

私立東海大學資訊工程與科學系研究所

碩士論文

指導教授：朱正忠 教授

以 SysML 為基礎之系統需求塑模流程

A SysML Based Modeling Process for System Requirement

研究生：陳嘉璟

中華民國九十八年七月二日

摘要

需求工程在系統開發的生命週期中扮演著重要的腳色，在進行系統開發時，系統需求的撰寫經常是透過自然語言來描述，但是由於自然語言通常會有人為的不確定性，或是需求工程師對於該問題領域相關知識與經驗的缺乏，往往造成系統需求規格的模稜兩可與錯誤的需求分析，為了避免使用者再次發生傳統系統開發因為需求描述不確實，而導致專案進行困難的現象，需要一套適當的需求塑模工具和方法來提供工程師適當的機制來瞭解客戶想要些什麼，分析需求、評估可行性、確認規格，提升需求描述的正確性，以降低專案失敗的機率。

研究目的在於提供一套引導使用者撰寫需求的方法，協助系統開發流程，避免使用者因為需求不明確導致專案進行困難的現象，SysML 是一種圖形塑模語言，能讓使用者在需求建立階段避免語言上的誤會以及需求的不一致，模型驅動架構之軟體設計概念則提供使用者一個需求擷取的流程避免錯誤的需求擷取，在本研究中我們導入 SysML 塑模方式以及模型驅動架構之軟體設計概念，設計提供複雜應用系統所需之圖形塑模環境以及需求內容相關模組，並套用一個數位錄影系統需求做為研究範例，最後產生一個以 XML 為基礎之整合式系統文件，提供給設計、架構、程式碼以及測試相關流程使用。

關鍵字: 需求工程，需求塑模，塑模語言，模型驅動

Abstract

Requirement engineering plays an important role in the life cycle of system develop, the system requirement specification is used to be written with natural language format. It can be easily leads to a situation that full of ambiguity, inconsistency, imprecision and incompleteness. It is necessary that to propose a proper requirement modeling tool and modeling approach to help system engineer. So that they could clearly realize what customers need. Then Analyze requirement and Feasibility, Consultations at a reasonable solution, present the solution clearly, confirm specification. So engineers improve the accuracy of requirement description that can reduce the project failure probability.

In this paper, we propose an approach to describe requirements, assist the system development process, and avoid the same fate as the ambiguously user requirement leads to project delay or fail. Also we use SysML modeling as well as model-driven architecture concepts. In our research, we provide a SysML based requirement modeling environment and some requirement related model for complex application system. And we applied research as an example of system requirements, and finally to generate a XML-based integrated system files, made it available for design 、 architecture analysis 、 coding and testing and other relevant processes.

Keywords: Requirement Engineering, SysML, Model-driven, Requirement Modeling

章節目錄

摘要.....	I
ABSTRACT.....	II
章節目錄.....	III
圖目錄.....	V
表目錄.....	VII
第一章 導論.....	1
1.1 前言.....	1
1.2 研究動機與目的.....	2
1.3 研究方法與步驟.....	4
1.4 章節導讀.....	5
第二章 知識背景與相關研究.....	6
2.1 專案生命週期.....	6
2.2 系統需求分析.....	7
2.2.1 需求分析探討.....	7
2.3 需求塑模.....	8
2.4 SysML.....	9
2.5 UML Profile.....	11
2.6 Stereotype(原型).....	11
2.7 Papyrus SysML profile.....	12
2.8 OMG MARTE.....	12
2.9 XML.....	13
2.10 OCL.....	14
2.11 需求管理工具.....	16
第三章 需求塑模流程.....	17
3.1 需求擷取流程.....	18
3.2 需求引導問題樣版.....	21
3.3 CIM 階段.....	28
3.4 PIM 階段.....	29
3.5 OCL 條件限制.....	31
第四章 案例研究.....	32
4.1 實做環境.....	32
4.2 系統畫面.....	33

4.3	塑模與 Profile.....	34
4.4	CIM 階段需求分析.....	38
4.5	PIM 階段.....	40
4.5.1	分析動態規則.....	40
4.5.2	描述靜態結構.....	42
4.6	匯入與匯出 XML 需求文件.....	45
第五章	結論與未來展望.....	47
參考文獻	48
附錄一	PVE 子系統 SYSML 需求圖.....	50

圖目錄

圖 1 系統開發流程示意圖.....	6
圖 2 SysML 與 UML2 涵蓋範圍分布圖.....	10
圖 3 SysML Requirement diagram stereotype.....	12
圖 4 系統架構圖.....	18
圖 5 需求資訊建立流程圖.....	19
圖 6 System State Diagram.....	20
圖 7 使用者輸出 XML 的使用案例.....	20
圖 8 需求圖描述功能需求.....	28
圖 9 VSS Connection Server block	30
圖 10 系統初始畫面.....	33
圖 11 建立初始的需求圖.....	34
圖 12 Select profile	34
圖 13 Requirement template Profile	35
圖 14 需求量化單位.....	36
圖 15 輸入需求樣版.....	36
圖 16 檢驗 Profile 內容錯誤.....	37
圖 17 驗證需求內容發現使用者輸入錯誤資料型態	38
圖 18 Use Case.....	39

圖 19 VSS system requirement diagram prototype	39
圖 20 要求啟動 Connection Server.....	40
圖 21 進入閒置狀態.....	40
圖 22 接收與傳輸.....	41
圖 23 回到閒置狀態.....	41
圖 24 Connection Server State Diagram.....	41
圖 25 Papyrus 產出 VSS 區塊定義圖	43
圖 26 PVE subsystem Requirement Diagram.....	44
圖 27 匯出 XML 需求文件.....	46

表目錄

表 1 OCL description type	15
表 2 DVR 需求訪談內容	23
表 3 功能性需求樣版 Profile.....	28
表 4 定義狀態轉移動作時的操作.....	30
表 5 Connection Server BDD Profile 內容	30
表 6 OCL 限制.....	31
表 7 定義狀態轉移動作時的操作.....	42

第一章 導論

1.1 前言

由於科技日新月異，使得系統開發的time to market在產品生命週期中逐漸縮短，產品開發的複雜度和異質性也與日俱增，然而與產品開發有關的專案關係人(Stakeholder) 一般來說並非全部都具備有良好的資訊相關知識，此時若是需求工程師又缺乏良好的訓練或者是該領域的相關知識和經驗，就會導致制定需求規格時的不明確，而影響到後續分析、設計與測試等相關流程的進行，這樣的因素會造成專案失敗或是產品上市時間延宕的機率相對的提升，如果能夠在系統需求建立的同時，輔以有系統的圖表，將有助於釐清產品開發時對系統功能特性的模稜兩可，產出明確的需求規格。

OMG SysML是一種利用一般性圖形塑模語言用來分析、設計和驗證系統設計規格，其中包括軟、硬體、資訊、人員以及設備之大型複雜系統。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模型分析結合，在本研究中我們將整合一個系統，來應用SysML對一個研究案例進行系統需求塑模及相關的分析。

1.2 研究動機與目的

檢視整個軟體開發流程與並探討各個階段之間的連貫關係，各階段發展的標準、方法步驟以及應用工具，卻因為缺乏彼此之間協調與整合的考量機制存在，致使軟體開發者在每一階段，必須以人工方式檢視並協調整合來自不同標準、方法及應用工具之來源資料，也因不同設計之間的統合、細部分工等作業程序，使得軟體開發與研究者無法以一個一致性、協同性的開發方式進行軟體系統的開發作業，而使得整個軟體開發成本居高不下，其中需求分析階段在一個系統開發生命週期中，是最初的階段也是一個很重要的流程，需求工程師經由需求訪談的過程來引導客戶並且與客戶確認需求內容後，取得最原始的使用者需求，這是最初的需求分析文件，而從需求擷取到最後的需求確認，這些與需求相關的流程稱為需求工程[1]，此時需求的描述大多數還是用自然語言的方式所撰寫，但是這種建立需求文件的方式，必須完全依靠需求工程師的經驗與知識來進行，如果需求工程師本身的專業涵養不足，則用這種方式來制定需求文件就很容易產生需求的模稜兩可(Ambiguity)、需求不一致(inconsistency)以及需求不完整(incomplete)、不精確(Imprecision)的狀況，不好的需求[2][3][4]將

導致專案的執行進度延宕，而專案執行過程所發生的錯誤也會倍增，因此，需求階段其實非常需要專業的投入，而多數的軟體產業都是以非專業的方式來處理相關的工作。

知名的研究機構 Standish Group 在一篇報告中對一群中高階資訊技術執行經理做了一份問卷調查[5]，這份調查報告中提供了一些關於影響專案成功與失敗的因素，其中一項因素是”清楚的陳述需求”，排名第三，有百分之十三的高階經理人認為這項因素是最為影響專案成敗的重要條件，而另一個因素是”不完整的需求”，有百分之十三點一的中高階經理人認為這是使專案失敗最大的不利因子。

由此可見，一份完整且清楚的需求描述是左右軟體專案流程順利與成功的關鍵要素，而本篇研究的重點就在於如何結合模型驅動架構與 OMG SysML 以及配合 OMG Profile，來增加需求撰寫的完整性以及清楚的需求陳述的流程，進一步使系統軟體開發流程更加順利。

1.3 研究方法與步驟

本研究的進行方式，是先搜集現有的相關文獻以及相關的開發工具和現有產品的調查，包括軟體工程相關文獻，例如需求工程、專案管理、需求管理、物件模組化等，以及模型驅動架構與 SysML 的規格，工具則是 borland、Eclipse、與 Papyrus 等圖形塑模工具以及相關的背景知識及原理，並加以研讀和熟悉工具操作介面與系統原理，以便於在開始實作範例研究時能順利進行。

接下來就是實作我們所提出的流程，在逐步推導流程的同時，一邊確認是否符合初步訂定的目標，然後對現有開放源碼資源的整合元件做系統整合性測試，並且套用一個範例來進行實際的案例研究，再依所得到的結果來對我們提出的方法做客觀的結論與評估。

1.4 章節導讀

論文的章節安排分述如下：

第二章 在 第 這 個 章 節 們 將 介 紹 本 篇 論 文 的 相 關 研 究 以 及 背 景 知 識 來 陳 述 基 本 觀 念 ； 第 三 章 我 們 將 對 研 究 的 系 統 需 求 塑 模 流 程 做 進 一 步 的 說 明 ； 第 四 章 我 們 將 使 用 塑 模 工 具 並 套 用 一 個 範 例 來 做 案 例 研 究 ； 第 五 章 最 後 將 對 研 究 做 一 個 結 論 與 說 明 未 來 的 工 作 以 及 展 望 。

