行政院國家科學委員會專題研究計畫 成果報告

適性化遊戲教學平台開發及其對輔教成效研究- 結合角色扮演遊戲與動態評量
研究成果報告(精簡版)

計 畫 類 別 ： 個別型
計 畫 編 號 ： NSC 97-2221-E-343-001-
執 行 期 間 ： 97 年 08 月 01 日至 98 年 07 月 31 日
執 行 單 位 ： 南華大學電子商務管理學系

計畫主持人： 王昌斌
共同主持人： 謝昆霖、施能木、黃豐國、張介耀、陳宗義
陳仁義

處 理 方 式 ： 本計畫涉及專利或其他智慧財產權，2 年後可公開查詢

中　華　民　國　98 年 10 月 22 日

行政院國家科學委員會補助專題研究計畫期末報告

適性化遊戲教學平台開發及其對補教成效研究-結合角色扮演遊戲與動態評量

計畫類別：■個別型計畫　　整合型計畫

計畫編號：NSC97-2221-E-343-001-

執行期間：97 年 8 月 1 日至 98 年 7 月 31 日

計畫主持人：王昌斌　南華大學電子商務管理系

共同主持人：謝昆霖　國立台東大學資訊管理學系

施能木　國立台東大學美勞教育學系

黃豐國　國立台東大學數學教育學系

計畫參與人員：

張介耀　南華大學電子商務管理系

陳宗義　南華大學電子商務管理系

陳仁義　南華大學資訊管理學系

陳萌智　國立成功大學製造工程研究所

楊惠媚　大同商業專科學校

曾清義　南華大學資訊管理研究所

蔡政宇　南華大學資訊管理研究所

陳育銘　南華大學資訊管理研究所

藍鈺凱　南華大學資訊管理研究所

張堯榮　南華大學資訊管理研究所

執行單位：南華大學電子商務管理系

中　華　民　國　98　年　9　月　18　日

摘要:

從先前學者研究遊戲融入學習，能提升學生學習興趣及學習成效，但這種系統必須針對不同主題開發各別軟件，其缺點為成本高且內容缺少彈性化，本研究將知識管理概念結合角色扮演遊戲，期望藉由遊戲情境闖關之動機，激發其對學習之意願及學習效率；系統分為兩大主題--「遊戲」與「學習」分開處理，有別於目前市場將學習融入遊戲之軟件。此種架構下，不但可以節省開發成本，且內容可彈性化，目前尚未有相關學者研究或開發該領域。

本研究主題有下列主要研究項目:

1. 如何將知識管理應用於學習診斷；
2. 透過學習診斷判斷學生程度並提供學習導讀；
3. 提高學生對學習之興趣；
4. 協同學習及存取控制。

- 研究動機

教育部之教育改革行動方案中曾規劃補救教學之辦理內容:「研究補救教學之教材與教法，提供更多元的學習機會，以配合學生個別差異，激發學生潛能，達到適性教育目標」。然而補救教學之實際運作上仍遭遇一些困境，例如：課程開設時間利用下課後及假日，常面臨教師人力不足、學生上課情緒低落、缺席率居高不下的窘境等。

有鑑於此，國內許多研究提出了利用數位學習來施行補救教學，數位學習有著傳統教學所沒有的特點：不受制於時間與空間的分隔，實現任何時間與地點皆可學習的理想，另外還有其它特點：自調式學習進度、豐富的學習資源、降低教學成本與用途廣泛等，因此如將數位學習應用於補救教學上應有很大之助益，但是如果數位學習之設計只是一味地將教材上網，仍無法解決補救教學在學習動機與適性化方

面之問題。因此如何提供適當的教材給予適當的學習者，此一適性化的教學策略與技術是值得研究的主題；再者，在國內施行補救教學之實作上甚少針對如何提高學習動機與學習成效進行技術上之開發與實證之研究，針對此目的是我們的研究中心所在。本計畫之研究動機所述如下：

本計畫研究動機之一：如何將遊戲融入數位學習中，藉此有效提昇補救教學施行對象之學習動機。

傳統的遊戲教學乃是利用遊戲設計達到教學的目的，在我們先前的研究中發現此類型的遊戲設計不僅不易訂定遊戲主題與遊戲內容來符合教學，而且整個遊戲的完成時間較短，當遊戲進行結束後重新再玩的意願亦會明顯降低，對於以學科為補救教學的遊戲教學方式，實有改良之必要性，因此如何製作不同於以往的遊戲教學，建置具共享機制之遊戲教學平台，可擴增的遊戲主題與劇情、評量題庫與教材內容，配合教師的教學策略並持續提昇學習者的動機與學習成效是本計畫亟欲探討的方向。

本計畫研究動機之二：如何利用循序漸進之評量方式在補救教學過程中提供更適性化的學習內容與知識。

動態評量又稱協助式評量，乃是在測驗進行中給予學生提供暗示、線索及協助，對學生的認知能力或特定學科領域進行持續性學習歷程的評量。透過動態評量之實施配合循序漸進的選題策略，在評量過程中配合受試者解題之需要給予必要之協助，使受測者得到適性化的協助，並獲得最大之潛能發展，達到因材施教之教育目標。

本計畫之研究動機之三：結合領域之教師、學生、家長與專家合作經營，並對於本計畫之結合角色扮演遊戲與動態評量進行教學實驗與成效研究。

雖然數位學習已能方便提供多元化的學習環境，但仍有許多值得改進之處，如教育部在「挑戰2008國家發展重點計畫」中提到，在資訊化時代的社會，國民具有資訊應用知能與完善的數位化學習環境是國家發展的基礎條件，目前各級學校已完成網路學習基礎硬體建設，但是中小學師生要應用網路資源協助教與學及網路學習其內涵則有待加強建置。故自教育部極力推展「資訊科技融入各科教學」教育政策以來，網路教學系統的實施與推展遇到瓶頸及缺失如下：

- 教師研習各資訊科技後只是片斷零亂的資訊科技使用，無法將其整合和運用，造成資源浪費。
- 資訊科學技術日新月異，要熟悉各程式語言及軟體使用往往要投入大量時間與腦力，現實環境無法配合，造成老師排斥及望而卻步。
- 同學與同學、教師與同學之間缺乏教學互動機制，無法結合教學理論提昇學習效果。
- 網路課程架設在其他教學平台，受制他人操作常要一段適應期，無法達到隨時補充教材及更新教學內容的機制。

本計畫所建置之遊戲教學平台除了達到提昇補救教學之學習成效外，最終乃希望能可以永續經營以造福補救教學之學生、家長與教師，因此，除持續改良資訊技術外，最重要的是真正證實此一研究以結合角色扮演遊戲與動態評量之方式實際具有可提昇補救教學之學習成效，因此有必要進行一系列教師專家會談並實地推行與實際進行施測，以證實此研究之可行性。

