

行政院國家科學委員會專題研究計畫成果報告

供應鏈系統資訊物件模型之建構與整合

Object-Oriented Modeling and Integration of Supply Chain Information System

計畫編號：NSC 90-2212-E-002-206

執行期限：90 年 8 月 1 日至 90 年 7 月 31 日

主持人：陳正剛

(e-mail: achen@ccms.ntu.edu.tw)

執行機構及單位名稱：國立台灣大學工業工程學研究所

中文摘要

企業的全球化經營是一個無法抵擋的趨勢，伴隨而來的是更複雜且頻繁的企業間互動關係，如何有效地整合企業內與企業間資源頻繁的傳遞活動與快速的資訊交換，以保持企業競爭的優勢是目前各個企業無時無刻思考的問題。於是『供應鏈管理』成為極重要的關鍵。本研究以物件分析與設計的方法建構供應鏈管理系統的物件模型作為探討的重點，本研究先建立一個擴充型的供應鏈參考模型做為物件導向設計的起點，利用 UML 為模型開發工具，提出以產品為主要物件觀點的被使用案例模型與分析方法，並藉此建構出供應鏈管理系統的物件模型。最後以物件導向觀念的特質，提出評估物件模型架構優劣的參考觀點。我們亦將 EDI 的內容資料格式（以 Computing Industry 為例）予以物件化，最後並利用分散式物件仲介環境（CORBA）與物件導向資料庫（OODB）來整合供應鏈管理系統。

關鍵詞：供應鏈管理、電子資料交換、分散式物件導向技術、物件導向資料庫

Today's companies are facing increasing competition and volatility in a globally interconnected economy. To remain competitive now, industrial organizations are continually looking for ways to reduce product development time, improve product quality and service, enhance flexibility, and reduce production cycle time. In order to achieve these goals, the coordination among related enterprises plays an important role. Because processes and activities in an enterprise and among enterprises are not only heterogeneous but also complex, an effective Supply Chain Planning (SCP) system becomes very important. Our goal in this research is, therefore, to construct an integrated supply chain information system using OO approaches. We first define an "extended" Supply Chain Operation Reference (SCOR) Model, originally proposed by Supply Chain Council, as a requirement model for the SCM information system. Based on this extended reference

model, we begin OOA and OOD using Unified Modeling Language (UML). We develop a "Used Case" requirement analysis approach and analyze the characteristics and advantages of the approach using a set of criteria. Following construction of the most adequate object model, an integrated OO SCM information system is proposed using the CORBA (Common Object Request Broker Architecture) technology and an OO database. A prototype is developed in this research to demonstrate the proposed integrated system.

1. 緒論

供應鏈管理系統是一種需要高度整合的系統，這包括企業內資料管理的整合與企業間資料交換的整合。為了讓供應者與顧客能夠彼此快速且準確地回應彼此的需求，有效率的資料交換方式尤其是企業間所關心的一項重要技術。電子資料交換（EDI, Electronic Data Interchange）是目前廣為大家運用的資料交換方式，由於它的準確性、快速性與安全性，著實提升了企業的競爭力，然而也因為它在成本、整合能力、擴充客戶能力與資料『及時（Real Time）性』上的缺點，也讓許多企業望之卻步。

本研究利用物件導向大師 Booch, Rumbaugh 與 Jacobson 所發展出的 UML (Unified Modeling Language) [8] 做為物件導向分析與設計的工具，並參照 Supply Chain Council 所制訂的參考模型（Reference Model）來建構一完整的供應鏈模型，我們亦將 EDI 的內容資料格式（以 Computing Industry 為例）予以物件化，期望發展出一高度整合之分散式物件環境的架構，將所有企業內與企業間的資料傳遞都包裝成分散式的物件，以物件做為企業內與企業間資料交換的單元，藉以改善傳統系統的缺點與不足之處。

2. 供應鏈系統的需求模型

發展物件導向系統，首先要建立系統使用者的需求描述，進而產生系統的需求模型（requirement model）。對供應鏈系統而言，必須先瞭解供應鏈的運作，方能建立正確的系統需求描述。本研究的第一步驟，即是建立一個足以描述供應鏈行為而且可以進一步產生系統需求描述的參考模型（reference model）。

