國　立　交　通　大　學
工業工程與管理學系

碩士論文

可加註語言樹轉換法的分析：從記號階梯圖到記號圖

Analysis of XML Tree Transformation Method: from Marked Ladder Diagrams to Marked Graphs

研究生：魏良榮
指導教授：梁高榮博士

中華民國九十九年七月

可加註語言樹轉換法的分析：從記號階梯圖到記號圖

Analysis of XML Tree Transformation Method: from Marked Ladder Diagrams to Marked Graphs

Student：Liang-Ying Wei

Advisor：Gau-Rong Liang

A Thesis Submitted to
Department of Industrial Engineering and Management
College of Management
National Chiao Tung University
In Partial Fulfillment of the Requirements
For the Degree of Master of Engineering in
Industrial Engineering and Management
July 2010
Hsinchu, Taiwan, Republic of China

可加註語言樹轉換法的分析：從記號階梯圖到記號圖

研究生：魏良榮　　指導教授：梁高榮　博士

國立交通大學工業工程與管理學系

摘要

裴氏圖是一套可以改善現場監控系統績效的卓越數學模型。相對地，較低成本的階梯圖卻成為最普遍的現場監控器。若可以將階梯圖自動轉換為裴氏圖，我們就可以容易地為現場監控器建立其數學模型。在本論文裡，用可加註語言樹為基礎的裴氏圖可加註語言文件與 PLCopenXML 文件來分別代表裴氏圖與階梯圖。接著發展出可加註語言樹轉換法將 PLCopenXML 文件轉換為裴氏圖可加註語言文件。由於轉換上的計算複雜度，本論文對階梯圖與裴氏圖加以限制。更精準地說，可加註語言樹轉換法只針對記號階梯圖與記號圖來實作。一些簡化的實作案例則驗證了本研究方向的可行性。

關鍵字：
記號圖(Marked Graph)
記號階梯圖(Marked Ladder Diagram)
裴氏圖加註語言(Petri Net Markup Language)
PLCopen 可加註語言(PLCopenXML)
可加註語言樹轉換法(XML Tree Transformation Method)

Analysis of XML Tree Transformation Method:
from Marked Ladder Diagrams to Marked Graphs

Student：Liang-Ying Wei　　Advisor：Dr. Gau-Rong Liang

Department of Industrial Engineering and Management
National Chiao-Tung University

Abstract

Petri Net is an excellent mathematical model for improving the performance of shop floor control system. In contrast, the most used controller on shop floor is ladder diagram because of its lower cost. If it is possible to transform ladder diagrams into Petri Nets automatically, then we can build their mathematical models for shop floor controllers easily. In this thesis, XML trees named PNML document and PLCopenXML document are used to represent Petri Nets and ladder diagrams, respectively. Then an XML tree transformation method has been developed for transforming a PLCopenXML document into a PNML document. Because of the computational complexity of the transformation, some limitations are applied to the Petri Nets and ladder diagrams. More precisely, only the XML tree transformation method for marked ladder diagrams and marked graphs has been implemented. The simplified implementation for some case study shows the feasibility of this research direction.

Key words：Marked Graph, Marked Ladder Diagram, Petri Net Markup Language, PLCopenXML, XML Tree Transformation Method

致謝

兩年的研究所求學生涯，到最後這份論文的誕生，首先要感謝梁高榮教授的栽培。老師於學術研究與論文撰寫方面，孜孜不倦的指導並且提供豐富的資源，讓我學習到細心研究、謹慎思考的做事態度。並且從老師對工作與生活心態方面，學習到一個”Master”應該具備的做人處事的態度及道理。

這份論文得以順利完成，也要感謝唐麗英老師、陳文智老師以及張永佳老師，不吝提供寶貴的意見，並且給予指導，使我的論文臻於完善。感謝唐老師與張老師在我的研究架構與論文寫作方面，提供精闢的見解；感謝陳老師對於我的研究內容，有細心的建議。

在我的求學過程中，非常感謝實驗室中大家的陪伴。謝謝教練、便便、顧胖、花花、小鄭的互相砥礪；謝謝學長猴猴的經驗傳授；謝謝學弟妹們老恩、大頭、惠珍、思思、朱明典的支持鼓勵。你們豐富了我的研究所生涯，讓我過得充實、快樂。

另外，謝謝我的朋友們，在這兩年中給予我溫暖的陪伴。謝謝扣斯、PPTRA、咪咪、添盛、紹楷，你們的存在給予我很大的支持力量。

最後，我要感謝我的父母，給我非常大的自由與空間，讓我可以朝自己想走的方向邁進。也謝謝你們給我完全的信任與支持，讓我對於自己的人生懂得規劃與負責。我所有的成就都應該屬於你們。

目錄

摘要 i
致謝 iii
圖目錄 vi
第一章 緒論 1
1.1 研究動機 1
1.2 問題界定及研究目的 3
1.3 研究方法 5
1.4 論文架構 6
第二章 文獻回顧 7
2.1 階梯圖 7
2.1.1 IEC61131 標準 7
2.1.2 階梯圖基本元件 7
2.1.3 階梯圖基本性質 8
2.1.4 階梯圖之邏輯迴路 9
2.2 裴氏圖 10
2.2.1 裴氏圖基本元件 10
2.2.2 裴氏圖基本性質 10
2.2.3 裴氏圖之數學特性及規則矩陣 11
2.2.4 記號圖 12
2.3 裴氏圖與階梯圖之轉換 13
2.4 可加註語言 14
2.4.1 可加註語言技術及發展 14
2.4.2 PLCopen 加註語言 15
2.4.2.1 AutomationML 及 PLCopen 標準規範 15
2.4.2.2 PLCopen 加註語言架構 15
2.4.3 裴氏圖加註語言架構 17
2.5 可加註語言處理技術 19
2.5.1 SAX 和 StAX 19
2.5.2 文件物件模型 19
2.5.3 XML 資料繫結 20
2.5.4 XML 文件處理技術選擇 20
2.5.5 PLCopenXML 及裴氏圖加註語言之文件物件模型 21
第三章 可加註語言樹轉換法的設計 25
3.1 記號階梯圖與記號圖之映射分析 25
3.1.1 狀態元件與條件元件 25
3.1.2 標記條件與不標記條件 26
3.1.3 自我控制迴路 26
3.2 記號階梯圖至記號圖的修正梯級與記號階梯圖的定義 27

3.2.1 原始映射梯級 27
3.2.2 修正映射梯級 27
3.3 記號階梯圖與記號圖的等構關係 31
3.4 可加註語言樹轉換法的分析 33
3.4.1 可加註語言樹轉換法的架構 33
3.4.2 可加註語言樹轉換法的三階段流程與說明 34
3.5 偵查方程式的轉換 40
第四章 可加註語言樹轉換法程式的實作 42
4.1 可加註語言樹轉換法程式的需求分析 43
4.2 可加註語言樹轉換法程式的設計實作 45
4.2.1 程式操作流程 45
4.2.2 程式套件設計 45
4.2.2.1 程式套件關聯與程式處理流程 46
4.2.2.2 GUI 套件設計 47
4.2.2.3 xmltools 套件設計 49
4.2.2.4 ladderdiagram 套件與 petrinet 套件設計 50
4.2.2.5 transformtools 套件設計 53
4.2.2.6 fileGenerator 套件設計 55
第五章 案例分析 57
5.1 液體加熱系統案例分析 57
5.1.1 液體加熱系統案例分析的操作流程 57
5.1.2 建立液體加熱系統的記號階梯圖 58
5.1.3 液體加熱系統記號階梯圖至記號圖的轉換 60
5.1.4 液體加熱系統記號圖的驗證 60
5.2 灌模系統案例分析 62
5.2.1 灌模系統案例分析的操作流程 62
5.2.2 建立灌模系統的記號階梯圖 62
5.2.3 灌模系統記號階梯圖至記號圖的轉換 65
5.2.4 灌模系統記號圖的驗證 65
第六章 結論 67
6.1 結論 67
6.2 後續研究 68
參考文獻 69

圖目錄

圖 1.1 裴氏圖與記號圖的關係 3
圖 1.2 階梯圖與裴氏圖之等構關係 3
圖 1.3 現場監控流程 4
圖 1.4 本論文研究方法流程 5
圖 1.5 本論文研究方法與論文架構對照 6
圖 2.1 階梯圖的基本圖形 8
圖 2.2 OR/AND 迴路 9
圖 2.3 裴氏圖 11
圖 2.4 裴氏圖之狀態方程式 11
圖 2.5 裴氏圖之規則矩陣 12
圖 2.6 裴氏圖分類 12
圖 2.7 裴氏圖與對應之階梯圖 13
圖 2.8 XML 與網路之互動[5] 14
圖 2.9 XML Tree 結構的辨別 15
圖 2.10 簡單的階梯圖 16
圖 2.11 PLCopenXML 的 XML Tree 17
圖 2.12 簡單的裴氏圖 17
圖 2.13 PNML 的 XML Tree[1] 18
圖 2.14 JAXB 的資料處理流程[19] 20
圖 2.15 PLCopenXML 的 DOM 模型 21
圖 2.16 PNML 的 DOM 模型[1] 23
圖 3.1 記號圖之條件元件及記號階梯圖之條件元件 25
圖 3.2 記號階梯圖的自我控制迴路 26
圖 3.3 自我控制迴路對照的記號圖及轉換的布林表示法 27
圖 3.4 為圖 3.3 的原始映射梯級[7] 27
圖 3.5 記號圖 28
圖 3.6 為圖 3.5 映射的記號階梯圖 28
圖 3.7 利用記號圖解釋記號階梯圖的非同步特性 28
圖 3.8 交換暫存點 P_i 及暫存點 P_{i+1} 發生順序 29
圖 3.9 利用修正映射梯級轉換圖 3.5 後之記號階梯圖 29
圖 3.10 等構的記號階梯圖與記號圖 29
圖 3.11 記號圖及其規則矩陣 31
圖 3.12 P1 依照規則矩陣寫出的邏輯式 31
圖 3.13 記號階梯圖及其布林邏輯式 31
圖 3.14 P1 依照布林邏輯式寫出的邏輯式 32
圖 3.15 記號階梯圖與記號圖的等構關係 32
圖 3.16 轉換法示意圖 33

圖 3.17　可加註語言樹轉換法的設計流程 34
圖 3.18　可加註語言樹轉換法三階段與程式套件的對應關係 35
圖 3.19　階段 II 的物件對應關係 37
圖 3.20　裴氏圖 40
圖 3.21　利用表 3.7 中邏輯式繪出之階梯圖 41
圖 4.1　可加註語言樹轉換法程式開發程序 42
圖 4.2　使用者需求與程式需求 43
圖 4.3　程式規格 44
圖 4.4　程式操作流程 45
圖 4.5　程式處理流程 46
圖 4.6　程式套件關聯 47
圖 4.7　GUI 套件架構 48
圖 4.8　GUI 套件的 UML 類別圖 48
圖 4.9　xmltools 套件架構 49
圖 4.10　OPLCXML 類別之流程圖 49
圖 4.11　xmltools 套件類別圖 50
圖 4.12　ladderdiagram 套件架構 50
圖 4.13　ladderdiagram 套件處理流程 51
圖 4.14　ladderdiagram 套件類別圖 51
圖 4.15　petrinet 套件架構 52
圖 4.16　petrinet 套件處理流程 52
圖 4.17　petrinet 套件類別圖 53
圖 4.18　transformtools 套件架構 53
圖 4.19　LD2PN 類別處理流程 54
圖 4.20　transformtools 套件類別圖 55
圖 4.21　fileGenerator 套件架構 55
圖 4.22　fileGenerator 套件類別圖 56
圖 4.23　outputPNML 類別及 outputPLCopenXML 類別處理流程 56
圖 5.1　液體加熱系統案例分析流程圖 57
圖 5.2　液體加熱系統的記號階梯圖 59
圖 5.3　轉換程式將液體加熱系統的記號階梯圖轉換為記號圖 60
圖 5.4　原始液體加熱系統的記號圖 61
圖 5.5　轉換程式轉換出的液體加熱系統記號圖 61
圖 5.6　灌模系統案例分析的流程圖 62
圖 5.7　灌模系統的記號階梯圖 64
圖 5.8　轉換程式將灌模系統的記號階梯圖轉換為記號圖 65
圖 5.9　原始灌模系統的記號圖 66
圖 5.10　轉換程式轉換出的灌模系統記號圖 66
圖 6.1　階梯圖形態 67

表目錄

表 2.1 階梯圖組成元件圖形對照及代表意義 8
表 2.2 裴氏圖各元件之圖形對照及代表意義 10
表 2.3 裴氏圖與階梯圖的邏輯式架構分類 13
表 2.4 PLCopen 加註語言文件 16
表 2.5 裴氏圖加註語言文件 18
表 2.6 不同狀況下，使用不同 XML 處理技術的最佳選擇 21
表 2.7 PLCopen 文件檔物件說明 22
表 2.8 PNML 文件檔物件說明 23
表 2.9 PNML 文件檔物件說明(續) 24
表 3.1 記號階梯圖的物件群 36
表 3.2 記號圖的物件群 37
表 3.3 階段 II 記號階梯圖物件轉換為記號圖物件的詳細對應關係 38
表 3.4 轉換程式中的方法名稱及意義 38
表 3.5 轉換程式中的方法名稱及意義(續) 39
表 3.6 圖 3.19 之偵查法則 40
表 3.7 表 3.6 之對應條件式與邏輯式 41
表 4.1 程式套件名稱及功能 46
表 5.1 液體加熱系統記號階梯圖的輸入接點及輸出線圈名稱對照之意義 58
表 5.2 灌模系統記號階梯圖的輸入接點及輸出線圈名稱對照之意義 63

第一章　緒論

本章主要在說明本論文研究方向及整體架構，本章節共分為四小節。第 1.1 節『研究動機』說明本論文的研究動機，第 1.2 節『問題界定及研究目的』提出本論文的研究假設並說明研究目的及界定研究範圍，第 1.3 節『研究方法』為本論文的研究方法及階段說明，第 1.4 節『論文架構』為論文的整體架構。

1.1　研究動機

多年來，工業界的自動控制領域，普遍使用可程式邏輯控制器(Programmable Logic Controller, PLC)[13]做為控制製造系統的工具。在可程式邏輯控制器出現之前，自動控制系統是利用成千上百的繼電器(Relay Coil)與計數器組成，但可由程式語言控制的可程式邏輯控制器，基本上可以完全取代這些大型裝置。

PLC 的程式語言[13]很多，其中以階梯圖(Ladder Diagram, LD)[11]為工業界自動製造系統中最普遍使用的程式語言。階梯圖利用電路的串並聯邏輯構成，在學習及應用上都較簡易。但階梯圖的設計規模與設計者的經驗相關；若使用者設計經驗較為豐富，繪出之階梯圖會較為精簡。在驗證方面，階梯圖只能透過實驗與模擬確認邏輯是否正確。當製造系統與控制程式的複雜度增加，驗證步驟也會更加耗時且增加成本。

由於階梯圖為較低階的程式語言，執行階梯圖時，常無法得知驅使系統流程正常運作的執行順序、同步性以及擁有一致性的事件，因此很難對已編程好的階梯圖程式語言做修改且再利用[38]。因此階梯圖只能做到"控制"，但無法做到系統分析、評估及模擬。而裴氏圖(Petri Net, PN)[36]之設計，可以畫出製造系統的雛形，並且利用裴氏圖之數學性質分析、模擬及控制自動製造系統[3]。

針對裴氏圖轉換為階梯圖，已有研究提出使用映射分析進行裴氏圖及階梯圖的轉換[5, 28]，但卻缺乏電腦程式的輔助、實作。目前的轉換僅能利用裴氏圖的繪圖程式畫出裴氏圖後，以人工比對的方式，對照其邏輯再繪出階梯圖。當裴氏圖過於複雜時，繪製階梯圖者誤判邏輯的機率也會增加，如此繪出的階梯圖也等同於無效。且利用圖形轉換的方法，缺點是難以自動化轉換。

而利用裴氏圖轉換為階梯圖的法則，可再將符合此法則的階梯圖轉換為對應的裴氏圖，如此可以達到雙向轉換的目的。若轉換至裴氏圖，便可利用裴氏圖的數學性質分析圖形，計算出控制圖形的法則，再將此法則鑲嵌回階梯圖，使階梯圖更加智慧化。

由於 XML 技術[30]的成熟，裴氏圖加註語言(Petri Net Markup Language, PNML)[21]與 PLCopenXML[26]的發展也受到重視。裴氏圖加註語言是將裴氏圖的圖形轉換為文字格式，且利用 XML 的格式表達，因此裴氏圖加註語言不會有格式不同的問題。由於裴氏圖加註語言的出現，使得許多裴氏圖繪圖程式增加支援讀取裴氏圖加註語言的功能，因此在裴氏圖繪圖程式間便能以這套標準規格溝通，不同繪圖程式繪出的裴氏圖不再需要特定繪圖程式才能修改，解決了單一裴氏圖繪圖程式功能不足的缺點。而 PLCopenXML 為 PLCopen 國際組織[27]針對可程式邏輯控制器中使用的程式語言，制訂出的一套可供驗證之 XML 標準格式。將階梯圖圖型轉為此種文字格式，便可供使用者於網際網路中傳遞文

件並加以利用。

本論文將針對階梯圖與裴氏圖的 XML 關係進行研究，並利用 XML 有利於自動化轉換的優點，實做記號階梯圖至記號圖的自動轉換。

1.2 問題界定及研究目的

由於在階梯圖中，邏輯的判斷為即為系統資訊的流動，因此裴氏圖必須以沒有分流的資訊流型態表示，此種型態的裴氏圖即為記號圖(Marked Graph, MG)[36]。記號圖為裴氏圖的一種特例，兩者關係如圖 1.1 所示。因此本論文假設階梯圖轉換後的裴氏圖即為記號圖，並在第三章中定義了與記號圖對應的記號階梯圖。

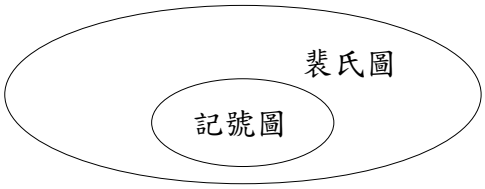


圖 1.1　裴氏圖與記號圖的關係

階梯圖雖然最廣泛被應用於 PLC，但其純粹為邏輯的判斷，不具備數學計算性質，因此只能做到控制系統，卻不能用於分析系統。而裴氏圖的數學特性，則常用於分析、模擬系統。本論文之研究目的在於將兩者的優點做結合，利用目前控制系統的記號階梯圖自動轉換為對應的記號圖。利用轉換後的記號圖進行系統的分析，可計算出監控系統的守恆方程式，達到便於控制的目的。

階梯圖的組成為布林邏輯式(Boolean Logic)[9, 26]，裴氏圖則可表達為矩陣模式[36]。在階梯圖轉換為記號圖的演算法設計上，必須要利用專家系統的 If-Then 規則[3]，將記號圖的規則矩陣數學特性與階梯圖的邏輯式做等構的對應，如圖 1.2 所示。而階梯圖與裴氏圖有各自的 XML 格式，演算法也必須在兩種不同的 XML 格式之中做判讀及轉換。

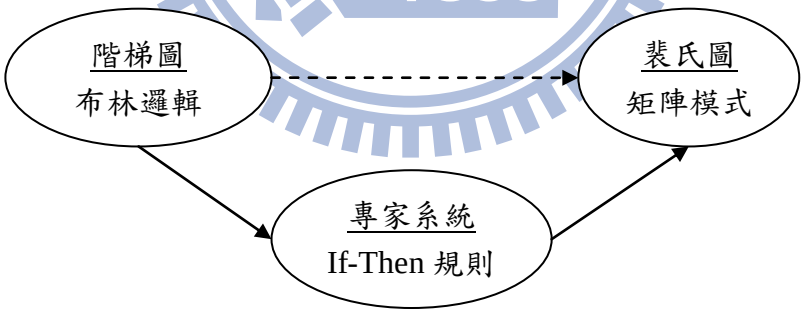


圖 1.2　階梯圖與裴氏圖之等構關係

在現場監控領域中，本論文將目前工業界最普遍使用於 PLC 的階梯圖轉換為具有數學分析性質的裴氏圖。研究範圍著重在記號階梯圖至記號圖的分析與轉換，如圖 1.3 所示。階梯圖控制 PLC 的實作以及裴氏圖的分析應用則不在本研究的探討範圍。

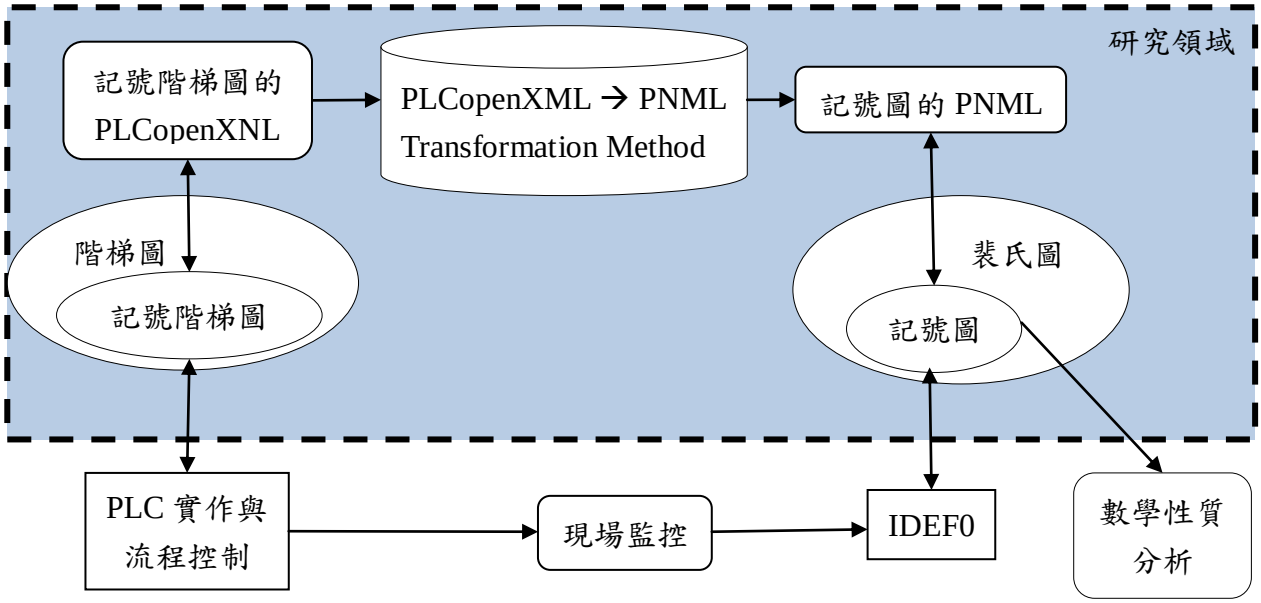

研究領域
記號階梯圖的
PLCopenXNL
PLCopenXML → PNML
Transformation Method
記號圖的 PNML
階梯圖
記號階梯圖
裴氏圖
記號圖
PLC 實作與
流程控制
現場監控
IDEF0
數學性質
分析

圖 1.3　現場監控流程

1.3 研究方法

本論文的重點在於記號階梯圖至記號圖轉換程式的設計與實作。主要研究方法架構分成四個階段，如下所示：

第一階段：『確認研究目的與研究範圍』了解目前學術界對於裴氏圖轉換為階梯圖的相關研究，並分析實作的可行性以及釐清研究範圍。

第二階段：『裴氏圖與階梯圖之回顧』了解裴氏圖與階梯圖的相關元件及其特性。

第三階段：『建立轉換標準階段』分析裴氏圖轉換為階梯圖的映射關係，並研究數學特性及邏輯式之間對照的關係。

第四階段：『程式實作及測試』將階梯圖至記號圖的轉換實作成電腦程式，並利用液體加入系統及灌模系統進行本論文結果之驗證。

圖 1.4 為本論文研究方法之流程。

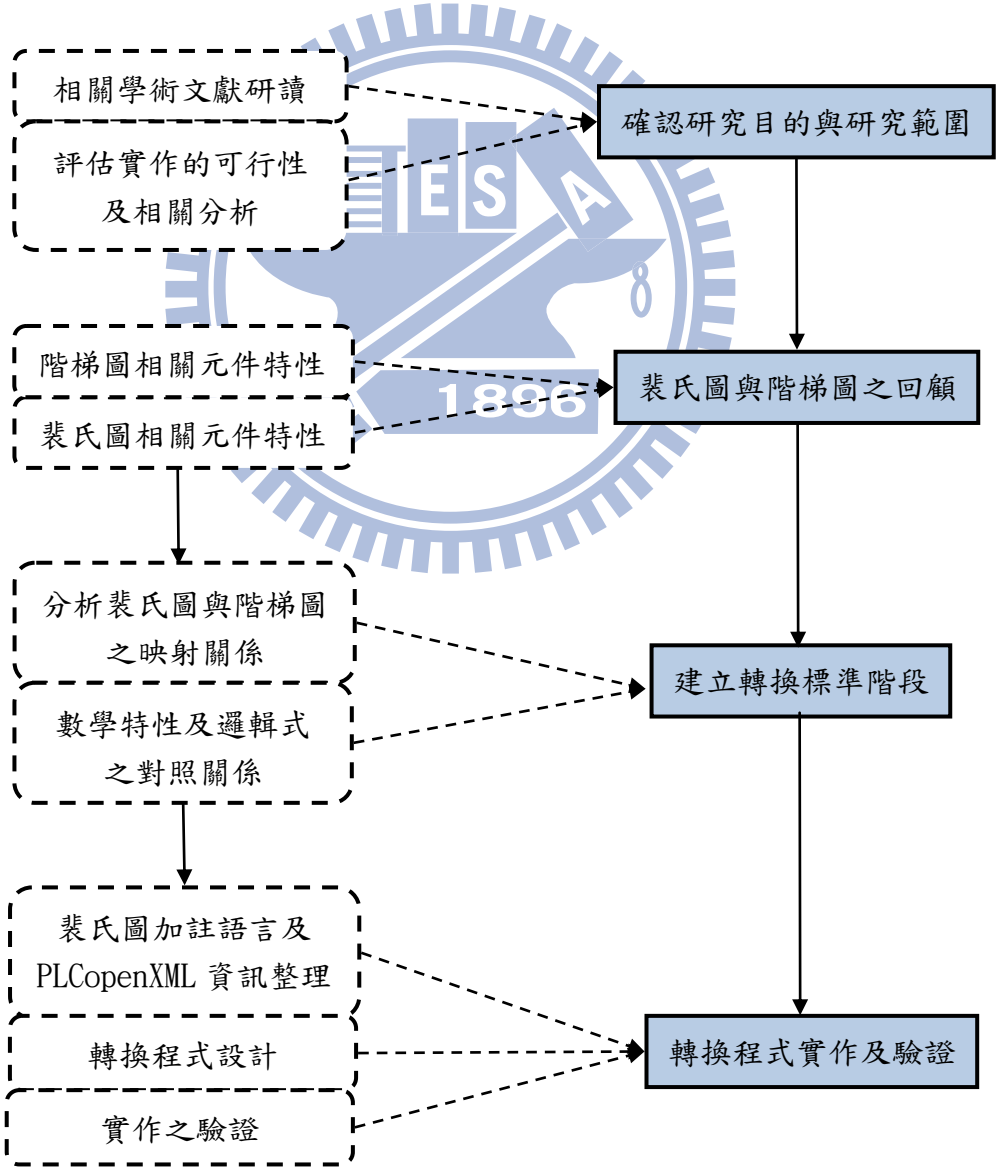


圖 1.4　本論文研究方法流程

1.4 論文架構

本論文共分六章，其中各章節內容大綱如下：

第一章：『緒論』說明本論文的研究動機、界定研究範圍、介紹研究方法及論文整體架構。

第二章：『文獻回顧』回顧與本論文相關之技術文件，並對相關基礎觀念做整理。

第三章：『可加註語言樹轉換法的分析』說明階梯圖中之記號階梯圖轉換為裴氏圖中之記號圖的映射分析方法。

第四章：『可加註語言樹轉換法程式的實作』將轉換之映射分析以程式碼撰寫實作。

第五章：『案例分析』利用灌模系統進行本論文的實作說明與驗證。

第六章：『結論』說明本論文之優缺點及未來方向。

圖 1.5 為本論文研究方法各流程與論文架構各章之對照。

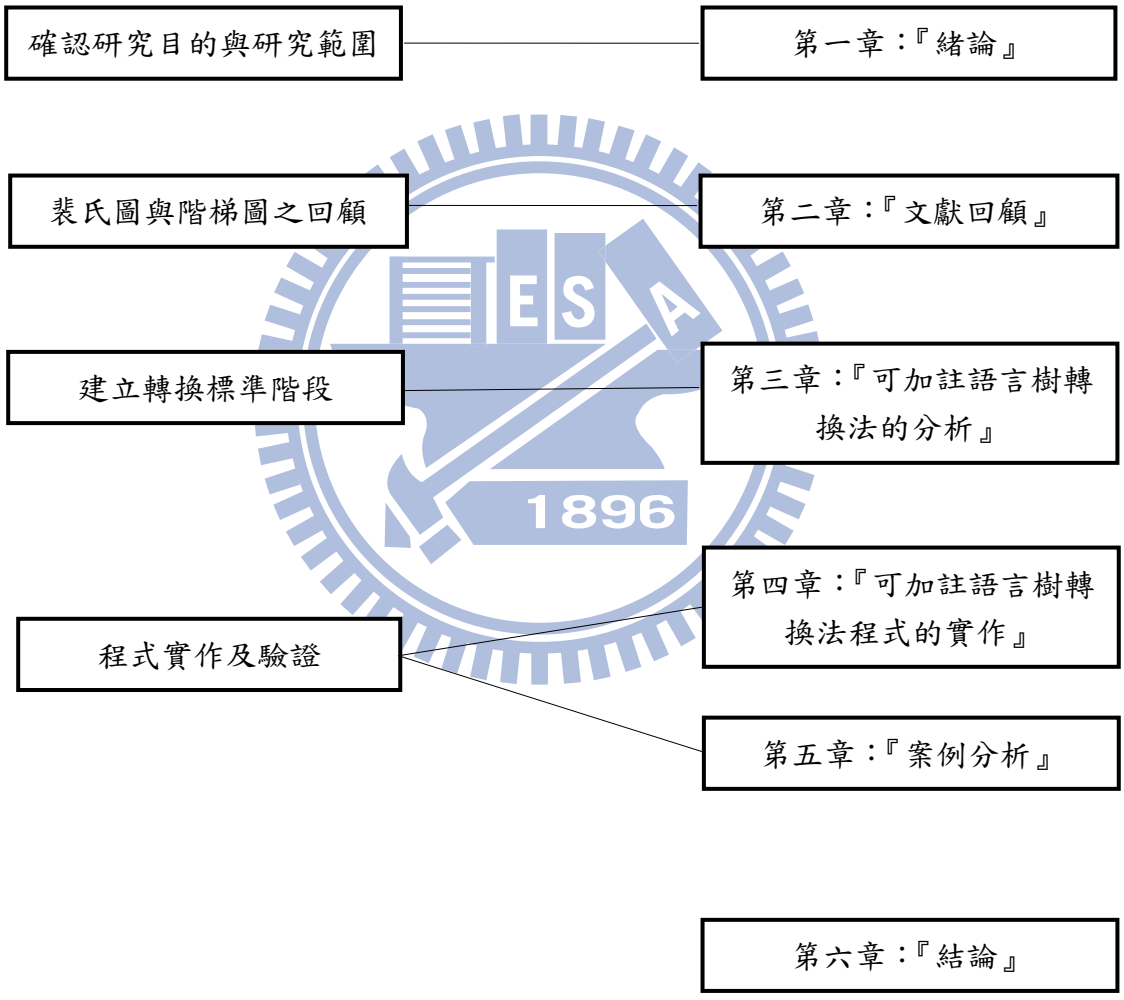


圖 1.5　本論文研究方法與論文架構對照

第二章　文獻回顧

本章主要在回顧階梯圖、裴氏圖、可加註語言與可加註語言處理技術。以及說明本論文所用到的各項流程圖與相關概念。第 2.1 節『階梯圖』說明階梯圖的基本元件、性質及邏輯，第 2.2 節『裴氏圖』說明裴氏圖的基本元件與記號圖以及裴氏圖之數學性質，第 2.3 節『裴氏圖與階梯圖之轉換』介紹裴氏圖轉換為階梯圖之法則，第 2.4 節『可加註語言』介紹可加註語言、裴氏圖加註語言及 PLCopen 加註語言，第 2.5 節『可加註語言處理技術』介紹可加註語言目前的處理技術分類。