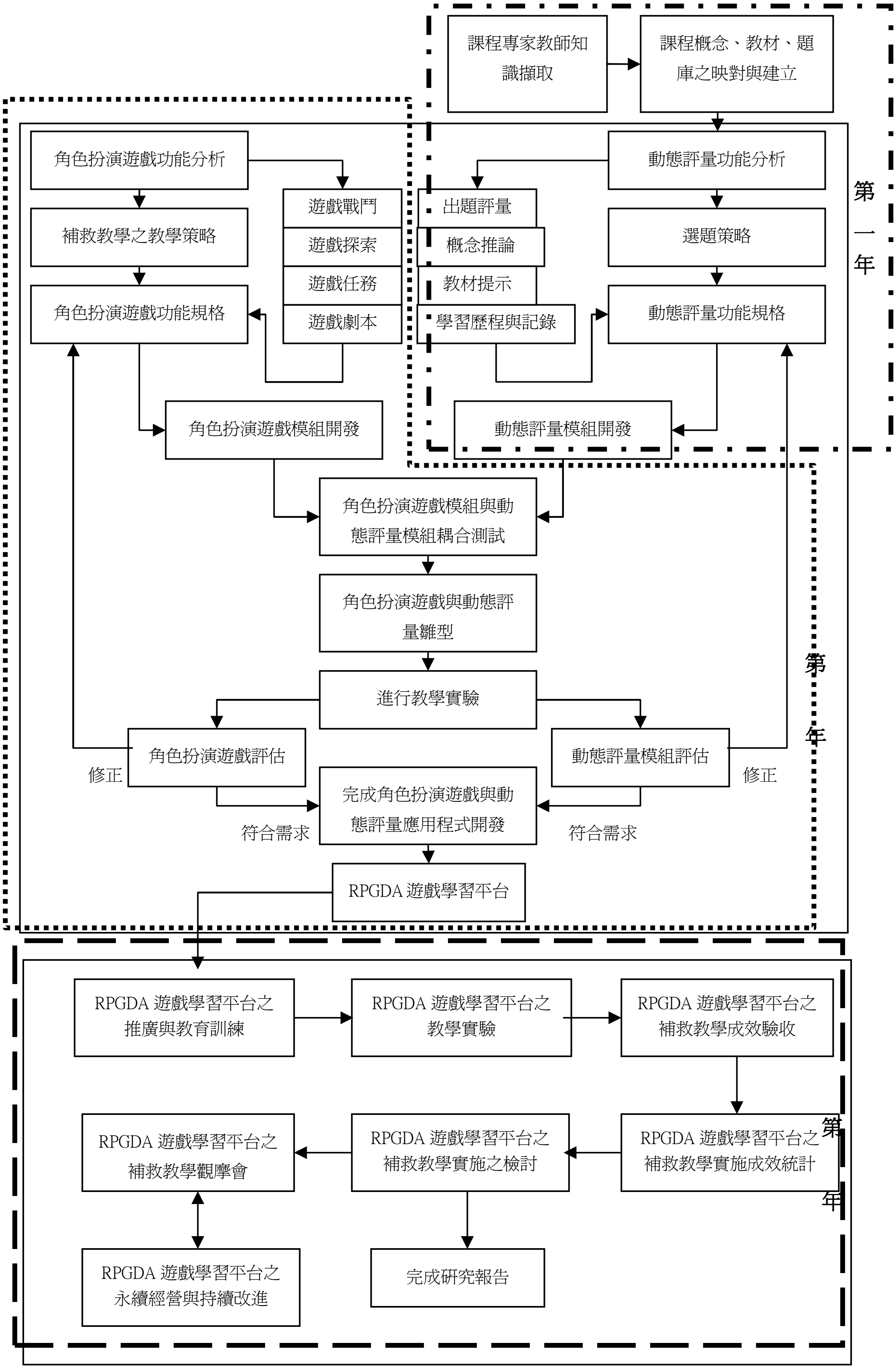
- 研究目的

本計畫目的在於利用數位學習方式輔助國小學童補救教學，並透過結合角色扮演遊戲與動態評量之學習方法以提高學習動機，並收到實質之教學成效，達到教育改革之目標。

針對上述研究目的，本計畫之預期目標如下：

1. **利用遊戲教學方法提高學習動機**:將補救教學之課程內容轉化至遊戲教學之中，學生藉著角色扮演遊戲進行學習，引發學習的動機，增加學習的經驗,促進認知結構之發展,並且建立長久的學習效果。
2. **利用動態評量方式持續診斷學習能力**:將評量的題目與隱含在題目中的知識透過系統化的組織與關連,利用動態評量的方式在遊戲過程中持續診斷學生個別的能力水準,協助學生發現與分析學習不足之處及迷思的學習概念，以達到學生適性發展之目的。
3. **提供適性化學習內容**：利用診斷學生的能力發現學習概念迷思之處，然後分配給學生適當的學習單元內容與學習路徑，經由評量—教學—再評量的方式，一再的反覆進行，使得學習者能進行自我學習、自我診斷，來補足自己所欠缺的概念，建構適合個別學習者的學習鷹架，幫助學生達到最佳的學習發展。
4. **提供領域專家與成員共同發展補救教學之平台**:透過補救教學之平台讓教育專家對於某一教學內容之分析找出隱含的知識與概念,設計相關的評量題目與引用的相關概念,藉由本體論的方法將題庫、教材與知識概念做一連結與表徵,而遊戲設計專家則透過平台設計遊戲劇本與遊戲內容,最後由教師設定相關的教學進度與選題策略讓學生在平台上進行補救教學之遊戲。
5. **探討其成效以作為將來補救教學研究之參考**:透過長期持續實際運作與施測,以證實此一研究以結合角色扮演遊戲與動態評量之方式實際具有可提昇補救教學之學習成效。

- **預定計畫執行之進度與工作項目**



本研究計畫之進行步驟與工作項目

■ **動態評量系統建置與測試(第一年)**

1. 利用「專家教師知識誘導」來建構「專家教師課程知識概念ontology」，並透過實際之知識概念之評量與評價(knowledge verification and validation)，進行ontology之修正與確認。
2. 依據課程知識概念ontology，在每個概念中建置評量題目與教材內容，完成數位知識內容建構。
3. 發展適用的概念評量與診斷機制，討論其信、效度，作為監控學生概念迷思的工具。
4. 依據課程知識概念ontology配合診斷機制，發展動態評量子系統各功能運作，進一步豐富整合計畫評量學生的概念迷思補救教學之教材。
5. 完成動態評量子系統核心技術與功能模組之整合與測試。

■ **角色扮演遊戲開發(第二年)**

1. 探討角色扮演遊戲之劇本模式、戰鬥模式、探索模式與事件模式，各模式之間的關係，並分析與規劃適用於補救教學專屬元件之原則與流程。
2. 以系統工程之方法與技術及物件導向分析與設計之方法與技術，設計各功能模組之規格。
3. 使用系統雛型化開發方法與軟體元件技術，分別針對角色扮演遊戲

模組各別開發應用程式，並使其具開放性、分散性與整合性。

4. 結合附近國民小學學童，針對國小補救教學對象實施 RPGDA 雛型之測試、評估與調查。

5. 依測試結果改進與檢討 RPGDA 雛型，如須改進 RPGDA 之雛型，則重回到步驟 3，修正兩模組之規格、設計與結合方式，依此反覆進行，直至找出最佳之結合模式。

6. 完成 RPGDA 之應用程式開發，並使應用程式所完成之 RPGDA 課程元件具有開放性、分散性與整合性。

7. RPGDA 中動態評量系統與角色扮演系統整合。(可能延續至第三年上半年)

■ **遊戲教學實驗與成效研究（第三年）**

1. 依據前二年所開發之 RPGDA 之遊戲學習平台，舉辦說明會並邀請具實務經驗之教師參與教育訓練，並將其推展至日常之補救教學活動中，使其與補救教學同時並行運作，並記錄學習成效與具體指標之達成。
2. 依據補救教學之實務運作與平行測試之結果，調整動態評量與遊戲之參數，使系統更具實用之價值。
3. 發展融入本計畫的實驗組與控制組。
4. 探討學生不同的「學習型態」，其「學習成效」之相關議題。

5. 探討教師之不同的「教學策略」，其「學習成效」之相關議題。

6. 進行研究樣本學生施測和資料分析，討論補救教學之學生實驗教學效益。

7. 針對補救教學之學生研究和實務人員舉辦補救教學績效之應用研習會。

8. 成立補救教學遊戲學習與動態評量之網路開放資源與永續發展中心。

本研究計畫原規劃為三年期計畫，下列第一年已完成之進度：

1. 依據課程知識概念 ontology，在每個概念中建置評量題目與教材內容，完成數位知識內容初步建構。(附件一)
2. 發展適用的概念評量與診斷機制，討論其信、效度，作為監控學生概念迷思的工具。(附件二)
3. 角色扮演遊戲結合動態評量教學平台架構研究及開發。(附件三)
4. 協同學習及存取控制之研究。(附件四)

