

# 跨國企業之 ERP 會計系統架構分析--以 SAP R/3 系統為例

An Architectural Analysis for Multinational Firms' ERP Accounting Modules  
-- A Virtual Case Study of SAP R/3 Systems

周國華

屏東商業技術學院會計系講師

## 摘要

本文根據分散式系統基本原理對 SAP R/3 系統的會計模組進行架構分析，並建構一個虛擬的跨國性分散式情境，探討如何讓 SAP R/3 系統維持一個適當的軟硬體及網路架構，以取得良好的分散式計算績效。虛擬個案公司為一家東亞區域型的電子製造業公司，其製造重心在中國大陸及東南亞，銷售通路則遍及東亞各國；此外，除台北為集團總部外，另在上海（下轄北京、香港兩個據點）、新加坡（下轄吉隆坡、曼谷兩個據點）、東京（下轄大阪、漢城兩個據點）設立區域總部。為符合各國法令規定，個案公司在各國均設有具法人身份的獨立子公司，部份城市設有該國子公司之分公司。此一個案之探討結論如下：

(1) 台北總部與三個區域總部之間的參考架構是聯邦式系統或標準分散式系統的架構。

(2) 三個區域總部與其轄下據點之間的參考架構是標準的主/從式系統架構。

(3) 前述的討論係以兩個端點之間的關係來探討，比較著重中央端點與本地端自主性的關係。若從全區資料庫查詢的觀點來看，也可以採用多元化系統做為參考架構。

關鍵字：企業資源規劃；會計模組；分散式系統；自主性；透通性

**ABSTRACT:** The technical structure of multinational companies' ERP accounting modules involves multi-tier structures of software and hardware, and the related network structures. The paper gives a well-rounded but concise discussion of them all. Then, for applying the theories and techniques on the implementation of distributed ERP systems, the paper gives an architectural analysis of a virtual implementation of SAP R/3 system. The virtual case study shows that:

1. The reference architectures between headquarter and the three regional headquarters are federated systems or standard distributed systems.

2. The reference architectures between the three regional headquarters and their subordinate sites are standard client/server systems

3. From a database query standpoint, the multi-systems could be another reference architecture.

**KEYWORDS:** ERP; Accounting Modules; Distributed Systems; Autonomy; Transparency

## 壹、前言

會計資訊系統通常位居企業資訊系統的核心，且隨著電腦軟硬體在功能上不斷的強化，及網路技術的進展，採用多層分散式架構的企業資源規劃系統會計模組(ERP Accounting Modules)已成為企業會計系統的主流。由於跨國企業營業及生產據點散佈在不同國家或地區，要建構一套能有效管控所有據點的整合性 ERP 會計系統，不論在軟體系統、網路架構或組織管理上都是一項重大挑戰。本文針對分散式會計資訊系統之理論基礎及技術架構作深入的探討，並從分散式資料庫理論中找出建立分散式 ERP 會計系統的參考架構(Reference Architecture)，以了解在開發一個適切的分散式 ERP 會計系統時，應該注意的決策思考重點及資訊技術環節。

完整的分散式會計系統包含使用者端、會計應用程式伺服器端、會計資料庫端等三個部份，其理論基礎及技術架構都有相當程度的複雜性。對於跨國企業的會計長(Controller)或財務副總裁(CFO)而言，如何透過分散式會計系統來讓散佈各地的會計部門可以有有效的執行其本地端(Local site)的職能，又可分享區域總部或全球總部的資源，並讓總部可以在不過分干擾各本地端職權下取得適當的區域或全球控制權，是個重要的會計系統設計議題。

跨國企業在導入 ERP 會計系統時，須為位處不同國家及地區的公司或分公司建立合乎所在地法規又兼顧企業整體管控需要的會計模組。在財務會計部分，通常牽涉不同記帳幣別、會計科目表及跨國合併報表的處理；在成本管理會計部分，則牽涉到不同的成本流程及責任中心的設定。SAP R/3 系統的財務模組中，財務會計(FI)及成本管理會計(CO)兩項會計子模組對上述問題提供了完善的解決方案。FI 及 CO 的會計資訊系統組織結構中有許多組織單元，透過不同層次組織單元之多樣性結合，經理人員即可用 SAP R/3 的會計模組控管跨國經營的大型企業。本文對此將有詳細說明。

本文後續內容如下：第貳節為理論背景，探討分

散式會計系統的相關理論；第參節為技術架構分析，探討分散式會計系統必須面對的軟、硬體及網路技術問題；第肆節為 SAP R/3 系統介紹，簡要說明 SAP R/3 系統的主從架構(client/server)技術，及其會計相關模組的內容；第伍節為實作架構分析，探討在一個虛擬的跨國性分散式情境下，如何讓 SAP R/3 會計系統維持一個適當的軟硬體及網路架構，以取得良好的分散式計算績效；第陸節為結論。

## 貳、理論背景

### 一、ANSI/SPARC 架構及其衍生

分散式會計系統的原理主要源於資料庫理論。自 1970 年代開始，陸續有文獻及專書探討資料庫標準化的相關議題，首先是美國國家標準局(American National Standards Institute, ANSI)的電腦及資訊處理委員會(Computer and Information Processing Committee, X3)，於 1972 年在其標準規劃及需求委員會(Standards Planning and Requirements Committee, SPARC)下成立了一個資料庫管理系統的研究小組，致力於制定出資料庫管理系統的標準，該小組於 1975 年發表期中報告(SPARC, 1975)，並於 1977 年完成最後報告(Tsichritzis and Klug, 1978)，在這兩份報告中所提出的架構，全名為“ANSI/X3/SPARC DBMS Framework”，日後則以“ANSI/SPARC architecture”為其流通名稱(以下稱為 ANSI/SPARC 架構)。ANSI/SPARC 架構係以資料為中心，分析出資料庫的外部綱要(External Schema, 使用者端)、概念綱要(Conceptual Schema, 企業邏輯端)及內部綱要(Internal Schema, 實體端)等三個構面，並建構其間視界(views)的關係，圖 1 顯示 ANSI/SPARC 架構的概要(請參閱附錄圖 1)。

ANSI/SPARC 架構在直覺上及邏輯上都獲得資訊系統專業界的認可，並引發許多衍生的研究。由於該架構並不是針對分散式資料庫系統所制定，許多資料庫學者遂嘗試將 ANSI/SPARC 架構衍生至分散式環境，舉例而言，Mohan and Yeh(1978)嘗試作出一個簡單的分散式 ANSI/SPARC 架構，Schreiber(1977)則建

構出一個複雜的分散式 ANSI/SPARC 架構，包含了資料模式中的異質性在內。

## 二、C. J. Date 的十二項規則

Date 在 1987 年發表了建立分散式資料庫的十二項規則(12 rules)<sup>1</sup>，目前已成為探討分散式系統的重要理論依據。本文以會計情境闡述其內容如下：

*規則 0：對使用者而言，分散式系統應該看起來與非分散式系統無差異*

本規則意指一個架構良好的分散式會計系統，應該讓使用者感受不到它是個分散式系統(例如：不會出現系統回應速度過慢、指令不同等現象)，而以為是在使用本地端的集中式系統；換言之，分散式系統應提供良好的系統透通性(Transparency)。以資料庫的運算而言，SQL 中的 SELECT、INSERT、UPDATE、DELETE 等指令的執行在分散式環境下不應有差異。此項規則又稱為基本原則(fundamental principle)，後續的十二項規則皆導源於此項原則。

*會計釋例：*甲公司全球總部設在台北，並在上海設有子公司，香港及北京設有分公司，以上海子公司為大陸總部；另在曼谷、吉隆坡及新加坡設有子公司，以新加坡子公司為東南亞總部。無論會計系統之應用程式或資料庫如何配置，其散佈在各地之會計部門在透過分散式資訊系統執行會計職能時，就像在使用一個集中式系統一樣方便、快速。