關於供應鏈的活動，Supply Chain Council 提出了一個泛用的模型，詳細的定義了 Plan、Source、Make 與 Deliver 四者各擔負的功能與彼此之間的關連性。如圖 2-1 所示(來源：Supply Chain Council)，Plan 的活動在上層擔任其餘三者之間的規劃與協調工作；Source 則是物料的準備(採購)者；Make 負責的是實際的生產作業；而 Deliver 則掌控了產品的派送工作。四種活動中都以數字註標表示該項活動中的特別行為，以 Source 為例，S1 代表 Make To Stock 的物料準備活動；S2 是 Make To Order 的物料準備活動；S3 則描述 Engineer To Order 的物料準備活動。

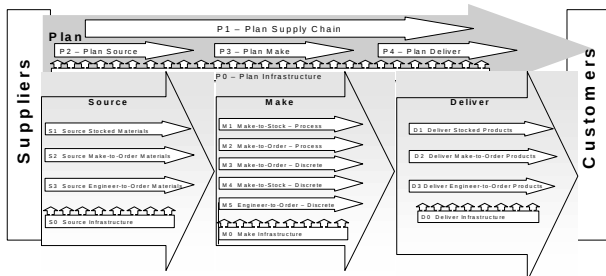


圖 2-1 Supply Chain Council 所定義的供應鏈的 19 種 Process 來源:Supply Chain Council

Supply Chain Council 所提出的模型中，並沒有強調一個企業體在擁有多家子公司時，其子公司之間在供應鏈活動中相互協調的情形。而且該模型基於通用性的考量，在活動細節的描述上相當抽象，並不適合直接引用來當做本研究的參考模型。因此，以 Supply Chain Council 的供應鏈模型為基礎，參考電腦系統製造業的產業結構，同時考慮企業體內有多個子公司時相互協調的運作情形，得到如圖 2-2 的供應鏈模型。

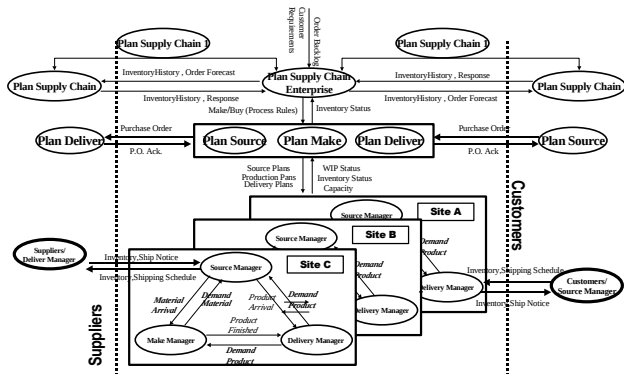


圖 2-2 新的供應鏈參考模型

本模型分為決策、規劃與執行三層，各個層級的功能、責任與單位之間的訊息溝通如下：

決策層 (Plan Supply Chain Enterprise)：以客戶需求 (Customer Requirements)、上下游間的預測訂單 (Forecast Order) 以及物料、貨品庫存記錄 (Inventory History) 等資料為依據，產生企業運作規則 (Process Rules) 供規劃層作為行動的依據。

規劃層 (Plan Source、Make、Deliver)：以上一層產生的運作規則與下層回應的 WIP (Work In Process)、現場產

能 (Capacity) 以及庫存 (inventory status) 等狀況為依規，接收下游訂單 (Purchase Order)、決定物料訂單並產生物料計畫 (Source Plan)、生產計畫 (Production Plan)、產品派送計畫 (Delivery Plans) 等資訊供執行層使用。

執行層 (Manager Source、Make、Deliver)：執行實際的原物料 (Source) 生產、製造 (Make) 與派送 (Delivery) 等工作，並且將 WIP、庫存狀況以及現場產能等資訊傳遞給上層。