附件一

運用模糊理論支援電腦適性化評量的探討

林明宏
南華大學資管所在職專班
cyc-htes0019@mail.htes.cyc.edu.tw

王昌斌
南華大學資管所
cbwang@mail.nhu.edu.tw

摘要

本研究探討以複習評量為前提的電腦適性化評量，目的在根據學生的表現提供合適的題目，並診斷學生的學習成果及提供教學上的建議。

在本文中以一組向量屬性來表示一道題目，這組屬性分成兩個部份，即主題概念和解題能力。首先以分數主題，從分數涵義的構面分成七個屬性，數學解題能力的構面則分成三個屬性，這十個屬性分別以模糊統計試驗及屬性空間測度的方式，利用教師對題目的敘述所隱含的概念及題目敘述方式加以判斷以獲得數值，然後再以模糊聚類分析及判斷屬性值強度的方式分別建立兩種題目群組以提供兩種選題策略的運用，以根據學生的表現提供不同難度或不同概念題目的個別化施測。

本研究期望以複習評量為前提下，建立評量題目可以自由增刪的電腦適性化評量系統，並且可適用於不同科目的設計。

關鍵詞：模糊理論、分數概念、電腦適性化評量

1.前言

1.1 研究背景及動機

隨著電腦的普及，將可利用電腦所提供之互動學習的指導，幫助學生自行學習或複習。因此，探討如何利用電腦做適性化動態評量將是一個值得研究的方向。在電腦適性化動態評量的探討，為何使用模糊理論，模糊理論可能有以下的特點：

一、模糊理論可用於電腦系統的設計。題目可在電腦中建立適合的群組以供選用。

二、模糊理論模仿人類的推理方式。在自然和社會現象中，差異往往要通過一個過渡的形式，而處於過渡中的差異，便具有“亦此亦彼”的性質，因此學生是否學會了？會到什麼程度？是否熟練了？熟練到什麼程度？其實都具有中間過渡的不明確性。因此從這種巨觀思維上，嘗試如何運用模糊理論。

三、從應用的角度思考。本研究則從應用的角度著手，不探討細節的內部認知歷程。概念與概念之間、概念與題目之間、題目與題目之間，有可能形成複雜的連結關係，因此運用模糊理論來處理系統的複雜性，將既有的題目大量並且直接的納入題庫中，並且允許評量題庫能自由的增刪題目，在此情況下建立動態評量的方法，根據學生不同的表現給予合適的題目，以判斷學生的學習狀況。

1.2 研究目的

本文實作一個分數的電腦適性化評量，評量六年級學生在分數方面的學習狀況，以作為補救教學或加強學習的參考。實施電腦適性化評量，能依據個別學生不同的表現給予不同評量題目，以判斷個別學生的學習狀況並提出建議，不僅節省老師在施測上的心力，施測後對每位學生都可加以指導，有些部分重新教學，有些部分指導學生自行學習，如此可不必全面性的複習，達成教師補救教學及學生強化學習上的效率。

1.3 研究限制

實驗的誤差至少考慮以下三項

（1）屬性值歸屬度的穩定性是否足夠

（2）題目難易程度的歸類方式是否恰當

（3）題目形式的多樣性是否充足

因此在研究過程的嚴謹性方面並不足夠，只是嘗試提出一個可行的方向。

2.文獻探討

2.1 電腦適性化評量

現今測驗專家們也正積極地發展電腦化測驗(CBT)。Ferraris（1991）認為電腦使用在教育上的

施測大約可以分為以下三大類：

一、電腦管理測驗(Computer Managed Testing，CMT)：檢查學生的答案、計分並計算有關的統計數。

二、電腦實施測驗(Computer Administered Testing，CAT)：電腦與學生直接對話以測驗學生所具備的知識，電腦不僅自動進行施測，並分析測驗的結果，且可根據學生的答題反應自行選擇試題，組織測驗的題目。

三、組織測驗(Test Synthesis，TS)：只要給予測驗的特徵（如水準、主題），電腦就會自動從資料庫中提取所需要的試題，自行組織試題。

在一般考試中如果學生考 60 分，通常只能知道學生還有 40％不會，卻無法得知究竟是哪些數學概念不會，如能讓教師知道學生還有哪些數學概念需要進行補救教學或加強，這樣的評量，就是一個成功有效的評量。

2.2數學解題歷程

Mayer(1993)討論解題的四個基本認知過程為：(1)問題轉譯 (2) 問題整合 (3) 解題計畫及監控 (4) 解題執行。根據以上解題的四個基本認知過程，如何將教師對題目的判斷轉為電腦可以處理的方式。對於「問題轉譯」的部份，可以判斷題目的表達方式是以數字符號為主或以文字描述為主。對於「問題整合」的部份，則探討分數概念內涵的分類，對於「解題計畫及監控」及「解題執行」的部份則由教師判斷這道題目的困難度有多高。

2.3 分數概念的探討

2.3.1 從分數的數學建構與應用涵義看分數概念

有不少學者分析分數在生活情境中所表現的意義，Ohlsson（1988）及 Kieren（1988）認為分數有四種建構：商的函數、有理數、二元向量、合成函數。

（1）分數是商的函數：分數基本的性質是兩個數之間除的關係。

（2）有理數：x/y 表示一個對象（object）或代表商的函數的一個不變值，例如 3/6，2/4，1/2，代表相同的值。

（3）二元向量：x/y 表示兩個量之間的比較關係。

a. 比（ratio)：兩個量之間的比較關係，如班上男生、女生的比是 21：16，比值為 21/16。

b. 比例（propotion）： 表示兩個比之間的等值，如速率是一個量與時間的比，p/q 表示為一速率時，代表兩種量數的函數關係：時間增加工作量也增加。

（4）合成函數：若某數 x 乘以 p/q，表示某數 x 經過兩次變化：先增加 p 倍在縮小 q 倍。

2.3.2 從單位概念的發展與分數的學習看分數概念

Steffe認為兒童在形成分數概念時，必須形成相當有彈性或變通性的單位概念。

Saenz－Ludlow,Adalira 認為兒童先具有彈性運用複合單位以解決整數問題之後才建構分數概念。分數概念的發展則經過三個水準，每個水準建構一個基模。

（1）測量的部份整體基模（metric part－whole scheme)：能將一個不可數的連續量如長度、面積、容積等看成一個整體，並將整體等分成幾個部份，理解部分與整體的關係，1/4 是等分成4份中的1份

（2）多重等分的協調基模（multiple－partitioning coordinating scheme)：

一個量可以作不同的等分，例如 5000 元的四十分之三十九大或是百分之九十九大的問題。能比較抽象的分數的大小。例如 2/7 和 2/14 那一個大，能直接說 1/7 是兩個 1/14，所以 2/7 比較大。

（3）部分等分的協調基模（part－partitioning coordinating scheme）

一個量可以等分成幾個部份，每個部份又可以等分，例如用所有錢的 3/10 買菜，用剩餘錢的 1/4 買蛋，，這兩個分數是不同的單位。

2.3.3 從教學單元看分數的概念

（1）分數概念和分數加法過程

a. 明白變動的基數「1」

b. 有一對多和比例的概念。

c. 簡易分數的倍數概念及合併。

d. 有擴分的操作概念和能夠比較分數大小

e. 合一過程，例如3/8＋5/8=1或1/8＋2/8＋5/8＝1。

（2）分數乘法的基本知識

a. 8×（1/4）也是8÷4有除的概念。

b. 8×（1/4）是8個（1/4)，可用單位分數操作。

c. 8×（1/4）=8÷4×1是比例的概念。

d. 8×（1/4）相當於每個1抽出1/4。

e. 12×（3/4）把12放大了3倍在縮小4倍。

綜合以上的探討，把分數題目所隱含的概念列舉如下：◎除法　◎比的概念　◎單位分數◎等值分數◎比例推理 ◎應用相同的基數「1」◎ 應用不同的基數「1」

2.4 模糊理論的探討

2.4.1 模糊集合

令論域 U=｛$x_1, x_2, \cdots, x_n$｝

模糊子集 A=$\sum_{i=1}^{n}\frac{\mu_A(x_i)}{x_i}=\frac{\mu_A(x_1)}{x_1}++\cdots+\frac{\mu_A(x_n)}{x_n}$