*規則 1：本地端自主性(Local autonomy)*

本地端自主性意指會計資訊系統上任何一個端點(site)之運作均由該端點自行控制，任何端點均不需依賴其他端點才能成功運作；因此，其他端點發生當機或暫停運作時，本地端不受影響，仍能照常運作。此外，此規則也意涵本地資料由本地擁有及管理，並承擔責任(local accountability)。

由於資料複製及網路傳輸上的限制，完全的本地端自主性並不可行，因此，Date 認為此規則可調整為

「本地端應儘可能做到最大程度的自主性」。此外，會計資訊系統的特徵之一是強調內部控制，從內控的角度思考，完全的本地端自主型也不符合會計系統運作的原理。

*會計釋例：*甲公司分散在各地的會計部門獲得充分授權可透過資訊系統自行完成本地端的會計交易處理，即使全球或區域總部的系統當機了，各地的會計部門仍可繼續運作。但為了內部控制之需，若干交易仍需獲得全球或區域總部的系統授權，並存取總部資料庫之資料。

*規則 2：不依靠中央端點(No reliance on a central site)*

本規則與「本地端自主性」係相輔相成的。一個依賴中央端點才能運作的會計資訊系統至少有兩項缺點：1. 容易在中央端點形成系統運作上的瓶頸；2. 一旦中央端點當機，整個系統就會癱瘓。

*會計釋例：*甲公司分散在各地的會計部門大部份均有可獨立運作的應用程式，且除了少部分資料集中在台北總部外，其他資料均儲存在各城市的本地端或區域總部之資料庫內。

*規則 3：連續性運作(Continuous operation)*

維持系統連續運作不中斷，是集中式系統與分散式系統都必須做到的，但在分散式系統下有不同的技術考量。本規則要求分散式系統在增加新端點或為任何端點進行軟體升級(如 DBMS 版本更新)時，整個系統仍能運作如常。

*會計釋例：*甲公司增加東京、漢城及大阪三個營業據點，並以東京為東北亞區域總部。在進行系統跨國連線時，透過備援系統之運作，甲公司全球網路上之其他端點並不需要暫停，仍可照常運作。

*規則 4：位置獨立性(Location independence)*

位置獨立性又稱為位置透通性(Location transparency)，意指使用者不必知道資料儲存在何處，即使資料是儲存在其他端點，在操作時所有資料就像存在本地端一樣。這可以簡化使用者端的程式及端點上的作業，且當資料在不同端點之間移動時，不會對端點上的程式及作業造成影響。

<sup>1</sup> C. J. Date 在其第七版的資料庫系統專書(Date, 2000)中，已經把沿用多年的「12 rules」改稱為「12 objectives」，但其內容意旨均相同。由於絕大多數資料庫文獻在引用相關概念時均慣稱「12 rules」，故本文仍以「12 rules」稱之。

*會計釋例*：甲公司大陸地區的部分營業資料集中儲存在上海總部之資料庫內，除了少部分需獲得額外的系統授權外，其餘資料是開放性的。北京或香港分公司之會計人員在存取這些儲存在上海總部資料庫內之開放性資料時，並不知道它們是存在上海總部，而以為是在本地端的系統內。

*規則 5：分割獨立性 (Fragmentation independence)*

為了提昇資料庫績效，許多系統都支援資料分割 (data fragmentation) 的功能；所謂資料分割，係將資料庫中的關連資料表 (relations) 作水平或垂直的分割。垂直分割是把資料表中的欄位切割成不同區塊，由不同部門進行新增、刪除、修改等維護作業，但使用時則以包含全部欄位的整筆值組 (tuples) 為主；水平分割則是由分散在各處的端點進行包含全部欄位的整筆值組之新增、刪除、修改等維護作業，且該筆資料就近儲存在產生資料的端點。

規則 5 強調一個支援資料分割的資訊系統，也必須支援分割獨立性 (又稱為分割透通性 (Fragmentation transparency))，亦即，讓使用者感受不到資料被分割。當使用者要查詢實體上被分割的資料時，使用的是一般的查詢指令，系統的後端則透過資料庫代數的合併 (join) 及聯集 (union) 等運算來將分割的資料組合起來。分割獨立性可以簡化使用者端的程式及端點上的作業，並且可讓資料在必要時進行再分割，以便進一步提昇系統績效。

*會計釋例*：甲公司的上海子公司下轄香港及北京兩個分公司，三地之新增交易資料均就近儲存在本地端，以舒緩網路傳輸之擁塞情形。但三地為一子公司所轄，在面對客戶或稅務機關詢問時，常需以整個大陸地區之統合資料為回應之依據，故會計系統設計成實體資料儲存在本地端，但會計人員在查詢時則呈現出整個大陸地區之統合資料 (水平分割之透通性)。

又，甲公司的客戶主檔包含許多欄位，部分欄位與銷售作業有關，由營業部門負責維護；部分欄位與信用查核有關，由財務部門負責維護；部分欄位與會計處理有關，由會計部門負責維護。查詢客戶資料時，則可全攬整個欄位 (垂直分割之透通性)。

*規則 6：複製獨立性 (Replication independence)*

資料複製也是提昇分散式系統績效的重要方法，若能本地端複製一份儲存在其他端點的資料，則不但可就近取得資料、免除網路擁塞的困擾，也增加了資料的可得性 (availability)。但複製會增加資料更新時的作業量，且必須確保「每一份」複製的資料都更新了。

支援複製獨立性 (又稱為複製透通性 (Replication transparency)) 的系統可讓使用者察覺不到資料是否被複製了，這種特性可以簡化使用者端的程式及端點上的作業，且可配合需求的改變，隨時進行資料複製及銷毀的工作，而不會影響使用者端的程式或作業。

*會計釋例*：甲公司大陸地區的客戶主檔由各分支機構負責建置與維護，由於客戶也有許多分支機構散佈大陸各地，上海、香港、北京三地複製彼此的客戶資料，以就近服務客戶、舒緩網路傳輸之擁塞情形。此複製係由系統後端自動進行，使用者 (公司職員) 及客戶均不知悉，因此，當總部設在上海的客戶因信用額度用罄而無法再賒購商品時，即使轉向北京分公司賒購也會被拒絕。若因系統建置不良，使資料更新有明顯的時差現象，讓使用者或客戶知悉部分資料係複製自其他端點，則在上海被拒絕信用交易的客戶即可能在北京獲得信用交易額度。

*規則 7：分散式查詢處理 (Distributed query processing)*

在分散式系統下，資料散佈在不同端點，各端點之軟、硬體效能也有一定程度差異。在進行查詢 (query) 時，需根據各端點之軟硬體效能來決定在哪一個端點進行資料的合併 (join) 作業，以便取得較佳的運算績效，達成分散式查詢的最適化 (optimization)。

*會計釋例*：甲公司總裁赴曼谷考察業務，發現部分客戶的應收帳款金額龐大，信用堪虞，於是透過曼谷分公司的資訊系統進行東南亞地區的應收帳款金額查詢，以了解應收帳款在一定金額以上的客戶有多少。此查詢需合併曼谷、吉隆坡、新加坡的客戶資料進行運算，因為是透過曼谷的系統進行查詢，此時曼

谷的系統就扮演分散式資料管理者(Distributed Data Manager, DDM)的角色，透過全區資料庫字典(Global database dictionary)的分析，決定由系統效能較強、客戶資料較多的新加坡端點來進行三地資料之合併運算，再將結果傳回曼谷端點。

