

東 海 大 學

工業工程與經營資訊學系

碩士論文

以知識推論法則建構機械加工系統控制

研 究 生：張瀟方
指 導 教 授：黃欽印 博士
 陳武林 博士

中 華 民 國 一 〇 七 年 一 月

Knowledge Driven Approach for Machining System Control

By
Ching-Fang Chang

Advisors : Dr. Chin-Yin Huang
Dr. Wu-Lin Chen

A Thesis

Submitted to the Institute of Industrial Engineering and Enterprise
Information at Tunghai University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in
Industrial Engineering and Enterprise Information

January 2018
Taichung , Taiwan

以知識推論法則建構機械加工系統控制

學生：張瀟方

指導教授：黃欽印 博士

陳武林 博士

東海大學工業工程與經營資訊研究所

摘 要

智慧工廠是工業 4.0 中主要精隨之一，其主要是智慧型的生產設備聯網並與實體達到「虛實合一」的生產系統，使管理者迅速掌握廠內狀況。簡單來說，工業 4.0 就是在連結與製造相關的元素。以機械加工為例，工具機透過物聯網將設備訊號整合後，便可以隨時掌握機台狀況，當有警報發生時也可以及時處理。目前，已經有許多工廠為了提高生產力和降低人工所帶來的錯誤，使用了多種自動化的機械設備，但機械設備在協同運作上仍有些困難。例如：當機台發生故障時，判定錯誤如何排除以及其他設備如何因應多數仍仰賴人工處理；當工廠內部要擴充或減少機台時，往往需要重新進行設備規劃整合，甚至需要重新撰寫程式。這些困難將會是阻礙自動化朝向協同合作智慧化前進的部份原因。

本研究架構建立了實體層、虛擬層及知識推論層，實體層為機械設備如工具機、機械手臂、無人搬運車等，虛擬層則透過 C# 以及精密機械研究發展中心 (PMC) 所開發的 SkyMars 軟體截取實體層的資料，而知識層面將建立本體論框架並透過 CLIPS 進行法則推論。本體論具有概念化、正規化、明確及可分享的特性，讓不同領域專家理解與達到知識共享。CLIPS 則是以 C 語言寫成的法則式推論專家系統，具有推論能力，會藉由事實 (facts) 的狀態觸發法則 (rules) 以產生新事實。因此，本研究系統將透過虛擬層所蒐集到的實體層資料進入到知識推論層面，產生結果後回傳至虛擬層再進行實體層控制，以解決智慧工廠內如機械設備如何能在發生錯誤時，在能不影響生產過程下使設備之間互相協調，或者當工廠內要增加或減少機台時應如何快速調整。最後，本研究利用 C# 建置介面來整合實體層與知識推論層並情境模擬與驗證，證明專家系統確實能解決前面所提到之問題並使製造業邁向更智慧化的一步。

關鍵字詞： 工業 4.0、網宇實體、本體論、專家系統、知識法則推論

Knowledge Driven Approach for Machining System Control

Student : Ching-Fang Chang

Advisors : Dr. Chin-Yin Huang

Dr. Wu-Lin Chen

Department of Industrial Engineering and Enterprise Information
Tunghai University

ABSTRACT

Industry 4.0 is a new trend of manufacturing management. It creates what has been called the "smart factory." The core value of the smart factory is that it combines the intelligent equipment with communication technology to let the managers online monitor the manufacturing situations and collaborate with the intelligent equipment. Although industries intend to apply the concept of the smart factories in the manufacturing, they still has difficulties about how to let different kinds of equipment collaborate with one another intelligently. For example, an unexpected rush order or a machine error may shut down the whole machining system to reschedule. Meanwhile, if the production line needs to add or reduce a machine, it usually needs to re-plan and even to re-write the control program. Thus, the manufacturing system usually relies on humen to resolve the error when some small unexpected errors occur. These difficulties are a part of the reasons that hinder the applications of the smart factories.

This research defines three layers to intelligently manage a machining shop: the physical layer, the virtual layer and the knowledge reasoning layer. The physical layer contains the mechanical equipment such as CNC machines, robots, AGVs, etc. The virtual layer uses C# and the SkyMars which was developed by the Precision Machining Center of Taiwan to capture the data from the physical layer. The knowledge layer uses ontology to establish the knowledge framework and utilizes CLIPS to build the knowledge rules. Facts in the physical layer will trigger the knowledge rules to generate new facts as an actuator command to plan new jobs, to execute the planned jobs, or to resolve errors. At the end, this research will use C# to establish interface which integrates with the physical layer and the knowledge reasoning layer to prove the expert system can truly let the manufacturing be more intelligent.

Keywords: Industry 4.0, CPS, Knowledge Reasoning, Ontology, Expert System

誌謝

研究所的生活一眨眼就走到了尾端，因為是五年一貫生的關係所以從大學四年級就進到了研究室學習，當時對於很多自動化的觀念非常薄弱，但是在指導教授黃欽印老師和陳武林老師的細心與耐心指導下，我進入了狀況而且找出了自己的研究方向，老師們除了每周抽空與我們討論進度之外，也經常帶著我們到不同公司參訪，甚至帶領我們參加國際研討會，讓我們增加國際視野也讓自己更加有信心，這兩年下來受益良多。

在讀研究所的過程中，有著同儕的力量也同時是我堅持下來的原因之一，謝謝學長姐鳳芷、昊騰、紹璿、顏晁、啟鑫、依璇還有上 Auto 的各個學長姐平時在我有困惑時給予協助；謝謝研究所好友兆廷、碧玲、子寒、靖以、珮儀和盛皇一起共患難了這兩年研究所生涯，除了一起開會、討論報告也一起放鬆娛樂甚至耍寶，這些回憶我將會一直放在心裡；謝謝研究所的學妹宜禎幫忙處理了研究室的大小事，祝福學弟妹們在未來做研究的路上也都能夠擁有非常高的成就。

謝謝多年的好友們在身旁的陪伴，讓我在低潮的時候有可以放鬆宣洩的出口，謝謝十二年閨蜜唯禎、璨語、詩芸、佳心、昀臻、瑩珊、萌萌經常聽我吐苦水和一起放鬆；謝謝佳心、高高在我寫程式遇到困難時給予我幫助；謝謝雅婕、惟凱在我忙碌的研究所生活中默默地給予我心靈上的鼓勵和陪伴；謝謝大學的好友們，雖然大家已經各奔東西，但仍不忘彼此聯繫與耍寶，感謝你們在我忙碌的時候沒有忘記我的存在；謝謝高中數學老師湯姆克魯施定期關心我的學習生活狀況。

最感謝的是我的家人們，提供我學習環境和一個溫暖的家，在我迷茫的時候也給予我方向，讓我也堅持完成學業，在我需要幫忙的時候都無條件地答應我，謝謝你們在我成長路上的付出和陪伴。

最後，謝謝自己沒有放棄完成碩士學業，才能在這一路上學到了很多想像不到的知識。祝福自己和身旁的大家在未來都能乘風破浪。

張瀟方 謹誌於
東海大學工業工程與經營資訊學系研究所
中華民國一〇七年一月

目錄

摘要.....	i
ABSTRACT.....	ii
誌謝.....	iii
目錄.....	iv
表目錄.....	vi
圖目錄.....	vii
第一章 緒論.....	1
1.1 研究背景與動機.....	1
1.2 研究問題與目的.....	2
1.3 研究流程與步驟.....	3
1.4 論文架構.....	4
第二章 文獻探討.....	5
2.1 工業 4.0 與智慧工廠.....	5
2.2 本體論.....	6
2.2.1 本體論介紹.....	6
2.2.2 本體論工具.....	8
2.3 專家系統.....	10
2.3.1 專家系統介紹.....	10
2.3.2 專家系統知識表現.....	13
2.3.3 法則式推論專家系統-CLIPS.....	15
2.4 製造環境的本體理論.....	17
2.5 通訊協定格式.....	19
第三章 研究方法.....	20
3.1 系統架構層.....	20
3.2 建置製造環境的知識本體框架.....	22
3.3 建置專家系統 CLIPS 法則.....	26

第四章 情境模擬.....	45
4.1 開發環境和工具.....	45
4.2 實體設備介紹.....	46
4.2.1 截取 CNC 資訊之軟體-SkyMars Professional	46
4.2.2 機械手臂-HIWIN RA605	47
4.2.3 模擬無人搬運車(AGV)動作之 PLC-Mitsubishi FX3G-24MR.....	48
4.3 情境描述與環境設定.....	49
4.4 設備與專家系統整合.....	50
4.5 情境推論結果.....	53
4.6 小結.....	61
第五章 結論與未來發展方向.....	62
參考文獻.....	64

表目錄

表 2.1 常用本體論發展工具功能分析.....	8
表 2.2 專家系統發展史.....	10
表 4.1 本研究電腦開發環境.....	45
表 4.2 本研究軟體需求工具.....	45

圖目錄

圖 1.1 研究方法與步驟.....	3
圖 1.2 論文架構圖.....	4
圖 2.1 工業 4.0 智慧工廠概念圖.....	5
圖 2.2 典型專家系統結構圖(Giarratano & Riley, 1998).....	12
圖 2.3 專家系統運作基本概念.....	12
圖 2.4 語意網路知識表示法簡易範例.....	14
圖 2.5 Deftemplate 範例說明圖.....	15
圖 2.6 Defrule 範例說明圖.....	16
圖 2.7 製造環境本體主要類別和物件屬性 (Lemaignan et al., 2006).....	17
圖 2.8 製造環境本體論知識模型 (Alsafi & Vyatkin, 2010).....	18
圖 2.9 TCP/IP 架構層.....	19
圖 3.1 簡易系統架構層.....	20
圖 3.2 詳細系統架構圖.....	21
圖 3.3 監控架構層.....	22
圖 3.4 本研究之製造環境本體知識類別.....	23
圖 3.5 知識本體類別之物件屬性.....	24
圖 3.6 類別與物件屬性關聯圖.....	25
圖 3.7 本研究建立之情境圖.....	26
圖 3.8 情境圖之各項圖示說明.....	27
圖 3.9 本研究正常加工之流程圖.....	27
圖 3.10 各設備之狀態資訊.....	28
圖 3.11 本研究中與各 Order 相關之 deftemplate 設計.....	29
圖 3.12 本研究中與各 Storage Devices 相關之 deftemplate 設計.....	29
圖 3.13 本研究中與各 Handling Devices 相關之 deftemplate 設計.....	29
圖 3.14 本研究中與各 Machining Devices 相關之 deftemplate 設計.....	29
圖 3.15 第一階段法則範例.....	30

圖 3.16 第一階段法則系統活動流程圖.....	31
圖 3.17 第二階段法則範例.....	32
圖 3.18 第二階段法則系統活動流程圖.....	33
圖 3.19 第三階段法則範例.....	34
圖 3.20 第三階段法則系統活動流程圖.....	35
圖 3.21 第四階段法則範例.....	36
圖 3.22 第四階段法則系統活動流程圖.....	37
圖 3.23 第五階段法則範例.....	38
圖 3.24 第五階段法則系統活動流程圖.....	39
圖 3.25 第六階段法則範例.....	40
圖 3.26 第六階段法則系統活動流程圖.....	41
圖 3.27 第七階段法則範例.....	42
圖 3.28 第七階段法則系統活動流程圖.....	43
圖 3.29 於 CLIPS 中確定所建立之 template 和 rule 無錯誤.....	44
圖 4.1 SkyMars Professional 軟體介面	46
圖 4.2 上銀六軸關節式機器手臂-RA605.....	47
圖 4.3 上銀六軸關節式機器手臂離線模擬軟體.....	47
圖 4.4 透過集線器(HUB)將設備端與 PC 端串聯之實際圖.....	48
圖 4.5 本研究整合介面圖.....	50
圖 4.6 專家系統 CLIPS 嵌入 C#專案之示意圖	51
圖 4.7 引用 CLIPSNET.dll.....	52
圖 4.8 載入 researchCLIPS.CLP 檔之方式.....	52
圖 4.9 情境模擬圖一.....	53
圖 4.10 情境模擬圖二.....	54
圖 4.11 情境模擬圖三.....	54
圖 4.12 情境模擬圖四.....	55
圖 4.13 情境模擬圖五.....	56
圖 4.14 情境模擬圖六.....	56

圖 4.15 情境模擬圖七.....	57
圖 4.16 情境模擬圖八.....	57
圖 4.17 情境模擬圖九.....	58
圖 4.18 情境模擬圖十.....	58
圖 4.19 情境模擬圖十一.....	59
圖 4.20 情境模擬圖十二.....	59
圖 4.21 情境模擬圖十三.....	60
圖 4.22 情境模擬圖十四.....	60
圖 4.23 情境模擬圖十五.....	61

第一章 緒論

1.1 研究背景與動機

製造業過去經歷了三次工業革命，從最初機器取代人力一直到自動化設備的使用都使得生產力逐漸提升。如今，製造業已經來到了第四次工業革命(Industry 4.0)，主要是以虛實系統(Cyber Physical System, CPS)、物聯網(Internet of Things, IOT)以及大數據(Big Data)為基礎，在很多領域上都有著工業 4.0 的運用，其中「智慧工廠」為工業 4.0 當中重要範疇之一，智慧工廠中廣泛運用通信技術來串聯實體設備並使工廠管理者能夠線上監控工廠內的現況，且讓設備與設備之間能夠智慧地溝通協調，達到所謂智慧製造(Intelligent Manufacturing)。

在台灣，製造業是推動經濟成長的關鍵，其中金屬工業又為台灣製造業重要產業之一，但由於台灣現今的製造業面臨了人口老化以及勞動力不足等問題，為了能夠即時的掌握與控制生產資訊，許多工廠除了使用自動化設備也開始導入虛實系統，走向智慧製造。在金屬工業當中，主要使用到的生產設備為工具機，目前大多數使用的工具機為電腦數值控制工具機(CNC)，其他自動化設備如機械手臂(Robot)、無人搬運車(Automated Guided Vehicle, AGV)與感測器(Sensor)等也逐漸的導入，而由於越來越多自動化設備的應用，更多的問題將聚焦在如何將不同設備進行整合控制，並且讓設備與現場資源能互相溝通協調，以提高製造生產彈性、自主性等更智慧化的特性。

本研究將運用本體論的技術，收集製造環境中的相關概念，並建立一以知識法則為主之專家系統，利用製造環境中所發生的事實進行推論，產生出新的解決方案。

1.2 研究問題與目的

現今，有許多的機械加工業者已慢慢導入許多自動化設備與技術，並漸漸開始導入虛實系統，然而目前許多的監控技術大多仍處於監視並未達到可直接控制，在面對機台異常或是產線需要增減設備時，仍需仰賴人工去做異常排除、恢復的動作，甚至需要重新撰寫程式。在實際的工廠內，可能會遭遇到的問題有許多種，例如設備出現警告或是故障、產線設備需要增加或移除、網路連線異常或斷線以及工具機內的機械元件需要更換或維護等許多狀況。

在本研究當中，因目前研究限制，因此只針對兩項議題做探討，如下：

1. 當設備出現警告或故障時，如何透過法則推論的方式去趨動其他設備協助。
2. 當產線中有設備需要增加或是移除時，如何透過法則推論的方式去找出讓產線能自動地符合需求和能輕易地擴展的方法。

1.3 研究流程與步驟

本研究的研究步驟將分成六個階段，首先定義問題與蒐集資料並確定所使用的研究方法工具，接著為本體論的建置與專家系統之法則發展，初步驗證法則無誤後，最後建立一情境模擬來驗證，如圖 1.1 所示。

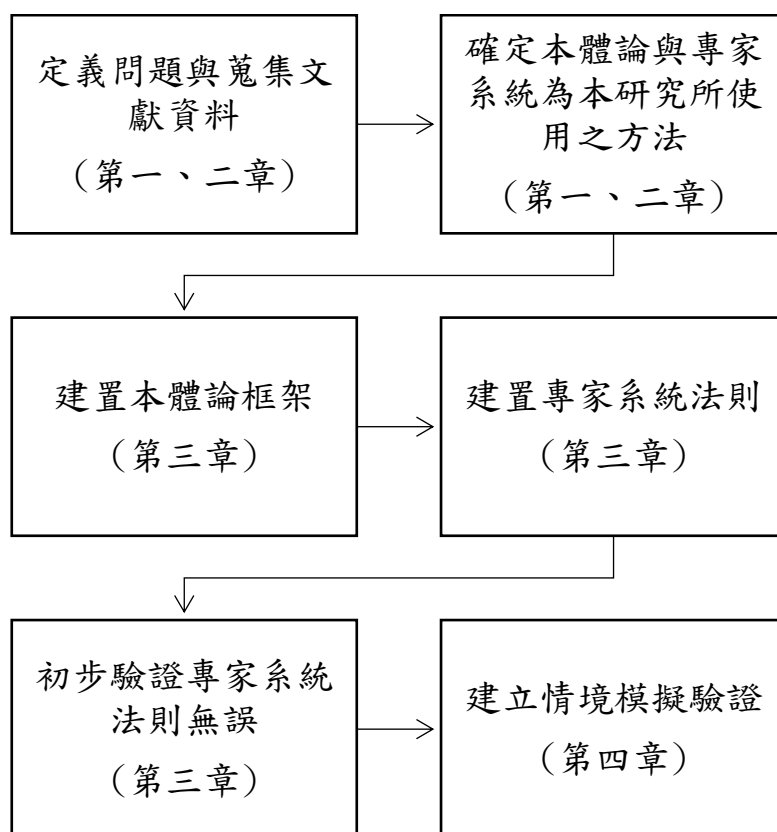


圖 1.1 研究方法與步驟

1.4 論文架構

本論文的架構如圖 1.2 所示，共分成五個章節進行討論。第一章為研究背景動機、研究問題與目的等介紹。第二章文獻探討將蒐集針對製造環境之本體論資料，並介紹本研究所使用的方法與工具。第三章將說明本研究的系統架構層以及建模與推論的流程。第四章則設計一個情境來驗證本研究提出的方法。第五章則為本研究的結論，並提出未來研究方向。

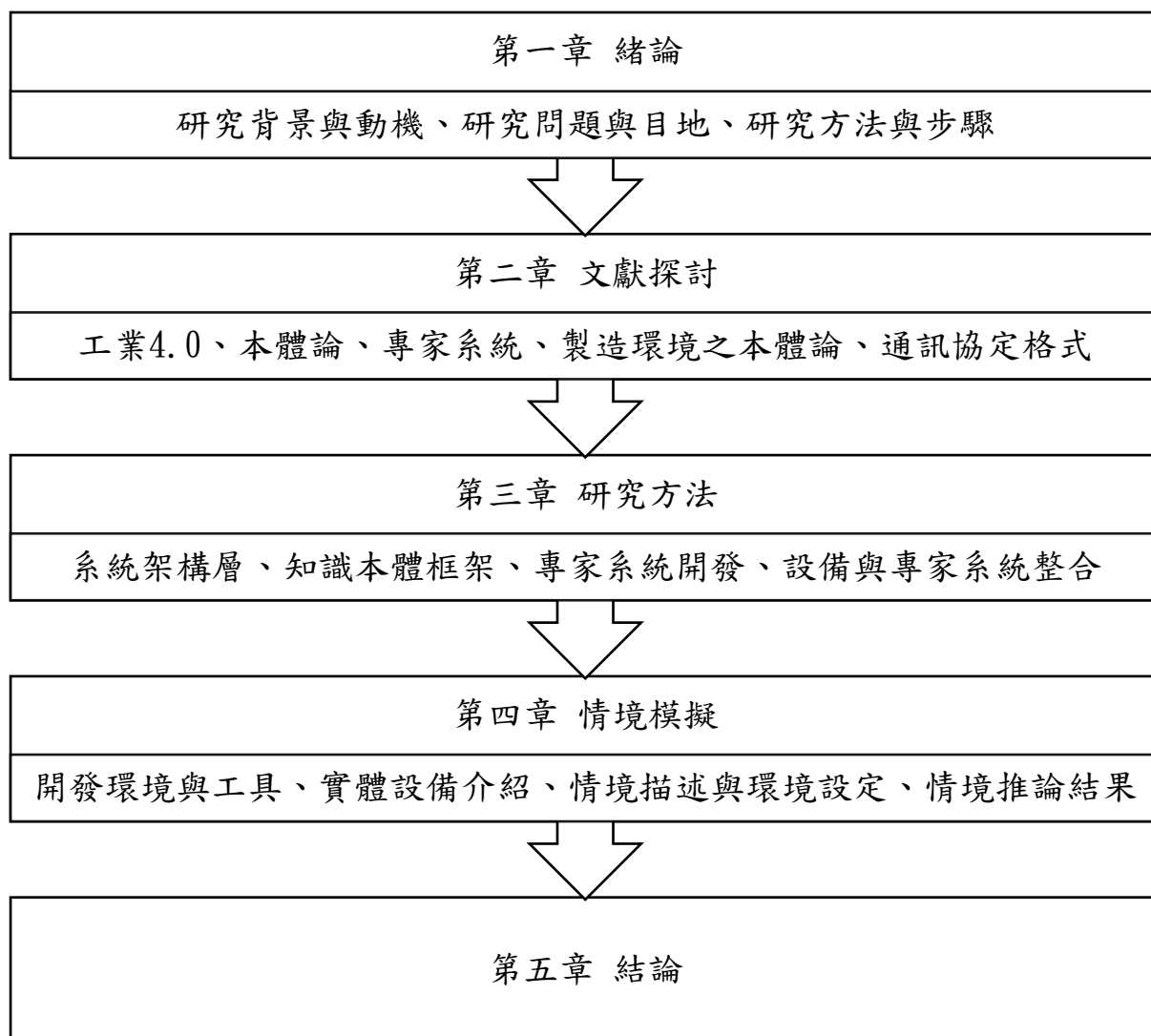


圖 1.2 論文架構圖

第二章 文獻探討

2.1 工業 4.0 與智慧工廠

製造業已經經歷了三次的工業革命，第一次革命是蒸氣機的使用，機器開始取代人力；第二次是「電」的使用，實現了大量製造；第三次則是開始自動化的設備使用，一次次的工業革命都大幅提升了工廠的生產力。而目前製造業已經迎來第四次的工業革命(Industry 4.0)，其中智慧工廠則是工業 4.0 中非常重要的應用，Think Act 雜誌在 2014 年說明了工業 4.0 智慧工廠的運作模式如圖 2.1，從此概念圖當中我們可以分出軟體與硬體的部分，軟體部分包含雲端計算、大數據及網路安全等議題，硬體部分則由感測器、先進製造系統、3D 列印、機械手臂等所組成，並從此圖中可以看出供應商和顧客之間的關係。智慧工廠中主要由智慧型的設備所組成，這些生產設備具有可互通的通訊格式亦或是可聯網，在同個環境下互相協調，管理者可透過所開發的監控系統即時掌握工廠內的生產狀況，達到所謂「虛實合一」的智慧系統。