亦可用向量形式表示模糊子集，但歸屬度為 0 的項不能略去。

A=（（x_1，$\mu_A(x_1)$），，⋯（x_n，$\mu_A(x_n)$））

或 A=（$\mu_A(x_1), \mu_A(x_2), \cdots, \mu_A(x_n)$）

以上所描述的模糊集合的表示法是在一個論域上的表示法。在『模糊集合於學習成效分析之應用』這篇論文中，對於一道題目的表示則是在五個論域上分別給予歸屬程度，將每道題目分成五個學習概念。即使兩個人的總分相近，因為在不同的學習概念有不同的得分，可解釋他們在不同的學習概念了解的程度不同，這種對於題目給予向量屬性的表示方法是本研究思想上的啟發。

2.4.2 歸屬函數的建立

本文中歸屬函數以統計試驗和屬性測度加以確定。

一、模糊統計試驗

在機率統計中，事件 A 發生的機率 P（A）定義為

$$P(A)=\lim_{n\to\infty}\frac{\text{A發生的次數}}{n}$$

實際應用中，只要 n 求足夠大就可以了

與此類似定義模糊統計試驗

$$\mu(\mu_0)=\lim_{n\to\infty}\frac{\mu_0\ \text{屬於}\ A^*\ \text{的次數}}{n}$$

A^*是模糊子集

模糊統計試驗具有歸屬頻率穩定性的特點。

若一道題目中某一概念，在 10 人中有 8 人勾選此一概念，則此一概念的歸屬數值為 0.8。

二、屬性測度

例如設 F 為消費者對某種商品的滿意度，把 F 分為五類，C_1＝｛很滿意｝C_2＝｛滿意｝C_3＝｛中等｝C_4＝｛不滿意｝C_5＝｛很不滿意｝，這五類彼此不相交，｛$C_{1,}C_{2,}\cdots C_5$｝稱為屬性空間。

x∈C_i 表示 x 具有屬性C_i，，記為 $\mu_x(C_i)$，屬性測度在$[0,1]$之間取值。須滿足以下的規則。

1. $\mu_x(C_i)$≧0，i=1～5
2. $\mu_x(F)=1$
3. , $C_i\cap C_j=\psi(i\neq j)$，$\mu_x\left(\bigcup_i C_i\right)=\sum_i\mu_x(C_i)$

即消費者對某種商品的滿意度，在屬性空間

｛$C_{1,}C_{2,}\cdots C_5$｝中$\sum_{i=1}^{5}\mu_x(C_i)=1$

對於題目中表示強度的屬性 □1 □2 □3 □4 □5 可使用 p_1×1＋ p_2×0.75＋ p_3×0.5＋ p_4×0.25＋ p_5×0。獲得數值。此時 $p_1+p_2+p_3+p_4+p_5=1$（p_i≧0，i=1～5）符合屬性測度的規則。

2.4.3模糊理論在選題策略上的應用

2.4.3.1 題目向量屬性值的測度

綜合上述文獻的探討，將題目向量的屬性，分成兩個部份，即（分數概念，解題知識能力）

例如，某人對以下兩道題目所勾選的選項如下

1.（　）兩個真分數相乘，所得的積必為　①整數②真分數③假分數④帶分數。

（以下二個選項為二選一）

□應用相同的基數「1」　■ 應用不同的基數「1」

（以下選項可以複選）

□除法　□比的概念　■單位分數 □等值分數 □比例推理

題意表達方式：□1 □2 □3 □4 ■5

解題策略：　□1 □2 ■ 3 □4 □5

計算難度：　■ 1 □2 □3 □4 □5

2.（　）$9\times\frac{5}{8}$＝　① 9×5×8 ② 9÷5×8 ③ 5÷8÷9 ④ 9×5÷8。

（以下二個選項為二選一）

■ 應用相同的基數「1」　□應用不同的基數「1」

（以下選項可以複選）

■ 除法 ■ 比的概念 ■ 單位分數□等值分數 □比例推理

題意表達方式：■ 1 □2 □3 □4 □5

解題策略：　□1 □2 ■ 3 □4 □5

計算難度：　■1 □2 □3 □4 □5

解釋如下

題意表達方式數字愈小表示主要是數字符號，數字愈大表示題意表達方式主要是文字。

解題策略和計算難度部分數字愈大表示愈難。

第一題純為文字說明；真分數即有單位分數的概念；了解的方式可用實際數字來驗算。真分數的乘法中第二個真分數以前一個真分數為整體的「1」，故本題的題意中有兩個不同的基數「1」。

第二題純為數字符號；$9\times\frac{5}{8}$ 可視為 9 個 $\frac{5}{8}$，有單位分數的概念；又可視為一條八格的積木，其中五格有記號，如此相同的積木共有九條，總共有 9x5 個有記號的格子，這是比的概念；9x5 個有記號的格子可形成（9x5）÷8 條積木，有除法的概念。兩題皆不需實際運算。

向量屬性值在不同題目間可能會有明顯的差異，而並非同一道題目在多數人之間產生很大的歧異。

2.4.3.2 利用模糊聚類分析將題目作分群

聚類分析就是使用等價關係把原來的集合分成互不相交的子集，不同的子集就相應不同的類。Fuzzy 聚類分析的步驟首先要建立 Fuzzy 相似關係，然後構造 Fuzzy 等價關係。

（一）建立 Fuzzy 相似關係

設為 U=｛u_1，u_2，u_3、、、u_n｝待分類的全體

每一待分類物由一組資料表示，$u_i=$｛x_{i1}，x_{i2}、、、x_{im}｝如何建立之間的相似關係，有許多方法：

本文中使用數量積法

$$r_{ij} = \begin{cases} 1 & i=j \\ \dfrac{1}{M}\sum_{k=1}^{m} x_{ik}\cdot x_{jk} & i\neq j \end{cases}$$

（二）構造 Fuzzy 等價關係

由第一步得到的矩陣 R 為相似矩陣一般只滿足自身反性和對稱性，需將 R 改造成 Fuzzy 等價矩陣。若 R 滿足以下規則，稱 R 為 Fuzzy 等價矩陣。

1 自身性　$r_{ii}=1$　即 I⊆R 或 R（u，u）=1

2 對稱性　$r_{ij}=r_{ji}$　即 R= R^T

3 傳遞性　$r_{ij}\geq\bigvee_{K=1}^{n}\left(r_{ik}\wedge r_{Kj}\right)$　即 R∘R⊆R

相似矩陣 R 不一定滿足傳遞性，此時利用

$R^2=R\circ R=$〔$r_{i'j'}$〕=〔$\bigvee_{K=1}^{n}\left(r_{ik}\wedge r_{Kj}\right)$〕。如果

$R^2=R$，則 R 為 Fuzzy 等價矩陣，不然則繼續檢查是否 $R^4=R^2$，以此類推下去。在得到題目間的相關係數並形成 Fuzzy 等價矩陣後，便可以根據 λ 值分成數個群聚，從各群聚中出題。使得選出的題目間有較大的差異性。