2.1　階梯圖

階梯圖源自繼電器控制接觸器的電路圖，圖形類似階梯。階梯圖是通過連線，把具有指令的階梯圖符號接在一起的連通圖，主要可用來控制離散事件系統。2.1.1 節介紹 IEC61131 標準，2.1.2 節說明階梯圖基本元件。2.1.3 節說明階梯圖基本性質。2.1.4 節說明階梯圖之邏輯迴路。

2.1.1　IEC61131 標準

IEC61131[30]為國際電工協會(International Electrotechnical Commission, IEC)為可程式邏輯控制器(Programmable Logic Controller，PLC)[13]制定的標準，其中包含了八個部分，訂定了關於一般資訊、設備需求與測試等的標準。其中第三部分 IEC61131-3 為其程式規格，制定了關於可程式邏輯控制器的編程語言標準，這包含了階梯圖、時序流程圖(Sequential Function Chart, SFC)、基本指令(Instruction List, IL)、功能區塊圖(Function Block Diagrams, FBD)、結構化文字(Structured Text, ST)五種規格[13]。

此標準統一了可程式邏輯控制器的控制語法，並提出一套可跨不同目標平台的可程式邏輯控制器實現機制。規範中透過模組化的規劃與設計，將控制動作分為邏輯運算與硬體動作兩個部份，邏輯部分以共同的描述格式做統整，硬體動作則針對各硬體設計專屬之韌體函式庫，使得控制邏輯可以在各目標平台上使用硬體資源，這樣的設計使不同的控制晶片皆可執行以 IEC 61131-3 語法[30]所設計的控制動作。

設計人員只需學會 IEC 61131-3 語法[30]，便可使用所支援的控制晶片進行可程式邏輯控制器設計。此外，由於所設計的程式碼可以在不同的目標平台間重複使用，因此，透過自行建立的函式庫及利用重複使用的特性，更可縮短自動化流程的開發時程。

2.1.2　階梯圖基本元件

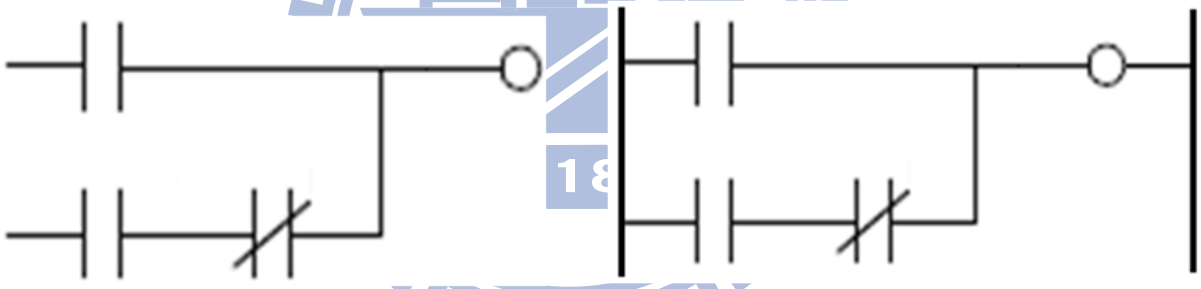
階梯圖的組成元件有左電源線(Left Power Rail)，右電源線(Right Power Rail)，輸入接點(Input Contact)，輸出線圈(Output Coil)[28]。左電源軌為系統電流的輸入端，右電源軌為系統電流的輸出端；輸出線圈代表系統的動作，也就是系統目前所處的狀態；輸入接點則可以依照邏輯組合成不同的條件，用以控制之後的輸出線圈動作。表 2.1 為階梯圖的組成

元件圖形對照及其代表意義。

表 2.1　階梯圖組成元件圖形對照及代表意義

名稱		圖形	代表意義
左、右電源軌 (Left /Right Power Rail)			系統電流的輸入與輸出端
輸入接點 (Input Contact)	常開接點 (Positive Contact) 常閉接點 (Negative Contact)		不同組合代表不同條件
輸出線圈(Output Coil)		( ) 或	代表系統的狀態

一個梯級(Rung)由輸入接點及輸出線圈組成，梯級與左右電源軌共同組成階梯圖。當系統通電後，電流由上至下、由左至由流動，便可透過梯級做出不同的動作達到控制系統的目的。圖 2.1 為階梯圖的基本圖形，圖 2.1(a)為梯級、圖 2.1(b)為梯級與左右電源軌組合成階梯圖。



(a) 梯級　　(b) 梯級與左右電源軌組合成階梯圖

圖 2.1　階梯圖的基本圖形

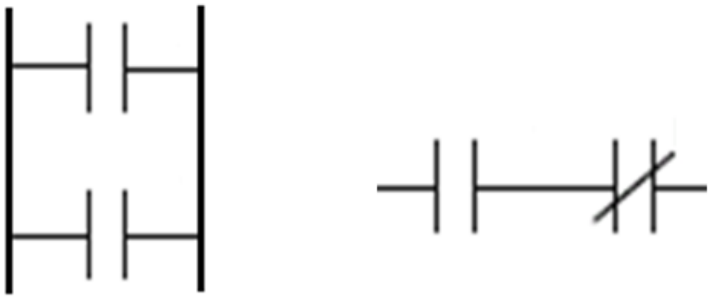
2.1.3　階梯圖基本性質

階梯圖的基本概念來自於電路圖，由許多繼電器組成的電路圖發展成由許多接點與線圈組成的階梯圖。與電路圖不同的是，階梯圖中的接線並非表示系統的實際接線狀況，而是表達運作的順序，因此階梯圖強調的是功能上的邏輯運作，而非實際線路的連接。

階梯圖的常開接點與常閉接點皆可不受限制的使用於階梯圖中，但每一級階梯必須至少有一個輸出，否則會造成邏輯錯誤。

2.1.4 階梯圖之邏輯迴路

在階梯圖中，使用 OR/AND 迴路達到控制的目的，並且可利用布林邏輯式表達。OR/AND 迴路在階梯圖中的作用就如同在電路圖中，OR 迴路指並聯電路組，當其中一個接點成立，則此組電路為通路；AND 迴路為串聯電路組，必須所有接點成立，此組電路才為通路。圖 2.2(a)為 OR 迴路、圖 2.2(b)為 AND 迴路。



(a) 並聯(OR)迴路　　(b) 串聯(AND)迴路

圖 2.2　OR/AND 迴路

2.2 裴氏圖

裴氏圖最早由德國 Carl Adam Petri 在 1962 年所發表，可以用來表達動態系統的離散事件，且可表達成矩陣模式，所以具備數學分析特性。只要有流程特性的資料，皆可利用裴氏圖表達，因此裴氏圖可應用於不同產業。2.1.1 節說明裴氏圖基本元件。2.1.2 節說明裴氏圖基本性質。2.1.3 節說明裴氏圖之規則矩陣。2.1.4 節介紹裴氏圖之特例：記號圖。

2.2.1 裴氏圖基本元件

裴氏圖主要由四個元件組成，分別為轉移點(Transition)、暫存點(Place)、弧(Arc)、浮標(Token)。而每個元件在裴氏圖代表的意義也各不同：轉移點表條件、暫存點表狀態、弧為系統的流程先後順序、浮標則在暫存點中構成不同的當前狀態。表 2.2 為各元件之圖形對照以及代表意義。

表 2.2 裴氏圖各元件之圖形對照及代表意義

名稱	圖形	代表意義
轉移點(Transition)		表示系統狀態改變時依據的條件
暫存點(Place)		表示狀態
弧(Arc)		為系統中執行流程的先後順序
浮標(Token)		浮標所在的暫存點會構成系統不同的狀態

其中弧具有方向性，且只能存在轉移點與暫存點之間。暫存點由弧指向轉移點，此暫存點稱為此轉移點的輸入暫存點；轉移點由弧指向暫存點，則此暫存點稱為此轉移點的輸出暫存點。由於暫存點表示狀態，轉移點表示進入某一狀態的條件，因此一個轉移點可以包含許多輸入暫存點及輸出暫存點，表示條件滿足的前後狀態。暫存點內可以同時擁有多個浮標，若暫存點中有N個浮標，代表有N筆資源或項目可供利用。裴氏圖中，浮標在賸存點中分布的狀況稱為標記狀態(Marking)。有浮標的暫存點代表系統目前處於此暫存點的狀態；反之，若暫存點中沒有浮標，代表系統目前不處於此暫存點的狀態。

2.2.2 裴氏圖基本性質

裴氏圖之基本性質可依初始標記狀態(Initial Marking)分為兩大類：行為性質(Behavioral Properties)與結構性質(Structural Properties)[36]。行為性質與裴氏圖的初始標記狀態有關，例如：可達性(Reachability)、活性(Liveness)等性質的探討。結構性質是指與初始標記狀態無關的性質，用來探討裴氏圖的結構包含安全性(Safeness)、限制性(Boundedness)、浮標不滅性(Conservativeness)、可逆性(Reversibility)與一貫性(Consistent)等[36]。

裴氏圖的分析方法有計算週期時間(Cycle Time)、繪製系統可達圖(Reachable Graph)、鎖死路徑分析(Deadlock Analysis)、計算不變量(Invariant)等[36]。隨著裴氏圖的複雜度增加，使用者難以自圖形觀察出裴氏圖是否鎖死，因此鎖死分析是初步設計裴氏圖相當重要的問題。若裴氏圖中的浮標數量較多，使用者很難判斷系統整體所有狀態的變更情況，因此可達圖亦可用來作為分析裴氏圖的方法之一。

由於裴氏圖為一動態系統，因此在分析裴氏圖的方法中，不變量是最基礎且最為重要的觀念。不變量是以數學為基礎用來表示系統狀態的一組數學聯立方程式，亦可看做一組矩陣，依照性質上的不同可分為暫存點不變量(P-Invariant)及轉移點不變量(T-Invariant)。利用兩種不變量都可用來監控動態系統。

2.2.3 裴氏圖之數學特性及規則矩陣

裴氏圖可以數學定義為五個面向的集合，PN = (P, T, F, W, M_0)。其中 P = {p_1, p_2, ..., p_m}表示一個暫存點的有限集合，T = {t_1, t_2, …, t_m }表示一個轉移點的有限集合。F⊆(P×T)∪(T×P)為弧的集合，表示 P 與 T 的關係，W:F→{1, 2, 3, …}，為權重函數(Weight Function)。M_0:P→{0, 1, 2, 3, …}，為暫存點的初始標記狀態。其中P ∩ T = ∅和P ∩ T ≠ ∅。裴氏圖若沒有初始標記狀態擇定義為 N = (P, T, F, W)。給予初始標記狀態值後，表示成(N, M_0)，即 PN = (P, T, F, W, M_0)[36]。

裴氏圖用狀態方程式(State Equation)描述其動態行為。狀態方程式定義：$M_k = M_{k-1} + A^T u_k$，k = 1, 2,…，令狀態M_k為m×1的向量，M_k為某個激發順序中第 k 次激發的浮標分布狀態；u_k為n × 1的向量，u_k為由狀態M_{k-1}要到狀態 M_k 第 k 次激發的規則矩陣，因此狀態方程式有可以初始標記狀態表達為$M_d = M_0 + A^T \sum_{k=1}^{d} u_k$。圖 2.4 為圖 2.3 所示裴氏圖之狀態方程式。

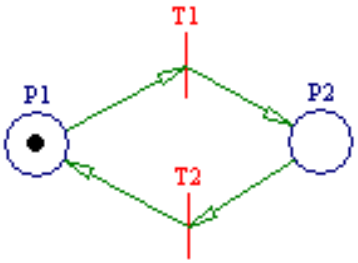


圖 2.3　裴氏圖

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

圖 2.4　裴氏圖之狀態方程式

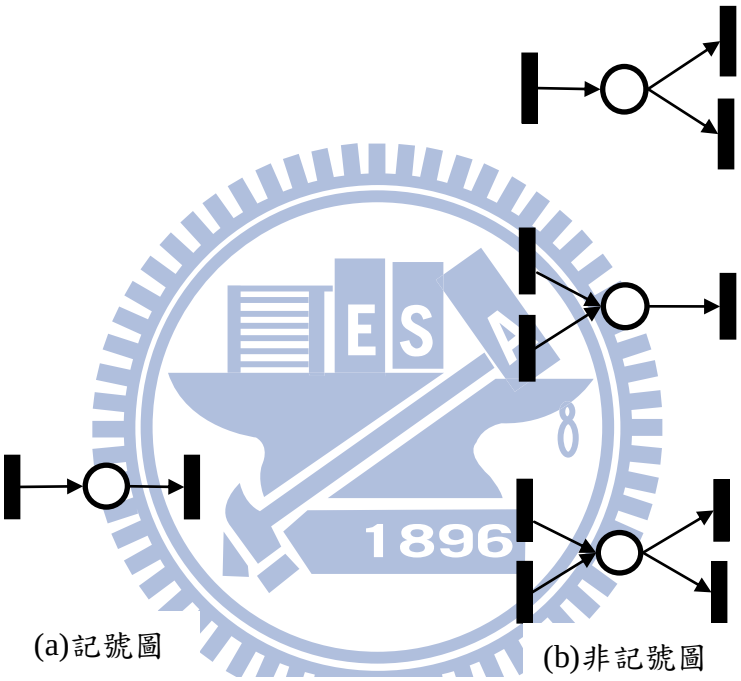
裴氏圖的規則矩陣(Rule Matrix)為此系統運行的規則依據，在其行列式中，行代表轉移點，列代表暫存點。其中數字代表通過某轉移點後，對應到的暫存點，其浮標數目之增減。此數學特性有助於降低在轉換過程中找尋觸發條件及對應轉移點的複雜度。以圖 2.3 之裴氏圖為例，圖 2.5 為其規則矩陣。

$$\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$$

圖 2.5　裴氏圖之規則矩陣

2.2.4　記號圖

記號圖為一種裴氏圖。記號圖中每個暫存點，必須只能有一個輸入及一個輸出，意即每個暫存點前端及後端只能連接一個轉移點。記號圖如圖 2.6(a)所示，圖 2.6(b)為非記號圖。記號圖是裴氏圖中，條件規定最嚴格的圖形且最常被用在分析系統上。而由於其沒有分流的特性，廣泛被應用於表達系統的資訊流。



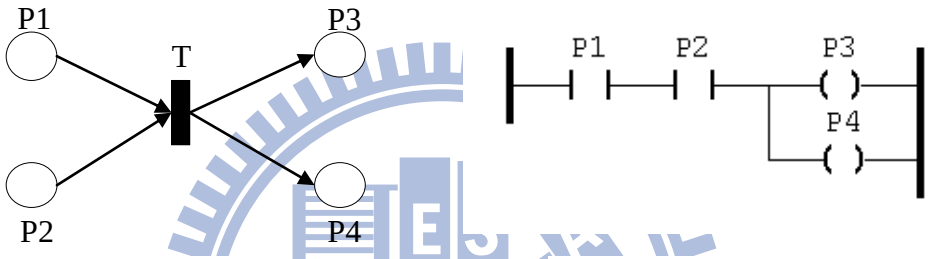
(a)記號圖　　(b)非記號圖

圖 2.6　裴氏圖分類

2.3 裴氏圖與階梯圖之轉換

對於裴氏圖與階梯圖，學術界多為裴氏圖轉換為階梯圖的文獻。最常見的轉換法則是以暫存點的角度[37]來考量。每個暫存點表示一個輸出線圈，找出以此暫存點為輸出暫存點的轉移點，並將此轉移點的輸入暫存點當做此輸出線圈的輸入接點條件式。因此在階梯圖中，所有的輸入接點與輸出線圈，標示的皆為暫存點。且對於輸入接點條件式的考量，只針對暫存點前的轉移點，未考量暫存點後之轉移點的部分。

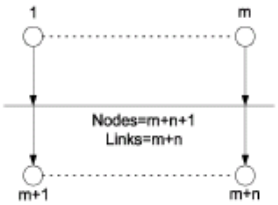
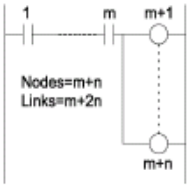
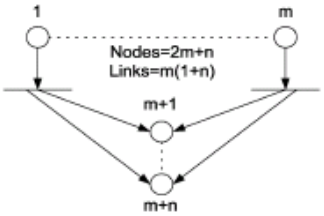
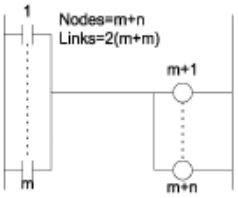
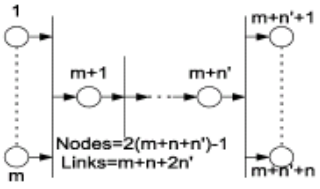
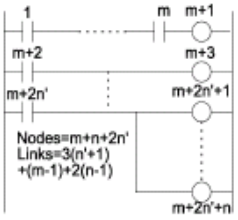
以圖 2.7(a)裴氏圖為例，為其相對應的階梯圖為圖 2.7(b)。中暫存點 P3 及 P4 前之轉移點皆為 T，因此只需考慮轉移點 T 之輸入暫存點 P1 及 P2 即可。P1 及 P2 需同時有浮標才能觸發轉移點 T，因此 P1 及 P2 為串聯之輸入接點；而轉移點 T 一但觸發，暫存點 P3 及 P4 皆會有浮標進入，故為並聯之輸出線圈。利用以上法則，可將裴氏圖與階梯圖的邏輯式大致分類為表 2.3 所示。



(a) 裴氏圖　　(b) 圖(a)對應之階梯圖

圖 2.7　裴氏圖與對應之階梯圖

表 2.3　裴氏圖與階梯圖的邏輯式架構分類

Constructs	Petri nets	Ladder diagrams
Logical AND		
Logical OR		
Sequential Model		

2.4 可加註語言

可加註語言(eXtensible Markup Language, XML)[30, 34]由全球資訊網聯盟(World Wide Web Consortium, W3C)發展而成，並在 1998 年二月發布為正式的標準規範。它是一種標籤語言，用於標示具有結構性資料訊的電子文件，主要用於跨平台傳遞資料、特別是能藉網路連結的平台。2.4.1 節介紹可加註語言的發展及技術。2.4.2 節介紹階梯圖的加註語言—PLCopen 加註語言。2.4.3 節介紹與裴氏圖有關的加註語言—裴氏圖加註語言。

2.4.1 可加註語言技術及發展

XML 由標準廣義加註語言(The Standard Generalized Markup Language, SGML)發展而來，可將文件中的標記與內容做明確的分隔為其特色，且所有文件的標示及使用方式均一致。XML 簡化 SGML 中較複雜與少用的特性，讓使用者可以輕鬆的定義需要的文件型態，也讓程式設計者更容易用程式做處理，如此的簡化使得 XML 更容易撰寫應用程式、更容易理解且更適合在網路上傳輸與整合。

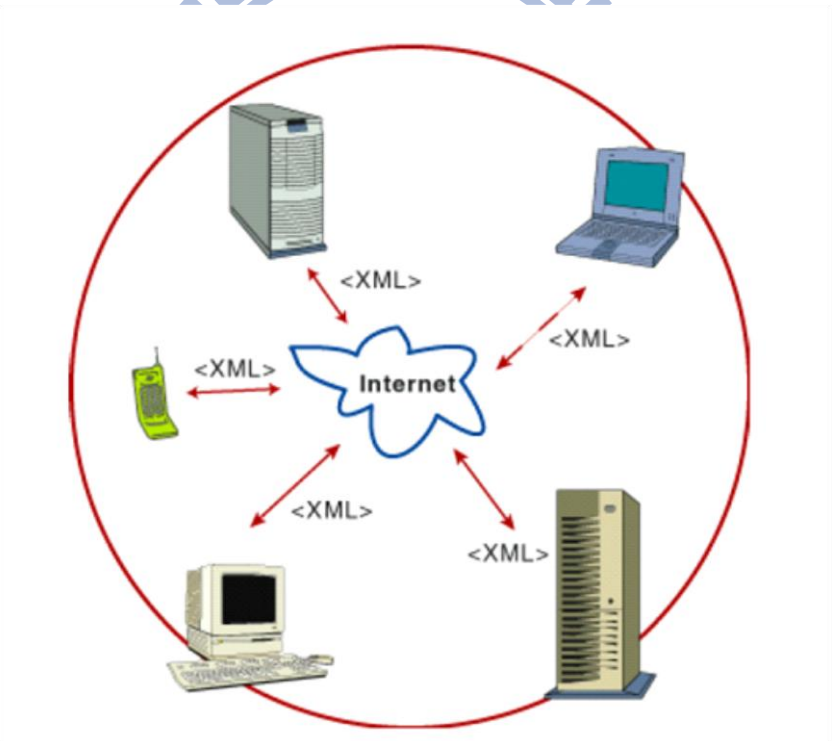
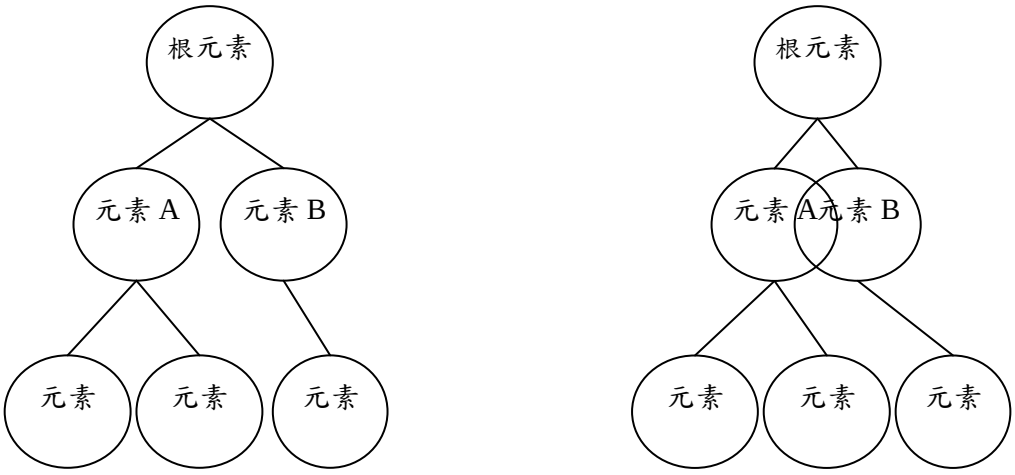


圖 2.8　XML 與網路之互動[6]

每份 XML 檔案必須有根元素，這可以代表最上層整體檔案的涵意。由於 XML 檔案採用結構式思考，所以在表達方式上就如同一棵樹，除了根元素外其他元素都必須有一父元素。在 XML 檔案中以<元素名稱>來表示起始標籤(Start Tag)，若元素沒有向下延伸的子元素或沒有元素本文，以</元素名稱>表示結束標籤(End Tag)。XML 檔案的元素結構上最大的禁忌為元素的重合，元素和元素之間必須要獨立，正確的 XML 樹狀結構(XML Tree)如圖 2.9(a)所示，元素重合的情況如圖 2.9(b)所示。



(a) 正確結構　　(b) 不正確結構

圖 2.9　XML Tree 結構的辨別

2.4.2　PLCopen 加註語言

PLCopen 加註語言為階梯圖之標準 XML 文件格式，2.4.2.1 節介紹 AutomationML、PLCopen 標準規範及本論文選用 PLCopenXML 之原因，2.4.2.2 節說明利用 PLCopen 加註語言表達階梯圖的架構。

2.4.2.1 AutomationML 及 PLCopen 標準規範

AutomationML(Automation Markup Language)[22]為一種以 XML 為基礎的文件標準格式，可用以儲存及交換工廠機械設備的資訊。AutomationML 主要目的在整合不同的近代工程設備工具，如可邏輯程式控制器、機械手臂控制(Robot Control)[14]等。其格式包含四大部分，分別為佈局(Topology)、幾何(Geometry)、運動學(Kinematics)及邏輯(Logic)[22]。

AutomationML 之邏輯部分使用的標準為 PLCopen 加註語言，此為 PLCopen[27]訂定的標準。PLCopen 為一個解決與工業控制程式設計相關主題的國際性組織，成立於 1992 年。其下分為兩個委員會，分別為科技委員會(Technical Committees, TC)及推廣委員會(Promotional Committees, PC)[27]。其中科技委員會的第六組(TC6)利用 IEC61131-3[30]中統一的標準程式規範，制定了 PLCopen 加註語言。PLCopen 加註語言為 XML 文件模式，提供使用者統一的文件格式，得以跨平台及利用網際網路傳送，做不同領域的應用。因本論文中轉換 XML 的部分與 PLC 的邏輯較為相關，因此採用 PLCopen 加註語言做為轉換的依據。

2.4.2.2 PLCopen 加註語言架構

PLCopen 加註語言為階梯圖的加註語言，與裴氏圖加註語言相同，每份 PLCopen 加註語言文件代表一個階梯圖。但與裴氏圖加註語言不同的是，PLCopen 不只可用於表達階梯圖，只要是 IEC61131-3 標準中的規範編程語言，皆可以 PLCopen 加註語言為其標準規範

語言文件。圖 2.10 為一個簡單的階梯圖，只包含了左右兩條電路、一個常開開關與線圈。

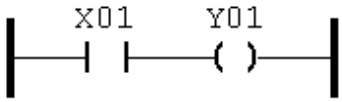


圖 2.10　簡單的階梯圖

圖 2.10 對應的元素可由 PLCopen 加註語言文件表示，標籤<leftPowerRail>及<rightPowerRail>代表左右兩條電路，<contact>和<coil>代表開關及線圈，而每一項標籤會有附屬標籤如 localId。

表 2.4 可以對圖形做更詳盡的說明，例如標籤<LD>代表這個 XML 文件為 PLCopen 加註語言形式寫成階梯圖，且其下所有標籤為階層式排列。利用標籤<leftPowerRail>、<rightPowerRail >、<contact>和<coil>定義左右兩條電路、開關及線圈。其附加屬性 localId 則給予這些元素一個容易辨識的代號，height 及 width 則決定了元素的圖形大小。標籤<position>代表元素的 x、y 座標位置，標籤<connectionPointOut>及<connectionPointIn>下層的<connection refLocalId>為元素之輸入輸出對應到之元素代號。<relPosition>為元素的相對位置，<variavble>則為使用者給予元素的代號。

表 2.4　PLCopen 加註語言文件

```
<LD>
  <leftPowerRail localId="1" height="40" width="2">
    <position x="39" y="100"/>
    <connectionPointOut formalParameter="">
    </connectionPointOut>
  </leftPowerRail>
  <contact localId="2" height="15" width="21">
    <position x="78" y="112"/>
    <connectionPointIn>
      <relPosition x="0" y="8"/>
      <connection refLocalId="1">
        <position x="78" y="120"/>
        <position x="41" y="120"/>
      </connection>
    </connectionPointIn>
    <connectionPointOut>
    </connectionPointOut>
    <variable>X01</variable>
  </contact>
 <coil localId="3" height="15" width="21">
    <position x="144" y="112"/>
    <connectionPointIn>
      <relPosition x="0" y="8"/>
      <connection refLocalId="2">
        <position x="144" y="120"/>
        <position x="99" y="120"/>
      </connection>
    </connectionPointIn>
    <connectionPointOut>
      <relPosition x="21" y="8"/>
    </connectionPointOut>
    <variable>Y01</variable>
  </coil>
  <rightPowerRail localId="4" height="40"width="2">
    <position x="203" y="100"/>
    <connectionPointIn>
      <relPosition x="0" y="20"/>
      <connection refLocalId="3">
        <position x="203" y="120"/>
        <position x="165" y="120"/>
      </connection>
    </connectionPointIn>
  </rightPowerRail>
</LD>
```

PLCopenXML 文件也可表達成樹狀結構，其中類別標籤為方型格子，屬性標籤及文字區域標籤為子彈形格子，圖 2.11 為 PLCopenXML 文件的 XML Tree。

