

國立交通大學

工業工程與管理學系

碩士論文

利用記號圖設計生產系統的推拉式指標

Marked Graph Based Push-Pull Index Design for
Production Systems



研究生：朱明典

指導教授：梁高榮博士

中華民國一百年六月



利用記號圖設計生產系統的推拉式指標

研究生：朱明典

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

本文提出兩項推拉式指標設計方法來描述重覆性製造系統裡物流的行為。這兩項設計方法都是建立在製造系統資訊流的基礎上。依據分析結果，發現歐氏記號圖可以用來表達資訊流的行為。因此歐氏記號圖可以用來設計推拉式指標。在此情況下，歐氏記號圖的兩項特性就成為計算推拉式指標的設計原理。其一是迴路分析；另一是暫存點分析。根據此設計原理，本文已開發出套裝軟體來計算歐氏記號圖的推拉式指標。



關鍵字：

重覆性製造系統(Repetitive Manufacturing System)

歐氏記號圖(Eulerian Marked Graph)

物流(Material Flow)

資訊流 (Information Flow)

推拉式指標(Push-Pull Index)

Marked Graph Based Push-Pull Index Design for Production Systems

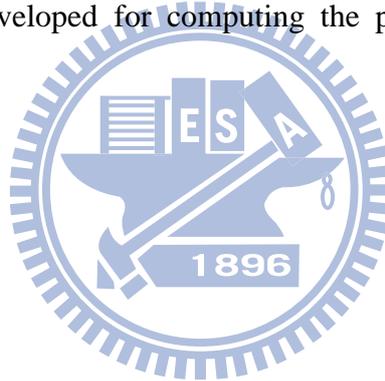
Student : Ming-Tien, Chu

Advisor : Dr. Gau-Rong, Liang

Department of Institute of Industrial Engineering & Management
National Chiao Tung University

Abstract

Two push-pull index design methods are proposed for charactering the patterns of material flow in repetitive manufacturing system. Both design methods are built on the information flow foundation of manufacturing system. Based on the analytical results, Eulerian marked graph is used for representing information flow. Then the push-pull index can be designed from the given Eulerian marked graph. In such a situation, two salient features of Eulerian marked graph become the design principles for computing push-pull index. One is circuit analysis; the other is place analysis. According to the design principles, a software package has been developed for computing the push-pull index of a given Eulerian marked graph.



Keyword :

Repetitive Manufacturing System

Eulerian Marked Graph

Material Flow

Information Flow

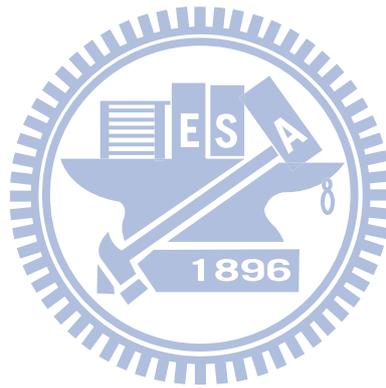
Push-Pull Index

致謝

本論文得以順利完成，首先要感謝我的指導教授梁高榮老師自入學以來整整二年寒暑從未間斷的辛勤教導及口試委員陳文智老師和林春成老師所提供之寶貴意見，俾使論文得以更加完善。再感謝有幸參與「無線頻譜執照釋出的機制設計」之專案計畫(計畫編號：99-2219-E-009-002)給予生活補貼，使得學生能專心於研究之中。

隱約還記得考上研究所的喜悅，轉眼間就寫致謝。靜下心來回想過去二年的時光的生活，除了學習上及生活上由博班學長范植宇、碩班學長黃柏勳，何青樺、鄭瑋廷、顧家樺、魏良縈及鄭惠華學長姊的細心指導。實驗室同學連惠珍、鄭仲元、劉思宇及彭思瑜共同研究的時光。還有學弟妹方本欣、張舜龍、林哲慧、徐宏智及張弘。

最後，要感謝的人實在太多，如有被遺忘的朋友在此亦一併感謝，感謝大家於這二年來對我的照顧謹以本論文獻給我最愛的父母、家人以及陪我伴我度過這段成長的老師和朋友們。





目錄

目錄	i
圖目錄	iv
表目錄	vii
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	4
1.4 論文架構	5
第二章 文獻回顧	6
2.1 推式與拉式的生產系統	6
2.1.1 推式及拉式生產系統觀念的發展	6
2.1.2 看板系統	8
2.2 裴氏圖模型	10
2.2.1 裴氏圖與記號圖	10
2.2.2 簡化法則	11
2.2.3 可達圖表達法	12
2.2.4 鎖死問題	13
2.2.5 系統不變量	14
2.2.6 週期時間分析	14
2.2.7 裴氏圖可加註語言	15
2.2.8 偵查法則	17
2.3 裴氏圖與生產系統	18
2.3.1 裴氏圖應用於看板系統	18
2.3.2 裴氏圖應用於推式及拉式系統	20
2.4 整合性裴氏圖模組	22
2.4.1 IDEFO 格式轉換至裴氏圖可加註語言格式模組	22
2.4.2 裴氏圖基本性質分析模組	23
2.4.3 裴氏圖自動化規範測試	23
第三章 生產系統的推拉式指標設計	24
3.1 推拉式指標分析流程	24
3.1.1 物流與資訊流的分離	25
3.1.2 資訊流與記號圖的關係	26

3.1.3.	修正記號圖步驟.....	26
3.2	純推式系統的記號圖表達法.....	27
3.3	純拉式系統的記號圖表達法.....	33
3.3.1	推式系統與拉式系統定義.....	34
3.4	迴路數導向指標設計.....	36
3.4.1	推拉介面.....	36
3.4.2	迴路數導向指標.....	40
3.5	暫存點導向指標設計.....	41
3.5.1	暫存點導向指標.....	41
3.6	複雜生產系統案例的推拉式指標分析.....	43
第四章	整合性裴氏圖軟體開發環境的設計.....	49
4.1	裴氏圖設計流程建立.....	49
4.2	程式模組架構.....	51
4.2.1	GUI 模組.....	51
4.2.2	IDEF0 格式轉裴氏圖格式模組.....	53
4.2.3	基本性質與不變量分析模組及規範測試模組.....	54
4.2.4	偵查法則模組.....	56
4.2.5	推拉式指標模組.....	58
第五章	範例實際操作.....	60
5.1.	程式介面及功能.....	60
5.1.1	檔案載入及圖形佈局.....	60
5.1.1.1	檔案開啟及 IDEF0 轉換.....	62
5.1.1.2	圖形佈局.....	64
5.1.2	裴氏圖應用區.....	64
5.1.2.1	裴氏圖分析與驗證.....	64
5.1.2.2	規範測試序列分析.....	68
5.1.3	偵查法則設計.....	68
5.2.	推拉式指標的計算.....	70
5.2.1	修正製造系統及計算推拉式指標.....	70
5.2.2	修正灌模系統記號圖及計算推拉式指標.....	72
第六章	結論與未來發展.....	74
6.1	結論.....	74
6.2	未來研究與發展.....	75
參考文獻	76



圖目錄

圖 2.1 裴氏圖與推拉式生產系統的文獻回顧關係.....	6
圖 2.2 看板圖示說明.....	8
圖 2.3 單看板系統流程.....	9
圖 2.4 雙看板系統流程.....	9
圖 2.5 轉移點激發法則.....	11
圖 2.6 簡化法則.....	12
圖 2.7 液體加熱系統.....	12
圖 2.8 液體加熱系統可達狀態.....	13
圖 2.9 XML 格式.....	15
圖 2.10 裴氏圖可加註語言核心模組.....	16
圖 2.11 裴氏圖文件資料交換.....	16
圖 2.12 子擬陣技術設計偵查器.....	17
圖 2.13 生產系統以站點表示.....	18
圖 2.14 前後站點之間的關連性.....	19
圖 2.15 三站點看板系統.....	19
圖 2.16 三站點看板系統(需求、供給無限).....	20
圖 2.17 推式系統.....	20
圖 2.18 拉式系統.....	21
圖 2.19 IDEF0 格式轉換至裴氏圖可加註語言格式流程.....	22
圖 3.1 推拉式指標的建立流程.....	24
圖 3.2 物流與資訊流分離.....	25
圖 3.3 生產資訊流回饋(資訊流).....	26
圖 3.4 修正生產裴氏圖流程.....	26
圖 3.5 例 3.1 生產流程的物流及資訊流.....	27
圖 3.6 例 3.1 純推式系統的資訊流.....	28
圖 3.7 例 3.1 純推式系統的物流.....	28
圖 3.8 例 3.1 純推式系統的資訊流.....	29
圖 3.9 例 3.2 生產流程的物流及資訊流.....	29
圖 3.10 例 3.2 純推式系統的資訊流-1	30
圖 3.11 例 3.2 純推式系統的資訊流-2	30
圖 3.12 純推式系統(N 個站點).....	31
圖 3.13 純推式系統生產.....	31
圖 3.14 例 3.3 生產流程的物流及資訊流.....	33
圖 3.15 例 3.3 資訊流.....	34
圖 3.16 純拉式系統(三站點).....	34
圖 3.17 純拉式系統.....	34

圖 3.18 推拉式推拉式指標	36
圖 3.19 例 3.4 的物料清單	36
圖 3.20 例 3.4 的物流	37
圖 3.21 例 3.4 的物流、資訊流及可達圖	37
圖 3.22 例 3.5 的物流及資訊流	38
圖 3.23 例 3.4 資訊流及推拉介面	39
圖 3.24 例 3.5 資訊流及拉式介面	39
圖 3.25 例 3.4 資訊流及暫存點來源	41
圖 3.26 例 3.5 資訊流及暫存點來源	42
圖 3.27 四作業生產物流	43
圖 3.28 例 3.6 的資訊流記號圖	43
圖 3.29 例 3.7 的資訊流記號圖	44
圖 3.30 例 3.8 的資訊流記號圖	44
圖 3.31 例 3.9 的資訊流記號圖	45
圖 3.32 例 3.10 的資訊流記號圖	45
圖 3.33 例 3.11 的資訊流記號圖	46
圖 3.34 例 3.12 的資訊流記號圖	46
圖 3.35 例 3.6 至 3.12 的IP及Ic分佈	47
圖 4.1 自動化開發流程	50
圖 4.2 模組關連資料	51
圖 4.3 自動化系統主畫面	53
圖 4.4 形化使用者介面類別	54
圖 4.5 IDEF0 格式轉換至裴氏圖可加註語言格式類別	55
圖 4.6 基本性質及不變量分析類別	57
圖 4.7 偵查法則套件(Package Diagram)	57
圖 5.1 程式開啟畫面	60
圖 5.2 灌模系統	62
圖 5.3 控制輸送帶	63
圖 5.4 開啟檔案畫面	63
圖 5.5 退出檔案畫面	64
圖 5.6*.idl 轉換裴氏圖預設佈局	64
圖 5.7 裴氏圖分析與驗證	65
圖 5.8 擷取模型性質	65
圖 5.9 可達狀態與鎖死分析	66
圖 5.10 可達狀態與鎖死分析	66
圖 5.11 轉移點不變量分析	67
圖 5.12 暫存點不變量分析	67
圖 5.13 規範測試序列分析	68

圖 5.14 偵查法則介面.....	68
圖 5.15 偵查法則計算 1.....	69
圖 5.16 偵查法則計算 2.....	69
圖 5.17 例 5.2 物流.....	70
圖 5.18 修正圖 5.17.....	71
圖 5.19 例 5.2 畫面展示.....	71
圖 5.20 灌模系統記號圖.....	72
圖 5.21 灌模系統記號圖推拉式指標.....	73



表目錄

表 2.1 推式及拉式系統定義列表	7
表 2.2 看板功能及使用規則	8
表 2.3 液體加熱系統暫存點及轉移點說明	13
表 2.4 裴氏圖格式發展整理	16
表 3.1 推式及拉式系統生產資訊流比較	35
表 3.2 四作業生產命令來源列表	46
表 3.3 推拉式系統指標值	47
表 3.4 推拉式指標優缺點比較	48
表 4.1 自動化模組與套件關係列表	52
表 4.2 使用者介面類別	53
表 4.3 IDEF0 轉換模組類別說明	55
表 4.4 基本性質模組類別說明	56
表 4.5 規範測試模組類別說明	56
表 4.6 偵查法則套件功能說明	58
表 5.1 例 5.1 暫存點列表	61
表 5.2 例 5.2 轉移點作業說明	61
表 5.3 灌模及運送裴氏圖對照列表	63





第一章緒論

本章主要說明本論文的研究方向、目的與論文的整體架構，本章共分四小節。1.1 節「研究動機」。1.2 節「問題界定及研究目的」。1.3 節「研究方法」。1.4 節「論文架構」。

1.1 研究動機

自工業革命以來，機械的發明取代人力。大規模的工廠化，成本與時間亦相對縮減，生產模式產生大幅的改變，由手工式生產慢慢轉變為集聚至工廠生產。為滿足消費者的多樣且多變需求，生產流程模式不斷的進化，發展出各式不同的生產模式，又為再增加競爭力減少浪費，大野耐一(Taiichi Ohno)[43]於 1988 年發展出及時生產系統(Just In Time Production System, 簡稱 JIT)，且採用看板(Kanban)作為生產排程式控制工具，改變原有生產模式。由於及時生產系統不同於以往的物料需求規劃(Material Requirement Planning, MRP)系統，其物料供給方式條件不同。霍爾(R.W. Hall)[18]於 1983 年針對這二種生產系統提出推式系統(Push System)及拉式系統(Pull System)的觀念。

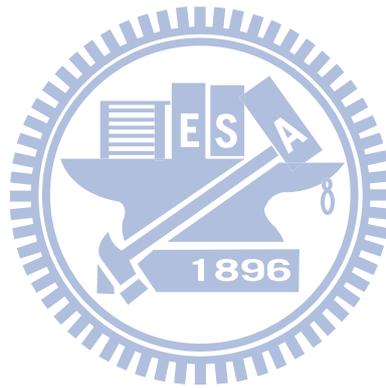
波尼等人(M.C. Bonney et al.)[13]有整理出歷年來各學者所提出之推式系統及拉式系統的定義，在英國標準協會(Britain Standard Institute, BSI)[23]及美國生產及存貨管理協會(The Association for Operations Management, APICS)[22]亦有各自的文字定義。推式系統及拉式系統存在著不同的優缺點。在混合式生產系統中所占推拉式系統之比重差異會突顯其優缺點，進而影響到生產績效。但目前尚未有數學上的定義，無法計算出其推式及拉式系統在生產系統之中的比重。

裴氏圖(Petri Net)[26]為同時具有圖形模型和數學性質之工具，可應用在許多系統上的建立與數學分析。近年來可加註語言(Extensible Markup Language, XML)[11]的發展，裴氏圖可加註語言(Petri Net Markup Language, PNML)[29]出現，使得裴氏圖電子資訊的交換更加便利。又馬斯科洛等人(M. D. Mascolo et al.)[40]提出裴氏圖應用在看板系統上及波尼等人[13]提出使用裴氏圖瞭解推式及拉式系統之間的差異，說明著推式系統及拉式系統是可用裴氏圖作為表示。

記號圖(Marked Graph)[42]是裴氏圖中的一種特例。修正馬斯科洛[40]及波尼[13]的圖型後，將其物流及信息流分離。利用推式系統及拉式系統在記號圖之間的差異，及記號的數學性質，可量化在生產系統內推式系統及拉式系統的比重。用來評估在混合式系統中推式系統及拉式系統的比重，藉由此量化指標可計算出在各產業在不同生產系統模式之績效。若建制完成後指標及績效的資料後，未來可供新開發之生產系統作為評估參考，亦或其他生產系統的改善。若擴大其應用內容，可延伸至整個供應鏈，找出最適合的推式系統及拉式系統指標。

在裴氏圖應用在生產系統自動化方面的軟體已有相當多人在開發，如由 IDEF0(Integration Definition 0)[24]格式轉換至裴氏圖可加註語言[2]、裴氏圖的基本性

質，如可達圖分析(Reachability Graph)、週期時間分析(Cycle Time Analysis)、不變數分析(Invariants Analysis)等[1]。亦有利用 Java 程式撰寫開放資料伺服器之連接架構[9]。自動化製造系統的設計與實作需歷經五大步驟[4]：(1)規格(Specification)、(2)驗證(Validation)、(3)撰碼(Coding)、(4)測試(Testing)、(5)偵查(Monitoring)。此五步驟之軟體皆已完備，目前只差一套整合性的軟體。加以整合後，若有 IDEF0 或是裴氏圖格式之檔時，即可立即使用其軟體上線，以減少建立自動化模組之時間。



1.2 問題界定及研究目的

在生產流程中，為滿足客戶需求、減少成本及在製品的浪費，需同時滿足多種生產條件。生產模式中推式系統及拉式系統的區別，波尼[13]整理不同角度提出各式定義及使用條件，生產現場管理者可依此作為生產流程的建制。但在現代極複雜的生產中，同時使用推式及拉式生產模式已成趨勢。在現有的研究領域中，雖已有推式系統及拉式系統的比較及說明，但目前較少同時擁有推式及拉式系統在生產系統內的研究。

在研討混合式生產系統的同時，首先需判斷系統內部推式系統及拉式系統所占之關係及比重。在過去的研究中，並無學者以數學模型來判別在生產系統內的推式系統及拉式系統。本論文的目的在分析在生產流程中，推式系統及拉式系統之間的比重，並量化製作成為指標。

在建制指標工具方面，本論文使用記號圖。自裴氏圖發展以來，又軟體的發展及可加註語言[11]的發明，使得在電子資料交換方面已非常便利。如波尼等人[13]已使用裴氏圖繪出推式系統及拉式系統並比較之、馬斯科洛等人[40]利用裴氏圖來表達看板系統、簡禎富[10]亦使用裴氏圖表達半導體生產流程。可說明裴氏圖的應用生產流程方面已十分廣泛。在修正馬斯科洛[40]及波尼[13]的理論後，會以記號圖作為建制指標工具。

本論文只著重于利用推式系統及拉式系統的資訊流流向差異建立推拉式指標，以表示在生產系統內推式系統及拉式系統之比重。在建立推拉式指標後的指標評估，及如何利用此指標作生產績效之評估，已非本論文所研討之範圍。

在生產製造流程中建立一套實作軟體，將利用 Java[28]語言建立一整合性裴氏圖自動化軟體，可簡化製造系統的設計與實作中規格、驗證、撰碼、測試、偵查，此五步驟。並利用物件導向程式語言 Java 撰寫整合相對應的監控員(Supervisor)、偵察員(Monitor)及維修員(Troubleshooter)軟體[4]，分別執行系統監控、偵錯以及故障診斷與修復的工作。

1.3 研究方法

本論文研究方法共分成四個階段，如下列所示。

第一步驟：針對過去波尼等人[13]及馬斯科洛等人[40]所提之方法中，將裴氏圖中暫存點定義以資訊流及物流的角度加以分析。

第二步驟：將其推式系統及拉式系統記號圖重新繪出，並給予定義。

第三步驟：依記號圖數學性質，找出推式系統及拉式系統之間的差異。

第四步驟：記號圖性質發展出推式系統及拉式系統推拉式指標。



1.4 論文架構

本論文共分六章，其各章內容大綱描述如下：

第一章：「緒論」，主要說明本論文的研究動機、問題界定及研究目的、研究方法
及論文架構。

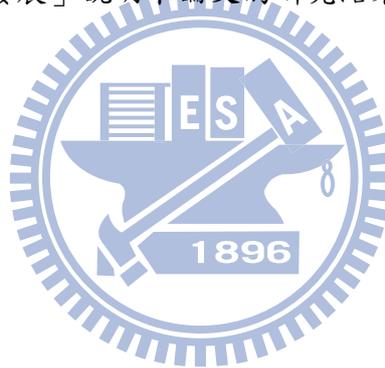
第二章：「文獻回顧」，主要說明推式系統及拉式系統之介紹、裴氏圖基本性質說
明及目前裴氏圖應用於推式系統及拉式系統或生產製造系統的架構及相關資訊。

第三章：「生產系統的推拉式指標設計」，針對推式及拉式系統再定義，並利用記
號圖性質找出其特性，及建立推拉式指標。

第四章：「整合性裴氏圖軟體開發環境的設計」，將由 IDEF0 格式轉置成裴氏圖可
加註語言格式，並計算裴氏圖數學性質，找出系統之特性，如鎖死狀態、路徑規範測
試、不變數等。

第五章：「範例實際操作」，以灌模系統為例，由 IDEF0 格式輸入後，可自動轉成
裴氏圖，並利用裴氏圖之特性，以作為決策系統之用。分析及計算灌模系統的推拉式
指標。

第六章「結論與未來發展」說明本論文的研究結果以及未來可延伸的發展方向。



第二章文獻回顧

本章主要目的在回顧本論文所用到之推式系統與拉式系統、裴氏圖及裴氏圖應用在生產系統文獻，以及過去文獻中裴氏圖在生產系統上(包含看板系統及推式及拉式系統)的應用。在 2.1 小節中整理部分學者對於推式系統與拉式系統之定義。在第 2.2 小節中說明裴氏圖與記號圖之定義及簡化法則。在第 2.3 小節中說明文獻中裴氏圖與看板系統及生產系統的應用。第 2.4 小節說明現有的裴氏圖軟體模組，如圖 2.1 所示。

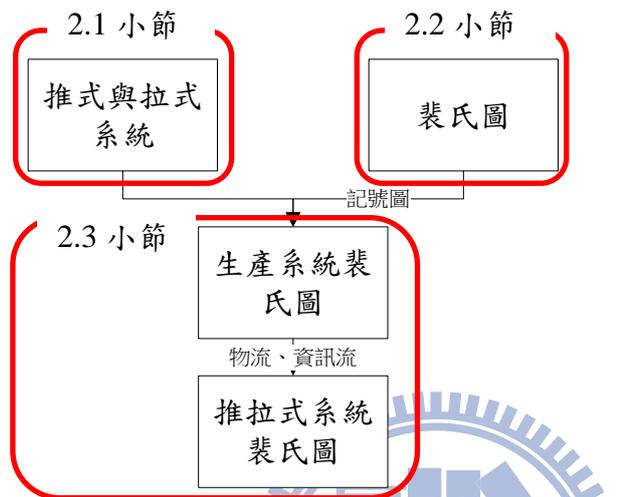


圖 2.1 裴氏圖與推拉式生產系統的文獻回顧關係

2.1 推式與拉式的生產系統

自大野耐一[43]參訪美國 Piggly Wiggly 自助零售商店後，於在 1948 年到 1975 年之間發展出及時生產系統，此系統利用看板系統作為生產依據。在此之後，生產系統開始提出在生產系統中有推式系統及拉式系統的區別。2.1.1 小節將介紹部分學者及組織所提出推式系統及拉式系統定義。2.1.2 小節說明看板系統使用及流程。

2.1.1 推式及拉式生產系統觀念的發展

在提出及時生產系統後近三十年，提出許多推式系統及拉式系統的定義。最早在霍爾在 1983 年[18]提到「推式」及「拉式」的觀念。下列由波尼等人[13]文中整理出各方學者所提出推式系統及拉式系統之定義。

李(L. C. Lee)[37]所提出的推式系統為當工件進入至生產系統時依生產優先順序生產；拉式系統為當有物料用盡的看板容器送達作業網站時才生產。文卡特許(K. Venkatesh)[48]提出在推式系統為前端機台生產零件不需要等待後端機台的需求；反之，拉式系統為前端機台只能在收到後端機台要求時才生產。

東尼(A. D. Toni)[47]試圖分析推式及拉式生產系統，將之分為三個子系統：(1)存貨子系統。(2)製造排序、物料選擇及移動子系統。(3)生產規劃子系統。而推式及拉式

區別于資訊的來源，使用了戈達德(W. E. Goddard)及布魯克斯(R. B. Brooks)[17]的定義，推式即為在市場預期的需求下生產；拉式系統意指當有需求時才生產。

利亞(F. D. Villa)[49]提出以邏輯的角度來說，推式為向後看(Look Back)；拉式為向前看(Look Ahead)。以存貨管理來說，推式是基於過去的消費來推動及決定需求；拉式為有需求時或已發出的訂單才作生產的動作。基於上述定義得可知，再訂購點(Re-Order Point, ROP)為推式存貨管理技術，物料需求規劃為拉式存貨管理技術。又說這二者定義並不是完全相反的，其中的誤解來自於存貨管理子系統及優先作業子系統。相同的來說，依推式及拉式的邏輯來說，基於預測的主生產排程(Master Production Schedule, MPS)為推式；基於訂單(Orders)的主生產排程為拉式。

利亞(A. Villa)及渡邊(T. Watanabe)[49]所描述的推式及拉式的特性為「在日式及西式的製造管理訂單中有明顯的不同。前者是著重要在於不要浪費，在生產制程中每一個網站向後“拉”著所需之專案及確切的資源。後者注重於對生產缺貨的風險管理，因此在生產制程中每一個網站需求使用合適的計畫來“推”動」。

肯沃(J. Kenworthy)及利特爾(D. Little)[36]提出「製造資源規劃(Manufacturing Resource Planning, MRP II)[12]及看板系統常被認為對立的技術，看板系統的支持者以貶抑的姿態來討論製造資源規劃的技術，意指物料被推至生產線比成為完成品來的快。但看板相反，看板為一拉式技術，只有在下一網站有而需求時才使用看板訊號(Kanban Signal)或卡片(Card)來拉物料向前走。」

波尼等人[13]提出定義是推式系統為物流及資訊流同方向，拉式系統為物流及資訊流反方向。

表 2.1 推式及拉式系統定義清單

學者或組織	定義解釋角度	推式系統定義	拉式系統定義
李	物料	直接進入生產系統	當物料用盡或看板
戈達德	市場	依預期需求	依實際需求
利亞	邏輯	向後看	向前看
利亞	著重點	不要浪費	缺貨風險管理
肯沃	生產速度	物料進入生產線較快	生產完成品較快
波尼	物流及信息流	物流及信息流同向	物流及信息流反向
BSI	在製品	有在製品	有需求時，在前置時間前生產
APICS	生產、物料控制及配銷	生產：依生產排程 物料控制：依工作訂單 配銷：統一決定	生產：看板或需求 物料控制：有需求時 配銷：依現場倉庫決定

英國標準協會制定 BS 5192[14]定義：推式系統的訂購系統是每一個零件都有在製品，訂單是當有零件由倉庫中被領走時才發出；拉式系統的訂購系統是在規定的期限前可完成之下，依預估的前置時間(Lead Time)發行訂單。例如拉式系統是基於存貨管制(Stock Control)及看板。

美國生產及存貨管理協會字典(APICS Dictionary)[35]之推式系統及拉系統定義為：推式系統定義為(1)在生產中，工件生產的需求是依事先的生產排程規劃決定。(2)在物料控制中，物料的發出是依已規劃的時間表或在一開始時依工作訂單(Job Order)需求發出。(3)在配銷中，各區域倉庫內存貨皆統一由生產現場或是中央支援設備(Central Supply Facility)決定。拉式系統定義為(1)在生產中，只有在有需求時或是有看板資訊時。(2)在物料控制中，需有來自使用者的需求，方能發出物料。(3)在配銷中，由現場倉庫決定補貨與否，而非中央倉庫或工廠決定。上述定義，依其解釋角度整理至表 2.1。

2.1.2 看板系統

由大野耐一[43]於 1988 年提出，由字面上意義為招牌(Signboard)或是看板(Billboard)。看板是為了達到及時生產的目的，而來控制生產現場的生產排程式控制制工具，依實際生產流程會有些許差異。看板的種類主要有二種為“領料看板”或“搬運看板”(Withdrawal Card)和“生產看板”(Production Card)。其功能及使用規則如表 2.2 所示。新鄉重夫(S. Shigeo)[46]提出看板功能主要有三個。(1)辨視標籤(Identification Tag)：確定產品類別。(2)工作指示標籤(Job Instruction Tag)：確定加工流程、加工時間、加工品質。(3)運送標籤(Transfer Tag)：確定何種專案及運送的起始點及目的地。

表 2.2 看板功能及使用規則[43]

功能	使用規則
“領料指令”或“搬運指令”	後一道生產流程依“看板”至前一道生產流程領貨。
“生產指令”	前一道生產流程依“看板”生產產品種類及數量。
防止“生產過量”及“運送過量”	無“看板”時，不生產、不製造。
作為“產品”證明檔	“看板”必須掛在產品上。
為“暴露問題”的工具、“庫存管理”的工具	“看板”的數量愈少愈好。

在生產製造的過程中，僅使用生產看板作為傳遞功能，而未使用取料看板(Withdrawal Kanban)者，稱為單看板系統。先由圖 2.2 定義其圖形意義。

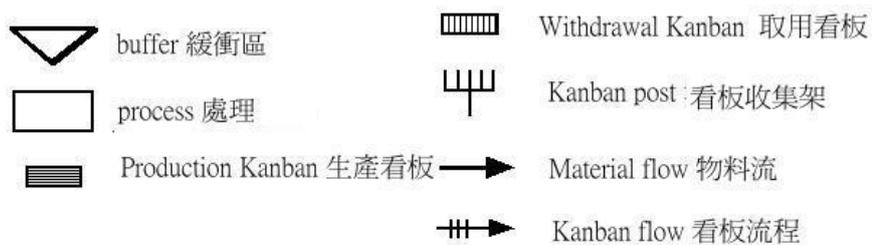


圖 2.2 看板圖示說明

圖 2.3 為生產看板之流程，圖上數位為一迴圈。數位 1 為生產流程前制程之緩衝區，數字 2 為當站制程，數字 3 為當站制程之緩衝區，數字 4、5 及 6 為看板運送方向。其流程為當站有看板送達時(數字 6)，將向前制程緩衝區領取當網站所需之原物料(或在製品)生產。領取後放入看板容器內，並將看板標示在看板容器等待生產(數字 2)，當生產完成後送至當站點的緩衝區(數字 3)等待下一站生產流程製程領料。當看板容器由下道製程領料完成後，形成空容器看板，再送(經過數字 4、數字 5 及數字 6)至前製程緩衝區領料等待生產。

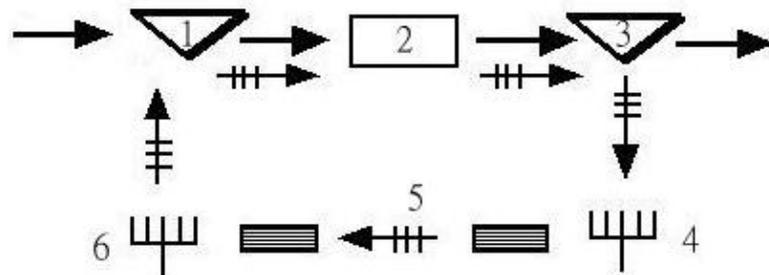


圖 2.3 單看板系統流程

雙看板系統較單看板系統多取料看板。以新鄉重夫[46]所提圖 2.4 為例，此生產系統共分三站，分別為零組件流程(Parts Processing)、零組件裝配(Parts Assembly)及最終裝配(Final Assembly)。在各站點之中前後共有二個暫存區，在製程前暫存區為 S_2 ，在製程後暫存區為 S_1 。其流程為，當最終裝配有需求時，向最終裝配的 S_2 拿取產品。若不足時，發出取料看板向零組件裝配的 S_1 取料。若 S_1 不足再發出在製品看板(Work-In-Process Kanban)要求生產，當站製程會向當站 S_2 拿取原物料(或在製品)生產，生產完成後產品及在製品看板一併運送至該站的 S_1 。若原物料(或在製品)不足時，會再發出取料看板至上一站點的 S_1 ，依此類推至前站點。即意為物料看板皆在二製程站點之流動，而在製品看板則在製程站點內中製程及暫存區之物料流動。

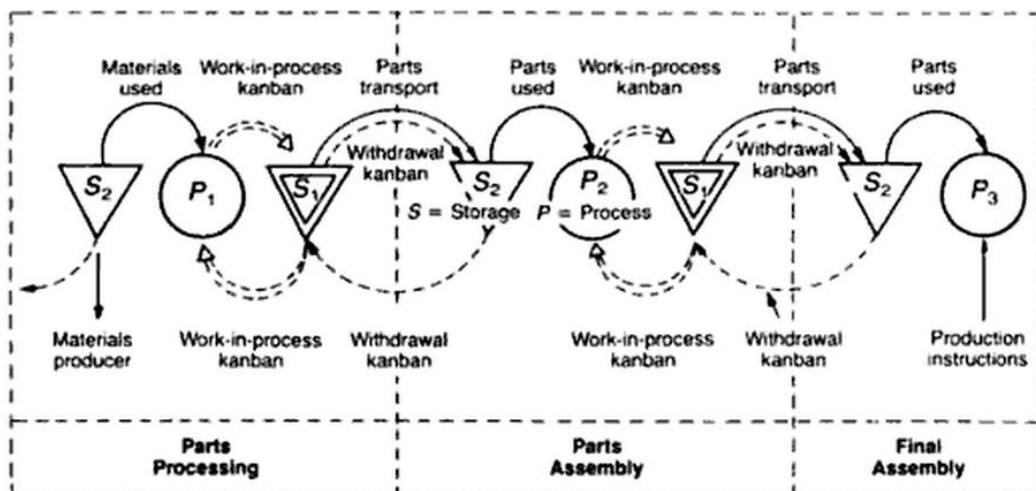


圖 2.4 雙看板系統流程[46]