本模型中另一個不同於 Supply Chain Council 之處在於考慮到同一企業體內，各個子公司 (Site) 之間的協調問題 (圖 2-3)：同屬的子公司之間供應/被供應的角色關係則是由規劃層來決定的，而且它們之間所傳遞的資訊和企業與企業之間所傳遞的資訊格式是相同的。

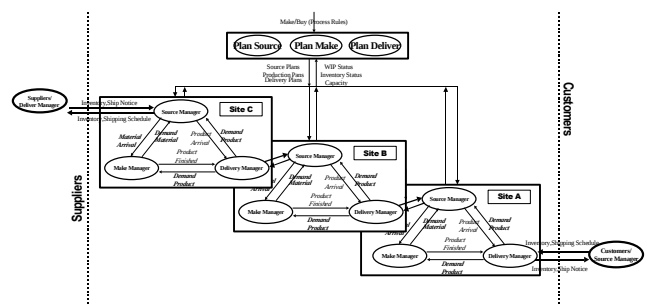


圖 2-3 企業體內部的供應鏈關係

本研究參考此自行發展的供應鏈模型，以該模型的運作情況為藍本，產生欲發展的供應鏈系統之需求模型，並引用物件導向方法，進一步的產生供應鏈系統的物件模型。

3.利用被使用案例 (Use Case) 建構供應鏈系統物件模型

物件導向式的系統開發方式，是從建立系統的需求文件開始，經物件導向分析 (OOA)、物件導向設計 (OOD)、到實作與測試 (Implementation & Testing) 乃至整合 (Integration) 等步驟來產生可以得到符合使用者要求的系統。Ivar Jacobson 在他的 Object-Oriented Software Engineering(1994)書中以使用案例 (Use Case) 來描述系統使用上的典型情節，他使用案例成為了專案發展以及規劃的主要元素，Rational Software 更在 1996 年將使用案例包含進 UML (Unified Modeling Language) 中[8]。目前，使用案例已經成為物件導向式系統開發程序中的重要工具，不論是在系統分析、系統設計以及系統測試時都會運用到它的模型，利用使用案例來描述系統的使用情境，並藉此過濾與辨識出描述系統需要物件以建構系統的物件模型的分析流程，有其優點與特色。特別是在使用者與系統之間互動的描述上，使用案例可以明確的定義出使用者需要的是什麼樣的介面與執行流程，使得開發人員在系統發展過程中不至迷失方向。但是，使用案例描述的重點是系統使用者的使用情境 (Scenario)，對於系統內部執行狀況與流程的描繪則較為不足，從使用案例模型中並不容易看出各個使用案例之間的執行情況與關連性，其架構出來的物件模型不見得與真實世界中的運作情形能夠真正相互的對應。

對於一些與真實世界的實體有相互對應與互動關係如

供應鏈、物流、快遞、倉儲或製造等類型產業的管理或執行系統，本研究試著利用使用案例的技巧但以不同的觀點來描述它們的行為，重點是把使用案例描述的焦點放在真實世界中，系統對哪些物體做了什麼運作上。以被系統處理的物體為主角，利用使用案例紀錄它們與系統間的真實行為，描述它們經過整個系統的過程並建立出這些物體的被使用案例模型。再以這些被使用案例為依歸，經由其中的描述辨識出物件，並組成系統的物件模型。

如同使用案例模型中的三個組成成分一般，被使用案例也是以被使用者、被使用案例與被使用案例圖形三者組成完整的被使用案例模型。參考先前提出的供應鏈參考模型的需求描述，供應鏈環境中需要系統處理的主要實體物質是產品、物料與在製品三者。因此，選擇『產品』、『物料』與『在製品』為本模型的被使用者。再根據供應鏈參考模型的行為描述，可以建立出這三個被使用者的被使用案例與使用情境。整個被使用案例模型的描述如下：