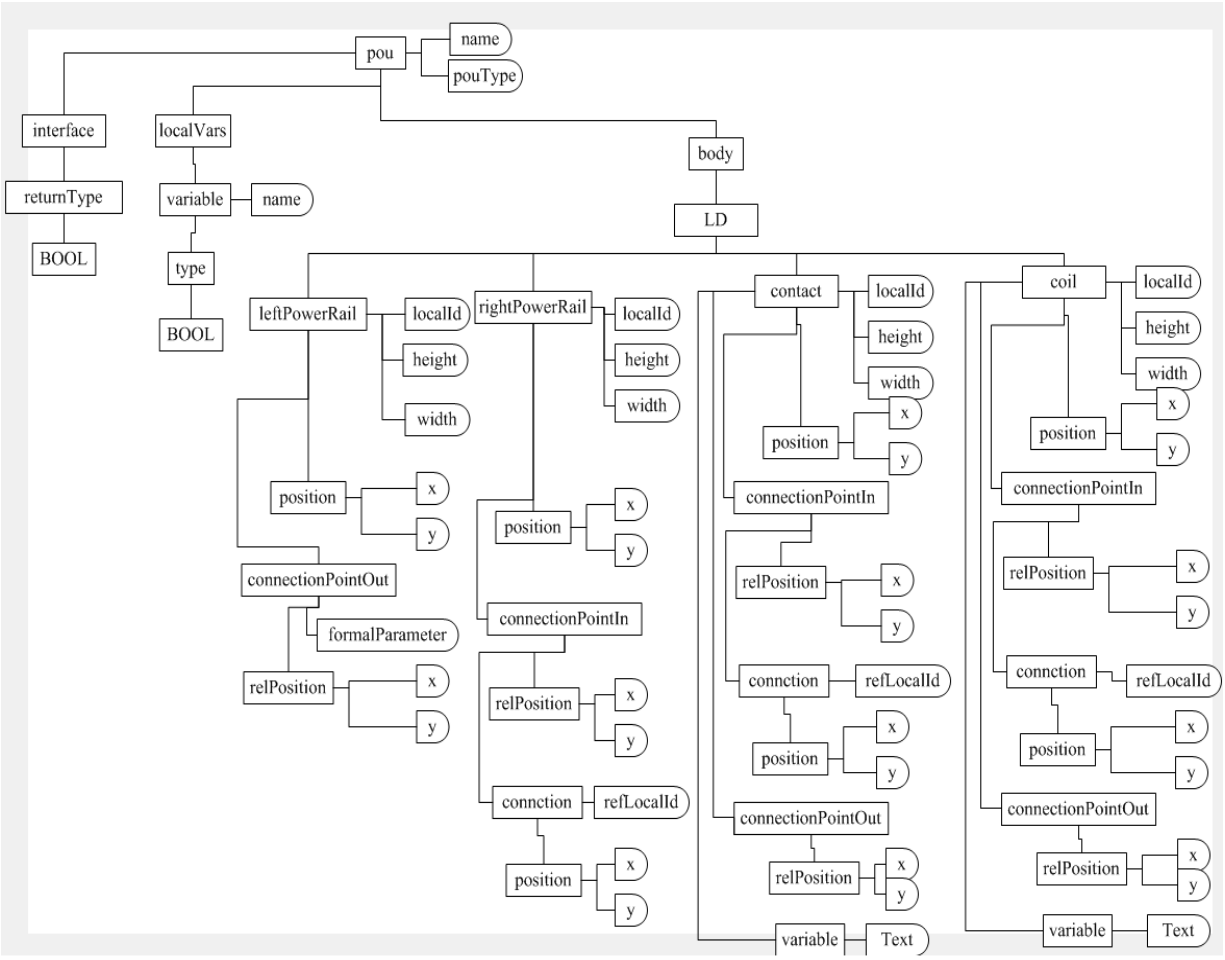


圖 2.11　PLCopenXML 的 XML Tree

2.4.3　裴氏圖加註語言架構

裴氏圖加註語言(Petri Net Markup Language, PNML)[4, 31]為裴氏圖的標準規範語言文件，國際標準組織(International Organization for Standardization, ISO)[10]及國際電工協會依據 ISO/IEC 15909[12]標準制訂，並在 2004 年 12 月首次成為規範[12]。每份 PNML 文件即代表一個裴氏圖。本論文第 2.1.3 節討論過裴氏圖的數學性質，若未給予裴氏圖初始標記狀態為 N = (P, T, F, W)，若加入初始狀態(N, M_0)，裴氏圖定義為 PN = (P, T, F, W, M_0)，其中 W 權重在一般的裴氏圖中皆設定為 1。因此一個裴氏圖模型在 PNML 文件中可以用暫存點 P、轉移點 T、箭號 F 及其初始狀態 M_0 描述，而圖形位置可用座標表示。而在 PNML 文件中分別用標籤(Tag)<place>、<transition>、<initialMarking>與<arc>來代表暫存點、轉移點、初始標記狀態與箭號。例如圖 2.12 是一個簡單的裴氏圖，各只有一個暫存點、轉移點及箭號，可用 PNML 文件來表示。

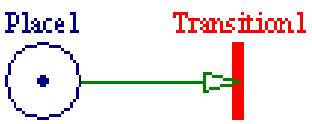


圖 2.12　簡單的裴氏圖

其中對應的標籤可由表 2.5 中的文件表示，且每一項標籤會有附屬標籤如 id，使得元素在文件中更容易辨識。且表 2.5 可以對圖形做更詳盡的說明，例如標籤<pnml>代表這個 XML 文件為 PNML 形式寫成，且其下所有標籤為階層式排列。標籤<net>代表之下為一個裴氏圖，附加屬性 type 說明此為一個更新裴氏圖(Updated Petri Net, UPN)[17]。接下來再定義每個暫存點<place>、轉移點<transition>、箭號<arc>，其附加屬性 id 則給予這些元素一個容易辨識的代號，其中標籤<arc>下的 source 及 target 表示這個箭號的來源及目標之 id。標籤<graphics>下的 position 為其代表元素的 x、y 座標位置，<name>為畫裴氏圖時使用者給予的代號。<place>下的標籤<initialMarking> value 為 1 代表其為初始標記狀態。由上述文件發現，利用文字表達圖形非常複雜，也顯示以人工準確繪製圖形的複雜度。

表 2.5　裴氏圖加註語言文件

```
<pnml>
    <net id="n1" type="UPNPNML">
        <place id = "p1">
            <graphics>
                <position x="166" y="177"/>
            </graphics>
            <name>
                <value>Place1</value>
            </name>
            <initialMarking>
            </initialMarking>
            <attributeX>
                <value>0.00</value>
            </attributeX>
            <attributeY>
                <value>0</value>
            </attributeY>
        </place>
        <transition id = "t1">
            <graphics>
                <position x="232" y="177"/>
            </graphics>
            <name>
                <value>Transition1</value>
            </name>
        </transition>
        <arc id = "a0" source = "p1" target = "t1">
            <graphics/>
                <value>1</value>
            <graphics/>
        </inscription>
        <type value = "normal"/>
        </arc>
    </net>
</pnml>
```

PNML 文件可表達成樹形結構，其中類別標籤為方型格，屬性標籤為箭號格，如圖 2.13 所示。

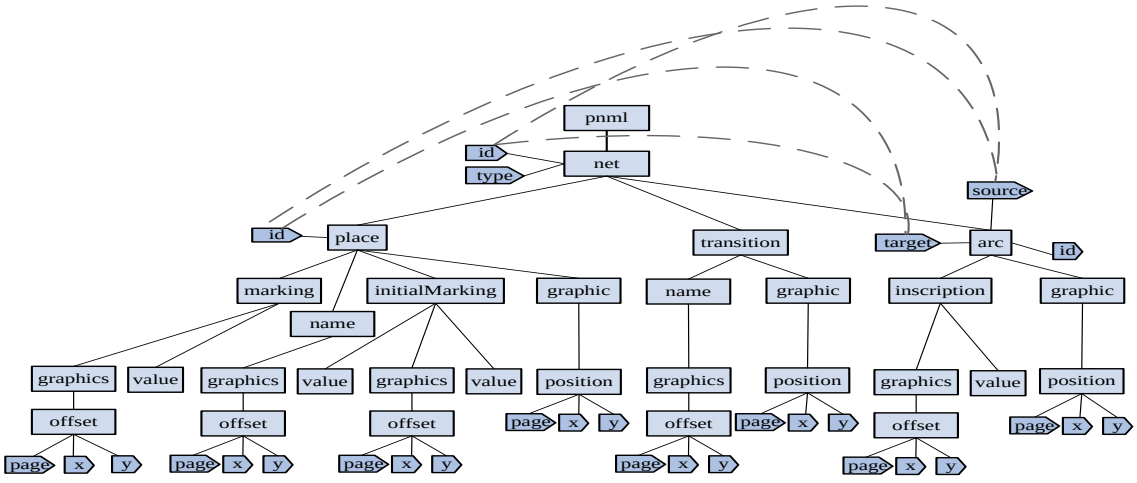


圖 2.13　PNML 的 XML Tree[1]

2.5 可加註語言處理技術

近幾年來，可加註語言的分析已大致被分類為三種不同處理類型的標準[20]。分別為SAX(Simple API for XML)、StAX(Streaming API for XML)、文件物件模型(Document Object Model, DOM)、及 XML 資料繫結（XML Data Binding）。2.5.1 節說明 SAX 和 StAX 這兩種處理技術，2.5.2 節說明何謂文件物件模型，並介紹兩種常用的文件物件模型解析器，2.5.3 節介紹何謂 XML 資料繫結，以及介紹常用於 XML 資料繫結的應用工具，2.5.4 節說明本論文之可加註語言樹轉換法中，XML 處理技術的選擇，2.5.5 節為 PLCopenXML 以及 PNML 之 XML 樹狀結構的物件說明。

2.5.1 SAX 和 StAX

SAX 解析 XML 時，採用串流的形式。當解析一份 XML 文件，由使用者定義回傳事件的函數，SAX 解析器會將文件從頭至尾分析一次，當遇到回傳事件的定義點時，將定義值回傳，便結束解析。且 SAX 處理為單方向的，若沒有重新開始，已解析過的資料無法再讀取[32]。此種解析法只在讀取文件時處理數據，不需將文件中的數據獨立於暫存記憶體中，因此較省記憶體空間，處理速度也較快。

StAX 也是採用串流的形式解析 XML 文件，與 SAX 不同的地方在於 StAX 可以設計應用程式碼，將想要觸發的事件拉出來，而非被動的等待解析器回傳事件。但對於程式程式開發者來說，需重複使用數據時，一次又一次的重新讀取將浪費大量的時間與記憶體空間，因此這種單方向性且不可重複的處理方式，雖然較簡便但卻也有使用上的困難。

2.5.2 文件物件模型

DOM(Document Object Model)稱作文件物件模型[30]，邏輯結構可以用節點樹的形式進行描述，因此其物件群為以樹形方式組合的概念。除了以說明為主的物件沒有與別的物件關聯外，其餘的物件都會與父元素或是子元素(Child)有關聯。因此 DOM 的概念最常用於設計 XML 格式上。通過對 XML 言文件的解析處理，XML 文件中的元素便轉化為 DOM 模型中的節點。DOM 模型的節點有 Document、Element、Comment、Type 等等節點類型，其中每一個 DOM 文檔必須有一個 Document 節點，並且為節點樹的根節點(Root)。它可以有子節點，或者葉子節點如 Text 節點、Comment 節點等。任何的格式良好的 XML 文件中的每一個元素均有 DOM 模型中的一個節點類型與之對應。利用 DOM 模型將 XML 文件轉化成 DOM 模型後，我們就可以自由的處理 XML 文件了。

在文件物件模型中，最常被使用的解析器為 JDOM 及 DOM4J。JDOM 的目的是成為 Java 特定文件模型，此解析器簡化與 XML 文件的互動並且比使用 DOM 轉換更快。由於是第一個 Java 特定模型，JDOM 一直被大力推廣和促進。而雖然 DOM4J 解析器代表了完全獨立的開發結果，但最初，其實是 JDOM 的一種分支。DOM4J 合併了許多超出基本 XML 文件可處理的功能，包括 XPath 支援、XMLSchema 支援等[24]。

2.5.3 XML 資料繫結

XML 資料繫結是一種處理 XML 文件的技術[16]，它可以利用應用程式讀取 XML 文件內容，甚至新增、修改、刪除 XML 文件內某筆資料。如此一來，XML 文件將可以做為不同應用系統之間交換資料的媒介。有別於 SAX 的串流形式及 DOM 的物件模式，此種技術是直接擷取 XML 文件中元素內的內容，所以不用透過繁雜的程式內容，也降低了程式碼的複雜程度。在 Java 應用程式中，XML 資料繫結的基本實作方式有兩種：(1.) 針對文件類型定義(Document Type Definition, DTD)的文法定義產生 Java 類別，這些 Java 類別會對應至 XML 文件內每個元素。(2.) 利用 Java 物件來處理 XML 元素值，或是檢查 XML 文件內容。

XML 資料繫結常見的應用工具有 JAXB(Java Architecture for XML Binding)及 Castor 等應用程式。JAXB 的資料處理流程如圖 2.14 所示[18]，首先在 Binding Complier 這個步驟，會先將 XML 架構(XML Schema)轉換為 Java 程式中的類別，再利用 JAXB 對 XML 文件做編組(Marshal)與解組(Unmarshal)的動作，編組與解組是將 XML 中的資料在儲存媒介與記憶體當中做轉換。

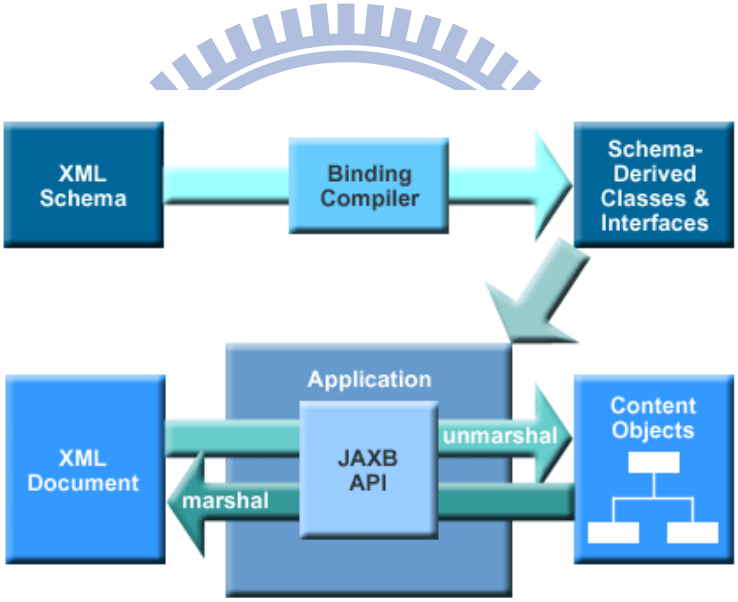


圖 2.14　JAXB 的資料處理流程[18]

2.5.4 XML 文件處理技術選擇

以上介紹的 XML 處理技術中，DOM 及 XML 資料繫結是目前普遍使用的兩種處理技術。雖然 XML 資料繫結中的 JAXB 可以直接將 XML 文件轉換為 Java 類別，且產生出 XML 內容的物件樹，但是對於 XML 文件的繼承關係，還是要由使用者判斷後自行擷取所要的物件。但 DOM 模型產生出的物件樹，已包含物件之間的繼承關係，可以直接對於某物件擷取其上下層物件，以供程式利用。

表 2.6 為在不同狀況下，使用不同 XML 處理技術的最佳選擇。當只是要傳遞 XML 文件至某一方進行解析時，若為從端(Server)則選擇 SAX，因為 SAX 處理技術是效率最高的，若是主端(Client)，可利用 XML 資料繫結，因為 XML 的格式已經給定。若是要完整的解析一份文件，可使用 XML 資料繫結，因為這個技術可以免去許多繁雜的程式。但若是要

進行兩份文件的轉換，建議使用 DOM，因為這個技術可以對於讀取進來的 XML 文件樹狀結構進行內容的修改及擷取，且對於轉換後要建立的 XML 文件先建立其 XML 樹狀結構，以讓要轉換出的文件可以依照其樹狀結構做制式的轉換。所以本論文選用 DOM 做為處理 XML 文件的技術。

表 2.6　不同狀況下，使用不同 XML 處理技術的最佳選擇

運用情況	選擇技術	優點
XML message transfer server	SAX	SAX is the fastest
XML message transfer client	XML 資料繫結	The data format is given
Configure file	XML 資料繫結	Easy to use
XML transfer	DOM	It can modify content and tree structure

2.5.5　PLCopenXML 及裴氏圖加註語言之文件物件模型

由於 DOM 模型可將 XML 文件表示為可加註語言樹(XML Tree)，因此 PLCopenXML 檔案及 PNML 檔案皆可表達為樹狀結構。利用 XML 的樹狀結構，可以標示出 XML 文件中的物件群，且可清楚的看出 XML 文件中父元素與子元素繼承的關係。PLCopenXML 的 DOM 模型包含了 variable 物件、leftPowerRail 物件、rightPowerRail 物件、contact 物件及 coil 物件，如圖 2.15 所示。

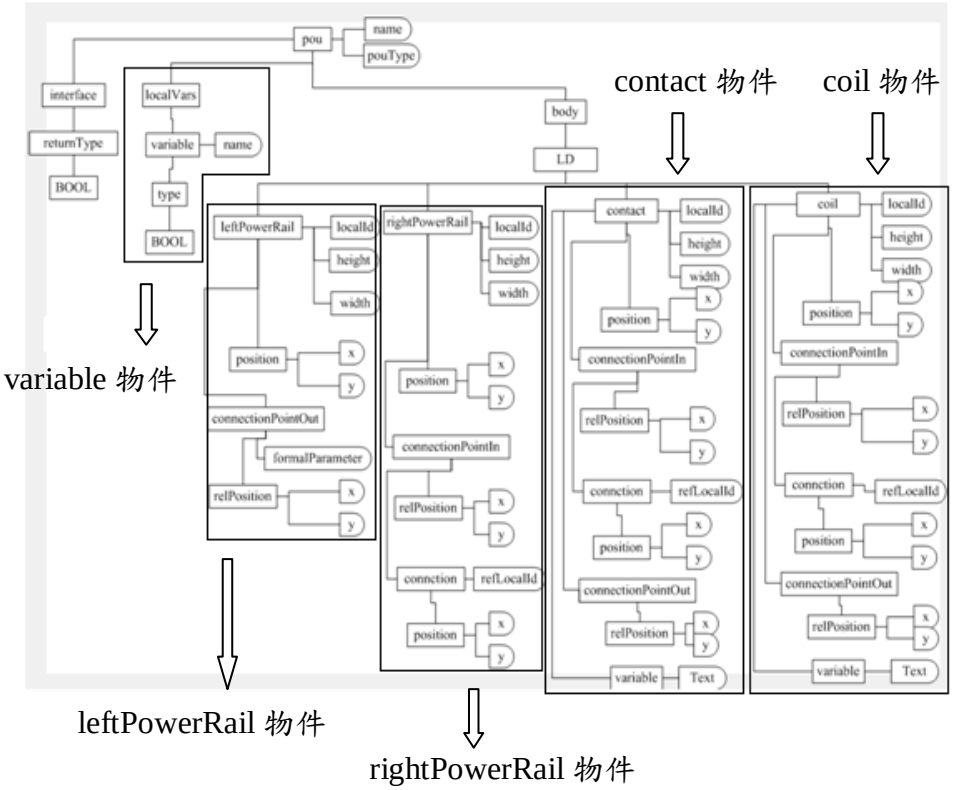


圖 2.15　PLCopenXML 的 DOM 模型

DOM 模型中的物件群，皆有其代表的意義，其中的物件可以對應到原本 XML 的文字、圖形，表 2.7 為 PLCopenXML 文件檔中與程式對應之物件說明。

表 2.7　PLCopen 文件檔物件說明

圖形	PLCopenXML 內容	物件	說明
	<project> 中略		
Variable	<localVars> <variable name="A"> 中略	name	圖中所有元件的名稱集合
	<leftPowerRail localId="1" height="40" width="2"> <position x="97" y="130"/> 中略 </leftPowerRail>	localId x y	電腦辨識的 Id 座標
	<rightPowerRail localId="2" height="40" width="2"> <position x="391" y="130"/> 中略 </rightPowerRail>	localId x y	電腦辨識的 Id 座標
 	<contact localId="3" height="15" width="21" **negated="true"**> <position x="143" y="142"/> <connectionPointIn> <relPosition x="0" y="8"/> <connection refLocalId="1"> 中略 <variable>A</variable> </contact>	localId negated x y refLocalId variable	電腦辨識的 Id 辨別常閉開關 座標 輸入端的 Id 圖形上的名稱
	<coil localId="5" height="15" width="21"> <position x="301" y="142"/> 中略 <connection refLocalId="4"> 中略 </connectionPointOut> <variable>C</variable> </coil>	localId x y refLocalId variable	電腦辨識的 Id 座標 輸入端的 Id 圖形上的名稱
	中略 </project>		

一個樹形結構可以標示出此 XML 文件的物件群，PNML 的 DOM 模型包含了 Place 物件、Transition 物件、Arc 物件，如圖 2.16 所示。

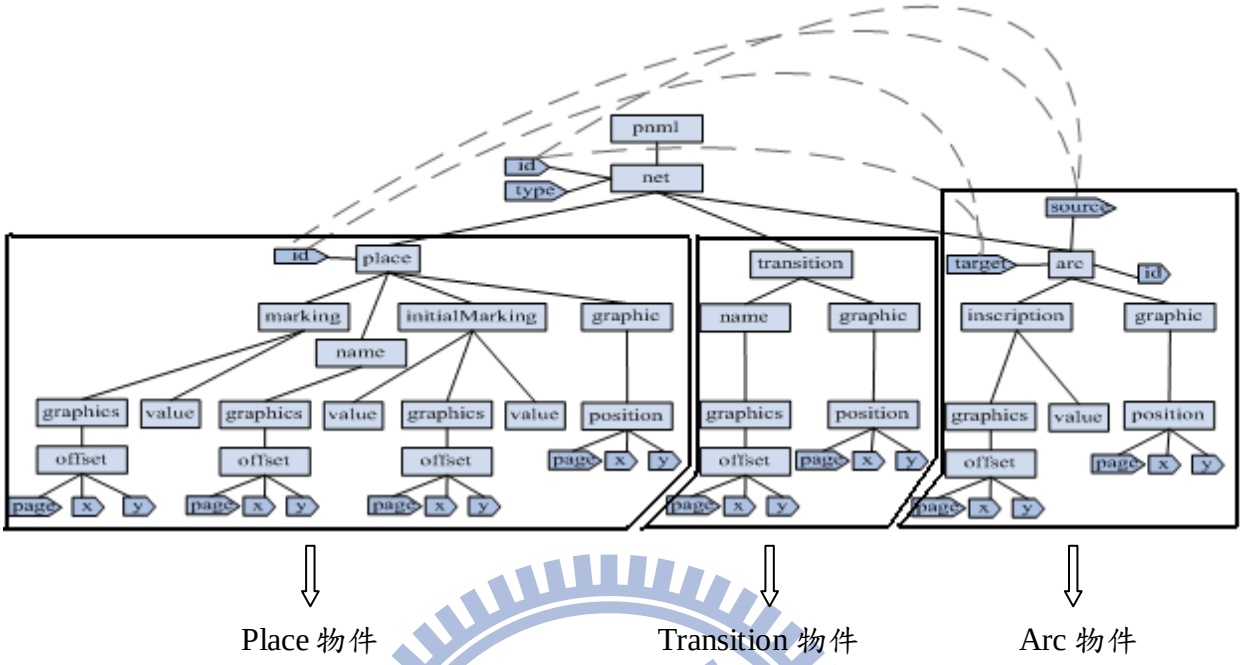


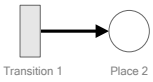
圖 2.16　PNML 的 DOM 模型[1]

DOM 模型中的物件群，皆有其代表的意義，其中的物件可以對應到原本 XML 的文字、圖形，表 2.8 及表 2.9 為 PNML 文件檔中與程式對應之物件說明。

表 2.8　PNML 文件檔物件說明

圖形	PNML 內容	物件	說明
	<?xml version="1.0" encoding="UTF-8"?> <PNML><net id="n1" type="PTNet">		
Place 2 	<place id="**p2**"> 中略 <value>**Place2**</value> </name> <initialMarking> 中略 </graphics> <value>**1**</value> </initialMarking> <graphics> <position page="1" x="**333**" y="**167**"/> </graphics> </place>	id name value x y	Id 為電腦所辨識 圖形上所看到的名稱。 此暫存點的 token 數目。 暫存點的座標。

表 2.9　PNML 文件檔物件說明(續)

Transition 1 	<transition id="**t1**"> <name> 中略 <value>**Transition1**</value> </name> <graphics> <position page="1" x="**191**" y="**167**"/> </graphics> </transition>	id value x y	Id 為電腦所辨識 圖形上所看到的名稱。 轉移點的座標
	<arc id="a528" source="**t1**" target="**p2**"> <inscription> 中略 </arc>	source target	此對應的是暫存點或轉移點的 ID source 代表前端，target 代表後端。
	</net></PNML>		

第三章　可加註語言樹轉換法的設計

本章說明可加註語言轉換法的設計。第 3.1 節『記號階梯圖與記號圖之映射分析』說明記號階梯圖與記號圖的狀態元件、條件元件，以及兩種圖形之間元件的映射關係，並且說明記號階梯圖的標記條件、不標記條件和自我控制迴路。第 3.2 節『記號階梯圖至記號圖的修正梯級與記號階梯圖的定義』中說明原始映射梯級由於可程式邏輯控制器的掃描方式不同，必須修正為修正映射梯級，並且定義記號階梯圖。第 3.3 節『記號階梯圖與記號圖的等構關係』利用記號階梯圖的邏輯判斷式與記號圖的規則矩陣驗證了兩者的等構關係。第 3.4 節『可加註語言樹轉換法的分析』說明轉換程式的架構，記號階梯圖與記號圖之間可加註語言樹的物件對應關係，以及轉換程式中三個階段的個別說明。第 3.5 節『偵查法則的轉換』說明記號圖的偵查法則數學式，與記號階梯圖邏輯判斷式的關係。

3.1　記號階梯圖與記號圖之映射分析

在記號階梯圖中，輸出線圈可以代表事件的狀態，而輸入開關的組合邏輯則用來控制相對應的輸出線圈；在記號圖中，各個暫存點代表圖中的狀態，而轉移點中的浮標組成不同的觸發條件使記號圖進入不同的狀態。3.1.1 節說明記號圖與記號階梯圖的狀態元件和條件元件，3.1.2 節說明記號階梯圖的標記條件及不標記條件，3.1.3 節說明自我控制迴路。

3.1.1　狀態元件與條件元件

記號階梯圖的狀態是由輸出線圈的動作表示；記號圖的狀態則由暫存點中的浮標來標記，浮標分佈的標記情形可以表示記號圖目前處於何種狀態。因此輸出線圈為記號階梯圖的狀態元件，而暫存點為記號圖中的狀態元件。在可加註語言樹轉換法中，當輸出線圈有動作，即代表暫存點被標記，所以這兩種元件在轉換時，一個輸出線圈就代表一個暫存點。

在記號階梯圖中，電流通過輸入開關組合成的串並聯電路，控制輸出線圈的動作，因此輸入開關中常開開關與常閉開關的串並聯方式，組合成輸出線圈的動作條件；在記號圖中，當轉移點的每個輸入暫存點皆為標記狀態時，轉移點才會被觸發，因此轉移點的輸入暫存點組合成此轉移點的觸發條件。但是轉移點不為記號圖的狀態元件，記號圖中的轉移點觸發條件必須搭配上轉移點本身才是控制記號圖暫存點的動作條件。所以轉移點觸發條件及轉移點本身，才能與記號階梯圖中常開及常閉開關構成的串並聯電路做映射。圖 3.1 為記號圖及階梯圖的條件元件。

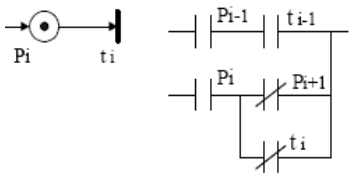


圖 3.1　記號圖之條件元件及記號階梯圖之條件元件

3.1.2 標記條件與不標記條件

標記條件與不標記條件為記號階梯圖中條件元件的分類。當記號階梯圖的標記條件成立，表示其對應到的記號圖暫存點會得到浮標，或者會一直處於有浮標的狀態；當不標記條件成立，代表其對應到的記號圖暫存點目前沒有浮標，或者浮標會轉移到下一個暫存點。標記條件與不標記條件若同時成立，表示記號圖中的狀態出錯，且會使得階梯圖的邏輯混亂，故標記條件與不標記條件不會同時成立。

3.1.3 自我控制迴路

圖 3.2 為記號階梯圖的自我控制迴路(Self-Holding Circuit)[8]，此控制迴路廣泛的被應用在階梯圖編程。在本研究中，此自我控制迴路可以用來表示記號圖中，一個暫存點最基本的行為模式。

當線圈 P 動作時，表示暫存點 P 被標記；常開開關 L 表示暫存點 P 的標記條件，當 L 開啟時，P 有動作；當 L 關閉時，由於 P 有一個自我維持的常開開關，所以 P 仍舊有動作；而常閉開關 U 表示暫存點 P 的不標記條件，當 U 開啟時，P 的動作便會取消；因此每一個記號圖中的暫存點，都可以發展出一個自我控制迴路。

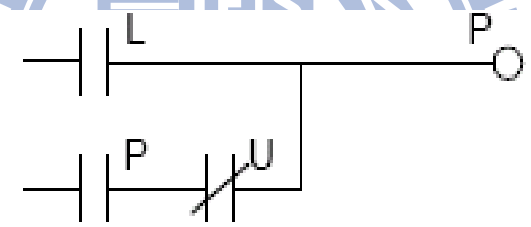


圖 3.2　記號階梯圖的自我控制迴路

3.2 記號階梯圖至記號圖的修正梯級與記號階梯圖的定義

記號階梯圖與記號圖皆為圖形表達的方式。兩者運行的邏輯若用圖形直接轉換，會造成轉換上的複雜及困難。因此本研究利用記號圖的數學矩陣模式，搭配記號階梯圖的布林表示法(Boolean Notation)[9]，降低轉換上的複雜度。3.2.1 節說明記號階梯圖轉換為記號圖的原始映射梯級，3.2.2 節說明針對記號階梯圖的非同步特性(Asynchronous Characteristics)[34]做修正的修正映射梯級，以及對記號階梯圖做定義。