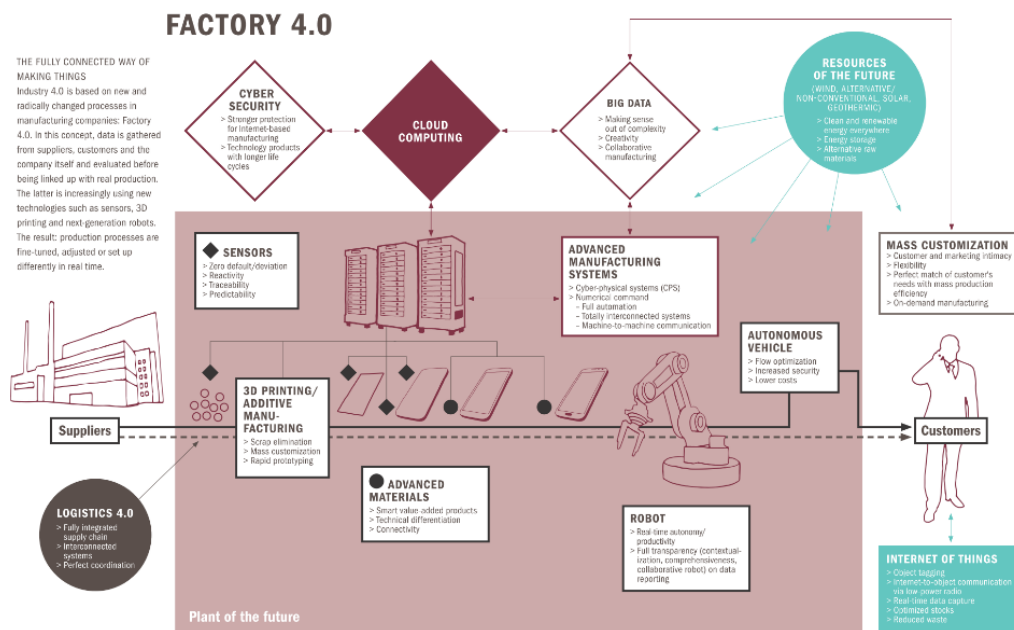


圖 2.1 工業 4.0 智慧工廠概念圖

資料來源：Think Act, 2014

2.2 本體論

2.2.1 本體論介紹

本體論(Ontology)起源自哲學，Bunge 提出本體論為「關於真實世界的基本特性哲學理論」(Bunge, 1977)，再導入至電腦科學領域當中，因其關係到對於真實世界的理解、認知、結構與表達，對於知識領域的定義和相關知識的取得等有較大的優勢。本體論主要以樹狀結構及關聯方式來分類與描述特定領域內，事件之間的關係，並用結構化的方式表達。Gruber 在早期將本體論定義為「對概念的詳細外顯描述」(Gruber, 1993, 1995)，藉由對概念的共同描述，讓不同專家領域的人能夠互相了解並擁有相同的認知，也能讓機器了解人類要表達的語意。

本體為建置特定領域的概念，統一詮釋對事物的認知，Gruber 指出，本體論具有以下四點特性(Gruber, 1993)：

(1) 概念化(Conceptualization)：

將真實世界予以概念化，以提供結構性知識的描述，讓相關成員相互溝通。

(2) 正規化(Formal)：

以清楚的語意定義及數學符號所形成的語言進行描述，可讓人類閱讀。

(3) 明確(Explicit)：

所使用的概念及使用這些概念的限制都有明確的定義，可供機器解讀。

(4) 可分享(Share)：

知識本體表達的是共同認可的知識，反映領域中公認的概念，因此可作為溝通的媒介。

透過以上四點特性清楚地描述與定義不同的知識，讓不同領域的專家能清楚的瞭解，以達到知識分享的目的。

本體論的主要組成要素為：Classes、Slots、Facets、Instances (Noy et al., 2001)：「Classes」指的是類別，描述某個領域知識的概念；「Slots」亦可稱

為 Properties，描述類別與類別之間的關聯屬性；「Facets」是 Slots 的限制；而每一個「Instances」則為 Classes 的一個實例，會繼承類別的屬性與關聯。結合系統論的 Ontology 則是由實體(Entities)、屬性(Attributes)及關聯(Relations)所組成(Uschold & Gruninger, 1996)：「實體」指特定領域中重要的事物；「屬性」敘述概念的特性及可能的範圍；「關聯」則說明實體或實體與概念之間的關係。

Ontology 可依據不同的性質和用途進行分類，其中「主題內涵」和「用途」是常見的分類標準(Guarino, 1997)。

依照主題內涵，可分為下列三種：

- (1) 術語本體論(Terminological ontology)：像詞彙一樣表達專業領域中的知識。
- (2) 資訊本體論(Information ontology)：用以詳細記錄資料庫。
- (3) 知識塑模本體論(Knowledge modeling ontology)：主要用在知識的塑模，建立完整的知識概念並表達。

依照用途，則可分為下列四種：

- (1) 上層本體論(Top-level ontology)：描述一般概念，如：空間、物件、時間、事件等，而這些一般化概念與特定領域無關。
- (2) 領域本體論(Domain ontology)：描述相關領域的字彙。
- (3) 作業本體論(Task ontology)：描述相關作業或活動的字彙。
- (4) 應用本體論(Application ontology)：描述和特定領域或是作業相關的概念，而這些概念通常相當於領域實體(Domain entity)在執行活動時的角色。

2.2.2 本體論工具

在建立本體論的過程中，不同專家的領域會影響實例的性質、概念等，使得發展出來的本體論不盡相同。因此，在建立本體論時，我們必須判斷知識本體框架應該屬於哪一種專家領域。常用的本體論發展工具如下：

- Protégé
- Webonto
- WebODE
- OntoEdit
- OILED
- Ontosaurus
- Ontolingua

在 2013 年，Shen 等人針對常用的本體論發展工具進行功能分析如表 2.1 (Shen et al., 2013)。

表 2.1 常用本體論發展工具功能分析

Name Function	PROTÉGÉ	Webonto	OntoEdit	WebODE	OILED	Ontosaurus	Ontolingua
OWL	√			√	√		
Visualization	√	√		√			
Ontology merging	√		√	√			√
Chinese support	√						
Network technology		√	√	√		√	√
Fuzzy ontology							
Cooperative development		√	√	√		√	√

資料來源：Shen et al., 2013

常用本體論編輯器其中之一的發展工具 Protégé 是由 Stanford University 所開發，是一個免費的開源平台，為不斷增加的本體論使用者提供一整套用本體構建領域模型和基於知識應用的工具。在 Protégé 中可以使用 Resource Description Framework (RDF)、Resource Description Framework Schema (RDFS)、Web Ontology Language (OWL)來編輯及修改本體的部分；

並納入 Open Knowledge Base Connectivity (OKBC) 模組，提供圖形化與框架式的知識表達方式和開發推論引擎的環境，讓使用者能自行建構本體論。

Protégé 本身除了上述特點之外，還有以下五種特點：

1. 圖形化介面：易於學習與建立本體框架邏輯。
2. 支援多種儲存格式及資料型態。
3. 提供 API，讓程式撰寫員可以依照自己的需求撰寫程式碼。
4. 提供開放源碼，支援 Plug-in 開發。
5. 免費開放源軟體。

由於 Protégé 的資源較開放完整易於開發，且圖形化的介面較易使用來理解並建立本體論，因此在本研究當中，將會先利用 Protégé 來建立與本研究相關之本體論框架，後續再利用專家系統建置知識法則。

2.3 專家系統

2.3.1 專家系統介紹

在 1980 年代學者們紛紛開始研究人工智慧技術，有非常多的專家系統開始被開發出來，而專家系統是早期人工智慧重要的一個分支，目的在模擬人腦，將專家知識和經驗建構在電腦上，讓電腦系統具備智慧推論能力，一般採用「知識表示法」和「知識推論」技術來對特定領域進行推論與解答。

在 2009 年，鄭弼仁整理列出專家系統之發展史，其發展史可分為五個時期如表 2.2：

表 2.2 專家系統發展史

	時間	過程
孕育期	1960 前	1937 年，Turing 提出理想計算機概念。 1956 年，「人工智慧」一詞首度提出。 1957 年，Newell 等人提出通用問題解決法。 1958 年，MaCarthy 提出 LISP 語言。
誕生期	1960~1970	1965 年，史丹佛大學開發第一個專家系統 Dendral，是一套能推論化學結構的專家系統。 1965 年，Robinson 提出 Resolution Refutation 理論，使符號定理證明的技術向前邁進。 1965 年，Zadeh 提出 Fuzzy Logic 理論，使數值或符號的處理具有模糊化之特性。 1969 年，麻省理工學院開發 MACSYMA(數學符號運算專家系統)。
開創期	1970~1980	1972 年，史丹佛大學開發 MYCIN(血液感染疾病診斷專家系統)。

	時間	過程
開創期	1970~1980	<p>1972 年，史丹佛研究院(SRI)開發 Prospector(礦床探測專家系統)。</p> <p>1972 年，法國 Colmerauer 與 Roussel 開發 PROLOG 邏輯程式語言。</p> <p>1974 年，卡內基-美隆大學開發 OPS 語言(人工智慧與專家系統專用語言)。</p> <p>1975 年，Minsky 提出框架式(Frame-base)知識標示法。</p> <p>1977 年，Feigenbaum 提出知識工程(Knowledge Engineering)一詞。</p> <p>1978 年，卡內基-美隆大學和 DEC 公司合作開發 R1(又稱 XCON)，可依據客戶需求，架構適當的工作站電腦系統。</p> <p>1979 年，Forgy 提出 Rete 演算法，提高專家系統的法則匹配速度。</p>
成長期	1980~1990	<p>1980 年，具有 LISP 語言之機器開始生產。</p> <p>1982 年，日本政府宣布開發第五代電腦，以 PROLOG 作為核心語言。</p> <p>1983 年，IntelliCorp 公司推出 KEE，其他工具也接著問世。如：ART、Knowledge Craft。</p> <p>1984 年，歐洲共同市場訂定歐洲資訊技術研究發展策略計劃。</p> <p>1988 年，Gallant 提出以類神經網路(Neural Network, NN)為基礎之專家系統架構。</p> <p>1989 年，日本宣布人類領域科學計畫(第六代電腦計畫)，希望藉著類神經網路突破人工智慧瓶頸。</p>
成熟期	1990 後	<p>進入商業競爭時代，專家系統在各行業廣泛應用。</p>

資料來源：鄭弼仁，民 98

典型的專家系統包含了知識庫(Knowledge base)、推論引擎(Inference engine)、工作記憶體(Working memory)、解釋功能(Explanation facility)、知識擷取功能(Knowledge acquisition facility)以及使用者介面(User interface)這六個要素，其結構圖如圖 2.2 所示(Giarratano & Riley, 1998)。

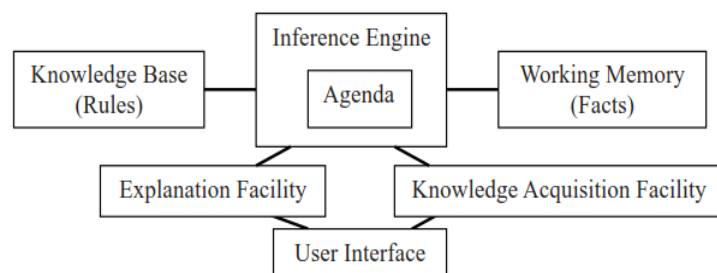


圖 2.2 典型專家系統結構圖

資料來源：Giarratano & Riley, 1998

專家系統解決問題主要是以知識庫內儲存的知識推論來獲得結果，而知識庫主要從該領域的專家獲取相關知識和方法，而目前推論系統所使用的知識表示法包含語意網路、框架、邏輯、法則和語意式，每一種表示法皆有其優點。簡單來說，在專家系統裡主要分為知識庫和推論引擎兩部分，知識庫負責記錄事實來提供推論的基礎，而事實會驅動法則，讓推論引擎進行推論，其運作概念如圖 2.3 所示(Giarratano & Riley, 1998)。

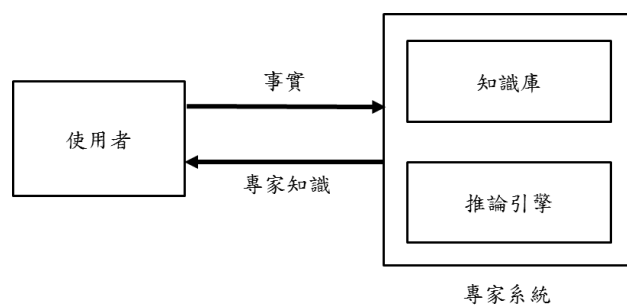


圖 2.3 專家系統運作基本概念

資料來源：Giarratano & Riley, 1998

2.3.2 專家系統知識表現

在 2.3.1 中提到專家系統是依據其知識庫來進行推論，而專家系統如何建構以及理解，必須依據知識的表示方式。而常用的知識表示法有五種，分別為：法則式、邏輯式、框架式、物件導向式以及語意網路式，以下將分別進行說明。

➤ 法則式知識表示法：

法則主要是「IF-THEN」的邏輯，其形式可分為「IF 前提 THEN 結論」和「IF 狀態 THEN 動作」，這兩者的意義為若前項的狀態或條件成立，則執行後項的動作。此種知識表示法的好處在於容易擴充與修改。

➤ 邏輯式知識表示法：

邏輯表示法於專家系統是屬於最嚴謹的方式，可定義為研究推論法則的科學，目的是從敘述的真偽來推論其他敘述的真偽。邏輯又可分為 Propositional Logic(命題邏輯)、Predicate Logic(術詞邏輯)、Fuzzy Logic(模糊邏輯)。

➤ 框架式知識表示法：

在 2.3.1 專家系統發展史中提到，框架式結構是 Minsky 在 1975 年首度提出用來表示固定形式思考模型的資料結構(Minsky, 1975)。基本原理是讓框架成為資料與程序結構化的架構，每一個框架組成：由一個框架名、一組槽及槽值所組成，這些槽可以是任何型態或值，每個槽包含 slot name、facet 和 value。框架常被組織成階層式架構，具備繼承的特性，而框架則屬於靜態的資料結構，缺乏靈活性。

➤ 物件導向式知識表示法：

此知識表示法由物件導向程式演化而來，表示法與框架式類似但仍有所不同，物件導向表示法將外界真實事件抽象成物件，使電腦與人的思考能夠更緊密，透過物件能使程式變得模組化並具有擴充性與易維護之特性。

➤ 語意網路式知識表示法：

早期生理學家為了研究人類記憶的模式，提出了語意網路(Semantic Networks)的觀念。其包含了「節點(node)」和「弧(arc)」，節點表示知識的組成元素或種類，也就是一般的理念、指定物件或是事件等，而弧則用來描述節點與節點之間的關係。語意網路圖形化的表現方式讓人容易理解，並且可以任意增減節點與弧，因此非常彈性化，但在處理較複雜的事情上，結構化稍顯不足也不易管理。圖 2.4 為語意網路表示法之簡易範例圖形。

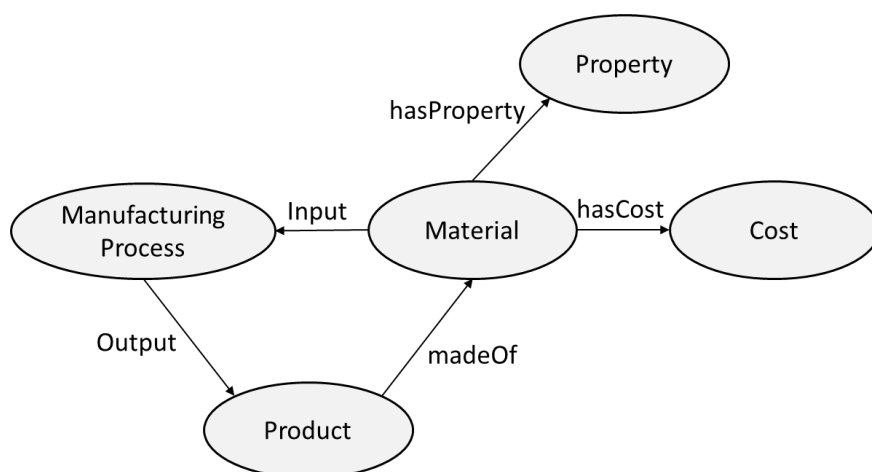


圖 2.4 語意網路知識表示法簡易範例

資料來源：本研究整理

2.3.3 法則式推論專家系統-CLIPS

本研究在推論的過程中會利用實體的狀態作為事實來驅動法則，因此所使用的專家系統為 CLIPS(C Language Intergrated Production System)，CLIPS 是一套以 C 語言寫成的專家系統開發軟體，在 1984 年由美國國家航空暨太空總署(National Aeronautics and Space Administration, NASA)的 Johnson Space Center 發展出來。CLIPS 提供了 COOL(CLIPS Object Oriented Language)物件導向程式語言，且具備程序性的設計方式及法則式知識表示法。

CLIPS 有三種呈現方式：

- (1) Rules 法則
- (2) Deffunctions and generic function 自定義及通用函數
- (3) Object-oriented programming 物件導向編程

在 CLIPS 裡，對於事件的定義有非常多種，此處僅針對本研究所使用到之定義作說明。本研究使用了 deftemplate 以及 defrule 作為建置我專家系統內知識法則的部分，分別說明如下：

- Deftemplate 可以為事實建立樣板，建立 deftemplate 時可以設定多個 slot，每一個 slot 各自代表不同的屬性。見圖 2.5 範例所示，以 deftemplate 建立 order(訂單)模板，建立相同類型但不同屬性的 Order，如編號 1 跟 2 有不同的產品別與數量，但都同樣屬於訂單資訊。使用 deftemplate 產生 fact 的時候，要設定其 slot 值，確保同一類型的事實也具備相同屬性，在法則的設定與判斷上會比較有利。

```
(deftemplate Order(slot Number)(slot Product)(slot Quantity))  
(Order (Number 1)(Product A)(Quantity 30))  
(Order (Number 2)(Product B)(Quantity 50))
```

圖 2.5 Deftemplate 範例說明圖

資料來源：本研究整理

- Defrule 用來定義法則，如前面所述，CLIPS 的事實會觸發法則來產生新的事實。在 defrule 中，「=>」符號前是觸發法則的條件，稱為 LHS(Left Hand Side)；而符號之後是法則被觸發之後要進行的動作，稱為 RHS(Right Hand Side)，此撰寫邏輯與「IF-THEN」所表示的相同。而為了避免定義完法則後，同時觸發了多條法則而引發錯誤，因此在撰寫法則時，可以使用 declare 來定義各個法則的優先次序，數字越大者優先順序越高。Defrule 簡易範例如圖 2.6 所示，當待加工原料之暫存架的狀態若為 Yes，表示架上存有待加工原料，並且當工具機狀態為閒置時，就會觸發機器手臂必須啟動的新事實。

```
(defrule Robot-transport-Material
(QueMaterialDesk Yes)
(CNC Idle)
=>
(assert (Robot go)))
```

圖 2.6 Defrule 範例說明圖

資料來源：本研究整理

CLIPS 亦可與 C++、JAVA 以及 C#等語言進行整合應用，本研究即使用 C#作為整合介面，但在進行整合的過程中不易發覺 CLIPS 內部設計是否有誤，因此在建置完法則之後，可透過 CLIPS 軟體進行 Offline 檢測，檢查所設計之模板以及法則和法則之間是否有相互抵觸或是不正確的撰寫。

2.4 製造環境的本體理論

建造一個智慧工廠是相當複雜且有非常多細節的過程，在這過程當中如何有效且更有系統的規劃，使設備達到更自主智慧的運用，一直以來都是製造業所關注並設法解決的課題。

2006 年時，Lemaignan et al.從實體(Entity)、作業(Operation)和資源(Resource)這三大類別角度提出了製造環境的本體論如圖 2.7 所示(Lemaignan et al., 2006)。

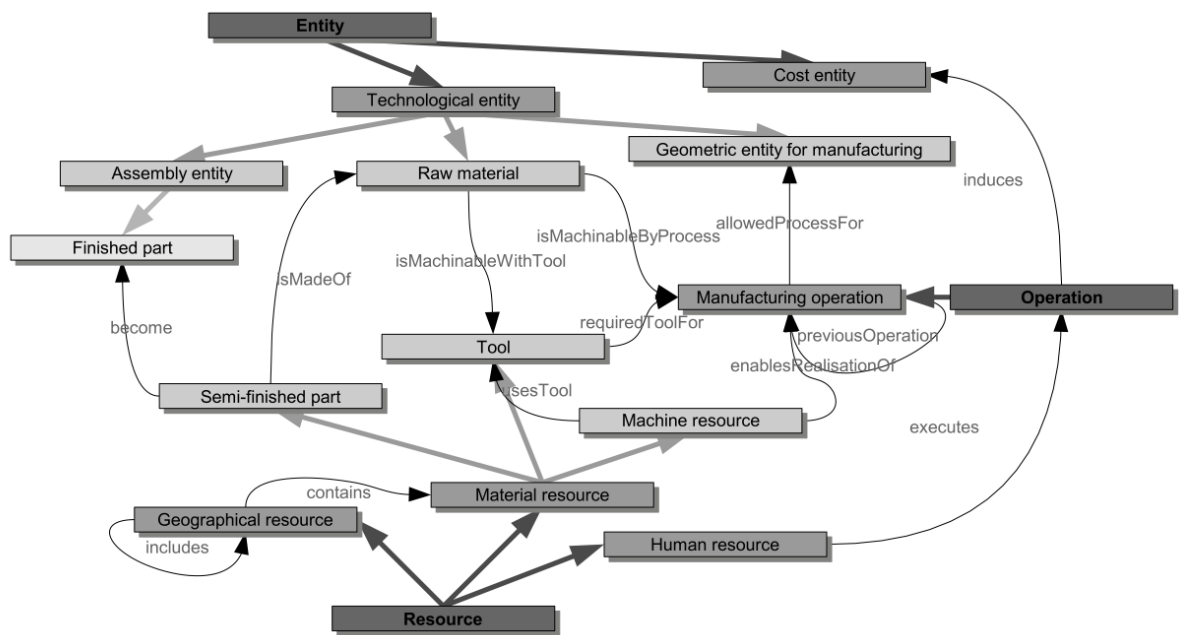


圖 2.7 製造環境本體主要類別和物件屬性

資料來源：Lemaignan et al., 2006

到了 2010 年，Alsafi 和 Vyatkin 則以物料資源(Material_resource)、作業(Operation)和控制器(Controller)來描述如圖 2.8 所示(Alsafi & Vyatkin, 2010)。