2.2 裴氏圖模型

裴氏圖[26]為1962年由西德的裴氏(C. A. Petri)所發表。一開始主要目的是在於描述化學流程。此圖形化的表達工具，已在許多離散事件系統的模式構建與分析研究中被應用著。裴氏圖可用於製造系統中活動與邏輯方面的敘述，包含了事件的同步性、事件因果性以及資訊的競爭與共享性。在2.1.1小節將介紹裴氏圖的基本元件及定義。2.1.2節說明裴氏圖簡化的簡化法則。2.1.3節明可達圖的表達法。2.1.4節說明鎖死問題的分類。2.1.5小節說明系統不變量的計算。2.1.6節說明週期時間分析的發展。2.1.7節說明裴氏圖可加註語言。2.1.8節說明偵查法則。

2.2.1 裴氏圖與記號圖

在最初的裴氏圖中不包含時間(Time)的概念，主要包含了五個部分，可依下列正式符號加以定義為 $PN = \{P, T, I, O, M_0\}$ 。

暫存點(Place)是以一空心圓圈 \bigcirc 做為表示。暫存點在圖形中代表物件(例如機台或是工作)的狀態。其代表符號如下所示：

$P = \{p_1, p_2, \dots, p_n\}$ ，其中 P 是一個有限的集合， $n \geq 1$ ， n 代表 PN 中暫存點的數量。

轉移點(Transition)以一粗型線段 \blacksquare 作為表示。在圖形中通常用於系統中事件的發生。例如起始加工事件，工作完成加工事件等。

$T = \{t_1, t_2, \dots, t_m\}$ ， T 為一有限的集合， $m \geq 1$ ， m 代表 PN 中轉移點的數量。

在暫存點及轉移點之間使用有向弧作為連結，下列 I 表示由暫存點指向轉移點，稱為轉移點的輸入弧(Input Arc)。相反的， O 表示由轉移點指向暫存點的，稱為轉移點的輸出弧(Output Arc)。其定義為 $I(p_i, t_j) : (P \times T) \rightarrow N$ ， $O(p_i, t_j) : (P \times T) \rightarrow N$ 。

暫存點內有浮標(Token)，以一實心圓 \bullet 作為表示， M_0 用來表示裴氏圖的起始狀態(Initial State Marking)。存有浮標的暫存點表示系統正處於此暫存點的狀態；反之若無浮標之暫存點表示系統並未處在此暫存點的狀態之中。每一暫存點可同時設置多個浮標，表示暫存點狀態可使用之資源數，若有一暫存點有 K 個浮標，代表有 K 筆項目或是資源可供利用。

在動態模式執行方面，裴氏圖藉由轉移點的激發(Firing)，進而改變各個暫存點中浮標的分佈狀況，以達成系統的動態描述。轉移點激發的規則為：在轉移點前所有的暫存點皆標記至少有一個以上的浮標點時，則稱此轉移點為可被激發(Enabled)。轉移點被激發後則暫存點中的浮標將會轉移。在一般的裴氏圖中有向弧的權重(W)為1，廣義裴氏圖(Generalized Petri Net)則帶有正整數的權重設定。轉移點的激發由例2.1表示之。

例 2.1：氫氣燃燒生水[42]

氫氣燃燒生水的化學方程式為 $2H_2 + O_2 \rightarrow 2H_2O$ ，可使用圖 2.5 表示，在未激發前為圖 2.5(a)有二個暫存點，分別為 H_2 及 O_2 ，各內含二個浮標。暫存點 H_2 至轉移點 t 之

權重為 2；暫存點 O_2 至轉移點 t 之權重未註明為 1。因滿足激發規則，經轉移點 t 激發後可得圖 2.5(b)，產生二個 H_2O ，暫存點 H_2O 內以二個浮標表示之。

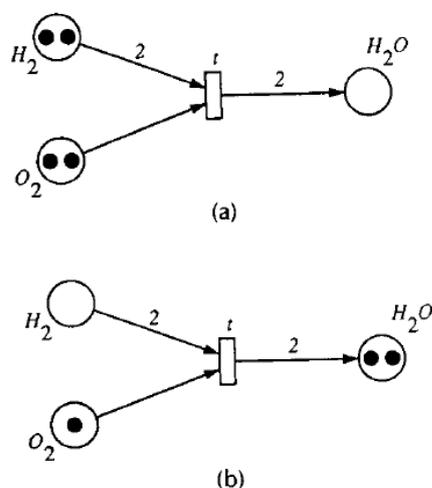


圖 2.5 轉移點激發法則[42]

裴氏圖是同時擁有數學且圖形性質之工具。基本性質中又可依初始狀態(Initial Marking)而分成兩大類[42]：(1)行為性質(Behavioral Properties)與(2)結構性質(Structural Properties)。行為性質為裴氏圖的初始狀態有關，如可達性(Reachability)、活性(Liveness)等性質的探討。結構性質指與裴氏圖的初始狀態無關的性質，包含了安全性(Safeness)、限制性(Boundedness)、浮標不減性(Conservativeness)、可逆性(Reversibility)與一貫性(Consistent)。

裴氏圖的分析方法上可分別由生產週期時間、可達圖、鎖死問題、不變量等。當裴氏圖的複雜度提升時，不易由圖形上判斷裴氏圖是否有鎖死現象，鎖死問題即成為在初步設計裴氏圖時相當重要的一門學問。在系統運作時，可利用浮標點的轉移來作為鎖死的判斷。但又可能因裴氏圖的流程過於複雜時，不易判系統所處狀態，因此裴氏圖可轉換至可達圖來描述系統的可達狀態。

記號圖[42]為裴氏圖中所有暫存點只有一個輸入弧及一個輸出弧稱之。其數學定義為 $\forall p \in P : |p \bullet| = |\bullet p| = 1$ 。

2.2.2 簡化法則

村田(J. Murata)[42]整理出數個簡化法則，主要為了使複雜的系統模型簡單化，但仍保留系統中原先的特性，如活性、安全性、限制性，如圖 2.6 所示。圖 2.6 中(a)表示合併連續的暫存點；(b)合併連的轉移點；(c)合併平行的暫存點；(d)合併平行的轉移點；(e)消除自循環的暫存點；(f)消除自循環的轉移點。

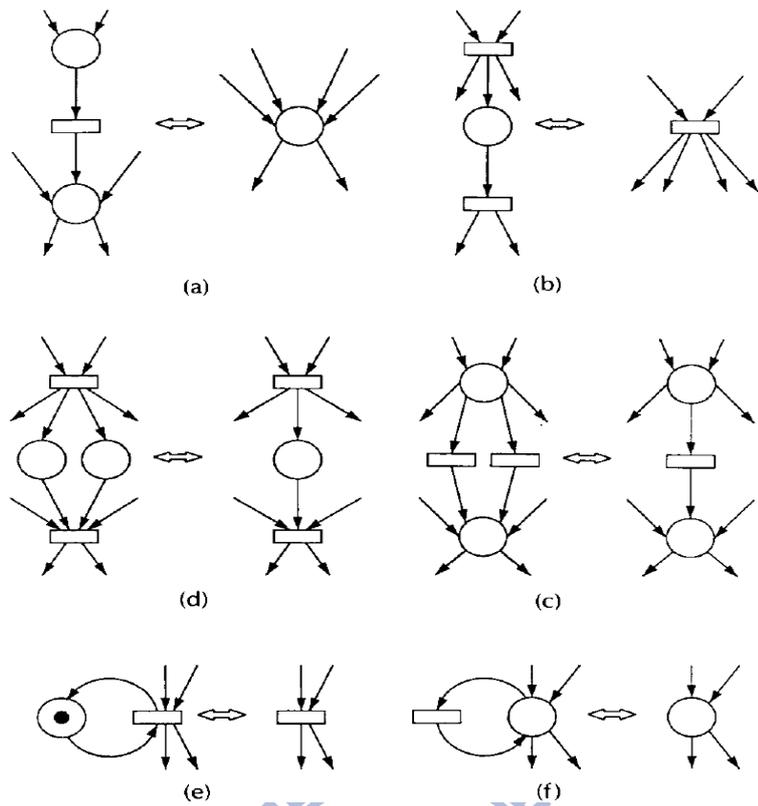


圖 2.6 簡化法則[42]

2.2.3 可達圖表達法

狀態的可達性質可使用可達圖作為表達，可達圖為一有向圖。一開始先繪出所選定系統裴氏圖之初始狀態，再將全系統中所有可能的狀態繪出。在分析系統性質，繪製裴氏圖的可達圖，為最基本的方法。

例 2.2：液體加熱系統[5]

液體加熱系統裴氏圖如圖 2.7 所示，暫存點及轉移點列表如表 2.3 所示。

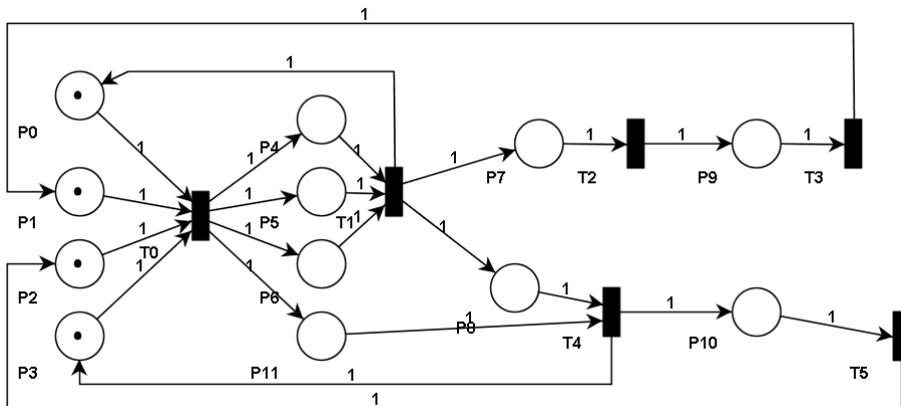


圖 2.7 液體加熱系統[3]

表 2.3 液體加熱系統暫存點及轉移點說明

暫存點及說明		轉移點及說明	
P0	Valve-2 on	T0	液體準備流入 Bessel-b2
P1	Vessel-a1 液面狀態為 full	T1	液體流入 Vessel-b2
P2	Vessel-b1 液面狀態為 full	T2	空的 Vessel-a1 準備補充液體
P3	Valve-1 off	T3	Vessel-a1 補滿液體
P4	Vessel-a1 液面狀態為 decreasing	T4	Vessel-b2 中之液體由 20 °C加溫至 100 °C後準備流出 Vesel-b2
P5	Vessel-b1 液面狀態為 decreasing	T5	100 °C之高溫液體流出 Vessel-b2 且流出後空的 Vessel-b2 恢復常溫 20 °C
P6	Valve-1 on		
P7	Vessel-a1 液面狀態為 empty		
P8	Vessel-b1 液面狀態為 empty		
P9	Vessel-a1 液面狀態為 increasing		
P10	Vessel-b1 液面狀態為 increasing		
P11	Valve-2 off		

以例 2.2 液體加熱系統裴氏圖為例，轉置為可達圖如圖 2.8。以本圖可說明此液體加熱系統共有 S0 到 S9 十種狀態。在起始狀態 S0 下，只能有轉移點 T0 可激發。最後可經由轉移點 T3 或是 T5 的激發後回到起始狀態 S0。

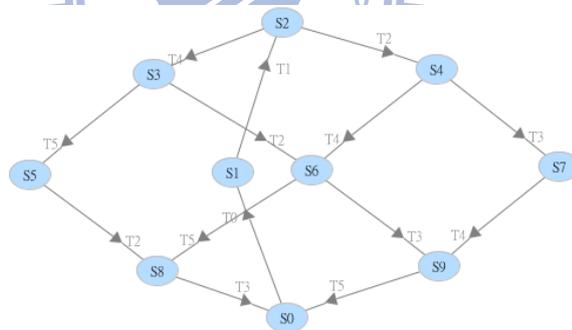


圖 2.8 液體加熱系統可達狀態

2.2.4 鎖死問題

鎖死現象(Deadlock)[50]指在製造系統中，二個或多個的工件彼此佔住所需資源。並等待其他工件釋放資源，但本身卻不釋放資源，使得此二個或多個工件無法完成。舉例說明，當有二個工程師 A 與 B，同時需要二種資源(筆及尺)才能工作。但資源只有一只筆及一把尺，且分別在 A 與 B 的手中。A 與 B 又不願釋放資源(筆及尺)給對方，亦無法正常工作，此時即為鎖死現象。

鎖死問題的解決方法一般可分為鎖死排除(Deadlock Prevention)、鎖死偵測與復原(Deadlock Detection and Recovery)及鎖死避免(Deadlock Avoidance)[50]。鎖死排除為將鎖死狀態去除之。鎖死偵測與復原為將已發生鎖死狀態復原至鎖死前狀態。鎖死避免

為使之不進入鎖死狀態。在裴氏圖的鎖死問題中，指無法再到達其他狀態下。可達圖表達法是可以解決鎖死問題的其中一種方法。

2.2.5 系統不變量

在系統長期動態運作下，可能會因為人為疏忽或是機器故障而導致系統失效或停止運作。因此在系統動態分析上會以不變量(Invariant)[3]作為基礎，不變量是以數學分析為基礎用來表達系統狀態的一組聯立方程式。利用兩種不變量可計算出聯立方程式作為監控動態系統。不變量是一組可表達系統狀態性質的聯立方程式，主要分為暫存點不變量(P-Invariant)[42]及轉移點不變量(T-Invariant)[42]。不變量只和系統有相關，不會隨著時間改變。因此不論在任何時候，系統的狀態皆需符合不變量。可利用此系統特性來偵察系統。當系統架構不變時，不變量改變，即表示發生故障，因此不變量可偵察系統是否處於正常運作的狀態。

對裴氏圖而言，其行為可使用狀態方程式(State Equation)[42] $P' = P + Rt$ 來表示。其中 P' 為未來暫存點的欄向量， P 表示目前暫存點的欄向量， R 表示(監控)法則矩陣(Rule Matrix)， t 為激發的轉移點。其公式推導如下。在狀態方程式同乘 w ，得 $P'w = (P + Rt)w$ ，再展開為 $P'w = Pw + Rtw$ ， $P'w = Pw + Rtw$ 表示不變，即 $0 = Rtw$ ，又 t 為常數，即可得 $0 = Rw$ ，用此方式推導即可得暫存點不變量方程式。同理 $R^T w = 0$ 可得轉移點不變量[42]。

2.2.6 週期時間分析

在週期時間分析的發展中，拉馬默西(C. V. Ramamoorthy)及加里(S. H. Gary)[45]定義其週期時間為最小循環中的最大值。裴氏圖可分有二類，為一致裴氏圖(Consistent Petri Net)及非一致裴氏圖(Inconsistent Petri Net)。在一致裴氏圖之下又分決策自由裴氏圖(Decision-Free Petri Net)、安全持續裴氏圖(Safe Persistent Petri Net)及一般裴氏圖(General Petri Net)。在文中定義出一致性系統(Consistent System)為在轉移點上輸入的浮標點數目等於輸出的浮標點數目，反之為非一致性系統(Inconsistent System)。而決策自由裴氏圖定義同於記號圖[42]，即為每個暫存點只有一個輸入及一個輸出。安全持續裴氏圖內的暫存點同時有多個輸入、輸出。一般裴氏圖為當有一個轉移點觸發時，就有另一個不會被觸發，在決策自由裴氏圖有一項性質為在迴路(Cycle)中浮標數必相等。再定義其最小週期時間，此篇是以列舉法作為計算公式。當安全持續裴氏圖時，可先轉換決策自由裴氏圖，其計算方法由列舉法可得之。

在馬戈(J. Magott)[39]針對每一個轉移點之間的時間差公式來計算最小週期，將時間裴氏圖(Timed Petri Net, TPN)轉換成裴氏圖後，使用暫存點不變量的觀念推導出下界公式 2(Lower Bounds 2, LB2)。並證明比拉馬默西及加里提出的下界公式 1(Lower Bounds 1, LB1)[45]好。

在盛岡(S. Morioka)及山田(T. Yamada)[41]提出在裴氏圖最小週期問題可以轉換成

線性規劃求解。在裴氏圖中的暫存點內的浮標點數量為流量，暫存點以有向弧取代，轉移點轉化為節點，即為網路流量形。如此一來等同於線性規劃，可使用線性規劃軟體求解。在線性規劃方面，使用最小開樹的概念作計算。

坎波斯等人(J. Campos, G. Chiola and M. Silva)[15]使用轉移點及轉移點之間的時間差，找出各轉移點正確被觸發的時間。以對偶(Dual)的角度去作計算，可使限制式大量減少。

2.2.7 裴氏圖可加註語言

裴氏圖可加註語言[29]是基於可加註語言[33][11]而發展出來的。可加註語言最早是由全球資訊網聯合會(World Wide Web Consortium, W3C)[31]所定義。其目的在於建立一套電腦資訊交流的標準語言。XML1.0 於 1998 年二月正式成為標準規範。可加註語言的前身是標準通用加註語言 (The Standard Generalized Markup Language, SGML)[11]，是自 IBM 從 1960 年代就開始發展的通用加註語言 (Generalized Markup Language, GML)[11]標準化後的名稱。可加註語言主要用途有三個，(1)豐富檔案(Rich Documents)為自定檔案描述並使其更豐富。(2)後設資料(Metadata)為描述其它檔案或網路資訊。(3)設定檔案(Configuration Files)為描述軟體設定的參數。可加註語言格式如圖 2.9 所示。

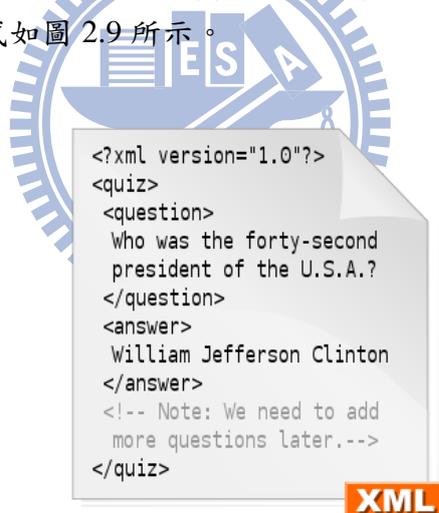


圖 2.9 XML 格式[11]

在裴氏圖的發明及可加註語言的發展之下，電腦交換資訊皆可基於可加註語言來發展。在發展軟體中，需要共同的格式以方便電腦資訊的交換。ISO/IEC[30]為國際標準化組織(International Organization for Standardization, ISO)[27]及國際電工委員會(International Electrotechnical Commission, IEC, 或譯國際電工協會)[25]，在一開始 ISO/IEC 15909 Part-1 先定義基本的裴氏圖及高階裴氏圖(High-Level Nets)。在 ISO/IEC 15909 Part-2 定義了以可加註語言為基礎的裴氏圖可加註語言[29]。預計在今年(2011年)會再發表 ISO/IEC 15909 Part-3 來定義有關裴氏圖家族的擴展性(Extensions)及變異性(Variations)。其重要的發展歷史由表 2.4 作為說明。

表 2.4 裴氏圖格式發展整理

年代	發表者	說明
1962 年	裴氏	Petri Net 的發明。
1996 年	W3C	可加註語言發明。
2004 年	ISO/IEC	15909-1：定義裴氏圖及高階裴氏圖。
2009 年	ISO/IEC	15909-2：定義裴氏圖交換格式裴氏圖可加註語言。
2011 年(預計)	ISO/IEC	15909-3：定義有關裴氏圖家族的擴展性及變異性。

在 2009 年定義完成後，希拉等人(L. M. Hillah et al.)[20]即提出建立新的應用程式介面(Application Programming Interface, API)。其核心模組如圖 2.10 所示。利用裴氏圖可加註語言架構(PNML Framework)，即可用在不同裴氏圖工具(Petri Net Tool)上的裴氏圖資料(PNML Document)作匯入(Import)及匯出(Export)，如圖 2.11 所示。

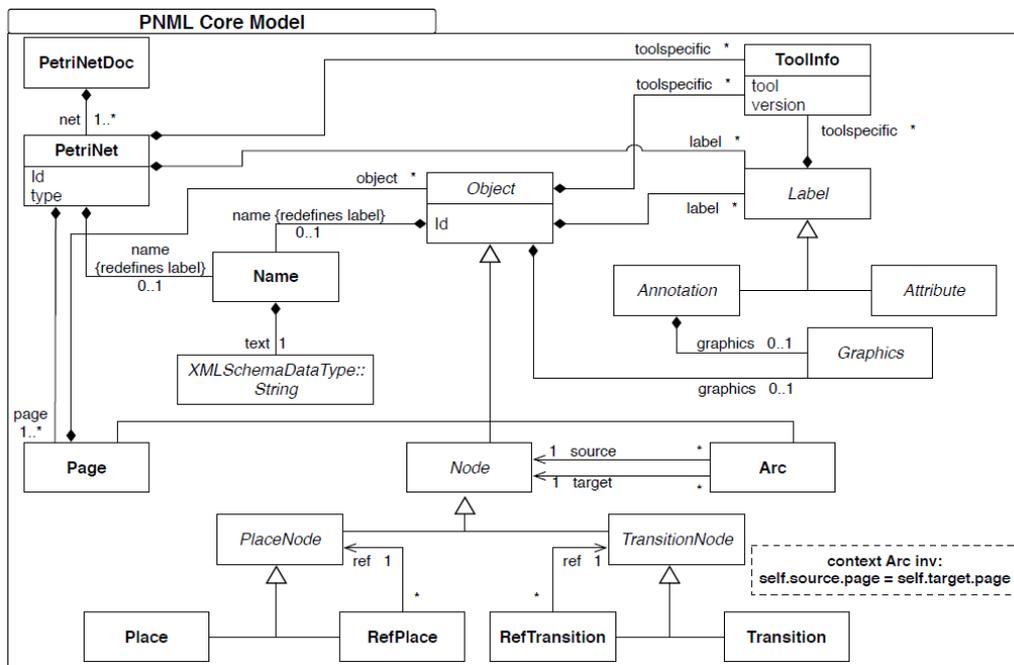


圖 2.10 裴氏圖可加註語言核心模組[20]

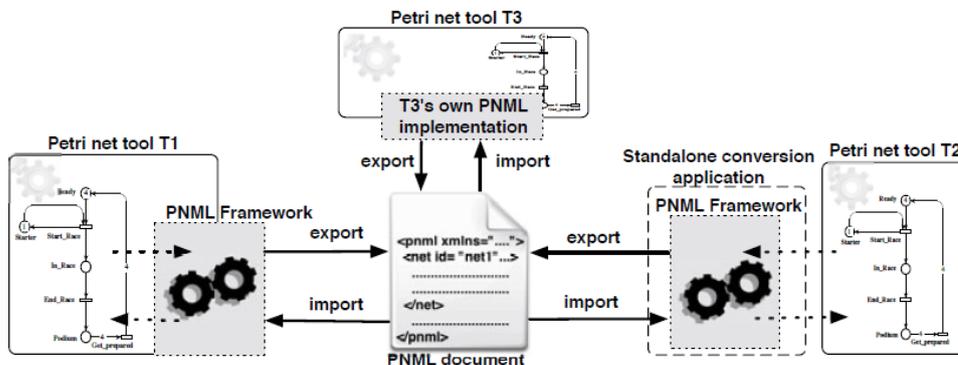


圖 2.11 裴氏圖文件資料交換[20]

2.2.8 偵查法則

梁高榮[6]提出可畫圖擬陣中，有二種方法可找出其偵查法則。第一是直接由可畫圖擬陣轉成可表達擬陣(Representable Matroid, RM)間接求解，第二是由其對偶形中找出答案。以例 2.2 為例，圖 2.12(a)中為液體加溫系統裴氏圖，可用實線圖來表達如圖 2.12(b)，之後將 P3 及 P4 合併，可得圖 2.12(c)。如此一來共有 11 個線段(暫存點)，又基底可知秩為 4(即浮標總數量)，再繪出對偶圖為虛線圖共有 8 個節點，因此秩為 7。依實線圖的基底可建構對偶圖的基底並使用粗虛線代表，可得額外增邊 P2、P6 及 P7。由實線圖的基底代入浮標後可得偵察法則 $P2 - P9 = 0$ ， $P6 - P9 = 0$

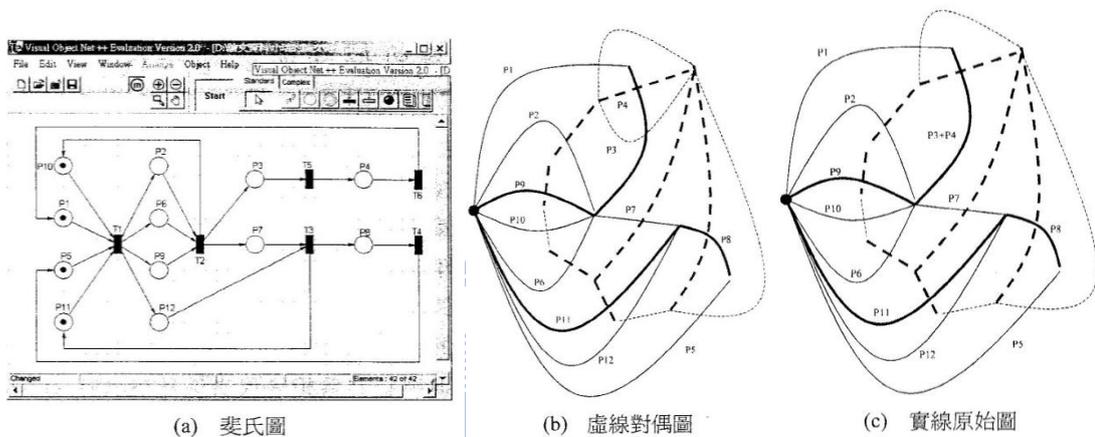


圖 2.12 子擬陣技術設計偵查器[6]

2.3 裴氏圖與生產系統

因裴氏圖同時具有圖型及數學性質，可應用在生產系統上。在第 2.3.1 小節中，將說明馬斯科洛等人提出裴氏圖應用在看板系統的例子。2.3.2 小節將介紹利用裴氏圖及推式系統與拉式系統之間的關係。

2.3.1 裴氏圖應用於看板系統

馬斯科洛等人[40]提出了在看板系統上可應用裴氏圖。在看板系統中，以圖 2.13 為例，將生產流程系統以圖表示。由原物料(Raw Materials)投入後，分別經過站點(Stage 1)及站點 2(Stage 2)後，再經過站點 3(Stage 3)，最後依序經過站點 4(Stage 4)及站點 6(Stage 6)完成後成為產品 1(Finished Products 1);或是站點 3 出來後亦可依序經過站點 5(Stage 5)及站點 7(Stage 7)完成後成為產品 2(Finished Products 2)。在每一個站點都包含二個部分，分別為製程流程(Manufacturing Process)及輸出暫存區(Output Buffer)。在製程流程中又包括了在製品等待及在製品加工。輸出暫存點包括了本站點完成的在製品。

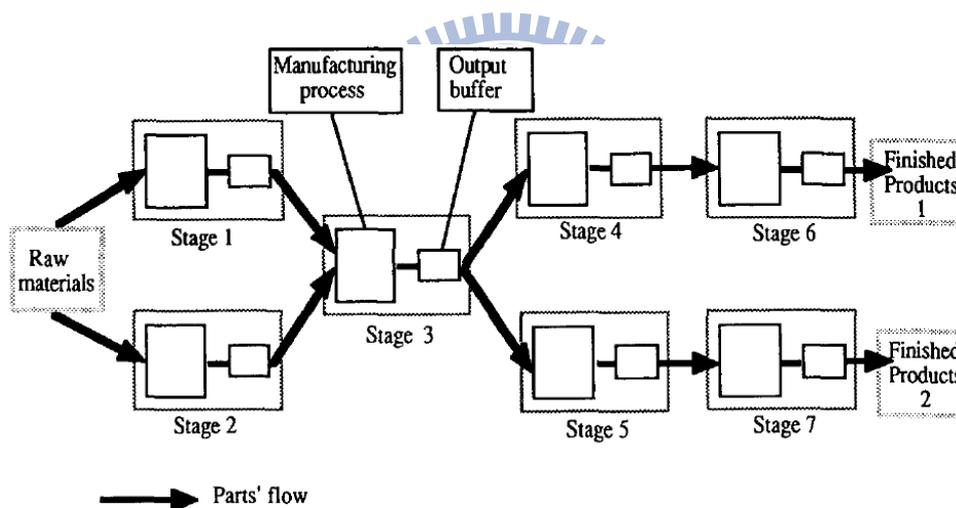


圖 2.13 生產系統以站點表示[40]

在圖 2.14 表示在每一個站點的前一道站點及後一道站點之間的關係。站點 i (Stage i) 表示生產流程中的某一站點，在站點 i 前一生產製造站點稱為站點 $i-1$ (Stage $i-1$)，完成站點 i 後下一製造站點稱為站點 $i+1$ (Stage $i+1$)。符號說明共有四個，空的看板容器 (Free Kanban)、滿載的看板容器 (Part Kanban)、看板流程 (Kanbans' Flow) 及在製品流程 (Parts' Flow)。先由站點 $i-1$ 加工完成後載滿的看板容器運送至站點 i ，站點 i 取出由站點 $i-1$ 來的看板容器內的在製品加工，拿取物料完成後看板會再回到站點 $i-1$ 。而站點 i 加工完成的在製品會放在自站點 $i+1$ 傳送過來的看板容器中，再運送回站點 $i+1$ 加工。以此類推至所有生產流程站點。

將圖 2.14 之看板系統，以三站點為例繪至成基本裴氏圖模型，如圖 2.15 所示。 I_i 是第 i 站點生產狀態， I_i 內部的浮標點數代表第 i 站點生產數量。 O_i 為生產完成後的暫存

點(Output Buffer)，其浮標數為該站點生產完成後數量。 F_i 內的浮標數表示空白看板容器數量。 N_i 內的浮標點數表示機台或是資源數量。轉移點 T_i 為站點 i 及站點 $i+1$ 之同步(Synchronization)動作。轉移點 W_i 為站點 i 為生產動作，延遲時間為 d_i ， d_i 亦為站點 i 的生產時間。需求設定方面為暫存點 T_D 為有需求之狀態， d_D 為產生需求。

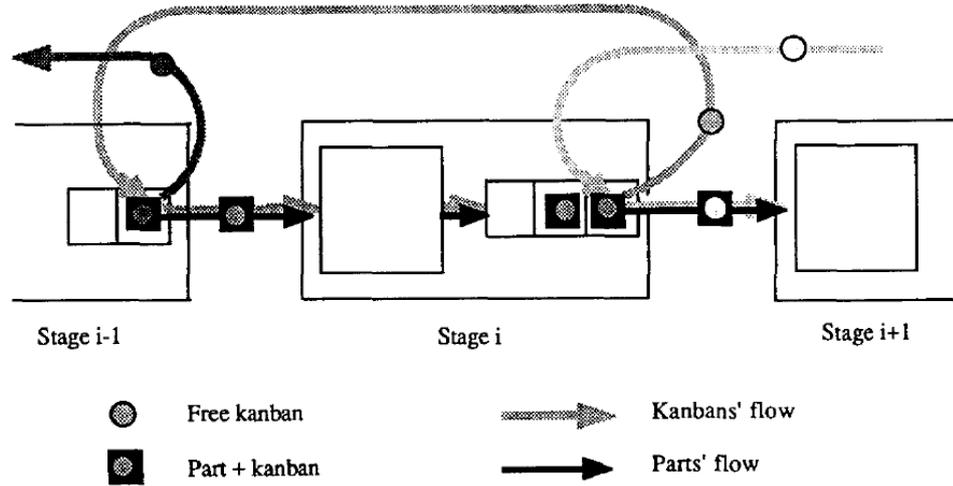


圖 2.14 前後站點之間的關連性[40]

圖 2.15 裴氏圖動態流程為先由 d_D 被激發後產生需求暫存點 D 狀態，同時若暫存點 O_3 有空的看板容器將激發轉移點 T_3 ，將運送至前一站點生產。同時激發轉移點 T_2 ，一方面向將看板運送至上一站點(站點 2)，另一方面開始生產(站點 3)，依此方式類推至向前製程站點 2，同理的方式向站點 1 拉貨。拉到站點 1 時，可由暫存點 R 提供原物料，以便進行生產動作。

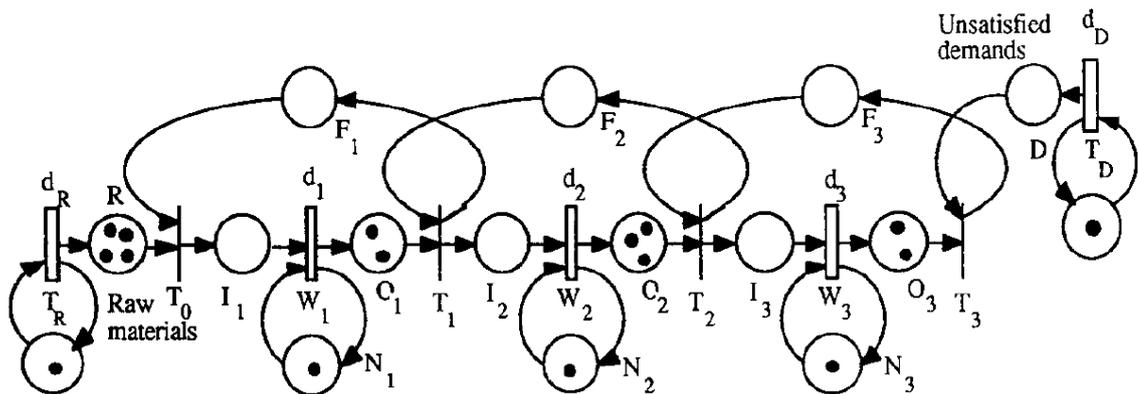


圖 2.15 三站點看板系統[40]