#### 規則 8：分散式交易管理(Distributed transaction management)

交易管理是確保資料在系統中的異動過程能被正確的處理，以免產生不正確資料而導致錯誤的決策。交易須維持單元性(Atomicity)、一致性(Consistency)、隔離性(Isolation)、持續性(Durability)等四個特性(簡稱為ACID)，此四個特性是靠交易管理中的回復控制(Recovery Control)與並行控制(Concurrency Control)兩大機制來完成。在分散式系統下，回復控制與並行控制都需要做額外的處理。在回復控制部分，可透過兩階段委任協定(two-phase commit protocol)來完成；至於並行控制部分，則藉由鎖定法(Locking)來完成。

會計釋例：甲公司的東北亞區域各公司由於剛設立，若干會計處理模式暫時沿用台北總部的通用方法。在取得經營上的經驗值後，東京子公司負責籌編東北亞區域三地通用的會計系統基本資料，並透過網路系統同時進行東京、大阪、漢城之資料更新動作。在更新過程中透過兩階段委任協定，在所有端點皆更新完成後，同時下達委任(Commit)指令，系統即改按新的經驗值運作；若因網路中斷或系統當機，導致若干端點未完成更新，則在網路重新連線或系統回復運作後，同時下達撤回(Rollback)指令，將系統回復到未執行更新前的原點，再重新根據兩階段委任協定來進行更新。

#### 規則 9：硬體獨立性(Hardware independence)

分散式系統各端點散佈在不同地理區域，很可能使用不同廠牌的硬體設備，會計資訊系統的軟體部分須盡量支援多廠牌的硬體設備，以讓各端點的使用者有使用同一系統的感受(single-system image)，俾達成分散式系統的透通性要求。

會計釋例：甲公司各端點的硬體設備為了就近維修方便，均採用當地知名廠商生產或代理的產品，如

大陸地區採用聯想電腦產品，台灣地區採用宏碁電腦產品，日本地區採用東芝公司產品等。採用的原則是這些硬體設備能夠承擔甲公司 ERP 系統(如本文後述之 SAP R/3 系統)的運作需求。

#### 規則 10：作業系統獨立性(Operating system independence)

分散式會計系統的應用程式及資料庫管理系統都必須在作業系統上執行，而不同的硬體可能搭配不同的作業系統，如 IBM 大型主機有其特定的作業系統，中小型伺服器與工作站電腦則可搭配各種廠牌的 UNIX 系統，PC 架構的電腦則可搭配 Windows NT、Windows 2000、Windows XP 等。為了能在不同的作業系統上執行，分散式系統的軟體程式需盡量支援不同的作業系統平台，所以需開發出不同平台之版本。

會計釋例：甲公司採用 SAP R/3 系統為其企業資源規劃系統，該系統支援多種主要的作業系統。台北總部的資料庫伺服器及應用程式伺服器採用 IBM 大型主機，各區域總部的資料庫伺服器及應用程式伺服器採用昇陽(Sun Microsystems)的主機，其他端點則採用 PC 架構的電腦；這些硬體分別搭配 IBM 大型主機專用作業系統、昇陽的 Solaris 作業系統、微軟的 Windows NT workstation、Windows 2000 server、Windows 2000 professional、Windows XP professional 等作業系統，SAP R/3 系統均可在這些作業系統上執行。

#### 規則 11：網路獨立性(Network independence)

只要能夠支援各種不同的網路協定(如 OSI，TCP/IP，IEEE)，分散式系統即可在各式各樣的網路架構及網路作業平台(如 LAN，WAN，VPN，Extranet，Intranet，Netware，Unix，Windows NT)上運作。

會計釋例：甲公司全球各營業據點均架設一個區域網路(LAN)，台北總部與各區域總部之間則藉由專線聯成一個廣域網路(WAN)，各區域總部與所轄子公司或分公司之間則藉由專線或 Internet 聯成廣域網路或虛擬網路(VPN)；各營業據點與重要供應商及客戶之間並聯成 Extranet。甲公司之 ERP 系統架設在此網路架構下，形成一個全球化的企業資訊系統。

規則 12：資料庫管理系統獨立性 (DBMS independence)

由於成本、功能、技術支援上的考量，分散式系統的不同端點可能需配置不同的資料庫管理系統，為了讓這些異質性的系統可以彼此互動 (interoperability)，需透過標準化的中介軟體 (middleware) 或閘道技術 (gateway) 來達成。

會計釋例：甲公司台北總部配合 IBM 大型主機採用 DB2 資料庫系統，在各區域總部採用 Oracle 資料庫系統，其他資料庫伺服器端點則採用 Sybase 或微軟的 SQL Server。甲公司的 SAP R/3 系統透過資料庫中介軟體 (ODBC、JDBC) 及物件中介軟體 (如 CORBA、DCOM、COM+)，讓不同端點之間的資料庫可以彼此溝通。

### 三、Özsu and Valduriez 的分散式資料庫架構

Özsu and Valduriez (1999) 以自主性 (Autonomy)、分散性 (Distribution)、異質性 (Heterogeneity) 等三個維度架構出分散式系統的不同組合。其中，自主性是指控制權的分散程度，分散性是指資料的分散程度，異質性是指硬體、網路協定、資料模式、查詢語言、交易管理協定等之差異性。

學者對分散式系統的自主性有不同定義，Gligor and Popescu-Zeletin (1986) 認為自主性應做到下列三點：

1. 本地端資料庫系統之運作，不受參與多元資料庫 (multidatabase) 系統之影響。

2. 個別資料庫系統處理查詢及最佳化的動作，不受需存取多個資料庫系統之全區查詢 (global queries) 所影響。

3. 個別資料庫系統加入或離開多元資料庫聯盟 (confederation)，不會影響系統的一致性或作業。

Du and Elemagarmid (1989) 指出自主性的三個層面：

1. 設計的自主性：個別資料庫系統可自由選用合適的資料模式及交易管理技術。

2. 溝通的自主性：個別資料庫系統可自由決定要提供何種資訊給其他的資料庫系統或應用程式。

3. 執行的自主性：個別資料庫系統可自行決定以何種方式來執行交易事項。

Özsu and Valduriez 則把自主性按程度區分為三種等級：

A0：高度整合 (tight integration)。個別系統之間有密切的資料分享，整個系統所提供的畫面是一致的，不同端點的使用者會覺得所有的資料似乎集中在一個單一資料庫內。

A1：半自主 (semiautonomous)。本地端的系統通常可獨立完成作業，但同時加入資料庫聯盟進行資料分享。由於進行資料分享需接受共通性的系統調校，因此喪失了一部分自主性。

A2：完全獨立 (total isolation)。個別系統是單獨存在的 (stand-alone)，不知道有其他系統，更不知道如何與其他系統溝通。

在分散性部分，Özsu and Valduriez 也按分散程度把分散性區分為三種等級：

D0：完全集中。只有一台電腦，無分散與否的問題。

D1：主從架構 (client/server) 分散。資訊系統分成伺服器 (servers) 與使用者端 (clients)，伺服器管理資料管理職能，使用者端則提供應用程式及使用者介面。

D2：主對主架構 (peer-to-peer)。完全分散，每部電腦均具備完整的功能，包含使用者介面、應用程式與資料庫系統，雖然彼此間進行資料分享，但無主從之別。

在異質性部分，Özsu and Valduriez 則將之區分為同質與異質兩種等級：

H0：同質性。

H1：異質性。資訊系統各端點在硬體、網路協定、資料模式、查詢語言、交易管理協定等方面均可能產生差異性，Özsu and Valduriez 特別強調在資料模式與查詢語言上的差異是異質性的真正來源。