3.2.1 原始映射梯級

記號圖若導入記號階梯圖的自我控制迴路，可以得到對應記號階梯圖的原始映射梯級。將自我控制迴路中的標記條件與不標記條件，對照記號圖的狀態/條件元件，轉換為布林表示法，便可得到階梯圖的原始映射梯級。圖 3.3 為圖 3.2 自我控制迴路與其對照的記號圖以及轉換後的布林表示法、圖 3.4 為圖 3.3 的原始映射梯級(Primary Mapping Rung)[8]。

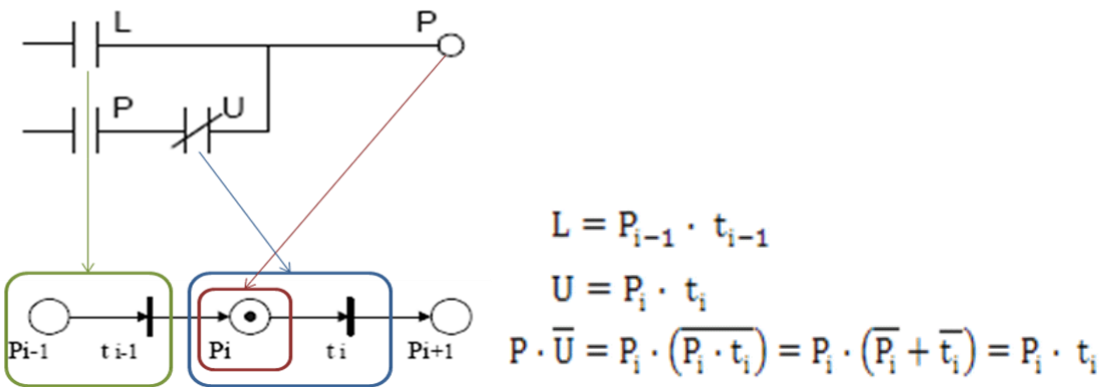


圖 3.3 自我控制迴路對照的記號圖及轉換的布林表示法

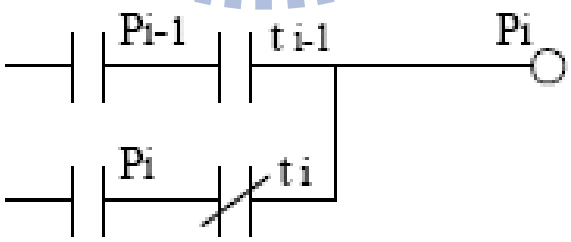


圖 3.4 為圖 3.3 的原始映射梯級[8]

3.2.2 修正映射梯級

目前可程式邏輯控制器使用的掃描模式(Scanning Modes)[8]大致分兩種，分別為垂直掃描(Vertical Scanning)以及水平掃描(Horizontal Scanning)。在垂直掃描模式時，以上的原始映射梯級可以正常運作；但若使用水平掃描模式時，以上的原始梯級便會因為記號階梯圖的非同步特性而不適用。

圖 3.5 的記號圖對照到圖 3.6 的記號階梯圖後，在圖 3.6 中箭號為水平掃描模式的運行

方向。假設線圈 P_i 為初始的標記狀態，掃描 B 線時，線圈 P_i 會因為常閉開關 t_i 的開啟而被關閉。因此理論上掃描 C 線時，線圈 P_{i+1} 會因為常開開關 t_i 的開啟而跟著動作，但事實上卻會因為常開開關 P_i 的關閉而導致線圈 P_{i+1} 無法動作。

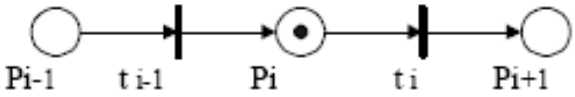


圖 3.5　記號圖

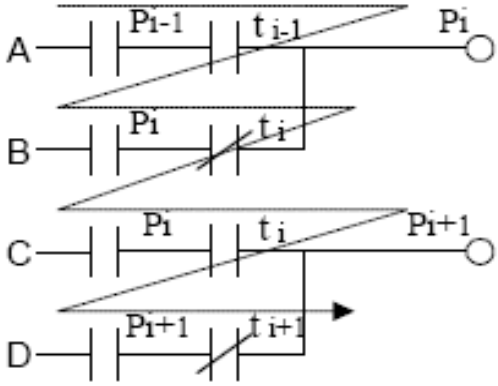


圖 3.6　為圖 3.5 映射的記號階梯圖

所以依照原始映射梯級的邏輯以及考慮非同步特性，修改原本映射梯級為修正映射梯級。在記號圖中，只要轉移點的觸發條件被滿足，轉移點會立即觸發。因此在記號圖中沒有不同步問題，所以由記號階梯圖回推至記號圖修改映射梯級，可以利用改變記號圖的暫存點動作順序，達到修正記號階梯圖非同步特性的效果。圖 3.7 為用記號圖解釋記號階梯圖的非同步特性。

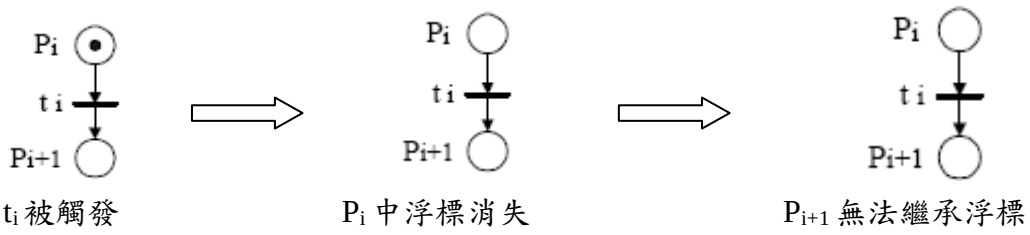
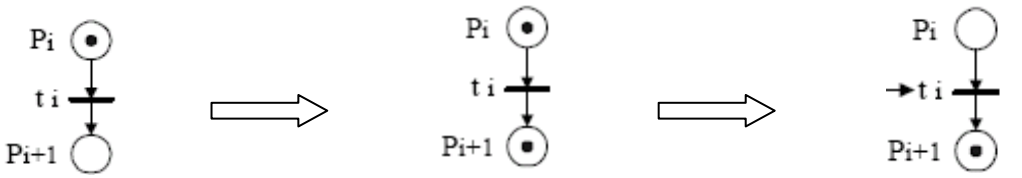


圖 3.7　利用記號圖解釋記號階梯圖的非同步特性

由圖 3.7 可以發現，當由記號階梯圖回推至記號圖時，暫存點 P_i 與暫存點 P_{i+1} 的不同步，造成浮標的消失。所以利用交換暫存點 P_i 及暫存點 P_{i+1} 的動作發生順序，來避免問題的發生。圖 3.8 為交換暫存點 P_i 及暫存點 P_{i+1} 的發生順序，浮標最後可以成功的到達暫存點 P_{i+1}。



t_i 被觸發　　P_{i+1} 先繼承浮標　　P_i 中浮標消失

圖 3.8　交換暫存點 P_i 及暫存點 P_{i+1} 發生順序

而由於暫存點的順序改變，在不標記條件的布林運算式 U 當中，必須加入暫存點 P_{i+1} 的成立，如此不標記條件才會將暫存點 P_{i+1} 的順序往前移。原本的$U = P_i \cdot t_i$ 要多加上 P_{i+1}，成為$U = P_i \cdot t_i \cdot P_{i+1}$，因此$P \cdot \overline{U} = P_i \cdot \overline{(P_i \cdot t_i \cdot P_{i+1})} = P_i \cdot (\overline{t_i} + \overline{P_{i+1}})$。圖 3.9 為利用修正映射梯級畫出圖 3.5 記號圖轉換後的記號階梯圖，其中常閉開關並聯組的邏輯式變為$(\overline{t_i} + \overline{P_{i+1}})$，比原始的梯級多了暫存點 P_{i+1}。

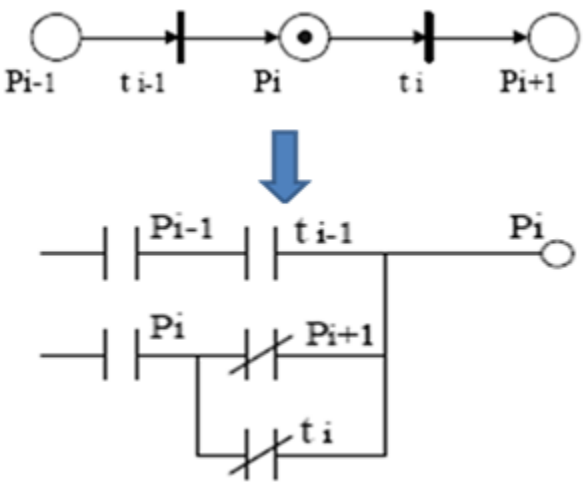


圖 3.9　利用修正映射梯級轉換圖 3.5 後之記號階梯圖

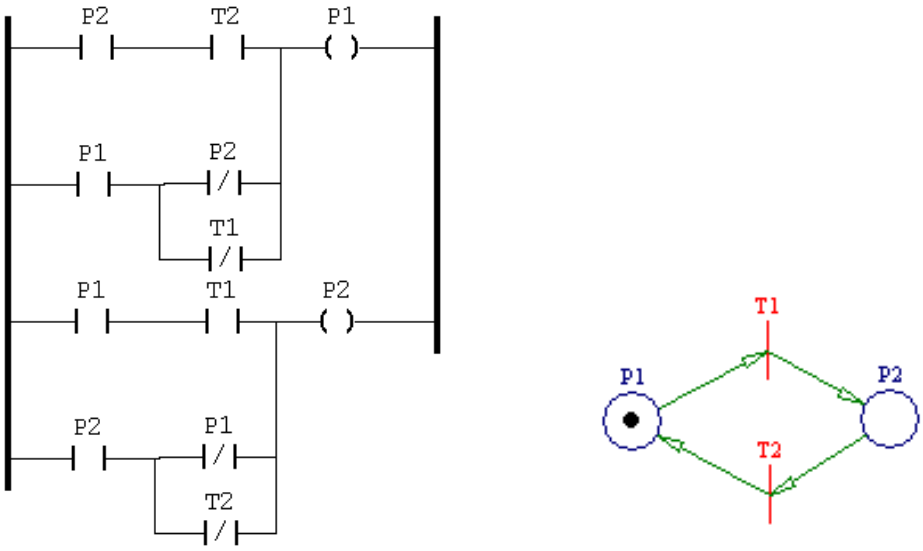


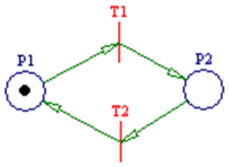
圖 3.10　等構的記號階梯圖與記號圖

第 3.3 節中修正後的映射梯級，可以組成一個完整的記號階梯圖。這種類型的階梯圖已排除非同步特性，每個層級包含一個動作線圈。每個動作線圈的控制條件分為標記條件與不標記條件，兩種條件用並聯邏輯控制動作線圈。有別於其他階梯圖，本論文將排除非同步特性的階梯圖定義為記號階梯圖(Marked Ladder Diagram)。記號階梯圖是與記號圖相對應的階梯圖，不僅解決非同步特性問題，也包含了記號圖各個暫存點只有一個輸出及一個輸入的特色。圖 3.10 為等構的記號圖與記號階梯圖。本論文著重在記號階梯圖與記號圖的對應轉換。

3.3 記號階梯圖與記號圖的等構關係

記號階梯圖與記號圖的等構關係，可以從專家系統的 If-Then 規則，針對記號圖的規則矩陣與記號階梯圖的布林邏輯式做驗證。其中 If-Then 規則又可以寫作邏輯式表達，因此記號階梯圖與記號圖的等構關係會以 If-Then 規則的邏輯式證明。

以圖 3.10 右邊的記號圖為例，其規則矩陣可以寫成圖 3.11 所示，在其行列式中，行代表轉移點，列代表暫存點。其中數字代表通過某轉移點後，對應到某暫存點的浮標數目之增減。每個暫存點對應到的數字，1 會使暫存點得到浮標，代表標記條件，-1 會使暫存點失去浮標，代表不標記條件。以圖 3.11 為例，P1 的不標記條件對應到 T1、P1 的標記條件對應到 T2；P2 的標記條件對應到 T1、P2 的不標記條件對應到 T2。



$$\begin{array}{c} \\ T1 \\ T2 \end{array}\begin{array}{c} \begin{array}{cc} P1 & P2 \end{array} \\ \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \end{array}$$

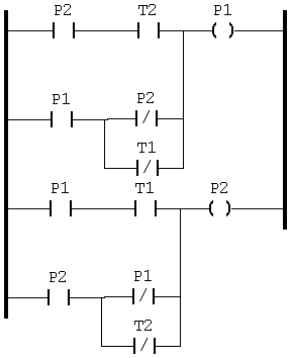
圖 3.11　記號圖及其規則矩陣

利用規則矩陣中暫存點的標記條件與不標記條件，就可以分別針對對應到的轉移點去作條件的邏輯判斷。以圖 3.11 的規則矩陣為例，P1 的不標記條件對應到 T1，再從 T1 為出發點找到 T1 觸發後到達的暫存點 P2，所以 P1 的不標記條件邏輯式為 $U = P1 \cdot (\overline{T1 \cdot P2})$。P1 的標記條件對應到 T2，從 T2 為出發點，找到使 T2 觸發的暫存點為 P2，所以 P1 的標記條件邏輯式為 $L = T2 \cdot P2$。綜合以上條件，P1 依照規則矩陣的 If-Then 規則寫出的邏輯式如圖 3.12 所示。

$$P1 = L+U = (T2 \cdot P2) + \left(P1 \cdot (\overline{T1 \cdot P2})\right) = (T2 \cdot P2) + (P1 \cdot (\overline{T1} + \overline{P2}))$$

圖 3.12　P1 依照規則矩陣寫出的邏輯式

圖 3.10 的記號階梯圖中，以 P1 為動作線圈的布林邏輯式如圖 3.13 所示。該圖分別為 P1 的記號階梯圖及其布林邏輯式。利用布林邏輯式，可寫出與其對應的邏輯式，如圖 3.14 所示。



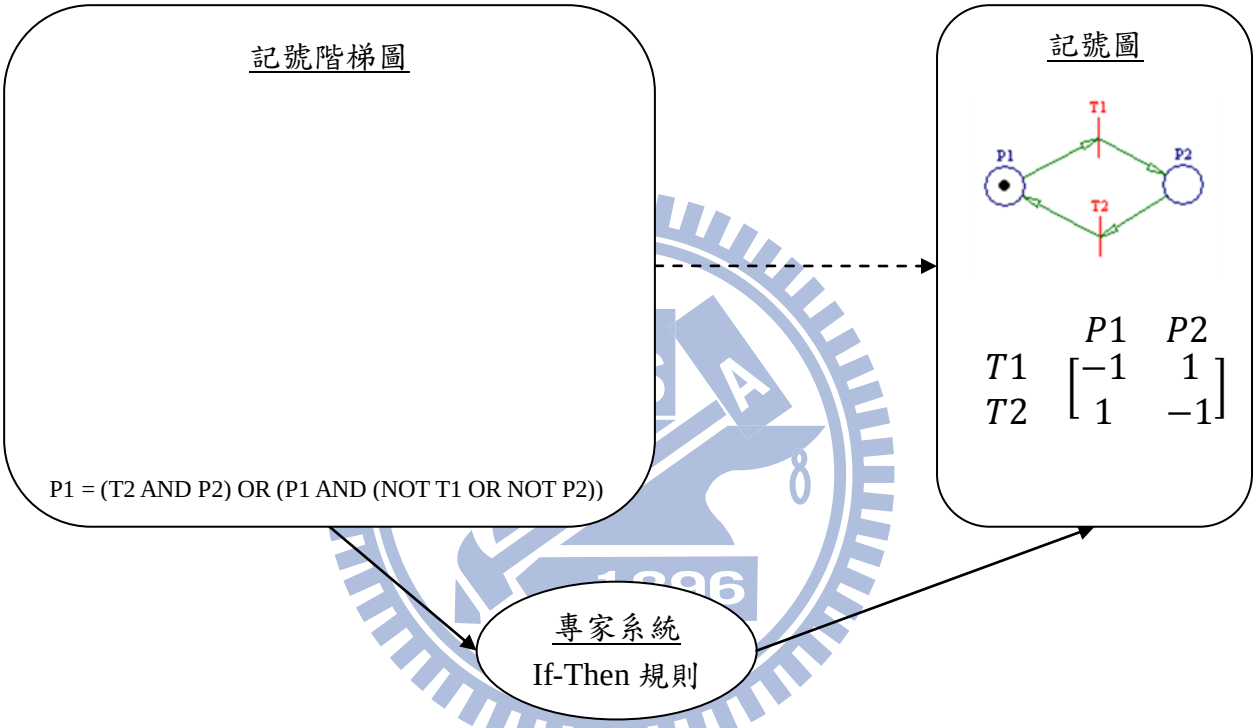
P1 = (T2 AND P2) OR (P1 AND (NOT T1 OR NOT P2))

圖 3.13　記號階梯圖及其布林邏輯式

$$P1 = (T2 \cdot P2) + (P1 \cdot (\overline{T1} + \overline{P2}))$$

圖 3.14　P1 依照布林邏輯式寫出的邏輯式

圖 3.12 是從記號圖的規則矩陣為出發點，依照其數學特性寫出的邏輯判斷式；而圖 3.14 是從記號階梯圖為出發點，依照其邏輯寫出邏輯判斷式。由圖 3.12 與圖 3.14 中可以看出，從記號圖與記號階梯圖兩種不同出發點，寫出的邏輯判斷式是相同的。因此不管從物理意義面，或從數學意義面，皆可以判斷這兩種圖形是等效的。圖 3.15 為這兩種圖形的等構關係圖。



$$P1 = (T2 \cdot P2) + (P1 \cdot (\overline{T1} + \overline{P2}))$$

$$P1 = L+U = (T2 \cdot P2) + \left(P1 \cdot (\overline{T1 \cdot P2})\right) = (T2 \cdot P2) + (P1 \cdot (\overline{T1} + \overline{P2}))$$

圖 3.15　記號階梯圖與記號圖的等構關係

3.4　可加註語言樹轉換法的分析

由於可加註語言樹轉換法轉換的為兩種圖形的可加註語言格式，因此轉換法必須從兩者的樹狀結構進行分析。3.4.1 節介紹可加註語言樹轉換法的架構、3.4.2 節說明可加註語言樹轉換法的三階段流程，以及每階段的說明及物件群介紹。

3.4.1　可加註語言樹轉換法的架構

圖 3.16 為本研究轉換過程的示意圖，記號階梯圖完成後，將 PLCopenXML 格式檔案輸入後，經由可加註語言樹轉換法，轉換成記號圖之 PNML 格式檔案，最後是轉換後的記號圖。

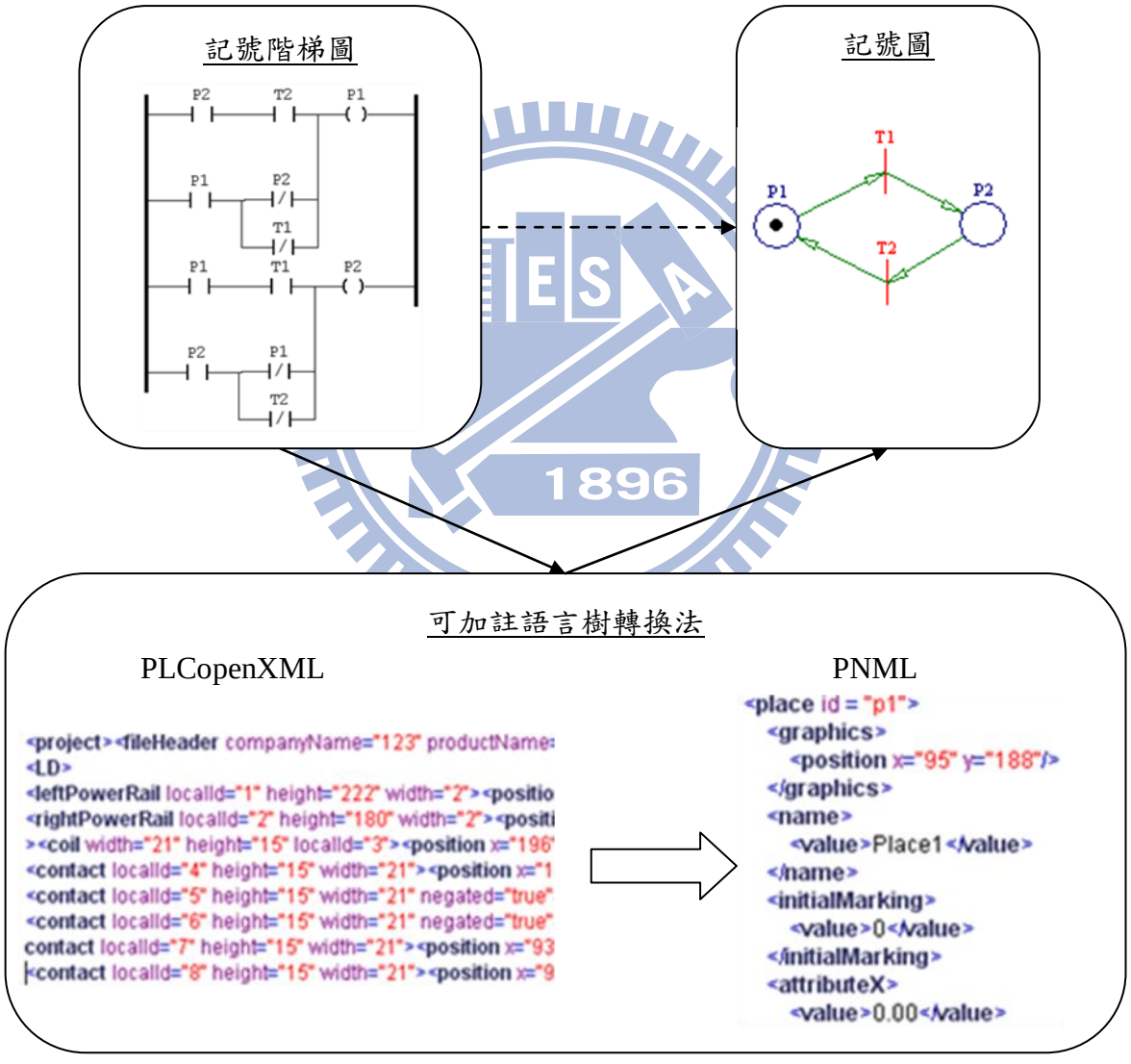


圖 3.16　轉換法示意圖

3.4.2 可加註語言樹轉換法的三階段流程與說明

PNML 及 PLCopenXML 皆可以樹狀圖表示。當兩者轉換為樹狀結構時，可以找出兩者對應的元素。本小節主要在介紹可加註語言樹轉換法的轉換過程部分，至於使用者介面部分與可加註語言文件的讀取(XML Path Language, XPath)[32]部分，會在第四章做說明，不在本小節討論範圍。

圖 3.17 為本論文可加註語言樹轉換法的設計流程，首先決定記號階梯圖及記號圖的物件，再將可加註語言樹轉換法的轉換過程分為三個主要階段，分別為階段 I、階段 II、階段 III。階段 I 主要是將記號階梯圖的可加註語言樹轉換為階梯圖的物件，階段 II 是將記號階梯圖的物件利用可加註語言樹轉換法轉換為記號圖的物件，最後階段 III 是將記號圖的物件轉換回記號圖的可加註語言樹。每個流程的對照表格會在以下說明。

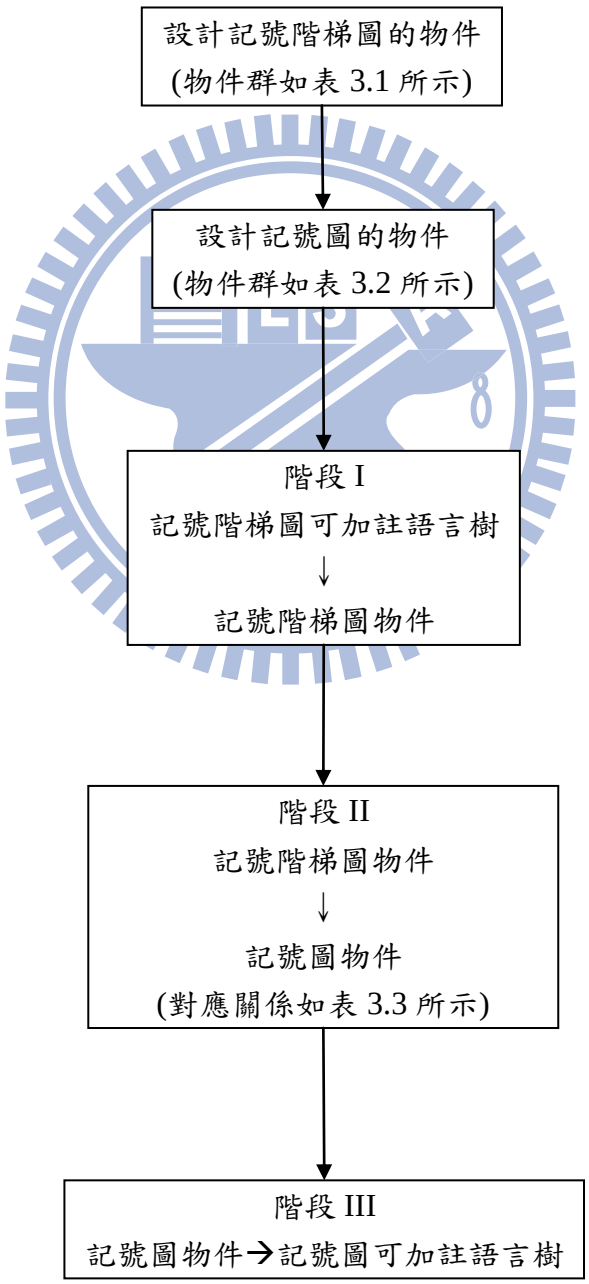


圖 3.17　可加註語言樹轉換法的設計流程

圖 3.18 為可加註語言樹轉換法三階段與程式套件的對應關係。每個階段對應到的物件會在以下說明，而每個套件的設計方式會在本論文第四章做說明。階段 I 利用 xmltool 套件將記號階梯圖的可加註語言樹轉換為記號階梯圖物件，轉換後的記號階梯圖物件利用 ladderdiagram 套件做暫存；階段 II 利用暫存的記號階梯圖物件搭配 transformtools 套件做樹狀對應轉換，產生出記號圖的物件利用 petrinet 套件做暫存；最後階段 III 利用 fileGenerator 套件將記號圖的物件轉換為記號圖的可加註語言樹。

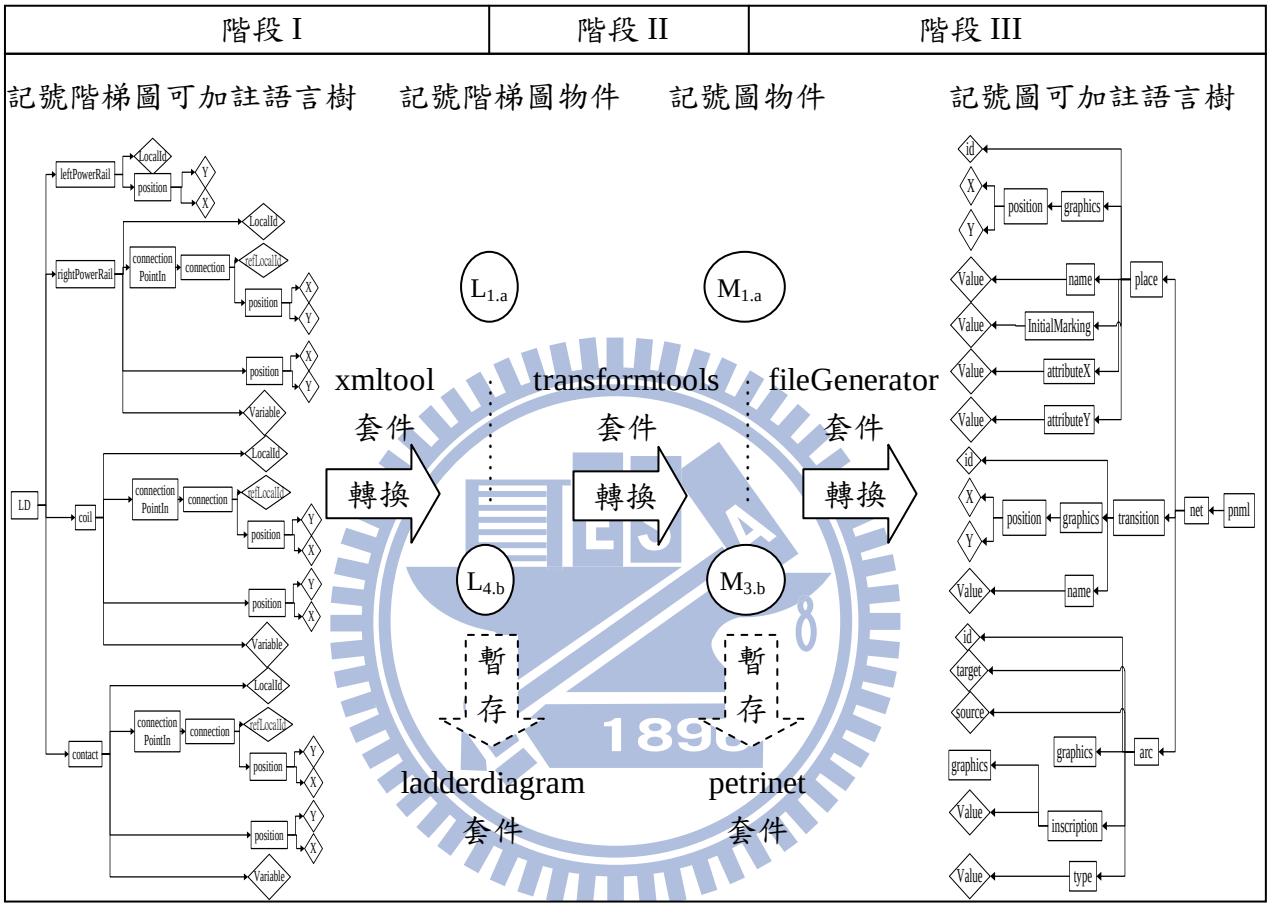


圖 3.18　可加註語言樹轉換法三階段與程式套件的對應關係

表 3.1 為記號階梯圖的物件群，根據第 2.1.2 節，記號階梯圖基本元件可分為四大類，分別為 coil、contact、rightRail 及 leftRail。階段 I 會利用 xmltool 套件將記號階梯圖可加註語言樹與這些物件群做對應。表 3.1 將記號階梯圖的四大物件分別命名為 L1. coil、L2. contact、L3. rightRail 及 L4. leftRail，物件群的數量會依照記號階梯圖的梯級數量不同而有所不同。