在圖 2.15 繪出看板系統裴氏圖後可假定需求、供給無限的狀態下，可將生產流程中原物料供給及需求產生的暫存點及浮標點加以簡化，如圖 2.16 所示。

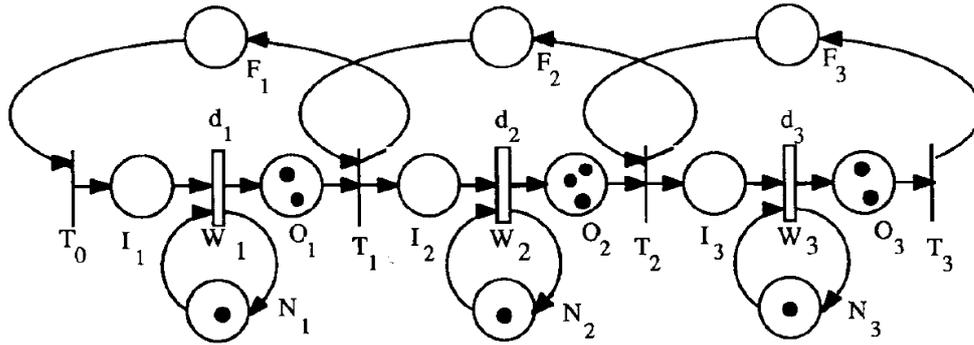


圖 2.16 三站點看板系統(需求、供給無限)[40]

2.3.2 裴氏圖應用於推式及拉式系統

波尼等人[13]在比較推式系統及拉式系統差異時，使用了裴氏圖的作為分析工具。並以一生產流程作為範例。在此推式系統及拉式系統的生產流程皆相同，由原物料進入生產系統後，依序經過三個生產步驟，分別為原物料取得(Materials Procurement)、組件生產(Parts Production)及裝配(Assembly)，最後再到訂單流程(Order Processing)，如圖 2.17 及圖 2.18 所示。裴氏圖繪製依生產流程由上至下，方向由左至右，訂單來源皆在左下方。

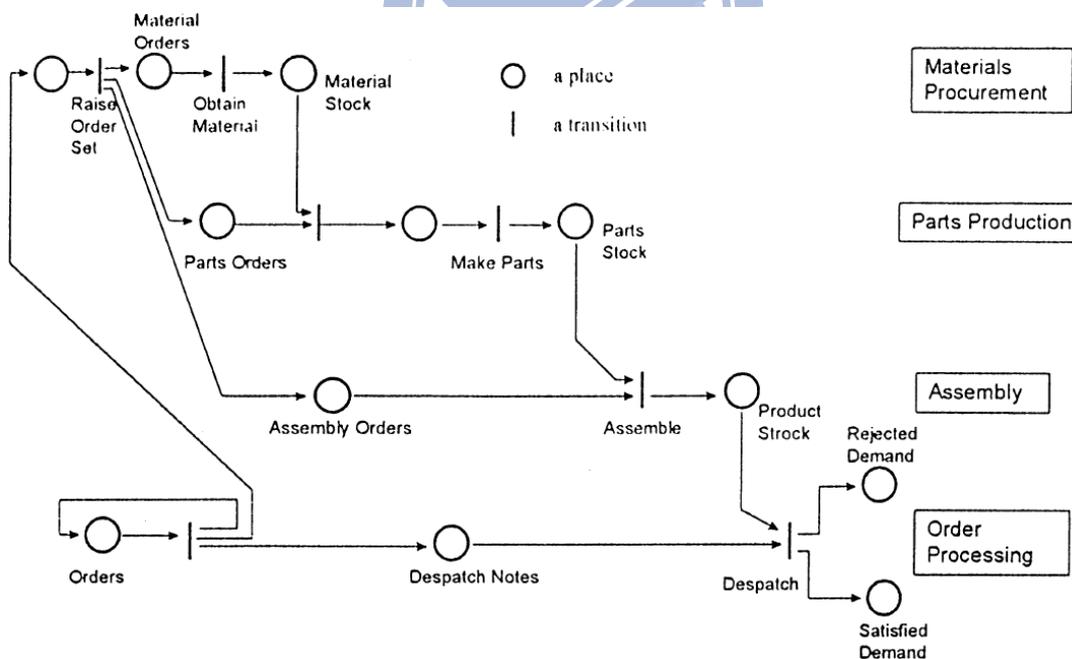


圖 2.17 推式系統[13]

圖 2.17 為推式生產系統裴氏圖，其流程為當(左下)訂單來時，激發轉移點，發出三道生產命令分別至各生產步驟及發運單(Dispatch Notes)至轉移點發運(Dispatch)，同時原物料運送至暫存點物料取得。激發轉移點取得物料，至暫存點物料存貨狀態。有

了物料及生產命令後，轉移點生產組件開始運作。同理再經由組件存貨，激發轉移點裝配，至產品存貨，和發運單一並出貨。若有滿足需求至暫存點滿足需求(Satisfied Demand)，若無滿足需求至暫存點不滿足需求(Rejected Demand)。

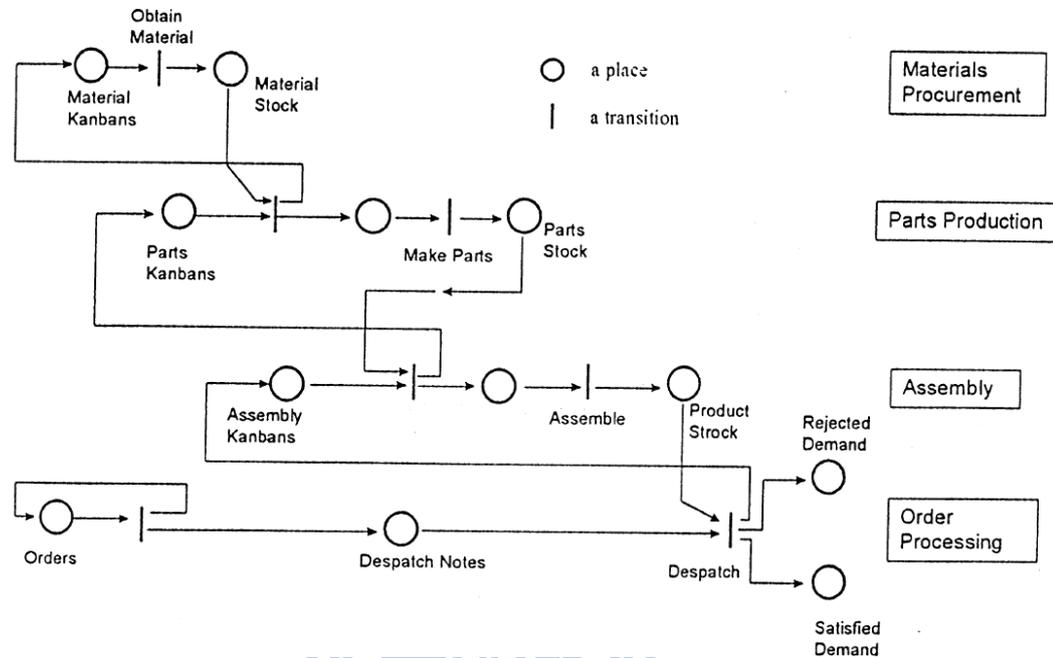


圖 2.18 拉式系統[13]

圖 2.18 為拉式系統裴氏圖，其流程為當訂單來時，只發出發運單至裝配站點，由裝配站點依產品存貨發出裝配看板(Assembly Kanbans)至裝配站，同理，裝配站會依組件存貨數量發出組件看板(Parts Kanbans)，最後向物料發出物料看板(Material Kanbans)，由原物料取得站點去向供應商採購。

2.4 整合性裴氏圖模組

在本小節將說明裴氏圖相關之已開發之模組。製造系統的設計與實作中規格、驗證、撰碼、測試、偵查，各別由范植宇[2]完成規格階段、林潔妤[1]完成測式階段及陳音帆[7]完成驗證與撰碼階段。在 2.4 小節中共分為 3 個小節，在 2.4.1 小節說明由 IDEF0 轉換到裴氏圖可加註語言模組。2.4.2 小節說明裴氏圖基本性質分析模組。2.4.3 小節說明裴氏圖路徑分析模組。

2.4.1 IDEF0 格式轉換至裴氏圖可加註語言格式模組

范植宇[2]利用 Java 程式語言以及開放資料伺服器連接架構的規格建置一個裴氏圖可加註語言導向的開放資料伺服器連接軟體。使用者可藉由此模組將 IDEF0[24]軟體商業套裝軟體 BPwin[21]所提供的 IDL 檔案格式將其格式轉換為裴氏圖可加註語言格式。范植宇[2]的系統研究架構中主要可分為數三個步驟，「樹形概念」、「文件物件模型(Document Object Model, DOM)[32]設計」及「程式設計」。因 IDL 檔案格式中並非基於可加註語言所建立的

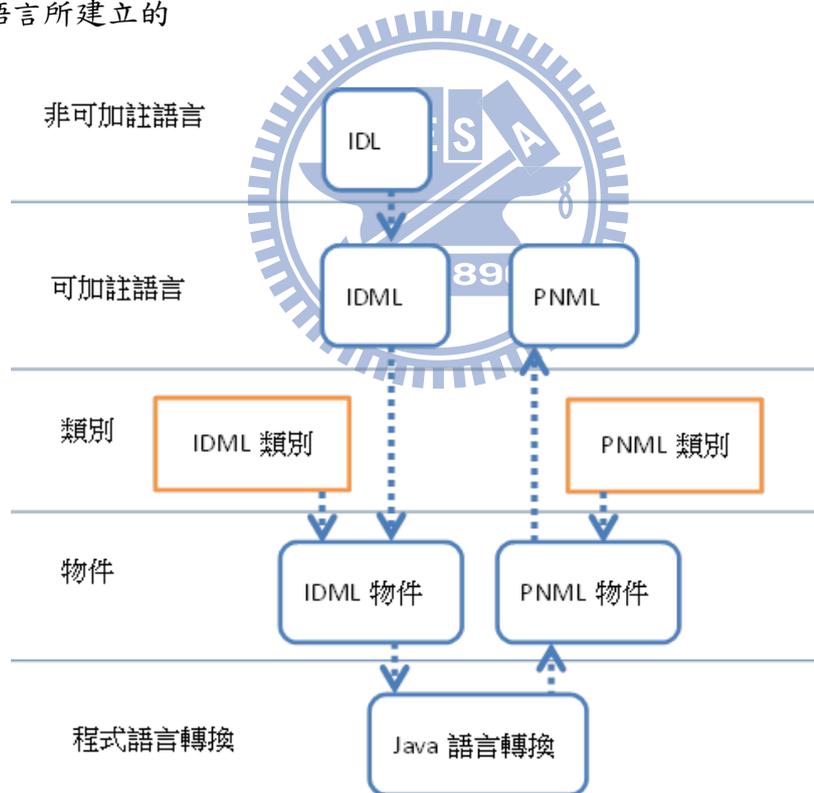


圖 2.19 IDEF0 格式轉換至裴氏圖可加註語言格式流程

第一步驟「樹形概念」為研究 IDML[2]及裴氏圖可加註語言結構樹之間的架構後，找出其樹形轉換法。

第二步驟「文件物件模型設計」利用樹形轉換的概念將 IDML 樹與 PNM 樹轉換成文件物件模型設計

第三步驟「程式設計」利用 Java 語言建立 IDML 類別及裴氏圖可加註語言類別。其轉換如圖 2.19 所示。

2.4.2 裴氏圖基本性質分析模組

陳音帆[7]在基本性質分析模組中，主要有五個子模組。(1)轉移點不變量分析、(2)暫存點不變量分析、(3)週期時間分析、(4)基本性質分析及(5)鎖死分析。基本性質分析可分析出模型規模及規模性，鎖死分析可分析其狀態及鎖死區域，轉移點不變量及暫存點不變量可各別找出其不變量之數量及其狀態方程式。週期時間分析可計算其最小週期，但其裴氏圖需為記號圖方能計算。

2.4.3 裴氏圖自動化規範測試

在林潔妤[1]的自動化規範測試系統中包含四個模組。分別為活可達圖(Live Reachability Graph)模組、歐氏有向圖模組、測試序列模組、循環分析模組。

可達圖為一有向圖，可用來描述系統內所有的可達狀態。因此可使用旅途法(Transition Tours)來得到可達圖的測試序列。若可達圖具有歐氏有向圖的特性時，可使用弗勒里演算法(Fleury's Algorithm) [1]求得此可達圖的一筆畫路徑。若不行，可使用中國郵差問題[1]，藉由增加額外的重複弧線讓可達圖變成歐氏有向圖。

模組功能中，活可達圖模組可讀取利用 INA(Integrated Net Analyzer)[34]軟體產生的可達圖檔，找出鎖死狀態並利用鎖死排除方法將之去除，並利用歐氏有向圖模組解決具有中國郵差問題的可達圖，再使用測試序列模組找出歐依勒路徑及狀態間轉移所需激發的轉移點。最後再利用生產循環分析模組列出所有的生產循環。

第三章 生產系統的推拉式指標設計

本章將說明如何以資訊流記號圖為主，來設計推拉式系統指標。首先分別以裴氏圖繪出物流、推式系統及拉式系統後，藉由二者記號圖的不同定義推拉式系統指標。定義完成後可依其迴路數及暫存點的性質以建立指標。最後再比較二指標之間的差異。在 3.1 小節說明建立模型流程。3.2 小節推導出推式系統的記號圖。3.3 節說明拉式系統的記號圖。3.4 節以迴路數導向建立推拉式指標。3.5 節以暫存點導向建立推拉式指標。3.6 節以複雜生產系統的案例作推式系的指標分析，各節之間的關係圖 3.1 所示。

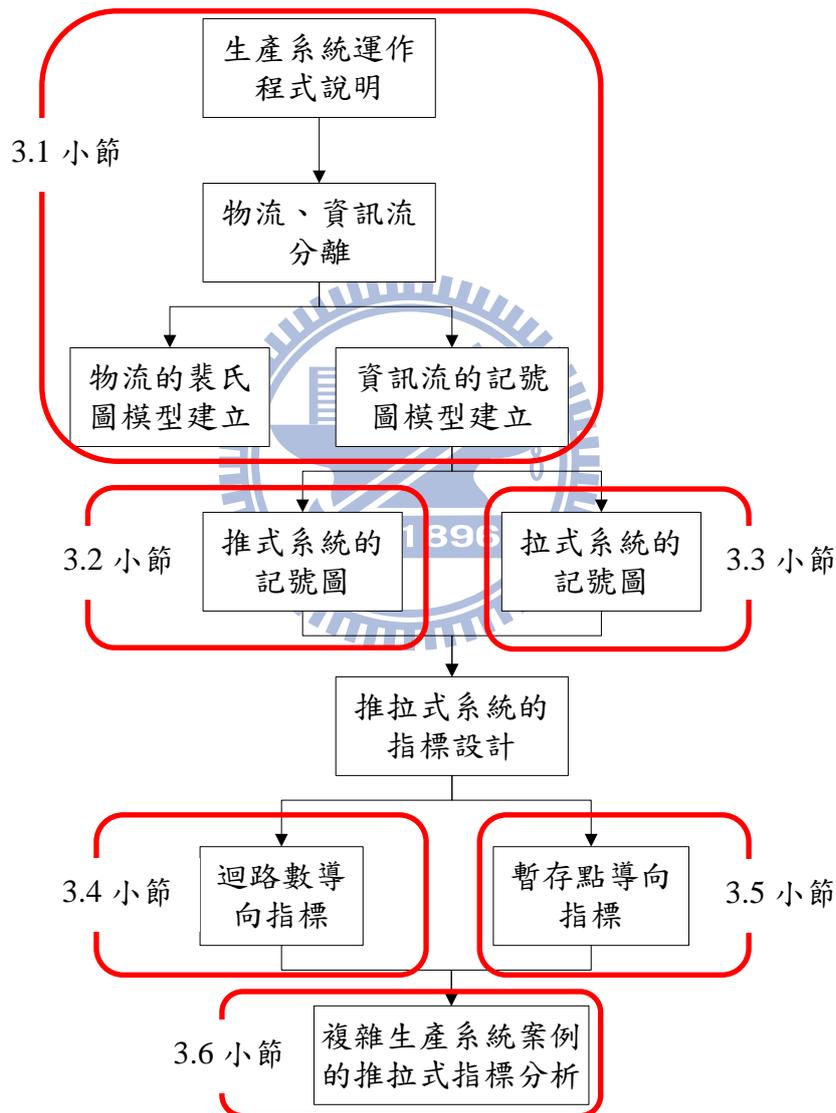


圖 3.1 推拉式指標的建立流程

3.1 推拉式指標分析流程

設計推拉式指標前，需先定義推式及拉式系統。本節利用波尼等人[13]的案例提

出推式記號圖及拉式記號圖，將物流及資訊流分離後，再說明資訊流與記號圖之間的相關性。

3.1.1. 物流與資訊流的分離

本論文中使用狹義的生產物流定義，指生產流程中原物料輸入至生產系統中經由生產後變成產品，最後運送至客戶手中的實體分配。資訊流亦狹義的定義為，在生產系統流程內為滿足生產條件所需的資訊交換。資訊流的流量大小，表示在生產流程中所需之資訊多寡。在此論文中不去定義資訊的流量大小問題，只判斷是否有資訊流的存在。

波尼等人[13]比較推式系統及拉式系統差異時，在同一生產系統之中因推式系統及拉式系統特性而建立二個不同的裴氏圖(圖 2.17、圖 2.18)。在這二張裴氏圖中，在同一暫存點皆同時包含了物流(Physical Distribution)及資訊流(Information Flow)活動，這會造成下列問題。(1)物流與資訊流同用暫存點，將造成定義不明，無法正確的得知正確的物流及資訊流。(2)實際生產線上，物流主要皆為由供應商提供原物料。經一系列的生產流程後交貨給客戶，為一單方向性的流程。浮標數會因新增原物料而增加浮標數，產品生產完成出貨後，浮標數會再減少。如此一來，整個系統的浮標數會變的不穩定，而難以分析。(3)物流即產品(在製品)為一實體，資訊流只有資訊(非產品)，沒有實體。二者本質差異太大。因此需將生產系統中的物流及資訊流作分離的動作。為表示資訊流及物流之間的差異，物流使用粗線，資訊流使用虛線表示，其分離方法如圖 3.2 所示。

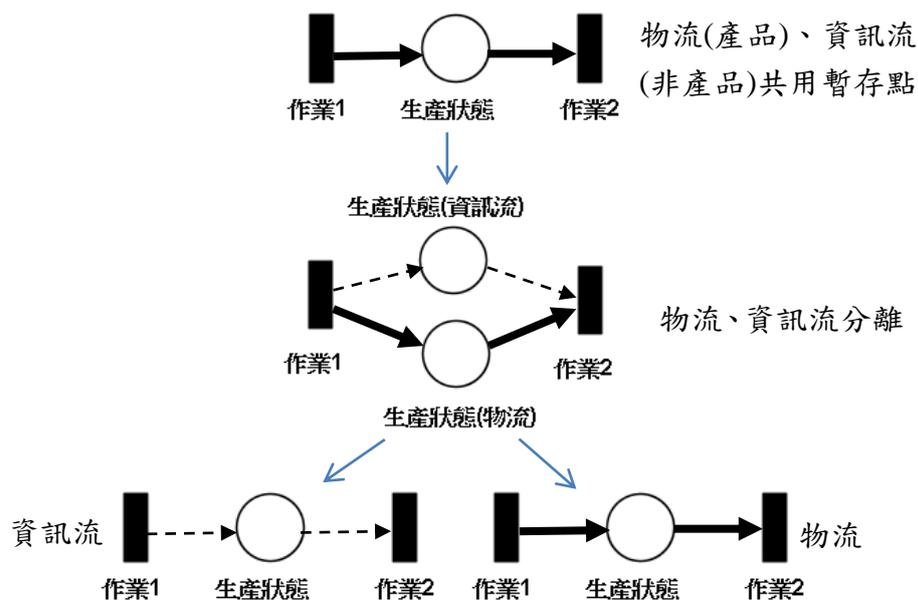


圖 3.2 物流與資訊流分離

3.1.2. 資訊流與記號圖的關係

物流方向不會因推式系統或拉式系統的不同而有所改變。在建構純推或純拉式系統時會以資訊流為主。定義裴氏圖中包含三個元素，暫存點 ○表示生產系統資訊流的狀態，轉移點 ■表示生產系統的動作及生產系統狀態的改變。在轉移點及暫存點之間的有向弧 → 表示轉移點激發後狀態改變的方向及可激發轉移點的方向。暫存點內有浮標 ●，表示目前所在的生產系統狀態。粗線有向弧表示物流方向，細線有向弧表示資訊流方向。

本論文只考量在一般情況下，只能有一個下達生產命令的部門(即生產規劃部門或是下游站點)。暫存點為資訊流狀態，因此暫存點只會有一個輸入(資訊流)及一個輸出(資訊流)。生產系統裴氏圖不論推式或是拉式皆為記號圖。

波尼等人[13]所使用的裴氏圖中尚有一項缺失。資訊流只有單方向由訂單經過生產流程後直到客戶或由客戶直接到生產作業，並無反向的資訊流動。但在生產流中應有資訊的交流，資訊流的方向應為雙向。在有回饋的資訊流下，生產作業才能進行修正，否則將會出現重大異常。在所有資訊流中除了生產命令外，需再新增一暫存點，其方向相反於生產命令。所有的生產命令皆會成為一迴路。此迴路以實線表示，其修正如圖 3.3 所示。

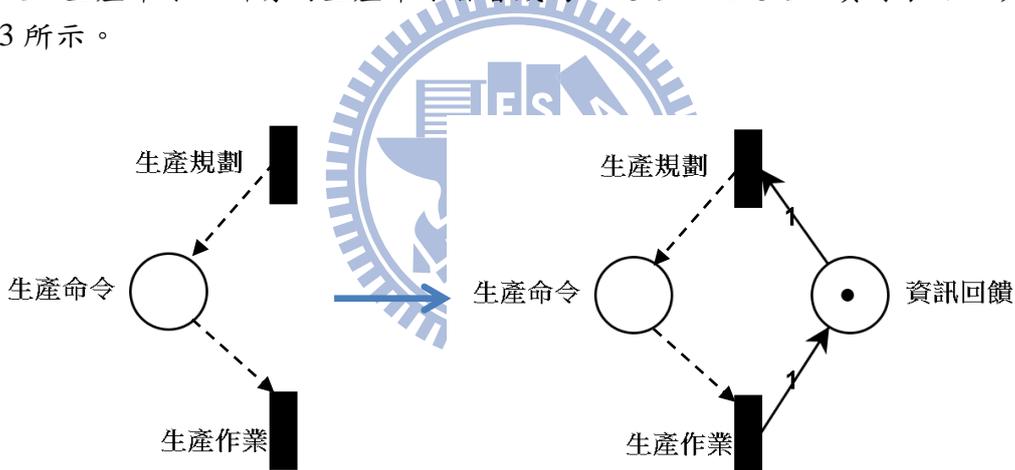


圖 3.3 生產資訊流回饋(資訊流)

3.1.3. 修正記號圖步驟

表達生產系統資訊流記號圖，需修正除上述二小節以外，尚有暫存點內未包含初始浮標。因此需修正三個部分，如圖 3.4 所示。

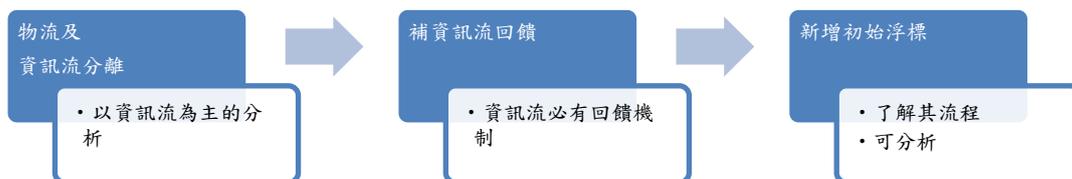


圖 3.4 修正生產裴氏圖流程

3.2 純推式系統的記號圖表達法

為使推式系統應用於記號圖上，本論文以圖 2.17 例子作延伸，推式系統為當有訂單(或預測)到達時會發出三道生產命令至各站點，分別為原物料取得(Materials Procurement)、組件生產(Parts Production)及裝配(Assembly) [13]，此時記號圖只能激發原物料取得站點。當原物料取得站點完成時，將物料送至組件生產站點，組件生產站點才能被激發。同理，當組件生產站點完成並運送至裝配站點才能被激發。

訂單、預測需求、物料(在製品)的到達或是其他生產流程上游的資訊，而激發生產動作為推式系統，可分成二個類型討論。第一類型為中央控制型。當生產規劃下達生產命令(資訊流)及物流(在製品)到達作業站點時生產。第二類型為地區控制型。物料及資訊流的到達時激發生產(資訊流及物流同行)。底下以例 3.1 說明第一類型，而例 3.2 說明等二類型。

例 3.1：修正圖 2.17 範例-1

令其生產站點同圖 2.17。由圖 2.17 修正三點，(1)物流及資訊流暫存點分離。(2)新增資訊流回饋。生產流程依序為原物料取得、組件生產及裝配。所有生產作業皆由生產規劃下達生命命令後才能生產，如圖 3.4 所示。

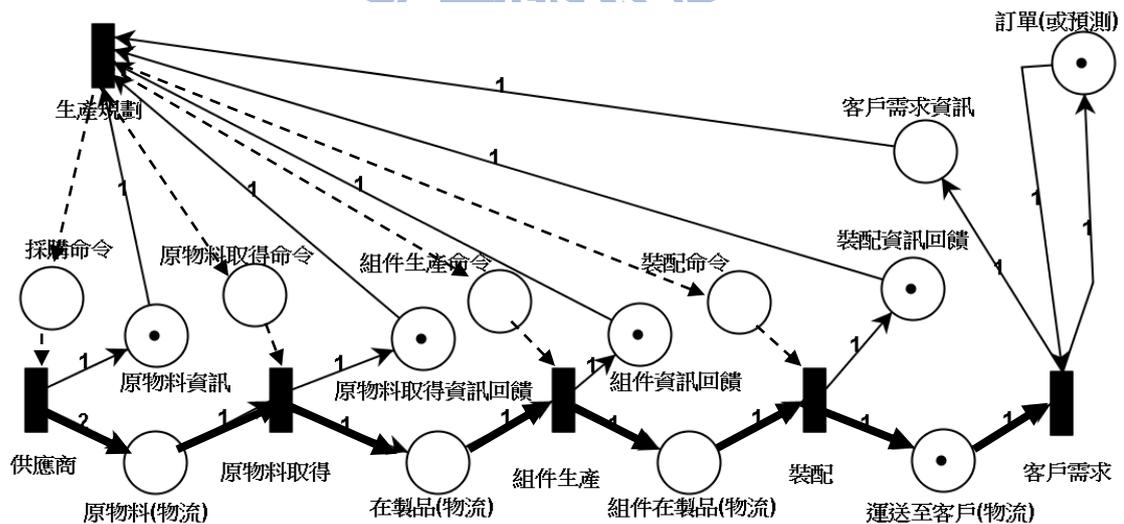


圖 3.5 例 3.1 生產流程的物流及資訊流

例 3.1 第一類型中當有訂單、預測需求時，生產規劃需將生產流程中各作業的資訊(包含人力、機台數、機台產能、機台情況、存貨、在製品、良率等)的資訊加以收集至生產規劃部門，決定原物料的訂購數量及下達生產命令至各流程，生產命令的內容主要包含所需生產產品種類、數量。

生產流程圖轉換至裴氏圖中，生產規劃為一動作。因此令生產規劃及所有生產作業為轉移點，資訊流為暫存點，資訊流的方向為有向弧的方向。去除物流暫存點後，推式系統記號圖會再將資訊流的回饋附加至各站點並簡化之。

圖 3.4 可分拆為資訊流圖 3.6 及物流圖 3.7。圖 3.6 內包含六個轉移點，包含了供應商、生產規劃、原物料取得、組件生產、裝配及客戶需求，分別代表其生產規劃動作及生產作業。當訂單來時，生產規劃需了解目前生產線的物料狀態，才能決定生產排程。因此暫存點訂單(或預測)而激發客戶需求。客戶需求資訊會傳送至生產規劃，原物料取得資訊回饋、組件生產資訊回饋與裝配資訊回饋(即各生產流程資訊)，其他生產作業亦需傳送各作業的資訊回饋。最後由生產規劃向各作業下達生命命令及採購命令。

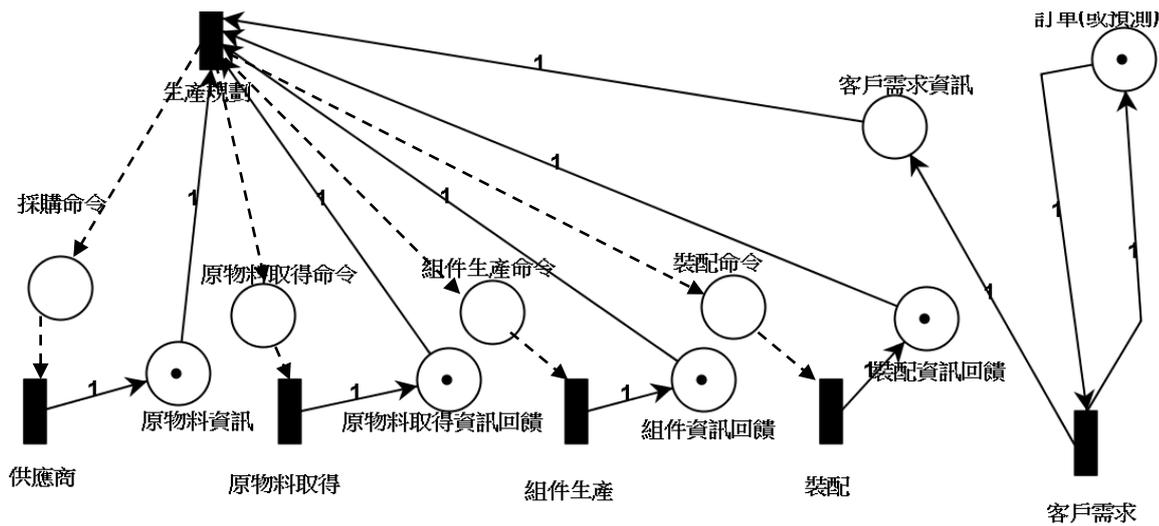


圖 3.6 例 3.1 純推式系統的資訊流

發出生產命令至各生產流程作業，資訊流狀態變更為暫存點原物料取得命令、組件生產命令及裝配命令。此三個暫存點各別可激發轉移點原物料取得、組件生產及裝配作業，當三個轉移點激發後，即生產完成。資訊流狀態會變回到暫存點原物料取得資訊回饋、組件生產資訊回饋與裝配資訊回饋，等待下一次的訂單。

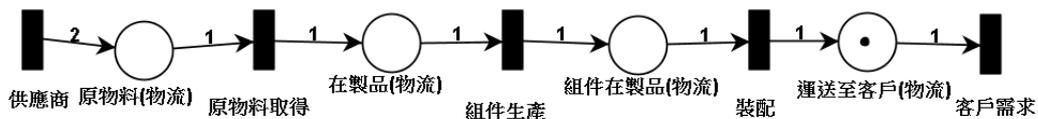


圖 3.7 例 3.1 純推式系統的物流

再以圖 2.16 為例，將其設為需求及供給無限。去除記號圖內的暫存點訂單，亦同圖 2.6(e)的簡化法則合併平行暫存點，不會使得記號圖的性質改變，記號圖只展示出生產流程如圖 3.8 所示。圖 3.8 為第一種純推式生產系統記號圖。

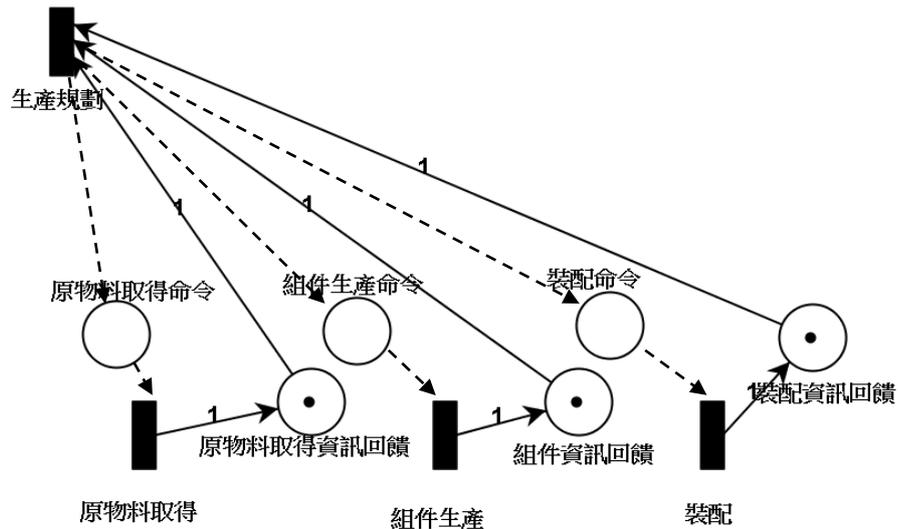


圖 3.8 例 3.1 純推式系統的資訊流

第二類型的推式系統，其生產流程相似於上述的生產流程。當有預測或是客戶需求來時，生產規劃部門會依生產種類及數量，對生產部門下達生產命令。只是在生產流程的各站點中，不須由生產規劃部門下等生產命令。當原物料取得完成時，自動運送至組件生產，組件生產作業收到由原物料取得完成的在製品時，亦自動生產並運送至裝配作業。裝配作業拿到組件生產完成的在製品後直接進行裝配作業，最後再回傳完成品的數量及種類的相關生產資訊至生產規劃部門，如圖 3.8 所示。