以上由自主性、分散性、異質性所構成的三個維度，可以建構出分散式系統的多種可能組合 (請參閱附錄圖 2)。

三個維度所構成的各種組合，從會計資訊系統觀點，有實作上的特定涵義<sup>2</sup>。若以(Ax, Dy, Hz)做為各種組合之代號(x、y、z為各維度之程度代號)，則(A1, Dy, Hz)通稱為聯邦式系統(federated systems)<sup>3</sup>，(A2, Dy, Hz)通稱為多元化系統(multi-systems)，(Ax, D1, Hy)通稱為主從架構系統，(Ax, D2, Hy)通稱為主對主架構系統。三個維度總共可以構成十八種組合，但部分組合並無實作上之意義。(各種組合之簡要說明及會計涵義，請參見附錄表 1)。

高度整合的主從架構系統(A0, D1, Hy)與主對主架構系統(A0, D2, Hy)是通稱的分散式系統，在資料庫架構上，係將 ANSI/SPARC 架構中的概念層拆成「本地概念綱要(Local Conceptual Schema, LCS)及視界」和「全區概念綱要(Global Conceptual Schema, GCS)及視界」兩層，且本地概念綱要及本地內部綱要隨著系統的分散而有不同內涵，使用者端的外部綱要也隨之形成多種版本。

分散式多元化系統(A2, D1, Hz)與(A2, D2, Hz)擁有最高的自主性，絕大部分情況下係自給自足，不需要與其他系統進行資料分享。即使在極少數情形下要與其他系統分享資料時，個別系統仍然享有絕大的自主性，故此系統在分散式架構上與前述的分散式系統有所不同。二者最大的差異是在全區概念綱要(GCS)上，分散式系統的 GCS 定義整個資料庫系統的概念視界，而分散式多元化系統的 GCS 僅是個別系統願意讓其他系統分享之部分的集合；換言之，在分散式系統下，全區系統是個別系統的聯集(union)，但在分散式多元化系統下，全區系統則僅是前述聯集的部分集合。此外，在系統實作上，分散式系統係採取由全區對映至本地的 top-down mapping 方式；分散式多元化系統則是採取由本地對映至全區的 bottom-up mapping 方式。

#### 四、多元化系統與聯邦式系統的相關文獻

在資料庫領域裡探討多元化系統的文獻甚多，對於前述的 GCS 之定義也各有不同。若分散式多元化之組合包含異質性在內，亦即表 1 中(A2, D1, H1)及(A2, D2, H1)兩個項目，則在實作上有兩種選擇：個別語法(unilingual)多元化系統及複合語法(multilingual)多元化系統。在個別語法系統下，認為多元化系統之使用者大部分不需要存取全區資料庫<sup>4</sup>，故使用者在存取本地端資料庫與全區資料庫時，必須用不同的資料模式與查詢語言，因此而有不同的外部綱要——存取本地端資料庫用本地外部綱要(LES)，存取全區資料庫則用全區外部綱要。在資料庫文獻中，Landers and Rosenberg(1982)及 Smith et al. (1981)之 MULTIBASE 系統、Templeton et al. (1987)的 Mermaid 和 Dwyer et al. (1986)的 DDTs 等都屬於此種個別語法多元化架構。

至於複合語法系統下，則假設所有人都會存取全區資料庫，故外部綱要在設計上不分本地或全區，而是完全以本地端的查詢語言來進行即可。對使用者而言，此種方法比個別語法系統方便；但在系統設計上，複合語法卻比個別語法複雜。Ferrier and Stanfret(1982)的 Sirius-Delta 系統及 Cardenas(1987)的 HD-DBMS 專案即屬於此種系統。

Litwin(1988)認為多元化系統既然擁有最大的自主性，就根本不需要有 GCS。因為不需要 GCS，所以比分散式系統在設計上更簡便，與原始的 ANSI/SPARC 架構一樣有三層，原先由 GCS 承擔的對映工作，轉交由外部綱要與本地概念綱要直接進行對映。此時，為了完成存取多個資料庫系統的工作，外部綱要可能須植基於多個本地概念綱要之上。

前述的聯邦式系統(A1, Dy, Hz)也不使用 GCS。在 Heimbigner and McLeod(1985)所設計的聯邦式系統中，每個系統定義一份輸出綱要(export schema)，載明該系統願意與其他系統分享的項目；全區資料庫，則是所有個別系統輸出綱要之聯集。當個別系統要存取全區資料庫時，則透過一份共同性的輸入綱要(import schema)來完成，此輸入綱要之功能與多元化

<sup>2</sup>各種組合在會計系統實作上之涵義，將於第肆節「虛擬個案分析」作深入探討。

<sup>3</sup>聯邦式系統(federated systems)，指個別系統在多數情況下可自行完成作業，但少部分情況須與其他系統進行資料分享才能完成作業，有如政治上聯邦體制之運作，故得此名。

<sup>4</sup>全區資料庫意指資料來自分散式系統中其他的資料庫系統。

系統下之全區外部網要相同。

### 叁、SAP R/3 系統介紹

以上所論及的分散式系統理論背景，是實作分散式 ERP 會計系統不可或缺的基礎。本文第肆節中，將以 ERP 系統中市場佔有率最高的 SAP R/3 系統為例，來進行分散式 ERP 會計系統的實作架構分析，以說明分散式系統之理論基礎及技術架構如何應用在 ERP 系統的實作上。由於 SAP R/3 系統的技術層次較為複雜，本節擬先對該系統作簡要之說明，以做為第肆節實作分析之基礎。

#### 一、R/3 簡史及基本架構

SAP 是全球前三大的商業軟體公司，它的 R 系列企業資訊系統一直位居全球同類產品市場佔有率的前茅。R/1、R/2 是以大型主機(mainframe)為主的集中式及時系統，R/3 則是主從架構的分散式系統<sup>5</sup>。

R/3 系統的原始構想起源於 IBM 公司在 1987 年所倡議的系統應用程式架構(System Application Architecture, SAA)概念，IBM 想藉著 SAA 架構來實作一套可以跨作業系統平台、擁有圖形使用者介面(GUI)、並整合關聯式資料庫管理系統的應用程式，這些現在看來不難達成的概念，在當時卻是很進步的構想。SAA 後來失敗了，原因是 IBM 不願把漸成市場主流的 UNIX 系統納入架構中，而堅持沿用 IBM 本身的大型主機架構做為發展平台。

SAP 在 1987 年把 SAA 的概念融入 R/3 的發展計畫中，並在次年確定用 UNIX 做為發展平台，這也是原本侷限在自然科學及工程領域中使用的 UNIX 系統第一次大規模的商業應用。R/3 的第一位企業客戶在 1992 年成功上線，此後即逐漸席捲全球主要企業客戶的 ERP 市場(Buck-Emden, 1996)。

R/3 的基本技術架構分為系統平台、基礎層(basis layer)及應用層(application layer)，基礎層主要是由中介軟體構成，藉由基礎層所提供的介面，R/3 應用程式可以跨各種作業平台、資料庫系統及通訊系統。基礎層是用 C 及 C++ 程式語言開發的。

應用層建立在基礎層之上，包括各種應用程式模組，是用 ABAP/4(Advanced Business Application Programming)程式語言開發的。ABAP/4 是 SAP 專用的程式語言，原本屬於第四代程序性語言，近年來增加了許多物件導向語言功能，R/3 的應用程式也因此大量的物件化及元件化。(R/3 的基本技術架構請參考附錄圖 3)。

在基本網路架構上，R/3 強調需在資料庫與應用程式伺服器之間建立一個 LAN 形式的伺服器網路(server network)，此網路與主幹網路分開，並需特別維護以確保系統能持續運作。在應用程式伺服器與使用者端之間，則透過路由器建立主幹網路(backbone 或 access network)，與各區域使用者端的 LAN 連結成一個 WAN。(R/3 的基本網路架構請參考附錄圖 4)。

