

# 協同產品開發模式下產品資料之控管

孫彰鴻

亞洲大學資訊科學與應用學系

g94251001@msl.asia.edu.tw

吳銜容

亞洲大學資訊科學與應用學系

psuhjw@asia.edu.tw

## 摘要

隨著社會環境快速變動，企業與企業之間的競爭顯得更為激烈，單一企業產品開發是無法適應現況環境。企業透過與其它企業的共同合作，協力開發產品，以提升產品開發的效率與效能，因而企業之間有了「協同」產品開發的概念。在「協同」產品開發模式下，企業之間在控管產品資料上產生許多問題，例如：企業之間產品資料版本不一致與不當之使用者存取等，常常造成企業產品開發的誤差與成本增加。為解決上述問題，本研究建構一個產品資料控管架構，此架構核心主要包含 User model、Data model 與 Rule function。當使用者存取產品資料時，系統會呼叫 User model 與 Data model，並附予相關資訊，然後再觸發 Rule function 加以規則式運算，由系統給予使用者與產品資料最佳化的控管。本研究藉由上述架構之探討與分析，提供在「協同」產品開發模式下，企業對於產品資料的控管，避免產品資料遭受損害，促進企業之間「協同」產品開發模式的成長與進步，進而提升企業競爭優勢與產品開發能力。

**關鍵詞：**協同產品開發，產品資料控管

## 1. 緒論

隨著經濟環境變動日益劇烈，企業面臨競爭壓力也相對地增加，因此在競爭激烈的環境下脫穎而出是現今企業所著重策略之一。又近幾年顧客需求多變化，企業為因應顧客之需求，其營運策略以顧客導向為出發點，在最快速的時間內提供客製化之產品。在這樣的競爭環境下，企業若只依賴自己本身的產品開發，是無法因應現況環境；因此企業為了提升本身之競爭優勢，有賴於企業之間共同合作開發產品。隨著網際網路發展普及，企業不再局限空間與時間限制，更可加速企業之間的合作。依上述所言，使得各個企業紛紛朝向「協同」產品開發模式發展。「協同」意指大家一起朝著自定目標前進，共同協力合作，進而達成自定目標。企業透過「協同」產品開發模式，共同合作開發產品，有助於降低企業成本及提升企業之產品開發效率與效能，進而增加企業競爭優勢。

「協同」產品開發模式是屬於一個開放的共同合作環境，若是企業缺乏有效產品資料控制與管理，將會導致花費大量成本。因此企業如何有效控管產品資料，以達到資料機密性、完整性與可用

性，顯得相當重要。在全球運籌管理的催化下，「協同」產品開發模式更加盛行，但是其相關問題也隨之而來，例如：企業之間產品資料版本不一致，導致產品開發的落差；不當的使用者存取權限設定，造成產品資料的損壞…。由上可知，在「協同」產品開發模式下，產品資料是屬於企業重要資產，若是企業沒有可靠的控管架構，控管產品資料，會使得企業付出相當成本代價。因此如何在「協同」產品開發模式下有效控管產品資料，是本研究所要探討之問題。

為了解決上述問題，本研究以 three-tier 概念，建構一個產品資料控管之架構，分別由 User-tier、Logic-tier 與 Data-tier 所組成。此架構主要核心為 Logic-tier，其包含 User model、Data model 與 Rule function。當使用者存取產品資料時，系統會呼叫 User model 與 Data model，並附予相關資訊，然後再觸發 Rule function 加以規則式運算，由系統給予使用者與產品資料最佳化的控管。企業可藉著此架構之機制，有效地控管產品資料，並且促進企業之間「協同」產品開發的成長與進步。

本研究藉著上述架構分析與探討，預期未來實

際以系統實做呈現，使得企業可以透過此系統所帶來之產品資料控管機制，達到產品資料機密性、完整性與可用性之效用，進而提升企業競爭優勢與產品開發能力。

## 2. 文獻探討

在全球化的競爭趨勢下，企業紛紛朝向「協同」概念邁進，藉著與企業之間的合作，一方面可以加快產品上市時間，另一方面可以因應顧客需求之不同，提供更彈性化與多樣化產品，進而提升企業核心價值。在「協同」產品開發模式裡，如何控管產品資料，顯得相當重要。有效控制與管理產品資料，可以進一步促進企業之間的協同合作；同時企業也藉著協同合作的效益，增加企業競爭優勢。本研究針對上述相關領域，本章節將予以整理與分類。