例 3.2：修正圖 2.17 範例-2

令其生產順序同圖 2.17。修正 3 個部分，(1)物流及資訊流暫存點分離。(2)新增資訊流回饋。(3)所有生產作業當有物料進入時，立即生產，無需等待生產命令，如圖 3.9 所示。

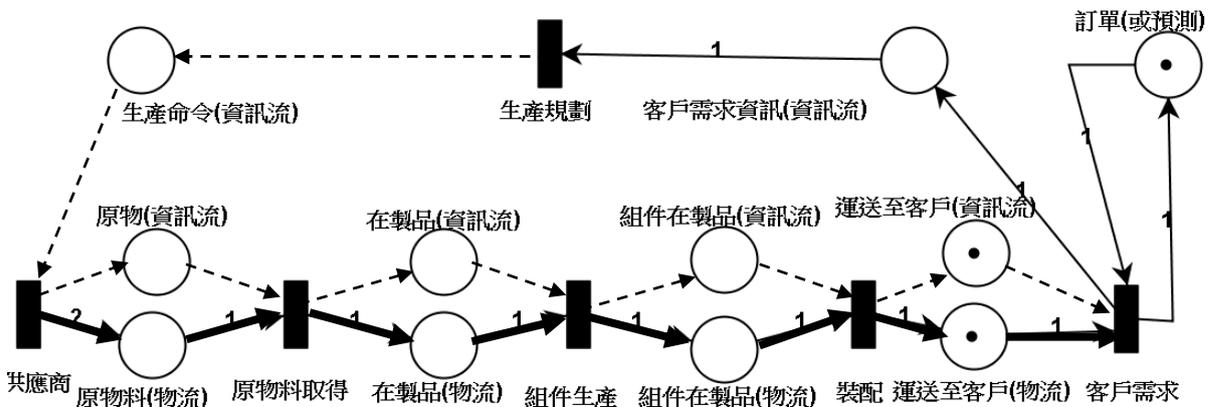


圖 3.9 例 3.2 生產流程的物流及資訊流

將圖 3.9 的資訊流及物流分離，資訊流記號圖為圖 3.10，物流裴氏圖同圖 3.7。例

3.2 資訊流流程為當訂單來時，確認目前的產品種類及數量後，傳送至生產規劃。生產規劃將決定需投入之物料數。

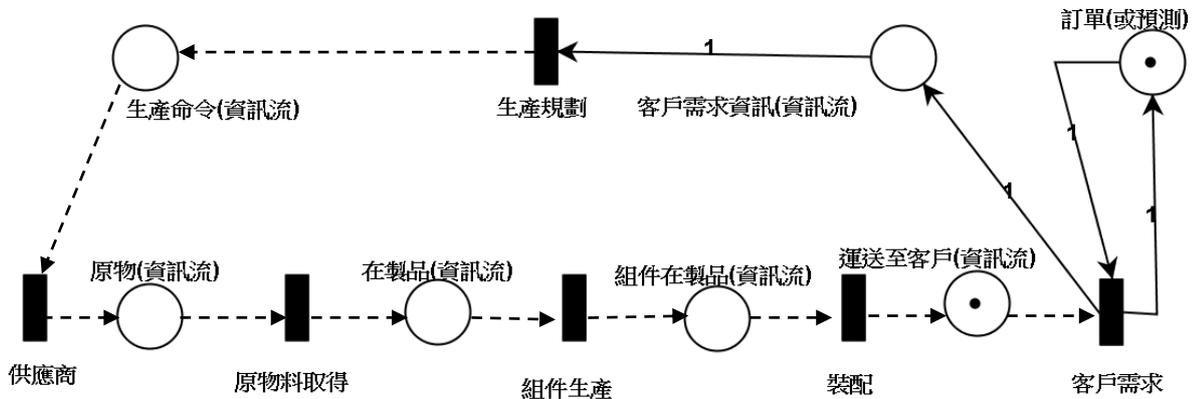


圖 3.10 例 3.2 純推式系統的資訊流-1

在圖 3.10 當有暫存點訂單(或預測)時，與收集生產流程中資訊即暫存點裝配資訊回饋(存貨、機台數、在製品等)，可激發轉移點生產規劃。生產流程狀態將改變至暫存點原物料取得命令，之後可激發轉移點原物料取得。原物料取得完成後，物流夾帶著資訊流，生產流程狀態為暫存點組件生產命令並運至組件生產，激發轉移點組件生產。同理，激發轉移點組件生產後，生產流程狀態為暫存點裝配命令，激發轉移點裝配。最後裝配完成後，會有資訊回饋至生產規劃，物料將運送至消費者。使用圖 2.16 方法及圖 2.6(e)的簡化法則，亦可將暫存點訂單簡化，如圖 3.11。

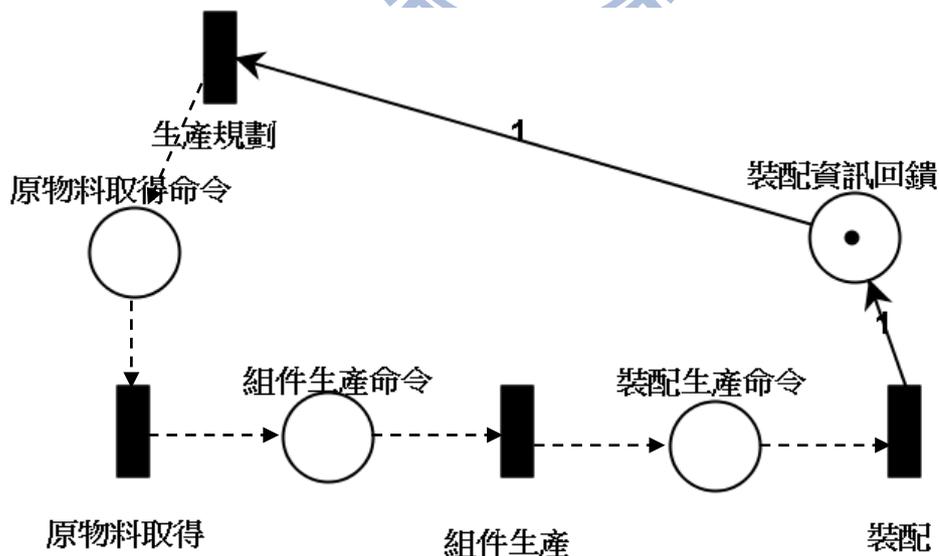


圖 3.11 例 3.2 純推式系統的資訊流-2

在圖 3.11 中，已假設供給及需求無限。在實務上圖 3.11 應適用於連續生產 (Continuous Production)或是大量生產 (Mass Production)。其資訊流記號圖明顯為一迴

路。裝配資訊回饋向生產規劃傳送目前產品種類及數量。生產規劃下達原物料取得命令(採購)，依序激發轉移點原物料取得。經由組件生產命令，激發組件生產。再由裝配命令，激發裝配。最後裝配資訊回饋再傳回資訊至生產規劃。此記號圖為一迴路，可利用圖 2.6(a)的簡化法則為簡化合併平行暫存點，因此可將所有生產流程作業合併，如圖 3.12 所示。

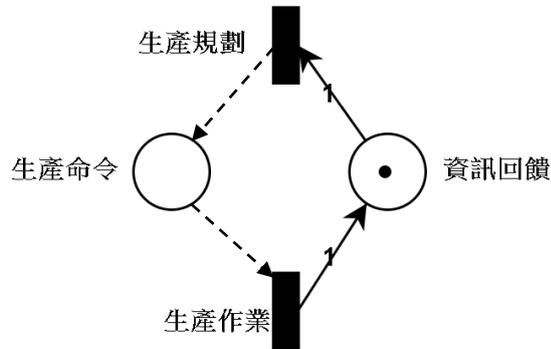


圖 3.12 純推式系統(N 個站點)

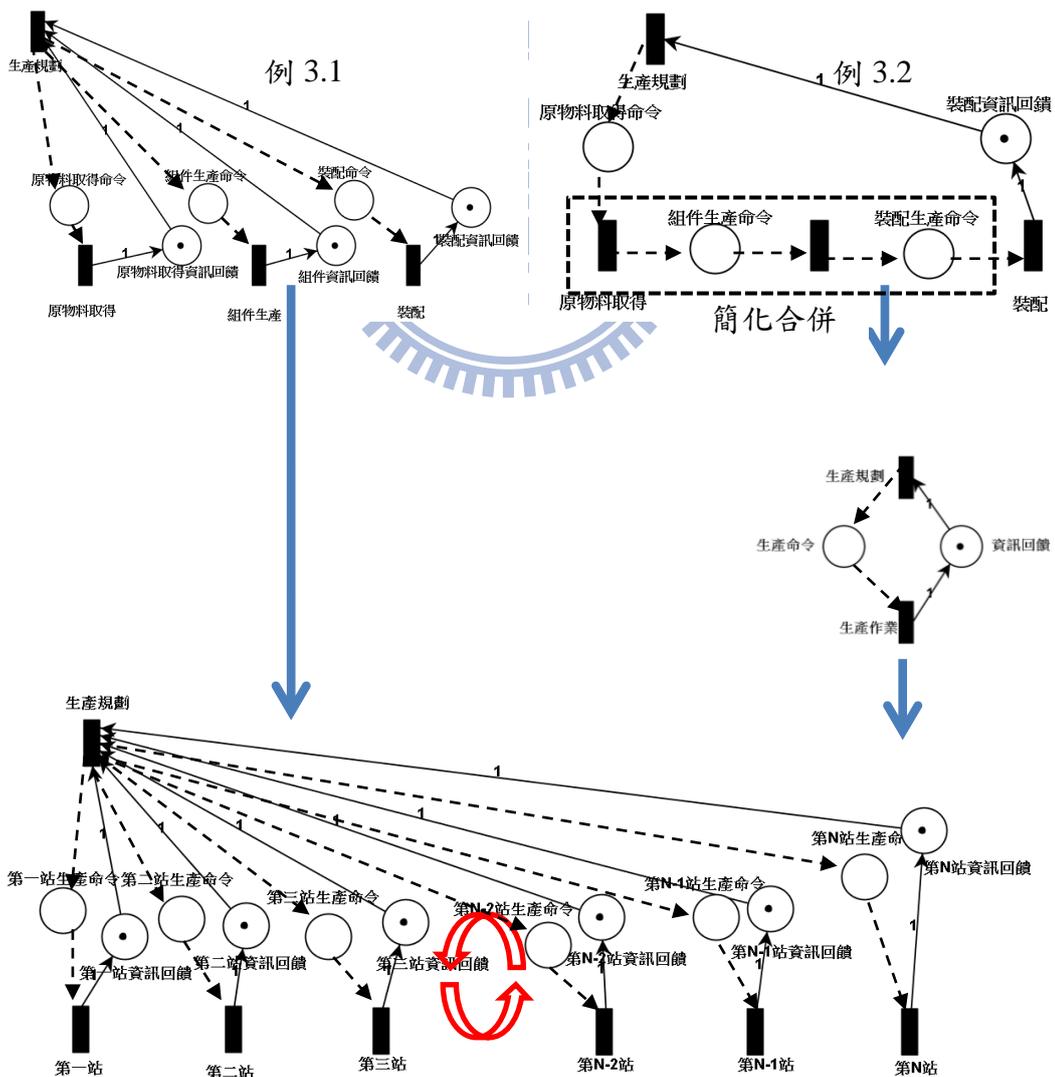


圖 3.13 純推式系統生產

圖 3.11 可簡化至圖 3.12。簡單來說即為資訊回饋後由生產規劃下達生產命令。在上述二種生產流程皆為推式系統，因生產命令來源不同使得資訊流流動方向不同。但由記號圖簡化的動作後，明顯可看出，在此二種推式系統最出皆為收到訂單(或預測)後由生產規劃下達生產命令。在生產作業完成後，資訊流(生產資訊回饋)亦會再回到生產規劃，等待下一次的訂單(或預測)。

雖例 3.1 及例 3.2 皆為相同的推式系統，因生產命令下達的資訊流而不同，而造成二種不同的記號圖。但例 3.2 簡化成圖 3.12 中，與例 3.1 的單作業記號圖皆相同。由轉移點生產規劃提出暫存點生產命令，激發轉移點的作業站點，再由暫存點資訊回饋回到轉移點生產規劃。因此可說明例 3.1 及例 3.2 本質為相同的。依此特性，可得知若有生產規劃，且與各站點之間有資訊流的往來即為推式系統。可推得假定有一系統，共有 N 個站點，其記號圖應有 $N+1$ 個轉移點(N 個作業站點及 1 個生產規劃轉移點)， $2N$ 個暫存點(N 個生產命令暫存點及 N 個資訊回饋暫存點)界於各作業站點轉移點及生產規劃之間。在資訊流相對物流的流向為一逆時鐘方向，如圖 3.13 所示。



3.3 純拉式系統的記號圖表達法

在豐田式生產管理中，設定了零存貨的目標，使用看板系統作為生產依據。以例 3.3 為例，當客戶產生需求時，生產流程中的最後一個裝配作業索取產品，裝配作業中看板容器內的產品被取走時，會再回到組件生產等待組件生產作業加工。組件生產作業會依看板上所標示之內容生產所需之物料及數量，再運送至裝配作業作最後的產品裝配。在組件生產將組件生產完後的在製品運送至裝配作業的同時，組件生產的看板容器亦會清空。此時看板容器會運至原物料取得作業，原物料取得作業收到空的看板容器，立即開始作業，直到裝滿看板容器為止。若原物料不足時，再向供應商下訂單。

例 3.3：修正圖 2.18

令其生產順序同圖 2.18。修正二個部分，(1)物流及資訊流分離。(2)新增資訊流資訊回饋。其生產流程依序為原物料取得、組件生產及裝配。所有生產作業需有訂單或需求產生時才能生產，如圖 3.14 所示。

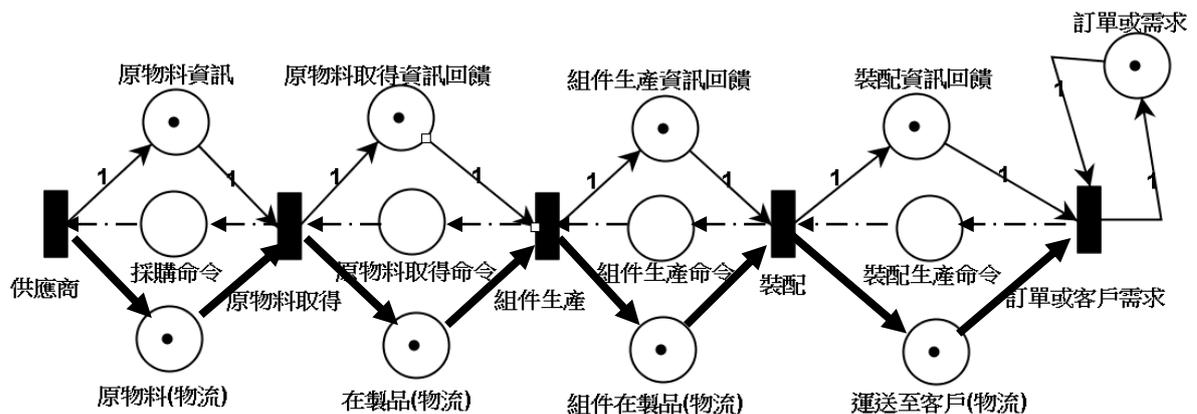


圖 3.14 例 3.3 生產流程的物流及資訊流

依拉式系統生產流程，例 3.3 的資訊流轉出至裴氏圖同於圖 2.18。但馬斯科洛的資訊流是將物流及資訊流作合併的動作，無法辨別物流及資訊流，在圖 3.14 將物流及資訊流各別繪出。將物流及資訊流分離可得資訊流記號圖，圖 3.15。物流圖如同推式系統物流圖，圖 3.6。為表示和推式系統資訊流差異，生產命令使用長虛點線表示。

圖 3.15 為在訂單(或預測)產生時，會激發轉移點裝配作業索取產品。裝配作業內的產品被取走後，看板容器會回到組件生產，而組件生產會依看板上的內容生產產品及數量。因此看板亦為暫存點組件生產命令。當組件生產完成後，看板容器裝滿後，看板及看板容器會送至裝配作業，即為暫存點組件生產資訊回饋。同理，當生產組件生產所需之物料時，運回至原物料取得。狀態為暫存點原物料取得命令，激發轉移點原物料取得。原物料取得完畢後，組件生產看板容器會及原物料取得完成品一起送至組件生產，狀態為暫存點原物料取得資訊回饋。

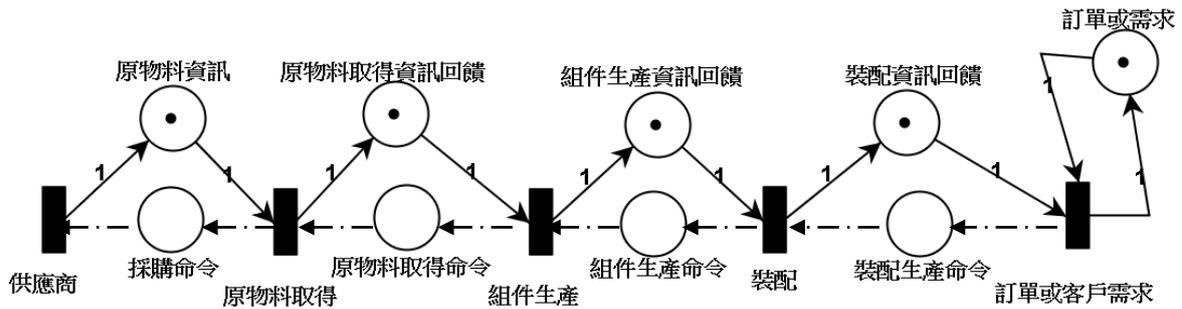


圖 3.15 例 3.3 資訊流

轉移點裝配激發條件有二，暫存點訂單(或預測)及暫存點組件生產資訊回饋，利用圖 2.6(e)的簡化法則，可得圖 3.16，為純拉式系統記號圖。

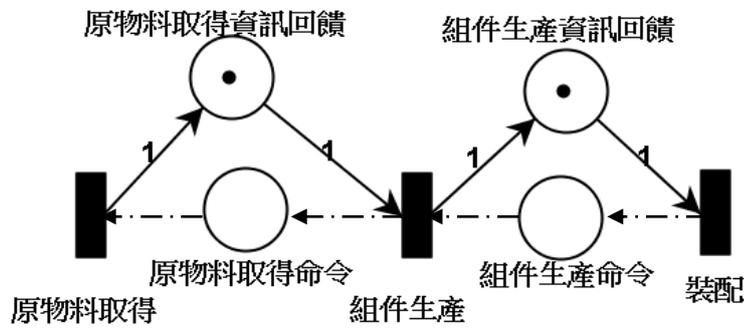


圖 3.16 純拉式系統(三站點)

在經由上述整理後，可推導出純拉式系統記號圖；生產作業站點為轉移點，各轉移點之間的資訊流一來一往。當有多個生產作業站點時，可計算出流程有 N 個作業時，就有 N 個轉移點， $2(N-1)$ 個暫存點。所有作業資訊(暫存點)只會與前一作業(轉移點)或後一作業(轉移點)有關，且資訊流為順時鐘方向，如圖 3.17 所示。

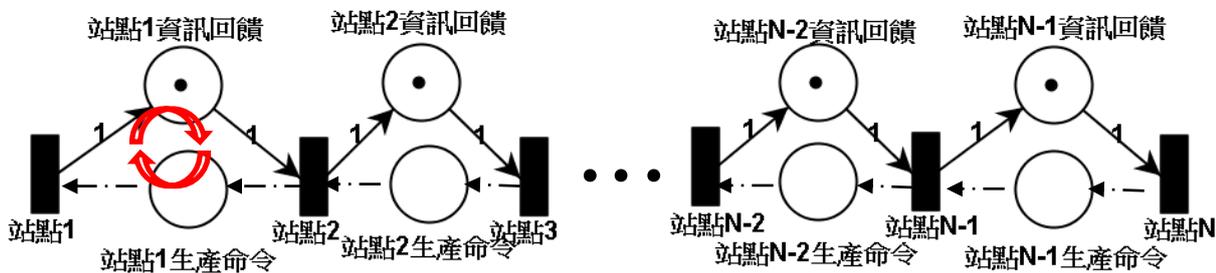


圖 3.17 純拉式系統

3.3.1 推式系統與拉式系統定義

波尼等人[13]定義當物流與資訊流同向時為推式系統，當物流與資訊流方向相反

時為拉式系統，但未明確定義物流與資訊流。在物流應指狹義的物流，在生產系統中的原物料及在製品流動；資訊流應為生產所需之資訊。實際在生產中，有非常多資訊的流動，不可能只有單向的流動，一定是雙向的資料交流。若如波尼等人的定義來說，資訊流只有一個方向，為同物流方向向客戶流動。沒有回饋資訊的生產流程，會無法修正生產流程中所產生的變異。以資訊傳遞來說，開始生產前是需要接收到生產命令來啟動生產，而生產命令的來源來有很多，包含預測、物料的到達、生產規劃、客戶需求及下游作業需求等。在生產完成後亦需回饋資訊，如完成品種類、數量及在製品等產品資訊。或機台狀態、線上異常等可能影響生產之因素，以滿足下次生產規劃所需之資訊。不是所有的生產命令所需的資訊都是相同的，會因生產系統不同而有作改變。推式生產系統的生產命令以生產規劃為主，在生產規劃中需求較多的資訊回饋。如機台狀態、在製品種類及數量、人員資訊等，才能作生產排程的動作。在拉式生產系統資訊流內容相較少，只需滿足下游作業看板上所需產品(或在製品)種類及數量，機台狀態、在製品數量及人員相關資訊皆不需知會下游作業，機台及在製品及人員相關事務皆需各作業站點自行管理。將上述推式生產系統及拉式生產系統暫存點資訊差異整理至表 3.1。

表 3.1 推式及拉式系統生產資訊流比較

生產系統比較	生產命令資訊內容	生產資訊回饋資訊內容
推式生產系統	預測、客戶需求、生產規劃、上游作業需求(物料的到達)	機台狀態、在製品數量、可能影響生產之因素之資訊
拉式生產系統	看板(下游作業需求)	看板(已完成種類、數量)

為建立數學模型，推式及拉式生產系統應具備明確的定義。即使生產流程為混合式(包含推式生產系統及拉式生產系統)，亦需明確指出各生產流程作業為何種生產系統，不能有同時為推式生產系統或是拉式生產系統。本論文會以各生產流程作業與上、下游之間資訊流關係的角度，來說明推式與拉式系統之間的差異。在推式及拉式系統間的資訊量，可由表 3.1 得知，拉式生產系統的資訊量相對推式生產系統的資訊量來的少及單純，在定義時會先由拉式系統開始說明。

拉式系統定義：生產命令來自下游作業。在生產作業與生產作業之間只存在三種關係，上游(Upstream Station)作業、平行作業、下游(Downstream Station)作業關係，任一作業被激發生產動作的資訊來源為下游作業時，此上、下游作業之間關係即為拉式系統。其資訊流方向為順時鐘方向。

推式系統定義：生產命令來自於非下游作業。非下游作業包含了其他所有可能性，意指生產命令來自上游作業資訊，或是生產規劃部門，而平行作業沒有上下游關係，因此不會互相干擾，不會有生產命令來自平行作業，在此情況下此二作業關係為推式系統。其資訊流方向為逆時鐘方向。

3.4 迴路數導向指標設計

本節將說明利用推拉介面作指標的設計。3.3 小節定義出推式及拉式系統，其辨別準則在於作業生產命令的下達，利用記號圖的數學性質來分析其系統之間的差異。本小節以生產命令發佈來源的角度作推拉式系統的指標。設立推拉式生產系統指標是依推式及拉式系統在生產流程中的比率而設置，因此有下列假定。推拉式生產系統指標介於 0 至 1 之間的數值，當指標為 0 時表示全生產流程為推式系統。當指標為 1 時表示為全生產流程為拉式系統。若介於 0 及 1 之間為推拉式混合式系統，拉式系統比重依指標值而遞增，如圖 3.18 所示：

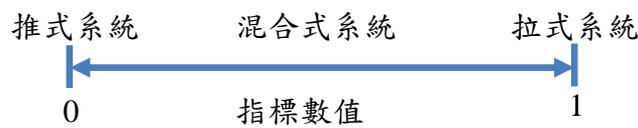


圖 3.18 推拉式推拉式指標

在設計中馬斯科洛[40]及波尼[13]只考量到單一生產流程，未考量組裝作業生產。本節在設計推拉式推拉式指標時，亦會考量到組裝作業。並以例 3.4 及例 3.5 為例計算推拉式指標。

3.4.1 推拉介面

哈里森等人(Harrison T. P. et al.)[19]提出在介於二種生產系統之間，可稱之為推拉介面(Push-Pull Boundary)[19]，此觀念亦可使用在記號圖上。在記號圖中在推式系統與拉式系統是使用暫存點資訊流作業判斷用，介於暫存點資訊流之間的為轉移點，因此推拉介面只有可能是記號圖中的轉移點。

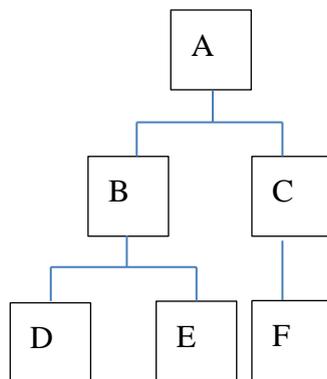


圖 3.19 例 3.4 的物料清單

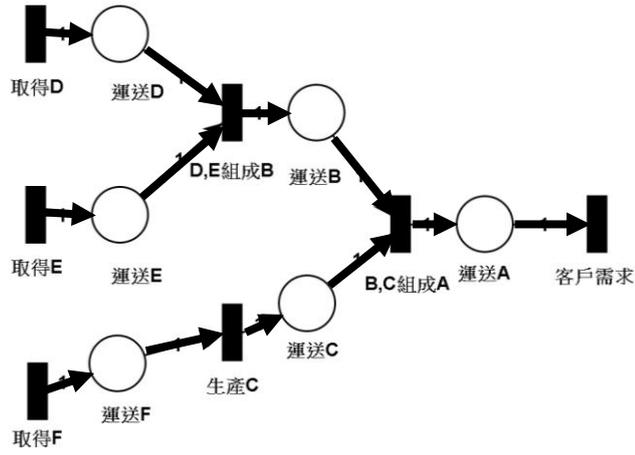
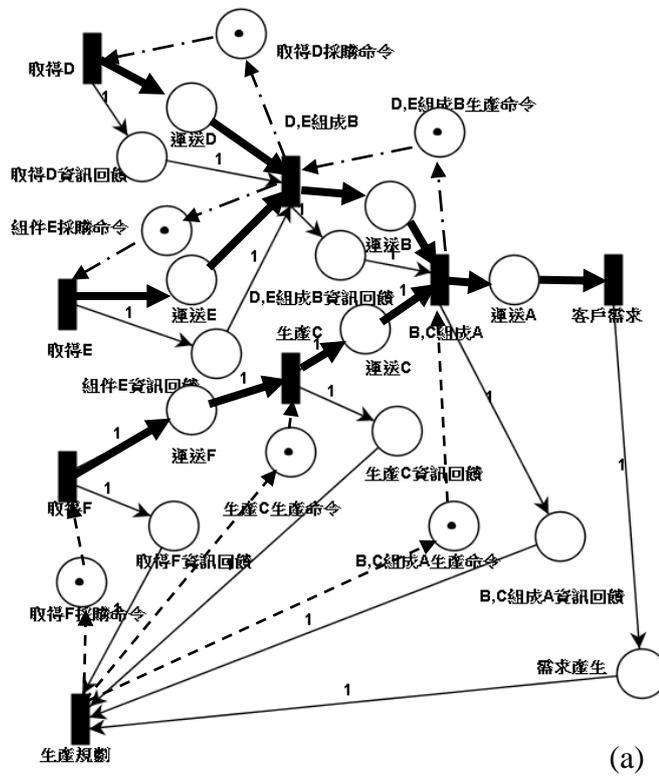
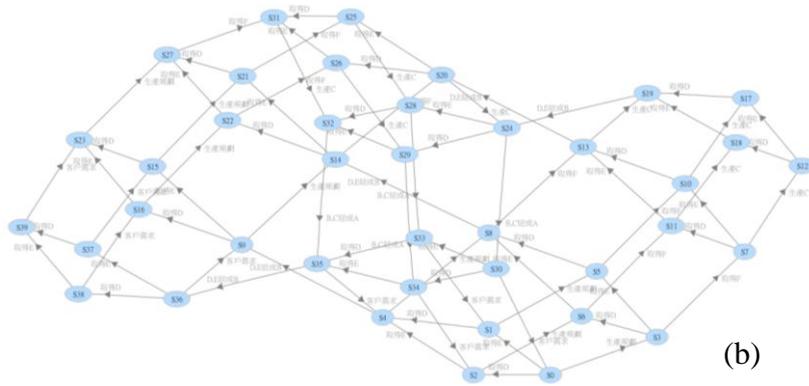


圖 3.20 例 3.4 的物流



(a)



(b)

圖 3.21 例 3.4 的物流、資訊流及可達圖

例 3.4：產品 A 組裝作業

有一產品 A 由組件 B 及組件 C 所組成。組件 B 由組件 D 及組件 E 所組成。組件 C 由組件 F 所組成。組件 F、組件 D 及組件 E 從供應商購買。其物料清單(Bill Of Material, BOM)如圖 3.18 所示。物流狀態裴氏圖如圖 3.19 所示，單一方向由左至右運送物料。產品 A 為有需求時生產。組件 B、組件 D 及組件 E 為有需求時生產。組件 C 及組件 F 生產排程由生產規劃制定。資訊流及物流如圖 3.21(a)所示，可達圖共有四十個狀態，如圖 3.21(b)所示。將資訊流及物流分離後可得資訊流記號圖，如圖 3.22 所示。

例 3.5：五生產作業

假定生產流程，設此一生產流程共 5 個作業(作業 1 至作業 5)依序生產。作業 1 及作業 2 為同一站點，皆由站點 1 下達生產命令。作業 3 及作業 4 為同一站點，皆由站點 2 下達生產命令。當作業 5 有需求時將發出生產命令至站點 2，當站點 2 有需求時將發出生產命令至站點 1，當有客戶需求時向作業 5 取貨，如圖 3.22(a)所示。其可達狀態共有 7 種，可達圖如圖 3.22(b)所示。

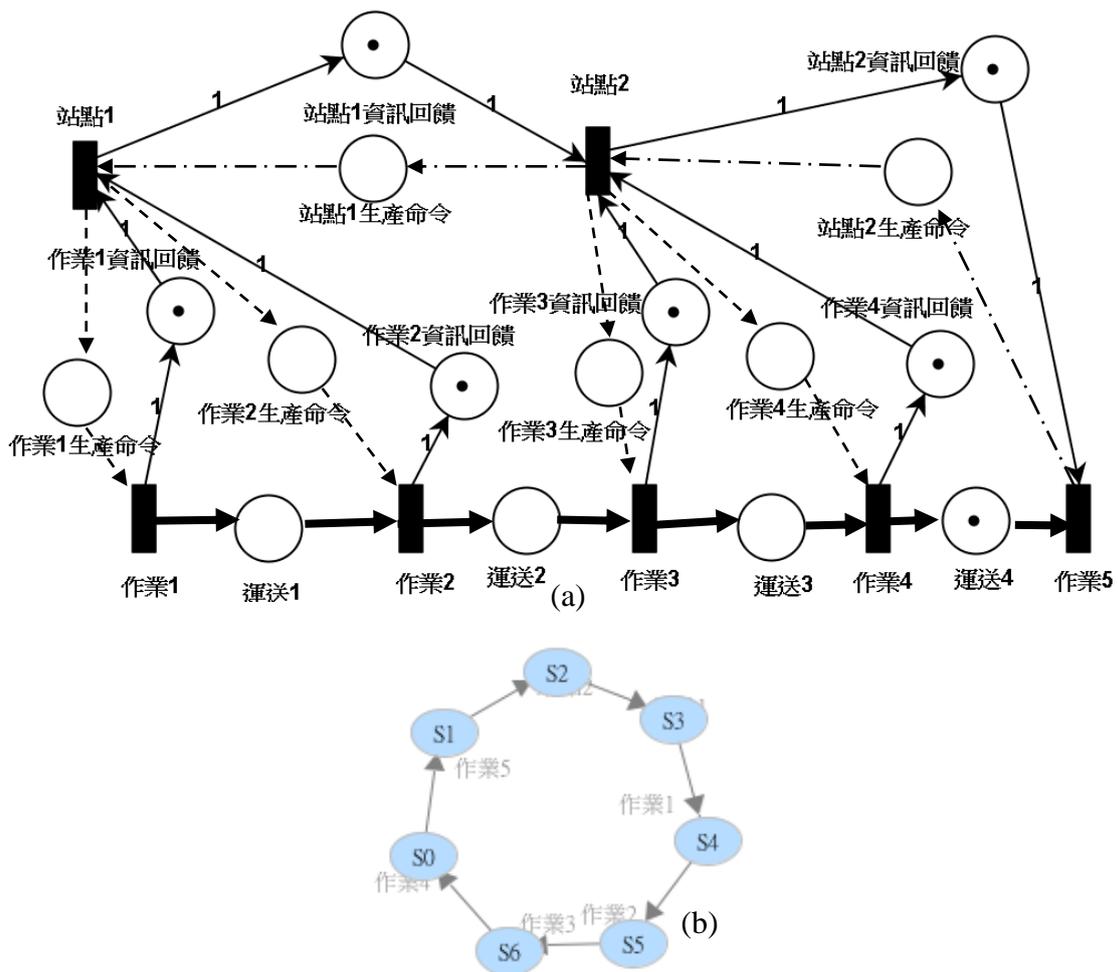


圖 3.22 例 3.5 的物流及資訊流

圖 3.21(a)依資訊流的差異可以找出轉移點「B,C 組成 A」為此生產流程的推拉介面，將資訊流迴路可分成二個部分。以推拉介面為分界線，以上為拉式系統，以下為推式系統，如圖 3.23 所示。