從圖 4 中可知 SAP R/3 的應用程式伺服器(Application host)通常在兩個以上，這種設計完全符合分散式系統應用程式分散但整合(distributed but integrated)原則，以方便系統進一步擴充，而且可就近設在使用者端附近以符合本地端自主性之需求。要讓這些分散在不同伺服器中的應用程式能整合性運作，SAP 發展出 ALE(Application Link Enabling, 應用程式連結促動)的機制。ALE 包含分散式服務層、應用程式服務層、通訊服務層等三層，分別負責「定義互動訊息機制」、「建立互動訊息機制」及「交換互動訊息機制」之功能。

#### 二、R/3 的會計應用程式模組

SAP R/3 是一個即時性、整合性、分散式的企業資源規劃系統，其完整的應用程式模組總共分為三大部份：

1. 財務(Financials)模組：有財務會計(FI)、成本管理會計(CO)、財務管理(TR)、投資管理(IM)、EC(企業控制)、RE(不動產管理)等主要模組及若干次要模組。

2. 運籌(Logistics)模組：有行銷配銷(SD)、材料管理(MM)、工廠管理(PM)、生產管理(PP)等主要模組及若干次要模組。

3. 人力資源(Human Resources)模組：有人事管理

<sup>5</sup> R/1、R/2、R/3 的 R 均表示及時系統(real-time system)。



(PA)、薪資(PY)、訓練管理(PE)、人員時間管理(PT) 等主要模組及若干次要模組。此模組的規模較小，通常以 HR 來表示。

企業在導入 R/3 系統時，可以從上述的模組中選擇適用的模組逐步導入。會計系統通常是企業資訊系統的核心，所以許多企業也選擇 FI 及 CO 模組作為導入 R/3 系統的起點。FI 及 CO 模組的次模組相當多，企業選擇適用的次模組來導入即可。(有關 FI 及 CO 次模組的內容，請參閱附錄表 2)。

FI 及 CO 的會計資訊系統組織結構中有許多組織單元，透過不同層次組織單元之多樣性結合，經理人員即可用 SAP R/3 的會計模組控管跨國經營的大型企業。這些組織單元中的重要項目說明如下<sup>6</sup>：

\* Client：R/3 系統中最高層次的組織單元，為三碼數字。在每次登錄使用 SAP R/3 系統時，第一件事就是輸入 Client 代碼。R/3 系統會為每個 Client 代碼建立獨立的主檔資料及關聯表格組。一個 Client 通常代表整個 Corporate Group，但在有設立多元式系統(multi system)之必要時，也可以僅代表一個次集團或個別公司。

\* Chart of accounts：會計科目表(COA)。分為 Operational COA(OCO A，供日常營運使用)、Local COA(LCOA，為符合營業所在地法律規定所另設之 COA)、Group COA(GCOA，整個企業集團共用之 COA)三種，集團內的個別公司可視需要設立一至三種 COA，其中 OCOA 為記帳之基礎，其他兩種則為 OCOA 之對映。

\* Company Code：擁有完整帳戶組織以符合外部財務報告規範的最小單位，通常為一家公司或分公司，以四位文數字(alphanumeric)編碼來代表。

\* Company：根據法律需編製完整財務報表的最小獨立單位。一個 Company 可以包含一個或數個 Company Codes，且這些 Company Codes 必須使用相同的 OCOA，但可以使用不同的記帳幣別。

\* Corporate Group：整個企業集團，包含所有需編入合併報表之公司。

\* Subgroup：為編製合併報表所設立的虛擬次集團，先將次集團內的公司報表合併，再將所有次集團的合併報表合併為整個企業集團的合併報表。

\* Business Area：可明確劃分的獨立經營領域，通常是為了編製部門別財務報告用。一個 Company Code 可以有多個 Business Areas，一個 Business Area 也可以跨多個 Company Codes。

\* Functional Area：費用發生之領域，如管理費用、行銷費用、生產費用、研究發展費用等，以利於編製損益表。

\* Controlling Area：為 CO 模組專用的組織單元，一個 Controlling Area 可以包含一個或多個 Company Codes，以讓成本管理會計的功能(如共同成本分攤)可以跨 Company Codes 來進行。同一 Controlling Area 中的 Company Codes 必須使用相同的 OCOA，但可以使用不同的記帳幣別。

\* Operating Concern：為 CO-PA 模組中用來做獲利力分析的組織單元，分析可按產品別、客戶別、國別、通路別來進行。一個 Operating Concern 可以包含一個或多個 Controlling Area。

上述組織單元之結合釋例，請參考附錄圖 5。

## 肆、虛擬個案分析

本節以第貳節中所虛擬的甲公司為例，說明 SAP R/3 系統的分散式實作架構。假設甲公司為一家東亞區域型的電子製造業公司，其製造重心在中國大陸及東南亞，銷售通路則遍及東亞各國；此外，除台北為集團總部外，另在上海、新加坡、東京設立區域總部。為符合各國法令規定，甲公司在各國均設有具法人身份的獨立子公司，部份城市設有該國子公司之分公司。表 3 為甲公司所採用 SAP R/3 系統的虛擬結構圖，以下各小節將以表 3 之內容探討甲公司在分散式系統實作架構上與理論背景之符合情形(請參閱附錄表 3)：

### 一、台北、上海、東京、新加坡之間

台北總部與三個區域總部之間，呈現的是聯邦式系統與標準分散式系統的架構：

<sup>6</sup> FI 及 CO 中的組織單元之英文中譯名稱與其實際意義不大吻合，故此處僅列出原文，以免混淆。

### 1. 台北、上海之間

從上海端模組的配置，可知除了必須將財務運作的結果傳回台北端做整合性管理外<sup>7</sup>，上海端點幾乎可以完全獨立運作。此二端點擁有各自的完整資料庫及應用程式伺服器，資料分享只有在必要時才進行，因此兩個端點之間呈現的是聯邦式或主對主分散式的系統架構。

### 2. 台北、新加坡之間

新加坡端的系統配置與上海端接近，但因為區域生產規模較小，所以所使用的 FI 及 CO 模組比上海端少。其資料庫及應用程式伺服器規模足敷該端點運作之需要，但自主性不若上海端強，故其與台北端之間屬於主對主的系統架構。

### 3. 台北、東京之間

東京端的系統配置比上海端少得多，因為東北亞區域並不負責製造業務。某些功能必須透過 ALE 連回台北使用台北端的模組進行運算，故東京端的功能並不完整，其與台北端之間屬於主對主或主/從分散式的系統架構。

若台北總部與三個區域總部之間使用的是不同的系統(如分別使用 Oracle、J. D. Edwards、Baan、Peoplesoft 等廠牌之 ERP 系統)，則其間的架構就可能是多元化系統或聯邦式系統。這種異質性情況對企業不見得有利，故實際上適用機會不大。

## 二、上海、北京、香港之間

上海、北京、香港三個端點屬於同一個 Company(2000)之下，在成本控管及編製集團合併報表時都以上海端點為中心，故三個端點均採用大陸版的 OCOA，再由上海端對映成台灣版的 GCOA；香港端則因屬於不同經濟體制，故另採用香港版的 LCOA。此區域是甲公司最重要的生產基地，故上海端及香港端分別擁有相當完整的 SAP R/3 模組，其中，上海端擁有 PP 模組及全套的 CO 模組，以肆應整個大中國區域生產管理及成本控制之需要。