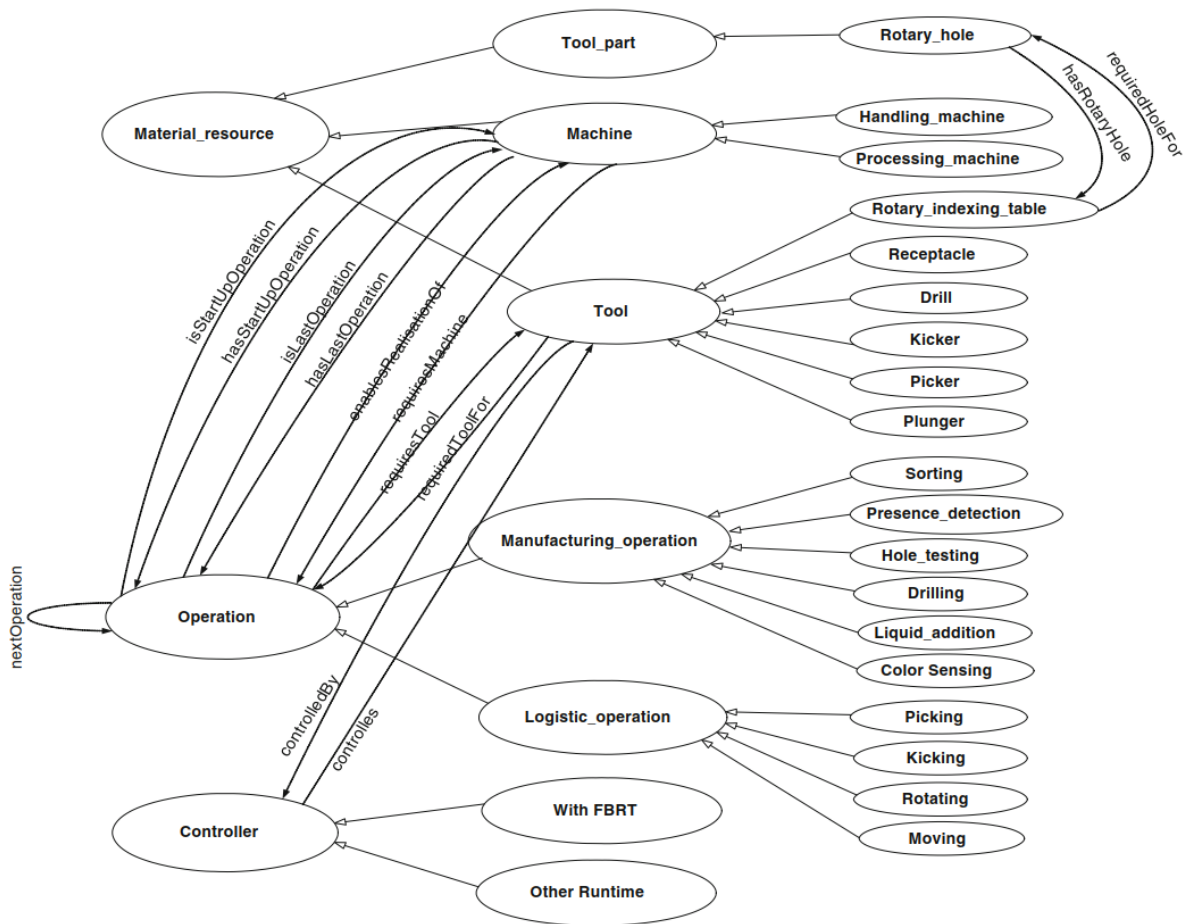


圖 2.8 製造環境本體論知識模型 (Alsafi & Vyatkin, 2010)

資料來源：Alsafi & Vyatkin, 2010

然而在製造領域中，要使工廠能夠更自主化，必須具備更智慧化的特性，因此，在 2016 年，Negri et al.指出在製造領域的語意語言中，有四個主要的要求如下(Negri et al., 2016)：

1. 必須支持概念化建模和數據儲存。
2. 必須支持模型易於使用和維護。
3. 必須支持相互操作性。
4. 必須支持自動推理。

本研究將參考過去所被定義的製造環境本體論，來定義本研究相關類別、物件屬性等本體框架。

2.5 通訊協定格式

在自動化製程中，有非常多種通訊協定格式如 TCP/IP、CC-Link、Modbus 等，而 TCP/IP (Transmission Control Protocol/Internet Protocol) 為非常廣泛使用的通訊協定格式，是一種網際網路通訊模型，其架構層次可分為四層，其說明如圖 2.9 所示。

架構層	說明
應用層 Application Layer	應用程式間的溝通協定，例如檔案傳輸。
主機傳輸層 Transport Layer	提供端點間的資料傳送服務並確定資料已被送達和接收。
網路層 Internet Layer	負責基本的封包傳輸功能，讓每個封包都能到達目的端。
網路存取層 Network Access Layer	定義如何用實際網路來傳送資料。

圖 2.9 TCP/IP 架構層

資料來源：本研究整理

工業自動化的設備種類非常多，且每個設備商都有自己的開發方式，不同的設備甚至提供不同的 API，因此設備在實際工廠內的整合相對困難，設備維護也相對不容易。為了讓設備有統一的整合協定，產業界紛紛推廣工業乙太網路(Industrial Ethernet)的通訊標準，讓設備與電腦之間的溝通化繁為簡，工業乙太網路運用乙太網路(Ethernet)作為標準，取用 TCP/IP 通訊協定，使設備得以互聯。

2008 年，Da'na 等學者運用 TCP/IP 技術在 PLC 的監測控制上，並說明 TCP / IP 在工廠控制設計中提供了所需的靈活性和可擴展性 (Da'na et al., 2008)。而在 2017 年，Bohuslava 等學者介紹了通過各種通訊協定來進行系統控制，並使用標準乙太網來設計基於工業 4.0 的機器人單元控制，主要也是利用 TCP/IP 來設計通訊(Bohuslava et al., 2017)。由此可知，TCP/IP 之通訊協定技術的應用相當具有整合性，本研究實體層設備即是透過 TCP/IP 進行資料擷取。

第三章 研究方法

3.1 系統架構層

本研究的系統架構層由實體層、虛擬層與知識推論層所組成，簡易系統架構層如圖 3.1。實體層部分包含工具機(CNC)、機械手臂(Robot)、無人搬運車(AGV)以及訂單資訊(Order Information)等實體方面的設備與資料；虛擬層作為實體層與知識推論層之間的橋梁，主要會截取並整合實體層的數據來讓知識推論層能進行法則推論；而知識推論層則是確定事實(Facts)、類別(class)、屬性(Property)和知識法則(Knowledge Rules)的核心。

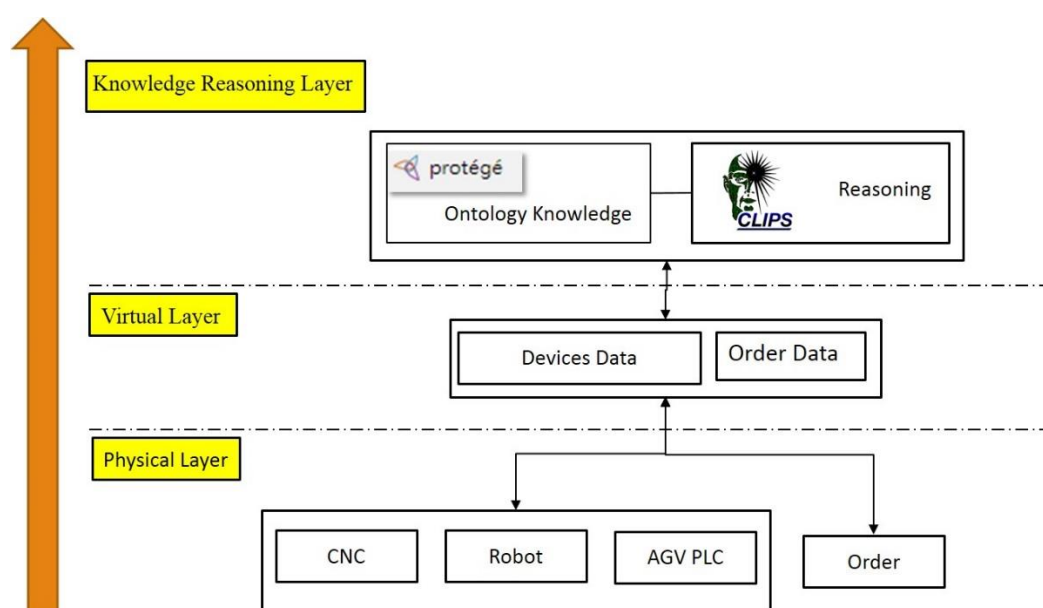


圖 3.1 簡易系統架構層

在本研究系統架構層中，較詳細的說明如圖 3.2 所示。首先，在實體層當中，工具機(CNC)主要區分成粗銑削(Rough Milling)、精銑削(Finish Milling)與雕刻機(Engraving)三種功能類型，而工具機的資料是透過台灣精密機械研究發展中心(PMC)所開發的 SkyMars 軟體來截取；機械手臂(Robot)將運用台灣上銀科技所開發的六軸關節式機械手臂 RA605；無人搬運車(AGV)的部分則因為研究限制，因此以 PLC 作為模擬 AGV 動作之工具，本研究所用的 PLC 為三菱所開發的 FX3G-24MR 型號款；訂單則會包含顧客所要的產品種類、數量、交期等資訊。實體層的機械手臂與 PLC 設備將會透過集線器(Hub)，且透過 TCP/IP 的通訊格式傳輸設備資訊至虛擬層進行整合，而工具機的設備資訊因已透過 SkyMars 截取，因此虛擬層可直接

取得 SkyMars 已整合之資訊。虛擬層將使用 C# 建立一 Windows Form 的介面將實體層設備資訊整合呈現，並且連結專家系統。知識推論層中，本研究先以 Protégé 建立本體論框架，接著依循著本體論框架，使用專家系統 CLIPS 來建置知識法則，透過虛擬層所截取的實際事實，觸發法則後產生新事實以解決本研究問題。

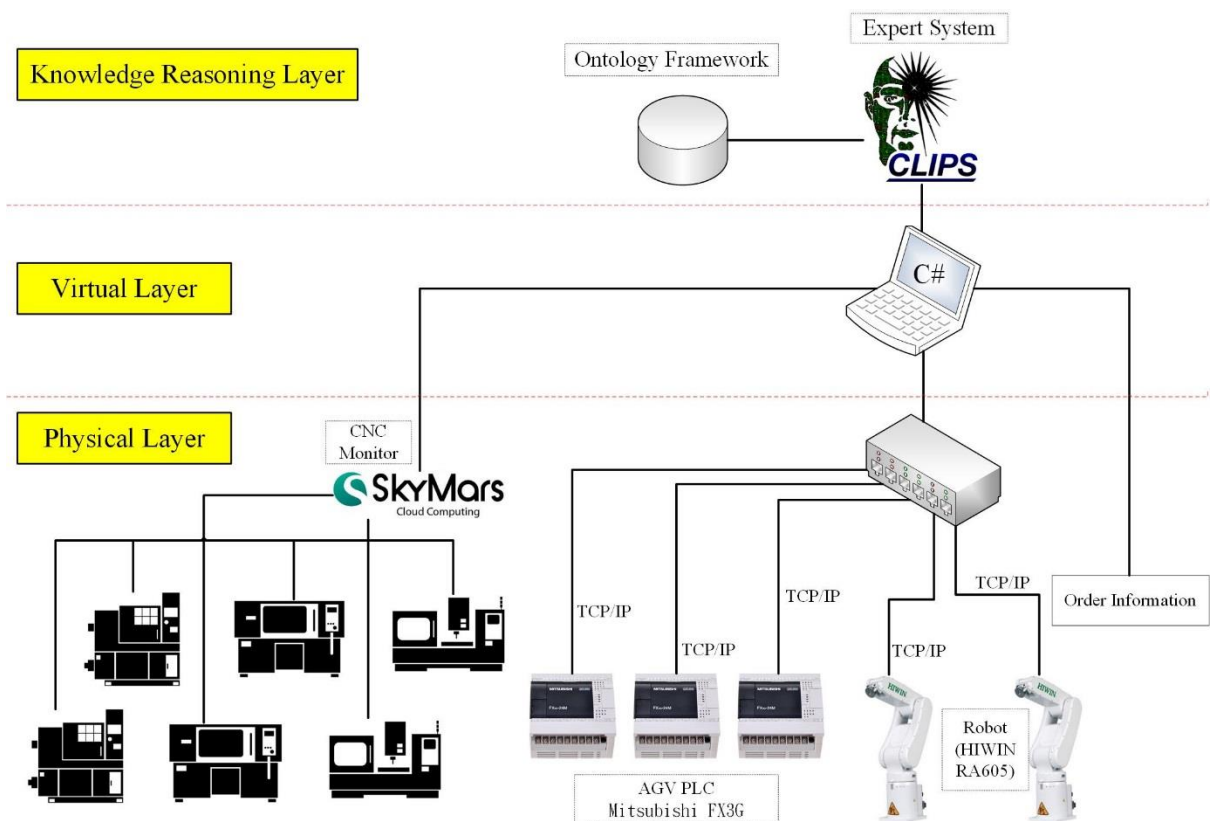


圖 3.2 詳細系統架構圖

建立完系統架構層後，接下來本研究將介紹知識本體框架與專家系統內法則的建置。

3.2 建置製造環境的知識本體框架

過去在定義製造環境知識本體的文獻中，大多主要以作業(Operation)和資源(Resource)為主再進行擴展，但若要使知識層能達到控制甚至規劃的層面，則必須將知識本體定義的更加完善。本研究主要核心在控制層面，在監控架構層中，屬於 Supervision 層，也就是所謂 Supervision Control and Data Acquisition(SCADA)，其功能為擷取工廠生產線之數據，並進行遠端設備或生產線「控制」，監控架構層如圖 3.3 所示。

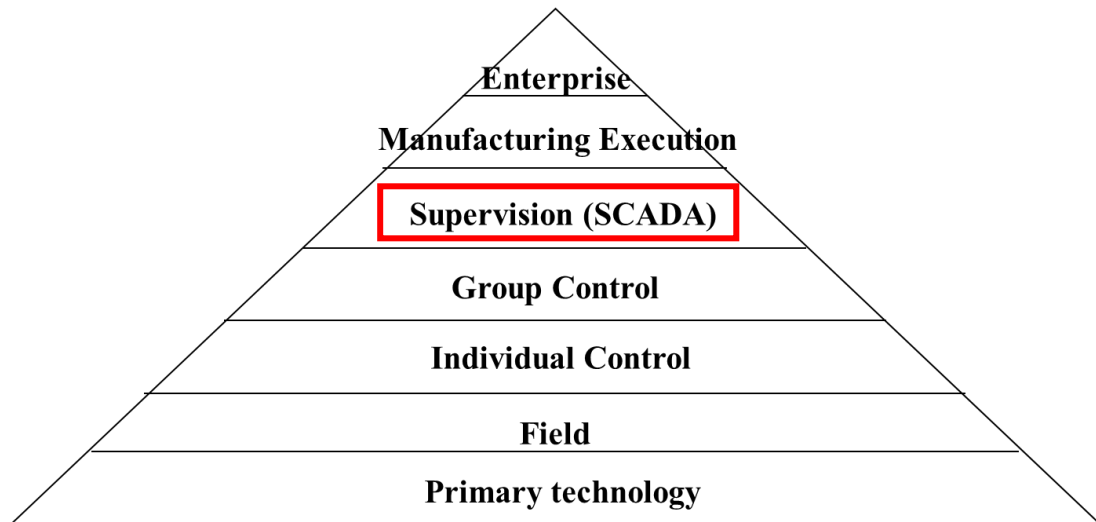


圖 3.3 監控架構層

接著，本研究所建置之知識本體框架中，主要分為七大主類別，分別為 Devices、Process、Status、Order、Material、Time 以及 Error，再分別建立其子類別，如圖 3.4 所示。

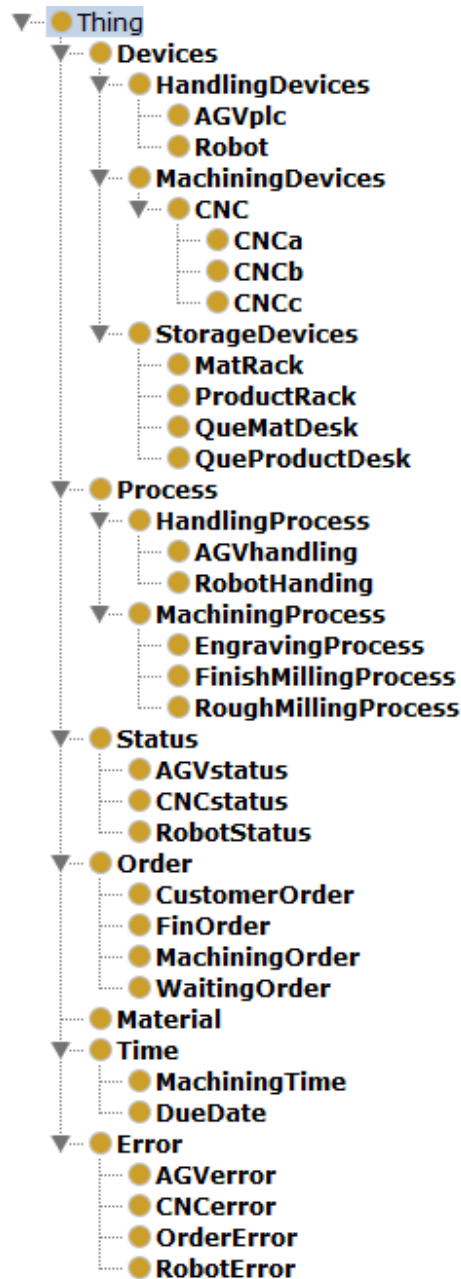


圖 3.4 本研究之製造環境本體知識類別

本研究知識本體框架之主類別描述如下：

- (1) Devices：此類別指在製造環境中的設備，其中搬運設備 (HandlingDevices)、加工設備 (MachiningDevices) 及儲存設備 (StorageDevices) 為 Devices 的子類別。三個子類別下，無人搬運車 (AGVplc) 以及機器手臂 (Robot) 為搬運設備之子類別；工具機 (CNC) 為加工設備之子類別；原料架 (MatRack)、待加工品料架 (QueMatDesk)、完成品待運送區 (QueProductDesk) 以及成品架 (ProductRack) 皆為儲存設備之子類別。

- (2) Process：此類別主要描述設備搬運以及加工之過程，分為搬運過程(HandlingProcess)以及加工(MachiningProcess)，搬運過程之子類別分別為無人搬運車搬運(AGVhandling)及機器手臂搬運(RobotHandling)。此類別的建立可供往後在製造環境的規劃中運用。
- (3) Status：此類別指在製造環境中設備的狀態，分為無人搬運車狀態(AGVstatus)、工具機狀態(CNCstatus)以及機器手臂狀態(RobotStatus)。如設備是否開機、待機或者是故障狀態。
- (4) Order：此類別指訂單資訊，訂單從接收後會歷經等待、進入加工一直到訂單處理完成，因此訂單分別有顧客訂單(CustomerOrder)、等待中訂單(WaitingOrder)、加工中訂單(MachiningOrder)以及完成之訂單(FinOrder)四個子類別。
- (5) Material：此類別指在製造環境中所使用到的原料。
- (6) Time：時間在製造環境中也是重要元素之一，其子類別有交期(DueDate)以及加工時間(MachiningTime)。
- (7) Error：此類別描述製造環境中設備的錯誤、故障等狀態，其子類別有無人搬運車錯誤(AGVerror)、工具機錯誤(CNCerror)、機器手臂錯誤(RobotError)以及訂單錯誤(OrderError)。

定義好類別之後，接著定義物件屬性，本研究建立了九個物件屬性來描述類別之間的關係，物件屬性如圖 3.5。

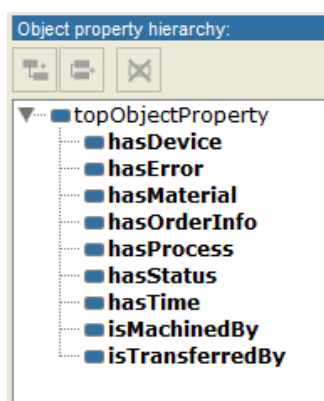


圖 3.5 知識本體類別之物件屬性

物件屬性建立完後，即可產生類別與屬性關聯圖，如圖 3.6 所示。

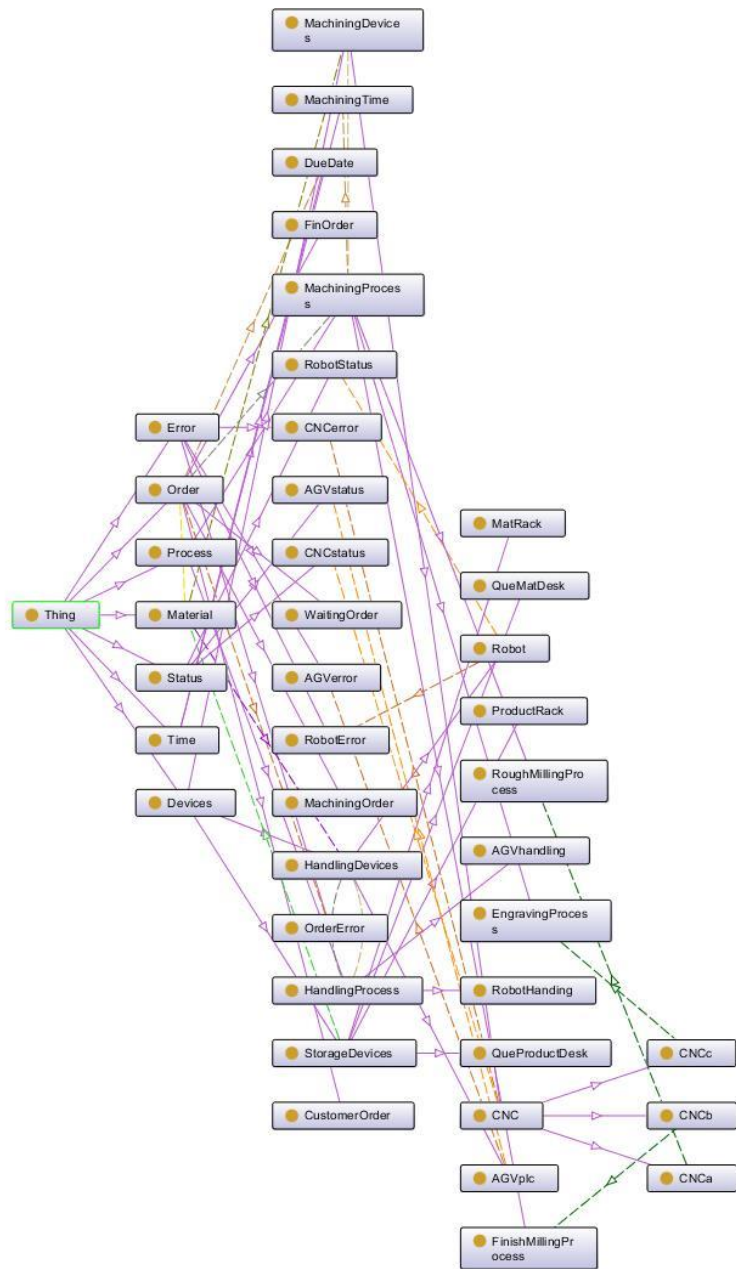


圖 3.6 類別與物件屬性關聯圖

3.3 建置專家系統 CLIPS 法則

在說明本研究所建置之法則之前，將先簡單介紹本研究之情境，本研究以金屬加工業為例，於廠內設有 6 台 CNC、三台 AGV 與兩台可移動式機械手臂(Mobile Robot)。如圖 3.7 與圖 3.8 所示，MatRack 與 ProductRack 分別為原料架與成品架；CNC 分為三種加工型態之工具機，CNCa 為粗銑削、CNCb 為精銑削而 CNCc 為雕刻機台，於本研究中 CNCa 與 CNCb 雖各為不同加工型態，但若有機台若忙碌或異常，也可互相支援其加工型態；各個工具機旁分別皆有原料待加工料架(Que Material Desk)，為圖中橘底的部分(a1、a2、b1、b2、c1、c2)；d1 與 d2 則為完成品待運送區，準備將加工後原料運送至成品架；R1 與 R2 則為可移動式機械手臂(Mobile Robot)；AGV 則有三台分別編號為 AGV1、AGV2、AGV3。