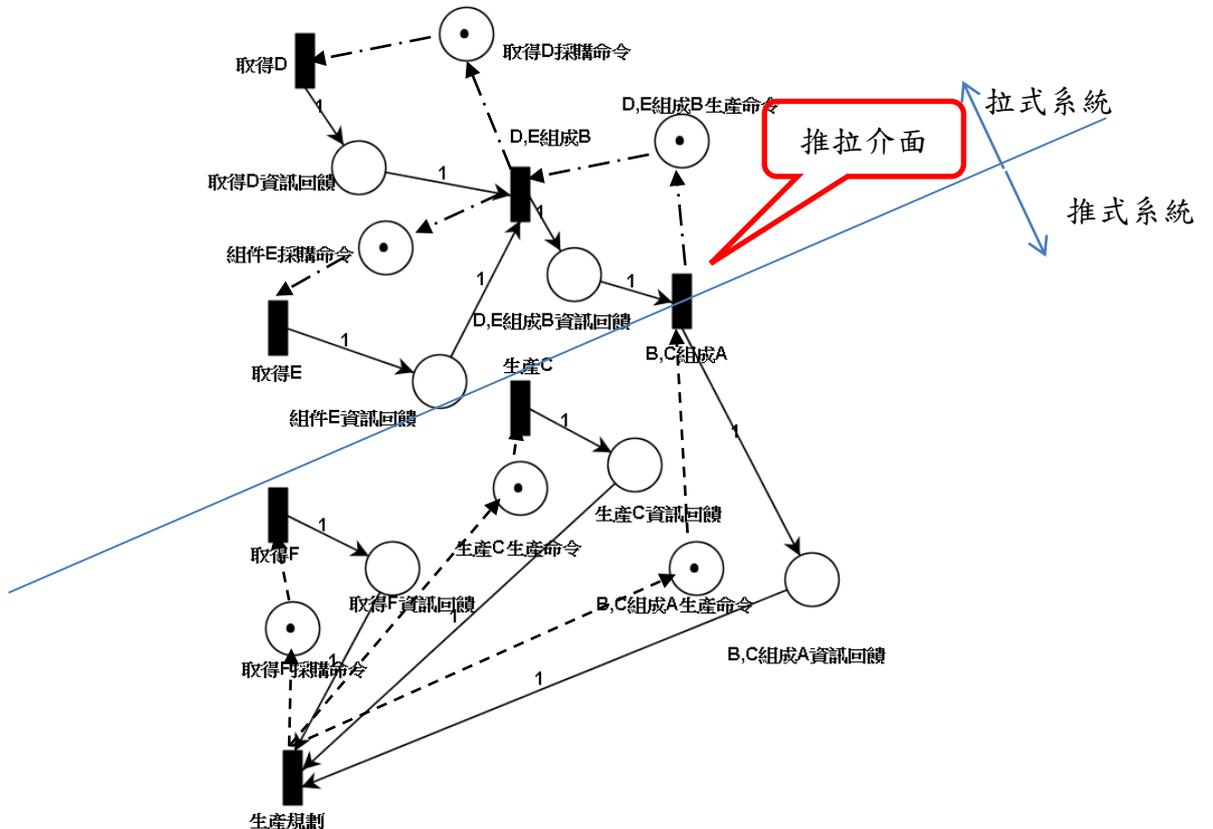


圖 3.23 例 3.4 資訊流及推拉介面

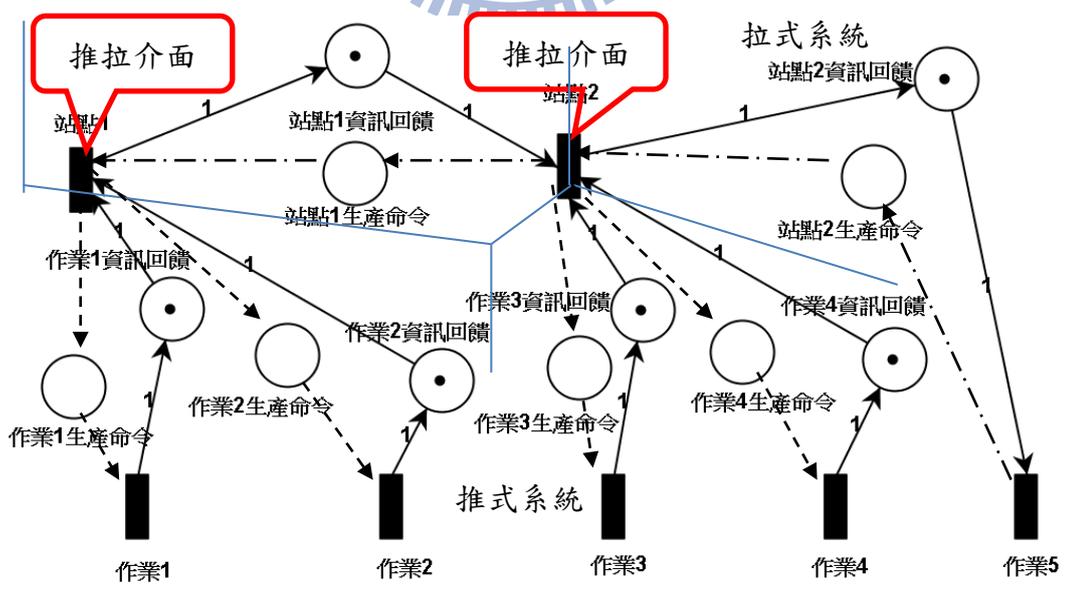


圖 3.24 例 3.5 資訊流及拉式介面

圖 3.22 的資訊流中，亦可找出其推拉介面，上半部分有二個拉式系統，下半部分

亦有二個推式系統，如圖 3.24 所示。

3.4.2 迴路數導向指標

整個生產流程中包含推式及拉式系統，利用第 3.1 節中推拉介面可區分各別推式及拉式系統。在明確的判別系統後，計算出拉式及全系統各別迴路數量。若是整個生產系統皆為推式或是拉式，則無推拉介面。若有包含推式及拉式系統，在分母設定為所有迴路數量，分子設定為拉式迴路數量，即可得拉式系統迴路在全生產系統中的比值。令 I_c 為迴路數導向指標。令 C_{Pull} 為拉式迴路數量， C_{All} 為所有迴路數量。拉式迴路數量及所有迴路數量值域為 $0 \leq C_{Pull} \leq C_{All}$ 。二者相除後，如公式(1)所示。

$$0 < I_c = \frac{C_{Pull}}{C_{All}} < 1 \quad (1)$$

以圖 3.22 為例，由推拉介面分成二個部分。共有二個迴路數，一個為推式，一個為拉式。 $C_{Pull}=1$ ， $C_{All}=2$ 依公式(1)可得 $\frac{C_{Pull}}{C_{All}} = \frac{1}{2} = 0.5$ 。以圖 3.23 為例，將其切割成

四個部分。上半部分拉式，下半部分推式。可得值 $C_{Pull}=2$ ， $C_{All}=6$ ， $I_c = \frac{C_{Pull}}{C_{All}} = \frac{2}{6} = 0.33$ 。



3.5 暫存點導向指標設計

在 3.1 節中說明生產命令及生產資訊回饋皆為一狀態，使用暫存點表示。又定義推式及拉式系統的差異在於生產命令來源的轉移點作業，生產命令只要是來自於下游作業(站點)，即為拉式系統。反之，非來自下游作業為推式系統。因此本小節由暫存點的數量為導向，建制推拉式生產系統指標。

3.5.1 暫存點導向指標

首先計算出在生產系統中所有生產命令暫存點數量，設為分母。再找出生產命令暫存點是由下游作業發出數量，設為分子。即可得知在全生產系統中，有多少比例的作業是拉式系統。推拉式生產系統指標可由下列方式定義。令 I_p 為暫存點導向指標， P_{Down} 為由下游作所下達生產命令的暫存點數量， P_{All} 為所有生產命令的暫存點數量。其值域為 $0 \leq P_{Down} \leq P_{All}$ 。再將二者相除，可得其推拉式系統指標，如公式(2)。

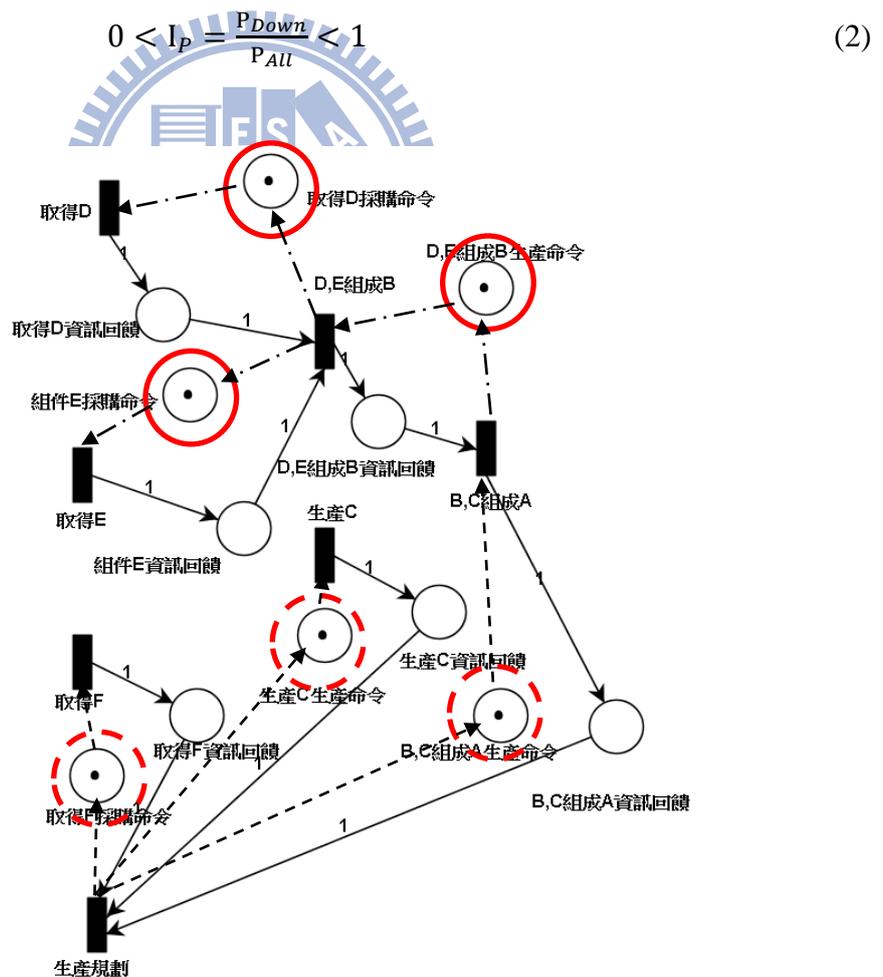


圖 3.25 例 3.4 資訊流及暫存點來源

以圖 3.23 為例，將生產命令暫存點來自下游的使用實線圓表示。生產命令暫存點非來自下游的，使用虛線圓表示，如圖 3.23 所示。代入公式(2)可得 $I_P = \frac{P_{Down}}{P_{All}} = \frac{3}{6} = 0.5$ 。

再以圖 3.26 為例， $I_P = \frac{P_{Down}}{P_{All}} = \frac{2}{6} = 0.33$ 。

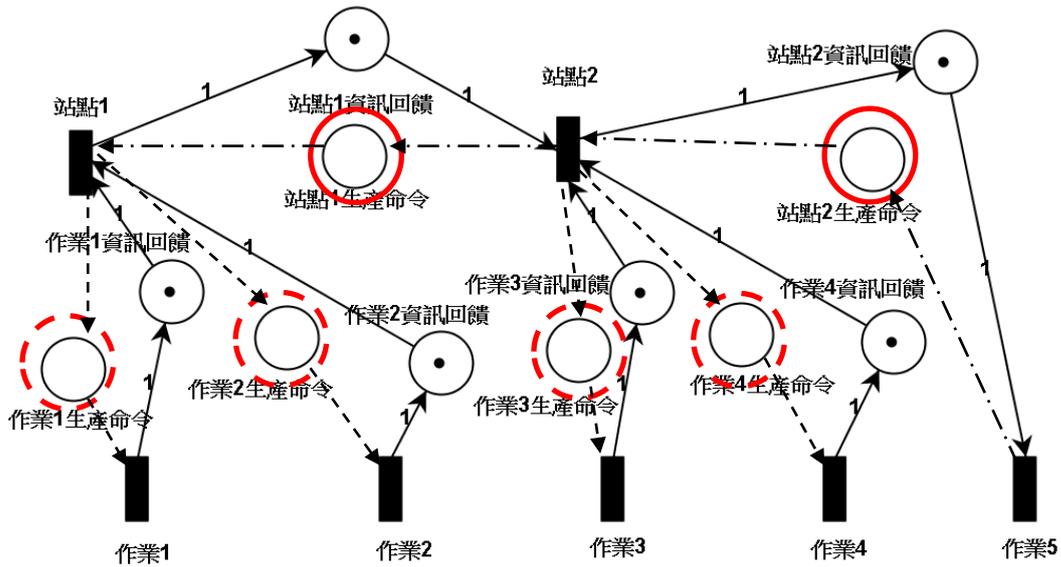


圖 3.26 例 3.5 資訊流及暫存點來源



3.6 複雜生產系統案例的推拉式指標分析

為計算及比較其二者推拉式系統推拉式指標，假定一個共有四個作業的生產流程。分別為作業 1、作業 2、作業 3 及作業 4 所組成，依序生產。並列出所有作業可能的生產命令來源(即指推式或拉式生產系統)，如圖 3.27 所示。

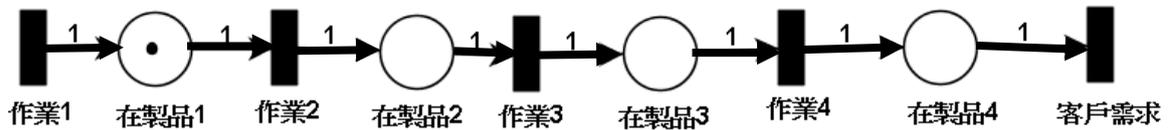


圖 3.27 四作業生產物流

3.3 節定義出的推式及拉式系統定義中，說明判斷推式或拉式在於資訊流的來源。在各作業中，依資訊流分類後只會有二種可能，推式系統與拉式系統。理論上共有 2^4 (推式或拉式)的 4 次方(作業數)種，即 16 種可能的組合。作業一沒有上游作業，無法向上拉貨。因此可能的組合減少至 2^3 為 8。又生產規劃至少處理二個以上作業，推式作業不會單獨存在，因此再扣 2 個可能性。作業 3 或作業 4 不單獨為推式系統。又有一特例為圖 3.26，同時有二個生產規劃。因此還有 7 種可能的生產流程。

在假定生產系統內共有四作業，分別為推式或是拉式的可能性組合約有七種繪出至記號圖。生產流程記號圖以例 3.6 到例 3.12 內共有七個例子。可先列出其生產命令來源如表 3.2 所示。因圖中會使用多個生產規劃，表中生產規劃(1)指若部分圖中使用多個生產規劃時表示為生產規劃 1。若只有一個時，生產規劃(1)表示為生產規劃。使用此表向上追溯生產命令來源，皆會指向訂單或預測，只是在生產流程中會因生產系統不同而改變。

例 3.6：四作業生產-1

生產流程為作業 1 至作業 4 依序生產，每一個作業的生產命令來源皆來自生產規劃單位，如圖 3.28 所示。

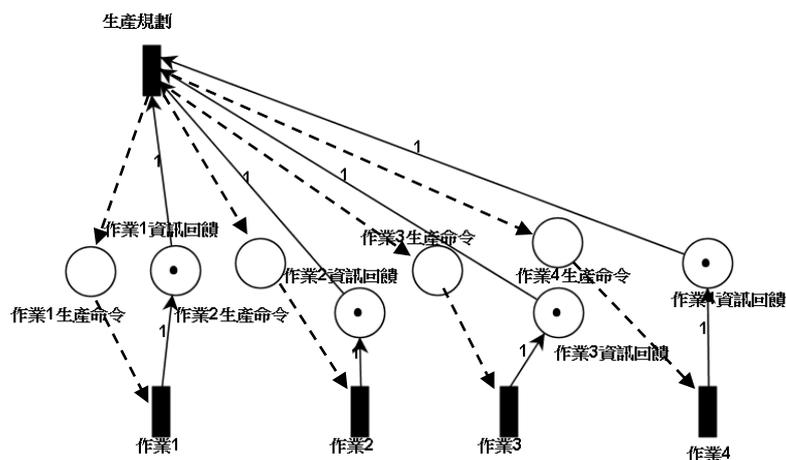


圖 3.28 例 3.6 的資訊流記號圖

例 3.7：四作業生產-2

四作業分成二個部分。作業 1 及作業 2 的生產命令來自生產規劃 1，作業 3 及作業 4 的生產命令來自生規畫 2。生產規劃 1 之需求來自生產規劃 2，如圖 3.29。

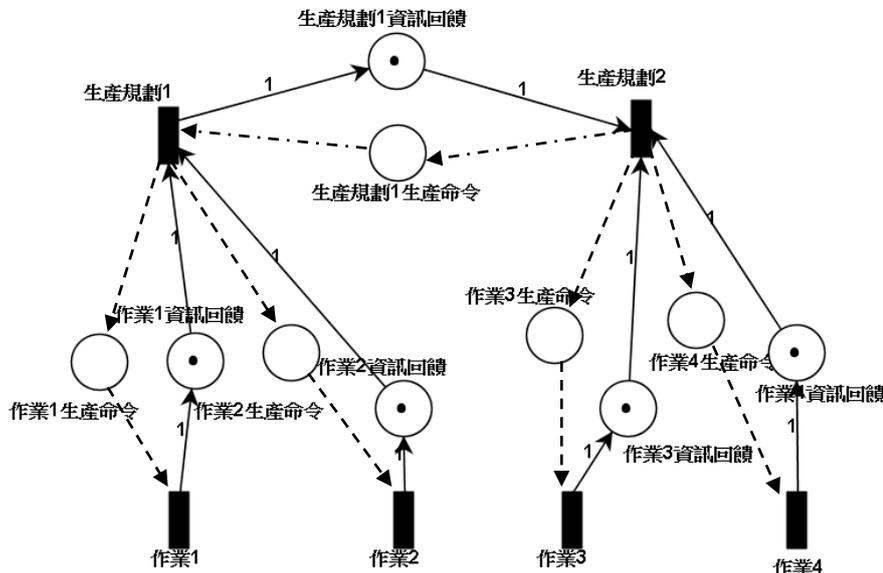


圖 3.29 例 3.7 的資訊流記號圖

例 3.8：四作業生產-3

作業 1 及作業 2 生產命令來自生產規劃。生產規劃生產命令來自作業 3，作業 3 生產命令來自於作業 4，如圖 3.30。

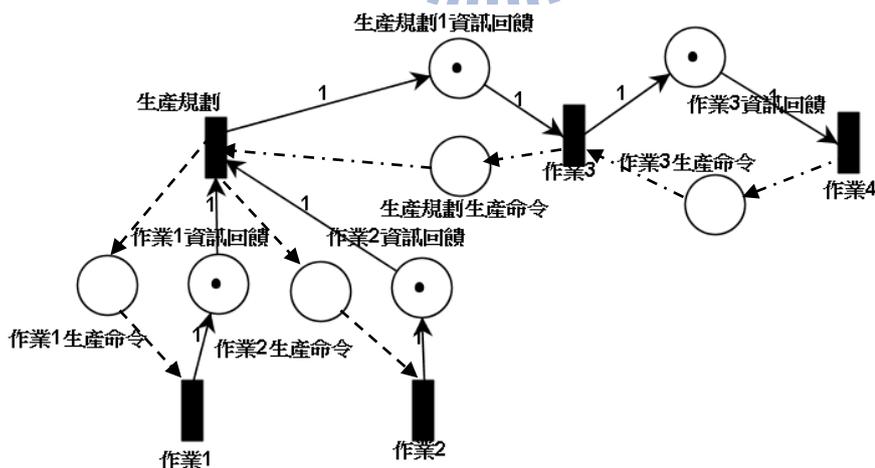


圖 3.30 例 3.8 的資訊流記號圖

例 3.9：四作業生產-4

作業 1、作業 2 及作業 3 的生產命令皆來自於生產規劃。生產規劃的生產命令來自於作業 4，如圖 3.31。

例 3.10：四作業生產-5

作業 1 生產命令來自作業 2。作業 2 生產命令來自作業 3。作業 3 及作業 4 生產命令皆來自生產規劃，如圖 3.32。

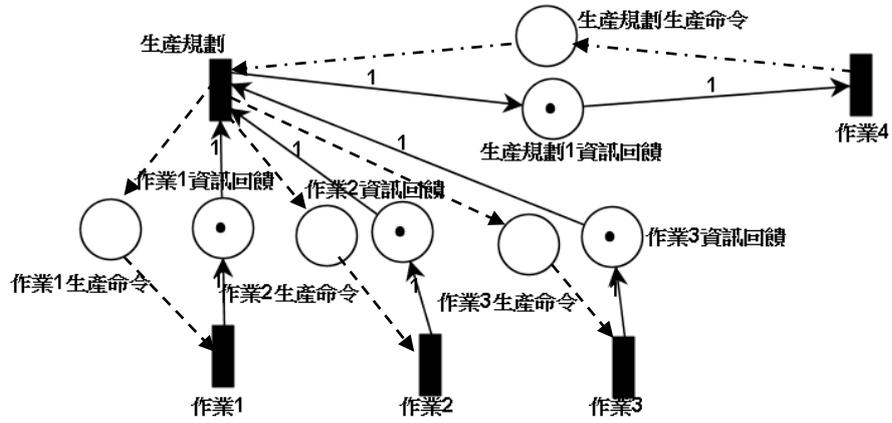


圖 3.31 例 3.9 的資訊流記號圖

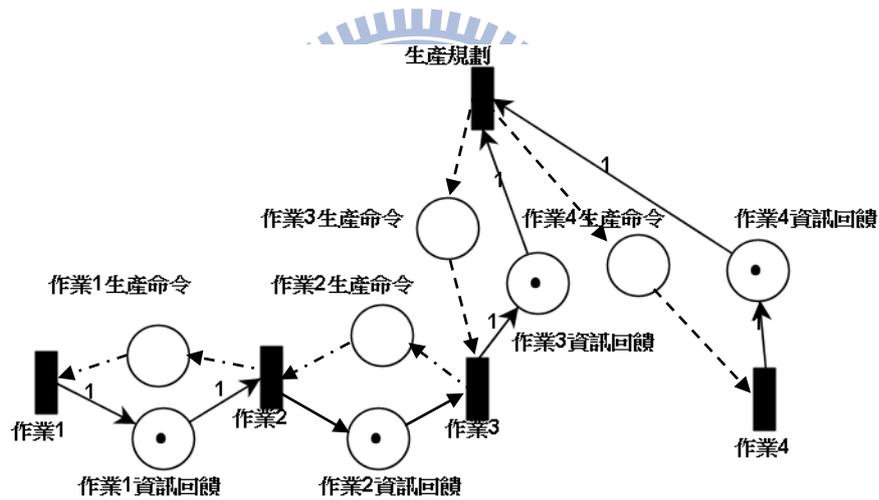


圖 3.32 例 3.10 的資訊流記號圖

例 3.11：四作業生產-6

作業 1 的生產命令來自作業 2。作業 2、作業 3 及作業 4 生產命令皆來自生產規劃，如圖 3.33。

例 3.12：四作業生產-7

作業 1 的生產命令來自作業 2，作業 2 的生產命令來自作業 3，作業 3 的生產命令來自作業 4。為純拉式系統，如圖 3.34 所示。

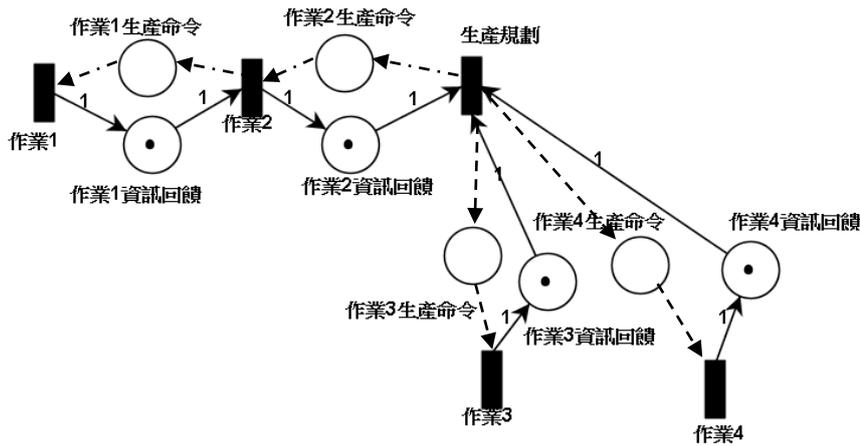


圖 3.33 例 3.11 的資訊流記號圖

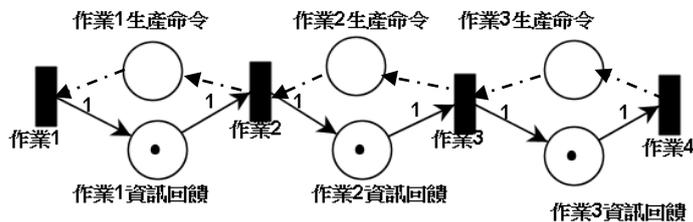


圖 3.34 例 3.12 的資訊流記號圖

利用表 3.2 即可直接看出在各二作業(或生產規劃)之間的是否有下達生產命令的關係。若有，則為拉式系統。若無，則為推式系統。表 3.2 資料可統計其生產命令來源是否來自下游作業的數量。在得知生產命令來自下游作業數量及生產命令總數量，使上述二數值相除即可求得暫存點導向指標之值。再使用推拉介面分割生產系統，計算拉式系統個數及全系統個數，相除後可得迴路數導向指標值如表 3.3 所示。

表 3.2 四作業生產命令來源列表

例\生產命令來源	作業 1	作業 2	作業 3	作業 4	生產規劃(1)	生產規劃 2
例 3.6	生產規劃	生產規劃	生產規劃	生產規劃	訂單或預測	-
例 3.7	生產規劃 1	生產規劃 1	生產規劃 2	生產規劃 2	生產規劃 2	訂單或預測
例 3.8	生產規劃	生產規劃	作業 4	訂單或預測	作業 3	-
例 3.9	生產規劃	生產規劃	生產規劃	訂單或預測	作業 4	-
例 3.10	作業 2	作業 3	生產規劃	生產規劃	訂單或預測	-
例 3.11	作業 2	生產規劃	生產規劃	生產規劃	訂單或預測	-
例 3.12	作業 2	作業 3	作業 4	訂單或預測	-	-

原例 3.6 中包含四個作業，應共有四個生產命令。例 3.12 在作業 4 的生產命令來源自於訂單或預測，又訂單或預測不在生產流程之中，因此去除。且其他作業 1、作業 2、作業 3 之生產命令來源各別來自作業 2、作業 3、作業 4，生產命令總數只會有

3 個。例 3.7、例 3.9，雖無作業四的生命令，但生產流程內含一個生產規劃，還是保有 4 個生產命令總數。例 3.7 中，雖作業 1、作業 2、作業 3 及作業 4 生產命令來源皆來自生產規劃部門，但卻是來自不同的生產規劃部門。各別為生產規劃 1 及生產規劃 2，又生產規劃 1 的生產命令來源為生產規劃 2。多一個生產命令，且判定此二個生產規劃部門有拉式系統的關係。

表 3.3 推拉式系統指標值

例\指標	生產命令來自下游	生產命令總數	拉式迴路數	全迴路數	I_P	I_C
例 3.6	0	4	0	1	0	0
例 3.7	1	5	1	3	0.2	0.33
例 3.8	2	4	1	2	0.5	0.5
例 3.9	1	4	1	2	0.25	0.5
例 3.10	2	4	1	2	0.5	0.5
例 3.11	1	4	1	2	0.25	0.5
例 3.12	3	3	1	1	1	1

表 3.3 中可明顯看出在混合的生產系統下，暫存點導向指標在 7 個圖形出現 5 種不同的值，皆介於 0 到 1 之間，如圖 3.35。迴路數導向指標只出現 4 種值，亦皆介於 0 到 1 之間。表示暫存點導向指標比迴路數導向指標更能表現其生產系統的差異性。在上表中所有 I_C 值 $\geq I_P$ 值，是例子本身因素，非指標值設計因素。在其他較大型的生產流程中，亦有可能出現 I_P 值 $\geq I_C$ 值。

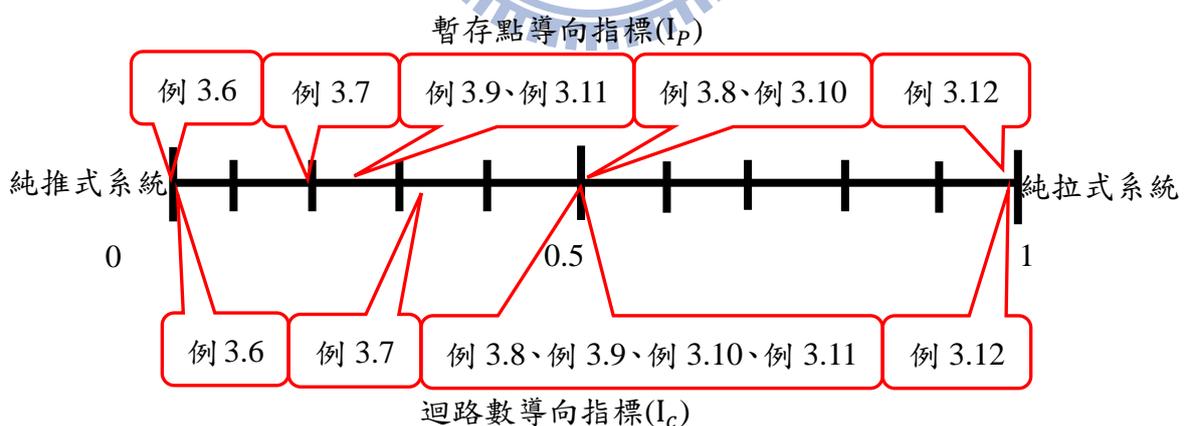


圖 3.35 例 3.6 至 3.12 的 I_P 及 I_C 分佈

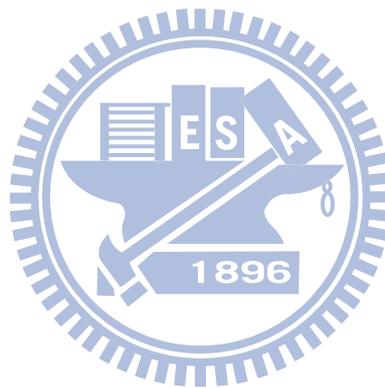
暫存點導向指標是由各站點的生產命令來源作為計算用，較迴路數導向指標來的敏感，所有生產站點不含權重都包含在暫存點導向指標的計算中。在實際生產中，並非所有站點的重要性都是相同的，會因製程的重要性或是其他生產因子的不同，干擾到暫存點導向指標的數值。在此情況下可採用迴路數導向指標作計算的依據。迴路數導向指標主要是依推式及拉式系統在生產系統內的數量而製定的。若在生產系統中有

某站點的生產型態改變時，較不會影響其數值的改變。優缺點比較如表 3.4 所示。

表 3.4 推拉式指標優缺點比較

指標\優缺點	優點	缺點
I_p	1.每一作業皆會計算。 2.數值精確。	1.數值敏感。 2.需辨別所有生產命令。
I_c	1.計算方便。 2.不受部分流程改變影響。	1.指標值差異不大。

在上述的優缺中，以暫存點導向指標較佳於迴路數導向指標。使用指標上，目的在於可以明顯的評比生產系統推拉式的差異，數值精確是十分重要的。在數值敏感部分上，可利用多作業組成一站點的方式解決。在找出所有生產命令暫存點的問題中，可使用電腦去作辨別。相對迴路數導向指標值差異不大的問題易解決多了。



第四章 整合性裴氏圖軟體開發環境的設計

本章說明使用物件導向程式語言 Java[28]撰寫的裴氏圖功能模組以及裴氏圖可加註語言導向資料伺單模組，建制完整的自動化程式軟體。在 4.1 節中說明裴氏圖設計流程建立，在第 4.2 節說明程式模組架構。

4.1 裴氏圖設計流程建立

有關裴氏圖應用在製造執行系統中已開發許多功能軟體，卻無整合性的軟體存在，且具有各自獨立作操作界面，使用者在使用其相關程式時，需不停切換各功能程式，所幸程式在開發時，皆是以功能模組為單位在作開發，方便其他使用者可再自行套用其試能模組，不必再自行撰寫程式開發，本節將已存在之模組依自動化系統開發為一套製造執行系統。

在自動化系統專案的分析中，包含規格、驗證、撰碼、測試、偵查，此五步驟。其流程如圖 4.1 所示。在一開始可使用 IDEF0 技術或是裴氏圖作為自動化系統的模式，若使用 BPwin 的 IDEF0 技術格式(*.idl)則可自動轉為裴氏圖可加註語言格式，若是使用裴氏圖可加註語言格式時，可直接進入裴氏圖分析工具，並使用形化使用者介面 (Graphical User Interface, GUI)表示其裴氏圖形，再利用裴氏圖基本模型了解其裴氏圖特性，繪出可達圖，並找出是否有系統鎖死情況產生。