### 2.1 協同設計模型

Z. Hang 等作者依協同操作的時間特性，主要分為並行模型（parallel mode）與循序模型（sequential mode）。並行模型屬於線上協同類型（on-line type），意指多人同時上線，對於同伴產品進行設計；循序模型屬於離線式協同類型（off-line type），意指在不同的時段裡，進行產品設計。[1]

Anumba 依時間與地點，將協同區分四種模型：同步分散式協同（synchronous distributed collaboration）、非同步分散式協同（asynchronous

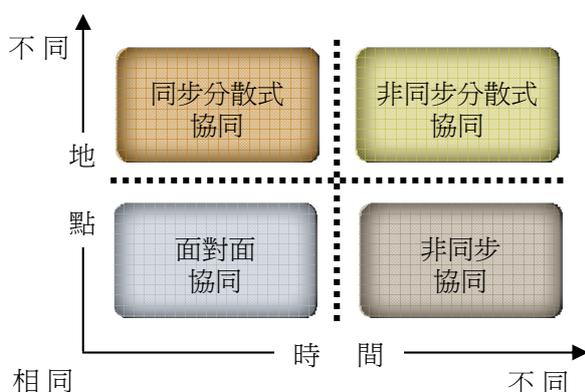


圖 1 協同設計模型

distributed collaboration)、面對面協同（face to face collaboration）、非同步協同（asynchronous collaboration），如圖 1。[2]

Kvan 將協同分為兩種模型：緊密耦合（close coupled）與疏鬆耦合模型（loosely coupled）。緊密耦合模型意指同一時間與地點裡，大家經由討論分析，進行共同合作完成工作，此模型優點可以綜合大家之意見想法，做即時編輯與修改；此模型缺點是無法做有效的個人績核考察。疏鬆耦合模型意指將工作依各種專業領域分派，每個人各自完成自己所分派之項目，此模型優點使各工作人員不需要在同一時間與地點裡，聚集共同完成工作；此模型缺點在於工作分派給各個工作人員，因此若是有部份工作需要修改時，可能會影響到其他部份工作的進行。[3]

### 2.2 存取控制

當使用者在存取相關資料時，系統需將使用者與相關資料進行控管，以保護資料安全性，避免資料遭受不當存取或惡意破壞。現今存取控制的概念，主要分為下列三種[4]：

- 1、Discretionary Access Control(DAC):系統直接授予使用者存取權限，例如：讀取與寫入權限，使用者透過系統所給予之權限，才可進行資料存取[5]。
- 2、Mandatory Access Control(MAC):系統裡定義一個安全層級，使用者與資料會分別與安全層級相關聯，此安全層級將會反應出使用者與資料的資訊。例如：安全層級裡 A 大於 B，又 B 大於 C，若使用者屬於 B 安全層級，其只可以對於 B 與 C 安全層級的資料，進行存取等動作[6]。
- 3、Role-based Access Control(RBAC):系統依據工作機能創造 role，再將 role 分配給使用者，每當使用者需求功能不同，系統將會分配不同 role 給使用者，使得使用者存取資料之權限更加彈性化[7]。