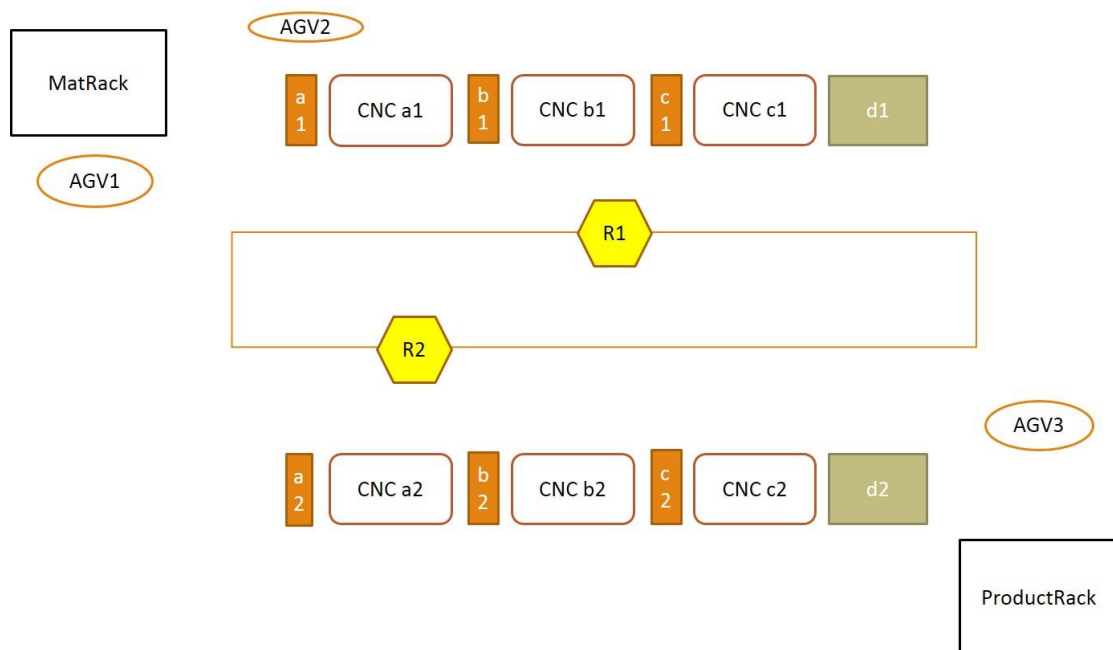


圖 3.7 本研究建立之情境圖






	Mobile Robot
	AGV
	CNC
	Que Material Desk
	Que Product Desk

圖 3.8 情境圖之各項圖示說明

本研究將製程訂為當原料開始加工，會經過粗銑削→精銑削→雕刻之流程，每一道製程間皆需要搬運設備之配合，正常加工流程圖如圖 3.9 所示。本研究強調設備間相互協調，將以此流程圖之七個階段流程作為建立法則之依據，在每一個環節皆有決策點，決定哪台機台做搬運與加工，甚至當機台異動時，如何能驅動別台閒置設備進行幫忙，因此每台設備的狀態資訊相當重要，各個設備狀態分類整理如圖 3.10，實體設備狀態皆以目前設備商所提供的為依據。

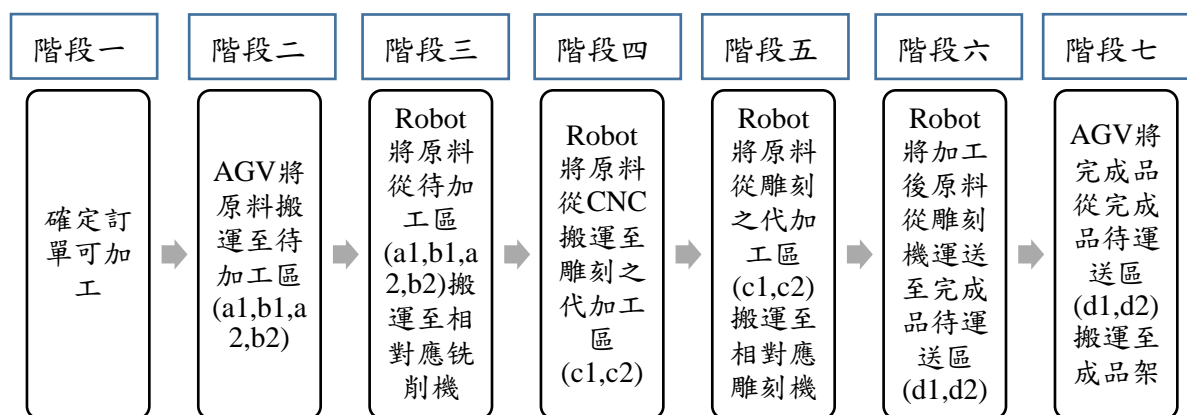


圖 3.9 本研究正常加工之流程圖

設備	狀態
CNC	0: 關機(Off)、1: 運動(Run) 2: 閒置(Idle)、3: 警報異常(Alarm)
Robot	1: 閒置(Idle)、2: 運動(Run) 3: 暫停(Pause)、-1: 失敗(Failed)
AGV	0: 關機(Off)、1: 運動(Run) 2: 閒置(Idle)、3: 警報異常(Alarm)
Que Material Desk Que Product Desk	0: 無原料(Empty)、1: 有原料(Not Empty)

圖 3.10 各設備之狀態資訊

在設計相關法則之前，deftemplate 可以為事實建立一個樣板，在建立 deftemplate 時可以設定多個 slot，代表各個不同屬性，而使用 deftemplate 時必須設定每一個 slot 的值，確保同一類型的事實有相同屬性，以利於後續法則的設計與判定。按照本研究所建立之本體知識與本研究目的，在 CLIPS 中定義了 12 個模板，圖 3.11 至圖 3.14 即為本研究所設計之各個 deftemplate。

```
(deftemplate CustomerOrder(slot OrderNumber)(slot Material)(slot
Quantity)(slot DueDate))
(deftemplate WaitingOrder(slot OrderNumber)(slot Material)(slot
Quantity)(slot DueDate))
(deftemplate MachiningOrder(slot OrderNumber)(slot Material)(slot
Quantity)(slot DueDate))
(deftemplate FinOrder(slot OrderNumber)(slot Material)(slot
Quantity)(slot DueDate))
```

圖 3.11 本研究中與各 Order 相關之 deftemplate 設計

```
(deftemplate MatRack(slot Material)(slot Quantity))
(deftemplate QueMatDesk(slot DeskType) (slot Status))
(deftemplate QueProductDesk(slot DeskType) (slot Status))
(deftemplate ProductRack(slot Material)(slot Quantity))
```

圖 3.12 本研究中與各 Storage Devices 相關之 deftemplate 設計

```
(deftemplate HandlingDeviceAGV(slot DeviceName)(slot Status)
(slot AGVProg))
(deftemplate HandlingDeviceRobot(slot DeviceName)(slot
Status)(slot RobotProg))
```

圖 3.13 本研究中與各 Handling Devices 相關之 deftemplate 設計

```
(deftemplate MachiningDeviceCNCmilling(slot DeviceName)(slot
Type)(slot Status)(slot MainProg))
(deftemplate MachiningDeviceCNCengraving(slot DeviceName)(slot
Status)(slot MainProg))
```

圖 3.14 本研究中與各 Machining Devices 相關之 deftemplate 設計

設計了 deftemplate 之後，defrule 用來定義本研究相關法則，其詳細運作過程如 2.3.3 所述，當 LHS 狀態同時符合時，法則才會被觸發。而本研究在 Mobile Robot、AGV 以及工具機等設備數量不只一台，因此若法則定義相同時，容易造成撞機等危險，因此在撰寫法則時，皆有利用 declare 來設置其優先次序，數字越大者，越優先判定。本研究依據圖 3.9 所述之七個階段流程，建立了法則來解決本研究情境問題，將一一說明如下：

(1) 第一階段法則-判斷原料架上之原料量是否足以進行加工：

如圖 3.15，此法則為訂單進入加工前先判斷原料架上的種類以及數量是否符合訂單所需，若原料充足時，訂單則進入等待，原料架之數量也隨之減少；反之，若不充足則無法進行後續加工，訂單需等待原料補足後，才得以進行。其法則系統活動流程圖如圖 3.16 所示。

```
(defrule MaterialQuantity-is-enough
(declare (salience 1000))
?CustomerOrder <- (CustomerOrder(OrderNumber ?OrderNumber)
(Material ?Material)(Quantity ?OrderQuantity)(DueDate ?DueDate)
(Program ?Program))
?MatRack <- (MatRack(Material ?MatRackMaterial)(Quantity ?MatRackQuantity))
(test (and
(eq ?OrderMaterial ?MatRackMaterial)(< ?OrderQuantity ?MatRackQuantity)))
=>
(retract ?CustomerOrder)
(retract ?MatRack)
(assert (WaitingOrder(OrderNumber ?OrderNumber)(Material ?Material)
(Quantity ?OrderQuantity)(DueDate ?DueDate)(Program ?Program)))
(assert (MatRack(Material ?MatRackMaterial)
(Quantity (- ?MatRackQuantity ?OrderQuantity))))))
```

圖 3.15 第一階段法則範例

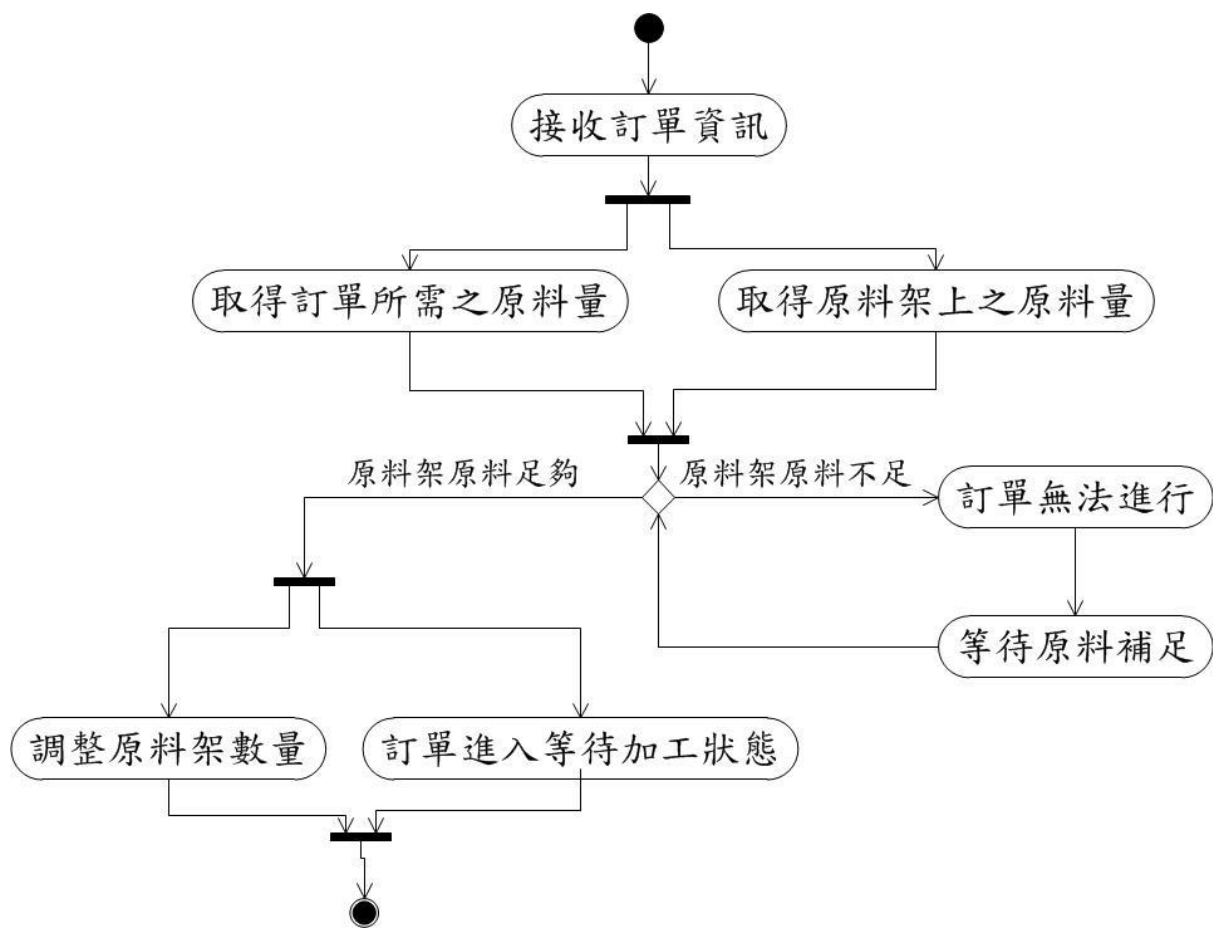


圖 3.16 第一階段法則系統活動流程圖

(2) 第二階段法則-決定哪一台 AGV 將把原料從原料架上搬運至屬於銑削機之待加工料架上：

由於原料進入加工首先會經過粗銑削與精銑削，因此在第一階段會先決定由哪台 AGV 將原料從原料架搬運到屬於銑削機的待加工料架上，本研究 AGV 假定為 3 台，法則中以 AGV1 為最高優先選擇權，次之為 AGV2，最後為 AGV3。以圖 3.17 的法則為例，當有等待之訂單時，若待加工料架的狀態為無原料且 AGV1 狀態有閒置，則驅動 AGV1，若此狀態條件無法滿足，則會驅動 AGV2 或 AGV3 之法則，若三台 AGV 皆忙碌，則等待 AGV 完成當下任務，若選擇的 AGV 發生異常，顯示異常狀態後，驅動其他閒置 AGV 進行搬運，其系統活動流程圖如圖 3.18 所示。

```
(defrule AGV1-transport-before-machining
(declare (salience 950))
?WaitingOrder <-
(WaitingOrder(OrderNumber ?OrderNumber)(Material ?Material)(Quantity ?
OrderQuantity)(DueDate ?DueDate)(Program ?Program))
?QueMatDesk <- (QueMatDesk(DeskType ?QueMatDeskType)(Status 0))
?AGV <- (HandlingDeviceAGV(DeviceName AGV1)(Status 2))
=>
(retract ?WaitingOrder)
(retract ?QueMatDesk)
(retract ?AGV)
(assert (MachiningOrder(OrderNumber ?OrderNumber)(Material ?Material)
(Quantity ?OrderQuantity)(DueDate ?DueDate)(Program ?Program)))
(assert (QueMatDesk(DeskType ?QueMatDeskType)(Status 1)))
(assert (HandlingDeviceAGV(DeviceName AGV1)(Status 1)
(AGVProg ?QueMatDeskType))))
```

圖 3.17 第二階段法則範例

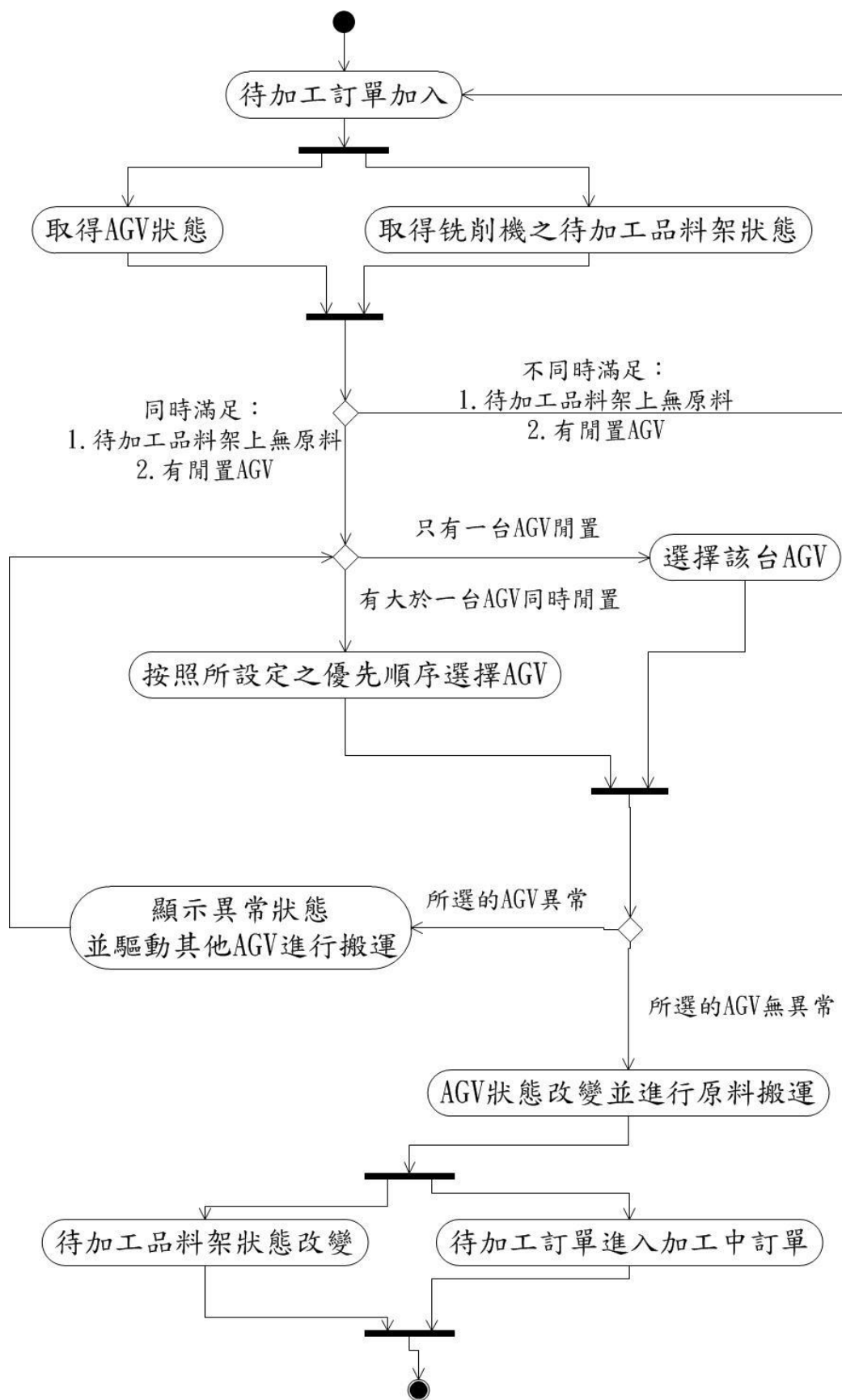


圖 3.18 第二階段法則系統活動流程圖

(3) 第三階段法則-決定哪一台 Mobile Robot 將把原料從屬於銑削機之待加工料架上搬運至相對應之銑削機進行加工：

當銑削機之待加工料架上有原料放置時，需要機械手臂將原料放置銑削機進行加工，本研究 Mobile Robot 假定為 2 台，法則中以 Robot1 為最高優先選擇權，再來為 Robot2。如圖 3.19 之 Robot1 法則範例，當待加工料架上有原料且銑削機之狀態為閒置時，若 Robot1 狀態也是閒置，則驅動 Robot1 執行；反之，若狀態條件無法滿足，則判定 Robot2 法則，若皆忙碌，則等待 Mobile Robot 完成當下任務。其系統活動流程圖如圖 3.20 所示。

```
(defrule Robot1-CNCa1-transport-during-milling
(declare (salience 800))
?MachiningOrder <-
(MachiningOrder(OrderNumber ?OrderNumber)(Material ?Material)(Quantity
?OrderQuantity)(DueDate ?DueDate)(Program ?Program))
?QueMatDesk <- (QueMatDesk(DeskType a1)(Status 1))
?CNC <- (MachiningDeviceCNCmilling(DeviceName CNCa1)(Status 2))
?Robot <- (HandlingDeviceRobot(DeviceName Robot1)(Status 3))
=>
(retract ?CNC)
(retract ?Robot)
(assert (MachiningDeviceCNCmilling(DeviceName CNCa1)(Status 1)
(MainProg ?Program)))
(assert (HandlingDeviceRobot(DeviceName Robot1)(Status 2)(RobotProg a1))))
```

圖 3.19 第三階段法則範例

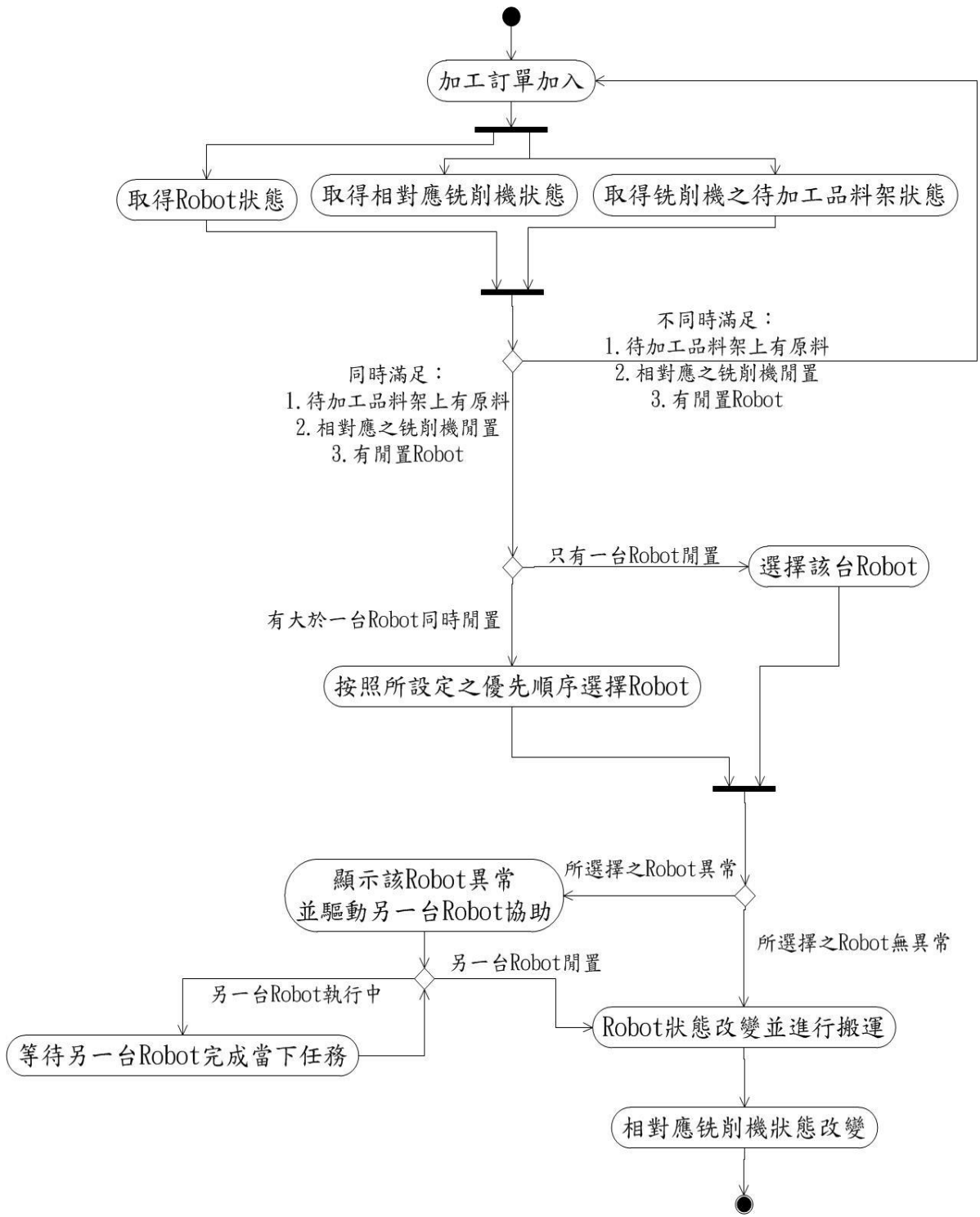


圖 3.20 第三階段法則系統活動流程圖

(4) 第四階段法則-決定哪一台 Robot 將把銑削後之原料搬運至屬於雕刻機之待加工料架上：

當原料經過粗銑削與精銑削之過程後，會進入雕刻作業，因此需要 Mobile Robot 將銑削後原料搬運至屬於雕刻機之待加工料架上，本研究 Mobile Robot 假定為 2 台，法則中以 Robot1 為最高優先選擇權，再來為 Robot2。如圖 3.21 之 Robot1 法則範例，當原料經過銑削後且雕刻機待加工料架之狀態為閒置時，若 Robot1 狀態也是閒置，則驅動 Robot1 執行將原料搬到雕刻機之待加工料架上；反之，若狀態條件無法滿足，則判定 Robot2 法則，若皆忙碌，則等待 Mobile Robot 完成當下任務。其系統活動流程圖如圖 3.22 所示。

```
(defrule Robot1-from-CNCa1milling-to-QueMatDesk-c1
(declare (salience 700))
?MachiningOrder <-
(MachiningOrder(OrderNumber ?OrderNumber)(Material ?Material)
(Quantity ?OrderQuantity)(DueDate ?DueDate)(Program ?Program))
?CNC <- (MachiningDeviceCNCmilling(DeviceName
CNCa1)(Status 2)(MainProg ?Program))
?QueMatDesk <- (QueMatDesk(DeskType c1)(Status 0))
?Robot <- (HandlingDeviceRobot(DeviceName Robot1)(Status 3))
=>
(retract ?CNC)
(retract ?QueMatDesk)
(retract ?Robot)
(assert (QueMatDesk(DeskType c1)(Status 1)))
(assert (HandlingDeviceRobot(DeviceName Robot1)(Status 2)
(RobotProg mA1C1))))
```