第二章 知識背景與相關研究

2.1 專案生命週期

標準程序的軟體開發流程可由四個階段所組成，分別為需求分析階段(Requirement Phase)、規劃設計階段(Analysis and Design Phase)、實作階段(Implementation Phase)、以及維護階段(Maintenance Phase)，如圖 1[6] 所示，系統發展流程由需求階段開始逐一進行，直到維護階段形成循環的開發流程，其間各階段間也可經由結果的度量做往上修正的動作，或對原先設定的目標做修正，一個計畫失敗的主要因素之一，就是缺少一份良善的需求文件[8]，在本篇論文中，我們將重心放在 Requirement Phase，並採用圖形化塑模的方式幫助使用者建立完整的需求圖。

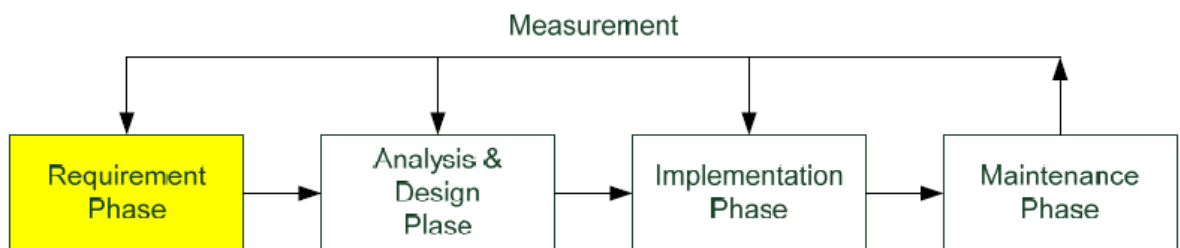


圖 1 系統開發流程示意圖

2.2 系統需求分析

一個系統的開發是從了解使用者需求為起點，不論是用戶簡單的需求描述或是正式的需求規格書，使用者需求是說明使用者在功能性、操作行為、產品效能以及一些限制的要求，系統分析師透過需求問卷、需求訪談等需求擷取方式得知使用者要利用所開發的系統來解決什麼樣的問題以及系統具有哪些目標和功能上的要求，分析師將這些文件內容具體整理之後，讓使用者確認需求是否符合使用者所期望，並且將整理後的文件提供系統後續流程使用。

使用者需求擷取是系統開發較為重要的部份，但也容易產生錯誤的需求擷取，因為系統需求具有模糊性、易變性還有使用者主觀想法的特性，所以要清楚明確的將需求表達出來是較有難度的，而系統需求在一個專案生命週期中，將會影響設計、原始碼開發、測試文件等內容，因此縮短掌握正確需求的時間及過程，是影響系統開發成功與否的要素。

2.2.1 需求分析探討

系統分析師將需求訪談之後的內容理解與分析，建立相關資訊、功能與系統行為的模型，將初步的使用者需求更加有條理並

且更精細完整，最後彙整成需求規格說明書，這一連串的步驟形成系統發展生命週期的需求分析階段，由 Kotonya[9]提出的觀點導向式需求定義(Viewpoint-Oriented)法，使用需求誘出與服務導向(Service-Oriented)架構，站在系統中不同的觀點描述需求，並定義成屬性、服務或者事件，情境(scenarios)的方式通常用於描述系統的行為需求(behavior requirement)，而使用案例圖是最常被使用的，使用案例圖示一種模組化的統一模式語言表示法，目的是表現使用者的目標而非系統功能，描述外界觀點，在使用案例中，一個人可能會是多種操作者(Actor)，一個角色也未必是指單一使用者，使用案例圖可以用來瞭解系統需求並同時協助系統的測試。

2.3 需求塑模

塑模工具可以說是軟體基礎工程主要的部份，讓我們在有效的溝通方式下，能充分掌握及滿足需求，以快速、有系統，可追溯的方式，完成各種軟體開發專案，OMG (Object Management Group)組織[10]提出的UML (Unified Modeling Language)[11]規範了標準的模型符號(Model Notations)來做為系統開發人員對系統的描述，UML是一套用來表達物件導向分析的標準語言，以UML來塑模系統，除了因為有共同的溝通符號能減少溝通時的誤會

外，也能以系統模型的資訊來驗證設計是否符合系統需求。

OMG在2001年又提出MDA (Model Driven Architecture)[12]的軟體發展概念，在MDA的架構中，包含三種重要的模型層次，分別是：CIM (Computational Independent Model) 、PIM (Platform Independent Model) 、以及PSM (Platform Specific Model) 。而其基本精神在於先不考慮任何資訊技術，根據需求來建構CIM級的需求模型，CIM試著去表示系統環境以及需求，而不涉及系統內部結構與運作細節；PIM著重系統內部細節，但不去干涉系統實體平台的制定；PSM則是真正落實於系統在某個特定實體平台的細節，例如.NET、Spring就是一種實體平台；而PIM級與PSM級的模型，可以彼此間根據MDA制定之標準以及模型間所定義的轉換規則做自動轉換。

2.4 SysML

OMG SysML(System Modeling Language)[13][14]是為系統工程應用開發的標準塑模語言，SysML 技術的設計是為了提供便利使用且功能強大的系統塑模架構，尤其 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，以解決範圍廣泛的系統工程問題，是一

個圖形化的塑模語言，其應用範圍相當廣泛，包括硬體與軟體的分析、設計、驗證等等都可透過 SysML 來表達，SysML 以 UML 2[15]為基礎做延伸發展，包含 UML 2 的圖形應用，並且修改或新增圖形來擴展 SysML 的功能面，所以 SysML 也繼承了 UML 的優點，SysML 沿用了部分 UML 2.0 的子集，並增加了一些圖形，包含九種基本圖形：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，圖 2 為 UML 與 SysML 的涵蓋範圍示意圖。

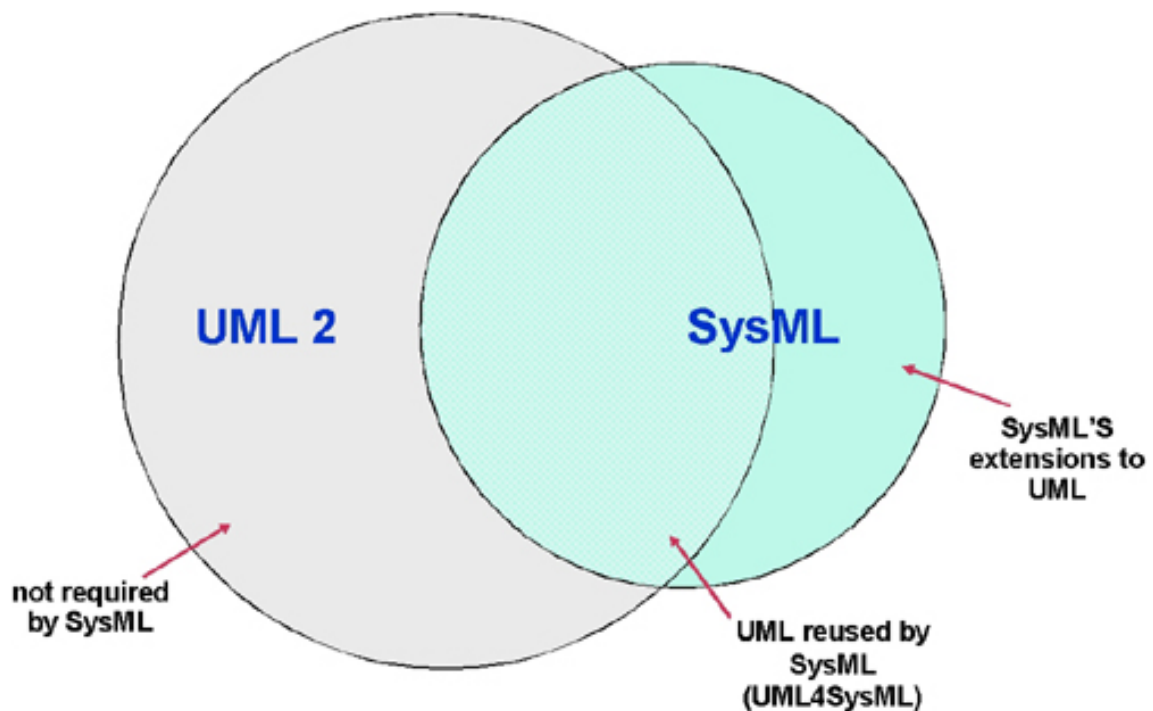


圖 2 SysML與UML2 涵蓋範圍分布圖

因此，在本研究中，我們利用 SysML 撰寫系統的需求，並整合現有的開放源碼(Open Source)來建立一個以 SysML 為基礎之

需求塑模流程，使用 SysML 來進行系統需求塑模，便能夠減少系統分析師在需求分析階段常見的錯誤。

2.5 UML Profile

Profile[16]是一種預先定義的延伸機制，它允許讓某個特定領域的專案關係人去利用擴充的方式來表達語義，來描述該特定領域的系統，Profile 並非用加入新觀念的方式來延伸 UML，相反的，它提供特定領域與 UML 之間的應用與銜接，有即透過 Profile 機制，讓我們能自訂屬於自己的 SysML 表達方式，並且利用 Profile 來建立客製化的需求樣版。

2.6 Stereotype(原型)

Stereotype[17]是一種將模型元素分類或是標記，以及引進新形態之形式元素的機制，例如 SysML 需求圖是 UML 類別圖的擴充原型而得，本研究中，我們藉由定義 Stereotype 來實作 Profile 以及制定需求樣版。

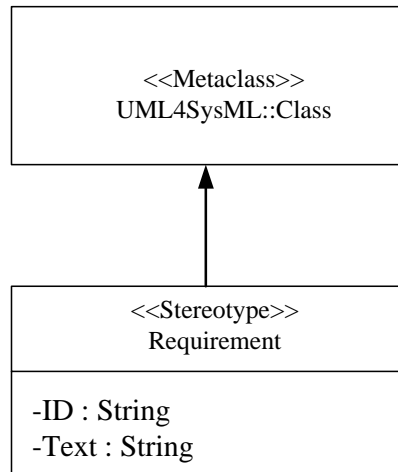


圖 3 SysML Requirement diagram stereotype

2.7 Papyrus SysML profile

Papyrus 是一套使用 Eclipse 為核心架構的非商業性 UML 圖形塑模工具，而其依照 OMG 官方所制定的 SysML 規格來製作出 SysML 的 profile，雖然 Papyrus for SysML[18]可以利用 UML 來模擬 SysML 的方式來表現 SysML 文件格式，但是它仍舊沒有提供一套有效的 SysML 需求塑模的流程，在此我們利用 Papyrus 並且加上 Profile 的特性來輔助使用者建立需求模型。