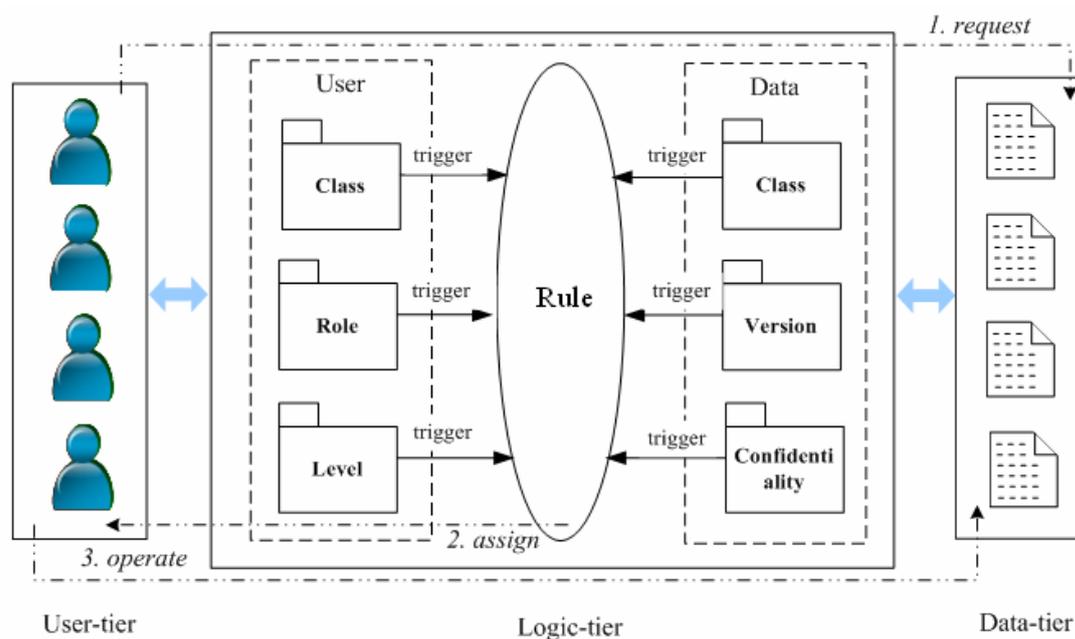


圖 2 協同環境下產品資控管之架構

### 2.3 資料安全性

資料安全性通常可分為有形 (tangible) 與無形 (intangible) 兩種特性。有形的安全保護屬於實體保護，例如：防火牆與防毒軟體。無形的安全保護則分為機密性 (confidentiality)、完整性 (integrity) 與可用性 (availability) 之特性，機密性意指唯有經過授權的使用者才可以對於資料進行存取與操作等動作；完整性意指確保資料內容安全與完整；可用性意指資料需要時可供使用者進行操作與使用之動作[8]。

### 2.4 小結

綜合上述文獻探討，本研究主要探討在協同環境下如何控管產品資料，以達到產品資料機密性、完整性與可用性。表 1，呈現三種存取方法之比較，

表 1 存取方法之比較

控管面 方法	使用者	產品資料	彈性化
DAC	O	X	低
MAC	O	O	中
RBAC	O	O	高

由表中可知，MAC 與 RBAC 存取方法都有衡

量使用者與產品資料之相關資訊，在產品資料的控管上相對地也比 DAC 較有彈性。因此本研究將會朝著 MAC 與 RBAC 存取概念，建構一個控管產品資料之架構，以有效地管理在相關產品資料。

### 3. 協同環境下產品資料控管之架構

本研究之架構以 tree-tiers 方式建置，其依序分為 User-tier、Logic-tier 與 Data-tier (如圖 2)。User-tier 是使用者相互作用的資訊介面，使用者透過此介面控管相關產品資料；Logic-tier 包含 User model、Data model 及 Rule function；Data-tier 是儲存相關產品資料部份。當使用者需要存取產品資料時，User-tier 將會呼叫 Logic-tier 至 Data-tier 找尋所需要之產品資料，當 Logic-tier 找尋符合使用者需求的產品資料時，則會啟動 User model 與 Data model，將相關使用者與產品資料之資訊匯入。其間並觸發 Rule function，Logic-tier 藉著 Rule function 邏輯運算，給予使用者適合的存取權限，存取所需要產品資料，以提升控管產品資料的安全性。由於產品資料會不斷地更新與變動，若是系統只給使用者固定存取權限，可能無法符合使用者需求及控管產品資料安全性。因此本系統利用使用者與產品資料的資訊進行交互比對，以找出適合的存取權限，使得系統