在有鎖死情況下發生時，可利用活可達圖模組排除其鎖死路徑，即可利用循環分析模組列出相同的生產循環路徑，並確認此可達圖是否為歐氏圖。若否則使用歐氏有向圖模組增加所需之路徑。若為歐氏圖，使用測試序列模組找出最短生產路徑。

若系統無鎖死下，可對其作轉移點不變量分析、暫存點不變量分析，再確認是否為強連性記號圖(Strong Connected Marked Graph)。若是則可載入時間參數分析最小週期時間，並找出其偵查法則，若不是可直接設定自動化控制軟體。

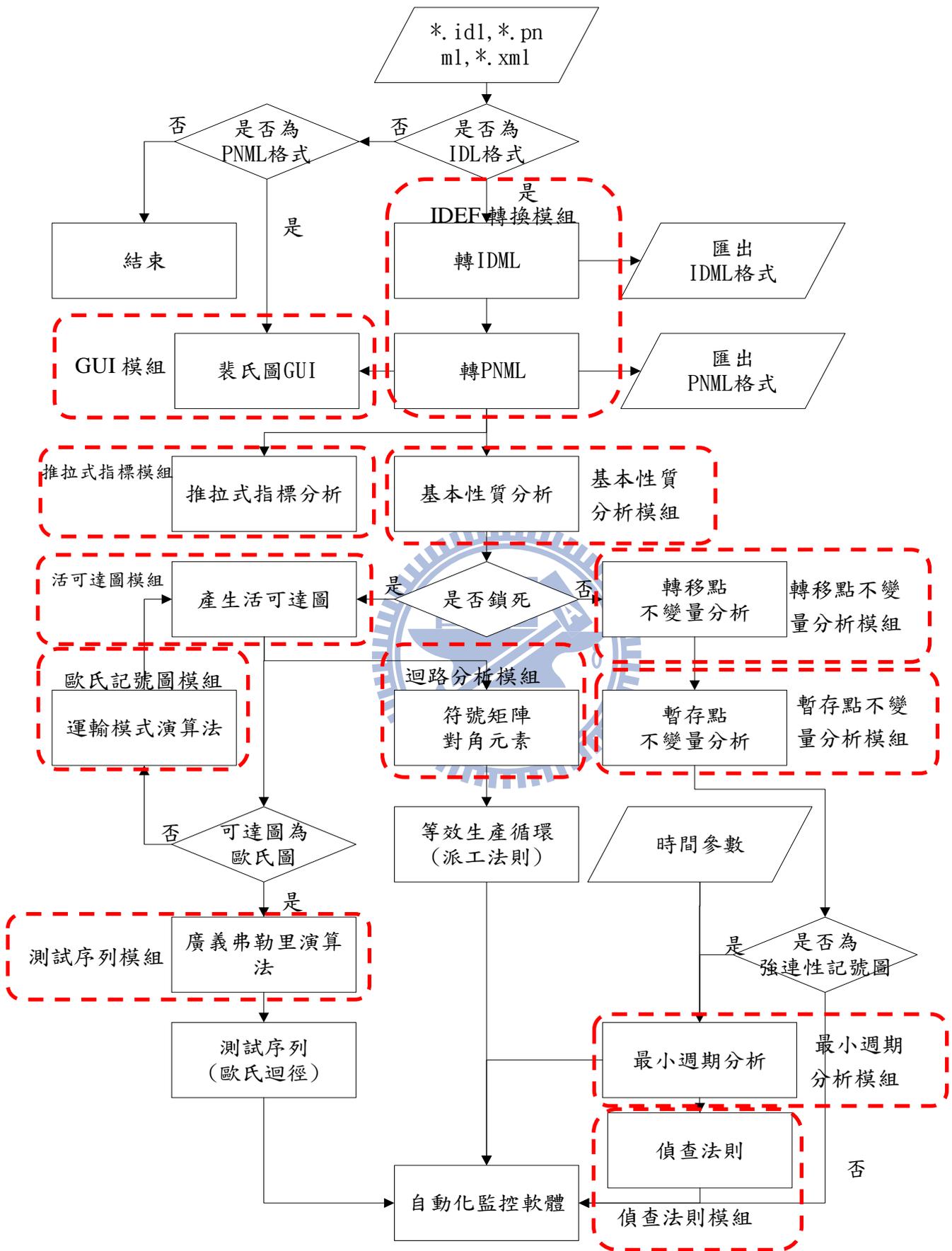


圖 4.1 自動化開發流程

4.2 程式模組架構

在本小節中將說明自動化系統的程式模組架構，依圖 4.1 的自動化流程系統中，雖有 11 個模組，但可主要分會五個部分，包含 GUI 模組、IDEF0 轉裴氏圖可加註語言模組、基本性質及不變量分析模組(內包含基本性質分析模組、轉移點不變量分析模組、暫存點不變量分析模組及最小週期分析模組)、規範測試模組(內包含活可達圖模組、歐氏有向圖模組及測試序列模組)、偵查法則模組及推拉式指標計算模組。共六個大部分，如圖 4.2 所示。

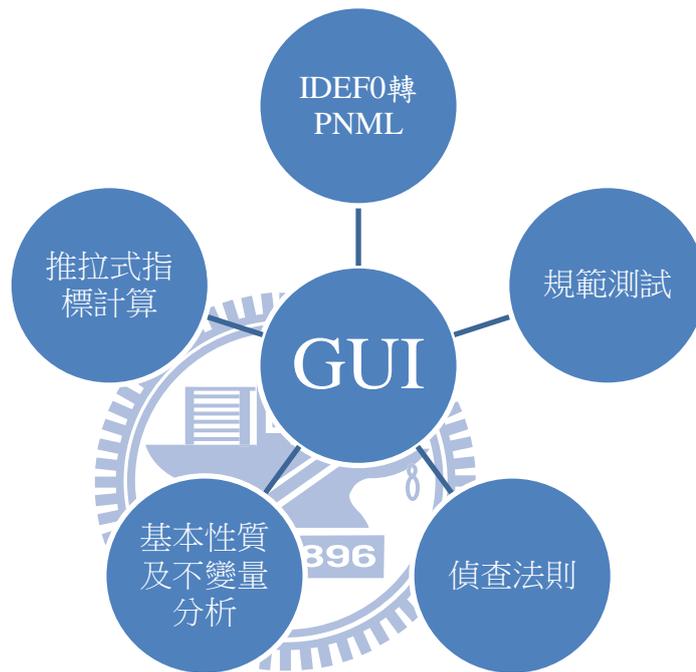


圖 4.2 模組關連資料

在此六大部分中，因演算法或是功能不同可再細分為 19 個套件(Package)或是功能較為簡潔而共用套件，如表 4.1 所示，並將其功能列出。其操作說明將於第五章再作介紹。

4.2.1 GUI 模組

在自動化系統中，需人性化的操作介面，本論文以功能為導向的方式將 GUI 模組作畫面的切割。畫面切割主要可分為三個部分，最上面的檔案處理區、操作介面左邊的功能列表區及操作介面右邊的裴氏圖展示區。功能列表區以列表的方式呈現，可具有擴充新增功能的能力。在裴氏圖展示區中預留了大部分的空間可供其他細部功能的彈性。

表 4.1 自動化模組與套件關係列表

模組名稱	套件
IDEF 轉移模組	idef0_transform_Module.build_component
	idef0_transform_Module.generate_idef0
	idef0_transform_Module.generate_petrinet
	idef0_transform_Module.transform_idf0_into_petrinet
GUI 模組	AMSdesigner
推拉式指標模組	Pduct.Ind
基本性質分析模組	analysis_Module.app.gui
活可達圖模組	conformance_test_Module
歐氏有向圖模組	
循環分析模組	
測試序列模組	
轉移點不變量分析模組	analysis_Module.app.gui
暫存點不變量分析模組	
最小週期分析模組	
偵查法則模組	rule_design_Module.custombutton
	rule_design_Module.custopane
	rule_design_Module.cyclebasis
	rule_design_Module.drawcurve
	rule_design_Module.dualinfo
	rule_design_Module.Module.troubleshooter
	rule_design_Module.monitor
	rule_design_Module.perinet.component
	rule_design_Module.perinet.xmltools
	rule_design_Module.srm

GUI 套件中可分成五個類別，主要負責使用者介面主畫面、主畫面切割、功能展示及檔案輸入處理，其類別及功能由表 4.2 所示，其類別關聯圖為如圖 4.4 所示。

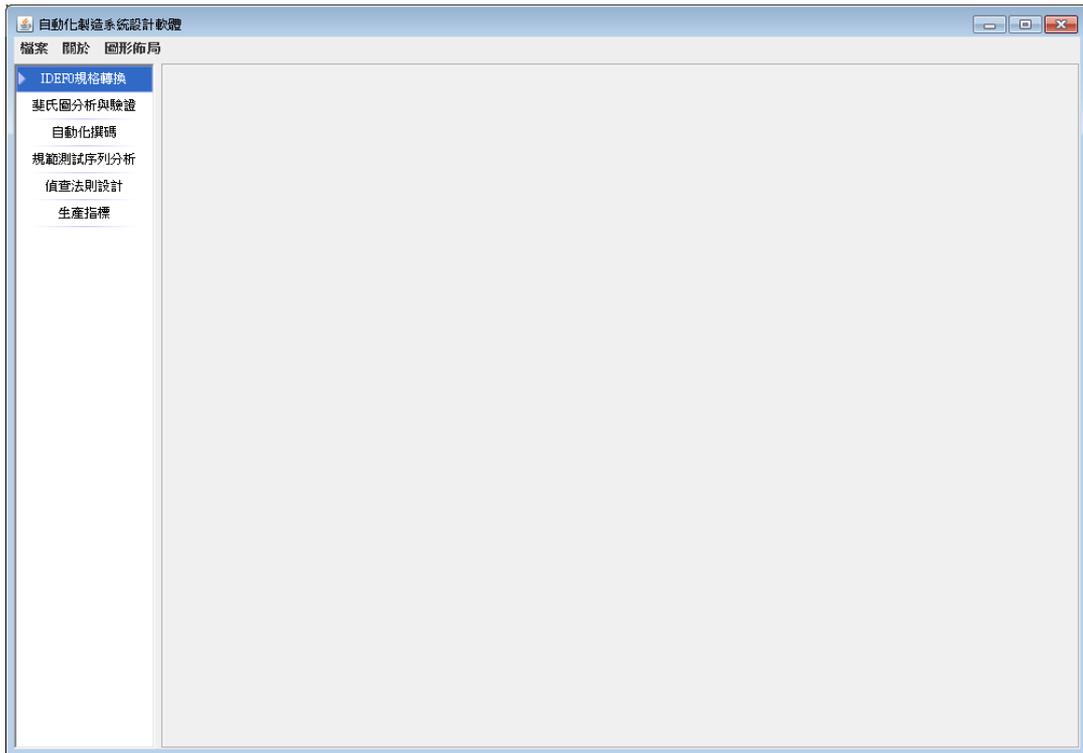


圖 4.3 自動化系統主畫面

表 4.2 使用者介面類別

類別	功能
Designer	檔案輸入處理
DesignerUI	檔案處理使用者介面
Initable	設定頁面格式
ListBar	程式功能列表
TxFileFilter	過濾檔名



4.2.2 IDEF0 格式轉裴氏圖格式模組

在此模組轉換流程如圖 2.19 所示，主要分三大步驟，共使用四個套件分別為 `idef0_transform_Module.build_component`、`idef0_transform_Module.generate_idef0`、`idef0_transform_Module.generate_petri_net` 及 `idef0_transform_Module.transform_idf0_into_petri_net` 套件其內包含十三個類別，其類別關聯圖中如圖 4.5 所示使用。

在 `idef0_transform_Module.build_component` 套件中主要先定義 IDEF0 格式及裴氏圖可加註語言格式之類別。`idef0_transform_Module.generate_idef0` 是對於 IDEF0 格式的處理。`idef0_transform_Module.generate_petri_net` 只有一個類別，目的在於將物件轉化為實際的裴氏圖可加註語言格式。`idef0_transform_Module.transform_idf0_into_petri_net` 套件處理 IDEF0 轉換至裴氏圖的主程式。其各套件類別說明如表 4.3 所示。

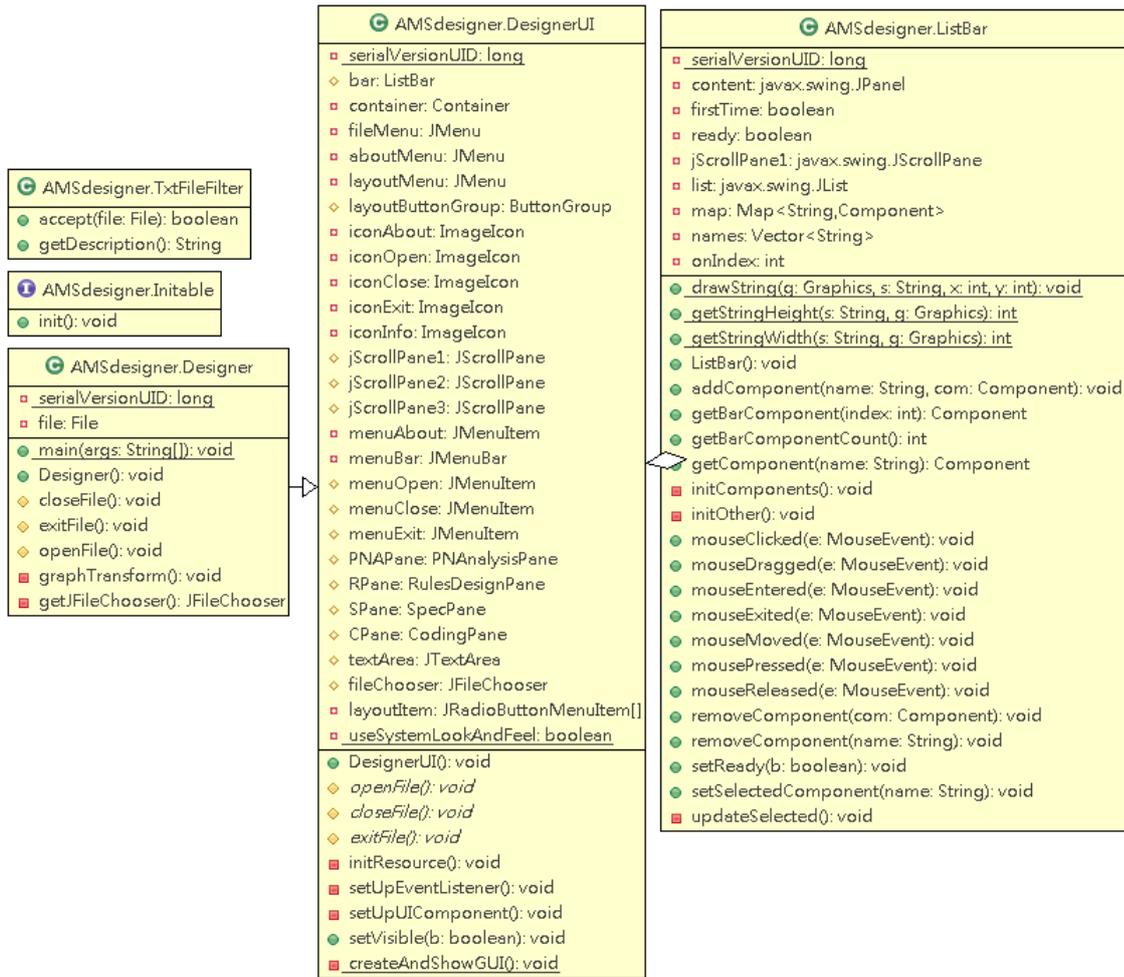


圖 4.4 圖形化使用者介面類別

4.2.3 基本性質與不變量分析模組及規範測試模組

在基本性質分析模組中將說明裴氏圖的性質共有十二個類別，主要是利用輸入裴氏圖可加註語言格式後，分析其數學性質。分析裴氏圖的主要類別為“BasicAnalysis”、“MinCycleTimeAnalysis”、“PlaceAnalysis”、“TransitionAnalysis”及“RG Analysis”。

在“BasicAnalysis”類別中主要判斷是否為純(Pure)裴氏圖、普通(Ordinary)裴氏圖、保守(Conservative)裴氏圖、狀態機(State Machine)裴氏圖、記號圖、強連性(Strongly Connected)裴氏圖、狀態界(Structurally Bounded)裴氏圖、安全(Safe)裴氏圖、活(Live)裴氏圖及重複性(Reversible)裴氏圖，共 11 個裴氏圖性質。

“MinCycleTimeAnalysis”、“PlaceAnalysis”、“TransitionAnalysis”及“RG Analysis”類別主要計算最小週期計算、暫存點不變量、轉移點不變量及不變量分析。其他類別為在裴氏圖展示中裴氏圖的表達及操作介面設定。其說明如表 4.4 所示。UML 圖形如圖 4.6 所示。

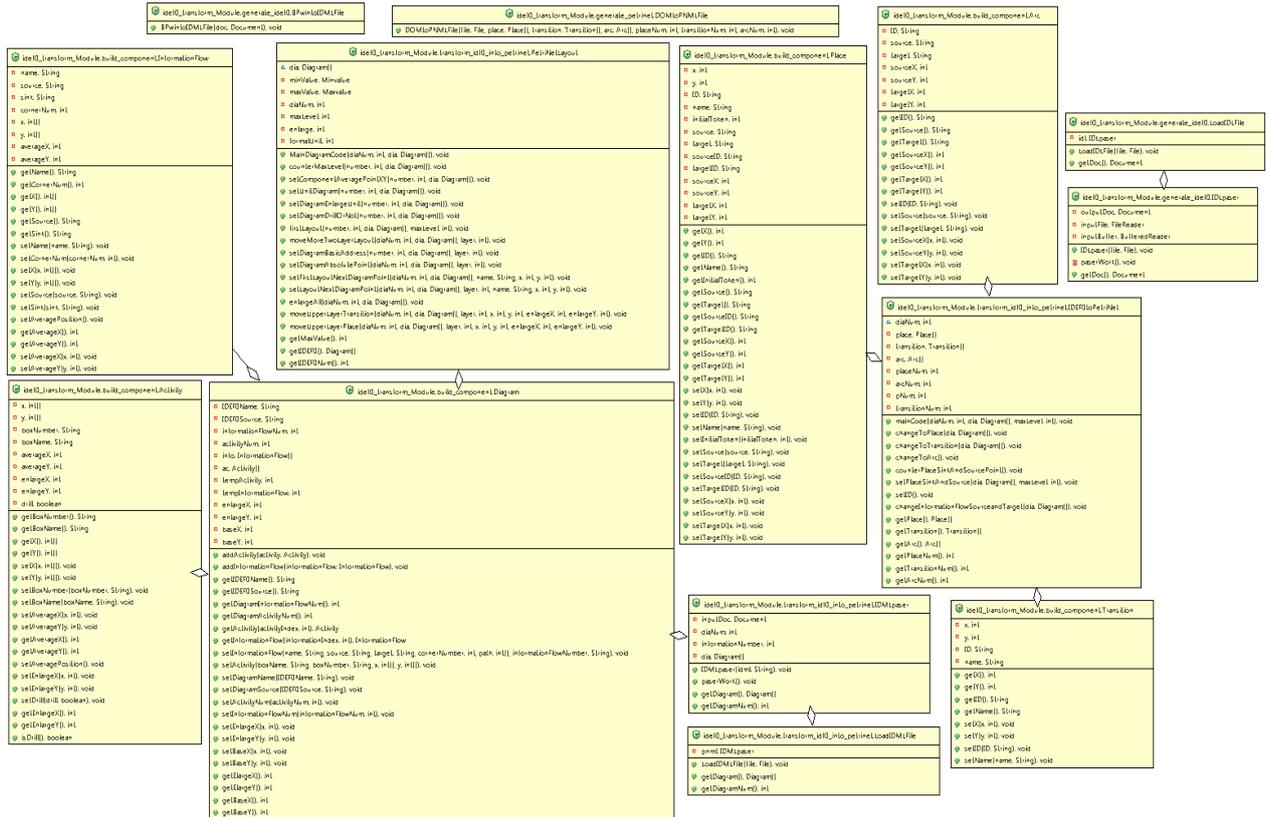


圖 4.5 IDEF0 格式轉換至裴氏圖可加註語言格式類別

表 4.3 IDEF0 轉換模組類別說明

套件名稱	類別名稱	說明
idef0_transform_Module.build_component	Activity	IDEF0 動作類別
	Arc	IDEF0 有向線類別
	Diagram	IDEF0 方格類別
	InformationFlow	IDEF0 資料流類別
	Place	裴氏圖暫存點類別
	Transition	裴氏圖轉移點類別
idef0_transform_Module.generate_idef0	BpintoIDEFFile	實體化 IDL 檔案
	IDLpaser	辨別 IDL 資料格式
	LoadIDLFile	處理 IDL 檔
idef0_transform_Module.generate_petriNet	DOMtoPetriNet	實體化裴氏圖可加註語言檔案
idef0_transform_Module.transform_idf0_into_petriNet	IDEF0toPetriNet	格式轉移主程式
	IDMLpaser	辨別 IDML 資料格式
	LoadIDMLFile	處理 IDML 檔
	PetriNetLayout	處理裴氏圖元件位置

表 4.4 基本性質模組類別說明

類別	說明
BasicAnalysis	裴氏圖基本分析
BasicSet	版面基本設定
CodingPane	裴氏圖區版面格式設定
DrawPane	裴氏圖展示區版面
GraphLayout	裴氏圖圖示排版
MinCycleTimeAnalysis	最小週期分析
PlaceAnalysis	暫存點不變量分析
PNAnalysisPane	裴氏圖區版面設定
RGAnalysis	可達圖分析
RunPLCSet	PLC 連接設定
SpecPane	載入時間參數檔案介面
TransitionAnalysis	轉移點不變量分析

規範測試模組所包含一連串的活可達圖模組、循環分析模組、歐氏有向圖模組及測試序列模的處理皆在同的一套件及類別中其程式流程只需輸入裴氏圖可加註語言格式後其程式流程如圖 4.1 即求得其規範測試路徑。其類別包含五個函式如表 4.5 所示。

表 4.5 規範測試模組類別說明

模組名稱	套件名稱	類別名稱	函式
活可達圖模組	conformance_test_Module	conformanceTest	costTest()
			deadlockTest()
歐氏有向圖模組			cpaTest()
循環分析模組			cycle()
測試序列模組			fleury()

4.2.4 偵查法則模組

在偵查法則模組使用擬陣理論方法，用裴氏圖形及數學性質來找出其偵查法則，因操作上包含了矩陣的計算，裴氏形的展示及對偶的觀念，其所需功能較多。rule_design_Module.custombutton、rule_design_Module.custopane 及 rule_design_Module.cyclebasis 三個套件主要是在處理裴氏圖區中操作介面、按鍵及展示功能。rule_design_Module.dualinfo、rule_design_Module.srm 二套件為利用擬陣理論對裴氏圖作對偶及 SRM 計算。rule_design_Module.perinet.component 及 rule_design_Module.perinet.xmltools 為處理裴氏圖可加註語言之套件。

rule_design_Module.Module.troubleshooter 及 rule_design_Module.perinet.component 為偵查法則之計算。其套件及說明如表 4.6 所示。其圖 4.7 為套件圖相關圖。

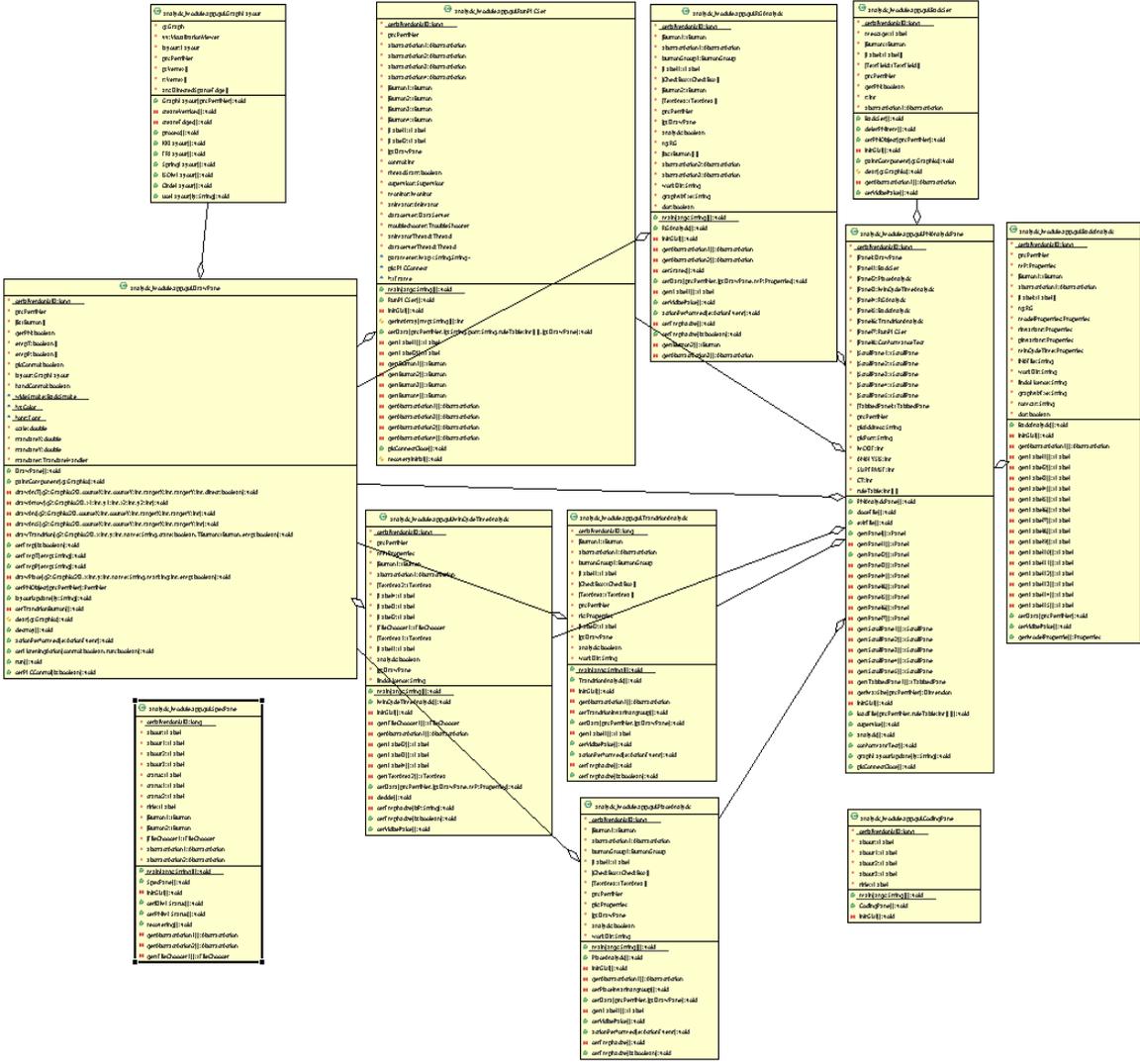


圖 4.6 基本性質及不變量分析類別

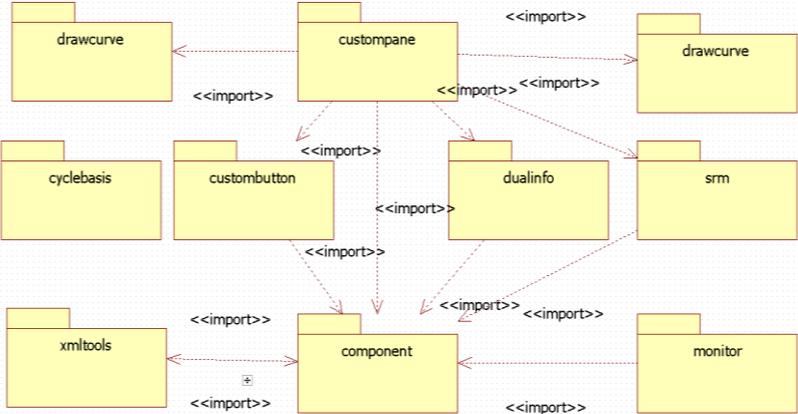


圖 4.7 偵查法則套件(Package Diagram)

表 4.6 偵查法則套件功能說明

套件	功能
rule_design_Module.custombutton	按鍵功能執行
rule_design_Module.custopane	裴氏圖區版面設定
rule_design_Module.cyclebasis	迴路處理
rule_design_Module.drawcurve	線段繪畫設定
rule_design_Module.dualinfo	對偶資訊處理
rule_design_Module.Module.troubleshooter	偵查法則監控程式
rule_design_Module.monitor	偵查法則處理
rule_design_Module.perinet.component	裴氏圖元件類別
rule_design_Module.perinet.xmltools	XML 格式處理功能
rule_design_Module.srm	計算 SRM

4.2.5 推拉式指標模組

依本論文對推式及拉式系統的定義，依生產命令來源作為判斷因素。推拉式指標模組主要包含三項，(1)PNML 資料擷取及分析。(2)版面設定。(3)推拉式指標計算。在 PNML 資料擷取及分析方面，使用黃柏勳已建製之裴氏圖元件類別 (rule_design_Module.perinet.component) 作為開發工具。在裴氏圖弧類別 (rule_design_Module.petrinet.component.Arc) 中，需再新增方法(Method)包含取得資料來源(Arc.getSource())及取得資料輸出(Arc.getTarget())。版面設定只需再新增，即可直接套用。推拉式指標套件命名為“Pduct.Ind”即可計算推拉式指標。

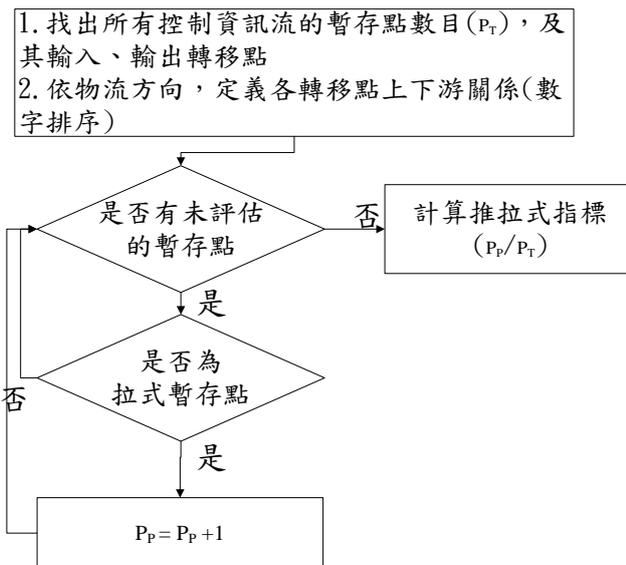


圖 4.8 推拉式指標計算處理流程圖

程式流程如圖 4.8 所示。首先找出所有控制資訊流，及其輸入輸出轉移點。再判斷是否未評估的暫存點，若無，直接計算推拉式指標，若有，判斷是否為拉式暫存點。直到評估完所有暫存點為止。再利用暫存點導向指標之定義，即可求得 I_p 值。



第五章 範例實際操作

本章以二個已存在之實例作為說明生產系統指標。在 5.1 節說明本程式之介面及使用，並以灌模系統中的傳輸系統作為範例。5.2 節將使用製造系統計算推拉式指標。5.3 節使用灌模系統計算推拉式指標。

5.1. 程式介面及功能

程式介面分成三個部分，第一部分檔案處理區，為(1)檔案載入及退出檔案。(2)圖形佈局。第二部分為裴氏圖應用區，包含(1)IDEF0 規格轉換模組。(2)斐氏圖分析與驗證模組。(3)自動化撰碼模組。(4)規範測試序列分析模組。(5)偵查法則設計模組。(6)推拉式指標。第三部分為斐氏圖區，作為圖型處理及各模組內操作功能。程式開啟畫面如圖 5.1 所示。

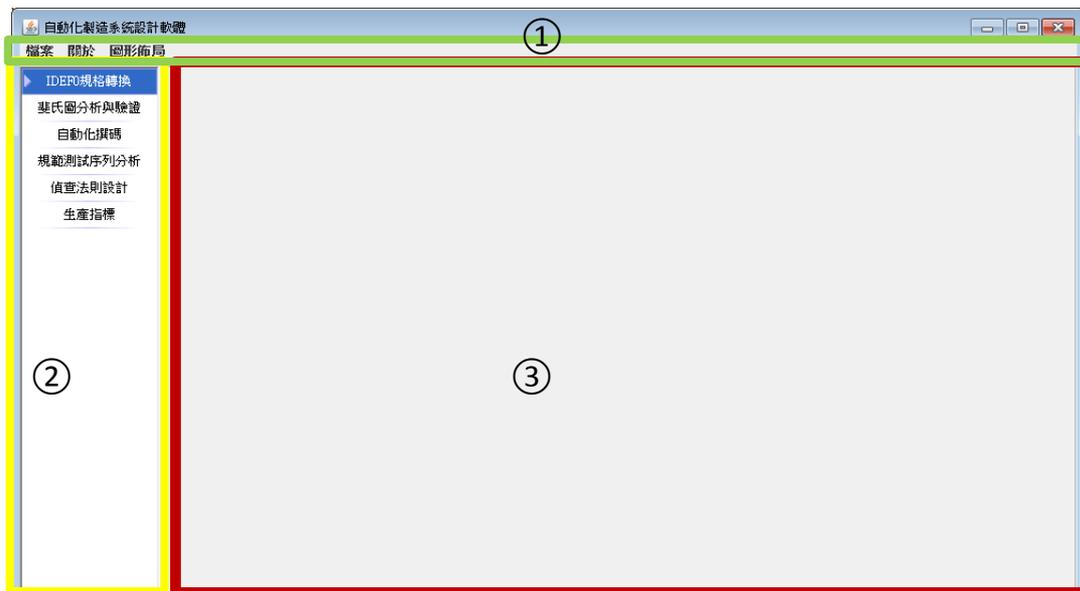


圖 5.1 程式開啟畫面

5.1.1 檔案載入及圖形佈局

下列將以灌模系統內的運輸系統作為例子說明下列四項模組。(1)IDEF0 規格轉換模組。(2)斐氏圖分析與驗證模組。(3)規範測試序列分析模組。(4)偵查法則設計模組。