圖 3.21 第四階段法則範例

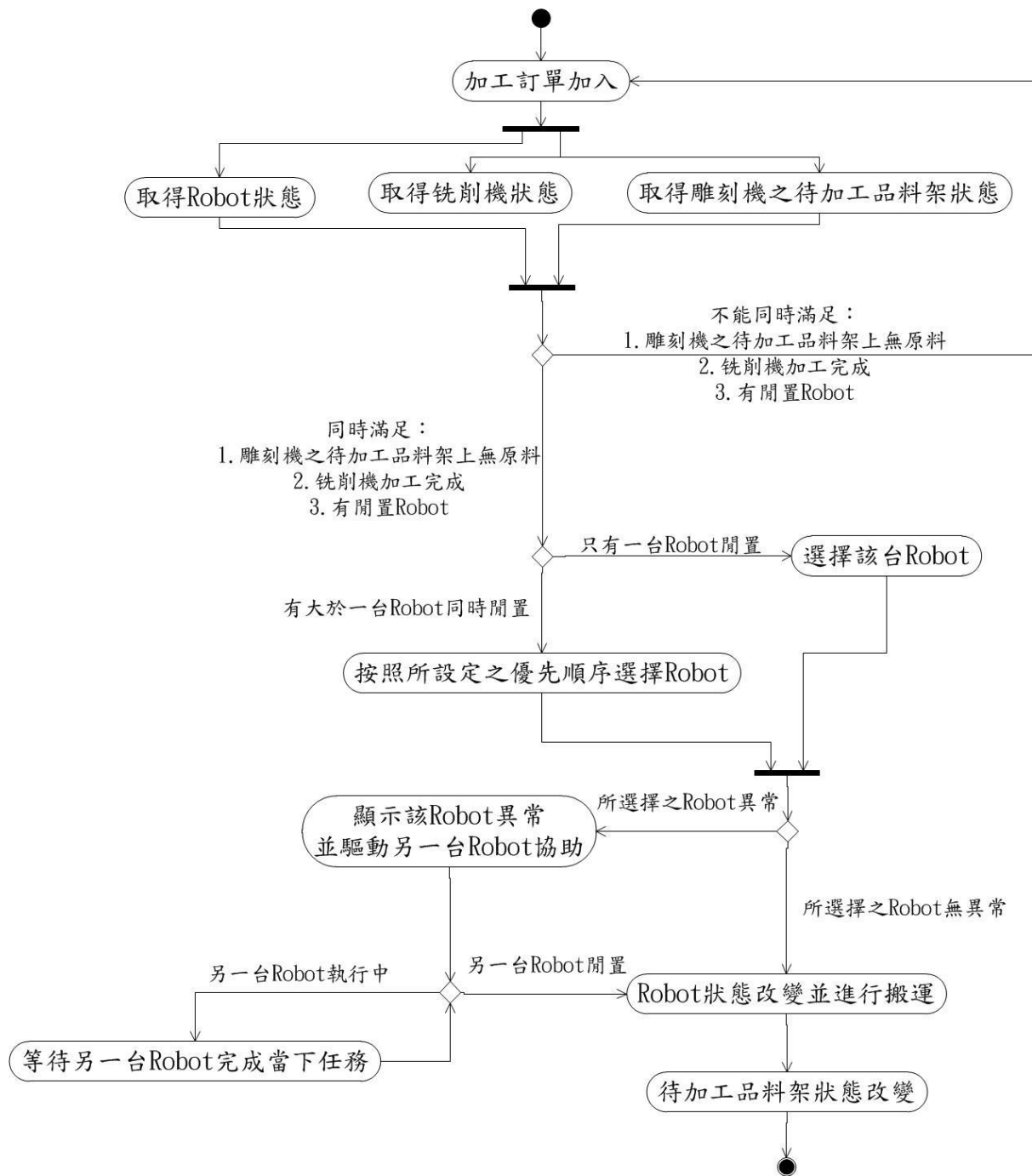


圖 3.22 第四階段法則系統活動流程圖

(5) 第五階段法則-決定哪一台 Robot 將把原料從屬於雕刻機之待加工料架搬運至相對應之雕刻機進行加工：

當雕刻機之待加工料架上有原料放置時，亦需要機械手臂將原料放置雕刻機進行加工，本研究 Mobile Robot 假定為 2 台，法則中以 Robot1 為最高優先選擇權，再來為 Robot2。如圖 3.23 之 Robot1 法則範例，當待加工料架上有原料且雕刻機之狀態為閒置時，若 Robot1 狀態也是閒置，則驅動 Robot1 執行；反之，若狀態條件無法滿足，則判定 Robot2 法則，若皆忙碌，則等待該 Mobile Robot 完成當下任務。其系統活動流程圖如圖 3.24 所示。

```
(defrule Robot1-CNCc1-transport-during-Engraving
(declare (salience 650))
?MachiningOrder <-
(MachiningOrder(OrderNumber ?OrderNumber)(Material ?Material)
(Quantity ?OrderQuantity)(DueDate ?DueDate)(Program ?Program))
?QueMatDesk <- (QueMatDesk(DeskType c1)(Status 1))
?CNC <- (MachiningDeviceCNCengraving(DeviceName
CNCc1)(Status 2))
?Robot <- (HandlingDeviceRobot(DeviceName Robot1)(Status 3))
=>
(retract ?CNC)
(retract ?Robot)
(assert (MachiningDeviceCNCengraving(DeviceName
CNCc1)(Status 1)(MainProg ?Program)))
(assert (HandlingDeviceRobot(DeviceName Robot1)(Status 2)
(RobotProg c1))))
```

圖 3.23 第五階段法則範例

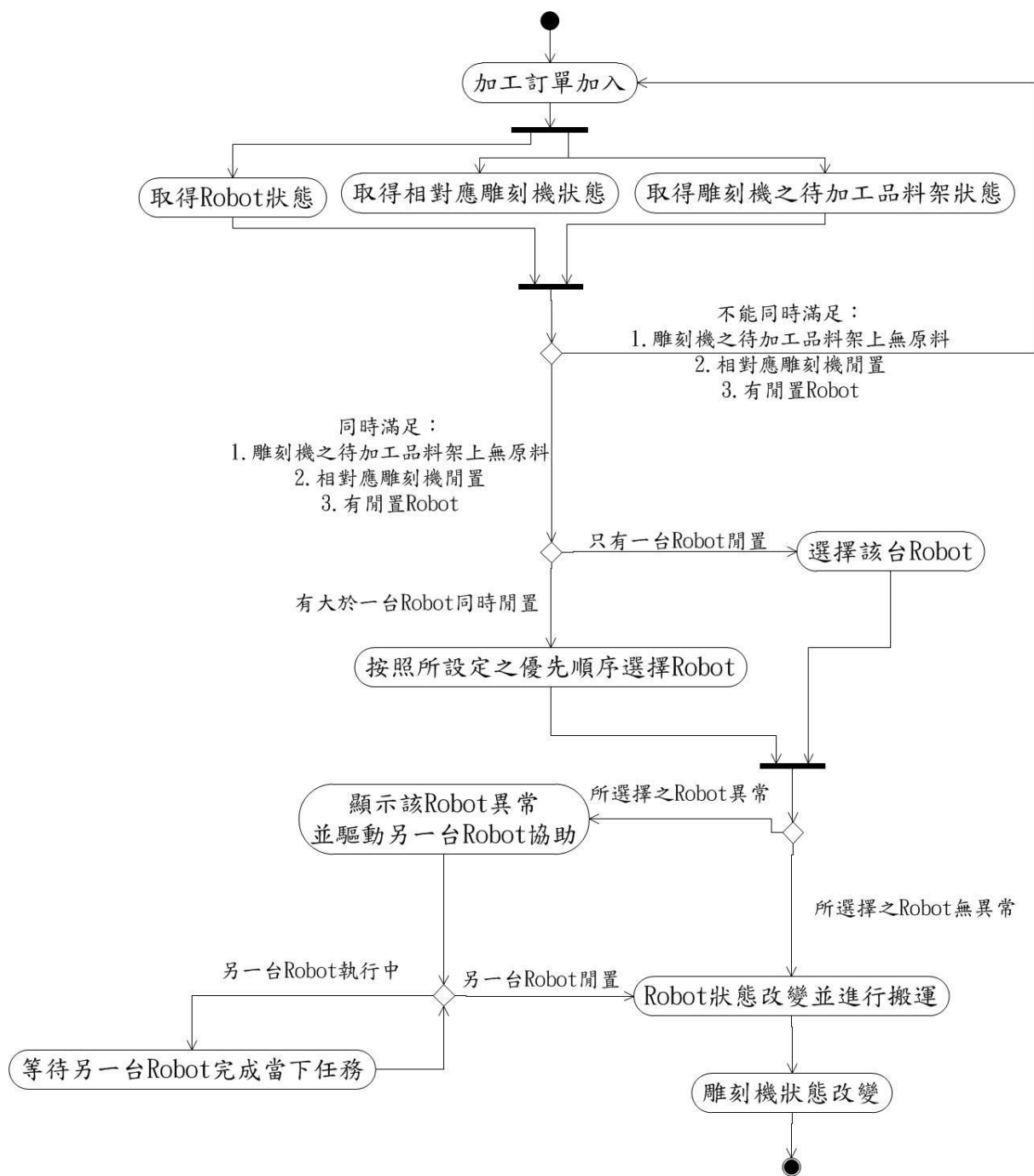


圖 3.24 第五階段法則系統活動流程圖

(6) 第六階段法則-決定哪一台 Robot 將加工完之原料搬運到完成品待運送區：

當原料經過最後雕刻加工過程後，需要 Mobile Robot 將加工後原料搬運至完成品待運送區，本研究 Mobile Robot 假定為 2 台，法則中以 Robot1 為最高優先選擇權，再來為 Robot2。如圖 3.25 之 Robot1 法則範例，當原料加工完且完成品待運送區狀態為沒有原料時，若 Robot1 狀態也是閒置，則驅動 Robot1 執行將原料送至完成品待運送區；反之，若狀態條件無法滿足，則判定 Robot2 法則，若皆忙碌，則等待 Mobile Robot 完成當下任務。其系統活動流程圖如圖 3.26 所示。

```
(defrule
Robot1-transport-from-engravingCNCc1-to-QueProductDesk-d1
(declare (salience 550))
?MachiningOrder <-
(MachiningOrder(OrderNumber ?OrderNumber)(Material ?Material)
(Quantity ?OrderQuantity)(DueDate ?DueDate)(Program ?Program))
?CNC <- (MachiningDeviceCNCengraving(DeviceName
CNCc1)(Status 2)(MainProg ?Program))
?QueProductDesk <- (QueProductDesk(DeskType d1)(Status 0))
?Robot <- (HandlingDeviceRobot(DeviceName Robot1)(Status 3))
=>
(retract ?QueProductDesk)
(retract ?Robot)
(assert (QueProductDesk(DeskType d1)(Status 1)))
(assert (HandlingDeviceRobot(DeviceName Robot1)(Status 2)
(RobotProg c1d1))))
```

圖 3.25 第六階段法則範例

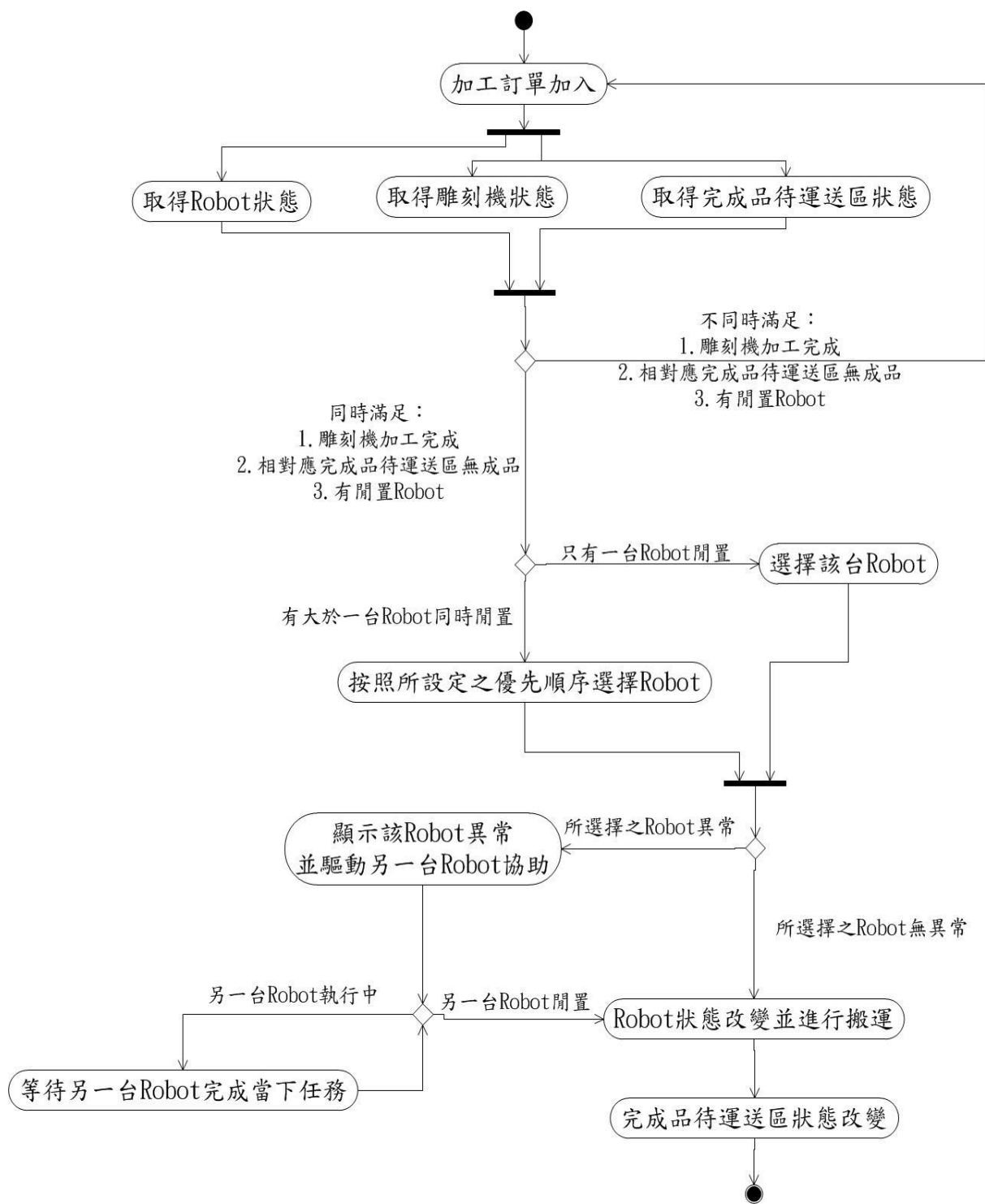


圖 3.26 第六階段法則系統活動流程圖

(7) 第七階段法則-決定哪一台 AGV 將完成品從完成品待運送區搬運至成品架上：

由於原料加工完畢後須送入成品架上，因此決定由哪台 AGV 將完成品搬運至成品架上，而本研究 AGV 假定為 3 台，法則中以 AGV1 為最高優先選擇權，次之為 AGV2，最後為 AGV3。以圖 3.27 的法則為例，當完成品待運送區有成品時，若 AGV1 狀態有閒置，則驅動 AGV1，若此狀態條件無法滿足，則會驅動 AGV2 或 AGV3 之法則，若三台 AGV 皆忙碌，則等待 AGV 完成當下任務，其系統活動流程圖如圖 3.28 所示。

```
(defrule AGV1-transport-after-machining
(declare (salience 450))
?MachiningOrder <-
(MachiningOrder(OrderNumber ?OrderNumber)(Material ?Material)
(Quantity ?OrderQuantity)(DueDate ?DueDate)(Program ?Program))
?QueProductDesk <-
(QueProductDesk(DeskType ?QueProductDeskType)(Status 1))
?AGV <- (HandlingDeviceAGV(DeviceName AGV1)(Status 2))
=>
(retract ?MachiningOrder)
(retract ?QueProductDesk)
(retract ?AGV)
(assert
(FinOrder(OrderNumber ?OrderNumber)(Material ?Material)(Quantity ?OrderQuantity)(DueDate ?DueDate)))
(assert (HandlingDeviceAGV(DeviceName AGV1)(Status 1)
(AGVProg ?QueProductDeskType)))
(assert (QueProductDesk(DeskType ?QueProductDeskType)(Status 0))))
```

圖 3.27 第七階段法則範例

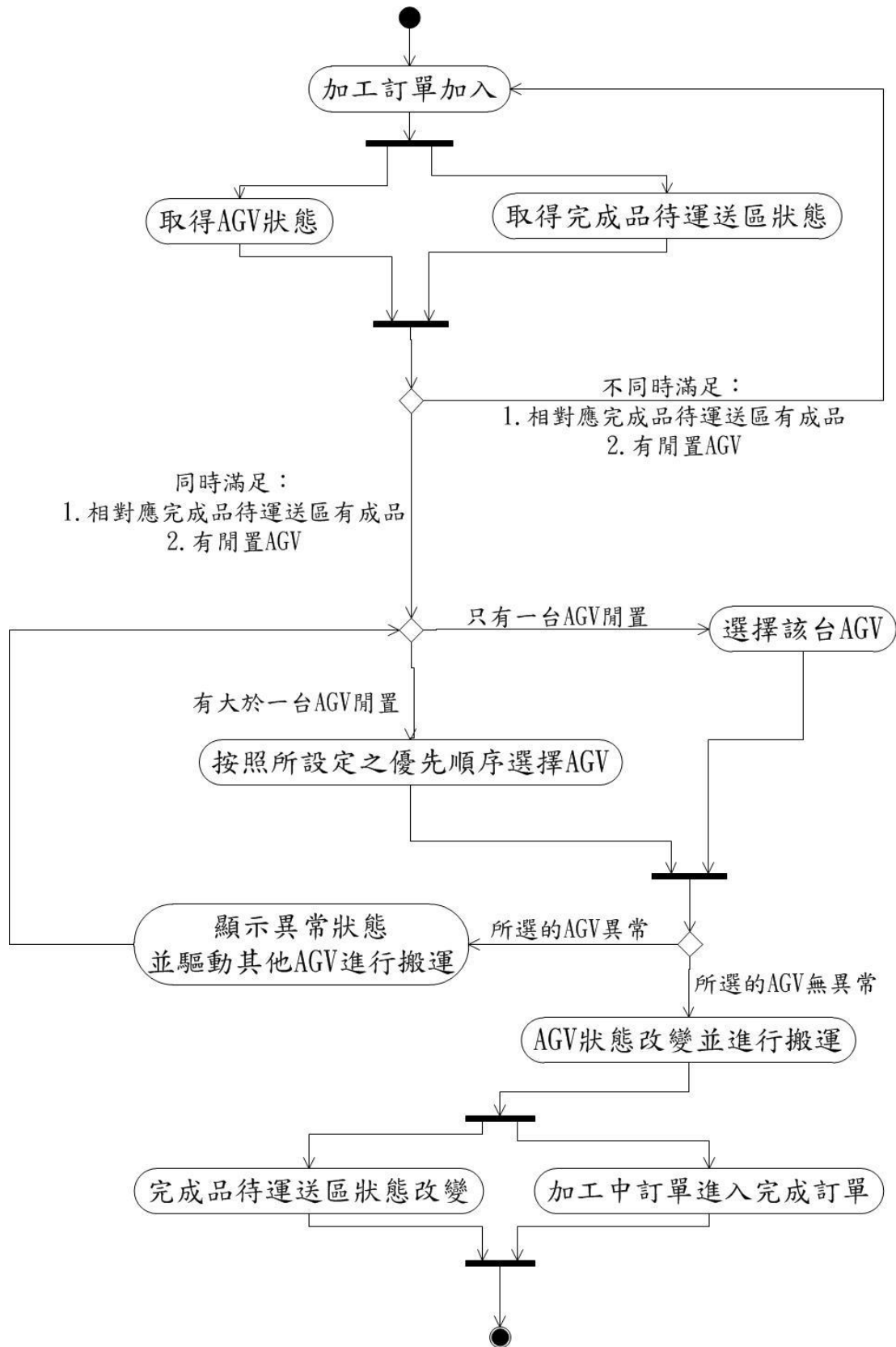
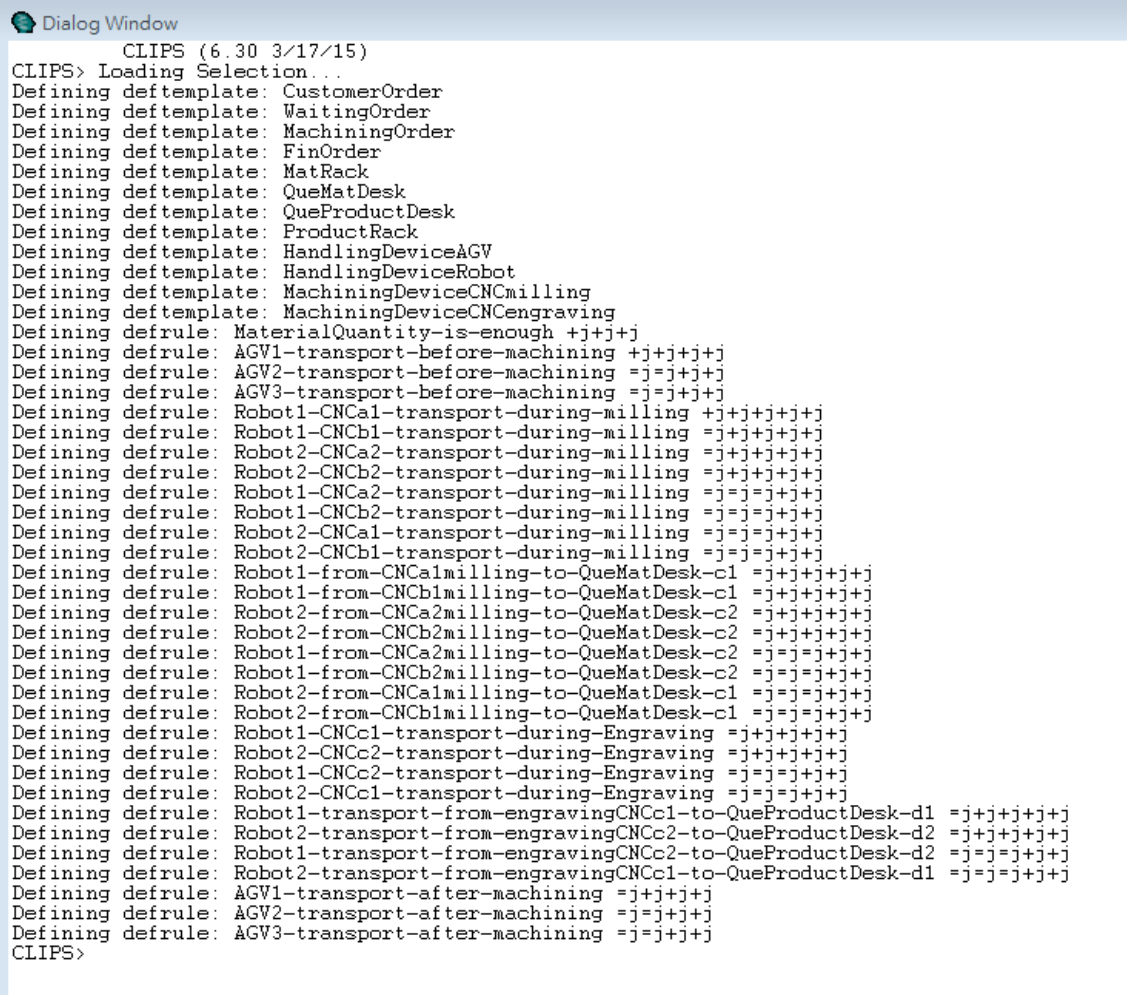