3 研究方法

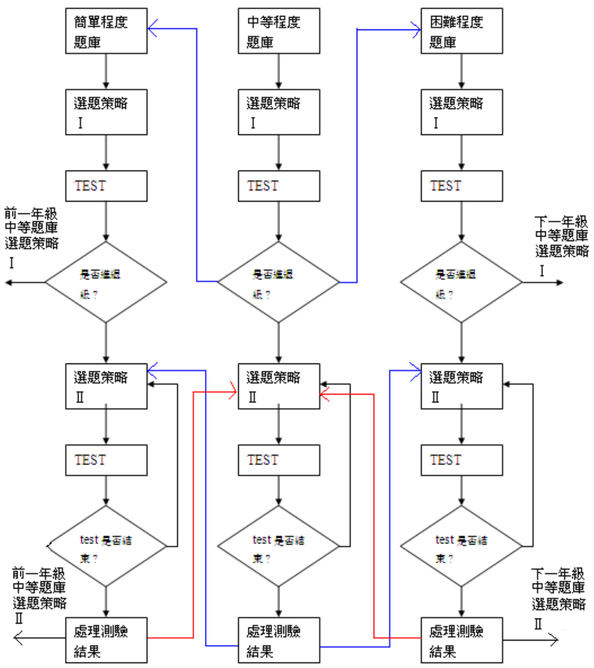
3.1 問卷設計

在填答問卷前需先作問卷說明，本研究的問卷說明除了簡短文字說明並輔以例題說明。

3.2 選題策略的探討

3.2.1 選題流程圖

這是以年級為單位的選題流程



3.2.1選題流程說明

（1）建立題目群組

a. 配合選題策略 I 的題目群組 I

簡單題目題庫	相同的基數「1」	模糊聚類分群
	不同的基數「1」	模糊聚類分群
中等題目題庫	相同的基數「1」	模糊聚類分群
	不同的基數「1」	模糊聚類分群

困難題目題庫	相同的基數「1」	模糊聚類分群
	不同的基數「1」	模糊聚類分群

題目先依難易度分類，再根據是否相同的基數分類，最後根據五個分數概念作模糊聚類分群。

選題策略Ⅰ：在某一難度等級中，由模糊聚類分析的不同群聚中選題，使題目間有較大的差異以作為進退級的考量。

b. 配合選題策略Ⅱ的題目群組Ⅱ

簡單題目題庫	相同的基數「1」	各概念群組
	不同的基數「1」	各概念群組
中等題目題庫	相同的基數「1」	各概念群組
	不同的基數「1」	各概念群組
困難題目題庫	相同的基數「1」	各概念群組
	不同的基數「1」	各概念群組

題目先依難易度分類，再根據是否相同的基數分類，各概念群組表示某一概念屬性值超過一定閾值的題目即可聚成一群。同一道題目有可能出現在不同的概念群組。

選題策略Ⅱ：針對某一概念作檢測，表現良好則進級至答對率降至某一閾值終止測驗，表現不佳則退級至答對率升至某一閾值終止測驗。

（2）學習者資料庫

	以數字符號表達為主（歸屬度低於0.2）	以文字表達為主（歸屬度高於0.8）	解題策略	計算難度	紀錄已測驗過的題目編號
答對率					

用來了解解題能力的表現及避免出現相同題目

4.預期結果與展望

一、與一些有關選題策略的論文直觀上的相比，本研究不試圖研究概念與概念之間、概念與題目之間、題目與題目之間，有可能形成複雜的連結關係。只要能大概判斷學生知識概念不會的範圍，或不熟悉的程度或大致判斷解題能力的狀況就能採取適當的教學措施。

二、題目給予一組向量屬性的好處

（一）在題庫中可自由的增刪題目，關注的焦點在題目本身，隨時新增的題目根據自身的向量屬性在題目群組Ⅰ和題目群組Ⅱ中分別加入適當的群組。

（二）題目藉由向量屬性來分類後，不僅可用來評量亦可用來教學。因此最重要的是對學科領域知識的分析才是主要關鍵，如何去找出合適的概念分類成為題目的向量屬性，然後才是借用模糊理論的觀念和方法加以應用。

三、可通用於其他科目複習評量的設計。例如在社會科中主題概念的部份則探究學科實際內容的概念，問題解決能力的部份則 以 Bloom 的四個知識向度:1. 事實知識 2. 概念知識 3. 程序知識 4. 後設認知知識。來表示，如此便可仿照本文的方式加以設計。

參考文獻：

涂金堂、林佳蓉 著，「如何協助學生解決數學應用問題」，高雄復文圖書出版社，民國 89 年。

郭信川、李朝宗，「模糊集合於學習成效分析之應用」，國立台灣海洋大學系統工程暨造船學系。

馮國臣 著，「模糊理論基礎與應用」，新文京開發出版股份有限公司，民國 96 年。

程乾生、吳柏林，「模糊統計分析的數學原理及其應用」，量化研究學刊，第一卷第一期，2007。

劉秋木著，「國小數學科教學研究」，五南圖書出版公司，民國 85 年。

鄭振初著，「分數教學分析：概念和運算」，九章出版社，民國 95 年。

Behr,Harel,Post, ＆ Lesh（1992）：Rational number ,ratio, and pro-potion. in Grouws,D.A.(1992,ed) ：Handbook of research on Mathematics Teaching and Learning acMillan

Mack,N.（1990）, Learning Fractions with Underatanding ：Building on informal knowledge. Journal for Research in Mathematics ducation.

Steffe,L.P.（1990）：Adaptive Mathematics Teaching. in Cooney＆Hirsch（1990）Teaching and Learning Mathematics in the 1990s

Steffe,L.,＆Olive,J.（1991）.The problem of fractions in the elementary school. The Arithmetic Teacher,38

附件二

以知識概念架構為基之數學能力診斷方法

王昌斌
南華大學資訊管理學系
cbwang@mail.nhu.edu.tw

張英燦
南華大學資訊管理學系
s2222012@mail.cy.edu.tw

摘要

一個好的學習診斷方法，不但要能夠有效地評估學習者的學習成效，更要能診斷學習者的學習障礙與迷思概念，才能幫助學生突破學習障礙，進一步改善學習效能。

評量測驗在學習過程中是一個重要的步驟，據此可以評估學生的學習效能以及學生的學習障礙。總結性評量是用來評估和總結學習者在某一特定時間的發展狀況，老師無法針對學習者所得到的成績給予真確的建議。因此本研究提出一個以知識概念架構為基礎之學習診斷方法，來評估學生的學習成效。

關鍵字：學習診斷方法、學習障礙、迷思概念、知識概念架構

A Diagnosis Method of Mathematics Ability base upon Knowledge Concept Construction.

Chin-Bin Wang
Department of Information Management, Nanhua university
cbwang@mail.nhu.edu.tw

Ying-tsan Chang
Department of Information Management, Nanhua university
s2222012@mail.cy.edu.tw

Abstract

A good studing diagnosis method not only can evaluate learners' study result effectively, but also can diagnose their learning disability and misconception, helps learners to break through the learning disability and improvement the learning result.

Assessment test is an important step in the learning process which evaluate the Study potency and learning disability. Summative assessment refers to the assessment of the learning and summarizes the development of learners at a particular time. However, teachers can not give valid suggestions to learners from their test scores. In this paper, a learning diagnosis method based upon the knowledge concept construction is proposed to evaluate the learning efficiency of students.

Keyword: studing diagnosis system, learning disability, misconception, knowledge concept construction

壹、前言

教學過程中，為了確認學生的學習成效，大量採用形成性及總結性的紙筆測驗，除了評鑑學生學習行為表現依據，並據此判斷學生的成績和等第。這樣的測驗方式，雖然能達到能力評定的功能，但對於進一步提供補救教學或加強，作用卻不明顯。在教學現場上，受限於時間與人力條件，教師難以診斷個別學生發生困難之所在，給予適當且有效的支援。學習者在學習過程中，經常因為模糊地帶未得到釐清，導致學習困難的情況不斷循環發生，進而影響後續的學習。因此，若能有效地推測出學習認知能力，找出學習者學習歷程上缺乏或迷思的節點，提供較佳之學習路徑給相關人員參考，有效地給予協助、補救，藉此作法有效改善學習者的迷思概念，提升其學習效能與信心。

近年來，資訊科技及網際網路的快速發展，各項資訊化的教學概念以及設備引進，增加了課堂活動的多樣性。除了傳統教學模式的改變，許多專家學者也投入多元化教育資訊輔具的相關研究，在教學現場提供其他的支援。例如電腦輔助教學系統（computer-assisted tutoring system）及電腦輔助測驗系統（computer-assisted testing system）（何榮桂，1997；許慶昇、杜淑芬、黃國禎，1998；張紹勳，1998）。電腦輔助診斷是現代測驗理論的主流，傳統的測驗方式，大多在施測完成後，產生形成性或總結性指標之後，便完成測驗的流程。對於學生的迷思概念或遺漏認知，也常因為課程進度的壓力、教師人力的不足而被忽略帶過，並沒有藉該次診斷的內容，將後續診斷、補救的工作完成。運用電腦輔助進行學習診斷，能有效率的輔助教師對施測結果進行錯誤模式的推斷，為教師在教學上提供實際且效能的協助。