- 被使用者：產品（細分為預測期、規劃期與執行期三者）、物料（細分為預測期、規劃期與執行期三者）與在製品（執行期）。
- 被使用案例：
 1. 與產品（預測期）（Product(Forecast)）有關的被使用案例
 2. 與產品（規劃期）（Product(Plan)）有關的被使用案例
 3. 與產品（執行期）（Product(Execution)）有關的被使用案例
 4. 與物料（預測期）（Material(Forecast)）有關的被使用案例
 5. 與物料（規劃期）（Material(Plan)）有關的被使用案例
 6. 與物料（執行期）（Material(Execution)）有關的被使用案例
 7. 與在製品（執行期）（WIP(Execution)）有關的被使用案例等。
- 被使用案例模型：直接借用 UML 的圖示來描述模型的型態，但符號的定義上有所變動。系統的被使用者（人形符號）與其被使用案例（橢圓形符號）之關係如圖 3-1 所示。其中人形符號與橢圓形符號之間的箭號表示被使用的關係，而橢圓形符號之間的三角形箭號則是被使用案例延續關係之描述。

與使用案例模型相較，被使用案例模型中案例的個數顯然多了許多，但案例的內容卻較為簡短。使用案例所描述的是使用者為了某一個目的使用系統的過程動作，案例與案例的內容之間是彼此獨立的，案例與案例之間的使用前後次序也不是那麼明顯。而每一個被使用案例卻是描述被使用者在某一情況下為系統其他物體所使用的過程，所以往往需要連續幾個被使用案例的連結，才能表達出系統使用者的目的。這種描述技巧的優點之一，就是可以在模型中敘述物體被使用的同時表現出其在系統中相對的狀態。

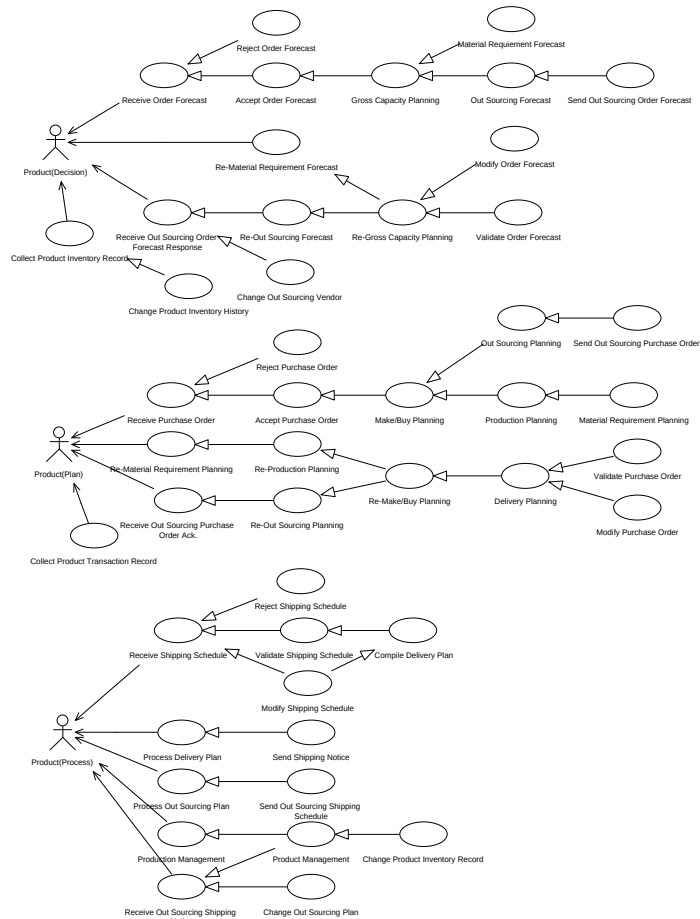


圖 3-1 產品之被使用案例模型

如同對使用案例描述的要求，被使用案例的內容也需具備用字遣詞以及專用名詞的統一、文句簡潔易懂的原則。與使用案例模型不同的是，被使用案例模型中的主角（被使用者）將直接被視為系統中的基本物件，而且可以直接建立這些基本物件之間的物件模型。圖 3-2 即是產品、物料與在製品三者的物件模型，模型的建構原則是以供應鏈管理系統參考模型的行為為依據。