2.8 OMG MARTE

MARTE[7]是 OMG 在 2007 年所提出的 UML Profile，是 OMG 針對嵌入式即時系統需求進行塑模而建立的規範，MARTE Profile 的內容包含了三個層次，分別是基礎層(Foundation)、分析層(Analyzing)以及塑模層(Modeling)三個部份，並且各自涵蓋在三

個封包(Package)之中，基礎層用途在於嵌入式即時系統中，對於優先層級(Priority)、反應時間(Response time)、資源限制(Resource Constraint)等基本需求資料的模型建立，也是 MARTE 的核心層面；而分析層則是對於系統效能與可靠度等資訊進行分析，例如系統的可用性(Usability)、可靠性(Reliability)、安全性(Safety)以及功率消耗(Power Consume)和資源配置(Resource Allocate)等系統目標相關資訊；塑模層則提供抽象層面的塑模元素，例如即時模組的控制、資源受限的模組或是即時系統的服務與行為等；但是 MARTE 在多核心系統的塑模並無多做描述，並且缺乏對於使用者撰寫需求的導引，在我們的研究中，Profile 則是用來引導使用者撰寫需求的樣版，並且在 Profile 中導入多核心系統塑模的元素，來強化多核心嵌入式系統需求塑模，補足 MARTE 所不足的部分。

2.9 XML

XML[19]是由 World Wide Web Consortium (W3C)所制定的標準，一種延伸式的標記語言，具有擴展性(Extensibility)、結構性(Structure)、描述性(Description)、確認性(Validation)等特性。同時 XML 具有跨平台的功能，對於不同的作業系統、硬體設備、應用軟體、多元的輸入模式，開發者可以自行制定符合己身需求的

標記(tag)，做結構性的描述，促使相同的一份文件呈現不同的規格，適用於不同的軟體，符合不同的設備、滿足多重的輸入方式。XML 藉著不使用特定的語言，來達到與平台無關的目的，XML TAG 不是預先定義的，所以我們可以自行定義及撰寫自己的標籤，這樣的好處是，我們可以依照案例需求來建立自我描述的 XML。

在我們的研究方法中，使用者將需求塑模之後，利用圖形塑模工具將 SysML 轉換成 XML 之後產生的文件，進一步將這些塑模資訊與其他系統文件整合，提供給後續專案流程使用。

2.10 OCL

OCL (Object Constraint Language)[20]是一種用來表示對模型物件之限制的語言，OCL 附著在模型元素上以條件或規範的方式來表達限制規則，在某些軟體開發流程中，是用來改善及補強一些在分析及設計階段時的產出物之表現能力，它的另一個用途則是提供支援部份的 UML 延伸資訊，在 UML 模型中，用 OCL 所描述的限制條件，通常是拿來擴充物件資訊表達能力的部份做敘述的補強，使物件模型的展現方式能夠更精確更完整，在此我們也將利用 OCL 對使用者所輸入的需求內容做導引與限制。

以下是 OCL 的一些特性：

- 在 UML 模型中連接在元件上的 OCL 表示式，是為了指明或限制模型的某個方面，這一點是 OCL 的基本觀念。
- OCL 可以讓適當的塑模工具來推論 UML 模型。
- OCL 可以更精確地描繪模型。
- 在模型元件上描述一些限制條件
- OCL 可以讓適當的塑模工具依據 OCL 的表示式來產生程式碼。

在本篇論文中，我們利用這些 OCL 表示式來對 Profile 內容建立條件限制式，然後藉由驗證 OCL 所描述與限制的條件來檢驗使用者所編輯輸入的需求內容是否合乎該需求樣版資料型態的合理範圍，由此來幫助使用者輸入正確的需求文件內容，如此便能對後續產出需求文件時的內容做初步引導，表 1 是 OCL 所描述的部分表示式內容。

表 1 OCL description type

type	operations
Integer	*, +, -, /, abs
Real	*, +, -, /, floor
Boolean	and, or, xor, not, implies, if-then-else
String	toUpper, concat

2.11 需求管理工具

在系統發展的流程裡，適當的工具對各階段的開發、需求管理及維護有相當助益，並且能夠簡化許多高複雜度的工作，雖然如此，如果對工具的不熟悉與使用不當，則會造成反效果，以下是幾種商用常見的需求管理工具：CaliberRM、DOORS、Rational Requisite Pro，這些工具研發至今仍在持續更新中，其管理需求的方式也可能會因為系統更新而有所變動，例如 Borland's caliberRM 是個企業軟體需求管理工具，對於協調、衝突分析及溝通方面是很方便的軟體，允許軟體之間能更精確且可預測的傳送 key project 的 milestone，它提供所有專案需求一個集中、更妥善的專案管理，求上述三套工具各自都具備有類似的功能，也能夠定義各種需求，經過幾年的擴充和發展，在功能上可說越趨完整，但是我們在此以不同的角度，提出以 SysML 為基礎之系統塑模流程，整合 MDA 開發方式，並使用 open source 的圖形化需求塑模工具來對系統需求進行塑模以及儲存需求文件。

第三章 需求塑模流程

在本篇論文中，我們提出一個基於 SysML 的模型驅動 (Model-Driven) 需求塑模流程，圖 4 為系統架構圖，我們用 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，論文的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。

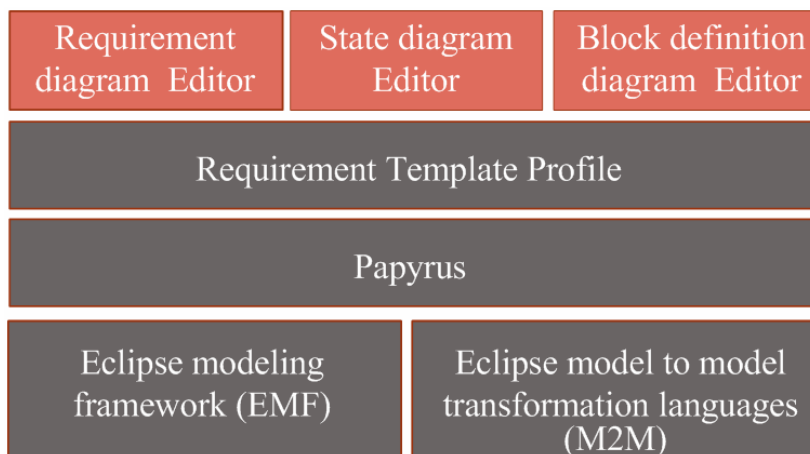


圖 4 系統架構圖

3.1 需求擷取流程

在系統開發的第一個步驟是需求文件撰寫，需求文件的品質將影響專案流程的進度，我們希望使用者在需求圖形的塑模過程中，能夠讓每個需求模型都附帶完整需求資訊，圖 5 為基於 MDA 架構的需求資訊建立流程，在 CIM 階段，使用者在需求訪談之後，套用我們所制定的需求樣版，將所得到的需求資訊輸入需求樣版，產出初步的 SysML 需求圖，並且透過需求樣版所得到的資訊來繪製使用案例圖，這個階段為需求的雛型(Prototype)，接下來 PIM 階段，建立 SysML 狀態圖(State Diagram)與區塊定義圖(Block Definition diagram)以及其所對應的需求樣版內容，透過狀態圖與區塊定義圖的分析內容可以分別得到系統靜態(Static)與動態(Dynamic)的需求，最後回到 Prototype，與原始 SysML 需求圖所提供的資訊整合之後，形成一份完整的 SysML 需求圖

(Requirement Diagram)，依據 MDA 的定義，我們所提出的塑模流程及產出，都歸納為 CIM 以及 PIM 階段，而因為產出的 XML 格式文件中，包含有系統的部署(deployment)等相關資訊，所以可提供給 PSM 階段使用，在我們的研究中並無對 PSM 階段做詳細探討。

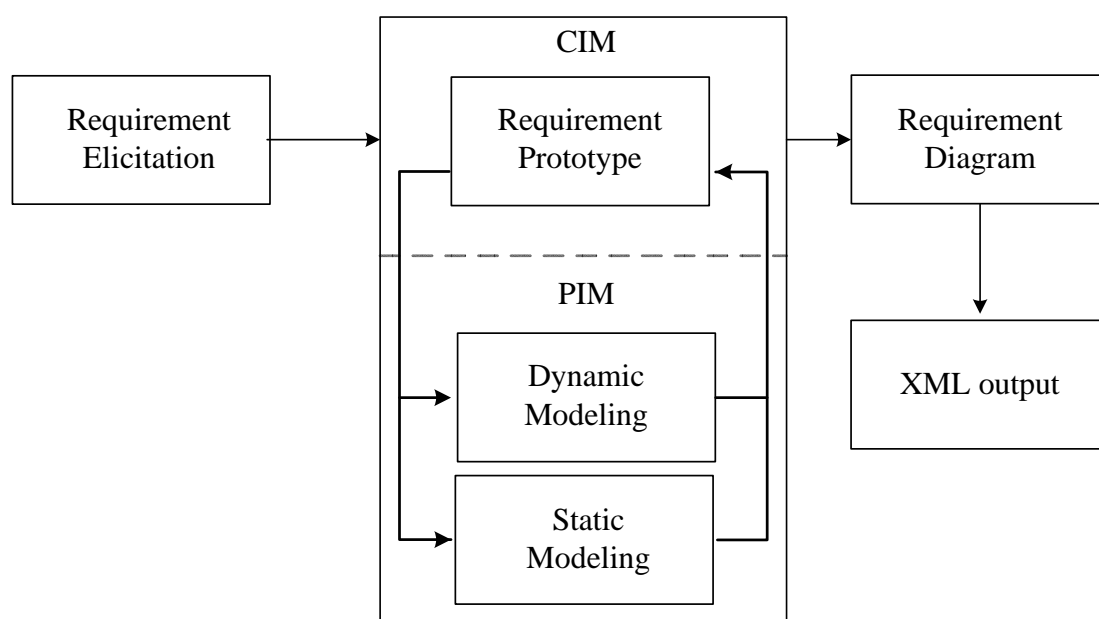


圖 5 需求資訊建立流程圖

圖 6 System State Diagram 以狀態圖來描述使用者在需求塑模的過程中，系統因為使用者的操作行為，而造成的狀態轉換及回應，使用者在開始編輯 SysML 圖形的同時，選取相關知識領域的需求樣版 Profile 之後，回到塑模編輯畫面做需求建立、修改、刪除的動作，最後存檔離開。