圖 3.28 第七階段法則系統活動流程圖

建立完法則後，將先利用 CLIPS 軟體進行法則檢查，確定所建立之所有 Template 和 Rule 無撰寫上的問題亦無互相抵觸、衝突等錯誤，確認之後再進行後續情境模擬。如圖 3.29 之畫面並無錯誤資訊即為法則編寫沒問題可進入後續整合驗證。



```
Dialog Window
CLIPS (6.30 3/17/15)
CLIPS> Loading Selection...
Defining deftemplate: CustomerOrder
Defining deftemplate: WaitingOrder
Defining deftemplate: MachiningOrder
Defining deftemplate: FinOrder
Defining deftemplate: MatRack
Defining deftemplate: QueMatDesk
Defining deftemplate: QueProductDesk
Defining deftemplate: ProductRack
Defining deftemplate: HandlingDeviceAGV
Defining deftemplate: HandlingDeviceRobot
Defining deftemplate: MachiningDeviceCNCmilling
Defining deftemplate: MachiningDeviceCNCengraving
Defining defrule: MaterialQuantity-is-enough +j+j+j
Defining defrule: AGV1-transport-before-machining +j+j+j+j
Defining defrule: AGV2-transport-before-machining =j=j+j+j
Defining defrule: AGV3-transport-before-machining =j=j+j+j
Defining defrule: Robot1-CNCa1-transport-during-milling +j+j+j+j+j
Defining defrule: Robot1-CNCb1-transport-during-milling =j+j+j+j+j
Defining defrule: Robot2-CNCa2-transport-during-milling =j+j+j+j+j
Defining defrule: Robot2-CNCb2-transport-during-milling =j+j+j+j+j
Defining defrule: Robot1-CNCa2-transport-during-milling =j=j=j+j+j
Defining defrule: Robot1-CNCb2-transport-during-milling =j=j=j+j+j
Defining defrule: Robot2-CNCa1-transport-during-milling =j=j=j+j+j
Defining defrule: Robot2-CNCb1-transport-during-milling =j=j=j+j+j
Defining defrule: Robot1-from-CNCa1milling-to-QueMatDesk-c1 =j+j+j+j+j+j
Defining defrule: Robot1-from-CNCb1milling-to-QueMatDesk-c1 =j+j+j+j+j+j
Defining defrule: Robot2-from-CNCa2milling-to-QueMatDesk-c2 =j+j+j+j+j+j
Defining defrule: Robot2-from-CNCb2milling-to-QueMatDesk-c2 =j+j+j+j+j+j
Defining defrule: Robot1-from-CNCa2milling-to-QueMatDesk-c2 =j=j=j+j+j+j
Defining defrule: Robot1-from-CNCb2milling-to-QueMatDesk-c2 =j=j=j+j+j+j
Defining defrule: Robot2-from-CNCa1milling-to-QueMatDesk-c1 =j=j=j+j+j+j
Defining defrule: Robot2-from-CNCb1milling-to-QueMatDesk-c1 =j=j=j+j+j+j
Defining defrule: Robot1-CNCc1-transport-during-Engraving =j+j+j+j+j+j
Defining defrule: Robot2-CNCc2-transport-during-Engraving =j+j+j+j+j+j
Defining defrule: Robot1-CNCc2-transport-during-Engraving =j=j=j+j+j+j
Defining defrule: Robot2-CNCc1-transport-during-Engraving =j=j=j+j+j+j
Defining defrule: Robot1-transport-from-engravingCNCc1-to-QueProductDesk-d1 =j+j+j+j+j+j
Defining defrule: Robot2-transport-from-engravingCNCc2-to-QueProductDesk-d2 =j+j+j+j+j+j
Defining defrule: Robot1-transport-from-engravingCNCc2-to-QueProductDesk-d2 =j=j=j+j+j+j
Defining defrule: Robot2-transport-from-engravingCNCc1-to-QueProductDesk-d1 =j=j=j+j+j+j
Defining defrule: AGV1-transport-after-machining =j+j+j+j+j
Defining defrule: AGV2-transport-after-machining =j=j+j+j+j
Defining defrule: AGV3-transport-after-machining =j=j+j+j+j
CLIPS>
```

圖 3.29 於 CLIPS 中確定所建立之 template 和 rule 無錯誤

第四章 情境模擬

4.1 開發環境和工具

表 4.1 和表 4.2 為本研究之電腦開發環境和軟硬體需求，包括 CPU 規格、記憶體規格、作業系統、推論工具、實體設備環境等。

表 4.1 本研究電腦開發環境

開發環境	說明
作業系統-Windows 版本	Windows 7 家用進階版
CPU 規格	I5-2450M 2.50GHz
記憶體規則	8.00GB
系統類型	64 位元

表 4.2 本研究軟硬體需求工具

軟硬體需求	說明
Protégé 4.3	建立本體論框架
CLIPS 6.3	建立法則推論工具
Visual Studio 2017	整合系統
SkyMars	截取所需 CNC 資訊
HIWIN RA605	機械手臂
Mitsubishi FX3G-24MR	PLC 模擬 AGV 動作
Visio 2013	繪製系統架構層工具
Access 2013	建立原料資料庫

4.2 實體設備介紹

4.2.1 截取 CNC 資訊之軟體-SkyMars Professional

SkyMars 是由台灣財團法人精密機械發展研究中心 (Precision Machinery Research Development Center, PMC)所開發的軟體，可遠端連接多個不同廠牌之工具機並進行監視，SkyMars Professional 版則提供了較完善的功能，主要為工具機資訊監視、NC 上傳下載、異常履歷警報、零組件維護管理、刀具維護管理、監測伺服及主軸的負載及轉速與機台稼動率服務等。除了上述幾項功能外，SkyMars Professional 亦提供了 API，SkyMars API 提供了 18 種類別資訊，使用者無須了解不同控制器廠牌的函式庫，即可方便地撰寫或是擴允應用程式，大幅降低了整合的困難度， SkyMars Professional 使用介面如圖 4.1 所示。

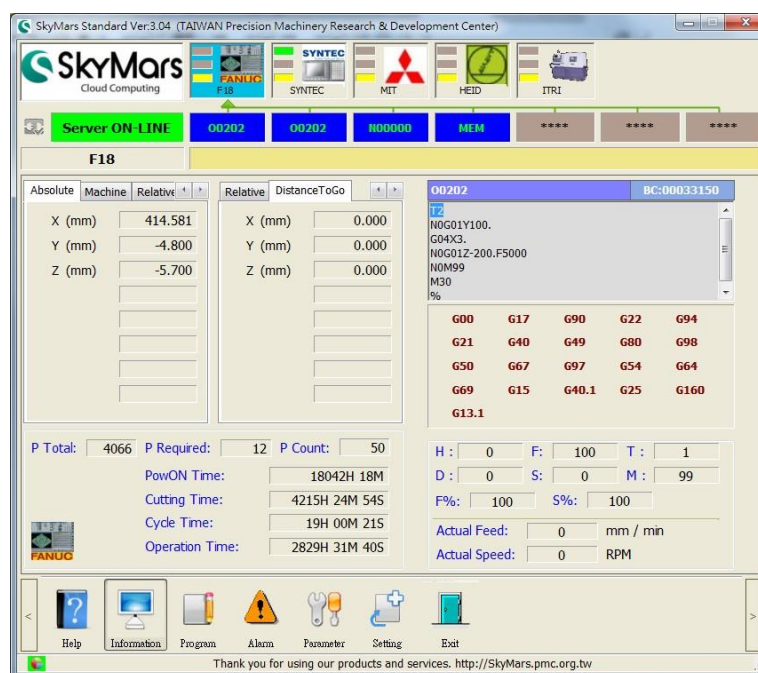


圖 4.1 SkyMars Professional 軟體介面

資料來源：PMC SkyMars 官網

目前 SkyMars 允許連線的控制器廠牌為：Fanuc、Mitsubishi、Heidenhain、新代控制器、寶元控制器與工研院控制器。

4.2.2 機械手臂-HIWIN RA605

機械手臂在自動化產線上扮演了非常重要的角色，而本研究所使用到的機械手臂為台灣上銀科技公司所開發的六軸關節式機器手臂-RA605，除了可應用於物件搬運之外，也可用於組裝作業或較複雜的加工任務，並具備了可聯網之功能，易於與其他設備串聯整合。圖 4.2 為此六軸關節式機械手臂實際圖。



圖 4.2 上銀六軸關節式機器手臂-RA605

此機械手臂亦提供了 Offline 的模擬軟體，讓使用者在實體操作前，能先透過離線軟體確認所撰寫之手臂程式是否正確，如圖 4.3 所示。同時也提供軟體開發套件，可使用外部程式如 C#、C++或 VB 來截取實體手臂或 Offline 模擬軟體之資訊。

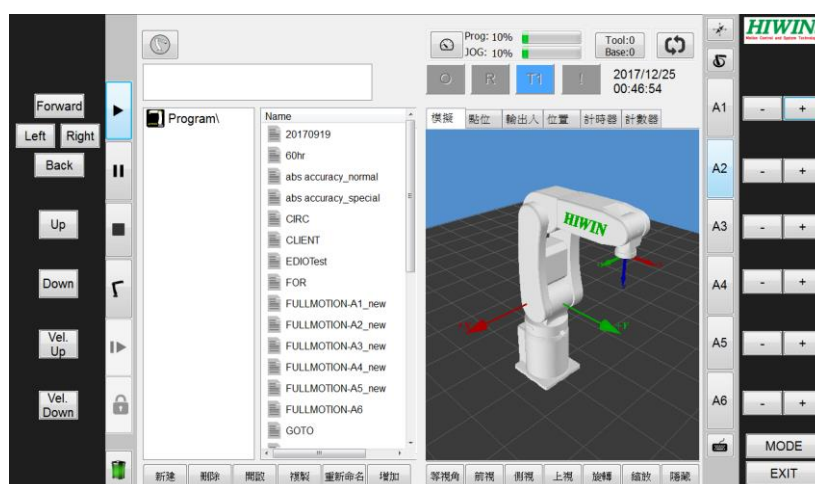


圖 4.3 上銀六軸關節式機器手臂離線模擬軟體

4.2.3 模擬無人搬運車(AGV)動作之 PLC-Mitsubishi FX3G-24MR

本研究中因研究限制，無實際無人搬運車(AGV)可做使用，因此利用可程式控制器(Programmable Logic Controller, PLC)來模擬 AGV 之動作。本研究所使用之 PLC 為三菱電機所開發之 FX3G-24MR 型號，輸入點有 14 點，輸出點有 10 點。本研究使用了 TCP/IP 之通訊協定格式來串聯各設備，因此為了能讓 PLC 連網，搭配了乙太網路模組，透過轉換介面卡讓 PLC 與乙太網路模組連接，如此一來，PLC 便具備了可聯網之功能，透過網路線將電腦(PC)與 PLC 直接連線，若需要與其他設備進行整合，也可透過集線器(HUB)，將設備與 PC 端串聯。本研究實際上即是透過 HUB 將設備端與 PC 端串聯，如圖 4.4 所示。

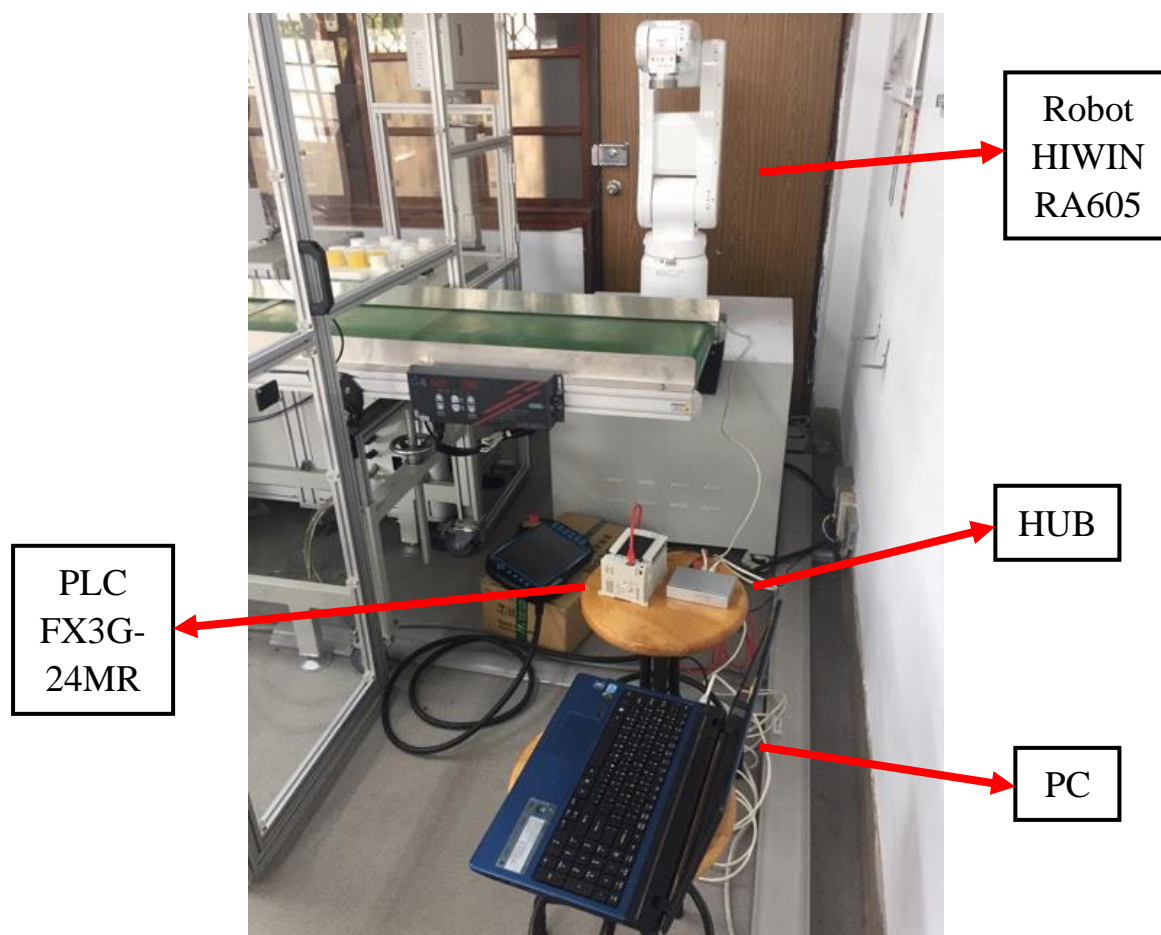


圖 4.4 透過集線器(HUB)將設備端與 PC 端串聯之實際圖

4.3 情境描述與環境設定

本研究之情境描述如 3.3 節所述，本章節將更進一步說明詳細的環境設定與相關實體限制等，一一條列如下：

1. 實體層設備狀態，CNC 與 Robot 皆直接以現有之設備的狀態資訊做判定依據，而 AGV、待加工料架與完成品待運送區之狀態則為本研究自訂義。實際上，這些設備狀態是可被擴展來應用的，在更多的法則判定上會更加完善，但仍需仰賴設備商來開發。
2. 本研究情境中，同種搬運設備有不只一台，因此為了預防撞機或不必要的危險，設立了機台優先次序。
3. 本研究設立的情境中，實際上會有很多種情況發生如網路斷線、機台需要維護等，但本研究僅聚焦在設備與設備之間如何藉由狀態資訊來協調合作，核心目的如 1.2 節所述。
4. 本研究無考慮插單、成本、詳細搬運與加工時間等因素。

4.4 設備與專家系統整合

本研究透過 C# 進行設備和專家系統之整合，其整合介面如圖 4.5 所示，包含訂單資訊輸入區、不同狀態之訂單區、原料架資訊、工具機資訊、機械手臂資訊、AGV 資訊以及設備狀態呈現區。

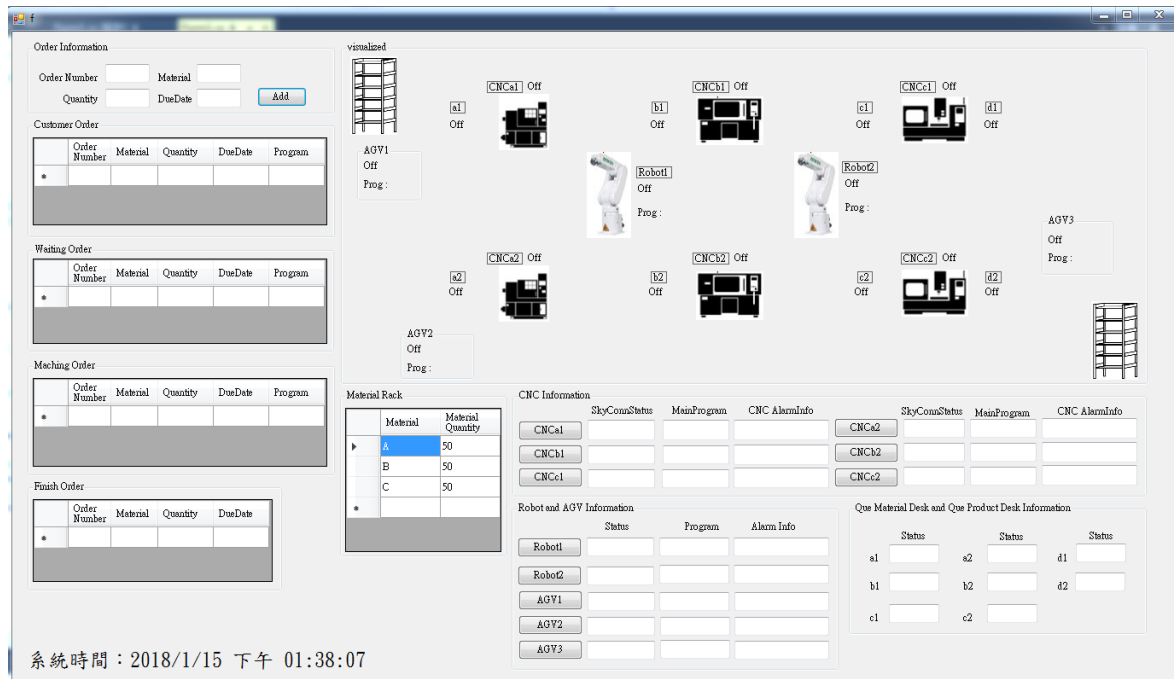


圖 4.5 本研究整合介面圖

功能分別描述如下：

- (1) Order Information：用來輸入顧客訂單的資訊，包含訂單編號、產品原料、數量以及交期。
- (2) Customer Order：顧客訂單輸入完畢後，資料會呈現於此區。此區資料應用在第一階段法則，與原料架之原料種類、數量做比較，比較後若顧客訂單確定可加工，則進入 Waiting Order 狀態。
- (3) Waiting Order：顧客訂單確定可加工後，將進入此區呈現。此區的資料會應用在第二階段法則，等待 AGV 將原料搬運至待加工料架後，此區的資料會轉為 Machining Order 狀態。
- (4) Machining Order：當 AGV 搬運原料至待加工料架後，訂單則進入此區呈現。此區的資料會應用在第三階段到最後第七階段，直到加工完畢後，此區的資料才轉為 Finish Order。

- (5) Finish Order：加工完畢後的訂單，將進入此區呈現。
- (6) Visualized：此區為以較易理解的方式呈現設備狀態。
- (7) Material Rack：顯示原料架內有的原料種類和數量，此區直接連結原料資料庫，並取用資料庫內的資料。此區的資料應用於第一階段法則。
- (8) CNC Information：呈現各個工具機機台資訊。
- (9) Robot Information：呈現各個 Mobile Robot 的機台資訊。
- (10) AGV Information：呈現各個 AGV 機台資訊。
- (11) Que Material Desk and Que Product Desk Information：呈現每一個待加工料架以及完成品待運送區之設備狀態。