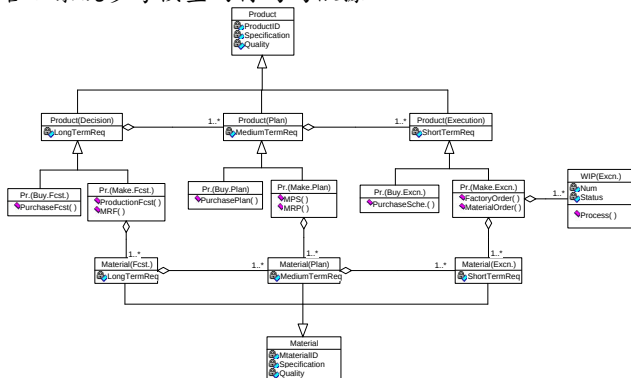


圖 3-2 產品、物料與在製品物件間的關連性

4. 物件模型的總體評估

就四種物件模型的各項物件評估進行彙整後，以從中辨識出那一種建構物件模型的分析方法才是最適用於供

應鏈管理系統的開發。針對各項觀點評估出來的結果，依其優劣的程度給予『4』、『3』、『2』與『1』四種等級，若是在某些項目上有無法明確評估次序的物件模型時，則給予相同的等級數字。表 4-1 即是四種物件模型集中比較的結果。

表 4-1 物件模型總體評估

模型名稱	功能導向式	OOSE	使用案例型	被使用案例
描述能力	1	3	2	4
發展一致性	1	4	2	3
系統彈性	3	4	2	4
重複使用性	1	3	2	4
可靠度與維護性	2	3	1	4
總體表現	8	17	9	19

由上表可知，針對供應鏈管理系統的開發過程，以被使用案例為分析方法所建構的物件模型最能符合系統的需求。而 OOSE 的分析方法所建構的物件模型與被使用案例的模型其實差距不大，在發展一致性的特徵上尤較被使用案例分析方法為佳。也就是說若只考慮如何建構出可執行的系統時，OOSE 的分析方式會是個不錯的選擇。功能導向式的系統開發過程是在有物件導向觀念以前，廣為設計者所使用的分析方法。但是以物件的精神來看，其高度的資料共用性與資料、功能分別處理的特徵，使得此方法的封裝性極差，完全不能表現出物件導向的優點與特性。只較功能導向式物件模型好一點的是以使用案例為基本物件的分析方法，其雖然以使用案例來描述使用者的需求活動。但是該物件模型中操作功能的分割過細，屬性的封裝程度也不佳，亦不合適以之進行物件導向分析的工具。

5. 物件模型實做

為明確的表現出用被使用案例分析方法所建構出來的物件模型在於實際執行時的特性，本研究選擇該物件模型與以 OOSE 分析方法所得的物件模型兩者的決策層進行系統實作的步驟。並利用一個簡單的使用情境，配合程式執行的動作與兩種物件模型中主要物件的互動圖之比較，可以清楚的描述出以被使用案例分析方法得到的物件模型的優點。為將焦點集中於物件的活動上，依據兩種物件模型所實作出的程式將採用簡單而完全一致的使用者介面。以下即是程式的執行過程：

系統接收預測訂單：系統接到預測訂單之後，會通知系統使用者準備處理（圖 5-1）。使用者按下『管理預測訂單』之後系統顯示『管理預測訂單對話框』。使用者按下『彙整預測訂單』之後，對話框內顯示出剛接收到的七張預測訂單。系統使用者可以點選任意一張預測訂單，並按下『檢視』確認過所有訂單之後，可以按下『產品需求預測』。使用者按下『總體生產規劃』，使用者按下『產-購預測』，將產品需求數量依據企業運作原則區分為自產與外購兩類。使用者按下『生產排程預測』，系統進行自產類產品的生產排程預測並顯示結果。使用者按下『物料需求預測』，系統進行自產類產品的物料需求排程預測並顯示結果。使用者按下『採購預測管理』，系統顯示對話框。使用者進行物料採購預測，系統顯示物料訂單資料。使用者按下選擇供應商，系統出現如圖 5-2 的畫面，供使用者