其中每個物件都有其對應到的屬性，雖然在擷取記號階梯圖物件時，會將所有屬性都擷取出來，但在轉換過程中只會用到其中某些屬性。例如：L1.1 的 coil_Position 是 coil 的座標位置，會在繪製使用者介面時利用到，但在轉換過程中不會用到 L1.1 這個物件；但 L1.5 的 coil_ID 是 coil 的 ID，在轉換過程中主要是利用 ID 來辨別不同 coil，因此轉換過程中會利用到 L1.5 這個物件。

表 3.1　記號階梯圖的物件群

物件	物件屬性	物件意義
L1. coil	1. Position	coil 的座標
	2. In	coil 的輸入點座標
	3. Out	coil 的輸出點座標
	4. Name	coil 名稱
	5. ID	coil 的 ID
	6. InConnectionID	coil 的輸入點 ID
L2. contact	1. Position	contact 的座標
	2. In	contact 的輸入點座標
	3. Out	contact 的輸出點座標
	4. Name	contact 名稱
	5. ID	contact 的 ID
	6. InConnectionID	contact 的輸入點 ID
	7. Negated	contact 是否為常閉開關
L3. rightRail	1. Position	rightRail 的座標
	2. Height	rightRail 的高度
	3. Width	rightRail 的寬度
	4. InConnectionID	rightRail 的輸入點 ID
	5. ID	rightRail 的 ID
L4. leftRail	1. Position	leftRail 的座標
	2. Height	leftRail 的高度
	3. Width	leftRail 的寬度
	4. ID	leftRail 的 ID

表 3.2 為記號圖的物件群，根據第 2.2.1 節，記號圖的基本元件可分為三大類，分別為 place、transition 及 arc。階段 III 會利用 fileGenerator 套件將這些物件群對應回記號圖可加註語言樹。表 3.2 將記號圖的基本三大物件分別命名為 M1. Place、M2. Transition、M3. Arc。

物件群雖然數量會因為記號階梯圖的繁簡程度有所不同，但分類還是以三大物件為主。與記號階梯圖物件相同，記號圖的每個物件也有其對應的屬性，在轉換過程中也只會用到某些屬性。例如：M1.2 的 Place_Point 是 Place 的座標位置，主要是用在使用者介面這部分，在轉換過程中不會用到 M1.2 這個物件；M1.4 的 Place_ID 是 Place 的 ID，在轉換過程中每產生一個 Place 會給予一個 ID，所以轉換過程需要用到 M1.4 這個物件。

表 3.2　記號圖的物件群

物件	物件屬性	物件意義
M1. Place	1. Name	Place 的名稱
	2. Point	Place 的座標
	3. initialMarking	Place 的初始標記
	4. ID	Place 的 ID
M2. Transition	1. Name	Transition 的名稱
	2. ID	Transition 的 ID
	3. Point	Transition 的座標
	4. inputPlace	Transition 的輸入 place 陣列
	5. outputPlace	Transition 的輸入 place 陣列
M3. Arc	1. Source	Arc 的來源座標
	2. Target	Arc 的目標座標
	3. source	Arc 的來源名稱
	4. target	Arc 的目標名稱
	5. ID	Arc 的 ID

圖 3.18 為階段 II 的物件對應關係圖，這個階段的轉換法會利用到 transformtools 套件中的物件，transformtools 套件會從 xmltool 套件轉換後的記號階梯圖物件，選出可加註語言樹轉換法會用到的物件，再轉換為此套件中的物件，以供轉換過程利用。表 3.3 主要在說明階段 II 記號階梯圖物件轉換為記號圖物件的詳細對應關係。

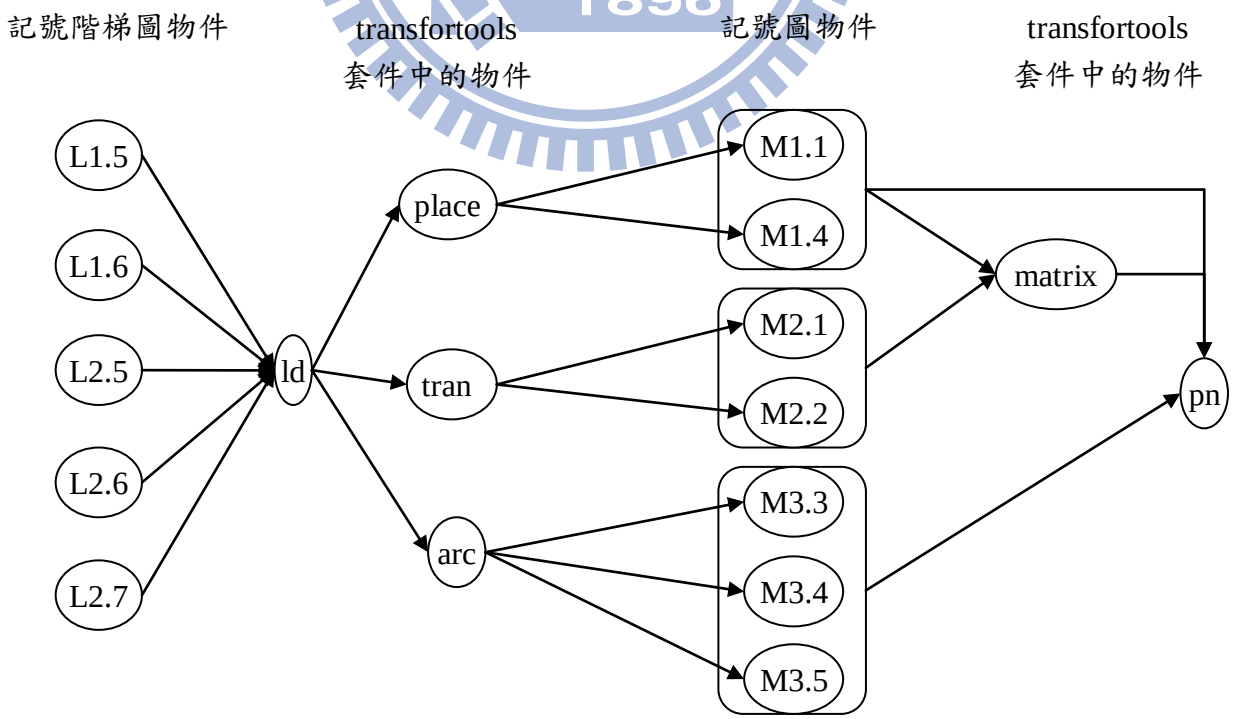


圖 3.19　階段 II 的物件對應關係

表 3.3　階段 II 記號階梯圖物件轉換為記號圖物件的詳細對應關係

記號階梯圖物件	對應	記號圖物件
L1.5 coil_ID	→	M1.1 Place_Name
		M1.4 Place_ID
L1.6 coil_InConnectionID	→	M2.1 Transition_Name
		M2.2 Transition_ID
		M3.3 Arc_source
		M3.4 Arc_target
		M3.5 Arc_ID
L2.5 contact_ID	→	M2.1 Transition_Name
		M2.2 Transition_ID
L2.6 contact_InConnectionID	→	M2.1 Transition_Name
		M2.2 Transition_ID
		M3.3 Arc_source
		M3.4 Arc_target
		M3.5 Arc_ID
L2.7 contact_Negated	→	M1.4 Place_ID
		M2.2 Transition_ID

可加註語言樹轉換程式中利用物件來作對應關係，而轉換程式中的各種方法(Method)則用來處理轉換作業。可加註語言轉換程式共用了 36 個物件及 34 個方法，如表 3.4 及表 3.5 所示。

表 3.4　轉換程式中的方法名稱及所屬套件

轉換程式的套件名稱	轉換程式中的方法名稱
xmltools	paserCoils()
	paserContacts()
	paserLeftPowerRail()
	paserRightPowerRail()
	getCoilname()
	getCoilID()
	getCoilInConnectionID()
	getCoilIDPosition()
	getCoilIn()
	getCoilOut()
	getContactname()
	getContactID()
	getContactInConnectionID()
	getContactIDPosition()

表 3.5　轉換程式中的方法名稱及所屬套件

轉換程式的套件名稱	轉換程式中的方法名稱
xmltools	paserCoils()
	paserContacts()
	paserLeftPowerRail()
	paserRightPowerRail()
	getCoilname()
	getCoilID()
	getCoilInConnectionID()
	getCoilIDPosition()
	getCoilIn()
	getCoilOut()
	getContactname()
	getContactID()
	getContactInConnectionID()
	getContactIDPosition()
	getContactIn()
	getContactOut()
	getContactname()
	getRightPowerRailID()
	getRightPowerRailHeight()
	getRightPowerRailWidth()
	getRightPowerRailPosition()
	getRightPowerRailID()
	getRightPowerRailHeight()
	getRightPowerRailWidth()
	getContactNegated()
transformtools	transformCoil()
	transformContact()
	setupPetrinet()
	setupMatrix()
	getPetrinet()
fileGenerator	outputPlaces()
	outputTrans()
	ouputArcs()
	output()

3.5 偵查方程式的轉換

裴氏圖的偵查方程式為利用不變量計算出來的方程式。由於偵查方程式為含有浮標數量的數學式，而不為布林邏輯式；因此偵查方程式的數值不可視為布林邏輯之真(True)或假(False)。一條偵查方程式可能轉換為許多條邏輯式，而轉換後邏輯式的多寡，必須由偵查方程式中包含的暫存點數及浮標數量決定。

以圖 3.20 的裴氏圖為例，表 3.6 為圖 3.20 對應的偵查方程式。第一條偵查方程式中，P6 + P10 = 1 的意義為，P6 及 P10 同時只能有一個暫存點有浮標。第二條偵查方程式表示，P2 及 P10 同時只能有一個浮標。第三條偵查法方程式表示，P7 及 P11 擁有的浮標總數會等於 P10 的浮標數。

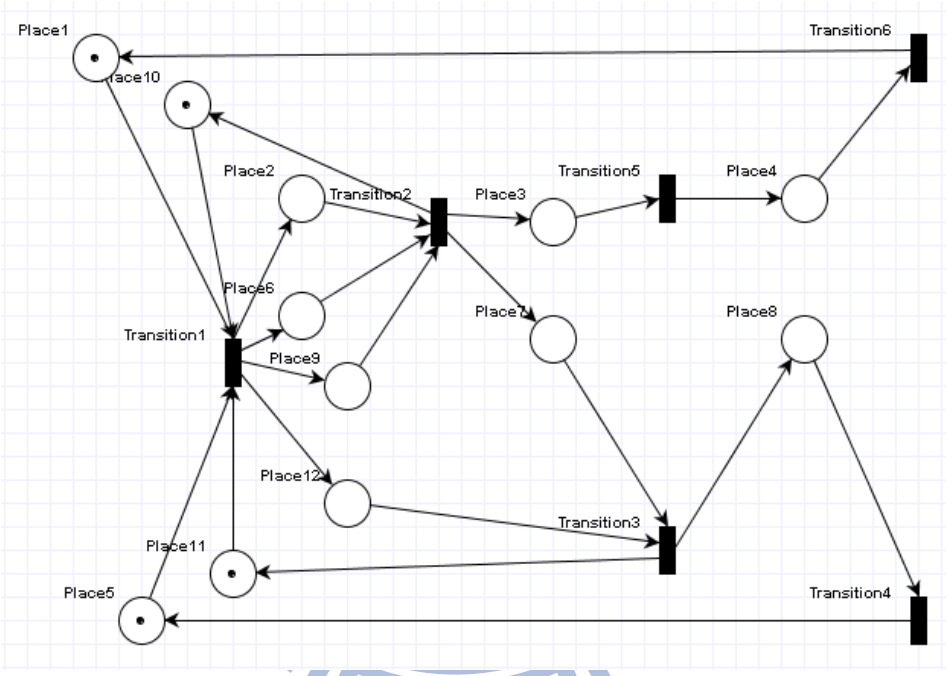


圖 3.20 裴氏圖

表 3.6 圖 3.20 之偵查法則

編號	偵查方程式
1	P6+P10 = 1
2	P2+P10 = 1
3	P7+P11-P10 = 0

表 3.7 為表 3.6 偵查方程式的對應條件式與邏輯式。由於此裴氏圖中的每個暫存點只會包含一個浮標，所以第一條偵查方程式代表 P6 有浮標則 P10 沒有浮標、P10 有浮標則 P6 沒有浮標，因此其對應的條件式為(P6,P10) = (1,0)及(P6,P10) = (0,1)，轉換後的邏輯式為 (P6∧~P10)∨(~P6∧P10) = 1；第二條偵查方程式的條件式與邏輯式同理可得；第三條偵查方程式先經過移項後得 P7+P11=P10，代表 P10 沒有浮標時，P7 及 P11 皆不會有浮標，P10 有浮標時，P7 或 P11 其中一個暫存點會有浮標，對應的條件式為(P7, P10, P11) = (0,0,0)、(P7, P10, P11) = (1,1,0)以及(P7, P10, P11) = (0,1,1)，與此條件式對應的邏輯式為

(~P7∧~P10∧~P11)∨(P7∧P10∧~P11)∨(~P7∧P10∧P11) = 1。

表 3.7　表 3.6 之對應條件式與邏輯式

編號	偵查方程式	對應條件式	邏輯式
1	P6+P10 = 1	(P6,P10) = (1,0)	(P6∧~P10)∨(~P6∧P10) = 1
		(P6,P10) = (0,1)	
2	P2+P10 = 1	(P2,P10) = (1,0)	(P2∧~P10)∨(~P2∧P10) = 1
		(P2,P10) = (0,1)	
3	P7+P11-P10 = 0 →P7+P11 = P10	(P7,P10,P11) = (0,0,0)	(~P7∧~P10∧~P11) ∨(P7∧P10∧~P11) ∨(~P7∧P10∧P11) = 1
		(P7,P10,P11) = (1,1,0)	
		(P7,P10,P11) = (0,1,1)	

當偵查方程式轉換為邏輯式後，便可以階梯圖的型式繪出。圖 3.21 為利用表 3.7 中邏輯式繪成之階梯圖。其中每條偵查方程式控制一個輸出線圈，而輸出線圈代表狀態燈號，燈亮表示此系統狀態正常。若其中一個燈熄滅，代表系統出錯。此邏輯式，表示由 P6 及 ~P10 的串聯電路與~P6 及 P10 的串聯電路，共同並聯用以控制第一個輸出線圈「MonitorRule_1」，以下類推。

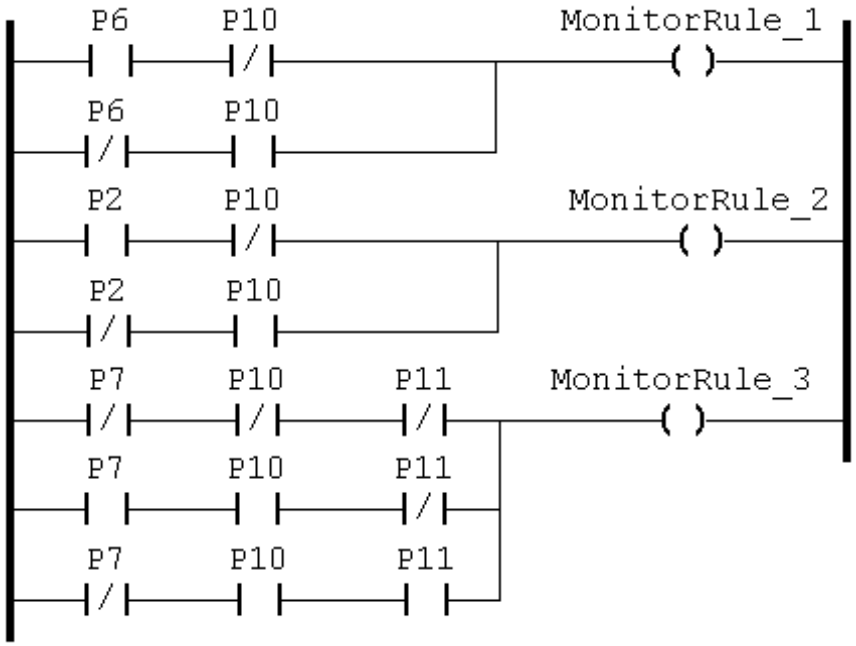


圖 3.21　利用表 3.7 中邏輯式繪出之階梯圖

第四章　可加註語言樹轉換法程式的實作

本章說明可加註語言樹轉換法物件導向的過程與程式碼的撰寫步驟及程式開發的程序與架構。4.1 節『可加註語言樹轉換法程式的需求分析』根據程式使用者的需求分析程式需具備的功能，4.2 節『可加註語言樹轉換法程式的設計實作』說明程式中介面與套件的設計。

可加註語言樹轉換法程式實作的流程大致可分成四階段，第一階段程式需求分析、第二階段程式設計實作、第三階段程式撰寫、第四階段程式驗證，開發程序如圖 4.1 所示。

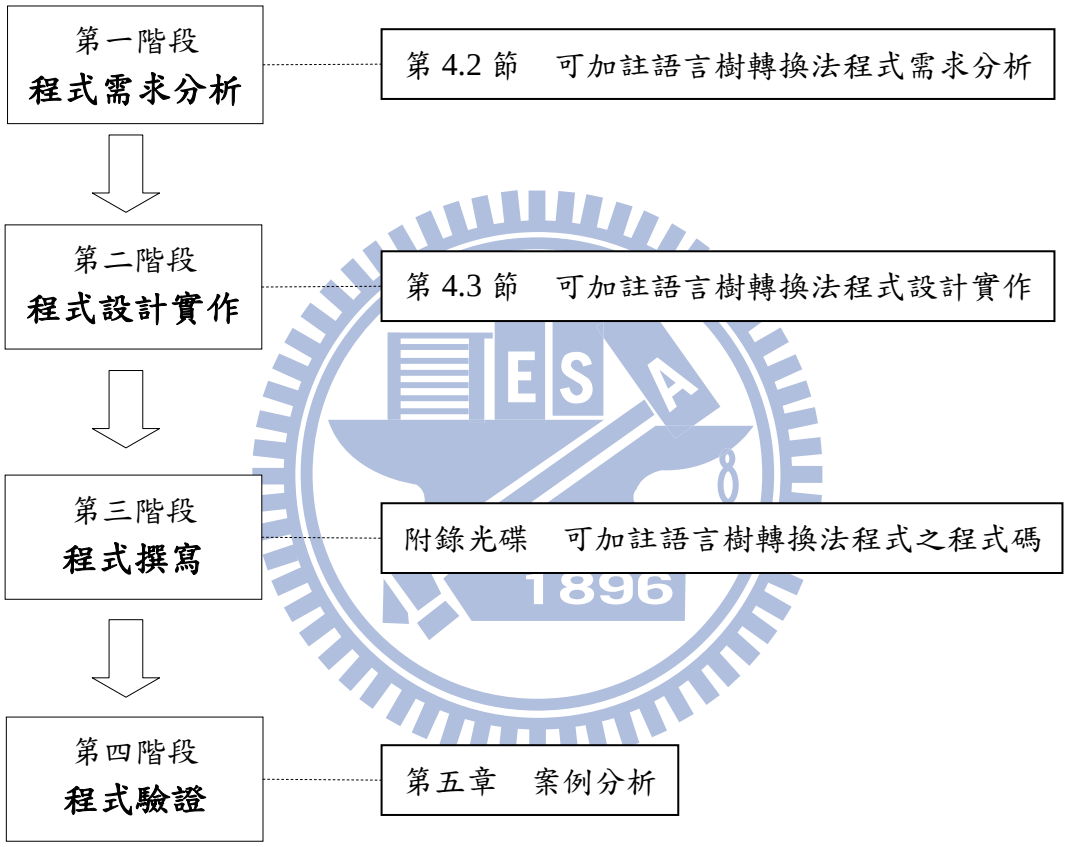


圖 4.1　可加註語言樹轉換法程式開發程序

第一階段「程式需求分析」依照程式使用者的需求，規劃轉換程式所需具備的功能，此部分於第 4.1 節中說明；第二階段「程式設計實作」依據第一階段需求分析後規劃的功能，設計程式所需的套件及介面，此部分於第 4.2 節中說明；第三階段「程式撰寫」利用 Java 語言撰寫程式，將設計後的套件以程式實現，程式碼於附錄光碟中說明；第四階段「程式驗證」，會在第五章中以液體加熱系統及灌模系統做此轉換程式的驗證。

4.1 可加註語言樹轉換法程式的需求分析

程式開發之初，首先要定義使用者需求，當使用者需求確定之後，便可確定程式需求。設計者可依照程式需求著手設計程式功能，程式功能設計完成再規劃程式規格，程式規格規劃完成後，等於確定程式套件的關聯，接著再進行程式套件的設計。圖 4.2 為本程式使用者需求與程式需求關係圖。

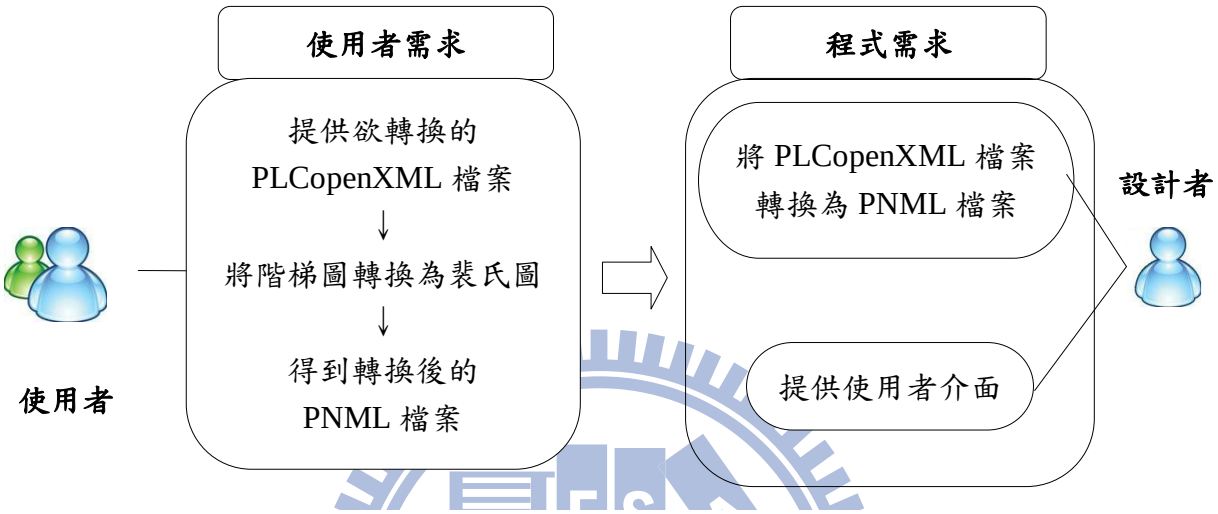


圖 4.2 使用者需求與程式需求

圖 4.2 中使用者希望提供記號階梯圖的 XML 格式檔案，並將記號階梯圖轉換為記號圖，再將轉換後的圖形輸出成 XML 格式檔案。程式設計者依照使用者的需求設計程式功能，先針對使用者提供使用者介面，再針對圖形轉換的部分提供轉換法則，接著依據程式功能決定程式規格。

此程式規格依照需求分為使用者介面、讀取檔案、分析輸入檔案、圖形元件暫存區、轉換法則及輸出檔案，如圖 4.3 所示。其中讀取檔案為讀取 PLCopenXML 檔案、分析輸入檔案為分析 PLCopenXML 檔案、暫存區分為裴式圖及階梯圖暫存區，轉換法則為記號階梯圖轉為記號圖法則、輸出檔案部分則分為輸出 PNML 檔案。

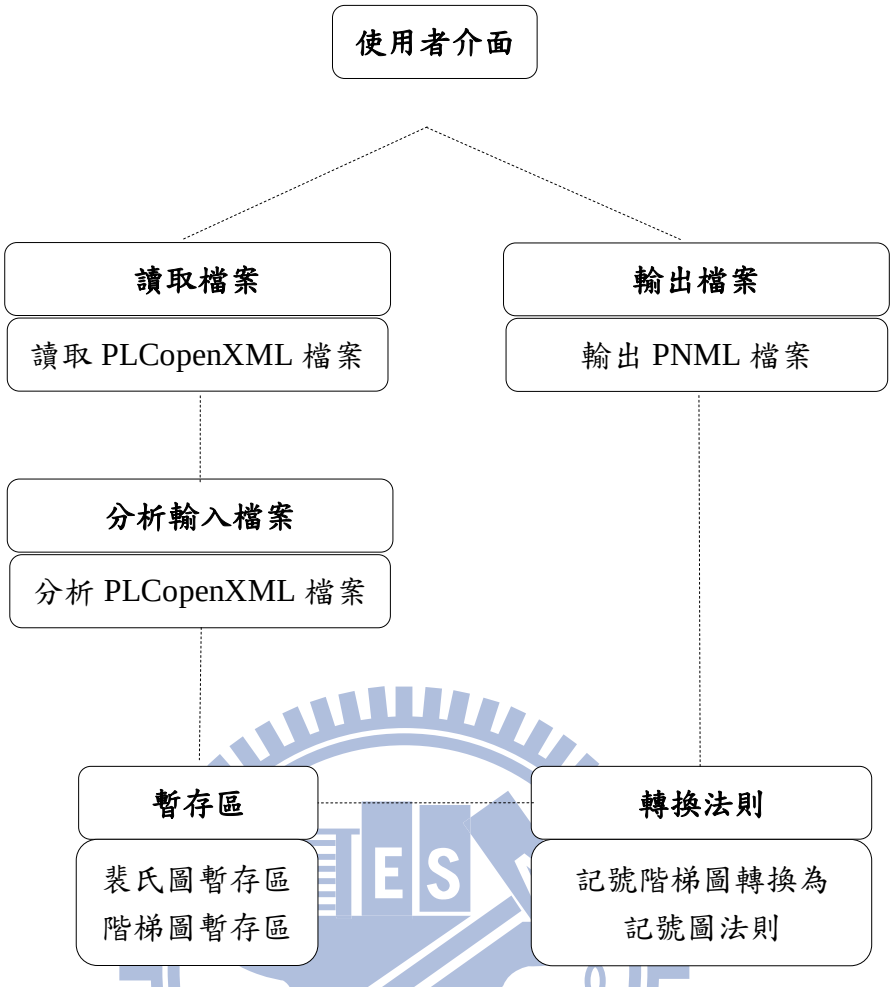

使用者介面
讀取檔案
讀取 PLCopenXML 檔案
輸出檔案
輸出 PNML 檔案
分析輸入檔案
分析 PLCopenXML 檔案
暫存區
裴氏圖暫存區
階梯圖暫存區
轉換法則
記號階梯圖轉換為
記號圖法則

圖 4.3　程式規格

4.2 可加註語言樹轉換法程式的設計實作

本節說明程式的設計與實作，其中程式設計包含了操作流程與程式套件的設計，而實作部分針對設計好的套件進行程式的撰寫。4.3.1 節『程式操作流程』說明使用者操作程式的流程，4.3.2 節『程式處理流程』說明程式處理資訊的流程，4.3.3 節『程式套件設計』說明程式套件的設計與實作部分。

4.2.1 程式操作流程

執行程式時，使用者在使用者介面可執行的操作分為載入檔案、退出檔案、選擇輸出檔案的位置及設定名稱，如圖 4.4 所示。當程式開始執行時，使用者先選擇欲載入的檔案，載入檔案可為 PNML 檔案或 PLCopenXML 檔案。若載入檔案格式正確，程式會自動進行轉換，使用者無須再進行操作。若載入檔案格式錯誤會有錯誤顯示窗格出現，此時使用者可重新選擇載入的檔案，之後可將轉換後的圖形輸出成 XML 格式檔案，此時使用者可以選擇輸出檔案的位置及設定名稱。而若使用者想要更換輸入檔案時，可選擇退出檔案，即可重新載入檔案進行轉換。