上海端和香港端之間屬於主對主的系統架構，至

於北京端點，因為完全透過 ALE 連結上海端進行運算，故與上海端之間屬於標準的主/從式系統架構。

## 三、新加坡、曼谷、吉隆坡之間

此三個端點位於不同國境內，分別是三個獨立的子公司，故均有部份 FI 模組以遵循當地會計準則處理財會事務，但在成本控管及編製集團合併報表時都以新加坡端點為中心。吉隆坡端因距離新加坡端很近，其實際營運決策均以新加坡端為主，故採用新加坡版的 OCOA。此區域是甲公司的次要生產基地，生產規模及產品等級均略遜於大中國區域，在 SAP R/3 系統的配置上也比較精簡，曼谷端及吉隆坡端均需透過 ALE 連上新加坡端做大部份的運算處理，此三個端點之間屬於標準的主/從式系統架構。

## 四、東京、大阪、漢城之間

此區域並無生產任務，故所配置的 R/3 系統較為精簡。漢城端因財務會計處理需符合當地會計準則，故設有部份 FI 模組；大阪端則只是一個使用者端；兩個端點的多數運算功能皆需透過 ALE 連上東京端進行，且東京端也需透過 ALE 連上台北端進行部份運算。此三個端點之間屬於標準的主/從式系統架構。

## 五、全區觀點

前述的討論係以兩個端點之間的關係來探討，比較著重中央端點與本地端自主性的關係。若從全區資料庫查詢的觀點來看，例如，當台北端要查整個集團的應收帳款帳齡資料，或新加坡端想查詢另外兩個次集團的應收帳款帳齡資料，就牽涉到整個系統資料分享、資料複製、全區概念綱要之設定等事項。Date 的十二項規則之核心觀點就是整個分散式系統的透通性，故甲公司的最佳做法是採用 ANCI/SPARC 分散式架構，並且不分端點採用一致的外部綱要。但從會計資訊系統所強調的內部控制及成本效益觀點而言，絕對的透通性並不適合企業經營實務；例如，若部份區域原有之舊系統(legacy system)仍數需求，且多數使用者並不需要進行全區查詢功能，則在異質性架構下，甲公司可以採用多元化系統，或者採用無全區概念綱要的多元化或聯邦式系統。

<sup>7</sup>三個區域總部除了 OCOA 外，均另有以台灣版設定的 GCOA，以供台北總部彙總營運資料及編製合併報表使用。

## 伍、結論

在企業經營範疇日益擴大下，採用分散式會計系統做為經營管理的核心工具已成為必要選擇。本文探討了分散式會計系統的理論基礎及技術架構，並用 SAP R/3 系統進行實作架構分析。分散式會計系統的基礎理論是建立在分散式資料庫理論上，根據理論上的探討，分散式系統可以分為標準分散式系統(主對主與主/從架構)、聯邦式分散式系統及多元化分散式系統等三大主流，這些分類是以資料庫的資料分散性及本地端系統自主性做為分類原則。

分散式會計系統除了要有後端的資料庫外，會計系統的核心其實是應用程式模組的集合。在大型 ERP 系統中會計模組的功能複雜，需要特別的應用程式伺服器來分別承載，為了讓應用程式、資料庫與使用者端之間能交互運作及分享資料，中介軟體及適當的網路模型扮演重要的角色。

在實務上，跨國企業所採用的分散式會計系統已偏向具跨功能整合的 ERP 系統，其中，多數大型企業又以 SAP R/3 為首選，本文所探討的 SAP R/3 會計系統虛擬個案架構分析，可以做為跨國企業建置 SAP R/3 ERP 系統之參考。至於其他 ERP 產品(如 Oracle、Baan、PeopleSoft、J. D. Edwards、鼎新等等)，也多有完備的會計模組。在 ERP 系統的各類模組中，會計模組內建的系統流程大多遵循國際會計準則 (IASC GAAP) 及美國會計準則 (US GAAP)，此二套準則也是大部分國家制訂一般公認會計原則的重要參考，因此會計模組是 ERP 系統中標準化程度最高的模組。故本文的虛擬個案架構分析，也可做為企業採用其他 ERP 產品時建置會計模組的參考。至於中小型企業，雖然不一定需要使用 ERP，但仍需要一個建置妥善的分散式會計系統，本文圖 6 所描述的多種層次分散式會計系統結構可供中小企業建置分散式系統之參考；若干中型企業也許無力自行建置、卻需要一個接近 ERP 系統的整合性企業資訊系統，則可考慮向 ASP(Application Service Provider，應用程式服務供應商)業者購買 ERP 系統的租借服務，在此架構下，企業可扮演單純的使用者端，卻可享受多層次分散式系統的效益。

## 參考文獻

1. Blain, J., B. Dodd, and D. Sandison (1998), *Administering SAP R/3: The FI-Financial Accounting and CO-Controlling Modules*, New York: QUE Company.
2. Buck-Emden, R., and J. Galimow (1996), *SAP R/3 System---A Client/Server Technology*, England: Addison-Wesley.
3. Cardenas, A. F. (1987), "Heterogeneous Distributed Database Management: HD—DBMS," *Proceedings of the IEEE*, Vol. 75(5): 588-600.
4. Date, C. J. (2000), *An Introduction to Database Systems*, 7th Edition. Massachusetts: Addison-Wesley.
5. Du, W., and A. Elemagarmid (1989), "Quasi-serializability: A Correctness Criterion for Global Concurrency Control in InterBase," *Proceedings of 15th International Conference On Very Large Data Bases*, pp. 347-355.
6. Dwyer, P., K. Kasravi, and M. Pham (1986), *A Heterogeneous Distributed Database Management*, Report CSC-86-7: 8216, Minnesota: Honeywell Corporate Research Center.
7. Ferrier, A., and C. Stanfret (1982), "Heterogeneity in the Distributed Data Management System SIRIUS-DELTA," *Proceeding of 8th International Conference on Very Large Data Bases*, pp. 45-53.
8. Gligor, G., and R. Popescu-Zeletin (1986), "Transaction Management in Distributed Heterogeneous Database Management Systems," *Information Systems*, Vol. 11(4): 287-297.
9. Goldman, J. E., P. T. Rawles, and J. R. Mariga (1999), *Client/Server Information Systems---A Business-Oriented Approach*, New York: John Wiley & Sons.
10. Heimbigner, D., and D. McLeod (1985), "A

- Federated Architecture for Information Management,” *ACM Transaction on Office Information Systems*, Vol. 3(3): 253-278.
11. Landers, T., and R. L. Rosenberg (1982), “An Overview of MULTIBASE,” In H.-J. Schneider (ed.), *Distributed Data Bases*, Amsterdam: North-Holland.
  12. Litwin, W. (1988), “From Database Systems to Multidatabase Systems: Why and How,” *Proceedings of British National Conference on Databases*, Cambridge: Cambridge University Press.
  13. Mohan, C., and R. T. Yeh (1978), *Distributed Data Base Systems: A framework for Data Base Design*. In *Distributed Data Bases, Infotech State-of-the-art Report*, London: Infotech,
  14. Özsu, M. T., and P. Valduriez (1999), *Principles of Distributed Database Systems, 2nd Edition*, New Jersey: Prentice Hall.
  15. Perez, M., A. Hildenbrand, B. Matzke, and P. Zencke (1999). *SAP R/3 in the Internet*, New York: Addison-Wesley.
  16. SAPAG. (2000a), *Functions in Detail*. (CD)
  17. ----- . (2000b), *R/3 4.6C Online Help*. (CD)
  18. Schreiber, F. (1977), “A Framework for Distributed Systems,” *Proceedings of International Computing Symposium*, pp. 475-482.
  19. Smith, J. M., P. A. Bernstein, U. Dayal, N. Goodman, T. Landers, K. Lin, and E. Wong (1981), “MULTIDATABASE: Integrating Heterogeneous Distributed Database Systems,” *Proceedings of National Computer Conference*, pp. 487-499.
  20. SPARC (1975), “ANSI/X3/SPARC Study Group on Data base Management Systems. Interim Report,” *ACM FDT Bulletin*, Vol. 7(2).
  21. Templeton, M., D. Brill, S. K. Dao, E. Lund, P. Ward, A. L. P. Chen, and R. MacGreger (1987), “Mermaid--A Front-End to Distributed Heterogeneous Databases,” *Proceedings of the IEEE*, Vol. 75(5): 695-708.
  22. Tsichritzis, D., and A. Klug (1978), “The ANSI/X3/SPARC DBMS Framework Report of the Study Group on Database Management Systems,” *Information Systems*, Vol. 1: 173-191.