進行物料供應商的選擇。在需求的物料皆指定供應商之後，即可將物料預測訂單送出。（外購產品的處理情況與物料相同，故不再贅述）。

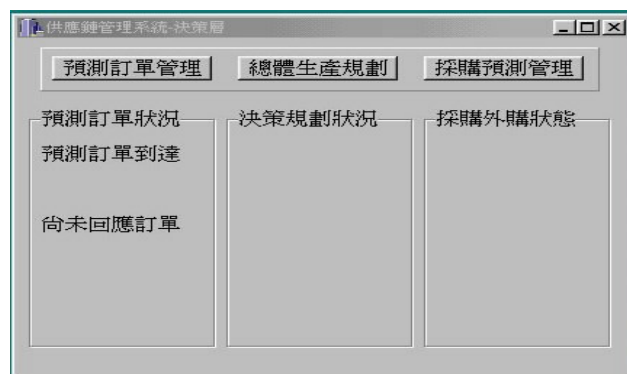


圖 5-1 供應鏈管理系統-決策層

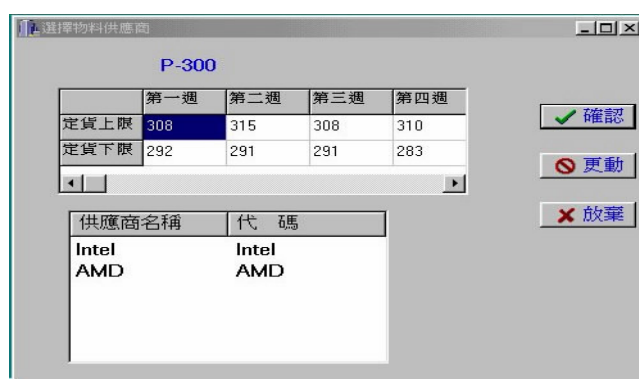


圖 5-2 選擇物料供應商

6. 供應鏈系統整合

為了讓整個供應鏈系統的整合更加容易，本研究採用 ObjectStore 的物件導向資料庫為資料管理的工具。於是在應用分散式環境與物件導向資料庫的情況下，本研究架構採圖 6-1 的架構。

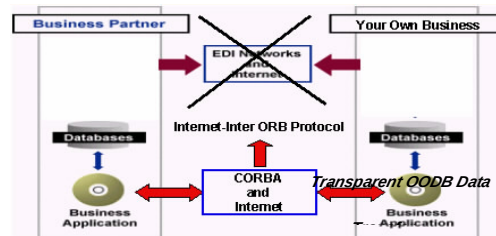


圖 6-1 分散式物件環境下企業間資料交換的架構

由於 EDI 是目前業界大多採用的資料交換方法，為了不與之抵觸並且要包含在之前提出的供應鏈模型中，類別（Class）間的 Generalization（is a kind of）和 Aggregation（is a part of）的關係成為劃分 EDI 類別（Class）的主要原則。以 Compaq（康柏克電腦）為例，它分別用到了 ASC X12 830（Planning Schedule to Supplier），846（Inventory Advice from Supplier），850（Purchase Order to Supplier），855（Purchase Order Acknowledgement from Supplier），856（Ship Notice from Supplier），862（Shipping Schedule to

Supplier)和 870(Order Status Report <Commit Signal> from Supplier)。一般直覺可能就讓 830, 846, 850...分別成為一個 Class,但是我們輕易發現如此之 Class 間的資料重複性非常大,以 850 與 855 為例,二者的內容相差僅三個 Attribute,如圖 6-2 所示。

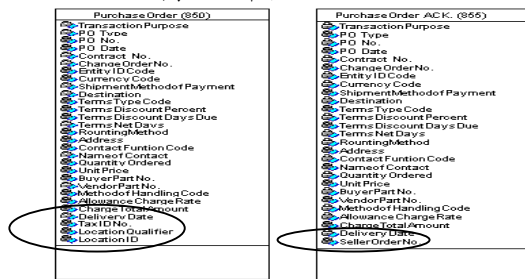


圖 6-2 ASC X12 850 與 855 傳輸資料內容的比較

另外在 Computing Industry 的 ASC X12 EDI 內容中發現每一個 Protocol 格式都有共同的 Control Header,因此可以將這個部分抽出成為 Super Class 讓所有其它的 Class 繼承。如圖 6-3 所示。