<pre>&lt;?xml version="1.0" encoding="Big5"?&gt; &lt;!DOCTYPE UserModel SYSTEM "user.dtd"&gt; &lt;UserModel&gt;   &lt;Class&gt;     &lt;company&gt;亞洲公司&lt;/company&gt;     &lt;actor&gt;承包商&lt;/actor&gt;   &lt;/Class&gt;   &lt;Role&gt;     &lt;name&gt;吳小明&lt;/name&gt;     &lt;appointment&gt;專案管理者&lt;/appointment &gt;   &lt;/Role&gt;   &lt;Level name="write"&gt;0.7&lt;/Level&gt;   &lt;Level name="read"&gt;0.7&lt;/Level&gt; &lt;/UserModel&gt;</pre>	<pre>&lt;?xml version="1.0" encoding="Big5"?&gt; &lt;!DOCTYPE DataModel SYSTEM "data.dtd"&gt; &lt;DataModel&gt;   &lt;Class&gt;     &lt;name&gt;產品類&lt;/name&gt;     &lt;condition&gt;open&lt;/condition&gt;   &lt;/Class&gt;   &lt;Version&gt;     &lt;filename&gt;產品資料_1&lt;/filename&gt;     &lt;edition&gt;1.0 版&lt;/edition&gt;   &lt;/Version&gt;   &lt;Confidentiality&gt;0.8&lt;/Confidentiality&gt; &lt;/DataModel&gt;</pre>
---	---

圖 3 User model 與 Data model

會依使用者需求與產品資料的資訊不同，給予彈性化及安全化的存取權限。

### 3.1 User model 與 Data model

User model 是管理使用者相關資訊，包含 Class、Role 與 Level 類別。Class 類別是分類使用者的類別，例如：使用者所屬單位與公司；Role 類別是使用者所屬的職稱，例如：使用者與管理者；Level 類別表示使用者的權限等級，權限等級愈高者，表示使用者存取權限也愈高。Data model 是管理產品資料相關資訊，其包含 Class、Version 與 Confidentiality 類別。Class 類別是將產品資料做分類，區分不同類型產品資料，例如：產品類與零件類；Version 類別管理產品資料的版本區別，例如：1.0 版與 2.0 版；Confidentiality 則顯示產品資料的機密程度，產品資料機密程度愈高者，表示其機密性也愈高。

User model 與 Data model 是利用 XML 語法建構，XML 是一套資料交換標準，具有結構化與跨平台等特性。如圖 3 所示，User model 與 Data model 包含了上述類別，在 User model 的 Level 類別裡，分成 write 與 read 權限屬性，每一個權限屬性都附予一個衡量值，以表示該屬性的權限範圍，衡量值

範圍可限定在 0.1 至 1。相同地，在 Data model 的 Confidentiality 類別裡，也附予一個衡量值，此衡量值表示產品資料是否具有高度機密性，衡量值範圍可限定在 0.1 至 1。User model 的 Level 值與 Data model 的 Confidentiality 值，系統將會依使用者需求不同與產品資料狀態不同，給予最適當的值，提供彈性化的產品資料控管。

本研究藉著將使用者與產品資料模組化概念，探討使用者與產品資料之間的相互作用，使得使用者存取產品資料時，系統給予使用者適合的權限範圍，並且當使用者需求與產品資料之間狀態改變時，相對地系統在權限控管方面，也會彈性地調整。

### 3.2 Rule function

當 User model 與 Data model 被系統呼叫時，此兩模組會將使用者與產品資料之相關資訊予以載入使用，並且觸發 Rule function。藉著規則式邏輯運算，進行此兩模組之間相關資訊的相互作用，然後由 Rule function 給予使用者適合的權限範圍。每一次使用者存取產品資料的情況，Rule function 會遵循 User\_Data\_Condition (UDC) pattern 記錄於資料庫中，以達到追蹤的效用。因此當使用者不當存