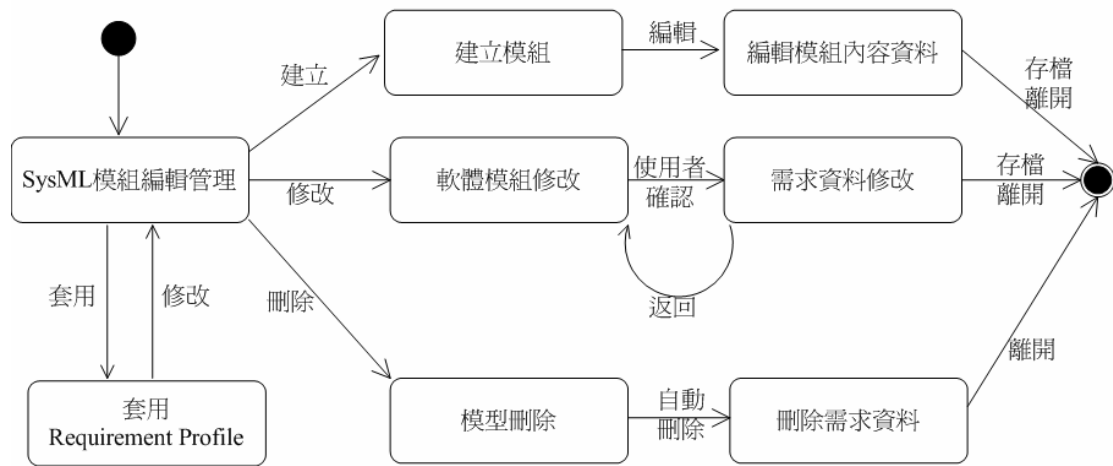


圖 6 System State Diagram

在第四章案例研究中，需求圖透過解析器將圖形轉換為 XML 文件，以 XML 格式提供系統需求文件給系統設計等後續系統開發階段所需要的資訊，圖 7 為使用者輸出 XML 的使用案例圖。

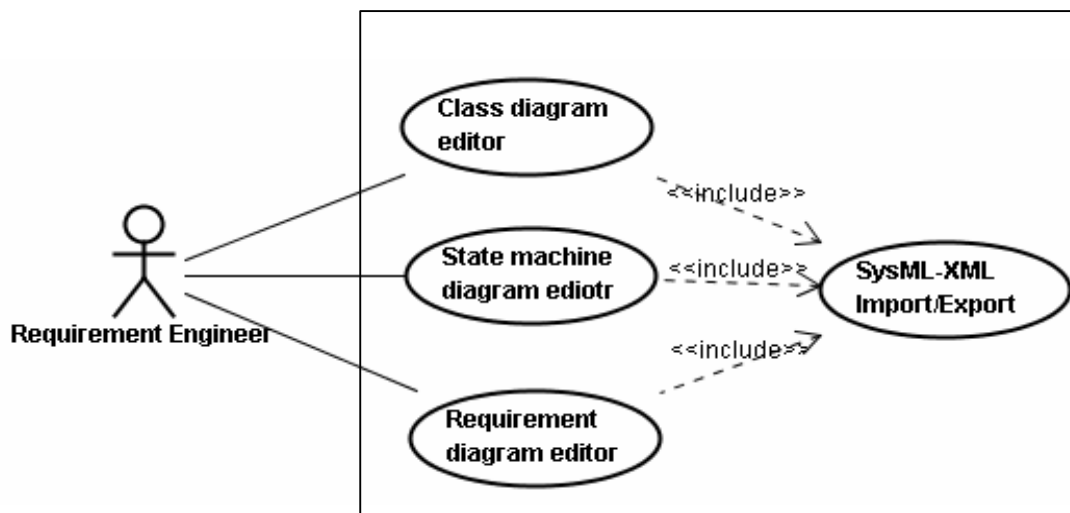


圖 7 使用者輸出 XML 的使用案例

3.2 需求引導問題樣版

需求擷取是相當重要的一個階段，為了避免因為人為的疏忽，例如工程師的資歷深淺不一，而導致需求不完整等問題發生，我們在這裡定義了一需求問題模組樣版，需求樣版(Requirement Template)的好處就是可以導引使用者依循制定的需求欄位來描述需求，避免因為前面所陳述因為需求的模稜兩可與不完整而造成需求品質的下降，提供系統分析師來進行使用者或是客戶的初步需求擷取，樣版定義如下：

1. 目標：

說明本軟體/系統的目標為何以及表現方式。

2. 軟體/系統使用者：

將會有哪些人員來使用本軟體/系統。

3. 功能性需求：

描述系統應該提供的服務，系統對輸入指令的回應方式以及特殊情況下的行為描述，例如：輸入方式為何？處理流程方法為何？輸出方式為何？

4. 非功能性需求：

系統的限制條件或是系統提供的功能，或者是系統執行效率、效能需求...等，能夠度量並且量化的項目，例如反應時間、可靠度與維護性。

5. 系統環境:

(i) 硬體需求

(ii) 執行環境系統，是否配合它種應用軟體。

(iii) 傳輸環境，是否需要網路傳輸。

表 2 為需求訪談內容，透過套用需求樣版來描述一個數位錄影系統(Digital Video Recorder, 以下簡稱 DVR) 的子系統之客戶需求，以表格來展示需求訪談內容。

表 2 DVR 需求訪談內容

項目	內容
系統目標	<p>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，系統主要目的為提供使用者一個使用介面，透過該介面，使用者可以經由網路連到系統主機，使用各種網路監控的功能。</p>
使用者	<ol style="list-style-type: none"> 1.警衛 2.住戶 3.保全人員
功能性需求	<ol style="list-style-type: none"> 1.系統要能提供遠端的客戶端連線。 2.系統要能夠依照 client 的要求去呼叫相對應的元件來提供 service。 3.系統可以進行即時影音傳輸。 4. ...
非功能性需求	<ol style="list-style-type: none"> 1.客戶端必須要在 5 秒內連線。 2.系統必須維持傳輸都在 50bps 以上。 3.系統最多可以同時處理 5 個客戶端連線。 4.系統可以同時進行 2 個以上的即時影音傳輸。 5. ...
系統環境	<p>(i) 硬體需求: 伺服器需要 4G 的記憶體/10TB 的儲存裝置 客戶端主機需要 1G 的記憶體</p> <p>(ii) 執行系統環境: 伺服器需要 SQL 以及 windows 2000/NT 以上作業系統 客戶端需安裝 windows XP 以上作業系統</p> <p>(iii) 傳輸環境: 伺服器主機:須具備 100Mbps 網路傳輸速度</p>

需求問題樣版有系統的定義問題來協助分析師擷取初步的需求，並定義出相關需求規格來引導分析師撰寫需求內容，以下為細部需求規格說明，分為兩個部份：

A:需求基本資料(Requirement information)

(1) 需求種類(reqkind):

定義需求的種類，例如功能性需求、非功能性需求、介面需求、安全性、可靠度需求等。

(2) 需求編號(reqID):

給定需求編號。

(3) 目前需求版本(currentVer)

標示目前本項需求修改至第幾個版本

(4) 上一個需求版本(previousVer)

標示本次修改前的版本

(5) 需求名稱(reqName):

為這個需求定義一個名稱。

(6) 專案名稱(projectName)

標示需求所屬專案名稱

(7) 建立日期(estDate)

標示建立需求的日期。

(8) 修改日期(adptDate)

(9) 文件建立者(estUser):

標示需求文件建立人員。

(10) 文件修改者(specAdapt)

標示需求文件修改人員

(11) 系統使用者(targetUser)

描述系統是為哪些使用者群所開發。

(12) 需求延伸來源(deriveFrom):

標示本需求是由哪個需求所延伸而來，以便需求追蹤。

(13) 需求細分來源(refineBy):

標示本需求經由哪個需求細分而來，以便需求追蹤。

(14) 需求驗證來源(verifiedBy):

標示本需求是經由某個需求所產出的物件來驗證。

(15) 滿足需求項目(satisfyTo)

B:需求內容

(16) 需求描述(Requirement Describe):

自然語言仍然有其描述需求的便利性存在，使用者在這個欄位用自然語言來描述這項需求，如果需求內容的知識領域在我們所定義的樣版中有不足的部份，那麼需求描述欄位能讓

使用者能自行輸入自然語言的敘述，來補充說明不足的部分。

(17) 程式撰寫語言(codingLang):

標示本項功能或是系統將由何種語言撰寫。

(18) 傳輸速度(transRate):

描述這項需求所需要的傳輸速度。

(19) 客戶端數目(ClientNum):

描述本項需求將允許有多少個客戶端連線數目。

(19) cpu 數量(cpuNum):

針對系統可能會有多顆 cpu，所以在此加入這個選項。

(20) 處理器核心數量(cpuCore)

考量到目前單顆 CPU 多顆核心的普及程度，故加入需求樣版。

(21) 記憶體限制(memoryConstraint)

對嵌入式系統內建記憶體的限制

(22) 功率消耗(powConsume)(跟多核心結合 考慮到功率問題)

描述功能或系統所需消耗的功率。

(23) 系統排程方式(systemSched):

描述系統預定的排程方式。

(24) 時間限制(timeConstraint):

描述時間性限制，例如完成時間、啟動時間、反應時間。

(25) 需求是否為 atomic(isAtomic)

定義這項功能是否為 atomic。

(26) Pre-Condition (preCondition)

定義需求的前置條件。

(27) Post-Condition (postCondition)

定義需求的後置條件。

(28) 優先順序(Priority)

定義需求的重要性高低。

(29) COTS

有時因為人員或資源上的不足，亦有可能是工程師並非具有該專業領域知識，而需要採用商用現貨(COTS)的方式。

(30) Time to market (time2Mkt)

對產品上市時間的要求。

(31) System platform (sysPlatform)

系統將在何種平台上執行

上述的需求樣版定義出一系列的需求欄位，以擷取精確以及量化的需求為目標，來定義每個選項，以及進而對使用者在編輯需求的同時對系統需求做出完整的需求導引，並且找出軟體的主

要需求、功能性與非功能性需求，提供 CIM 以及 PIM 階段所需的基本需求資料，勾勒出系統的主要架構。

3.3 CIM 階段

在 CIM 階段分析師根據需求問題樣版內容針對每個需求分別繪製出 SysML 需求圖(圖 8)，每個需求類別(Requirement class)都擁有各自對應的樣版 Profile，如表 3 是功能性需求的樣版 Profile。

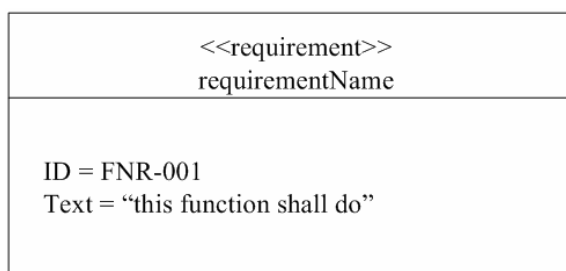


圖 8 需求圖描述功能需求

表 3 功能性需求樣版 Profile

Profile	Content
需求種類	Functional
需求名稱	Connection Server
需求編號	VSS-FNR-002
專案名稱	VMC_SYN
需求建立者	Joe

需求建立日期	2009.5.1
....	...