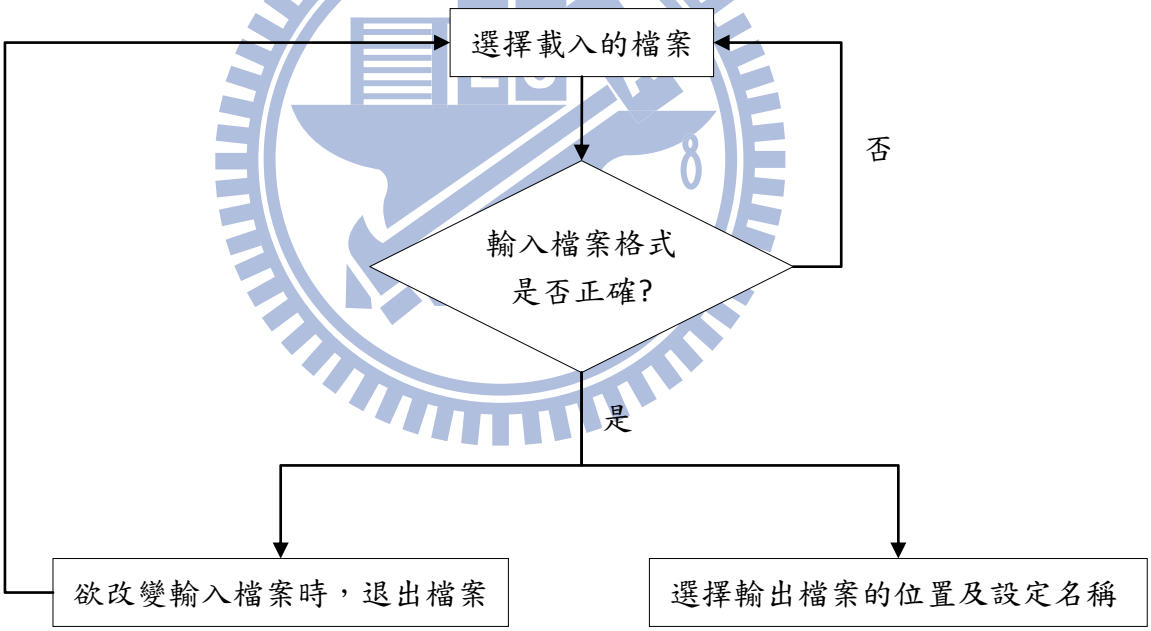


圖 4.4 程式操作流程

4.2.2 程式套件設計

在第 4.2 節中已設計好本程式的規格，針對設計好的規格，便可進行程式套件的設計。4.3.2.1 節『程式套件關聯』說明本程式所有套件之間的關係，4.2.2 節『GUI 套件設計』說明使用者介面設計，4.3.2.3 節『xmltools 套件設計』說明處理 XML 格式檔案作業，4.3.2.4 節『ladderdiagram 套件與 petrinet 套件設計』說明裴氏圖與階梯圖的暫存作業，4.3.2.5 節『transformtools 套件設計』說明裴氏圖與階梯圖轉換作業，4.3.2.6 節『fileGenerator 套件設計』說明檔案輸出作業。

4.2.2.1 程式套件關聯與程式處理流程

程式中包含了六個套件，分別處理主程式的執行作業、解析 XML 文件、提供暫存區暫存圖形元件、處理圖形的轉換作業、處理輸出 XML 文件作業、以及提供使用者操作介面，表 4.1 為程式套件及功能的對照。

表 4.1　程式套件名稱及功能

套件名稱	套件功能
MainFile	主執行程式
xmltools	解析 XML 文件
ladderdiagram	階梯圖暫存區
petrinet	裴氏圖暫存區
transformtools	圖形轉換作業
fileGenerator	輸出 XML 文件
GUI	使用者介面

程式處理流程為程式執行後，用以處理資訊的流程，其中包含了資料格式的判斷、資料分析、元件的存取、圖形的轉換以及最終輸出檔案的建立，如圖 4.5 所示。

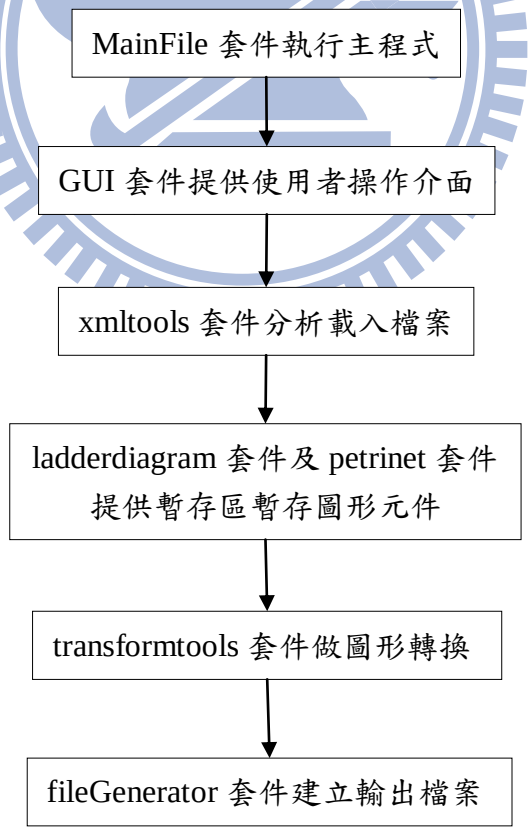


圖 4.5　程式處理流程

程式處理流程的先後順序與套件關聯有關。如圖 4.6 所示，主執行程式所在的 MainFile 套件需仰賴 GUI 套件才能工使用者使用以及做資訊的擷取，xmltools 套件需要 GUI 套件中提供的檔案才能做分析，xmltools 套件分析後的圖形元件暫存到 petrinet 套件及 ladderdiagram 套件提供的暫存區，transformtools 套件做轉換及轉換後皆需要 petrinet 套件及 ladderdiagram 套件暫存區中的暫存元件及提供的暫存區，fileGenerator 套件利用暫存區中的元件進行 XML 文件的建立，GUI 套件提供文件輸出的選項。

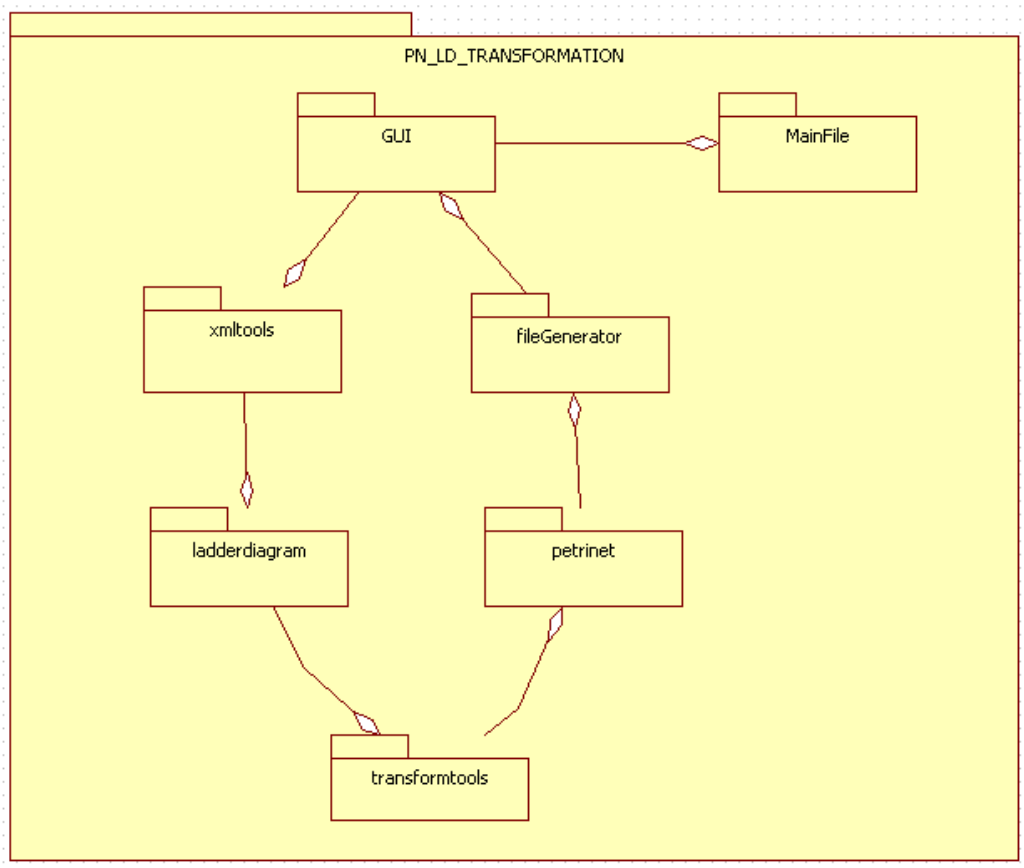


圖 4.6　程式套件關聯

4.2.2.2 GUI 套件設計

GUI 套件包含了 GUI 類別、custompane 子套件及 customdialog 子套件，其中 custompane 子套件中包含了 PNDrawPane 類別、LadderDrawPane 類別及 GraphLayout 類別，customdialog 子套件中有 MarkingNumberSettingDialog 類別，架構如圖 4.7 所示。

GUI 類別主要提供使用者操作介面，此程式操作介面分為左右兩大部分，並且提供使用者載入檔案、退出檔案、輸出檔案、圖形布局等功能。PDrawPane 類別為介面的左半部，提供裴氏圖圖形的呈現，LadderrawPane 類別為介面的右半部，提供階梯圖圖形的呈現，GraphLayout 類別提供轉換後的裴氏圖做不同布局的設定。MarkingNumberSettingDialog 類別提供使用者選擇轉換後的裴氏圖初始浮標。GUI 套件的 UML 類別圖[15]如圖 4.8 所示。

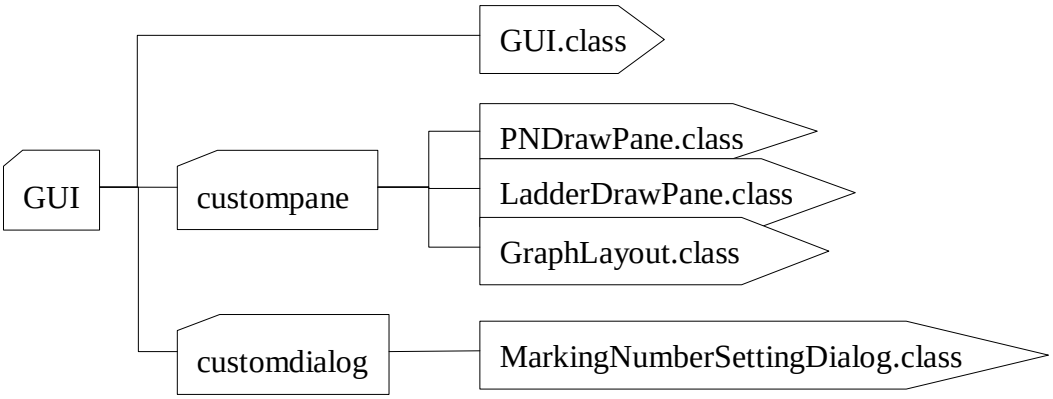


圖 4.7　GUI 套件架構

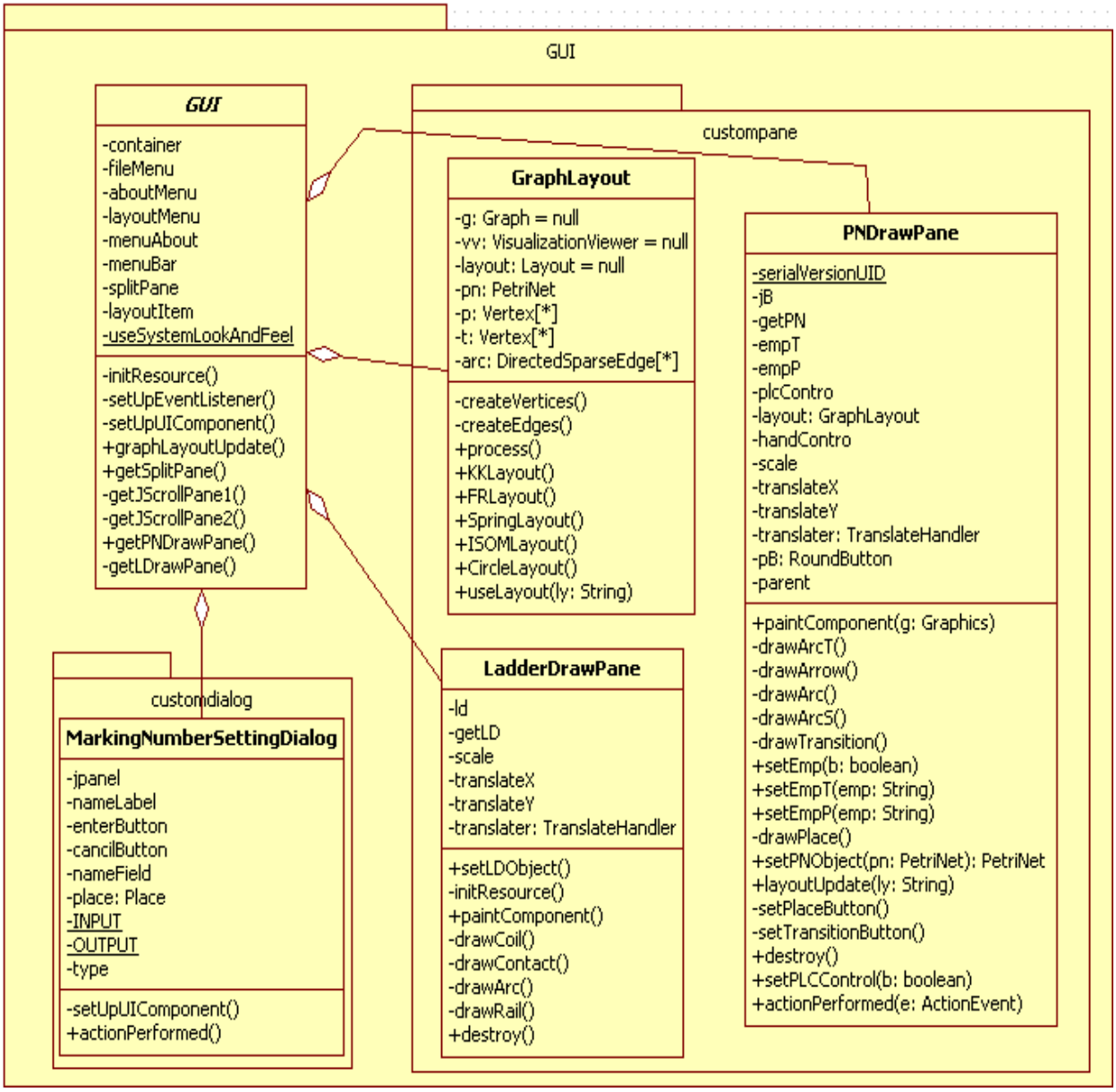


圖 4.8　GUI 套件的 UML 類別圖

4.2.2.3 xmltools 套件設計

xmltools 套件包含了 OPLCXMLpaser 類別，架構如圖 4.9 所示。OPLCXMLpaser 類別用來分析輸入的 PLCopenXML 檔案以及擷取輸入檔案中必要的元件，暫存至 ladderdiagram 套件提供的暫存區， OPLCXML 類別處理資訊的流程如圖 4.10 所示。xmltools 套件的類別圖如圖 4.11 所示。

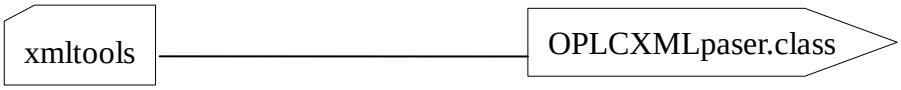


圖 4.9　xmltools 套件架構

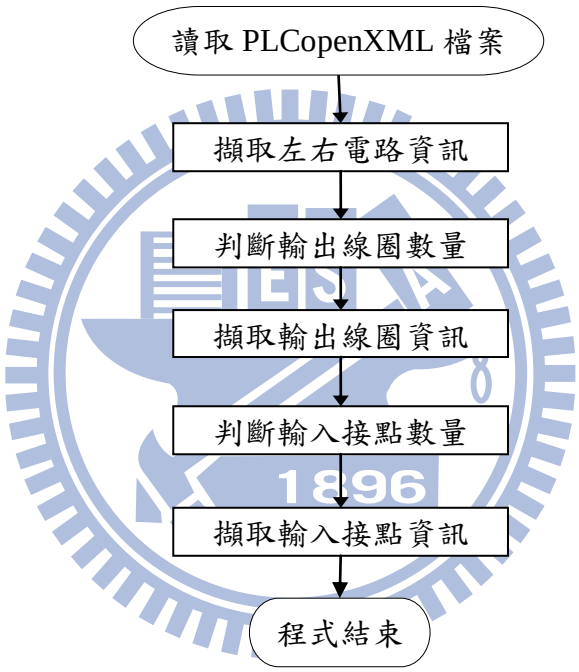


圖 4.10　OPLCXML 類別之流程圖

圖 4.11　xmltools 套件類別圖

4.2.2.4 ladderdiagram 套件與 petrinet 套件設計

ladderdiagram 套件中的 component 子套件包含了 LadderDiagram 類別、RightPowerRail 類別、LeftPowerRail 類別、Contact 類別及 Coil 類別，架構如圖 4.12 所示。RightPowerRail 類別、LeftPowerRail 類別、Contact 類別及 Coil 類別分別提供暫存區儲存來自 xmltools 套件分析後的階梯圖元件。

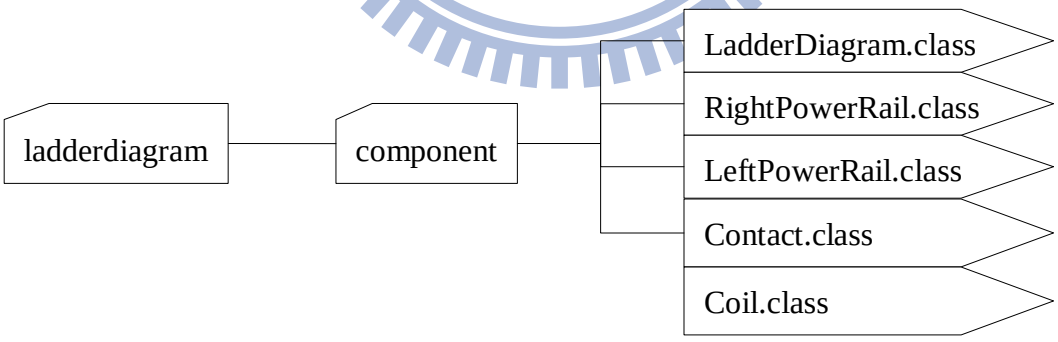


圖 4.12　ladderdiagram 套件架構

所有暫存區的元件需先整合到 LadderDiagram 類別，才能供其他類別使用。xmltools 套件分析後的元件可供 transformtools 套件做轉換處理以及供 LadderDrawPane 套件做繪圖作業。ladderdiagram 套件中各類別的處理流程如圖 4.13 所示。ladderdiagram 套件的類別圖如圖 4.14 所示。

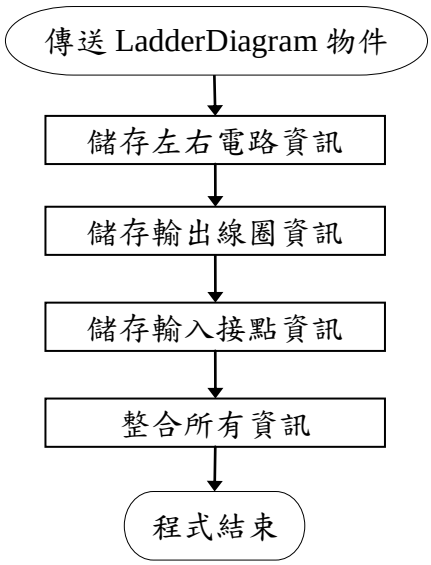


圖 4.13　ladderdiagram 套件處理流程

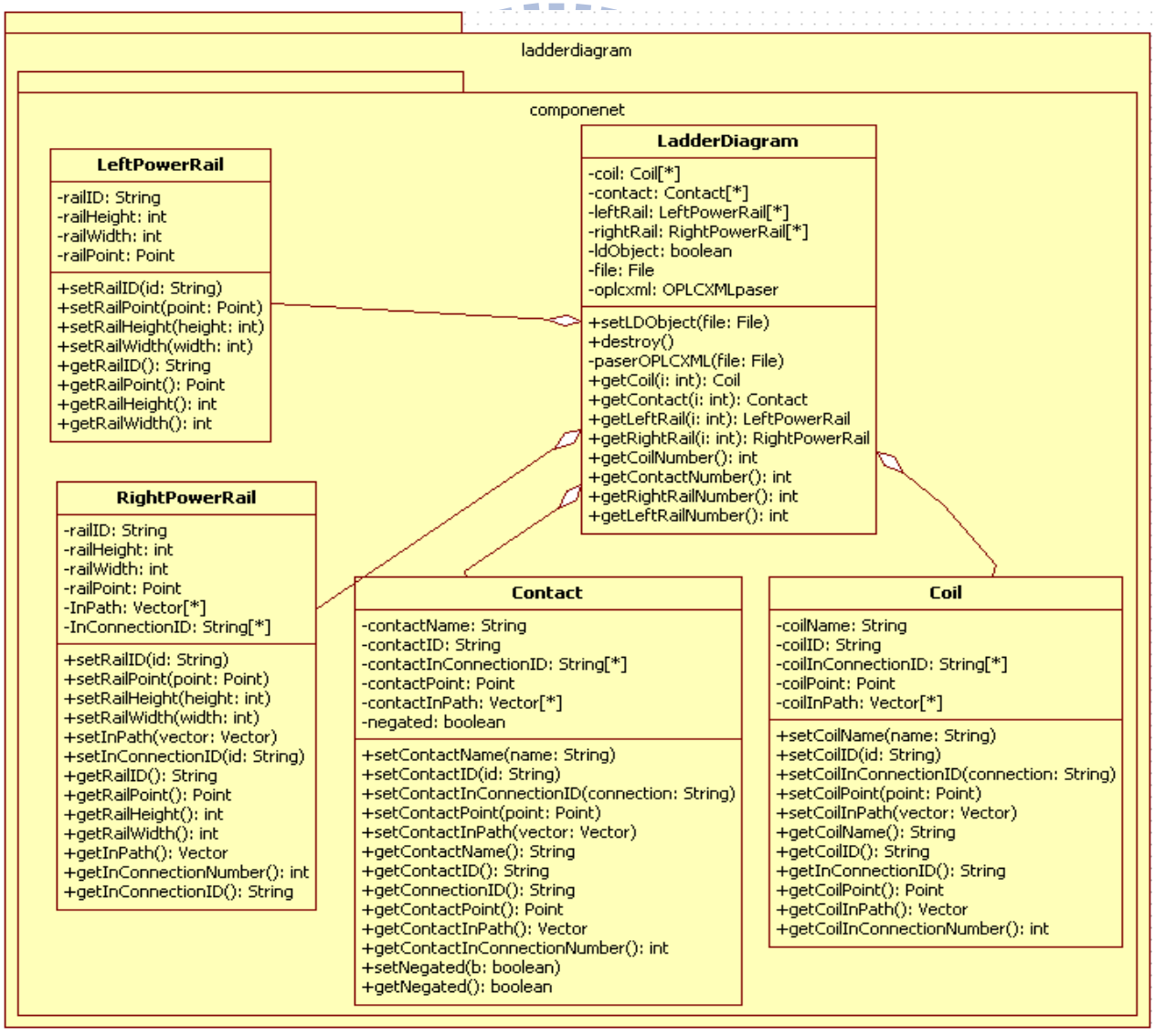


圖 4.14　ladderdiagram 套件類別圖

petrinet 套件中的 component 子套件包含了 PetriNet 類別、Place 類別、Transition 類別、Arc 類別及 VarMember 類別，架構如圖 4.15 所示。其中 Place 類別、Transition 類別、Arc 類別及 VarMember 類別提供暫存區儲存 transformtools 套件轉換後的裴氏圖元件。

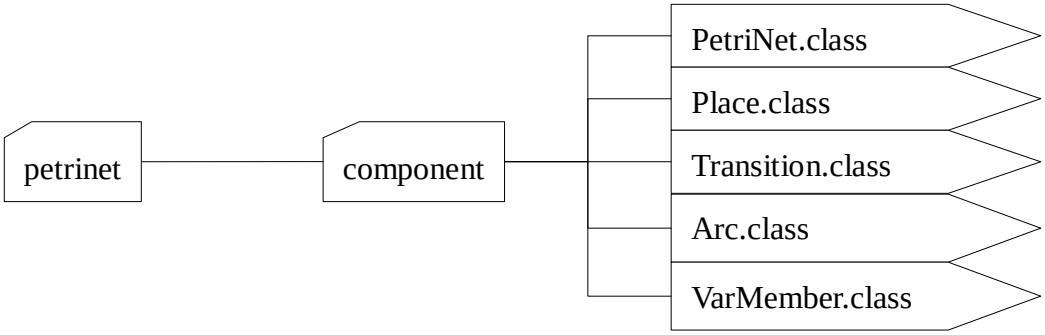


圖 4.15　petrinet 套件架構

所有暫存區的元件需先整合到 PetriNet 類別，才能供其他類別使用。transformtools 套件轉換後的元件可供 fileGenerator 套件建立 XML 文件，或供 PNDrawPane 套件做繪圖作業，petrinet 套件中各類別的處理流程如圖 4.16 所示。petrinet 套件的類別圖如圖 4.17 所示。

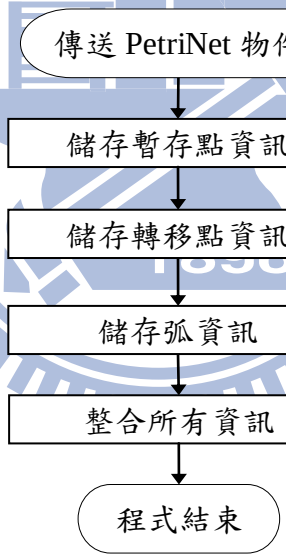


圖 4.16　petrinet 套件處理流程

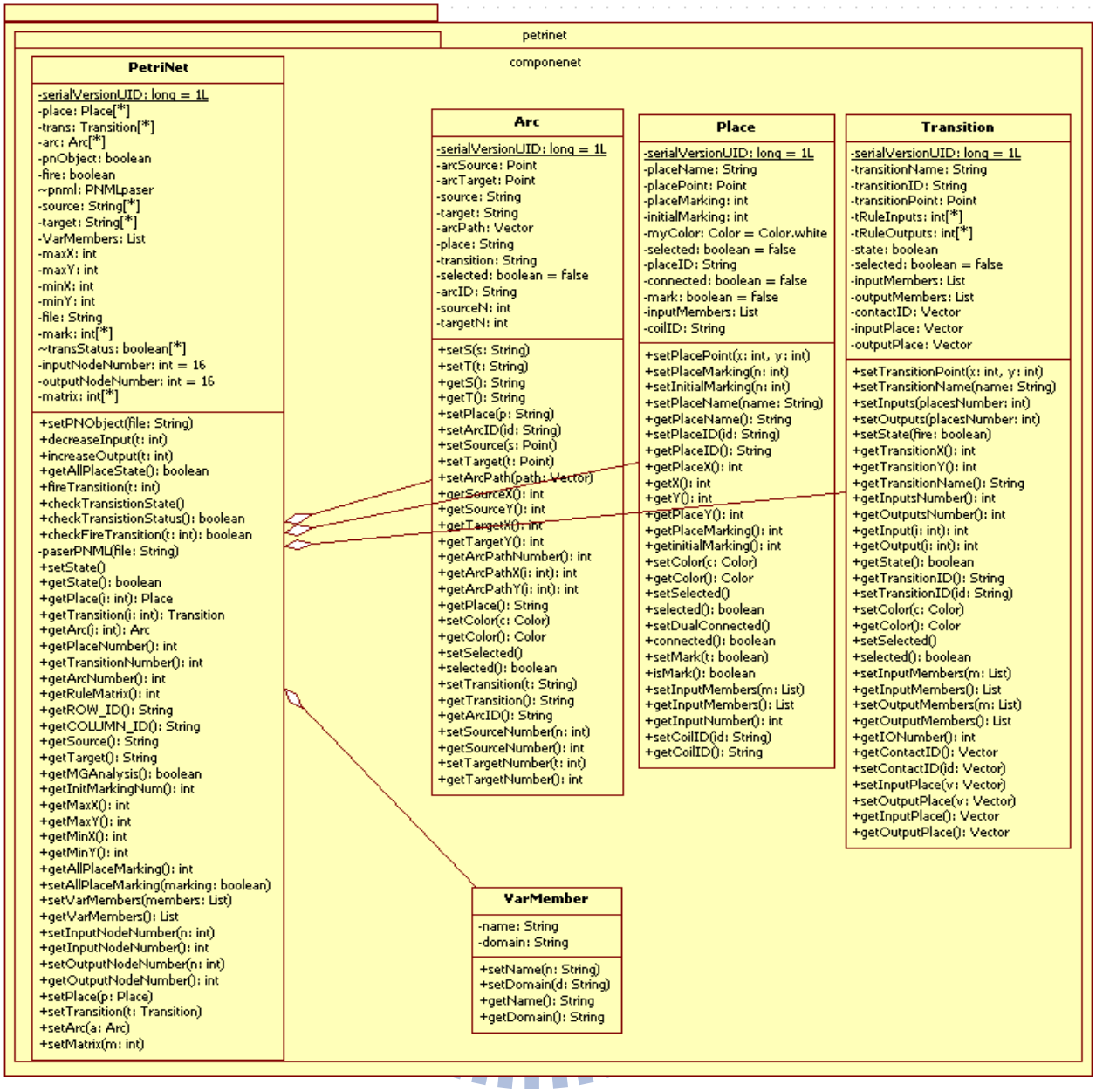


圖 4.17　petrinet 套件類別圖

4.2.2.5 transformtools 套件設計

transformtools 套件中包含了 LD2PN 類別，LD2PN 類別處理階梯圖轉為裴氏圖的作業架構如圖 4.18 所示。

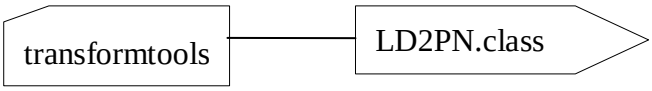


圖 4.18　transformtools 套件架構

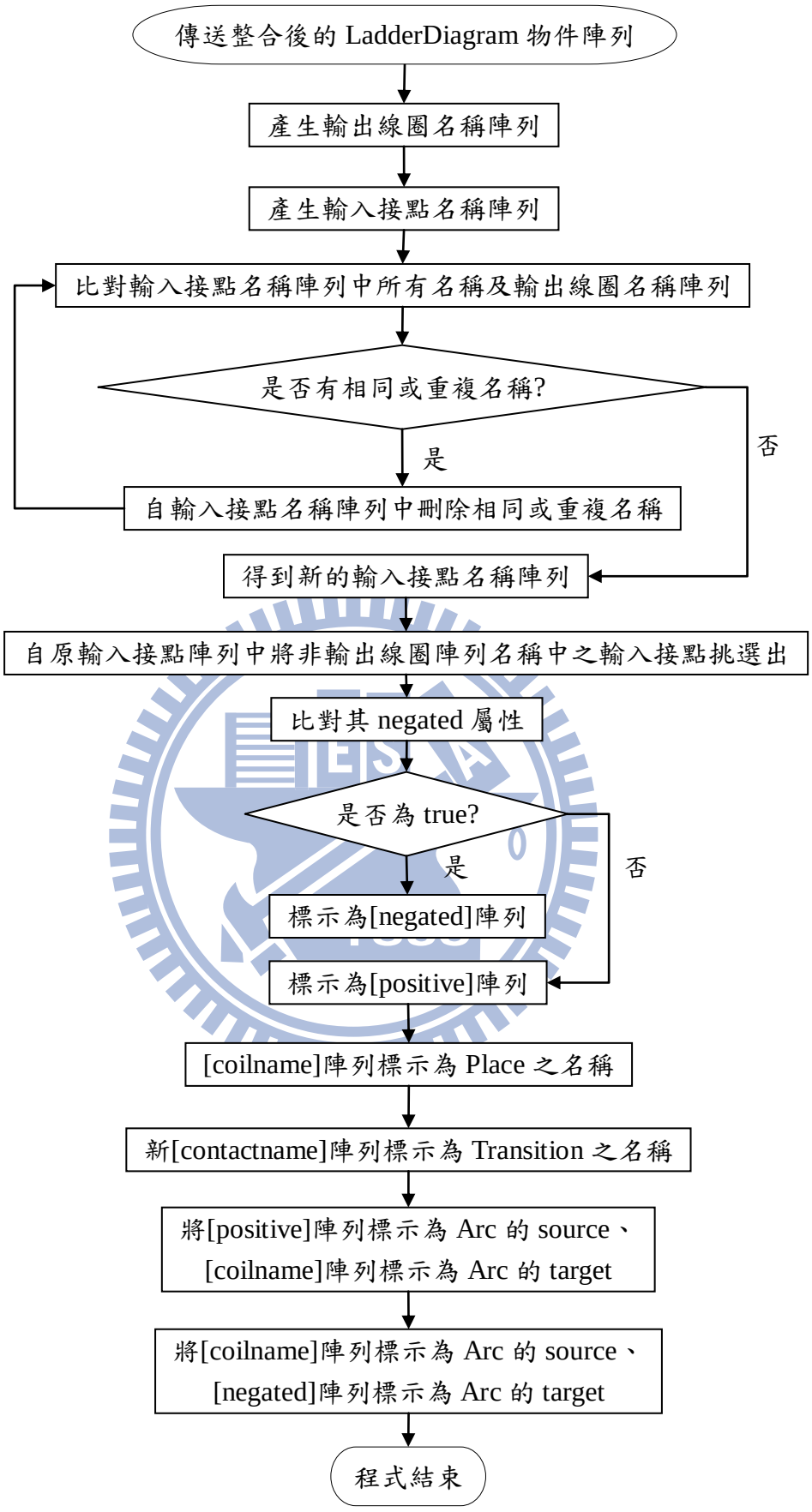


圖 4.19　LD2PN 類別處理流程

圖 4.19 為 LD2PN 類別的流程圖，傳送整合後的 LadderDiagram 物件陣列，先將所有輸出線圈與輸入接點標示為輸出線圈名稱陣列及原輸入接點名稱陣列，將原輸入接點名稱陣列扣除輸出線圈名稱陣列中重複出現的名稱，便可得到新輸入接點名稱陣列，再將原輸入接點名稱陣列中扣除輸出線圈名稱陣列名稱之輸入接點做分類，屬性 negated 為 true 者標示為[negated]陣列、否則標示為[positive]陣列。