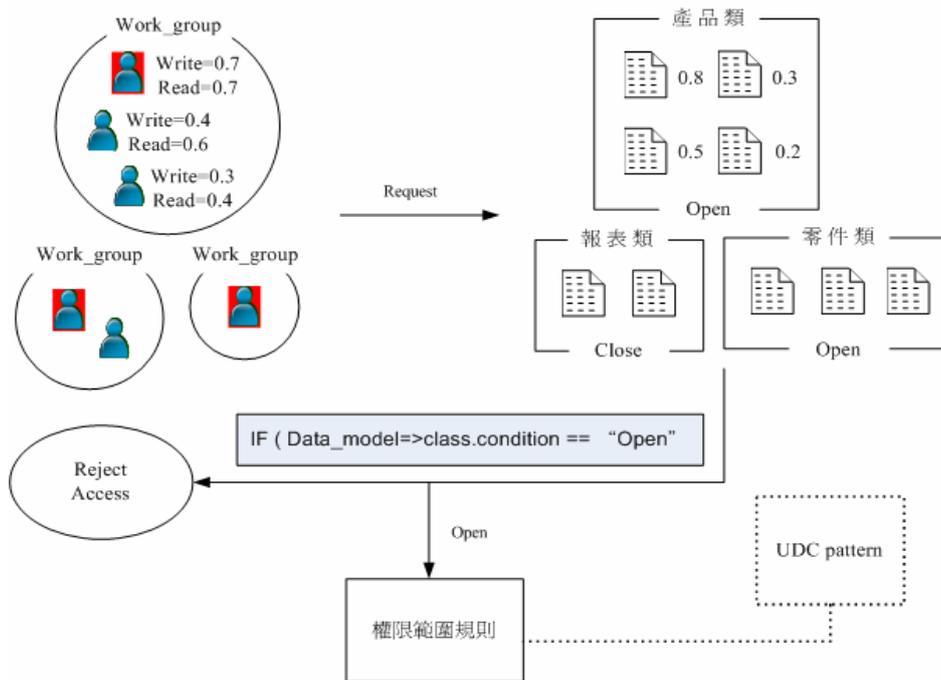


圖 4 Rule Function

取產品資料之操作時，系統可以透過 UDC pattern，找到不當之存取動作，以作為後續系統設定權限值之衡量與評估。

如圖 4 所示，呈現使用者存取產品資料之過程。每當使用者存取產品資料，系統將會為使用者建立一個 Work\_group，使用者處於 Work\_group 環境，進行產品資料的存取，使用者也可以透過加入其它 Work\_group 方式，共同合作存取產品資料。此 Work\_group 為協同合作環境，可以同時加入多

人使用者，共同使用產品資料。Work\_group 控制權，則交由其所有使用者裡，User model 的 Level 值最高者；擁有控制權的使用者，可以在 Work\_group 裡控管使用者加入與移除。在存取產品資料時，系統會以擁有控制權之使用者為依據，進行規則式邏輯運算，以給予適當的存取權限。當擁有控制權的使用者，同意讓 Level 值更高之新使用者加入時，系統則會將控制權轉移至該新加入之使用者。

```

    權限範圍規則
    User = Max(User1,User2,User3...); //Work_group 擁有控制權之使用者
    IF ( User_model=>Level.read > Data_model=>Confidentiality )
        Assign Workstation(read);
        Call UDC Rule;
    ELSE
        Reject User(read);
    
```

圖 5 權限範圍規則

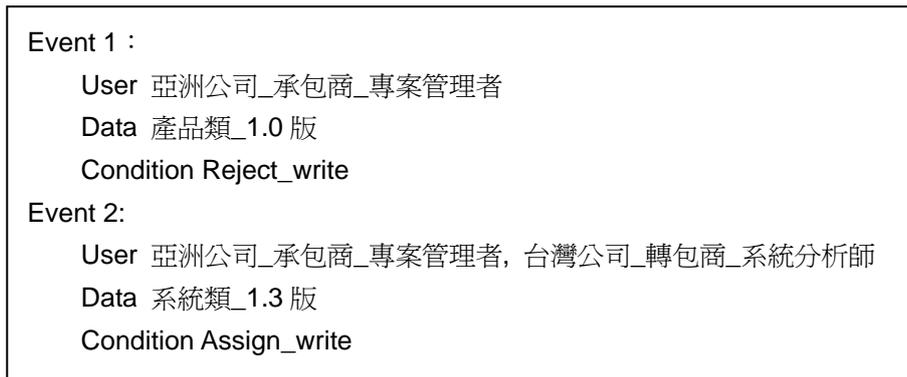


圖 6 UDC pattern

每個產品資料都有所屬之類別 (Data model 裡的 class)，類別裡包含類別名稱與類別狀態，例如：產品類\_open，零件類\_close，表示產品類資料是開放的，Work\_group 可以進入此產品類別，而零件類資料是不開放的，此 Work\_group 不可以進入此產品類別。因此 Work\_group 存取產品資料時，需視產品資料類別狀態是否為開放的狀態，若是開放狀態，系統才會進行規則式邏輯運算，給予 Work\_group 適當的權限；否則系統會顯示，此類別之產品資料並不開放進入。