透過分析初步的需求問題樣版與初步的需求圖，了解初始需求之間的關聯性，將 SysML 需求圖的雛型依照需求關聯來描繪，再配合使用案例圖將有助於快速釐清系統範圍以及功能，促進需求的完整性，同時也便於估算開發成本與時程。

3.4 PIM 階段

在 PIM 階段我們根據分析狀態圖來有說明關於人員、流程、系統的運作，以及其狀態的轉移的規則；接著為狀態圖內部各狀態之間轉移的動作，定義需求的操作(operation)，以及為狀態的轉換行為定義屬性(Attribute)，所得到的資訊如表 4，將被參照使用於區塊定義圖來表達系統內部的靜態結構，區塊定義圖(Block Definition Diagram 以下簡稱 BDD)通常會是系統設計的時候，被拿來當作重要設計藍圖，分析師經過套用需求樣版，得到初步的系統靜態結構，也就是區塊定義圖，而透過分析前面的狀態圖與需求樣版來得知區塊定義圖尚需加入的屬性與操作種類，並可透過每個區塊所屬的 Profile(表 5 Connection Server BDD Profile 內容)來描述元件運作方式。

表 4 定義狀態轉移動作時的操作

系統名稱	
A.轉移動作	a.操作定義
B.轉換行為	b.屬性定義

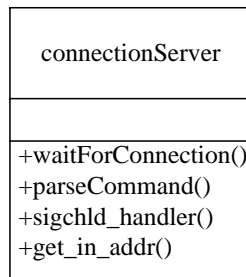


圖 9 VSS Connection Server block

表 5 Connection Server BDD Profile 內容

Profile	Content
區塊名稱	Connection Server
區塊編號	1
描述	提供連線給 RMC，並且接受 client 端的指令，經過指令分析之後提供相對應的服務。
相關性區塊	RecordedVideoListManager VideoStreamingManager
...	...

在分析狀態圖以及子系統內每個元件所屬的區塊，瞭解子系統內一群物件的互動情形之後，分析師將完整的區塊定義圖繪製出來，分析師在 CIM 階段取得系統環境以及初步的需求內容，經

過分析之後，於 PIM 階段將系統內部細節結構化並且產出 SysML 圖形，經過擷取系統的靜態以及動態需求的過程，就能得到繪制出完整的 SysML 需求圖所需要的需求資訊，而分析與整合這些需求資訊之後所得到的需求圖，是經由原始的使用者需求不斷的被延伸(derive)、細分(refine)、滿足(Satisfy)、驗證(verify)、追蹤(trace)之後，方能完成一個表現出需求之間關連性的 SysML 需求圖。

3.5 OCL 條件限制

我們利用 OCL 來規範使用者在輸入 Profile 時的敘述方式，這樣可以藉由驗證使用者所輸入的需求內容是否符合該欄位所制定的 OCL 條件，避免使用者輸入不當的數據或是需求文件內容，以下就是我們應用 OCL 做出條件限制的範例。

表 6 OCL 限制

樣版內容	資料型態	條件限制	OCL 定義
客戶端數目 (ClientNum)	Integer	客戶端數目不得 輸入小於 1 的值	self.ClientNum>0

第四章 案例研究

本章節將以 DVR 系統為範例，來說明以 SysML 為基礎之圖形塑模流程，使用 Papyrus 來進行編輯圖形系統需求，以及其輸出的結果。

4.1 實做環境

本篇論文所探討的以 SysML 為基礎所建構的圖形塑模流程，使用微軟的 Windows XP 為作業系統平台，並採用 Papyrus 做為圖形編輯工具以及開發 Profile 的工具，執行平台為 IBM 公司所釋出的 Eclipse 整合開發環境，使用 Sun Java 做為程式碼需要細部修正時的輔助工具。

使用 Java 的原因是因為其具有物件導向的特性，並且建立了相當龐大且齊全的程式庫，具有相當好的程式支援性以及能夠跨平台的高度移植性，所以我們採用 Java 來輔助開發撰寫，而 Eclipse 則是具備開發 Java 程式所需要的整合開發環境，並且能結合 GUI(Graphic User Interface) 外掛程式來做系統畫面建構，Papyrus UML 塑模工具是以 Eclipse 為核心的 Open source 系統塑模工具，並提供其本身使用 Java 所開發的可編譯原始碼，當必須

要修改原本的系統以配合實際應用時，有利於程式碼的改寫以及補強，因此我們採用 Papyrus 來做為應用實作案例研究的工具。

4.2 系統畫面

圖 10 為我們所使用之 open source 需求塑模工具的主要畫面，主要分為樹狀目錄、編輯視窗以及調色盤三個部份，使用者建立一個專案之後就能夠開始編輯需求圖，需求塑模編輯器主要提供圖形化介面與功能，讓使用者將訪談後之初步需求以格式化、圖形化的方式來描述需求。

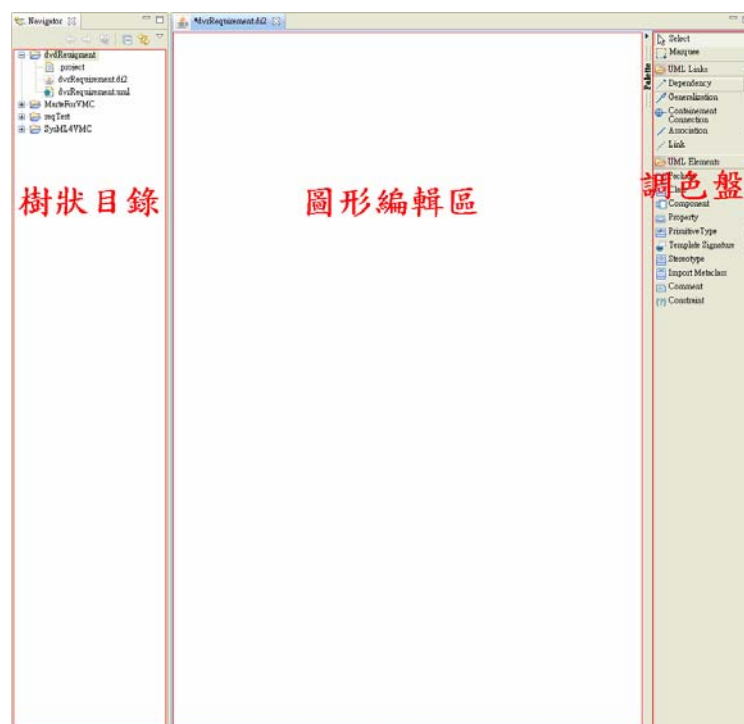


圖 10 系統初始畫面

4.3 塑模與 Profile

從工具中建立專案之後，將所得到的原始需求透過操作圖形化介面塑模系統來編輯，並選擇適當的樣版 Profile 需求範本來輔助系統需求撰寫，並對每項需求建立初始的需求圖。



圖 11 建立初始的需求圖

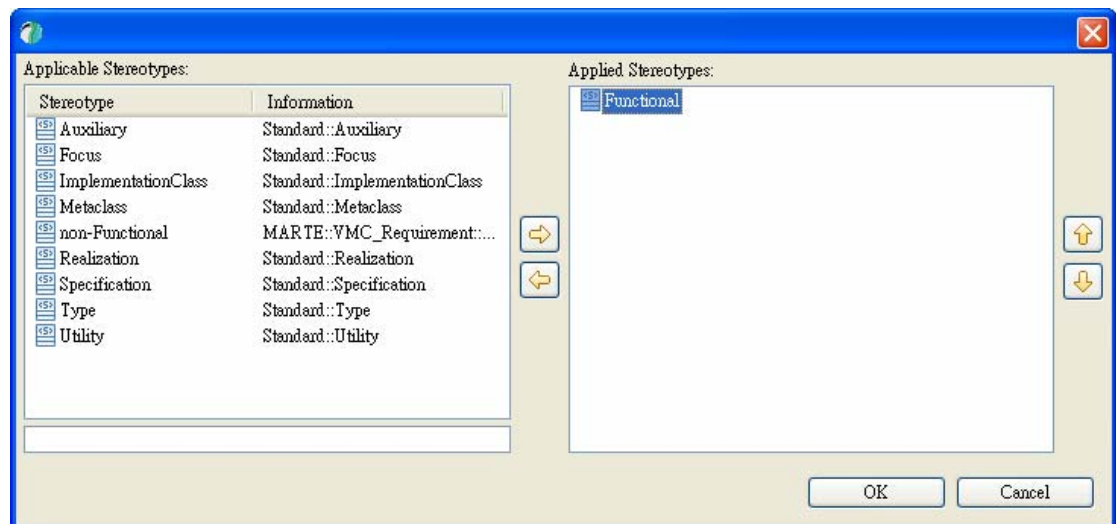


圖 12 Select profile

將塑模圖形套用需求樣版 Profile 的好處是，使用者在圖形塑模的同時，能夠依循我們所制定的需求樣版來建立需求，盡可能避免前述傳統需求建立方式常見需求不精確、不完整的錯誤，編輯出較完整的需求文件，我們利用 Stereotype 擴充機制的方式來實作需求樣版(圖 13 Requirement template Profile)，圖 14 是我們所定義的部分需求量化單位，透過量化單位，可以引導使用者更加精確地描述需求。