例 5.1 灌模系統

灌模系統[38]包含二個部分，第一部分為液體混合攪拌，第二部分為控制輸送帶。在原物料的部分有包含原料 A、B 及 H。A 及 B 混合後產生 Y，直到槽 1(Tank_1)所定之容量上限。Y 再和原料 H 混合後產生 X，直到槽 2(Tank_2)所定之容量上限。當有

感測器 RFID 感測到容器模型運送來時，會依 RFID 上之資料定義灌模時間。其暫存點狀態如表 5.1，轉移點說明如表 5.2 所示。依梁高榮[38]的方法轉換至裴氏圖，再將物流及資訊流作分離後，可得圖 5.2。

表 5.1 例 5.1 暫存點列表

設備暫存點狀態		狀態-1	狀態-2	狀態-3	狀態-4
閘	閘_A	開	關	-	-
	閘_B	開	關	-	-
	閘_H	開	關	-	-
	閘_Y	開	關	-	-
	閘_2	開	關	-	-
	閘_3	開	關	-	-
攪拌器	攪拌器_1	開	關	-	-
	攪拌器_2	開	關	-	-
物料	A	已裝滿	混合中	已混合	-
	B	已裝滿	混合中	已混合	-
	H	已裝滿	混合中	已混合	-
	Y	已裝滿	混合中	已混合	-
槽	槽_1	無作業	處理中	已準備	注入中
	槽_2	無作業	處理中	已準備	注入中
RFID 讀取器	開	關	-	-	-
資訊	進	出	-	-	-
感測器	開	關	動作	-	-

表 5.2 例 5.2 轉移點作業說明

轉移點名稱	作業說明	轉移點名稱	作業說明
synohronize_tokens_1	同步作業_1	mix_X	混合_X
synohronize_tokens_2	同步作業_2	mix_Y	混合_Y
synohronize_tokens_3	同步作業_3	start_re_X	開始注入_X
synohronize_tokens_4	同步作業_4	re_X	注入_X
synohronize_tokens_5	同步作業_5	activate_sensor	啟動感測器
fill_A	裝載_A	read_bar_code	讀取條碼
fill_B	裝載_B	ship_mould	運送模組
fill_H	裝載_H	pour_X	注滿_X
fill_Y	裝載_Y		

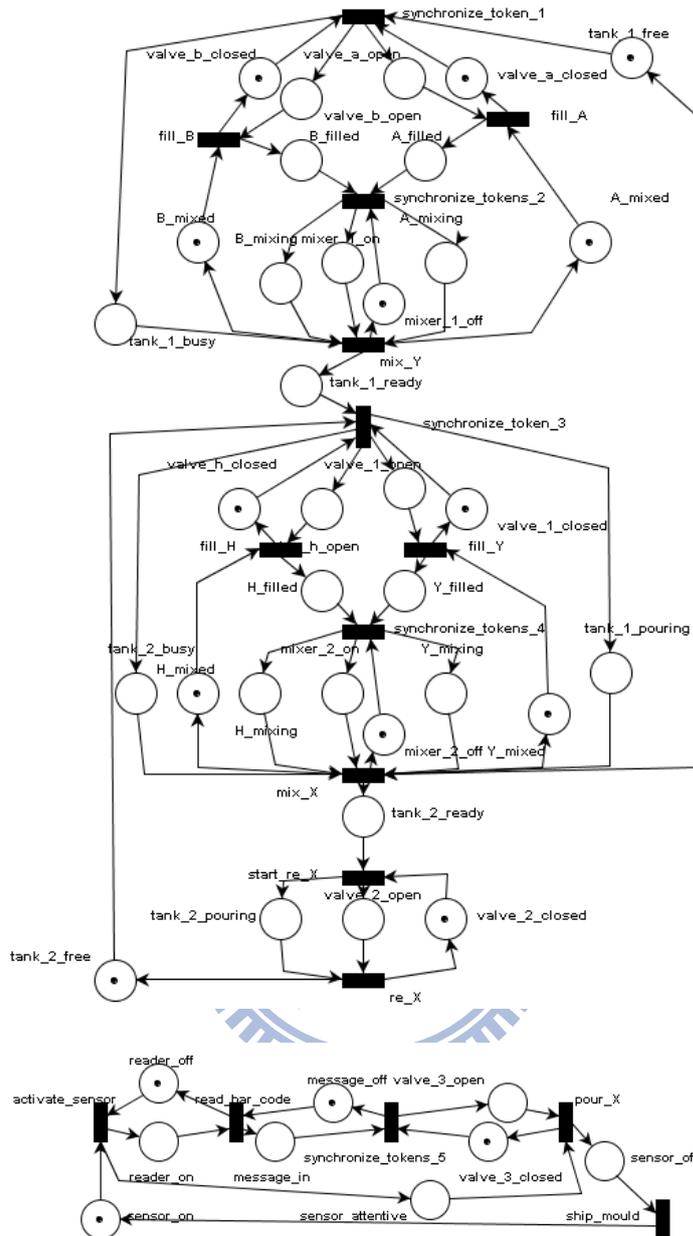


圖 5.2 灌模系統

5.1.1.1 檔案開啟及 IDEF0 轉換

下列以灌模系統中第二部分控制輸送帶為例說明程式應用，其 IDEF0 圖如圖 5.3，轉移點及暫存點對照表如表 5.3 所示。儲存格式為*.idl 格式。

檔案載入及退出檔案模組開啟三種檔案格式為*.xml、*.pnml、*.idl，此三種檔案格式中*.xml 及*.pnml 皆為 PNML 格式，只是副檔名所有不同。而*.idl 格式為 IDEF0 所使用之格式。本程式由“檔案”開啟畫面如圖 5.4。若是成功開啟*.idl 後，會自動轉換至 PNML 格式，並在畫面中出現“匯出 IDML 檔案”及“匯出 PNML 檔案”，如圖 5.5 右下選項所示。反之，若為*.xml 及*.pnml 則不會出現轉換成功資訊。

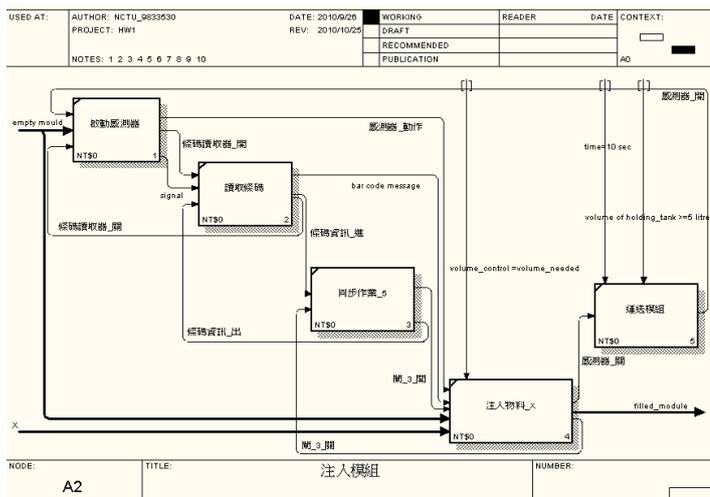


圖 5.3 控制輸送帶

表 5.3 灌模及運送裴氏圖對照列表

機台_狀態	暫存點名稱	動作	轉移點名稱
感測器_開	P1	啟動感測器	T1
感測器_動作	P2	讀取條碼	T2
感測器_關	P3	同步作業_5	T3
RFID 讀取器_開	P4	注滿_X	T4
RFID 讀取器_關	P5	運送模組	T5
資訊_進	P6		
資訊_出	P7		
閘_3_開	P8		
閘_3_關	P9		

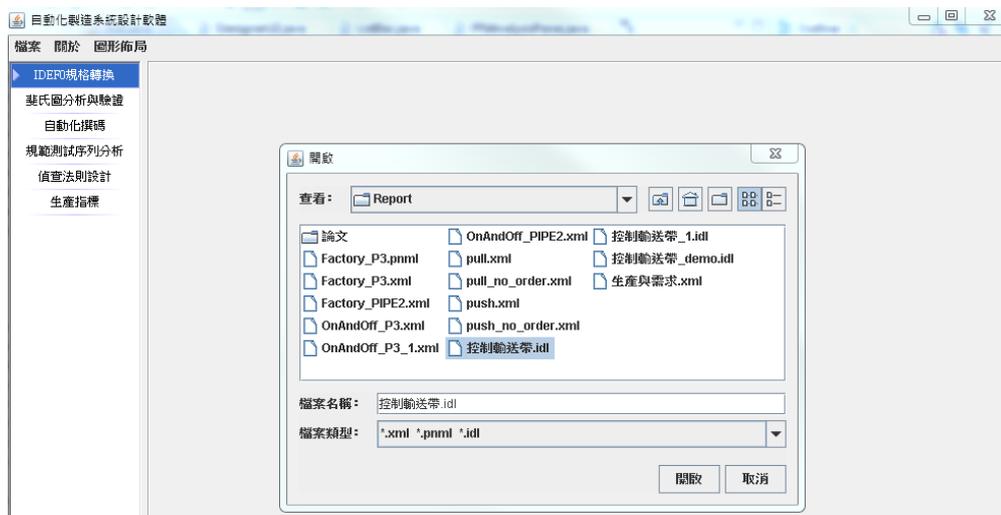


圖 5.4 開啟檔案畫面

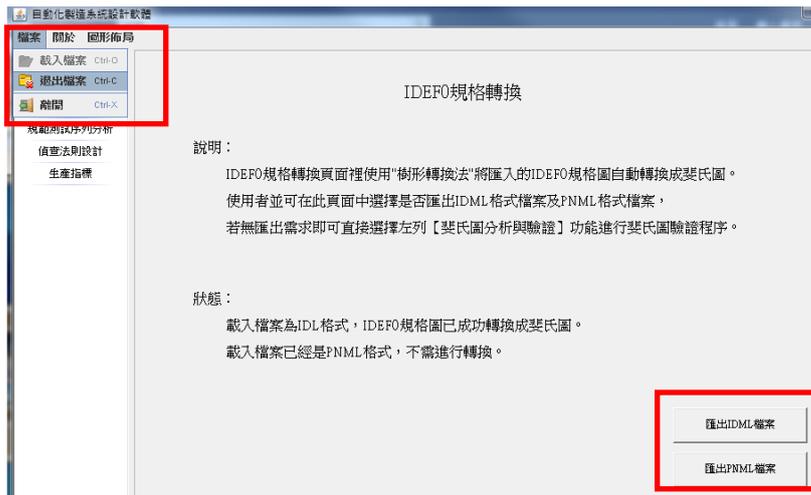


圖 5.5 退出檔案畫面

載入後亦可由“檔案”選擇退出檔案畫面如圖 5.5 左上所示。

5.1.1.2 圖形佈局

在*.idl 檔案格式轉換到 PNML 格式時，並無設定任何圖型佈局，如圖 5.6(a)。在圖形佈局中將預設 5 種圖形佈局(KK Layout, FR Layout, Spring Layout, ISOM Layout 及 Circle Layout)可供選擇。以 KK Layout 為例，可展示畫面如圖 5.6(b)所示。

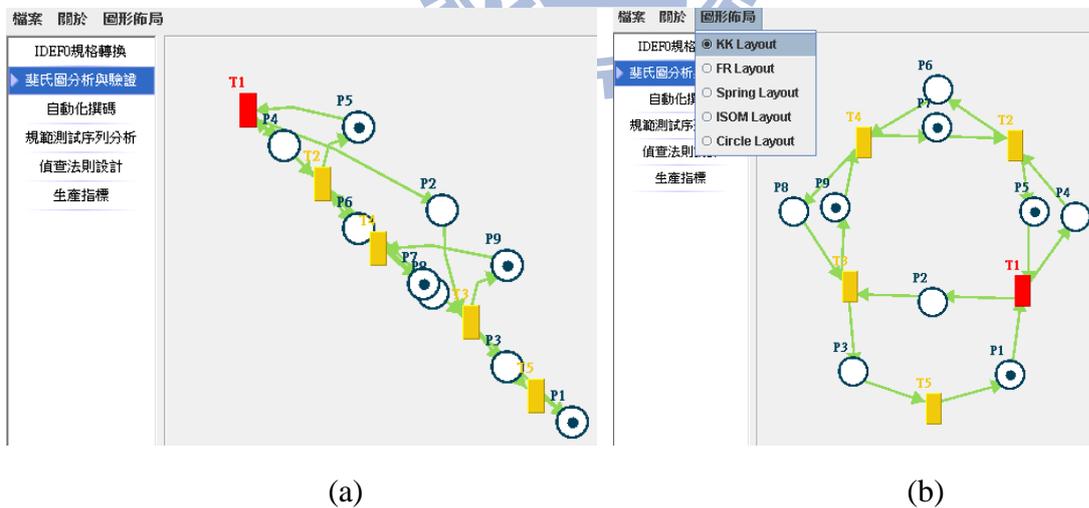


圖 5.6*.idl 轉換裴氏圖預設佈局

5.1.2 裴氏圖應用區

本節以控制輸送帶為例說明裴氏圖分析與驗證及規範測試序列分析。

5.1.2.1 裴氏圖分析與驗證

在點選裴氏圖分析與驗證後，右方共有五個子模組，分別為基本性質分析、可達狀態與鎖死分析、轉移點不變量分析、暫存點不變量分析及生產週期分析。如圖 5.7 所示。

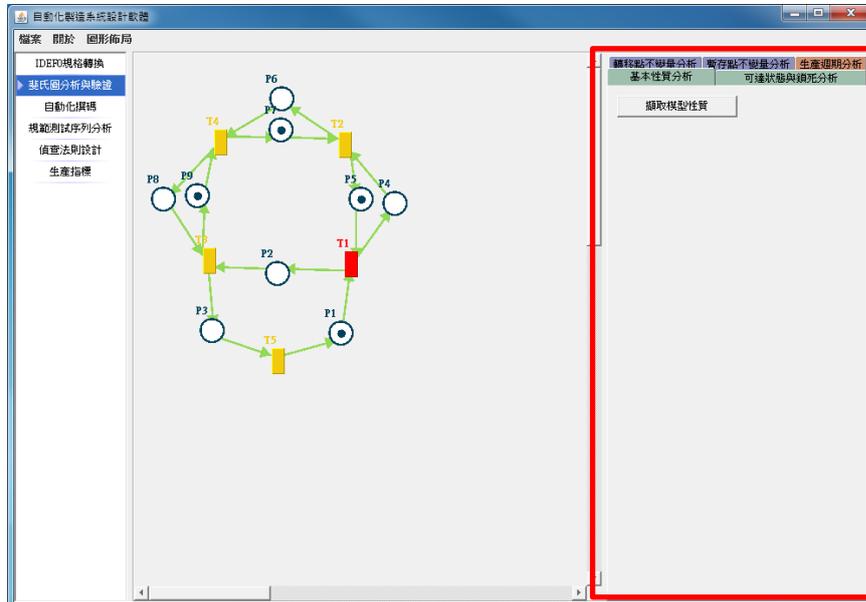


圖 5.7 裴氏圖分析與驗證

點選基本性質分析後，有”擷取模型性質”按鍵。點選後會出現可判斷出此裴氏圖之基本性質，包括模型規模(包含暫存點數及轉移點數)及規模性質(pure, ordinary, conservative, state machine, extended free choice, marked graph, strongly connected, structurally bounded, safe, live and reversible(resettable))共 11 項規模性質，如圖 5.8。

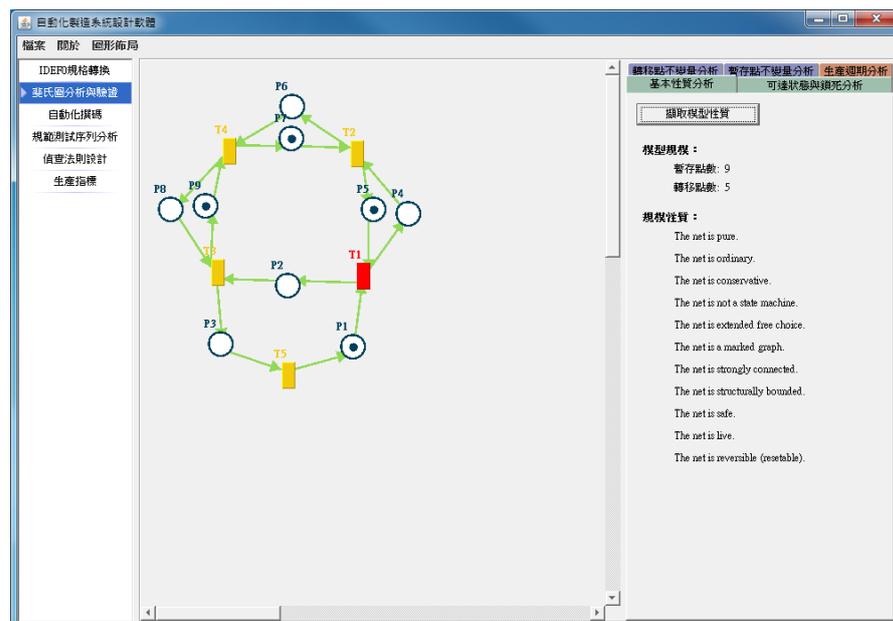


圖 5.8 擷取模型性質

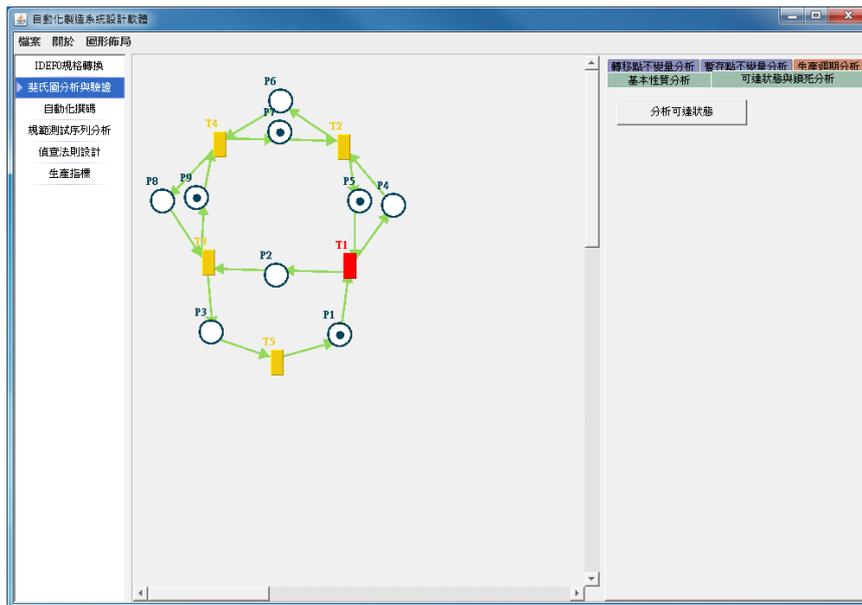


圖 5.9 可達狀態與鎖死分析

在點選”可達狀態與鎖死分析”可產生圖 5.9。點選分析可達狀態後，即產生此裴氏圖所有可達狀態，並列出所有可達狀態(s1, s2, s3, s4, s5)並說明是否為活(live)或是鎖死。在勾選任一狀態後，可出現一按鈕說明可將激發的轉移點。以圖 5.10(a)為例，s1 狀態為活，可經由 T1 激發後轉為 s2 狀態。圖 5.10(a)右上有開啟可達圖之功能，開啟後可產生可達圖如 5.10(b)。



(a) (b)
圖 5.10 可達狀態與鎖死分析

裴氏圖不變量分析共有二種，分別為轉移點不變量，及暫存點不變量分析。點選“轉移點不變量分析”後，可產生一按鈕“分析轉移點不變量”可點選。在點選“分

析轉移點不變量”列出所有轉移點不變量，以圖 5.11 為例，可得知只有一組轉移點不變量，為 T1, T2, T3, T4, T5。在勾選此組轉移點不變量時，畫面將現示出此組不變量。

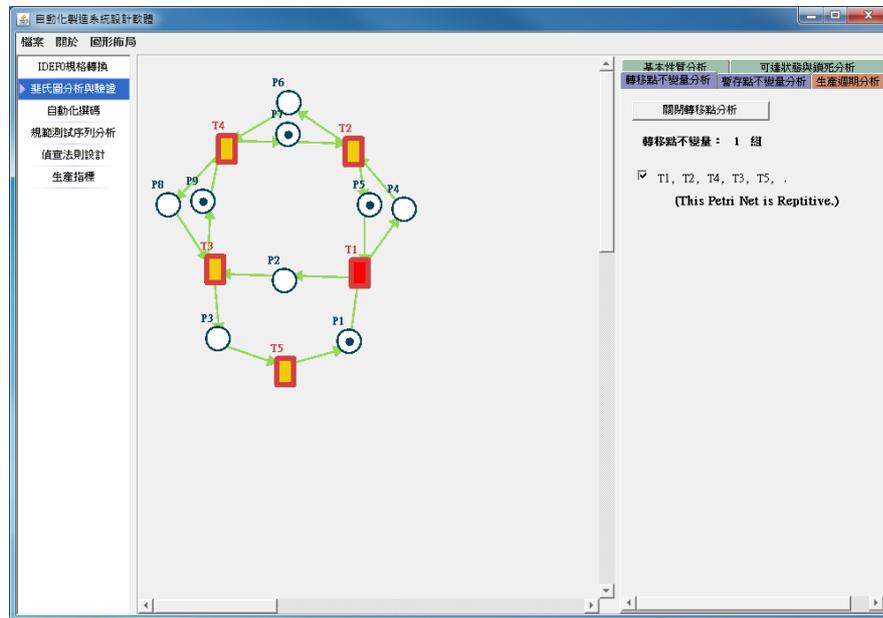


圖 5.11 轉移點不變量分析

點選“暫存點不變量分析”後，可產生一按鈕“分析暫存不變量”可點選。在點選“分析轉移點不變量”列出所有暫存點不變量，以圖 5.12 為例，可得知只有五組暫存點不變量，為 $P4 + P5 = 1$, $P2 - P4 - P6 - P8 = 0$, $P6 + P7 = 1$, $P2 - P4 - P6 + P9 = 1$, $P2 + P3 + P1 = 1$ 。在勾選此組暫存點不變量時，畫面將現示出此組不變量。

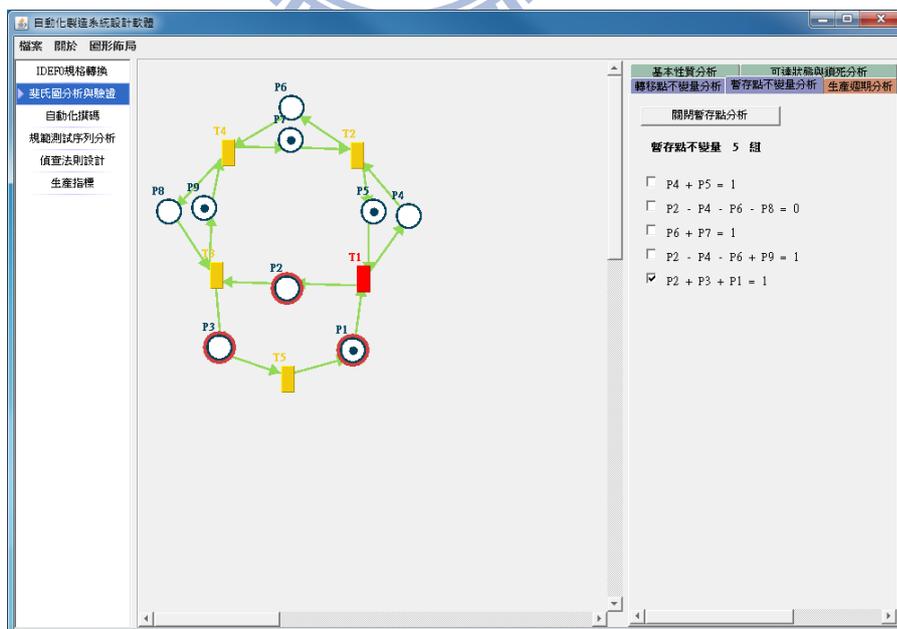


圖 5.12 暫存點不變量分析

5.1.2.2 規範測試序列分析

點選“規範測試序列分析”後右邊規範測試內會產生“分析測試路徑”，再點選後可出現圖 5.13(a)。此分析可自動列出並排除鎖死，並可點選可激發的轉移點。在此斐氏圖中，可看目前能觸發之轉移點，及其觸發路徑及狀態相對應的改變，依序為 state1 → T1 → state2 → T2 → state3 → T4 → state4 → T3 → state5 → T1 → state1。在點選轉移點前為圖 5.13(a)，點選“T1”後轉變為圖 5.13(b)。

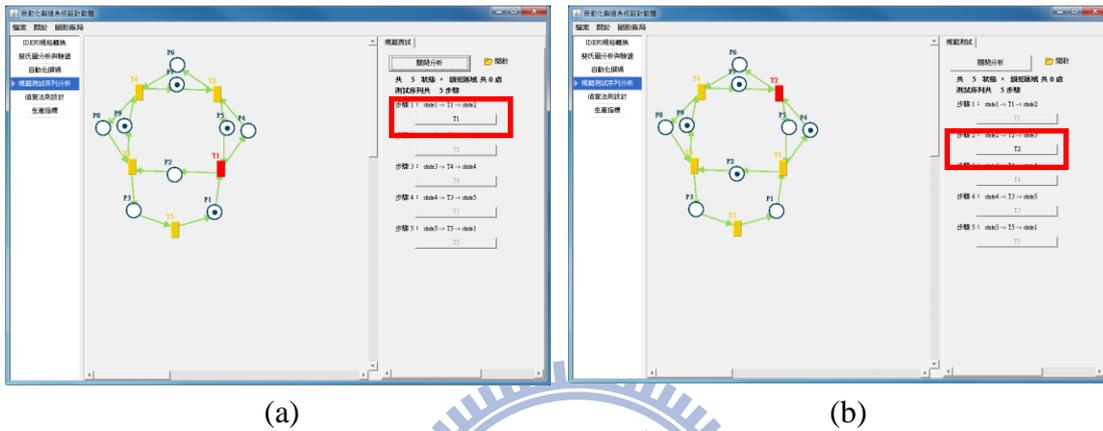


圖 5.13 規範測試序列分析

5.1.3 偵查法則設計

在點選”偵查法則設計”後，斐氏圖區會自動分成三大部分，第一部分依然為斐氏圖展示區，第二部分為偵查法則設定區，第三部分為偵查法則計算區。在 P-不變量中包含了屬性方程式及偵查法則，可人工作業找出偵查法則。在此第二部分為人工作業，利用展開樹及其對偶性質找出其偵查法則。為計算方便，程式會自行定義其暫存點及轉移點名稱，其對照表可由“暫存點，轉移點對照表”按鍵中查詢。第三部分為將所選之展開樹及載入額外邊，展示其計算過程使用。

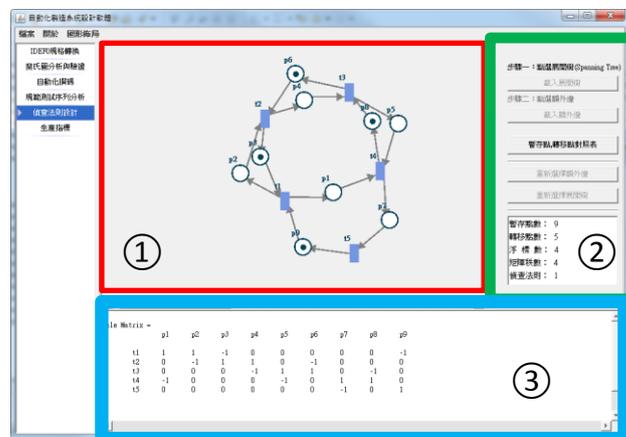


圖 5.14 偵查法則介面

第一步點選含有浮標點之暫存點，已點選之暫存點展示出紅色，如圖 5.15。選擇完成後即產生“載入展開樹”鍵。

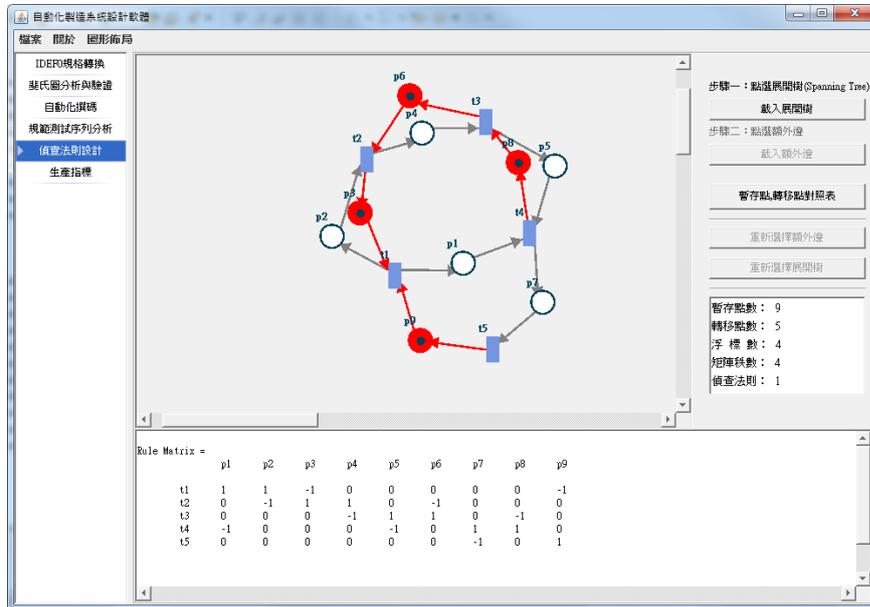


圖 5.15 偵查法則計算 1

點選“載入展開樹”鍵後，再點選載入額外邊之暫存點，產生“載入額外邊”接鍵，如圖 5.16。再點選後會馬上計算其偵查法則。載入後會跳出一視窗說明偵查法則，亦可由偵查法則計算區，查詢所有的計算推導過程。

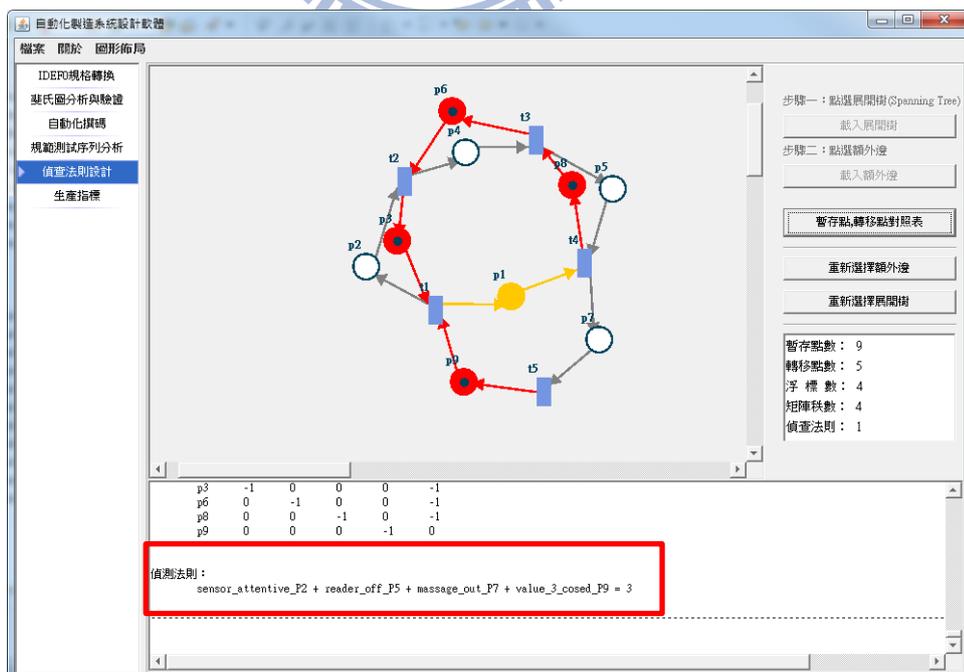


圖 5.16 偵查法則計算 2

5.2. 推拉式指標的計算

本節分別使用例 5.1 及齊默曼(A. Zimmermann)[51]的例 5.2 作為計算推拉式指標。此二實例暫存點資訊所包含需多資訊，在計算前需先作物流及資訊流分離的動作。先以例 5.2 作說明及計算，再計算例 5.1。

例 5.2：製造系統

假設有一製造系統由零件 A、B 及 C 組成一產品。零件 A 經由生產 A1 及生產 A2。待零件 B 經由生產 B1 及生產 B2。A、B 皆完成後再經由生產 AB。零件 C 亦依序經由生產 C1、生產 C2 及生產 C3 後，和 AB 一併進分生產 ABC。其物流圖如 5.17 所示。