當產品資料類別狀態為開放時，Work\_group 可以進入此類別之產品資料，其存取權限則由系統進行規則式邏輯運算後，再給予 Work\_group 適合之存取權限。如圖 5 所示，當 Work\_group 進入產品資料的類別時，Rule function 將會依據 Work\_group 裡擁有控制權之使用者，取出擁有控制權使用者 User model 裡 Level 值 (屬性：write 和 read)，並與 Data model 裡 Confidentiality 值做相對應比較。當 Level 值大於或等於 Confidentiality 值時，則表示擁有控制權使用者，權限範圍等級高於產品資料機密性，因此 Work\_group 裡的使用者，可以共同進行存取產品資料。反之，若是 Level 值小於 Confidentiality 值時，則表示擁有控制權使用者，權限範圍等級小於產品資料機密性，故此 Work\_group 裡的使用者，無法共同進行存取產品資料。例如：Work\_group 裡擁有控制權使用者的 User model 裡 Level(write) 值為 0.7，相對地，產品資料的 Data model 裡 Confidentiality 值為 0.5，經由 Rule function

比較後 ( $0.7 > 0.5$ )，Rule function 則會給予 Work\_group write 權限，允許 Work\_group 裡的使用者可以對於產品資料進行 write 動作。

UDC pattern 顯示與記錄每一次 Work\_group 存取產品資料的情況，並且將相關資訊存入資料庫，以供未來管理者追蹤記錄之依據。如圖 6 所示，UDC pattern，主要由 User、Data 與 Condition 三個構面所構成，以呈現使用者資訊、產品資料及權限控管之情況。系統藉著 UDC pattern 概念，將每次 Work\_group 的存取動作，當成一個事件加以管理與控制，因此使得管理者可以清楚瞭解整個協同環境下控管產品資料的事項，加以提升控管產品資料效率及效能，進而增加產品資料的安全性。

#### 4. 結論

隨著經濟環境快速變動，以往企業單打獨鬥的型式，已無法提升企業競爭力。為了因應現今的經濟環境及高度企業競爭壓力，企業藉著與企業之間的協同合作，以提升企業競爭優勢與降低企業成本支出。在協同合作環境裡，產品資料遍佈於各個不同地區之企業，然而產品資料屬於企業重要資產之一，因此企業之間如何有效控管產品資料，顯得相當重要。

本研究提出在協同開發模式下產品資料控管之架構。此架構主要依據使用者需求與產品資料狀態之間的相互作用，給予使用者適當的存取權限，並且透過 UDC pattern，詳細記入每次使用者存取產品資料之狀況，以作為後續管理者追蹤記錄的依

據，以達到在協同開發模式下產品資料的控制與管理。未來本研究將依此架構之分析，建構實際的系統環境，以驗證在協同開發模式下企業有效地控管產品資料。

### 參考文獻

- [1] 黃敬仁、張瑞芬、姚銀河，2003，『WfMC 為基之模組化網路協同設計系統分析與建置』，工業工程學刊，第四卷.第二十期：422~434 頁。
- [2] C. J. Anumba, O. O. Ugwu, L. Newnham and A. Thorpe, “Collaborative design of structures using intelligent agents,” *Automation in Construction*, Vol. 11, pp. 89-123, 2002.
- [3] T. Kvan, “Collaborative design: what is it? ,” *Automation in Construction*, Vol. 9, pp. 409-415, 2000.
- [4] R. S. Sandhu and P. Samarati, “Access control: principles and practice,” *IEEE Commun*, Vol. 32, No. 9, pp. 8-40, 1994.
- [5] B. Lampson, “Protection,” in *Proceedings of the Fifth Annual Princeton Conference of Information Sciences and Systems*, pp. 43-437, 1971.
- [6] S. L. Osborn, R. S. Sandhu and Q. Munawer, “Configuring role-based access control to enforce mandatory and discretionary access control policies,” *Inform Syst Security*, Vol. 3, No. 2, pp. 85-106, 2003.
- [7] R. S. Sandhu, E. J. Coyne, H. L. Feinstein and C. E. Youmn, “Role-based access control models,” *IEEE Comput*, Vol. 29, No. 2, pp. 38-47, 1996.
- [8] F. Biennier and J. Favrel, “Collaborative business and data privacy: Toward a cyber-control?,” *Computers in Industry*, Vol. 56, pp. 361-370, Jan. 2005.