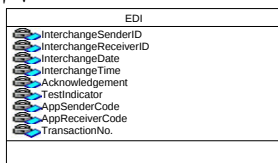


圖 6-3 EDI 的 Super Class

既然圖 6-2 所示的二個 Class 幾乎是一樣的,於是可以再將原本一大個 Class 分為幾個小 Class,再由這些分出來的 Class Aggregate 起來成為原本需求的"二"個(850 與 855) Class,這樣就已經達到減少資料重複性的目的(如圖 6-3 所示)。而且分出的 Class 並非只被 850 和 855 用到而已,事實上 NameInfo、Address 和 Contact 這三個 Class (圖 6-2 中圈起者)在其它的 Protocol 中都有被共用到,同時這些 Classes 也分別化為有意義的實體物件,此時已經漸漸顯現物件導向分析的好處。

有了與 EDI 相容之資料交換物件後,再配合 Used Case 所找出來 Object Model,可以將 EDI 類別輕易地融入物件模型中得到一個所謂供應鏈的 EDI 相容物件模型,由於完整的模型相當複雜且龐大,所以以圖 6-3 為例說明:圖中的 P.O. (850) 對應至 Prch.Odr.,而 P.O.ACK (855) 對應至 Prch.Odr.Ack。最後我們可以得到一個相較於前面提出之參考模型更加整合且與 EDI 相容的供應鏈系統整合模型,如圖 6-4 所示。

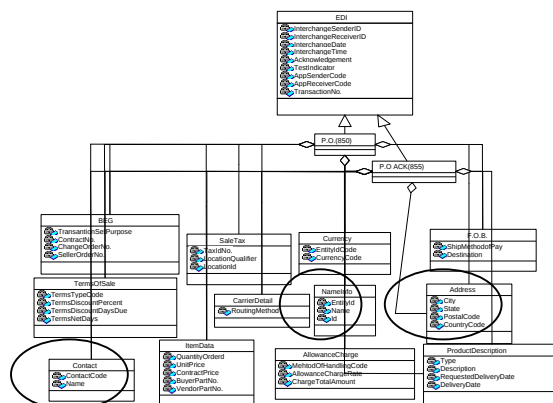


圖 6-3 ASC X12 850 與 855 共用的類別

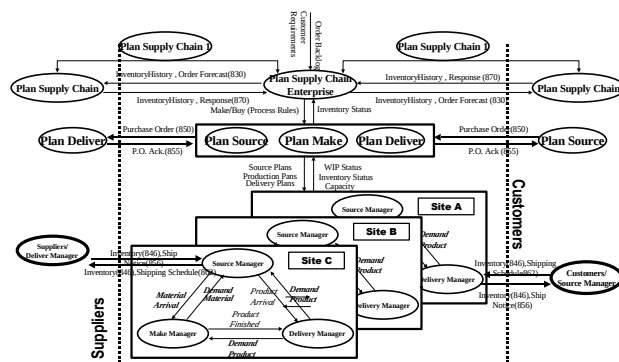


圖 6-4 EDI - Compliant Integration Model

參考資料

- [1] Tom Gormley and Bobby Cameron, "Supply-Chain Insights -- Multivendor Portfolios", Manufacturing Systems, January 1999, pp. 20.
- [2] Alan Pope, December 1997, The CORBA Reference Guide - Understanding the Common Object Request Broker Architecture.
- [3] Yen-Ping Shan, Ralph H.Earle, December 1997, Enterprise Computing With Objects - From Client/Server Environment To The Internet.
- [4] Sidney Hill, March 1996, "The Virtual Corporation-Electronic commerce ties far-flung business partners together, Manufacturing Systems", pp.38-40.
- [5] Donald Davis, August 1995, "Evolution In EDI-Vendors offer packaged solutions and better integration with MRPII, Manufacturing Systems", pp.22-28.
- [6] Ivar Jacobson, 1992, Object-Oriented Software Engineering.
- [7] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorensen, 1991, Object-Oriented Modeling and Design.
- [8] Hans-Erik Eriksson, Magnus Penker, 1998, UML Toolkit.