", *the Proceedings of SEKE 2005* ,pp.324-329, 2005
- [22] T.Y. Chao, P.Y. Lin, C.H. Chang ,C.W. Chu and C.T. Hsien, "XML-based Embedded Software Reusable Component Repository", *Object-Oriented Technology and Applications (OTA) and Software Engineering (TCSE)*,pp.353-358, 2010
- [23] C.H. Chang , C.W. Lu , C.W. Chu , C.T. Yang , P.A. Hsiung, N.L. Hsueh and C.S. Koong, "XML-based Reusable Component Repository for Embedded Software", *Proceedings of the 5th IEEE International Workshop on Quality Oriented Reuse of Software*,pp.345-350, 2011