而專家系統亦整合於 C# 中，如圖 4.6 示意圖，將本研究之 CLIPS 檔已嵌入至 C# 中，其檔名為「researchCLIPS.CLP」，此檔案必須同時放置於同專案的文件目錄底下，檔案內即是本研究所建立之各個 Template 與各項判斷法則。將檔案嵌入後，需將 CLIPSNET.dll 檔加入參考並引用才可使用 CLIPS 語法來進行後續串接，如圖 4.7 所示。引用完 CLIPSNET.dll 後，於程式碼內需給予 CLIPS 之環境並 Load 出欲使用的 researchCLIPS.CLP 檔，並讓其 reset，即可進行後續的串接，如圖 4.8。本介面功能在於透過 C# 所擷取到的設備資訊等實際事實，進入專家系統進行推論後，最後再將結果呈現回介面。

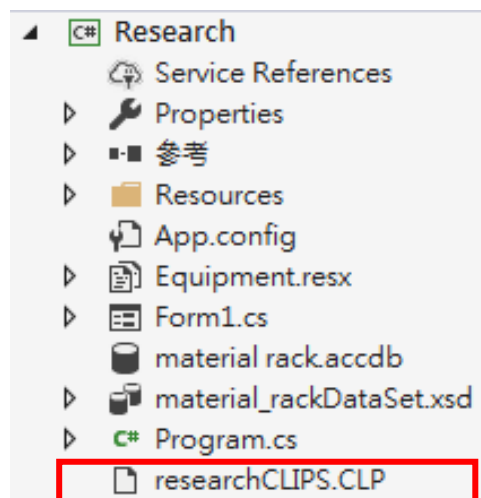


圖 4.6 專家系統 CLIPS 嵌入 C# 專案之示意圖

```
Research
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10 using System.Runtime.Remoting;
11 using System.Runtime.Remoting.Channels;
12 using System.Runtime.Remoting.Channels.Tcp;
13 using InterfaceLib;
14
15 using CLIPSNET;
16
17
```

圖 4.7 引用 CLIPSNET.dll

```
namespace WindowsFormsApplication1
{
    3 個參考
    public partial class Form1 : Form
    {
        private CLIPSNET.Environment clips = new CLIPSNET.Environment();

        1 個參考
        public Form1()
        {
            InitializeComponent();
            clips.LoadFromResource("WindowsFormsApplication1", "researchCLIPS.CLP");
            clips.Reset();
        }
    }
}
```

圖 4.8 載入 researchCLIPS.CLP 檔之方式

4.5 情境推論結果

本研究將利用情境模擬來說明本系統能透過所擷取到的實體設備狀態事實，利用專家系統的推論，產生因應方式。在情境模擬中，將進行一段連續時間的驗證，在這一連續時間內包含正常情況與機台異動情況，於本章節將會呈現一段系統從正常情況到異常情況，一直到回復正常情況的作動，並分步說明。

如圖 4.9 所示，原料架內有 A、B、C 原料且數量為各 50 個。現今，有一訂單需要 30 個原料 B 進行加工，而介面所擷取到的實體機台資訊為待加工料架 b1 上有原料，而 Robot1 正進行將 b1 之原料搬到 CNCb1 上，因此狀態為執行中(RUN)，其餘的設備皆為閒置(IDLE)。

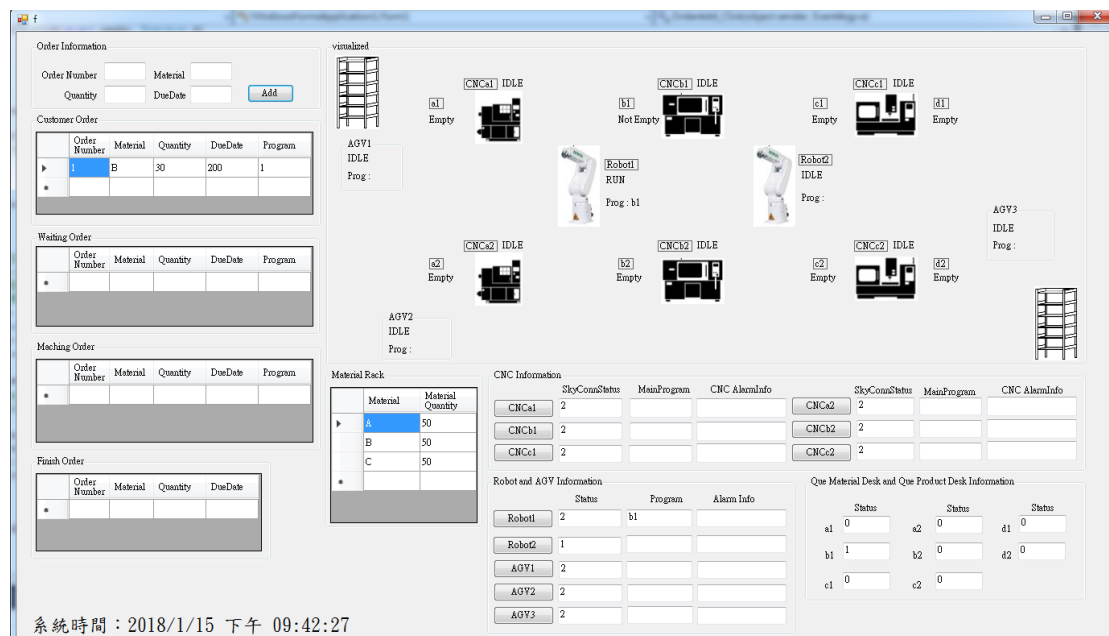


圖 4.9 情境模擬圖一

透過第一階段法則的判定，因訂單所需之原料數小於原料架上的數量，可進行後續加工，因此顧客訂單將轉換為等待中訂單，且原料架上之數量相對減少，如圖 4.10 所示。

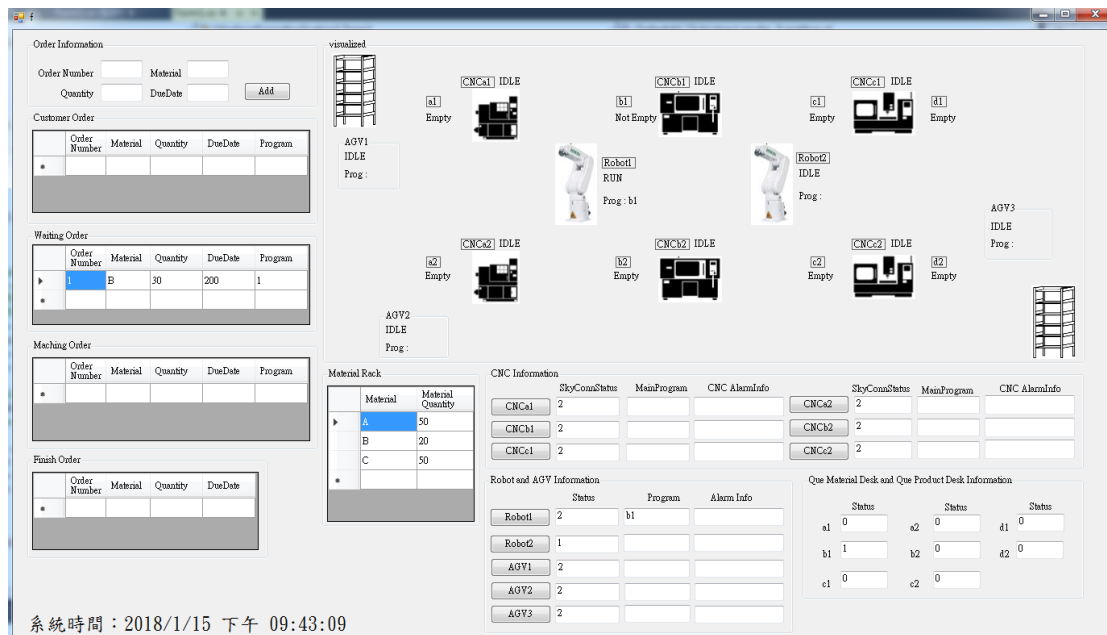


圖 4.10 情境模擬圖二

當有等待中的訂單時，系統則透過第二階段法則判定哪一台 AGV 將搬運原料到待加工料架上，在圖 4.10 的設備狀態可看到三台 AGV 皆屬於閒置狀態，且待加工料架 a1、a2、b2 皆無原料(Empty)，表示此三台 AGV 都可接受指令，因此第二階段法則將驅動 AGV1 搬運原料到 a1；AGV2 搬運原料到 a2 以及 AGV3 搬運原料到 b2，同時，等待中的訂單也轉變為加工中訂單，如圖 4.11。

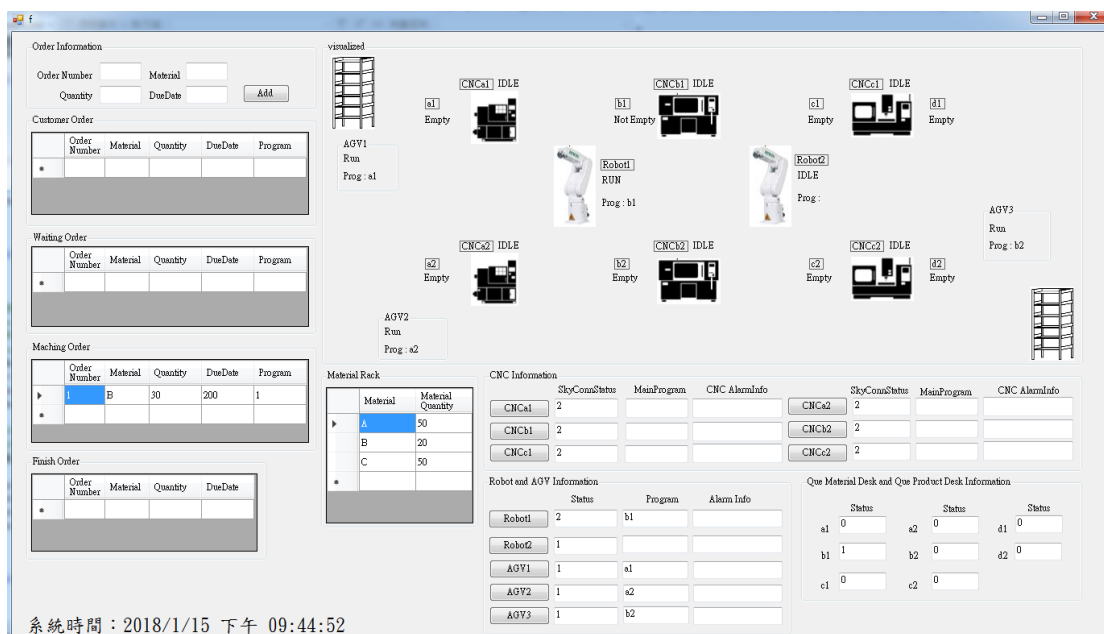


圖 4.11 情境模擬圖三

當三台 AGV 皆進行完搬運後，其狀態都恢復成閒置狀態，而待加工料架 a1、a2 與 b2 狀態則變為有原料的狀態(Not Empty)，如圖 4.12 所示，此時原先正進行將原料從待加工料架 b1 搬運到 CNCb1 的 Robot1 執行完任務後也恢復成閒置狀態，CNCb1 則變為加工中，而待加工料架 b1 亦變為無原料狀態。

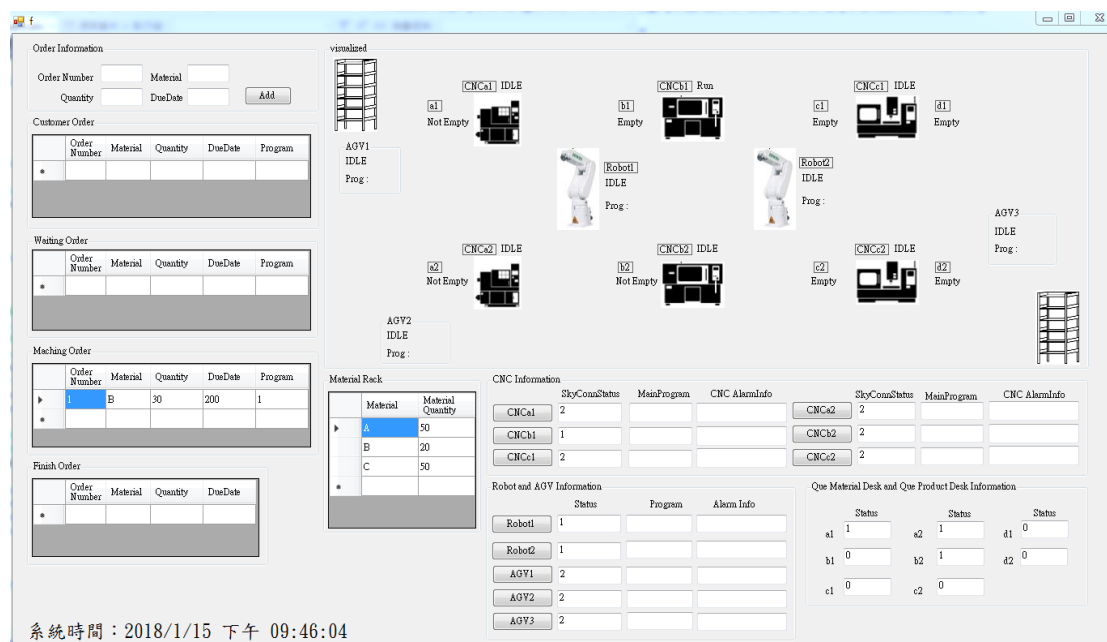


圖 4.12 情境模擬圖四

當待加工料架 a1、a2、b2 有原料時，Robot1 與 Robot2 將會負責將原料運送至相對應 CNC 機台上，此階段應用到的法則為第三階段法則。如圖 4.13 所示，Robot1 狀態改變為執行中且是執行 a1 程式，意即將待加工料架 a1 的原料搬運到 CNCa1 上，Robot2 狀態改變為執行中且是執行 a2 程式，意即將待加工料架 a2 的原料搬運到 CNCa2 上，而待加工料架 b2 上的原料則等待 Robot 執行完任務後再進行。

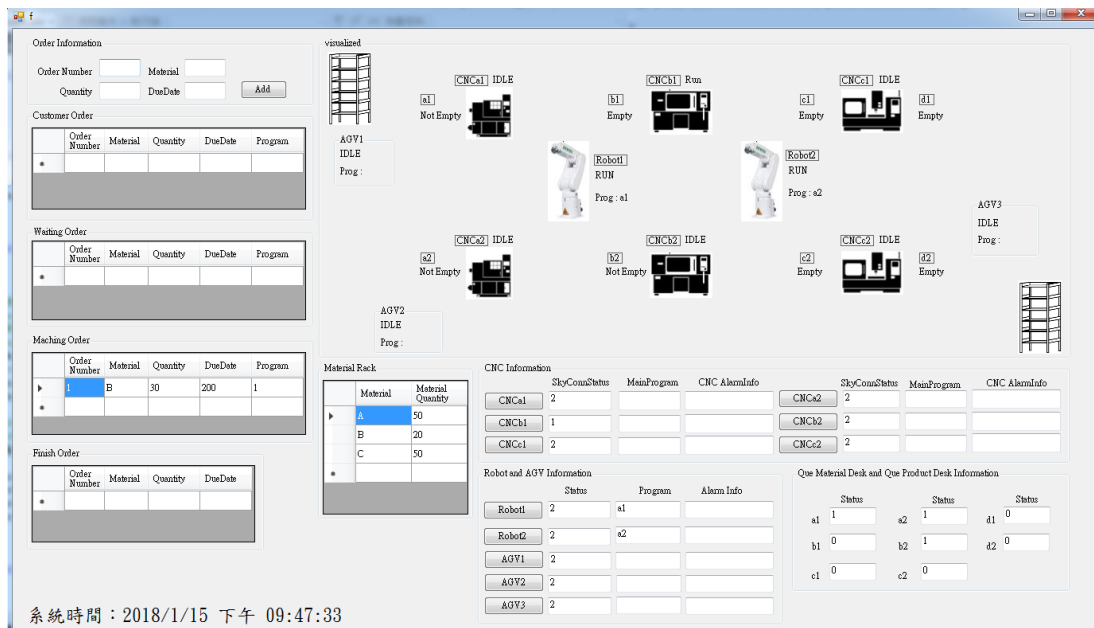


圖 4.13 情境模擬圖五

如圖 4.14 所示，當 Robot1 與 Robot2 搬運完 a1 與 a2 之原料到相對應的 CNC 後，Robot1 之狀態變回閒置中，Robot2 則進行待加工料架 b2 的搬運任務，同時，CNCa1 與 CNCa2 狀態改變為執行中，且依據訂單所匹配的加工程式進行加工，而待加工料架 a2 的狀態轉變為無原料。在此時機點，CNCb1 與 AGV1 發生了警報，如介面中所顯示之錯誤警報訊息，在此階段這兩台異常的設備必須進行修復，於此系統中無法進行任務，因此可視為機台移除。

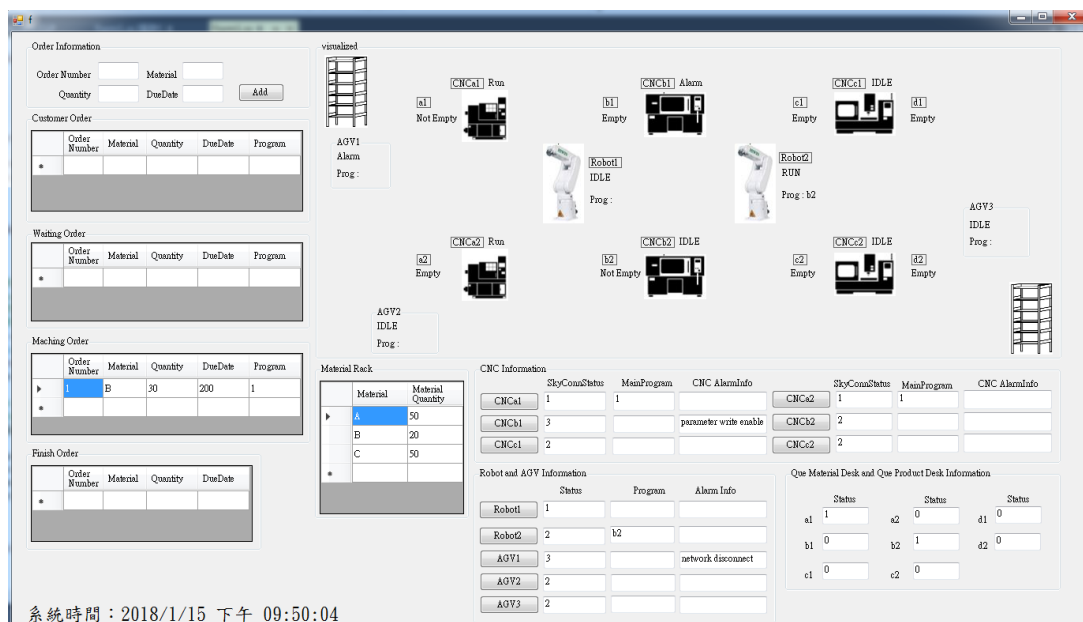


圖 4.14 情境模擬圖六

由於 AGV1 機台因發生異常被移除，原先應由 AGV1 將原料搬運至待加工料架 a2 的任務，將由 AGV2 來進行，此情況應用了第二階段法則，如圖 4.15 所示。而因 CNCb1 發生異常也被移除無法作使用，因此待加工料架 b1 亦無法接收原料。

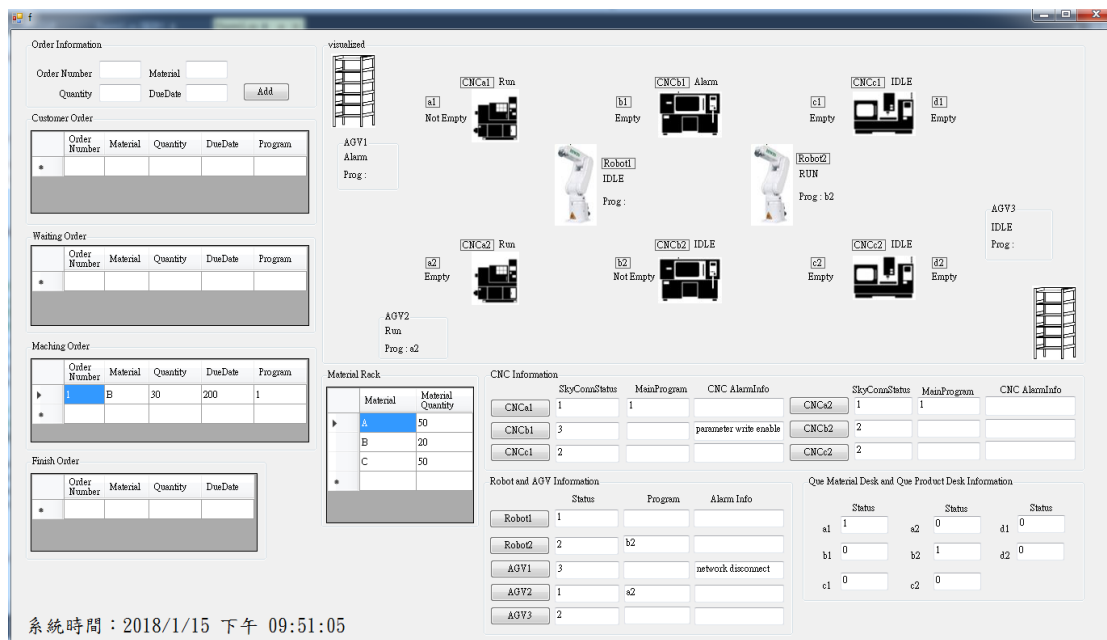


圖 4.15 情境模擬圖七

AGV2 將原料運送完後狀態則變回閒置，待加工料架 a2 則為有原料，如圖 4.16 所示。

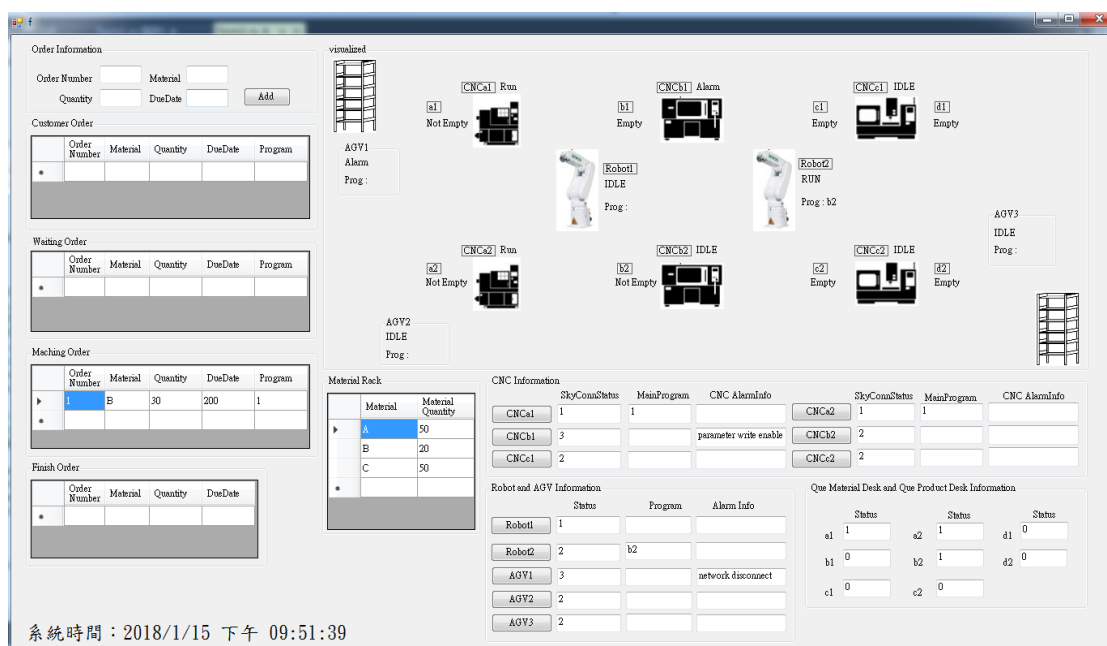


圖 4.16 情境模擬圖八

如圖 4.17 所示，CNCA2 狀態為閒置但有程式 1 存在，表示 CNCA2 加工完成，因此需要 Robot 將原料搬運到雕刻機之待加工料架 c2 上，但於圖中可看到 Robot2 正在執行待加工料架 b2 原料的搬運，因此驅動 Robot1 協助將 CNCA2 之原料搬運到 c2 上，如圖 4.18 所示，Robot1 的狀態改變為執行中，且執行將原料從 CNCA2 搬至 c2 之程式，此情境應用了第四階段法則。