將輸出線圈名稱陣列標示為暫存點，新輸入接點名稱陣列標示為轉移點，將[positive]陣列標示為弧的 Source、輸出線圈名稱陣列標示為 Arc 的 Target，將輸出線圈名稱陣列標示為 Arc 的 Source、[negated]陣列標示為 Arc 的 Target。圖 4.20 為 transformtools 套件類別圖。

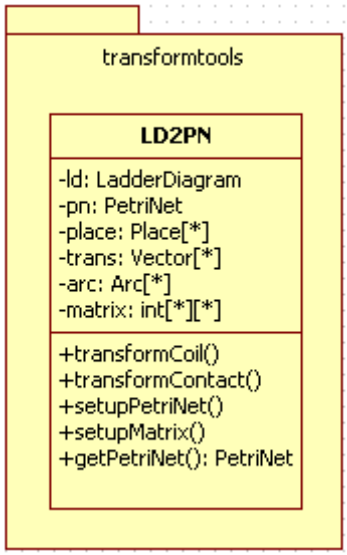


圖 4.20　transformtools 套件類別圖

4.2.2.6 fileGenerator 套件設計

fileGenerator 套件包含了 outputPNML 類別，架構如圖 4.21 所示。outputPNML 類別輸出 PNML 檔案，outputPLCopenXML 類別則輸出 PLCopenXML 檔案。，outputPNML 類別及 outputPLCopenXML 類別處理資訊的流程如圖 4.22 所示。xmltools 套件的類別圖如圖 4.23 所示。

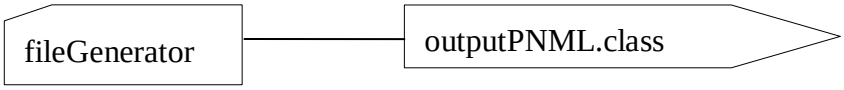


圖 4.21　fileGenerator 套件架構

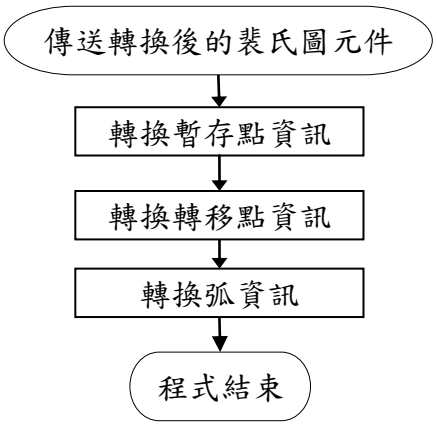


圖 4.22　outputPNML 類別及 outputPLCopenXML 類別處理流程

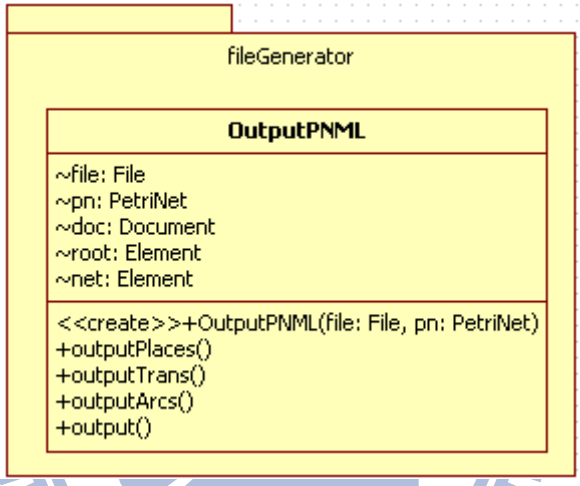


圖 4.23　fileGenerator 套件類別圖

第五章　案例分析

本章主要以兩大案例驗證轉換程式的正確性。5.1 節『液體加熱系統案例分析』說明利用階梯圖繪圖軟體繪出液體加熱系統的記號階梯圖並轉換為記號圖，再利用裴氏圖繪圖軟體做驗證的過程。5.2 節『灌模系統案例分析』說明灌模系統的記號階梯圖轉換至記號圖並利用裴氏圖繪圖軟體做驗證的過程。

5.1　液體加熱系統案例分析

本節案例利用液體加熱系統[4]做分析，5.1.1 節為液體加熱系統案例分析的操作流程及說明，5.1.2 節介紹液體加熱系統的記號階梯圖結構，5.1.3 節為液體加熱系統由記號階梯圖至記號圖的轉換，5.1.4 節為液體加熱系統的驗證部分。

5.1.1　液體加熱系統案例分析的操作流程

液體加熱系統的案例分析流程如圖 5.1 所示，分析流程分為建置記號階梯圖、轉換、驗證三部分。首先利用階梯圖繪圖軟體 PLCOpenEditor 建置液體加熱系統的記號階梯圖，再利用轉換程式進行記號階梯圖的圖形判斷、圖形轉換及 PNML 檔案輸出，最後利用裴氏圖分析軟體讀取 PNML 檔案來比對記號圖流程是否正確。

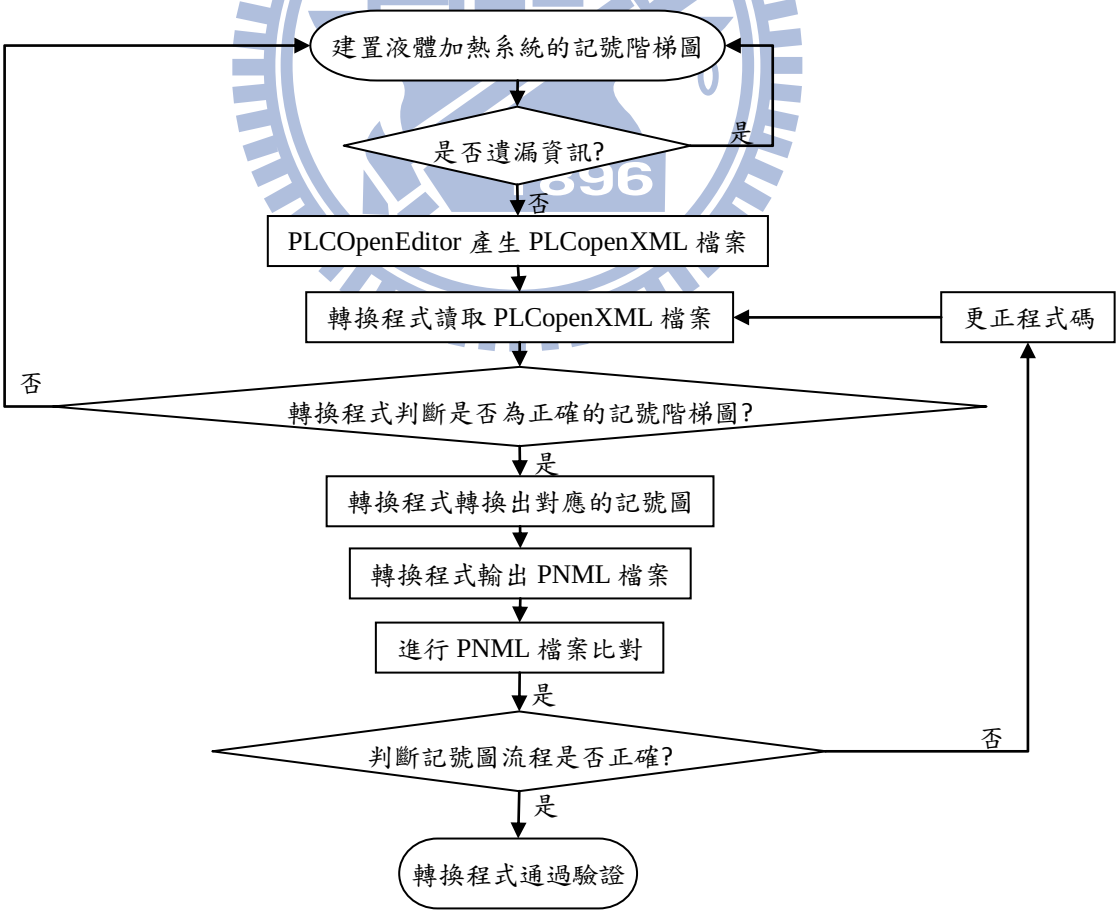


圖 5.1　液體加熱系統案例分析流程圖

5.1.2 建立液體加熱系統的記號階梯圖

分析流程的第一個步驟為建置液體加熱系統的記號階梯圖，表 5.1 為輸入接點與輸出線圈所代表的物理意義，圖 5.2 為液體加熱系統的記號階梯圖，其中每個輸出線圈為液體加熱系統的狀態，利用輸入接點組成的並聯條件控制輸出線圈，系統藉由不同的控制條件處於不同的狀態。

表 5.1 液體加熱系統記號階梯圖的輸入接點及輸出線圈名稱對照之意義

輸入接點及輸出線圈之名稱及對照意義			
P1	Vessel-a1 液體狀態為 full	T1	液體準備流入 Vessel-b2
P2	Vessel-a1 液體狀態為 decreasing	T2	液體流入 Vessel-b2
P3	Vessel-a1 液體狀態為 empty	T3	Vessel-b2 中之液體由 20℃加溫至 100℃後準備流出 Vessel-b2
P4	Vessel-a1 液體狀態為 increasing	T4	100℃之高溫液體流出 Vessel-b2 且流出後空的 Vessel-b2 恢復常溫 20℃
P5	Vessel-b2 液體狀態為 empty	T5	空的 Vessel-a1 補滿液體準備補充液體
P6	Vessel-b2 液體狀態為 increasing	T6	Vessel-a1 補滿液體
P7	Vessel-b2 液體狀態為 full		
P8	Vessel-b2 液體狀態為 decreasing		
P9	Valve-1 on		
P10	Valve-1 off		
P11	Valve-2 on		
P12	Valve-2 off		

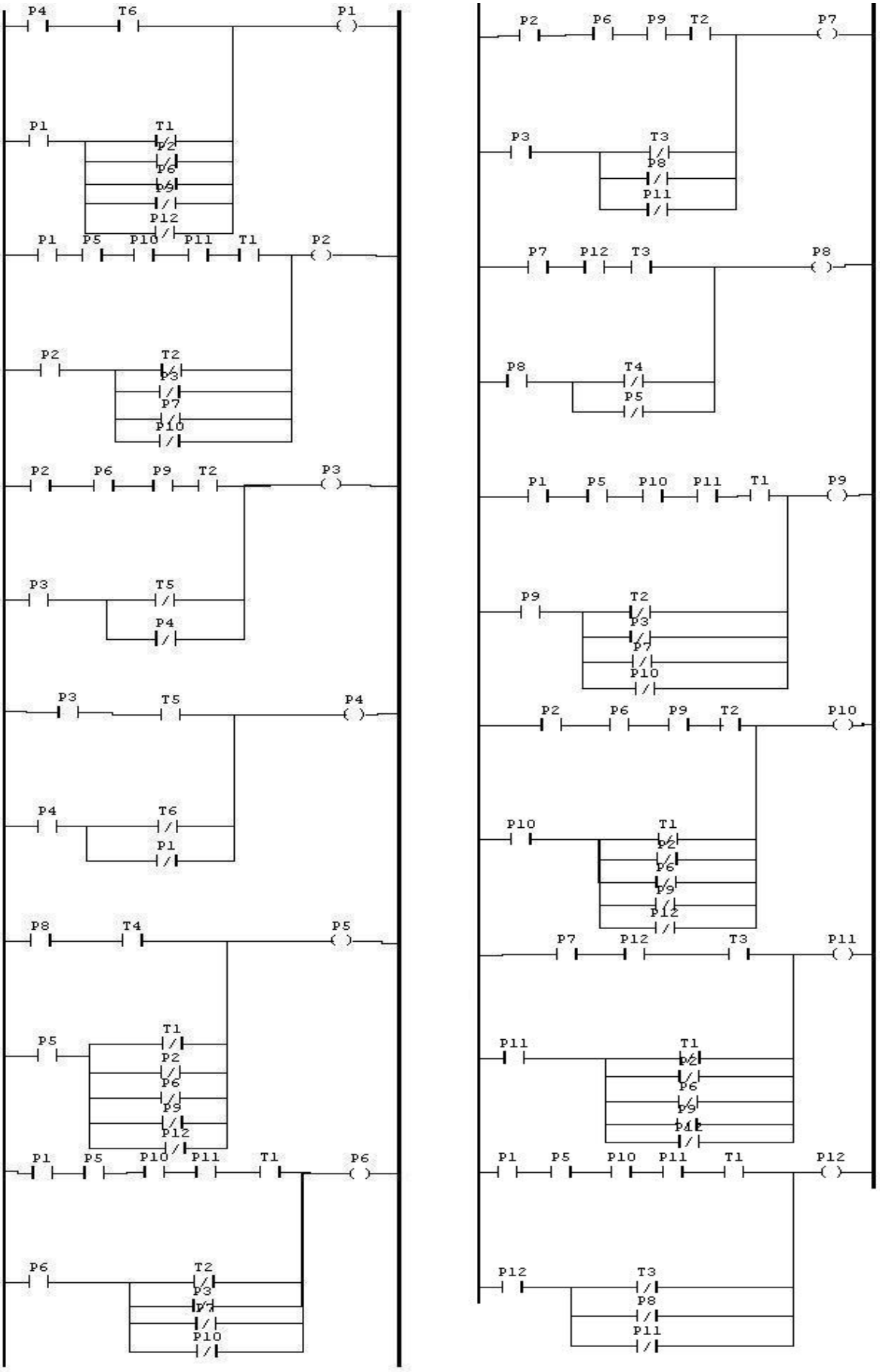


圖 5.2　液體加熱系統的記號階梯圖

5.1.3 液體加熱系統記號階梯圖至記號圖的轉換

轉換程式讀取液體加熱系統記號階梯圖的 PLCopenXML 檔案，會得到圖 5.3 左邊區域的記號階梯圖，隨即轉換出圖 5.3 右邊區域的記號圖。

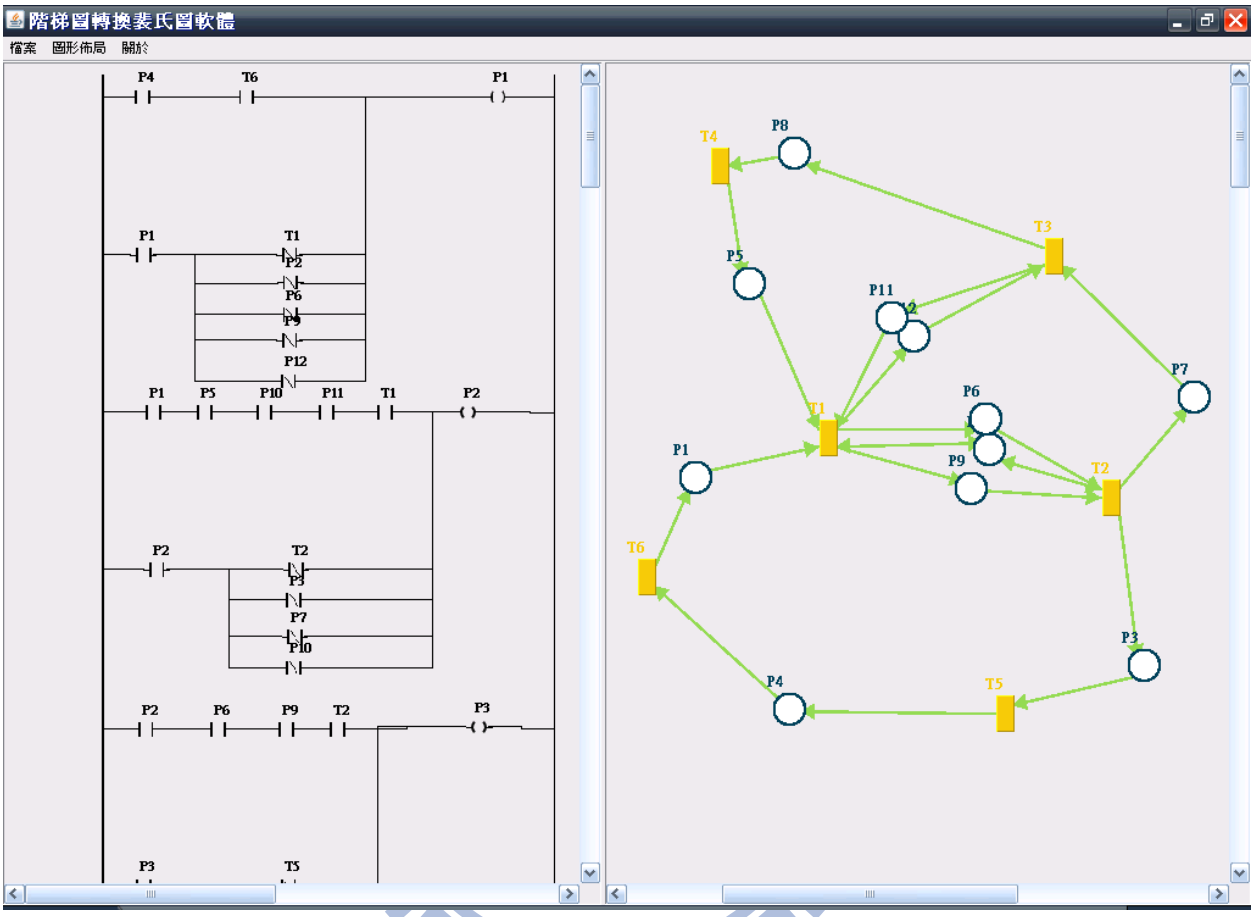


圖 5.3 轉換程式將液體加熱系統的記號階梯圖轉換為記號圖

5.1.4 液體加熱系統記號圖的驗證

利用轉換程式輸出液體加熱系統的 PNML 檔案後，進行分析流程最後的驗證步驟。為了驗證輸出的 PNML 檔案正確，本論文採用 PIPE2 免費裴氏圖分析軟體[19]，假若輸出檔案經由 PIPE2 軟體讀取之後，所呈現出的記號圖與原始液體加熱系統的記號相同，則可判定轉換程式輸出的 PNML 檔案正確無誤。

圖 5.4 為原始的液體加熱系統記號圖，利用 PIPE2 軟體讀取轉換程式輸出的 PNML 檔案後可得到圖 5.5，為轉換程式轉換後的液體加熱系統記號圖，若 PIPE2 軟體可產生記號圖，表式轉換程式輸出的檔案為正確的 PNML 格式，再比對原始的液體加熱系統記號圖之後，發現暫存點與轉移點之間的關係皆相同，因此即可判定轉換程式可將記號階梯圖轉換出正確的記號圖。

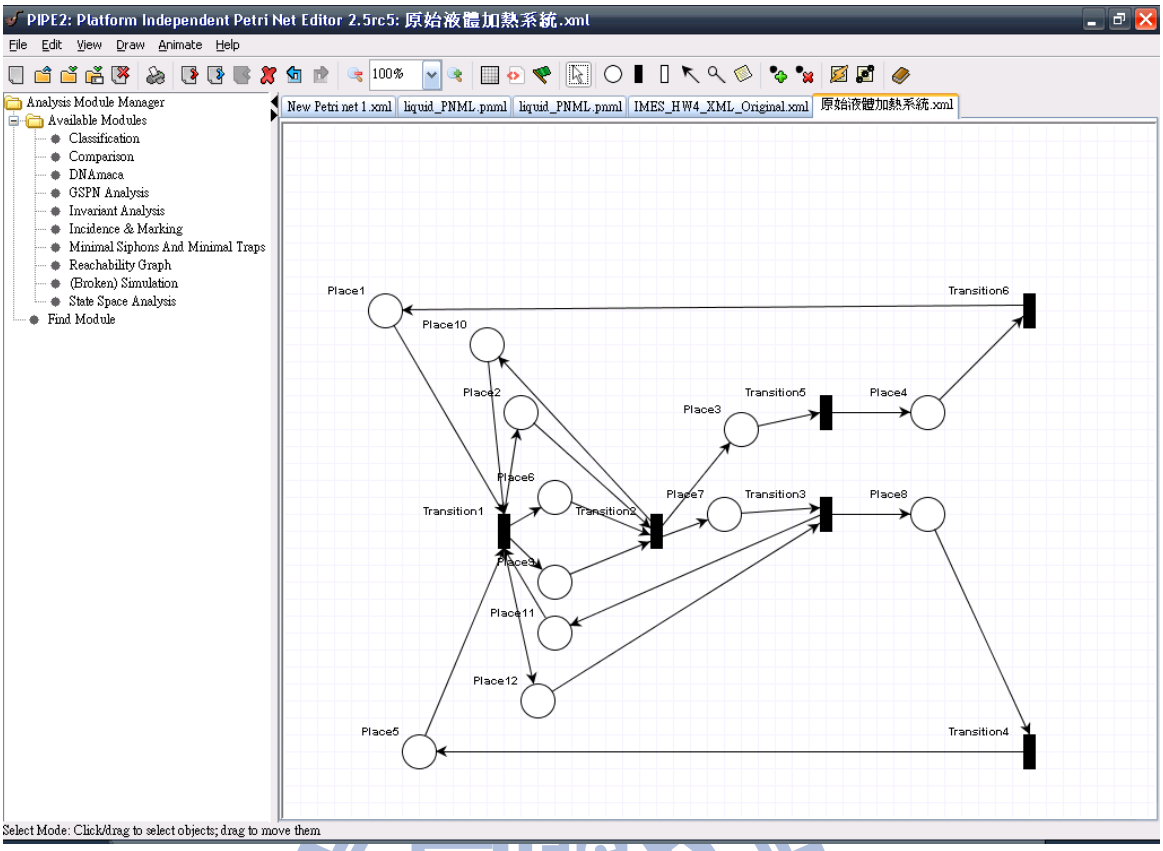


圖 5.4　原始液體加熱系統的記號圖

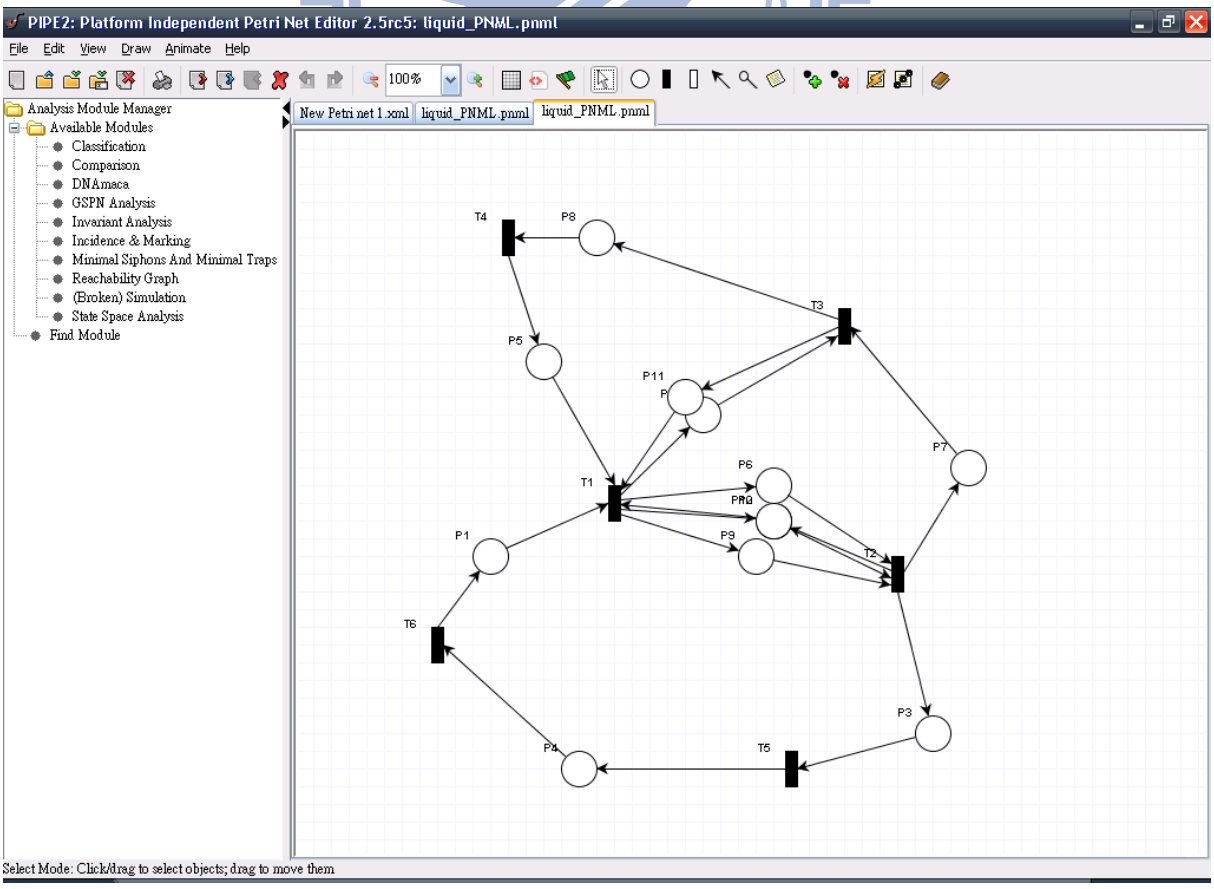


圖 5.5　轉換程式轉換出的液體加熱系統記號圖

5.2 灌模系統案例分析

本節案例利用灌模系統[2]做分析，5.2.1 節為灌模系統案例分析的操作流程及說明，5.2.2 節介紹灌模系統的記號階梯圖結構，5.2.3 節為灌模系統由記號階梯圖至記號圖的轉換，5.2.4 節為灌模系統的驗證部分。

5.2.1 灌模系統案例分析的操作流程

液體加熱系統的案例分析流程如圖 5.1 所示，分析流程分為建置記號階梯圖、轉換、驗證三部分。首先利用階梯圖繪圖軟體 PLCOpenEditor 建置液體加熱系統的記號階梯圖，再利用轉換成式進行記號階梯圖的圖形判斷、圖形轉換及 PNML 檔案輸出，最後利用裴氏圖分析軟體讀取 PNML 檔案來比對記號圖流程是否正確。