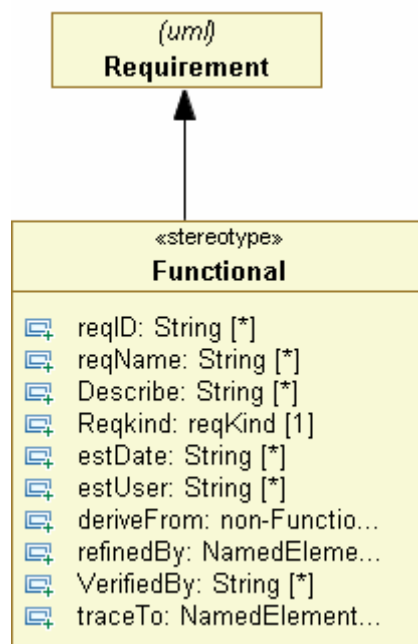


圖 13 Requirement template Profile

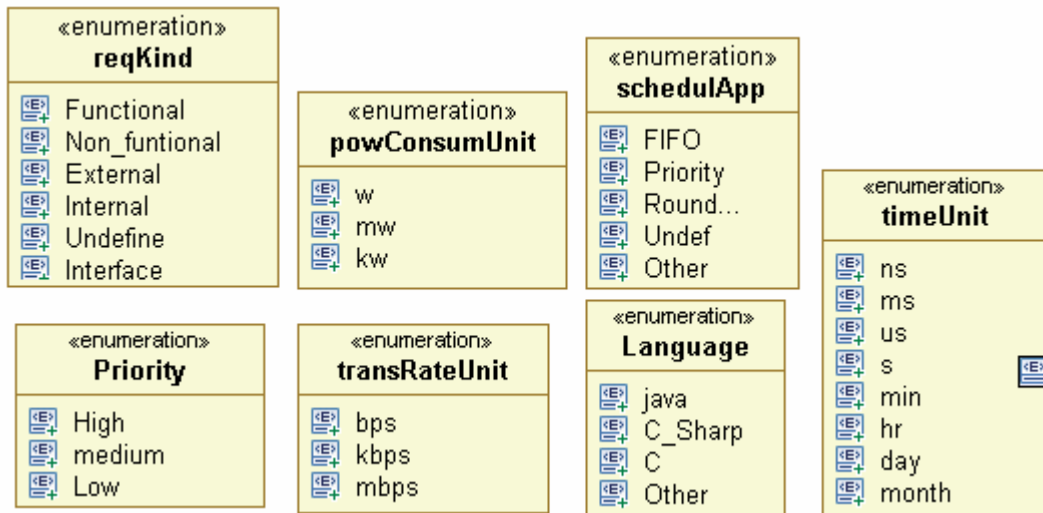


圖 14 需求量化單位

套用 Profile 之後，使用者在編輯 SysML 圖形的同時，依照我們定義的需求樣版來引導使用者填寫需求，將需求描述輸入需求樣版 Profile(圖 15)。

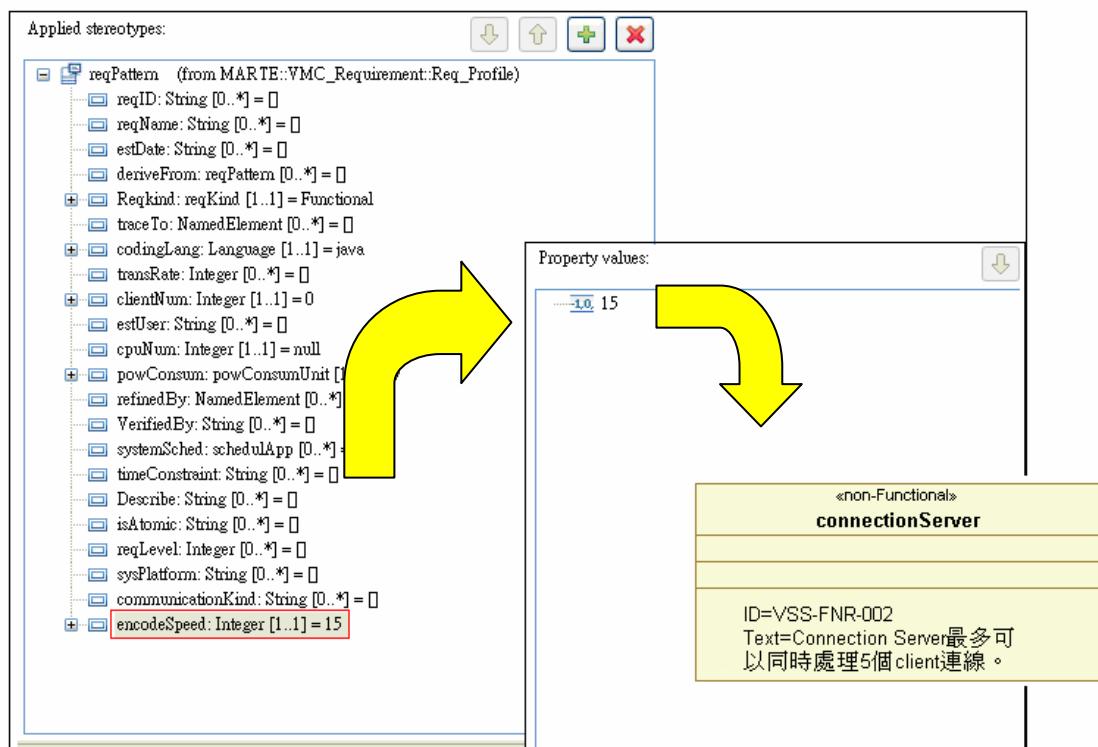


圖 15 輸入需求樣版

如果在編輯 Profile 的過程中，使用者不確定所輸入需求內容的值域是否超出需求樣版的規範，那麼使用者可以透過 OCL 驗證的功能來檢視內容是否有誤，若經過 OCL 驗證發現確實輸入錯誤，系統將會指出內容錯誤的欄位，來引導使用者做修正 Profile 內容的動作，如圖 16 所示，使用者在輸入 CPU 數量時，不慎誤植數據為-1，而在我們所規範的限制中，使用者若輸入了小於零的數值，在 OCL 驗證時，便會檢驗出錯誤的欄位(圖 17)，使用 OCL 的好處是我們能規範使用者輸入需求樣版的內容，如此就能盡可能避免需求的錯誤、不精確或是模糊不清。