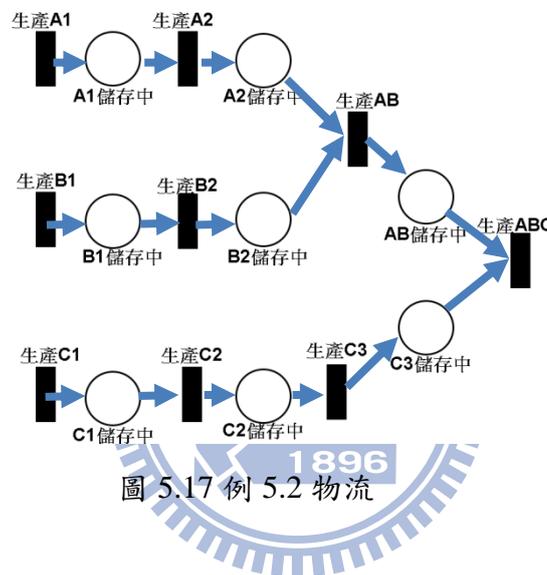


圖 5.17 例 5.2 物流

5.2.1 修正製造系統及計算推拉式指標

本論文使用生產命令資訊流作為計算生產系統的推拉式指標，因此還需將圖 5.17 的資訊流及物流分離。例 5.2 主要包含二種流向，物流及資訊流，分別以粗線及細線表示。又資訊流分推式資訊流、拉式資訊流、回授流及緩衝區資訊流。為方便說明使用四種線段作為表示(1)物流以粗線表示。(2)推式資訊流以虛線表示。(3)拉式資訊流以點虛線作為表示。(4) 回授流及緩衝區資訊流皆以細線表示，如圖 5.18 所示。

圖 5.18 流程主要是當有客戶有需求時，向暫存區 B9 拉貨。當 B9 緩衝區量不足時，再向生產 ABC 拉貨。其他的生產站點亦同，直到生產 B2 及生產 C3 時，會集中需求的資訊流至生產規劃。由生產規劃去規劃生產 B1、生產 C1 及生產 C2 的生產量。

對暫存點分析來說，將不列入計算的暫存點合併簡化後，放入程式中計算，計算過程如圖 5.19。共有 12 個生產命令資訊流(不含回授流)，其中包含有 4 個推式資訊流，

8 個拉式資訊流。即可得 $I_p = \frac{P_{Down}}{P_{All}} = \frac{8}{12} = 0.667$ ，其展示畫面如圖 5.19。

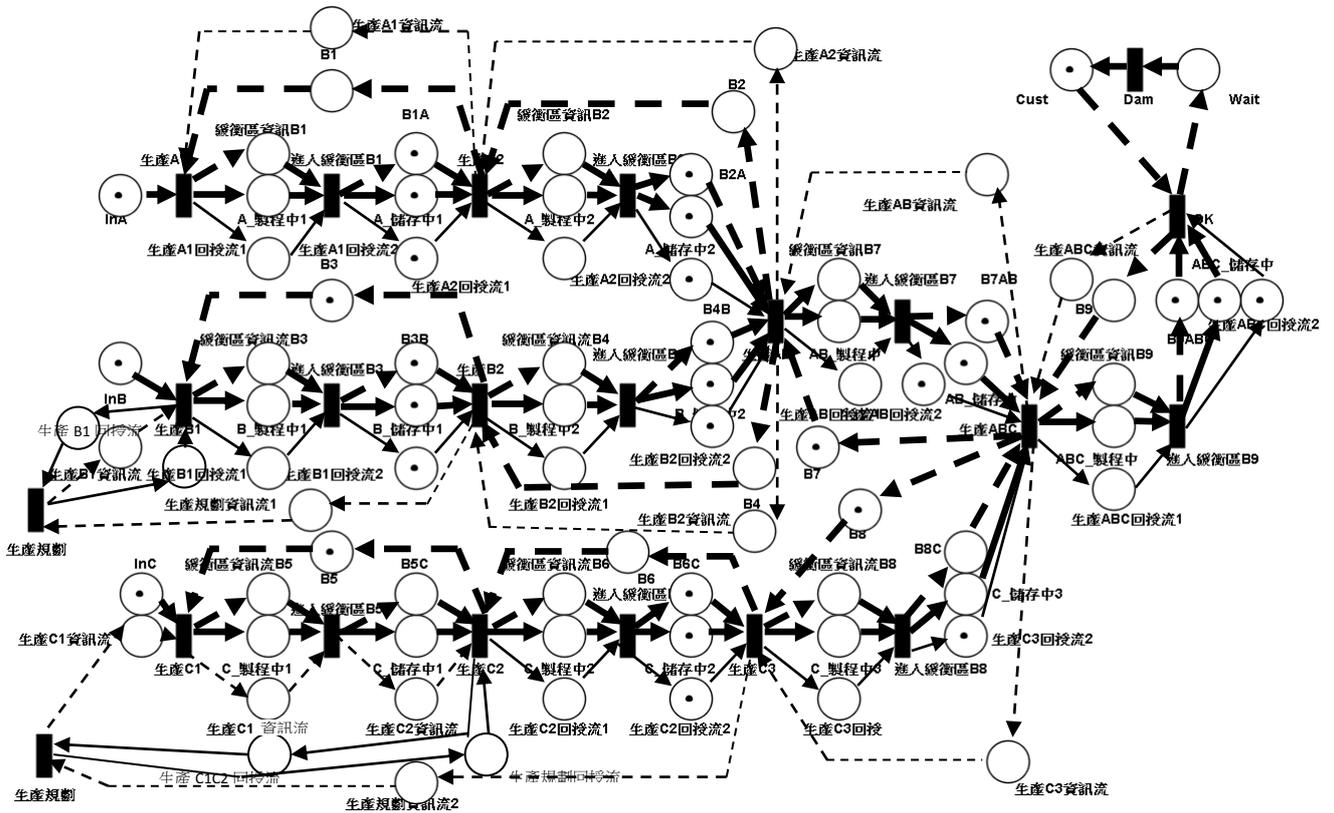


圖 5.18 修正圖 5.17

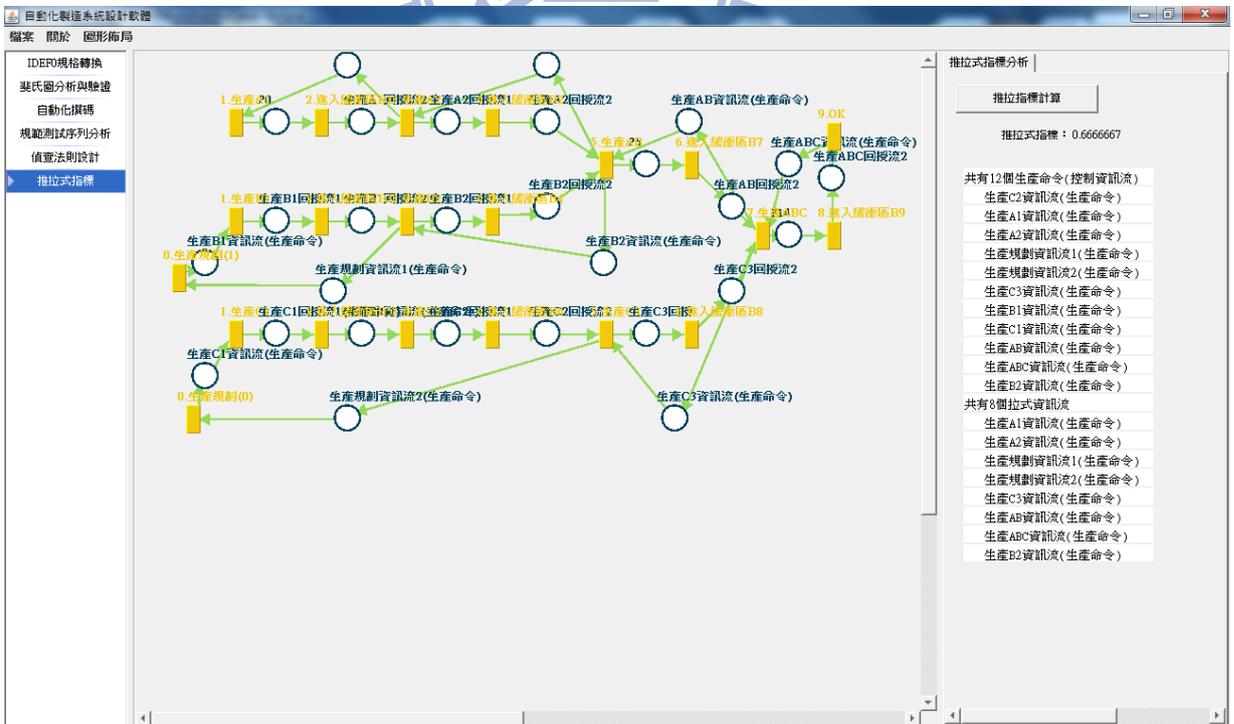


圖 5.19 例 5.2 畫面展示

5.2.2 修正灌模系統記號圖及計算推拉式指標

在例 5.1 中，雖物流由原料至灌模模組為一連串的动作，但資訊流不然。黃柏勳 [9] 所撰寫的控制系統記號圖，將其分成二大部分，分別為圖 5.20(a)及圖 5.20(b)。(1) 液體混合攪拌會依生產所需之配方，由生產規劃部門定義所需之數量(原物料比重)。(2) 當有需求時，灌模及運送會依 RFID 感測器感應容器上 RFID 資料，給予所需之溶液。在二者控制程式中，記號圖並沒有相連。

此圖 5.20(a)(b)皆為資訊流向，但激發生產動作的主要只有十個暫存點，分別為開_A_開(valve_a_open)、開_B_開(valve_b_open)、開_1_開(valve_1_open)、開_H_開(valve_h_open)及感測器開(sensor_on)等。例 5.1 以不同虛線表示推式資訊流，以點虛線表示拉式資訊流。

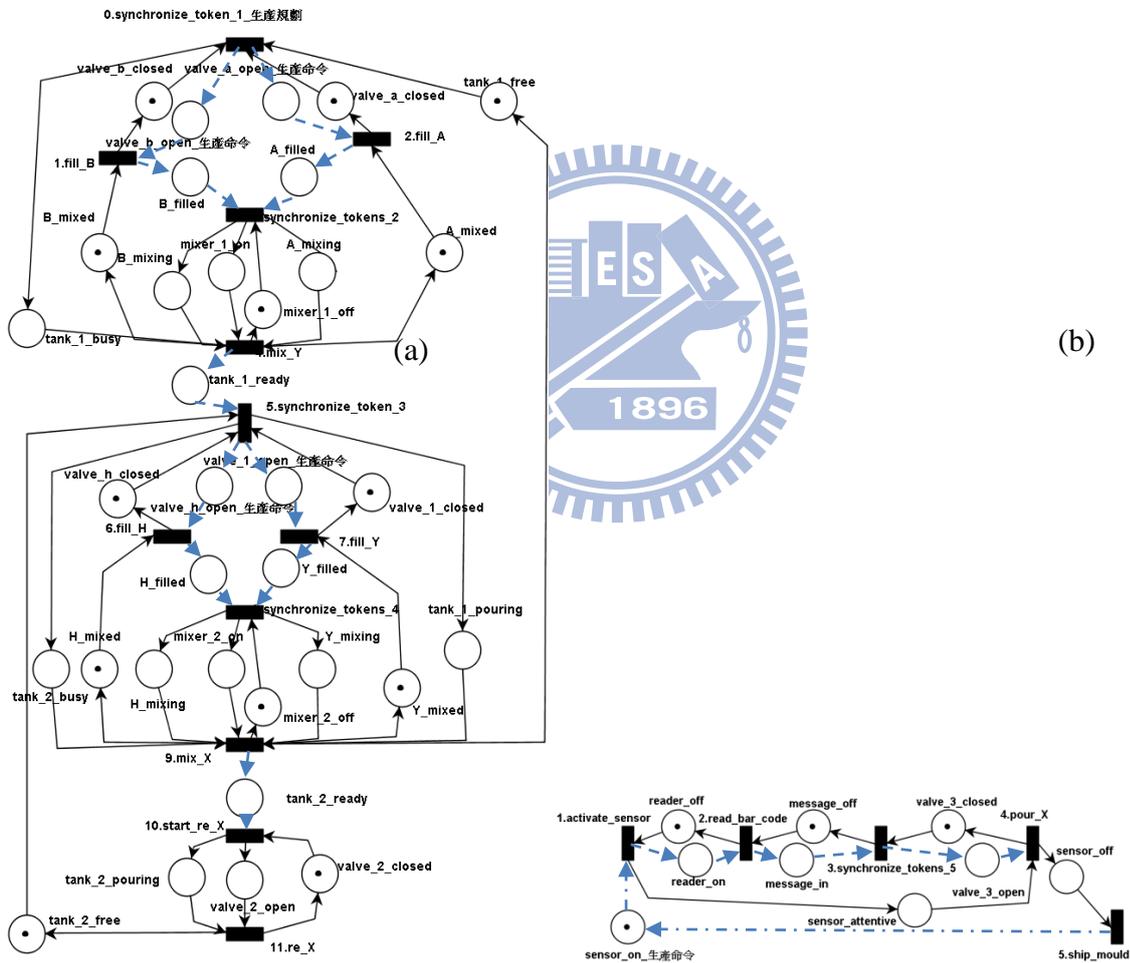


圖 5.20 灌模系統記號圖

檔案載入程式處理流程如圖 5.21 所示，共有 13 個生產命令(控制資訊流)，當中有

12 個是推式資訊流，一個為拉式資訊流可得 $I_P = \frac{P_{Down}}{P_{All}} = \frac{1}{13} = 0.0769$ 。

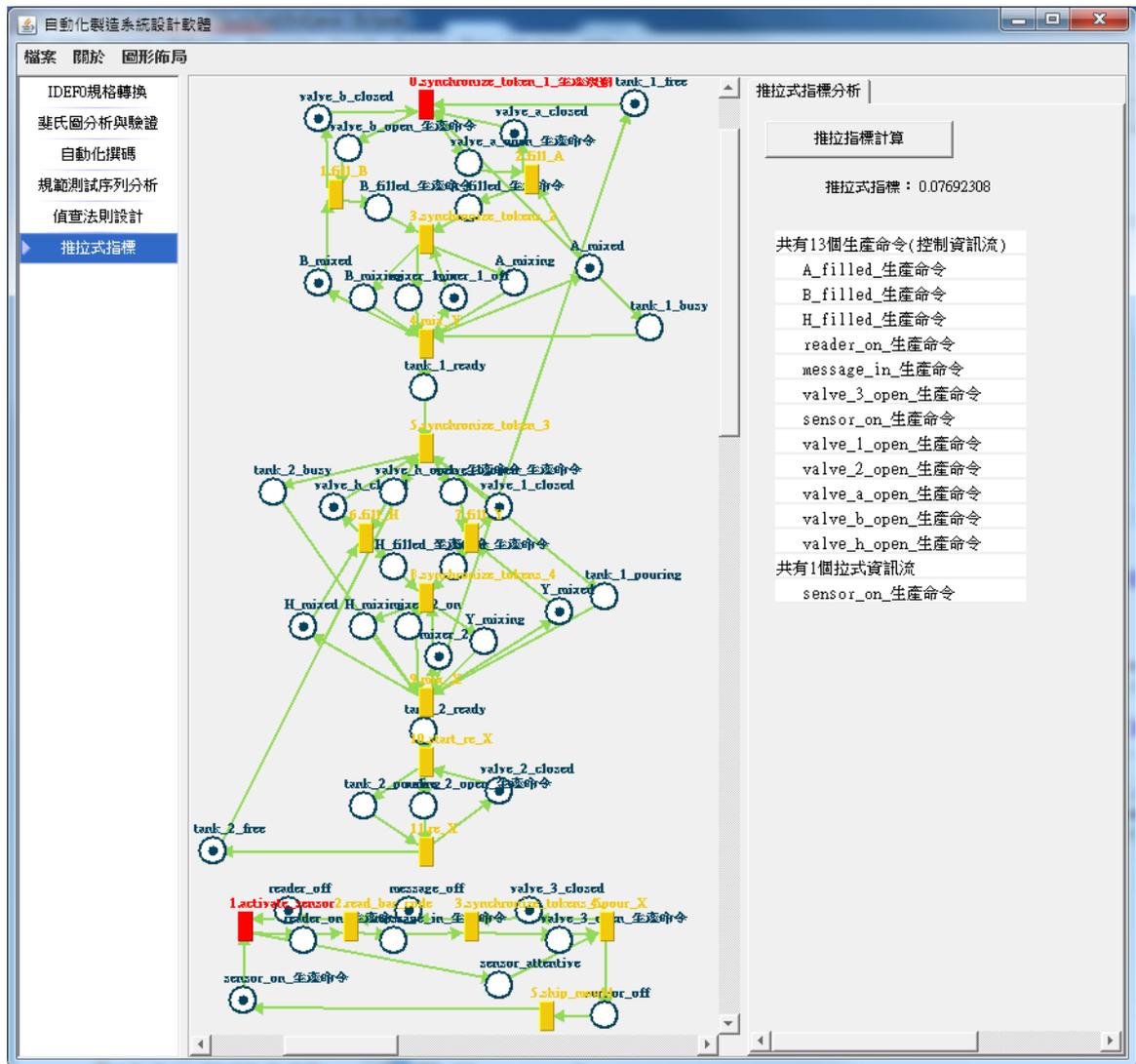


圖 5.21 灌模系統記號圖推拉式指標

第六章 結論與未來發展

本章的主要目的在說明本研究的結論及未來研究方向。本章共分成兩節。6.1 節說明本論文之結論。6.2 節說明未來研究與發展。

6.1 結論

在過去的文獻中，尚無學者研究生產流程推式及拉式的比重評比。本研究的指標是利用將生產命令數量的量化去計算生產推式及拉式之間的關係，能以生產命令的角度精確的表達推式及拉式的比重。

記號圖有著優異的數學性質的以分析整體系統，利用記號圖可快速的分析其生產命令資訊。以大量的實例說明推拉式指標的應用。以記號圖為工具，依記號圖的性質設計出一套能評估在一生產系統內推式及拉式系統的比重。

本論文以僅以生產命令資訊流計算生產系統內的推式及拉式系統比重，並無建立評比此指標之方法。亦在記號圖內作生產命令的分析，不考量在記號圖以外的範疇。

因科技的發展，利用 Java 程式語言，及結合已開發之程式套件。可由 IDEF0 轉換至裴氏圖，再由裴氏圖的性質分析其基本性質、鎖死問題、法則矩陣，最後可求得推拉式指標。上述功能皆整合在同一軟體之中，並預留其他開發空間。

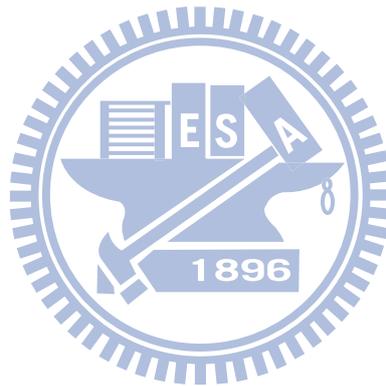


6.2 未來研究與發展

本研究僅考量如何建立推拉式指標，並無評量指標的好壞。在推拉式指標的設計方面，本論文主要只設計推拉式指標，尚未考量到各站點的比重。在生產製程中，並非所有製程的重要性都是相同的，但目前的推拉式指標設計，基本假定所有站的重要性是相同，在未來設定中可依考量依製程時間、成本、或製程的重要性作加權計算。

程式計算部分。裴氏圖可加註語言的發展使得在電子資料交換更加容易，又可加註語言綱要(XML Schema)的發展，可利用 JAXB 套件將可加註語言綱要約束(Bind)為類別(Classes)。籍由此類別，可輕易的使用 Java 語言來撰寫程式及計算推拉式指標。本程式所使用的裴氏圖可加註語言類別由黃柏勳所撰，已加快開發程式撰寫。但並非官方所定義之可加註語言綱要，尚有部分功能需再另撰寫。可待 ISO/IEC 15909 Part-3 發表後再由此規範修正其類別。

目前程式需先以人工對記號圖作物流、資訊流及機台資訊流作分離，未能作到全自動化計算。未來可撰寫自動修正與分離物流資訊流及機台資訊流的相關模組，即可直接載入檔案後直接計算，以減少人工處理時間。



參考文獻

- [1] 林潔妤，「裴氏圖軟體撰碼問題中自動規範測試系統的設計與實作」，國立交通大學工業工程與管理系碩士論文，2008年。
- [2] 范植宇，「階層轉換法的自動化：從規格到設計的樹形轉換」，國立交通大學工業工程與管理系碩士論文，2009年。
- [3] 梁高榮，「自動化製造系統內的即時故障察覺」，機械工業，五月，127-138頁，2006年。
- [4] 梁高榮，「虛體製造系統內的多緒架構」，機械工業，五月，241-261頁，2004年。
- [5] 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年。
- [6] 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年。
- [7] 陳音帆，「裴氏圖導向控制器開發系統的設計與實作」，國立交通大學工業工程與管理研究所碩士論文，2008年。
- [8] 陳音帆，梁高榮，「自動化製造系統裡裴氏圖模式的撰碼自動化」，機械工業，十二月，111-121頁，2006年。
- [9] 黃柏勳，「建立自動化製造系統的開放資料伺服器連接架構」，國立交通大學工業工程與管理研究所碩士論文，2010年。
- [10] 簡禎富、施義成、林振銘及陳瑞坤，「半導體製造技術與管理」，國立清華大學出版社，2006年。
- [11] Phillips, L. A., 莊金松 編譯, XML 全方位完全剖析, 第三波資訊, 2001.
- [12] Andreas, D., and Alf, K., Beyond Manufacturing Resource Planning (MRP II), Springer, 1998.
- [13] Bonney, M. C., Zhang, Z., Head, M. A., Tien, C. C., and Barson, R. J., "Are Push and Pull Systems Really So Different?," International Journal of Production Economics, Vol. 59, pp. 53-64, 1999.
- [14] British Standards Institution, BS 5192 : Guide to Production Control Part I, London, 1993.
- [15] Campos, J., Chiola, G., and Silva, M., "Ergodicity and Throughput Bounds of Petri Nets with Unique Consistent Firing Count Vector," IEEE Transactions on Software Engineering, Vol. 17, No. 2, pp. 117-125, Feb, 1991.
- [16] David, R., and Alla, H., Discrete, Continuous, and Hybrid Petri Nets, Springer, 2005,
- [17] Goddard, W. E., and Brooks, R. B., "Just-In-Time: A Goal for MRPII, Readings in Zero Inventory," Conference Proceedings APICS, 1984.
- [18] Hall, R.W., Zero Inventories, American Production and Inventory Control Society, 1983.

- [19] Harrison, T. P., Lee, H. L., and Neale, J. J., *The Practice of Supply Chain Management*, Springer, 2003.
- [20] Hillah, L. M., Kordon, F., Petrucci, L., and Tr`eves, N., "PNML Framework: An Extendable Reference Implementation of the Petri Net Markup Language," *Lecture Notes in Computer Science*, Vol. 6126, pp. 318-327, 2010.
- [21] <http://druyanov.tripod.com/RDBMS/BPWin.html/>, BPwin, Feb. 1, 2011.
- [22] <http://www.apics.org/default.htm>, APICS, Feb. 1, 2011.
- [23] <http://www.bsigroup.tw/>, BSI, Feb. 1, 2011.
- [24] <http://www.idef.com/>, Integrated DEFinition Methods, Feb. 1, 2011.
- [25] <http://www.iec.ch/>, International Electrotechnical Commission, Feb. 1, 2011.
- [26] <http://www.informatik.uni-hamburg.de/TGI/PetriNets/>, Petri Net World., Feb. 1, 2011.
- [27] <http://www.iso.org/iso/home.html/>, International Organization for Standardization, Feb. 1, 2011.
- [28] http://www.Java.com/zh_TW/, Java, Feb. 1, 2011.
- [29] <http://www.pnml.org/>, Petri Net Markup Language, Feb. 1, 2011.
- [30] <http://www.standardsinfo.net/info/index.html/>, ISO/IEC, Feb. 1, 2011.
- [31] <http://www.w3.org/>, World Wide Web Consortium, Feb. 1, 2011.
- [32] <http://www.w3.org/DOM/>, Document Object Model, Feb. 1, 2011.
- [33] <http://www.w3.org/XML/>, Extensible Markup Language, Feb. 1, 2011.
- [34] <http://www2.informatik.hu-berlin.de/lehrestuehle/automaten/tools/>, Integrated Net Analyzer, Feb. 1, 2011.
- [35] James, F. C. and John, H. B., *APICS Dictionary*, APICS, 2005.
- [36] Kenworthy, J. and Little, D., "When Push Comes to Shove is MRPII Infinite Push or Pull?," *BPICS Control*, University of Liverpool, pp. 31-32, 1995.
- [37] Lee, L. C., "A Comparative Study of The Push and Pull Productions Systems," *International Journal of Operations and Production Management*, Vol. 9, No. 4, pp. 5-18, 1989.
- [38] Liang, G. R. and Hong, H. M. "Hierarchy Transformation Method for Repetitive Manufacturing System Specification, Design, Verification and Implementation", *Computer Integrated Manufacturing Systems*, Vol. 7, No. 3, pp. 191-205, 1994.
- [39] Magott, J., "Performance Evaluation of Concurrent System Using Petri Net," *Information Processing Letters*, Vol. 18, pp. 7-13, 1984.
- [40] Mascolo, M. D., Frein, Y., Dallery, Y., and David, R., "A Unified Modeling of Kanban Systems Using Petri Nets," *The International Journal of Flexible Manufacturing Systems*, Vol. 3, No. 3-4, pp. 275-307, 1991.

- [41] Morioka, S. and Yamada, T., "Performance Evaluation of Marked Graphs by Linear Programming," *International Journal of Systems Science*, Vol. 22, No. 9, pp. 1541-1552, 1991.
- [42] Murata, T., "Petri Nets: Properties, Analysis and Applications," *Proceedings of The IEEE*, Vol. 77, No. 4, pp. 541-580, 1989.
- [43] Ohno, T., *Toyota Production System: Beyond Large-Scale Production*, Productivity Press Inc., 1989.
- [44] Oxley, J. G., *Matroid Theory*, Oxford University Press, 1992.
- [45] Ramamoorthy, C. V. and Gray, S. H., "Performance Evaluation of Asynchronous Concurrent System Using Petri Nets," *IEEE Transactions on Software Engineering*, Vol. 6, No. 5, pp. 440-449, 1980.
- [46] Shingō, S., *A Study of the Toyota Production System from an Industrial Engineering Viewpoint*, Productivity Press, 1989.
- [47] Toni, D. A., Caputo, M., and Vinelli, A. M., "Production Management Techniques: Push-Pull Classification and Application Conditions," *International Journal of Operations and Production Management*, Vol. 8, No. 2, pp. 35-51, 1984.
- [48] Venkatesh, K., Zhou, M. C., Kaighobadi, M., and Caudill, R., "A Petri Net Approach to Investigating Push and Pull Paradigms In Flexible Factory Automated Systems," *International Journal of Production Research*, Vol. 34, No. 3, pp. 592-620, 1998.
- [49] Villa, A. and Watanabe, T., "Production Management: Beyond the Dichotomy between 'Push' and 'Pull'," *Computer Integrated Manufacturing Systems*, Vol. 6, pp. 53-63, 1993.
- [50] Viswanadham, N., Narahari, Y., and Johnson, T. L., "Deadlock Prevention and Deadlock Avoidance in Flexible Manufacturing Systems Using Petri Net Models," *IEEE Transactions on Robotics and Automatic*, Vol. 6, No. 6, pp. 713-723, 1990.
- [51] Zimmermann, A., Rodriguez, D., and Silva, A., "A Two Phase Optimization Method for Petri Net Models of Manufacturing Systems," *Journal of Intelligent Manufacturing*, No. 12, pp.409-420, 2001.

附錄一

本附錄說明在推拉式指標計算之方法。下列以 Eclipse 軟體 Java 語言開發。原始

程式碼說明如下：

```
package Pduct.Ind;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Image;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Date;
import java.util.List;
import java.util.Properties;
import javax.swing.AbstractAction;
import javax.swing.BorderFactory;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.WindowConstants;
import rule_design_Module.petrinet.component.PetriNet;
import rule_design_Module.petrinet.component.RG;
import rule_design_Module.petrinet.component.RGState;
import sun.awt.util.IdentityArrayList;
import usual.tool.ShowImage;
import vms.modules.INAAnalysisFileGenerator;
import vms.modules.INAInputNetGenerator;
import vms.modules.ModelPropertiesAnalysis;
import vms.modules.ReachabilityHandle;
import AMSdesigner.DesignerUI;
import analysis_Module.app.gui.DrawPane;
import org.dom4j.Document;
import org.dom4j.DocumentException;
import org.dom4j.io.SAXReader;
import org.omegahat.Environment.DataStructures.integer;

import cern.colt.list.IntArrayList;
public class Ind extends javax.swing.JPanel {

    private JButton jButton1;
    private JLabel jLabel;
    private AbstractAction abstractAction1;
    private PetriNet pn;
    private JLabel[] jLabel;
    private DrawPane jp;
    private File file;
    private Document Doc;
    private ArrayList T_I= new ArrayList();//生產規劃ID及推式生產轉移點
    private ArrayList P_I= new ArrayList();//生產命令ID(含推拉)
```

```

private ArrayList Ph_I= new ArrayList();//找出推式生產命令
private ArrayList T_S=new ArrayList();//生產順序
private ArrayList P_T=new ArrayList();//暫存點下一個轉移點
private ArrayList T_P=new ArrayList();//暫存點前一個轉移點

public Ind(){
    initGUI();
}
private void initGUI() {
    try {
        jLabel = new JLabel[0];
        this.setPreferredSize(new java.awt.Dimension(295, 546));
        this.setLayout(null);
        this.setSize(295, 546);

        {
            jButton1 = new JButton();
            jButton1.setBounds(12, 12, 150, 30);
            jButton1.setAction(abstractAction1());
            jButton1.setText("推拉式指標計算");
            add(jButton1);
            jButton1.setVisible(true);
        }

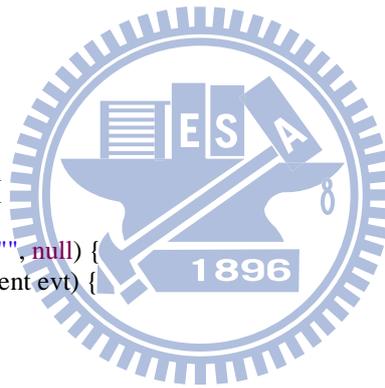
    } catch (Exception e) {
        e.printStackTrace();
    }
}
public AbstractAction abstractAction1(){
    if(abstractAction1 == null) {
        abstractAction1 = new AbstractAction("", null) {
            public void actionPerformed(ActionEvent evt) {
                jLl.setVisible(true);
            }
        };
    }
    return abstractAction1;
}
public void setVisibelfalse() {
    this.removeAll();
}

public void Ck_Trn(PetriNet pn){
    this.pn=pn;

    //找出生產規劃(推式專用)
    for (int i=0;i<pn.getTransitionNumber();i++){
        if(pn.getTransition(i).getTransitionName().indexOf("生產規劃")>=0){
            T_I.add(pn.getTransition(i).getTransitionID());
            //System.out.println(pn.getTransition(i).getTransitionName().charAt(0));
        }
    }
    //找出生產命令(含推拉)及所有暫存點的下一個轉移點
    for(int i=0;i<pn.getPlaceNumber();i++){

        for(int j=0;j<pn.getTransitionNumber();j++){
            for(int k=0;k<pn.getTransition(j).getInputsNumber();k++){
                if(pn.getTransition(j).getInput(k)==i){

```



```

// System.out.println(pn.getTransition(j).getInput(k)+" "+i);
// System.out.println(pn.getTransition(j).getTransitionName());
// System.out.println(pn.getPlace(i).getPlaceName());
double dPT=Double.valueOf(pn.getTransition(j).getTransitionName().substring(0,2)).doubleValue();
//System.out.println(dPT);
P_T.add(dPT);//找出下一個轉移點
}
}
for(int k=0;k<pn.getTransition(j).getOutputsNumber();k++){
if(pn.getTransition(j).getOutput(k)==i){
double dPT=Double.valueOf(pn.getTransition(j).getTransitionName().substring(0,2)).doubleValue();
T_P.add(dPT);
}
}
}
// System.out.println(T_P.get(i)+" "+pn.getPlace(i).getPlaceName()+" "+P_T.get(i));
if(pn.getPlace(i).getPlaceName().indexOf("生產命令")>=0){

// P_I.add(pn.getPlace(i).getPlaceID());
P_I.add(i);
}
if(pn.getPlace(i).getPlaceName().indexOf("生產命令")>=0
&& Double.valueOf(T_P.get(i).toString()).doubleValue()
< Double.valueOf(P_T.get(i).toString()).doubleValue()){
Ph_I.add(pn.getPlace(i).getPlaceName());
//System.out.println(Ph_I.get(i));
//P_I.add(pn.getPlace(i).getPlaceID());
}

}
System.out.println("共有"+P_I.size()+"個生產命令");
for(int i=0;i<P_I.size();i++){
System.out.println(" "+pn.getPlace(Integer.parseInt(P_I.get(i).toString())).getPlaceName());

}
System.out.println(Ph_I.size()+"個推式");
for (int i =0;i<Ph_I.size();i++){
System.out.println(" "+Ph_I.get(i));
}

jLI = new JLabel();
jLI.setText("推拉式指標： "+(float)(P_I.size()-Ph_I.size())/(float)P_I.size());
jLI.setVisible(true);
jLI.setBounds(60, 60, 160, 20);
add(jLI);

}
}

```