灰色理論（Grey system theory）相較於其他的統計方法，該理論對於系統模型之條件屬於不完整、不確定、多變量輸入、離散的數據資料的情況，能做有效的處理，提出解決之方法（翁慶昌，陳嘉欉，賴宏仁，2001）。因應學生測驗作答的實際狀況，作答結果屬於少條件、少資訊的情形，本研究提出一應用灰理論之學習評量模式，希望能以一般形成性或總結性評量的題庫，在少量測驗題數的限制下評估受試者之能力值，整合主觀與客觀的受試者評量資訊，作為教師評量學生之學習成效工具。

貳、文獻探討

一、學習評量

為了增進學生學習效能，達成課程目標，教學目標、教學方式與教學評量需同時計劃，才能密切配合，達成完整結合。教學評量在整個基本教學模式（The Basic Teaching Model）中，具有回饋和統整活動的功能（Glaser, R. , 1962），詳見圖 2。

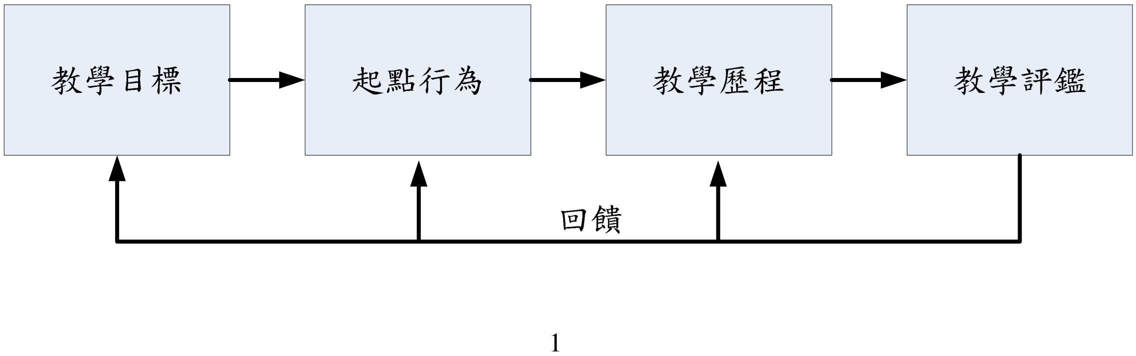


圖2 Glaser R.基本教學模式

教師根據評量的回饋，適切的修正教學目標、教學活動；學生可以修正其學習錯誤、診斷其學習方法、或增加其學習動機和學習成就，不僅能幫助學生達成學習目標，對於學生面對學習挫折的心理輔導也會有極大助益。

二、迷思概念

根據Piaget 與 Inhelder的看法(1969)，認知就是獲得知識(knowing)，而認知發展就是在獲得知識的過程。人類的學習歷程，來自於逐步建構其運作所需之結構，過程之中包含認識、理解、思考、問題解決、學習、概念化，分類及記憶。知識來自學習者與環境間互動的歷程，並在過程中累積所需的智識與能力。一個概念的形成完備，是一段新、舊經驗融合歸納歷程的結果，過程中學習者不斷嘗試將新的訊息和觀念，與已存的知識、經驗進行連結。

如果學生在進行學習之前，其前備知識(prior knowledge)，和專家所持有的概念形成及學習路徑並不一致，學習者的概念是不完整，甚至是不正確的，這些既有錯誤的概念將導致學生的學習成效不彰，也會妨礙另一概念的形成，稱為迷思概念(misconception)(Clark & Peterson, 1986)。迷思概念是蘊藏在內的知識能力，學生一旦產生迷思概念，容易在後續學習上造成困擾或障礙，無法達成預期的學習效果。這也更顯示了迷思概念對學習的影響；因此對於迷思概念的診斷及發覺，其重要性可見一斑。

三、知識地圖的建構

當我們思考知識獲得的歷程時，我們會發現，一項知識或技能的習得，往往需要許多先備知識或經驗作為基礎，這樣的進程與Piaget和Inhelder(1969)的認知發展論是一致的。完成某項主題的學習時，歷程中重要的節點，便稱之為「學習概念」(Learning Concept)。

學習活動進行中，發現學習歷程中有其一定的順序及路徑，如果依照概念的先後順序進行學習，將形成較佳學習路徑，我們將這樣的學習順序稱為「認知順序」。概念與概念之間，有其上下位關係或關聯性，我們將其關係用圖表具象化，稱之為概念構圖(Concept Mapping)(如圖3)。

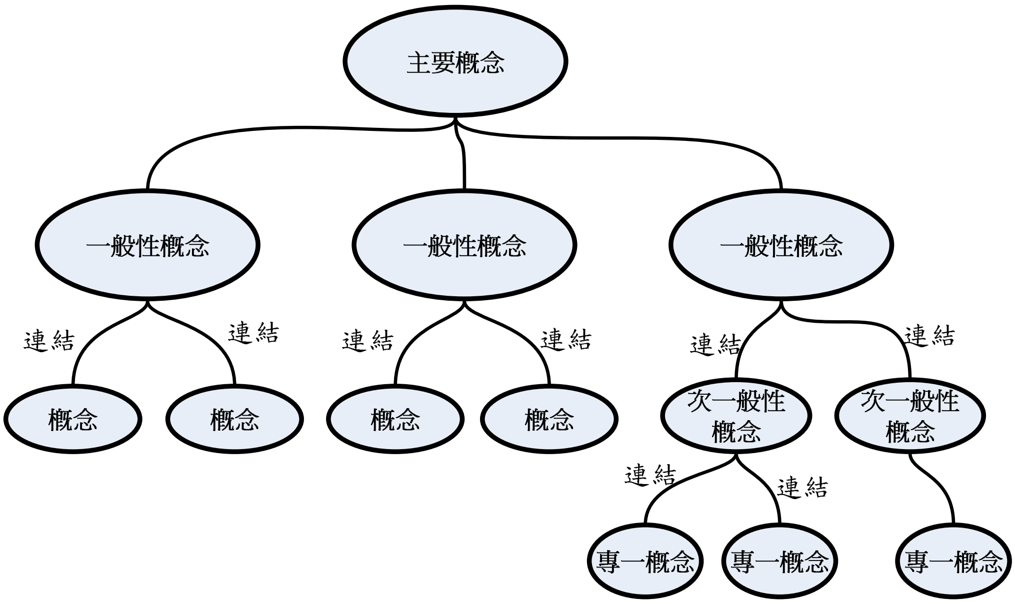


圖3 概念構圖

教師在課堂進行時,運用專家認定的學習概念圖為基礎,可具備兩大功用(黃達三,2006):一、教學的前置組織(Willerman & Mac Harg, 1993)。二、學生的先備知識的檢測(Wandersee, 1990)。教師的教學內容若能植基於學生的先備知識,那麼學生才要能進行有意義的學習(meaningful learning)。

四、模糊德菲法

模糊德菲法(Delphi method)亦稱為為專家法,它是利用專家集體智慧來確定各因素在評判問題或決策問題中重要程度係數的有效方法之一。若考量以反覆式的問卷調查,其彙整過程的時間將會過長,不符合時效,因此,Ishikawa, A., M. Amagasa, T. Shiga, G. Tomizawa, R. Tatsuta 與 H. Mieno 等人於 1993 年提出 Max－Min 法與模糊整合(fuzzy Integration)法,來解決此項的缺點。這樣的整合方法具有建構容易與運算簡單之特性,使之達成共識及一致性時,不但能節省調查時間與成本,而且能實際反應出專家之意見。

$$\tilde{W}_i = (\alpha_i, \beta_i, \delta_i), i = 1,2,3...k \quad (1)$$

$$\alpha_i = Min\{\alpha_{mi}\}, m = 1,2,3...n \quad (2)$$

$$\beta_i = \left[\prod_{m=1}^{n} \beta_{mi}\right]^{\frac{1}{n}}, m = 1,2,3...n \quad (3)$$

$$\delta_i = Max\{\delta_{mi}\}, m = 1,2,3...n \quad (4)$$

其中 $\tilde{W}_i$:模糊權重,

K :概念的數目

k :專家數

$\alpha_i,\beta_i,\delta_i$：三角模糊數的左端點、頂點及右端點

mi：第 m 個專家對第 i 個概念的看法及意見

接著採用重心解模糊化法來求得每一概念之權重數值：

$$W_i=\frac{\alpha_i+\beta_i+\delta_i}{3} \qquad (5)$$

W_i：第 i 個概念解模糊化後之權重值。

$\tilde{W}_i(\alpha_i,\beta_i,\delta_i)$：整合專家看法後的模糊權重值。

五、灰色理論與灰關聯度

中國大陸學者鄧聚龍教授於 1982 年提出灰色理論。灰色理論主要是針對系統模型的訊息不完全、關係不明確，進行關聯分析、模型建構、並藉著預測及決策的方法來探測並瞭解系統狀況（温坤禮、黃宜豐、張偉哲、張廷政、游美利、賴家瑞，2003）。灰色理論應用之範圍非常廣泛，主要是對於事件之間關係不明確，以及多變量之輸入，離散型之數據，及系統數據不完整等狀況，做有效率之處理。