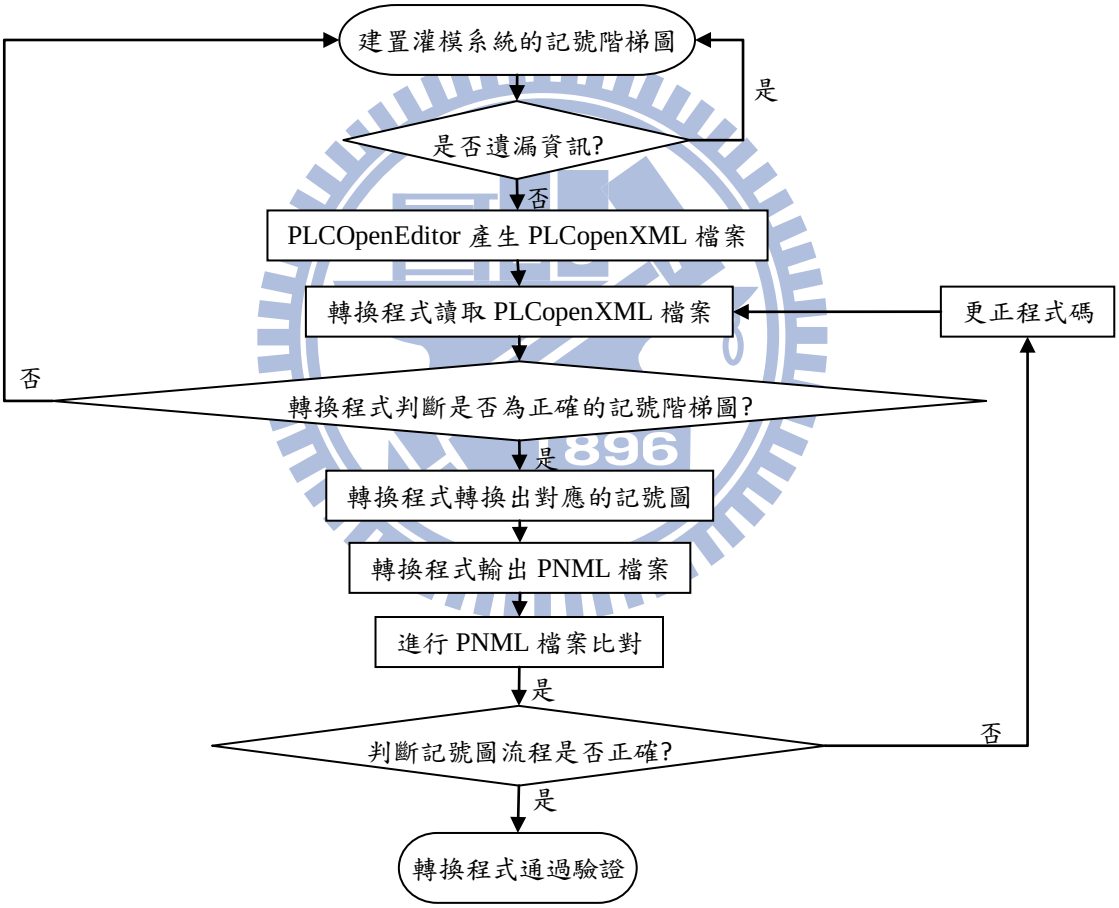


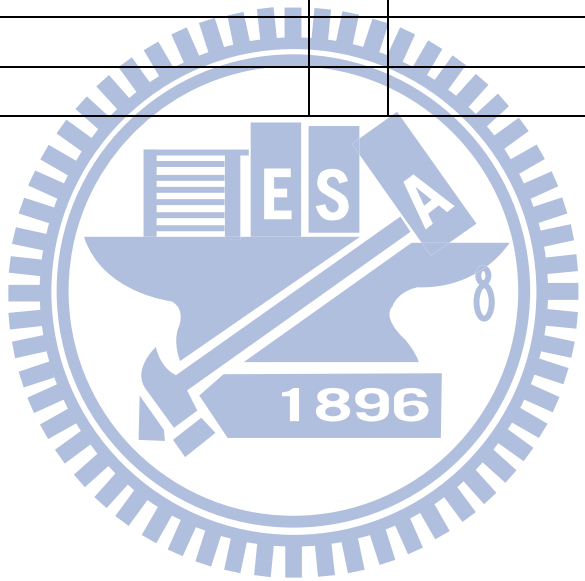
圖 5.6 灌模系統案例分析的流程圖

5.2.2 建立灌模系統的記號階梯圖

分析流程的第一個步驟為建置灌模系統的記號階梯圖，表 5.2 為輸入接點與輸出線圈所代表的物理意義，圖 5.7 為灌模系統的記號階梯圖，其中每個輸出線圈為灌模系統的狀態，利用輸入接點組成的並聯條件控制輸出線圈，系統藉由不同的控制條件處於不同的狀態。

表 5.2　灌模系統記號階梯圖的輸入接點及輸出線圈名稱對照之意義

輸入接點及輸出線圈之名稱及對照意義			
P1	閥 3 開	T1	液體 A、B 準備流入槽中
P2	零液位攪拌槽	T2	完成液體 A 注入槽中
P3	閥 1 關	T3	液體 B 注入槽中
P4	閥 2 關	T4	攪拌液體 A、B，使其混合
P5	閥 1 開	T5	準備讓槽中的液體流出
P6	閥 2 開	T6	槽中的液體流出中
P7	昇液位攪拌槽		
P8	液體 A 注入完		
P9	液體 B 注入完		
P10	液體 A 混合完		
P11	液體 B 混合完		
P12	滿液位攪拌槽		
P13	閥 3 關		
P14	降液位攪拌槽		



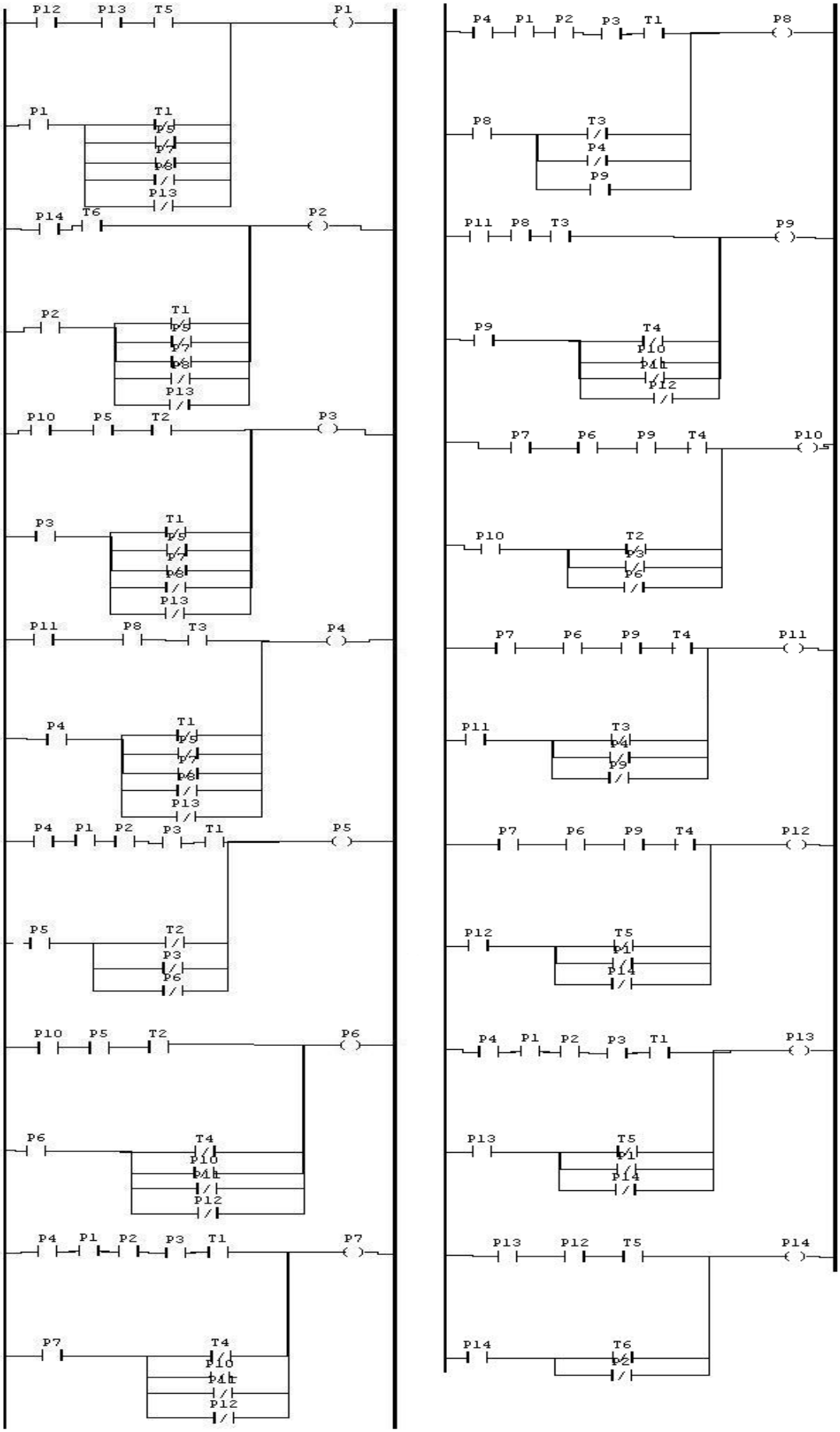


圖 5.7　灌模系統的記號階梯圖

5.2.3 灌模系統記號階梯圖至記號圖的轉換

轉換程式讀取液體加熱系統記號階梯圖的 PLCopenXML 檔案，會得到圖 5.8 左邊區域的記號階梯圖，隨即轉換出圖 5.8 右邊區域的記號圖。

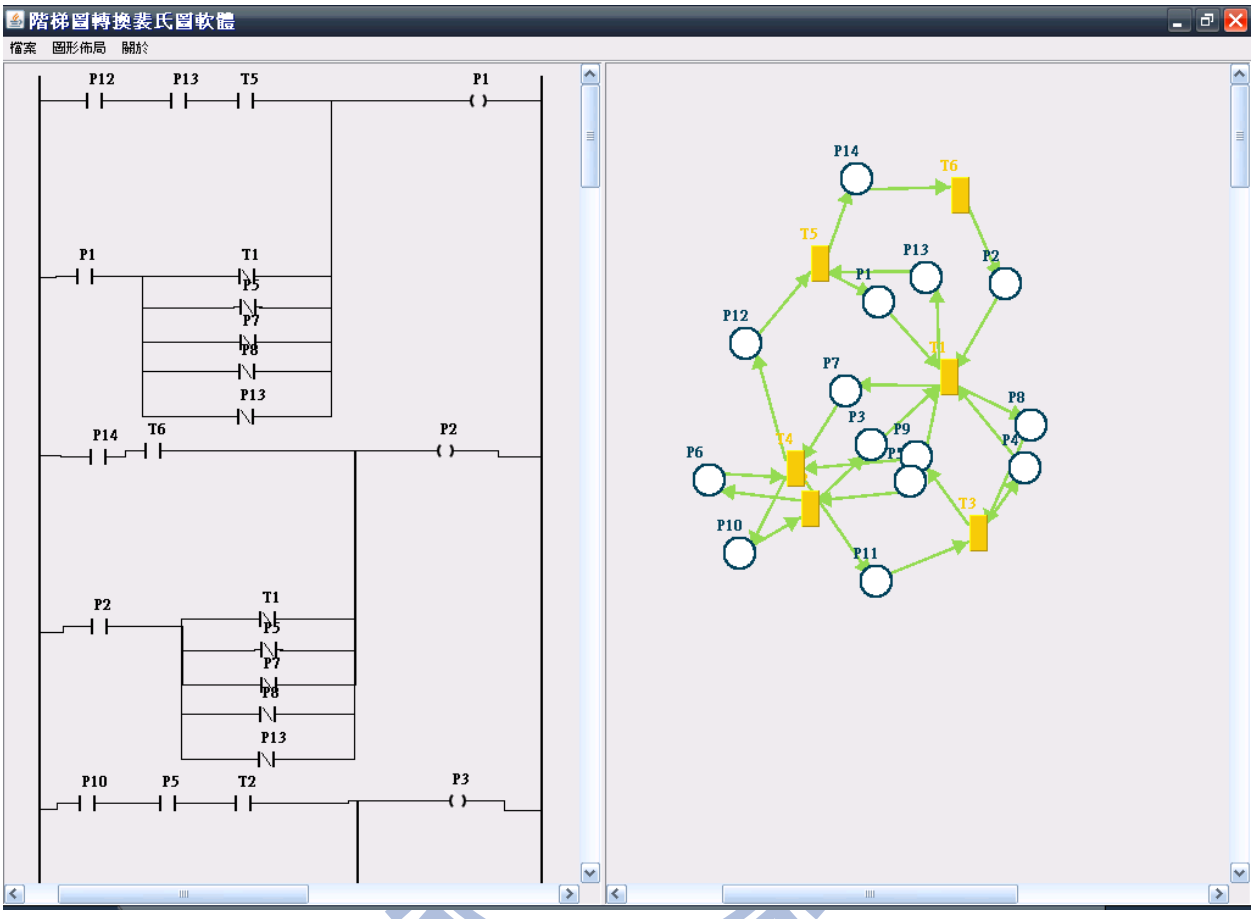


圖 5.8 轉換程式將灌模系統的記號階梯圖轉換為記號圖

5.2.4 灌模系統記號圖的驗證

利用轉換程式輸出灌模系統的 PNML 檔案後，進行分析流程最後的驗證步驟。為了驗證輸出的 PNML 檔案正確，本論文採用 PIPE2 免費裴氏圖分析軟體，假若輸出檔案經由 PIPE2 軟體讀取之後，所呈現出的記號圖與原始灌模系統的記號相同，則可判定轉換程式輸出的 PNML 檔案正確無誤。

圖 5.9 為原始的灌模系統記號圖，利用 PIPE2 軟體讀取轉換程式輸出的 PNML 檔案後可得到圖 5.10，為轉換程式轉換後的灌模系統記號圖，若 PIPE2 軟體可產生記號圖，表式轉換程式輸出的檔案為正確的 PNML 格式，再比對原始的灌模系統記號圖之後，發現暫存點與轉移點之間的關係皆相同，因此即可判定轉換程式可將記號階梯圖轉換出正確的記號圖。

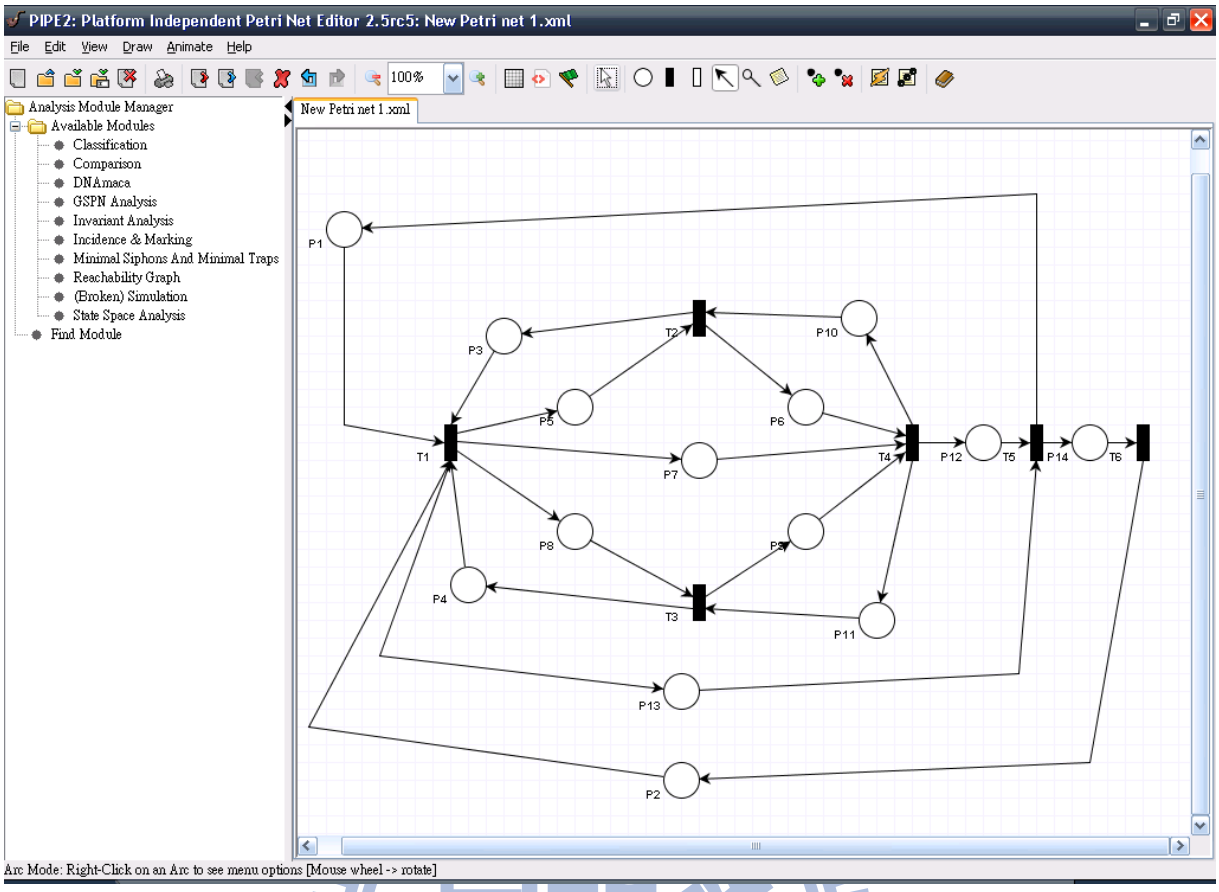


圖 5.9　原始灌模系統的記號圖

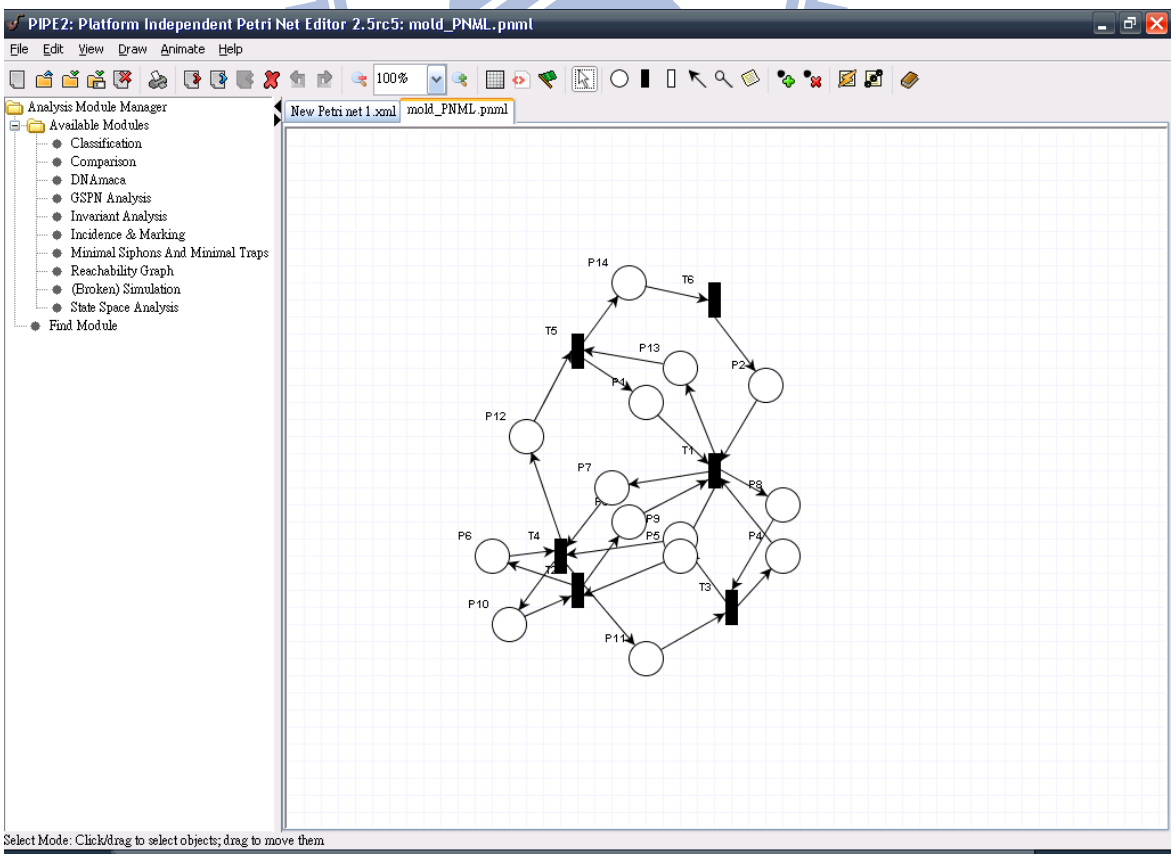


圖 5.10　轉換程式轉換出的灌模系統記號圖

第六章　結論

本章主要在說明研究結果與未來研究方向，本章共分為兩小節。6.1 節『結論』說明研究結果以及尚未研究的部分，6.2 節『後續研究』針對本論文尚未研究的部分提供未來可繼續深入研究的議題。

6.1　結論

目前已有的研究都只有提出階梯圖轉換為裴氏圖的轉換理論，第三章中利用階梯圖轉換為裴氏圖的研究理論，定義出記號階梯圖，且實作出記號階梯圖轉換至記號圖的轉換程式，為製造系統自動化建立更完善的流程。

階梯圖雖然在工業界已是普遍使用於控制可程式邏輯控制器的程式語言，裴氏圖也逐漸被工業界使用於控制製造系統，兩者各有優點。第四章研究記號階梯圖轉換至記號圖的轉換程式實作，加速導入自動製造執行系統的時間，也降低了人工轉換的錯誤率。有了時間的競爭優勢加上高度正確的自動製造執行系統，可以帶給工業界的自動化控制領域更嶄新的視野。

由於記號階梯圖為本論文第三章定義而成，因此記號階梯圖無法涵蓋所有階梯圖格式如所示，因此在研究階梯圖的各種樣式將是一個議題。階梯圖繪圖軟體可以繪製出各種階梯圖，但在轉換程式中無法判讀記號階梯圖以外的階梯圖形態，記號階梯圖形態以外的階梯圖，不在本論文研究範圍，轉換程式若遇到非記號階梯圖形態的階梯圖，會告知使用者此不為正確的記號階梯圖。

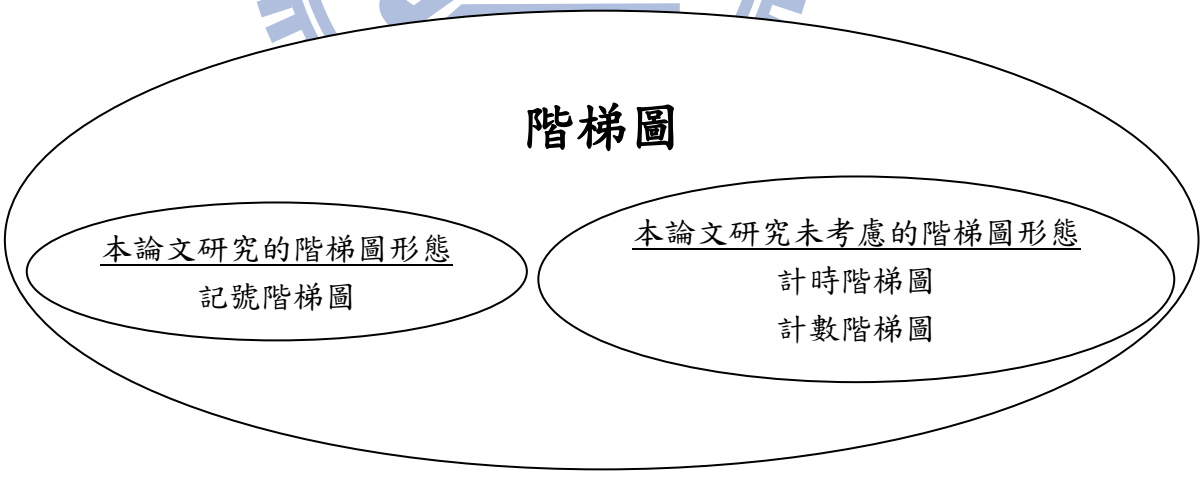


圖 6.1　階梯圖形態

6.2 後續研究

本論文的自動轉換程式主要針對記號階梯圖轉換至記號圖，並且對於控制記號圖的偵查方程式，建立了對應的階梯圖控制條件。未來可以針對逆向工程方面進行研究[5]，也就是針對記號圖轉換為記號階梯圖做更深入的研究與實作。逆向工程轉換部分，若可以先計算出記號圖的偵查方程式，並且在轉換過程將偵查方程式自動轉換為階梯圖的控制條件，鑲嵌進轉換好的階梯圖上，那對於製造系統的自動化控制與智慧化偵測會有非常大的幫助。

此外，本論文研究限制在記號階梯圖與記號圖的轉換，若是非記號階梯圖形態的階梯圖，轉換程式無法轉換為記號圖。假如沒有記號階梯圖與記號圖的限制，在實務應用面上，可以更加擴大自動化製造系統使用的範疇，讓更多產業可以快速導入製造系統的智慧化控制模式。

參考文獻

[1] 范植宇，「階層轉換法的自動化：從規格到設計的樹型轉換」，國立交通大學工業工程與管理研究所碩士論文，2009 年 2 月。

[2] 梁高榮，「虛體製造系統的多緒架構」，機械工業，十二月，111-121 頁，2006。

[3] 梁高榮，「裴氏圖與記號圖：可程式控制器的分析工具」，機械工業，十月，119-130 頁，2009。

[4] 梁高榮，「擬陣理論在現場監控的應用」，機械工業，六月，176-188 頁，2001。

[5] 黃柏勳、梁高榮，「三層式架構解裴氏圖/階梯圖轉換問題」，機械工業，八月，2010。(即將出版)

[6] 勞虎，無廢話 XML，1991 年 10 月。

[7] Billington, J., et al., "The Petri Net Markup Language: Concepts, Technology, and Tools," Applications and Theory of Petri Nets 2003, (Eds. W. van der Aalst and E. Best), LNCS, Springer,Vol.2679, pp. 483-505, 2003.

[8] Chirn, J. L. and D. C. McFarlane, "Petri Nets Based Design of Ladder Logic Diagrams," UKACC International Conference: Control 2000, Cambridge, UK, September 4, 2000.

[9] http://en.wikipedia.org/wiki/Boolean_logic, Boolean logic - Wikipedia, January 19, 2010.

[10] http://en.wikipedia.org/wiki/ISO_standard#International_Standards_and_other_publications, International Organization for Standardization - Wikipedia, January 21, 2010.

[11] http://en.wikipedia.org/wiki/Ladder_logic, Ladder logic - Wikipedia, October 23, 2009.

[12] http://en.wikipedia.org/wiki/Petri_Net_Markup_Language, Petri Net Markup Language - Wikipedia, January 21, 2010.

[13] http://en.wikipedia.org/wiki/Programmable_logic_controller, Programmable logic controller - Wikipedia, October 23, 2009.

[14] http://en.wikipedia.org/wiki/Robot_control, Robot control - Wikipedia, January 19, 2010.

[15] http://en.wikipedia.org/wiki/Unified_Modeling_Language, Unified Modeling Language - Wikipedia, January 19, 2010.

[16] http://en.wikipedia.org/wiki/XML_data_binding, XML data binding - Wikipedia, February 1, 2010.

[17] http://iaaa.cps.unizar.es/curriculum/09-Otras-Publicaciones-Congresos/cong_1992_INCOM_Upn.pdf, UPN: l\. Petri Net Based Graphical Representation for Company Policy Specifications in CIM, August 30, 1991.

[18] http://java.sun.com/developer/technicalArticles/WebServices/jaxb/#introjb, Java Architecture for XML Binding (JAXB), June 7, 2010.

[19] http://pipe2.sourceforge.net/index.html, Platform Independent Petri net Editor 2 software (PIPE2), April 2, 2010.

[20] http://www.360doc.com/content/06/0324/14/7095_85519.shtml, JAXP (Java API for XML Parsing), January 13, 2010.

[21] http://www2.informatik.hu-berlin.de/top/pnml/about.html, Petri Net Markup Language,

October 25, 2009.
[22] http://www.automationml.org/, AutomationML™, January 19, 2010.
[23] http://www.beremiz.org/documentation/the-plcopen-editor, The PLCOpen Editor, June 8, 2010.
[24] http://www.chuhu.com/docs/60.thtm, java 中四種操作 xml 方式的比較, CHUCHU, January 13, 2010.
[25] http://www.eclipse.org/, Eclipse.org home, June 8, 2010.
[26] http://www.plcopen.org/pages/tc6_xml/, TC6 - XML Schemes, October 25, 2009.
[27] http://www.plcopen.org/pages/organization/, PLCopen organization, October 25, 2009.
[28] http://www.plctutor.com/relay-ladder-logic.html, Ladder Logic, October 27, 2009.
[29] http://www.sfu.ca/~dgasevic/projects/P3net/, P3 - Petri net tool, June 8, 2010.
[30] http://www.w3.org/XML/, Extensible Markup Language (XML), October 25, 2009.
[31] http://zh.wikipedia.org/wiki/IEC_61131, IEC 61131, Wikipedia, December 23, 2009.
[32] http://zh.wikipedia.org/wiki/SAX, SAX, Wikipedia, January 13, 2010.
[33] http://zh.wikipedia.org/zh-tw/XPath, XML Path Language, Wikipedia, June 22, 2010.
[34] Lee, G. B., Zandong, H. and Lee J. S., "Automatic Generation of Ladder Diagram with Control Petri Net," Journal of Intelligent Manufacturing, Vol. 15, No. 2, pp. 245-252, 2004.
[35] Moller, A. and Schwartzbach, M., "An Introduction to XML and Web Technologies," Addison-Wesley, 2006.
[36] Murata, T., "Petri Nets: Properties, Analysis and Application," Proceedings of the IEEE, Vol. 44, pp 541-579, 2006.
[37] Peng, S. S. and Zhou, M. C., "Ladder Diagram and Petri-Net-based Discrete-Event Control Design Methods," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 34, No. 4, 2004.
[38] Venkatesh, K, Zhou, M. C. and Caudill, R. J., "Comparing Ladder Logic Diagrams and Petri Nets for Sequencecontroller Design through a Discrete Manufacturing System," IEEE Transactions on Industrial Electronics, Vol. 41, Issue: 6, p.p. 611-619, 1994.
[39] Venkatesh, K. and Zhou, M. C., "Modeling, Simulation, and Control of Flexible Manufacturing Systems－A Petri Net Approach," Singapore: World Scientific, 1998.
[40] Weber, M. snd Kindler, E., "The Petri Net Markup Language," Petri Net Technology for Communication-based Systems: Advances in Petri Nets (Eds. H. Ehrig, et al.), LNCS, Springer, Vol. 2472, pp. 124-144, 2003.