## 附錄

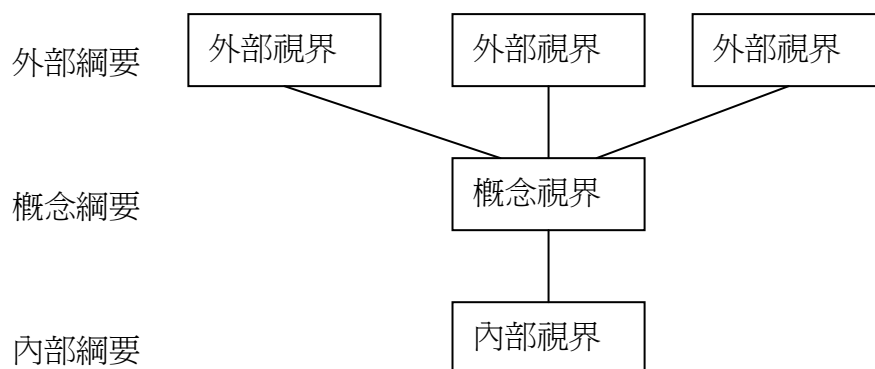


圖 1 ANSI/SPARC 架構

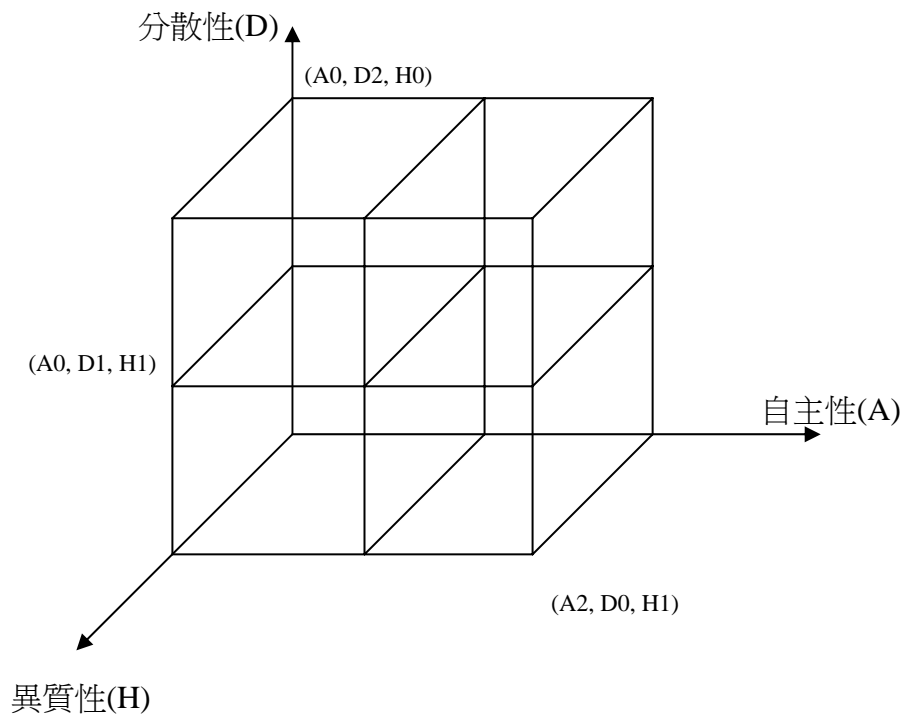


圖 2 分散式系統架構  
[取材自 Özsu and Valduriez(1999)]

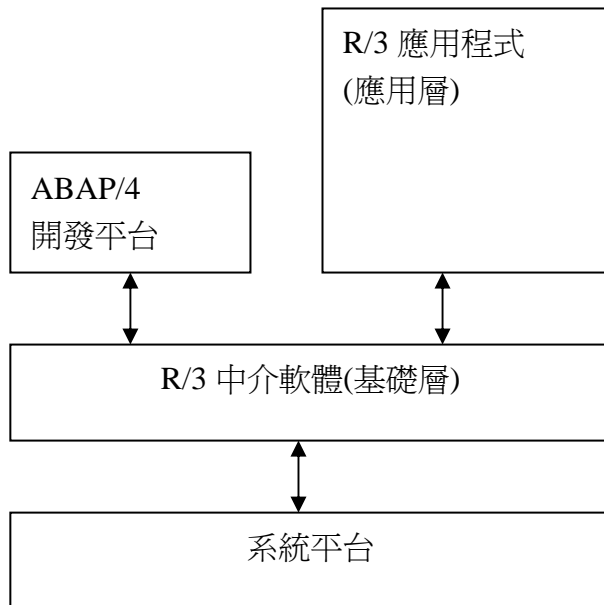


圖 3 SAP R/3 基本技術架構  
[取材自 SAP AG(2000b)]

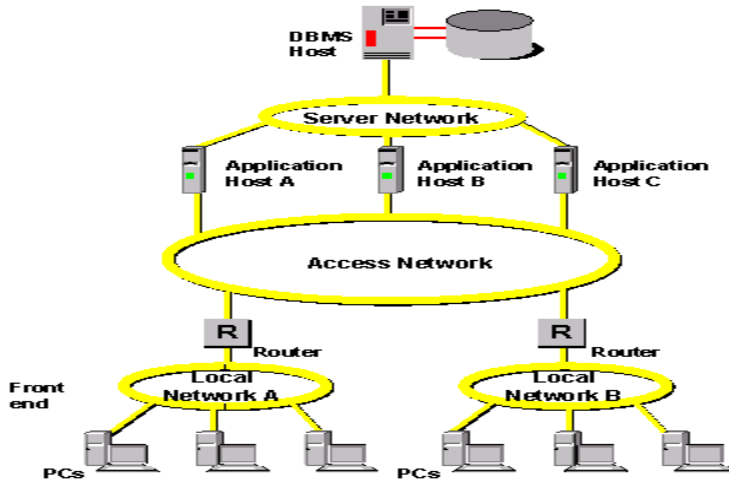
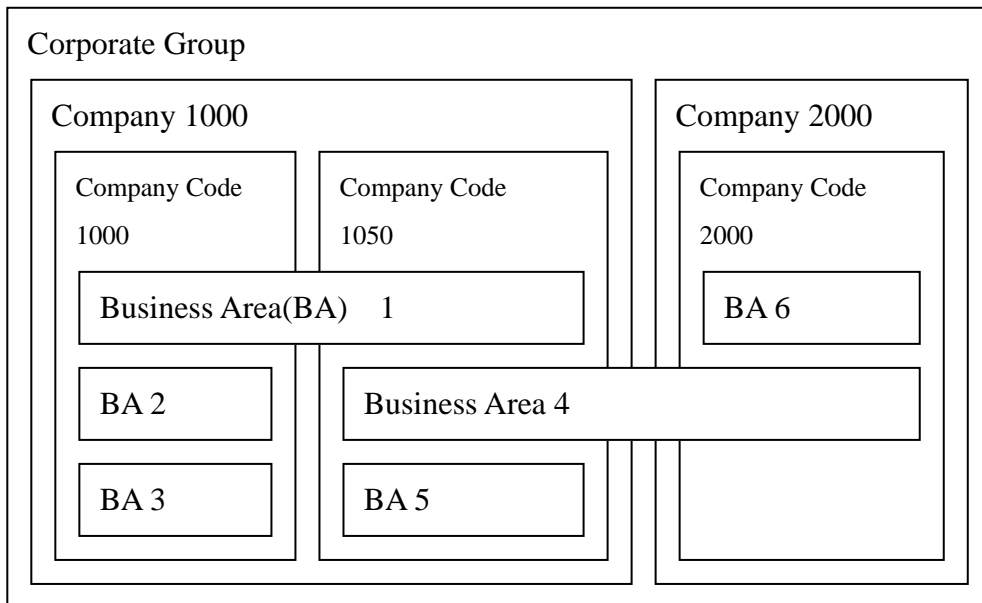