灰關聯分析主要是透過所得數據中，將各條件參數進行關聯性的運算，由這些已知條件找出不明確之訊息或規則，進一步釐清參數間之互動關係。藉由灰理論中灰關聯分析的計算，可求得各因子序列之灰關聯度，此灰關聯度即代表各因子與該研究主題（參考序列）之接近程度，藉由灰關聯度之比對結果來求得與目標函數或期望值之關聯程度。

在灰關聯空間$\{P(X);\Gamma\}$中，如果有一組序列：

$x_i=\left(x_i(1),x_i(2),...,x_i(k)\right)$

其中$i=1,2,...,m$，$k\in N$（正整數）

根據局部性灰關聯度定義：灰關聯係數$\gamma(x_0(k),x_i(k))$為

$$\gamma(x_0(k),x_i(k))=\frac{\Delta_{\min}+\varsigma\Delta_{\max}}{\Delta_{0i}(k)+\varsigma\Delta_{\max}} \qquad (7)$$

其中$i=1,2,...,m$，$k=1,2,...,n$

其中x_0為參考數列，x_i為一特定的比較數列。

$\Delta_{0i}=\left\|x_0(k)-x_i(k)\right\|$：$x_0$和$x_i$之間在第$k$個差的絕對值。

$\Delta_{\min.}=\overset{\min.}{\forall_j}\in i\overset{\min.}{\forall}k\left\|x_0(k)-x_j(k)\right\|$：代表所有數列間差距之最小值。

$\Delta_{\max.}=\overset{\max.}{\forall_j}\in i\overset{\max.}{\forall}k\left\|x_0(k)-x_j(k)\right\|$：代表所有數列間差距之最大值。

ς：辨識係數：$\varsigma\in[0,1]$（辨識係數的值可以依實際需要調整大小，一般均取 0.5 之值）

當求得數列各點之灰關聯係數，一般取灰關聯數的平均值為帶比較數列與參考數列間相對之灰關聯度。

六、學習錯誤類型

學習過程中，因為語言、基模、策略、程序知識的因素影響，產生了錯誤解題步驟，Stefanich & Rokusek（1992）指出：系統性錯誤是來自於穩定的錯誤、遺失技能知識、不完全或誤導學習而產生會持續出現在相類似的問題中，是對特別演算產生不正確反應。在數學解題過程中產生的錯誤步驟，依其出現錯誤的關鍵處作分類，分成幾種類型稱為錯誤類型（Kathlen，1987）。

在樣本空間中，將因產生學習障礙的成因，收斂成具錯誤類型關聯性較高之群組。藉由重要錯誤類型的比對，配合各個試題的內涵指標，便能協助快速而有效的掌握學習者的概念發展。經由錯誤類型特徵值的比對，我們可以進一步掌握學生的學習狀況，深入其學習歷程的問題。藉由診斷測驗工具的施行，收集學生的應答狀況並加以分析歸納，會發現學生的錯誤型態並非沒有關聯。教育測驗專家也認為，認知心理學和心理計量的發展應用於教育測驗，能更精確地診斷出學生的學習錯誤，於是發展出「認知診斷評量」之構想（Nichols, 1994; Snow & Lohman, 1989）。

参、研究方法與設計

一、研究方法

目前診斷技術中，只取得受試者對該試題的作答正確與否，無法對受試者的迷思概念提供真確指出，後續的補救教學也因此難以有效的規劃施行。以知識結構理論為基礎的兩階段題組式診斷模組，在同一診斷工具的檢測流程中，同時分析學習者的概念認知以及算式操作，以節省施測的時間，並提供個別學習診斷分析，讓學生可以立即知道自己的錯誤觀念，也有利教師進行補救教學，在可達成之時間與人力條件限制下，提升教學成效。