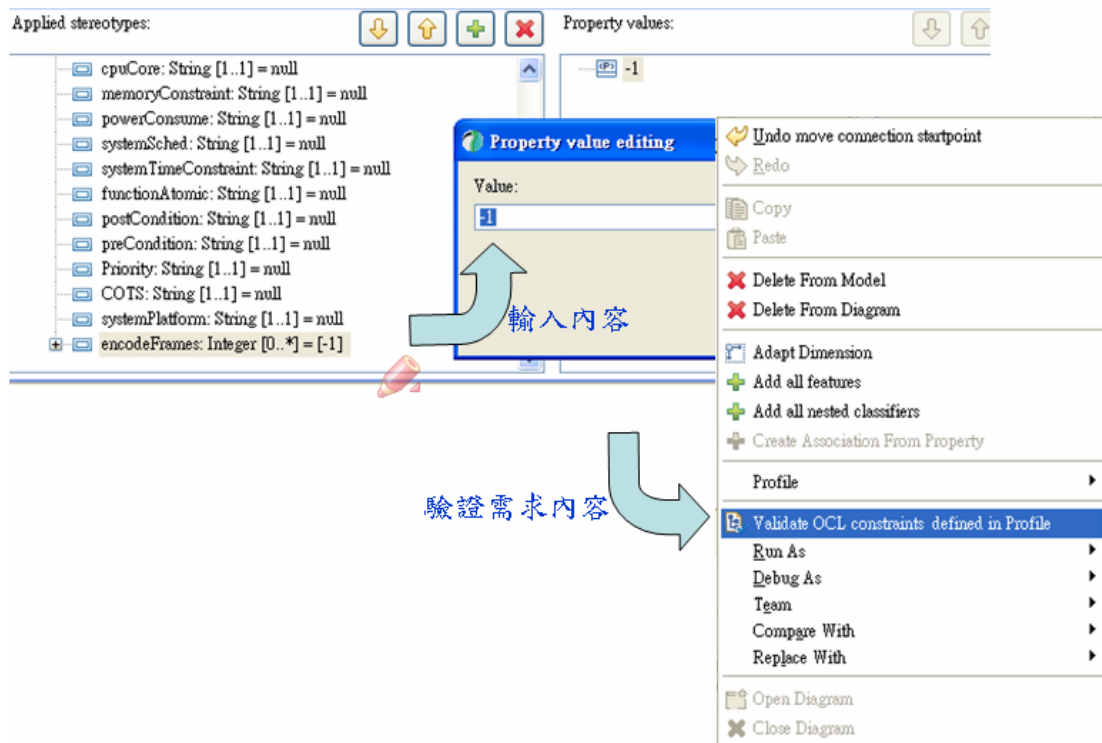


圖 16 檢驗 Profile 內容錯誤

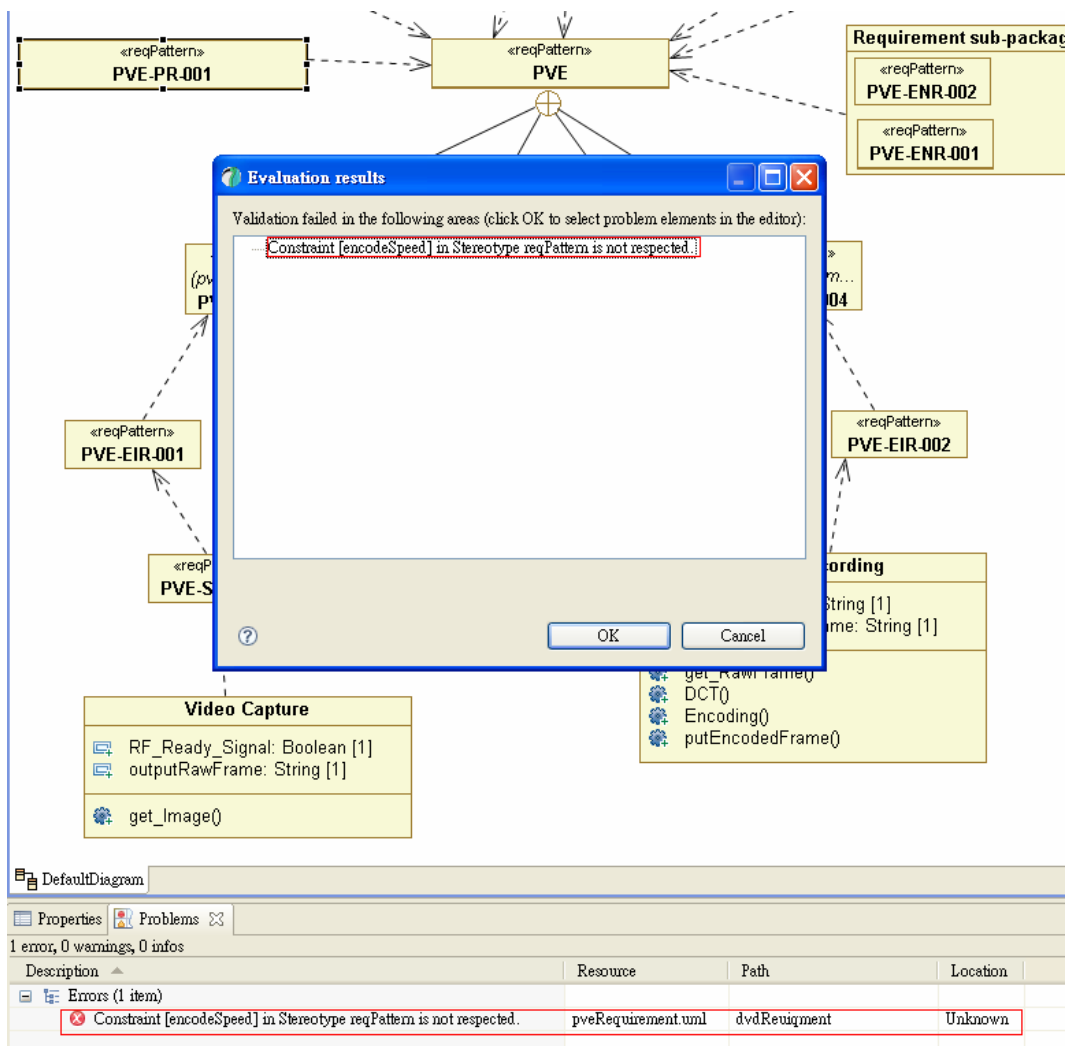


圖 17 驗證需求內容發現使用者輸入錯誤資料型態

4.4 CIM 階段需求分析

使用者在需求引導時依照需求問題樣版來撰寫客戶需求，經過初步分析問題樣版所得到的需求資訊之後，分析師將重要的功能歸納分類，並繪製使用案例以及雛型需求圖。

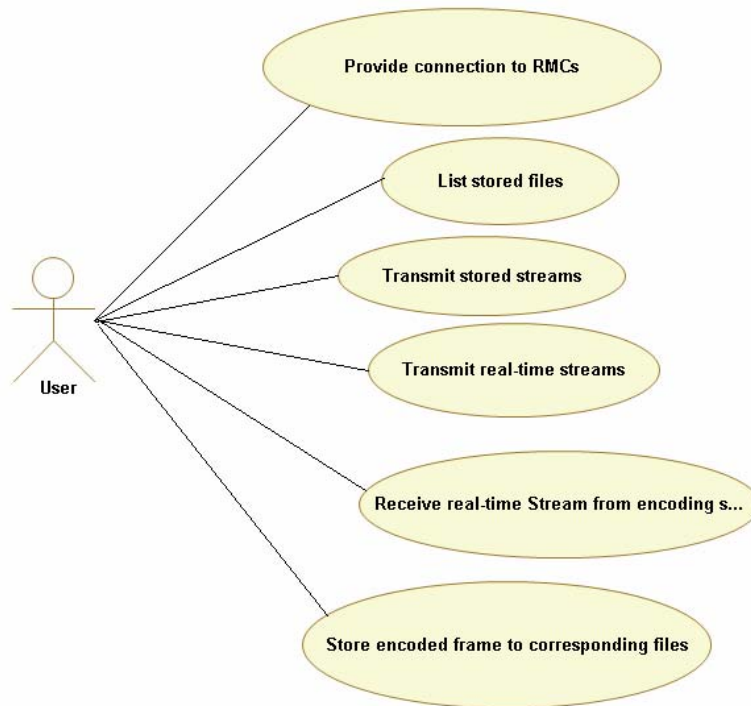


圖 18 Use Case

圖 18 為分析師所繪製的使用案例圖，定義出 User(Actor)將會擁有哪些基本功能。

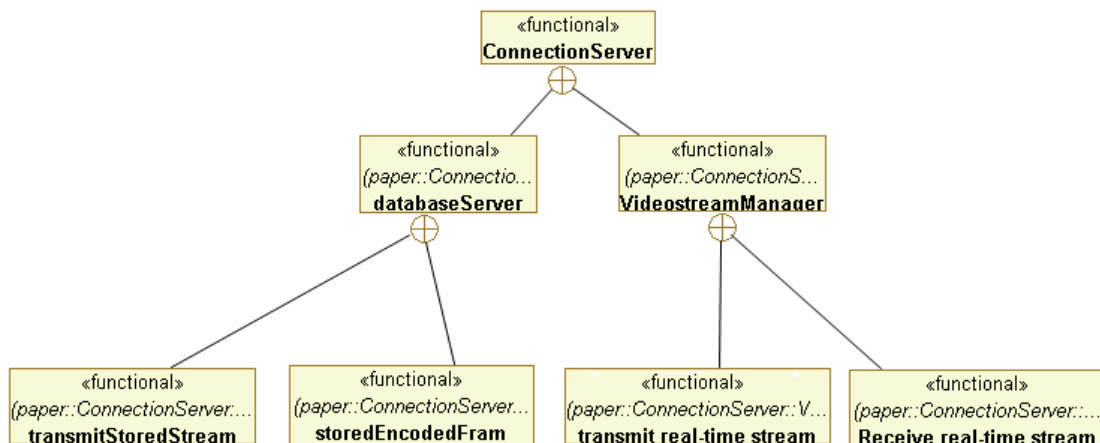


圖 19 VSS system requirement diagram prototype

圖 19 為將系統需求與使用案例做初步的歸納分析所產生的需求雛型，描繪出每個系統元件的原始需求之關連性。

4.5 PIM 階段

4.5.1 分析動態規則

分析師在經過分析使用案例之後，將 Connection Server 歸納為重要的需求項目，所以分析 Connection Server 的狀態轉移並繪製成狀態圖，以便了解與組織需求之間狀態的轉換。

1. 系統提出要求連線，Connection Server 啟動，Server 初始化。

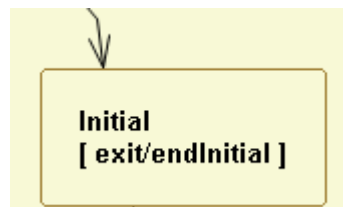


圖 20 要求啟動 Connection Server

2. 起始後無任何動作，進入閒置狀態。

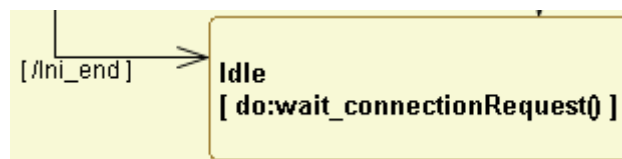


圖 21 進入閒置狀態

3. 接收 RMC 請求/傳送相對應要求服務。

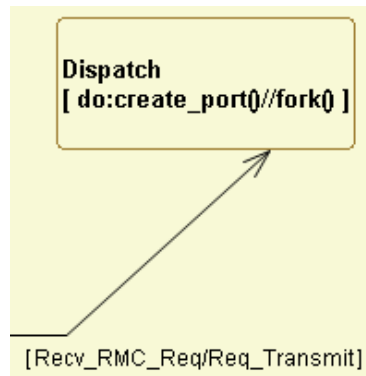


圖 22 接收與傳輸

4. 請求事項完成，轉移至系統閒置狀態。

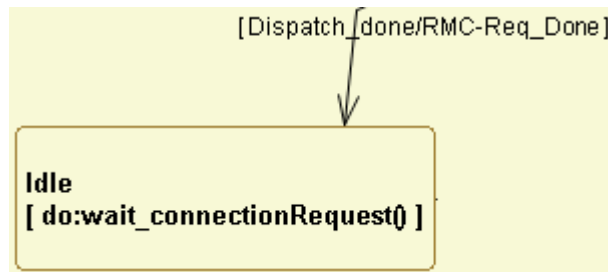


圖 23 回到閒置狀態

分析過狀態的轉移之後，接下來，將上圖四個片段組織，得到 VSS 系統元件之一的 Connection Server 狀態圖。

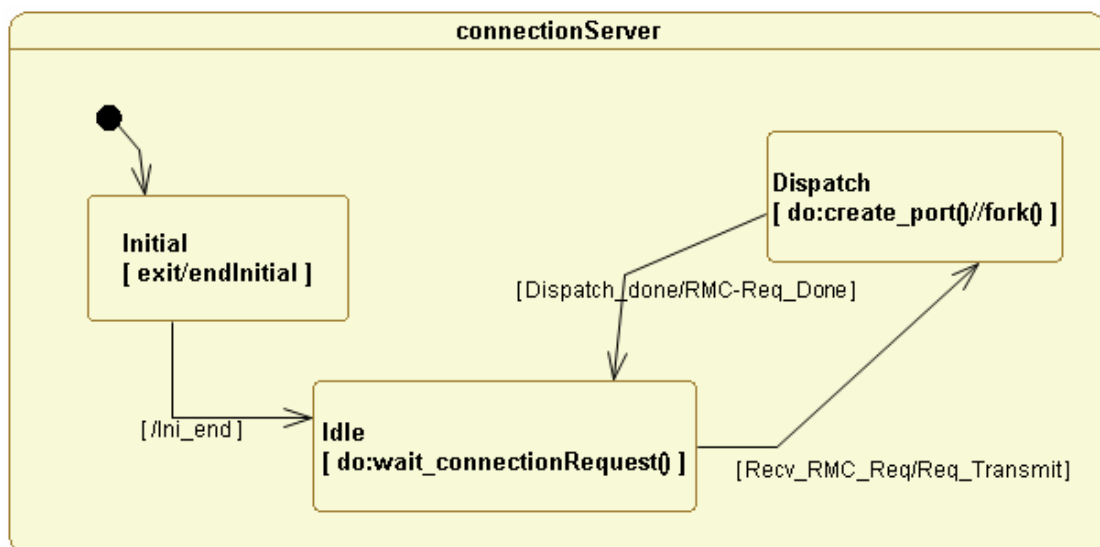


圖 24 Connection Server State Diagram

4.5.2 描述靜態結構

透過分析狀態圖，根據我們在第三章裡所提到，系統分析師可以針對狀態轉移的動作，定義需求的操作(operation)性質，為狀態的轉換行為定義屬性(Attribute)，表 7 即為系統元件定義相關操作的說明表，透過狀態圖以及需求引導階段所得到的表格內容來分析每個子系統區塊之間的組合關係，並且利用說明表將有關聯的操作以及屬性定義之後，進而產出 VSS 子系統的區塊定義圖，如圖 25，Video Stream Server 延伸出五個元件區塊，並且建立起每個子區塊的關聯性。

表 7 定義狀態轉移動作時的操作

Connection Server	
動作	定義操作
等待連線 接收/回應請求 進入/回到閒置	+waitForConnection() +parseCommand() +Sigchild_handler() +get_in_addr()