圖 4 SAP R/3 基本網路架構  
[取材自 SAP AG(2000b)]

(a) Corporate Group、Company、Company Code、Business Area



(b) Operating Concern、Controlling Area、Company Code

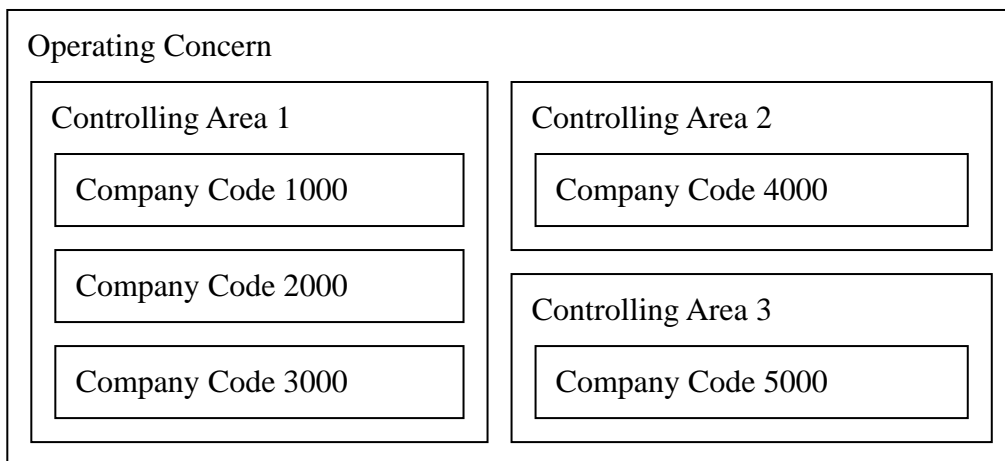


圖 5 FI 與 CO 模組組織單元之部份結合

表 1 分散式系統的不同組合

自主性	分散性	異質性	會計系統涵義
A0	D0	H0	在同一台機器上的高度整合會計系統
A0	D0	H1	在同一台機器上，用不同資料模式設計個別會計資料庫，卻高度整合的會計系統
A0	D1	H0	同質化高度整合的主從架構會計系統(分散式系統的典型)
A0	D1	H1	異質化高度整合的主從架構會計系統(分散式系統的典型)
A0	D2	H0	同質化高度整合的主對主架構會計系統(分散式系統的典型)
A0	D2	H1	異質化高度整合的主對主架構會計系統(分散式系統的典型)
A1	D0	H0	在同一台機器上，重複安裝相同資料模式的會計資料庫，每個資料庫負責不同會計模組功能，每個模組可以獨立運作，但部分作業須與其他模組協調
A1	D0	H1	在同一台機器上，用不同資料模式設計個別會計資料庫，每個資料庫負責不同會計模組功能，每個模組可以獨立運作，但部分作業須與其他模組協調
A1	D1	H0	主從架構同質化聯邦式會計系統
A1	D1	H1	主從架構異質化聯邦式會計系統
A1	D2	H0	主對主架構同質化聯邦式會計系統
A1	D2	H1	主對主架構異質化聯邦式會計系統
A2	D0	H0	不實際
A2	D0	H1	在同一台機器上，安裝由不同軟體公司所開發的不同會計模組(總帳、進銷存...等)，每個模組均獨立運作。
A2	D1	H0	不實際
A2	D1	H1	多元化主從架構會計系統(分散式多元化系統的典型)
A2	D2	H0	不實際
A2	D2	H1	多元化主對主架構會計系統，個別系統自成一完整體系(包含全部會計模組)，少部份作業(如編制合併報表)才需要跨系統進行資料分享(分散式多元化系統的典型)

表 2 SAP R/3 系統會計模組內容

(a)FI 模組

代號	名稱	內容
FI-GL	General Ledger Accounting	總帳會計
FI-AR	Accounts Receivable	應收帳款會計
FI-AP	Accounts Payable	應付帳款會計
FI-AA	Asset Accounting	財產會計
FI-SL	Special Purpose Ledger	特殊目的帳簿
FI-FM	Funds Management	基金管理
FI-BL	Bank Accounting	銀行會計
FI-LC	Legal Consolidation	編製合併財務報表
FI-TV	Travel Management	差旅管理
EC-CS <sup>a</sup>	Consolidation	編製內部及外部合併報表

(b)CO 模組

代號	名稱	內容
CO-OM <sup>b</sup> -CEL	Cost and Revenue Element	成本及收入要素會計
CO-OM-CCA	Cost Center Accounting	成本中心會計
CO-OM-OPA	Internal Orders	內部成本及收入單據控制
CO-OM-ABC	Activity-Based Costing	作業基礎成本會計
CO-PC	Product Cost Controlling	產品成本控制
CO-PA	Profitability Analysis	獲利力分析
EC-PCA <sup>c</sup>	Profit Center Accounting	利潤中心會計

註：a. EC-CS 為 EC 模組之子模組，但功能與 FI-LC 接近，故列於此。 b. OM 為 Overhead Management(費用管理)之簡稱。  
c. EC-PCA 為 EC 模組之子模組，但功能與 CO 模組有關，故列於此。

表 3 甲公司(虛擬公司)SAP R/3 系統組織結構

Company Code & 名稱	Company	Controlling Area & Subgroup	COA	營業重點	本地端 R/3 模組
1000 台北	1000	1	T(OCO A)	銷售, 管理	全套 FI, 部份 CO, SD, MM, HR
2000 上海	2000	2	C(OCO A) T(GCO A)	製造, 銷售, 管理	全套 FI, 全套 CO, SD, MM, PP, HR
2030 香港	2000	2	C(OCO A) H(LCO A) <sup>b</sup>	製造, 銷售	部份 FI, 部份 CO, SD, MM, PP
2070 北京	2000	2	C(OCO A)	銷售	透過 ALE 連接上海
3000 新加坡	3000	3	S(OCO A) T(GCO A)	銷售, 管理	部份 FI, 部份 CO, SD, MM, HR
4000 曼谷	4000	3	B(OCO A) T(GCO A)	製造	部份 FI, 其餘透過 ALE 連接新加坡
5000 吉隆坡	5000	3	S(OCO A) M(LCO A) <sup>b</sup>	製造	部份 FI, 其餘透過 ALE 連接新加坡
6000 東京	6000	4	J(OCO A) T(GCO A)	銷售, 管理	部份 FI, SD, HR, 其餘透過 ALE 連接台北
6050 大阪	6000	4	J(OCO A)	銷售	透過 ALE 連接東京
7000 漢城	7000	4	K(OCO A) T(GCO A)	銷售	部份 FI, 其餘透過 ALE 連接東京

註：a: T 表示台灣版的 COA; C 表示大陸版; H 表示香港版; S 表示新加坡版; B 表示泰國版; M 表示馬來西亞版; J 表示日本版; K 表示韓國版。

b: 假設香港分公司用大陸版 COA 記帳，但按香港規定須另行以香港版 COA 申報資料；假設吉隆坡分公司用新加坡版 COA 記帳，但按馬來西亞規定另行以馬來西亞版 COA 申報資料。