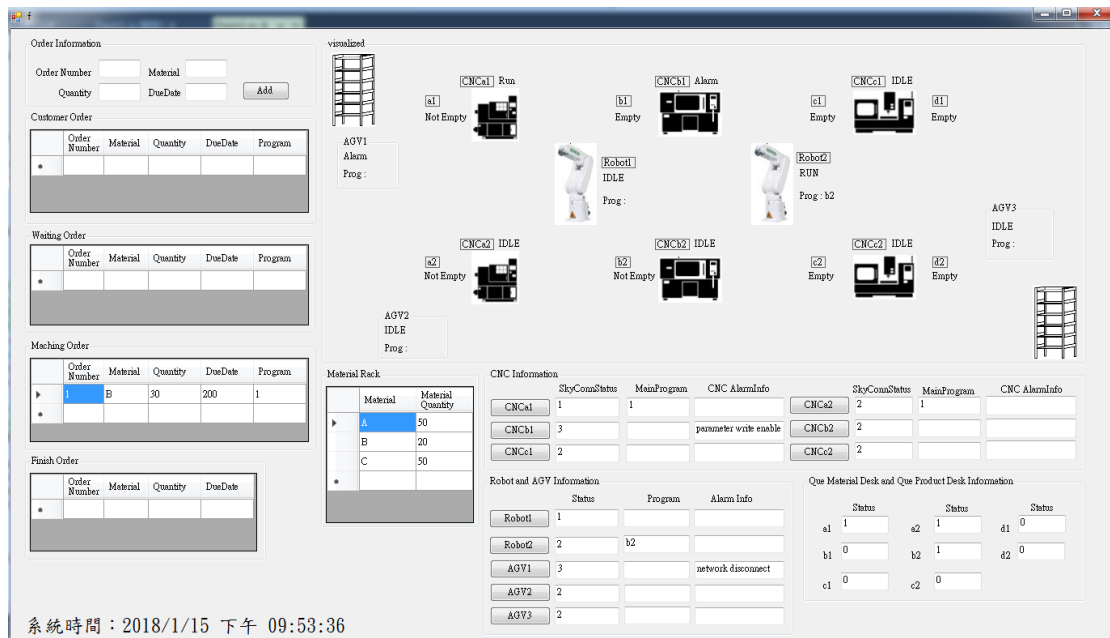


圖 4.17 情境模擬圖九

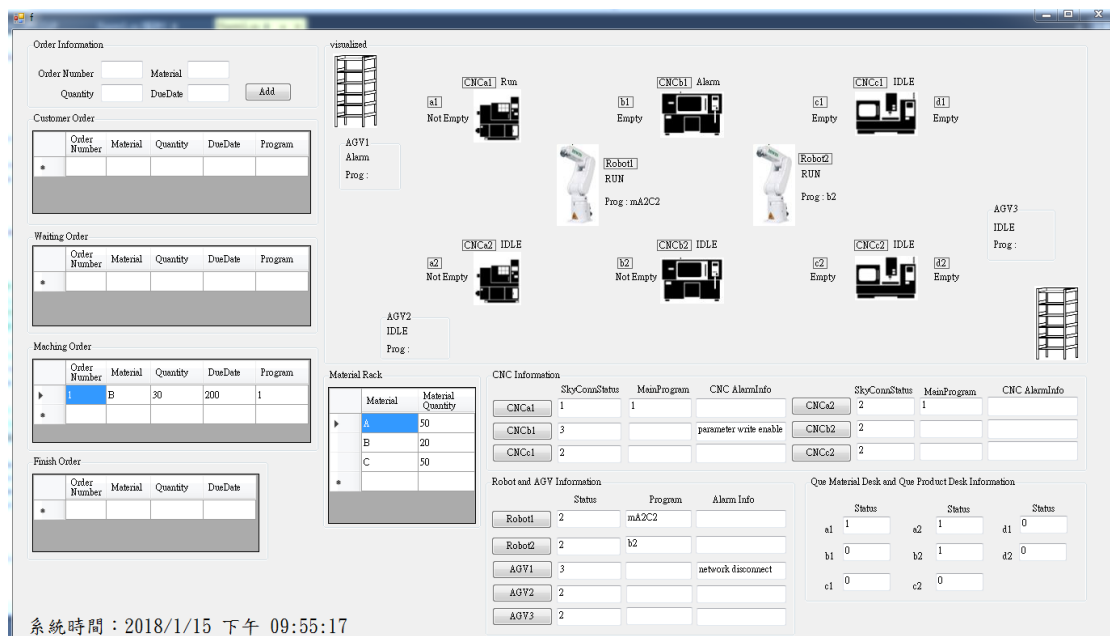


圖 4.18 情境模擬圖十

Robot1 搬運完後狀態恢復成閒置中，CNCA2 狀態則為閒置且無加工程式，而待加工料架 c2 的狀態變為有原料狀態。同時，Robot2 也執行完原先的 b2 搬運任務，狀態也恢復成閒置中，CNCA2 狀態則為加工中，如圖 4.19。

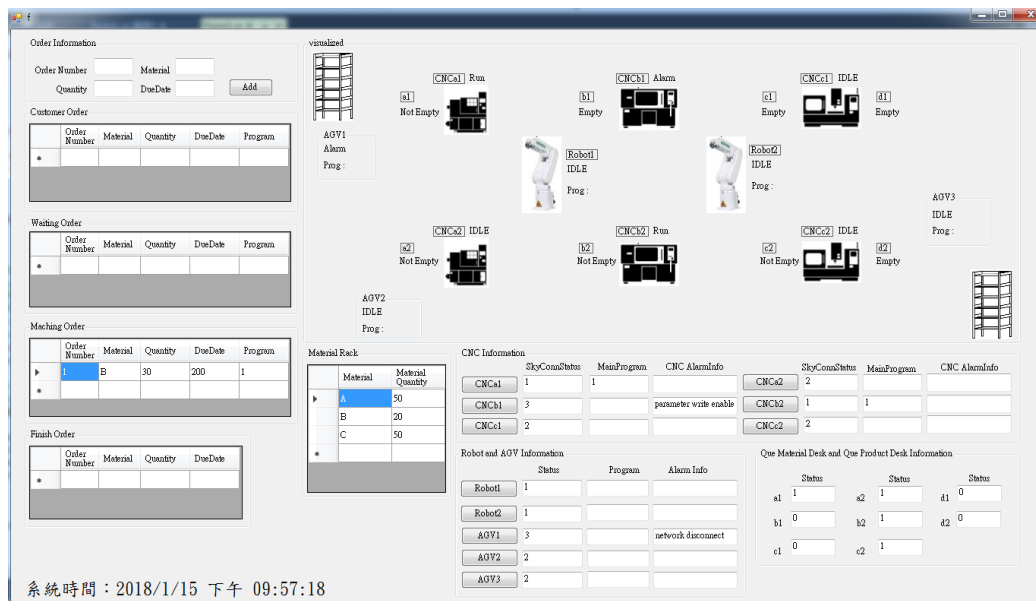


圖 4.19 情境模擬圖十一

CNCA2 恢復閒置後，由於先前已有原料透過 AGV2 搬運至 a2 上，因此會驅動 Robot2 將 a2 之原料搬運到 CNCA2 上做加工，此情境應用第三階段法則。而經過一段時間後，原先發生異常的 CNCb1 與 AGV1 已修復好並重新回到產線上，可視為機台之增加，如圖 4.20。

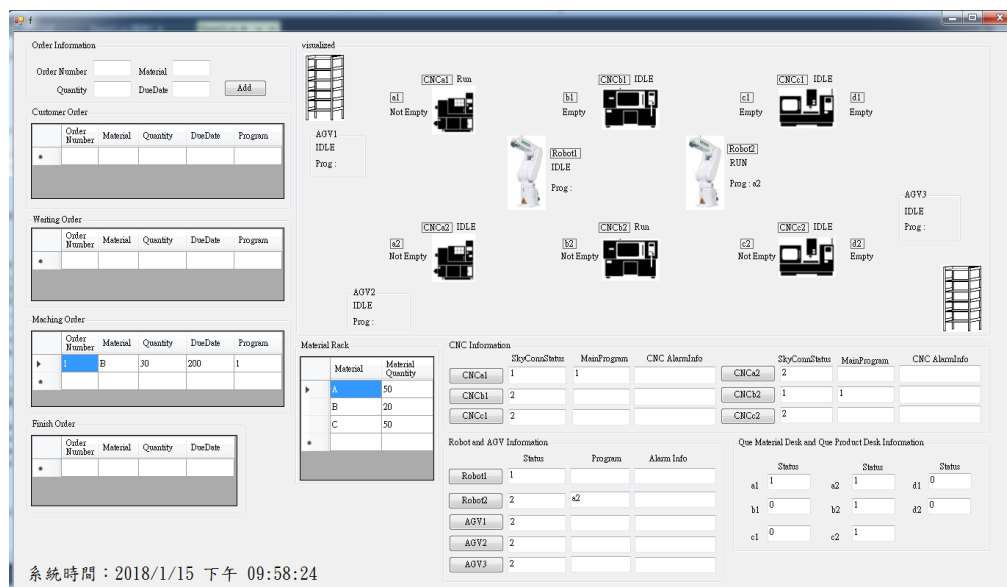


圖 4.20 情境模擬圖十二

CNCb1 與 AGV1 機台增加回來後，因原先待加工料架 b1 為無原料，因此則驅動 AGV1 進行加工，因此 AGV1 狀態改變，如圖 4.21 所示，此情境應用了第二階段法則。

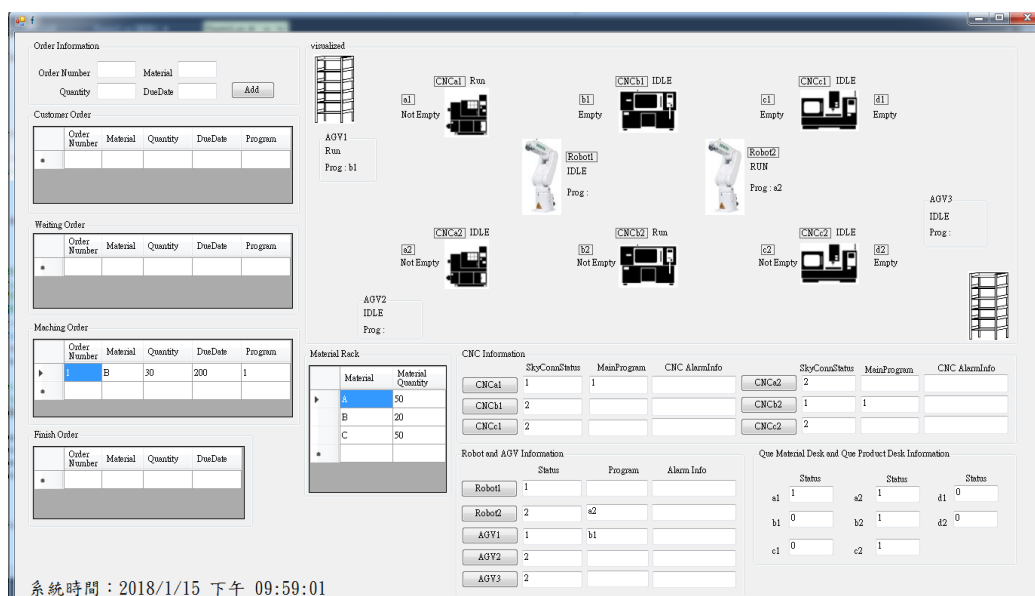


圖 4.21 情境模擬圖十三

AGV1 搬運完後狀態恢復為閒置，而待加工料架 b1 則為有原料狀態。而因 Robot2 正進行待加工料架 a2 的原料搬運，但同時待加工料架 c2 的原料也需要搬運至 CNCc1 上，此時因 Robot1 的狀態為閒置，所以驅動 Robot1 進行待加工料架 c2 原料的搬運，如圖 4.22，此情境應用了第五階段法則。

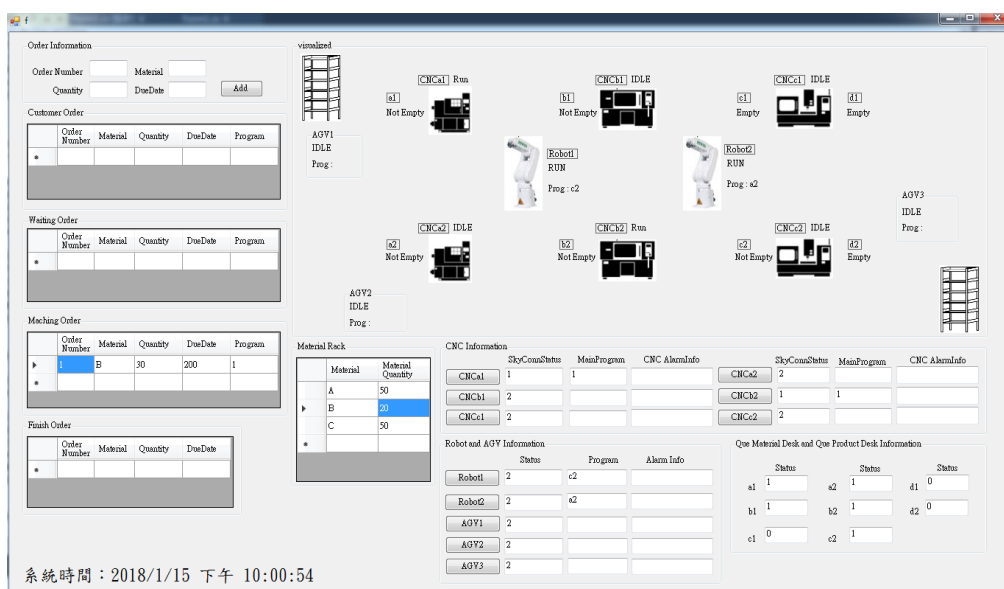


圖 4.22 情境模擬圖十四

Robot1 與 Robot2 分別搬運完後，狀態皆恢復為閒置，而 CNCa2 與 CNCc2 狀態則為執行中，且執行與訂單所匹配之加工程式，如圖 4.23。

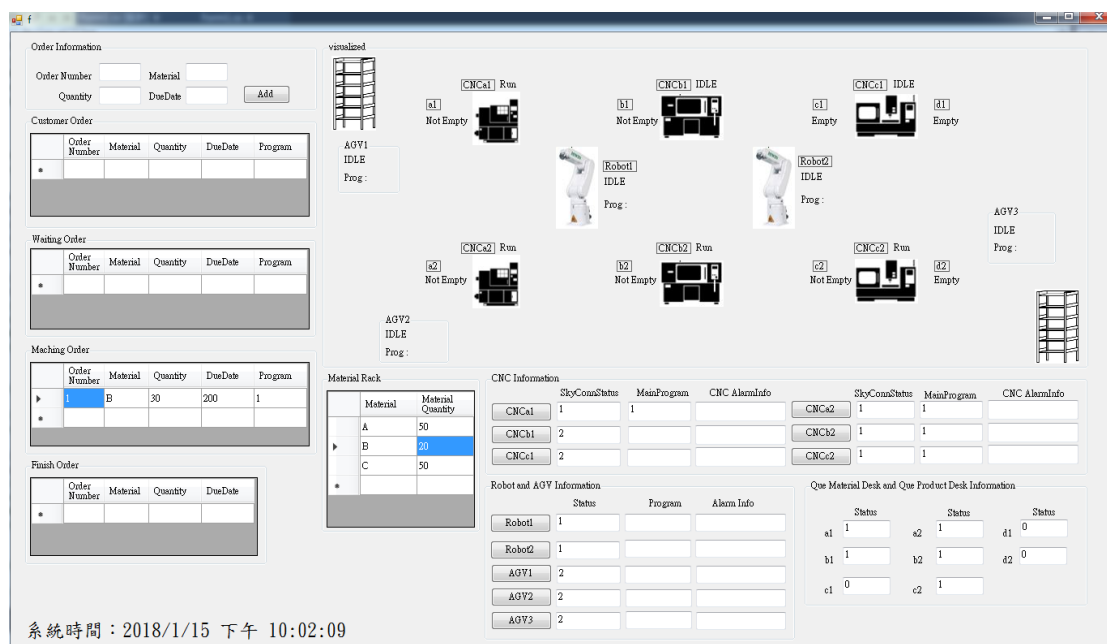


圖 4.23 情境模擬圖十五

4.6 小結

本章節利用金屬加工之情境模擬來驗證所建置之本體關係與相關事實之法則推論，主要利用實體設備狀態事實做為控制的依據，並說明系統如何解決並達到本研究中兩個欲解決之情境，而法則中所設立之機台優先次序為使用者可自訂義。

從此章節亦可以得知，搭配所建置的製造環境領域本體與法則推論知識表示，能讓系統更加智慧化去做判定，讓機台之間能互相因應且降低重新規劃控制的困難度，並更加走向「隨需而至」。

第五章 結論與未來發展方向

在工業 4.0 的浪潮下，將智慧工廠的概念導入機械加工業已是目前許多製造業者正努力進行的願景。然而，要使設備能提高自主性並達智慧化生產，除了工廠管理者無論是否遠端，都必須即時清楚了解工廠內所發生的事件之外，其設備之間的相互協調必然是重要的環節之一，因此，實際上工廠內的設備本體的關係、狀態、訂單狀態等知識共享的意義相當具有必要性。

本研究透過建置知識本體框架來描述製造環境領域之知識關聯後，再依據此框架撰寫後續法則推論之內容，主要建立以工廠內機台狀態為判定依據，驅動設備之間進行協調。本研究的特點如下：

1. 透過本體論定義，藉由知識共享，使得製造環境領域之間的物件關聯更明確。
2. 透過法則式知識推論，當物件和法則定義的更加完善時，可以達到實體控制。
3. 在本研究中，利用了相當具有可行性與整合性的 TCP/IP 通訊協定格式來進行實體層的設備整合，因此在本研究中能夠容易地組織實體層的訊號，在這一方面做了眾多實體資源管理。

本研究主要初步證實利用本體論以及知識法則推論的方法，能夠使得機械加工工廠以更智慧化的方式解決本情境模擬中所列舉的情況。然而，在機械加工製造領域內有更多可能發生的例外情況，例如緊急插單、網路異常甚至是工廠內部電供應不足等狀況，情境種類相當多且複雜，並不是本研究所能窮舉所有列出的，因此僅能針對特定情境去做解決。

本研究僅在設備狀態上進行知識推論控制，且因目前實際設備部分難以直接寫入控制之關係，因此在給予參數上為擷取現有實際設備狀態來驅動法則，未來若設備商開發的設備功能與可擷取、寫入的資訊更加開放，除了監視設備外亦可直接寫入控制時，可撰寫更加智慧化的控制系統，然而即使設備商提供了相當多的 API 以及所開放的資訊程度高，該如何擷取有效的資訊亦是一大問題，否則當設備資訊過多繁雜，將會對伺服器造成相當大的負荷。

另外，本研究在建模與撰寫法則的過程中，遭遇到了難以將專家系統法則「可視化」的困難，導致在建立法則時容易發生問題，難以快速了解法則是否與本體及實例相符合。未來，在法則建置的過程中，必須確定能夠因應欲解決之情境，未來，若在「可視化」這部分若能找到新的方法突破，在建構系統上將會大幅提升其方便性。

參考文獻

中文部分

1. 鄭弼仁(民 98)。基於 CLIPS 與 CSA 之分散式生產訂單允諾系統以 TFT-LCDArray 製程為例(碩士論文)。取自 <http://140.128.103.80:8080/handle/310901/1426>

英文部分

1. Alsafi, Y., & Vyatkin, V. (2010). Ontology-based reconfiguration agent for intelligent mechatronic systems in flexible manufacturing. *Robotics and Computer-Integrated Manufacturing*, 26(4), 381-391. doi:10.1016/j.rcim.2009.12.001
2. Bohuslava, J., Martin, J., & Igor, H. (2017, January). TCP/IP protocol utilisation in process of dynamic control of robotic cell according industry 4.0 concept. In *Applied Machine Intelligence and Informatics (SAMi)*, 2017 IEEE 15th International Symposium on (pp. 000217-000222). IEEE.
3. Bunge, M. (1977). *Treatise on Basic Philosophy: Ontology I: The Furniture of the World*. Dordrecht, Holland: D: Reidel Publishing Company.
4. Da'na, S., Sagahyoon, A., Elrayes, A., Al-Ali, A., & Al-Aydi, R. (2008). Development of a monitoring and control platform for PLC-based applications. *Computer Standards & Interfaces*, 30(3), 157-166.
5. Giarratano, J. C., & Riley, G. (1998). *Expert Systems*. Boston, MA, USA:PWS Publishing Co.
6. Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2), 199-220.
7. Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International journal of human-computer studies*, 43(5), 907-928.
8. Guarino, N. (1997). Understanding, building and using ontologies. *International journal of human-computer studies*, 46(2-3), 293-310.
9. Lemaignan, S., Siadat, A., Dantan, J. Y., & Semenenko, A. (2006, June). MASON: A proposal for an ontology of manufacturing domain. In *Distributed Intelligent Systems: Collective Intelligence and Its Applications*, 2006. DIS 2006. IEEE Workshop on (pp. 195-200). IEEE.
10. Negri, E., Fumagalli, L., Garetti, M., & Tanca, L. (2016). Requirements and languages for the semantic representation of manufacturing systems. *Computers in Industry*, 81, 55-66. doi:10.1016/j.compind.2015.10.009
11. Noy, N. F., Sintek, M., Decker, S., Crubézy, M., Fergerson, R. W., & Musen, M. A. (2001). Creating semantic web contents with protege-2000. *IEEE Intelligent systems*, 16(2), 60-71.

12. Shen, Q., Zhang, M., Song, Q., & Tang, Y. (2013, July). Research and Application of Web Information Retrieval Based on Ontology. In *International Conference on Computer, Networks and Communication Engineering (ICCNCE 2013)*. Atlantis Press.
13. Roland Berger. (2014, March). INDUSTRY 4.0 The new industrial revolution How Europe will succeed. *Think: ACT*, 10-11.
14. Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. *The knowledge engineering review*, 11(2), 93-136.