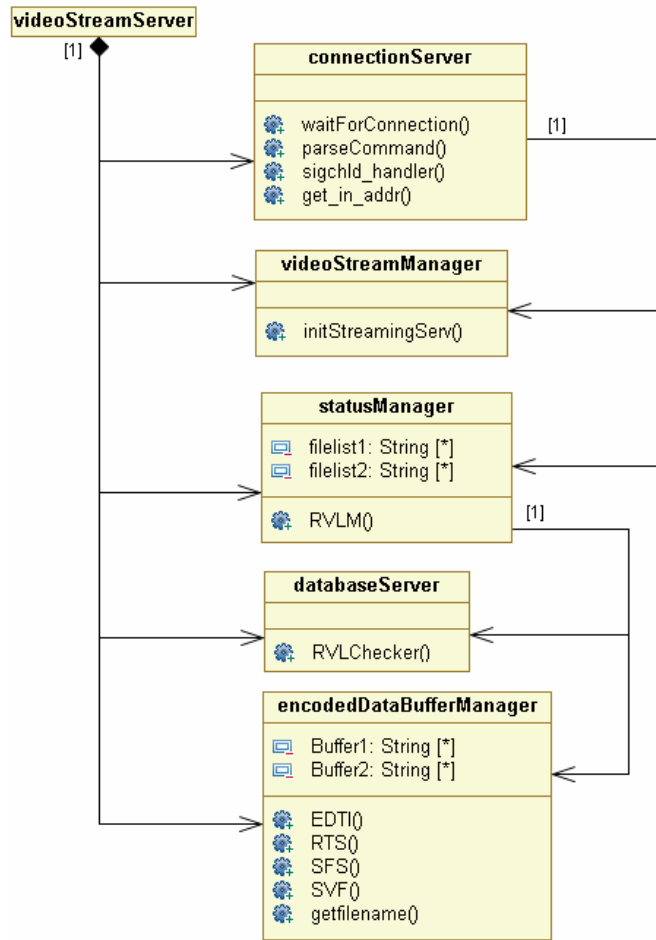


圖 25 Papyrus 產出 VSS 區塊定義圖

在分析狀態圖與區塊定義圖所提供的資訊後，系統分析師始能完成一個完整的 SysML 需求圖，我們以 DVR 子系統 Parallel Video Encoder (PVE)來討論，如下一頁的圖 26 PVE 需求圖所示，需求類別之間的關連性透過 SysML 需求圖的方式來表現，⊕表示較低階層的需求對上一層的需求有聚合關係，<<derive>>表示這項需求是由 PVE 這個需求衍生而來，具有衍生關係，一個命名為 PVE 的集合(set)，裡面元素是四個功能性需求的子集合(subset)，包含了四個功能性需求，PVE 對四個子集合具有組成

(composition)關係，而底層的區塊定義圖(Video Capture、Video Encoding)則是表示這兩個區塊是由分析師根據需求所衍生出的系統功能元件，此功能元件區塊具有滿足(satisfy)所指向需求的關連性，例如：影像捕捉(Video Capture)元件區塊是由 PVE-SYR 需求所衍生而來，PVE-SYR 的需求資訊為 PVE 子系統需能夠擷取影像，而影像捕捉區塊則是分析師為了滿足這項需求而產生具備有影像擷取功能以及特性的子系統元件，Requirement sub-package 的功用是將類似的需求歸類在同一個 package 內，以降低需求圖的複雜度，附錄一為 PVE 子系統詳細之需求圖。

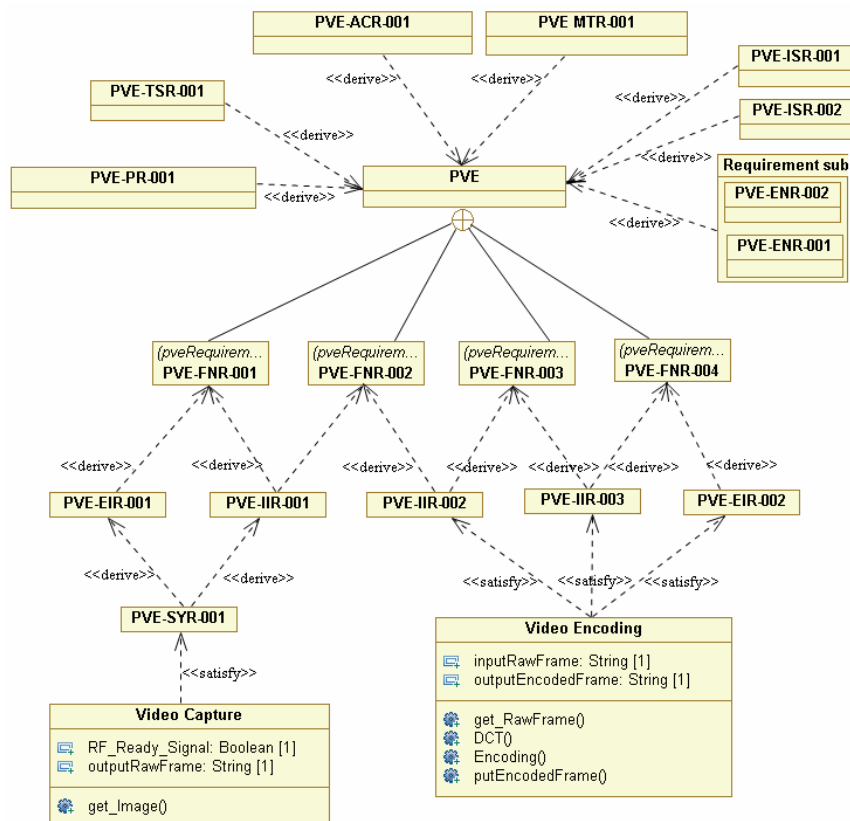


圖 26 PVE subsystem Requirement Diagram

4.6 匯入與匯出 XML 需求文件

經過模型驅動架構流程一連串的分析之後，再利用 XML 解析器(Parser)將塑模完成的 SysML 模型匯出成 XML 文件，使用者輸入的需求樣版內容，將完整的對應到所輸出的 XML 檔案，形成一份結構化的文件，提供專案後續分析設計流程使用，後續專案開發人員，只需使用自訂的解析器，利用解析辨識 XML 標籤(Tag)來擷取文件中內含的需求資訊，篩選合適的需求文件內容來做進一步的分析與設計，分析師也可以將 XML 文件留存，如果後續有類似的系統開發需求，就可以用匯入 XML 文件的方式來套用或是經過修改後使用。



圖 27 匯出 XML 需求文件

第五章 結論與未來展望

在這篇研究裡我們展現了一個基於 SysML 的圖形塑模流程，配合模型驅動方法以及定義需求樣版，讓使用者從編輯 SysML 圖形完成的流程中，將系統需求規格書中所需撰寫的部份基本內容完成，分析師使用 SysML 進行需求塑模以避免需求前後不一致，而 MDA 則提供分析師一個分析需求的流程，來降低因為錯誤的需求擷取而導致專案延宕的風險，再經由需求樣版來輔助使用者確認以及提高需求內容的正確性，縮短需求規格的制定與撰寫時間，增加需求可信度和可行性；目前工具的資源整合方面，Template Profile 的編輯對使用者並不友善，未來我們將對 Profile 的使用者介面做修正，讓使用者能更容易對 profile 進行修改需求樣版內容以及新增需求樣版，並藉由開發相關使用工具來輔助研究與實際應用，讓使用者在系統需求塑模的流程中擁有更便利的操作使用環境；在理論研究方面將持續對 Reusable database 做進一步的探討，從需求資料中截取出可用的再利用元件，增加元件再利用的效率，並持續加入多核心系統需求的特性，我們也將更進一步研究 SysML 需求圖、使用案例圖與其他 SysML 圖形之間的關聯性以及追蹤方法，讓需求文件撰寫流程更加完善。

參考文獻

- [1] Finkelstein A., Gabbay D., Hunter A., and Nuseibeh B., “Inconsistency Handling in Multi-Perspective Specifications,” *IEEE Transactions on Software Engineering*, Vol. 20, No.8, 1994, pp. 569-578
- [2] Hooks I., “Writing Good Requirements,” *the Proceedings of the 3rd NCOSE International Symposium*, 2003.
- [3] Kasser J.E., “Toward Improving the Recognition and Correction of Poor Requirements,” *the Proceedings of SETE 2005, ICE Australia, 2005*, pp.1-13.
- [4] Clarke S., Murphy J., and Roantree M., “Composition of UML Design Models: A Tool to Support the Resolution of Conflicts,” *Proceedings of the 5th International Conference on Object-Oriented Information Systems*, 1998, pp.464-479.
- [5] Conejero J.M., Hernandez J., Moreira A., Araujo, J., Mayumi I.K., Tetsuo T., “How Does Requirements Quality Relate to Project Success or Failure?”, *Requirements Engineering Conference, RE '07. 15th IEEE International*, Oct 15-19, 2007, pp.69-78. How dose requirements Quality relate to project success or failure? IEEE 2007.
- [6] Sommerville I., *Software Engineering*, Sixth Edition, Addison-Wesley Publishing Co. Inc.: Wokingham, England, 2001.
- [7] OMG Document, *A UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems*, June 09, 2008.
- [8] Standish, “The Chaos Report”, <http://www.standishgroup.com/chaos.html>, March. 19. 1998.
- [9] Kaindl H., “A Practical Approach to Combining Requirements Definition and Object-Oriented Analysis,” *Annals of Software Engineering*, Vol. 3, 1997, pp.

319-343.

- [10] Object Management Group (Online information: retrieved March 29, 2009)<http://www.omg.org/>
- [11] UML(Unified Modeling Language) (Online information: retrieved March 5, 2009) http://www.omg.org/gettingstarted/what_is_uml.htm
- [12] MDA Model Driven Architecture(Online information: retrieved March 5, 2009)
<http://www.omg.org/mda/>
- [13] SysML Specification(Online information: retrieved April 5, 2009)
<http://www.omg.sysml.org/#Specification>
- [14] OMG Document, *SysML Specification*, May 04, 2006.
- [15] OMG Document, *UML for Enterprise Application Integration*. Sept 17, 2001.
- [16] OMG Document, *UML Profile for Enterprise Distributed Object Computing*.
June 9, 2001.
- [17] Sinan Si A., *UML in a Nutshell*. O'Reilly & Associates Inc. Taiwan, 1998.
- [18] Papyrus, (Online information: retrieved March 5, 2009)
<http://www.Papyrusuml.org/>
- [19] Erik T.,R., *Learning XML*. O'Reilly & Associates Inc. Taiwan, 2001.
- [20] OMG adopted specification. "*UML 2.0 OCL Specification*". Oct 14, 2003.

附錄一 PVE 子系統 SysML 需求圖