二、研究架構

依據研究目的，本研究之研究架構，分成：學習概念分析、編製概念診斷工具、學習概念診斷等三大部分進行，其詳細研究步驟可分成下列步驟進行（詳如圖 5）。

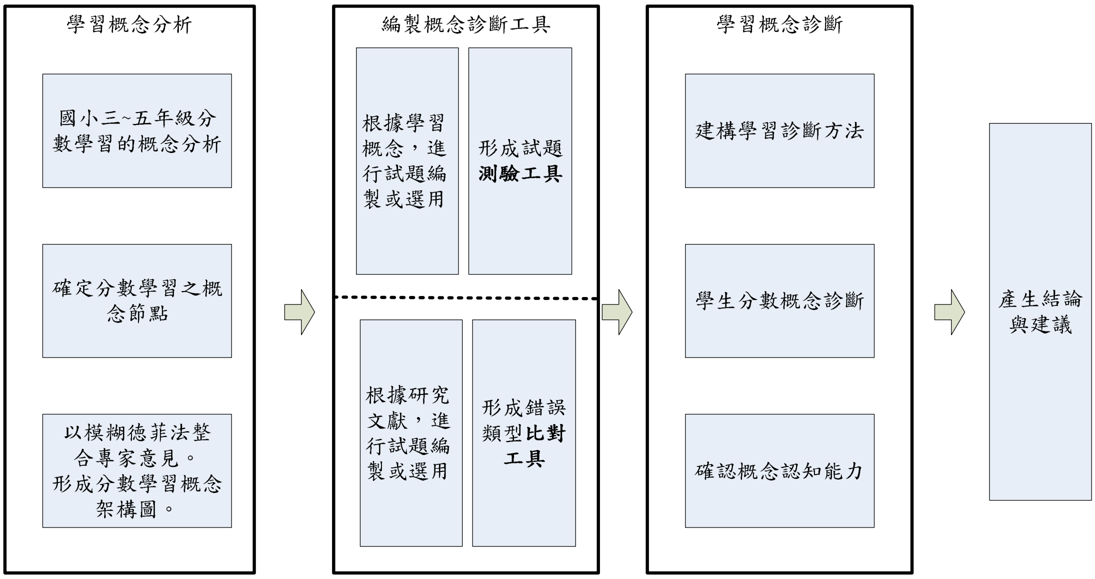


圖5 研究架構圖

本研究之研究架構，分成：學習概念分析、編製分數概念測驗工具、學習概念診斷三大部分進行，其詳細研究步驟可分成下列步驟進行：

（一）學習概念分析階段：

一、相關文獻探討

二、就診斷主題，進行教材概念初步分析

三、與擔任教學之教學人員進行討論，確認學習者通過測驗所需之關鍵概念與能力。

四、初步分析之各項概念，再參考相關研究之經驗，並委請該領域專家確認，分列出學習的關鍵概念與關鍵能力。

五、採用專家模糊德菲法，對學者專家及教學人員進行訪談與問卷調查，確認該學習主題概念關聯情形，產生概念關聯圖，作為後續步驟之基礎。

（二）編製概念診斷工具：

根據學習單元之主要概念，以及專家知識結構，進行試題的編製與篩選。

一、進行試題分析與修正。

二、進行組卷以利紙筆測驗的進行。

三、根據分數概念關聯圖及資料分析，製作出試題測驗工具。

四、參考相關文獻及實務經驗，進行常見錯誤類型之整理。

五、建立常見錯誤類型之屬性分析表，作為錯誤類型比對工具。

（三）學習概念診斷

一、診斷推論：以灰關聯法則，將學生的答題狀況與常見錯誤類型進行比對，藉此推算學生的概念認知狀況。

二、將學習狀況分為概念學習及解題技能兩部份，進行兩階段診斷程序（詳見圖 6），進一步確認其迷思概念之類型，診斷出其迷思概念或必需能力之缺乏。

三、提出補救學習建議路徑或是需加強的單一概念。

四、提出研究結論與建議。

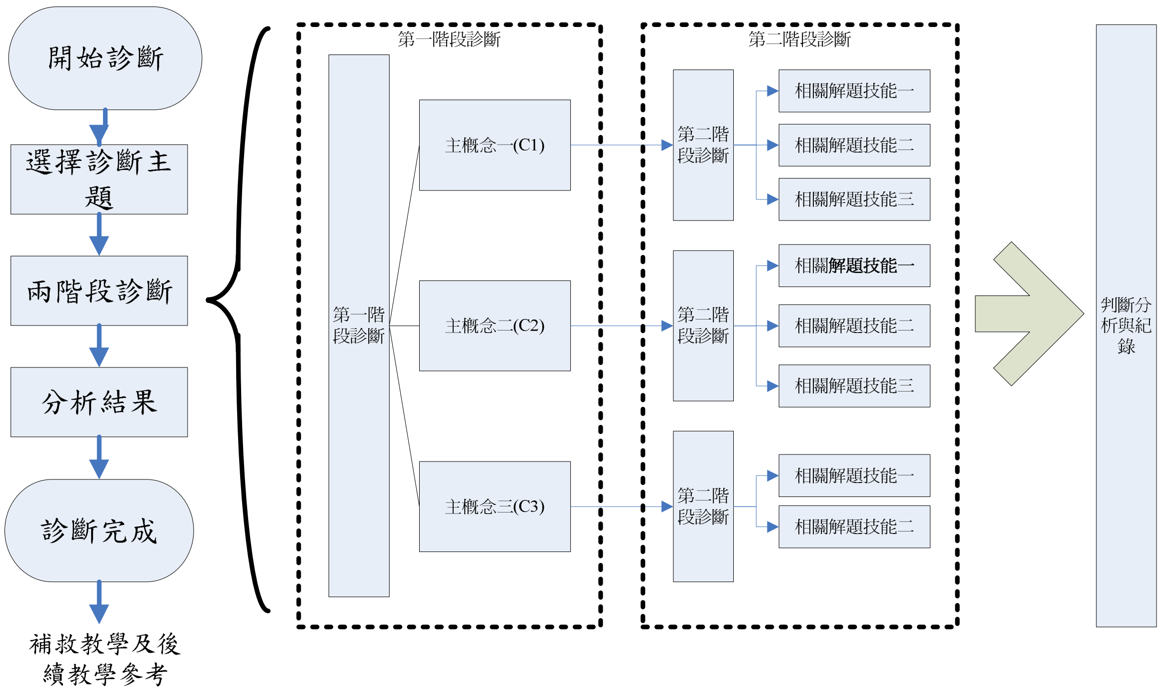


圖6 兩階段診斷評量流程圖

肆、方法實作

本章針對所提出之診斷流程方法做說明，並以國小三到五年級所學習之「分數加減」為診斷範圍，對以學習過該單元之學童進行診斷實作。整個診斷過程包括：

一、概念分析與知識概念架構圖的構建

（一）確定「分數加減」學習路徑之主要概念節點

本研究是以知識概念架構圖為基礎，經由文獻資料的整理，並透過與領域專家的訪談，佐以專家的看法，分列出完成該主題學習時，所需完成之學習概念，作為建立學習概念架構圖之節點依據（詳如表 2）。

表 2 「分數加減」學習路徑之主要概念節點說明表

概念編號	概念名稱	概念說明
C_1	單位量	當為了獲得一個與一待測定量相等的量時，一個已知量，或是此單位的一個被等分割部分，可重複累積複製此被界定量（甯自強，1998）
C_2	等分概念	等分是指將物品（連續量或離散量）細分，而細分的每個部分的量皆相等。（詹婉華、呂玉琴，2004）
C_3	單位分數	分母是大於 1 的正整數，而且分子是 1，這樣的分數被稱作單位分數，例如：$\frac{1}{2}$，$\frac{1}{4}$。Piaget, Inhelder, & Szeminska（1960）發現，孩

		童在處理和長度、面積等有關的分數問題時，先會處理 1/2，其次依序是 1/4、1/3、1/5、1/6。
C_4	分數概念及記錄方法	認識真分數、假分數、帶分數，包含： 一、 念法 二、 整數型的分數寫法 三、 認識分數在離散量的意義的能力 四、 認識分數在連續量的意義的能力 五、 分數在部分/全體的使用的能力
C_5	等值分數	一個數可以用無限的分數方式來表示，例如：$\frac{1}{2}=\frac{2}{4}=\frac{3}{6}\ldots$等等，這些不同的表示，便叫作這個分數的等值分數（Vance, 1992）；等值分數的特性就是部分可以再細分，部分可以再合併，等值分數既需要部分/全部的保留概念，也需要乘、除法的倍數觀念，可是乘除法運作的結果，並不會造成「量」的改變(引自詹婉華、呂玉琴，2004)
C_6	假分數、帶分數的轉換	當單位分數內容物為單一個物時，能認識假分數意涵並進行合成、分解活動。藉由 $\frac{n}{n}$ 與 1 的比較活動 認識帶分數，進一步了解同分母帶分數的合成、分解問題，並完成假分數和帶分數的互換程序；當單位分數內容物為多個個物時，能認識假分數、帶分數，藉由 $\frac{n}{n}$ 與 1 的比較活動，轉化假分數和帶分數的互換過程，完成同分母帶分數的合成、分解問題。
C_7	分數的大小比較	同分母分數的比較、異分母真分數的比較：經由通分過程，將待比較之分數，轉換為基準單位分數量相同（同分母）之等值分數，藉此轉換過程，進行分數大小之比較。
C_8	分數的加法	將分數調整成基準單位分數量相同（同分母）之等值分數，進行分數之加法計算。
C_9	分數的減法	將分數調整成基準單位分數量相同（同分母）之等值分數，進行分數之減法計算。

資料來源：本研究整理

（二）訪談並整合專家意見

收集專家意見，藉由領域專家對「分數概念」及「分數加減」單元的教學經驗，決定各元件節點內的概念權重大小，並將問卷訪談結果，以五等量模糊評估值（非常相關、大致相關、相關、不相關、非常不相關）（如圖 7）來表示轉換成模糊評估值。將模糊語意以模糊數參數化表示，且利用重心解模糊化法（式 6）來整合各個問項的語意值。

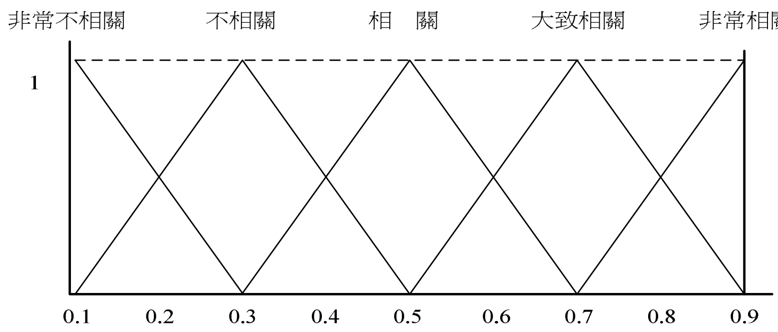


圖7 本研究模糊語意變數之三角模糊數

本研究採用模糊德菲法來整合這些意見，使之達成一致性，可將學生在國小階段學習分數加減時，主要概念的學習架構，形成國小階段分數加減學習的概念架構圖。本研究透過問卷調查，其中任意上下兩層之間存在包含的關係。經訪談及修正後，綜合並整理結果如圖 8 所示：

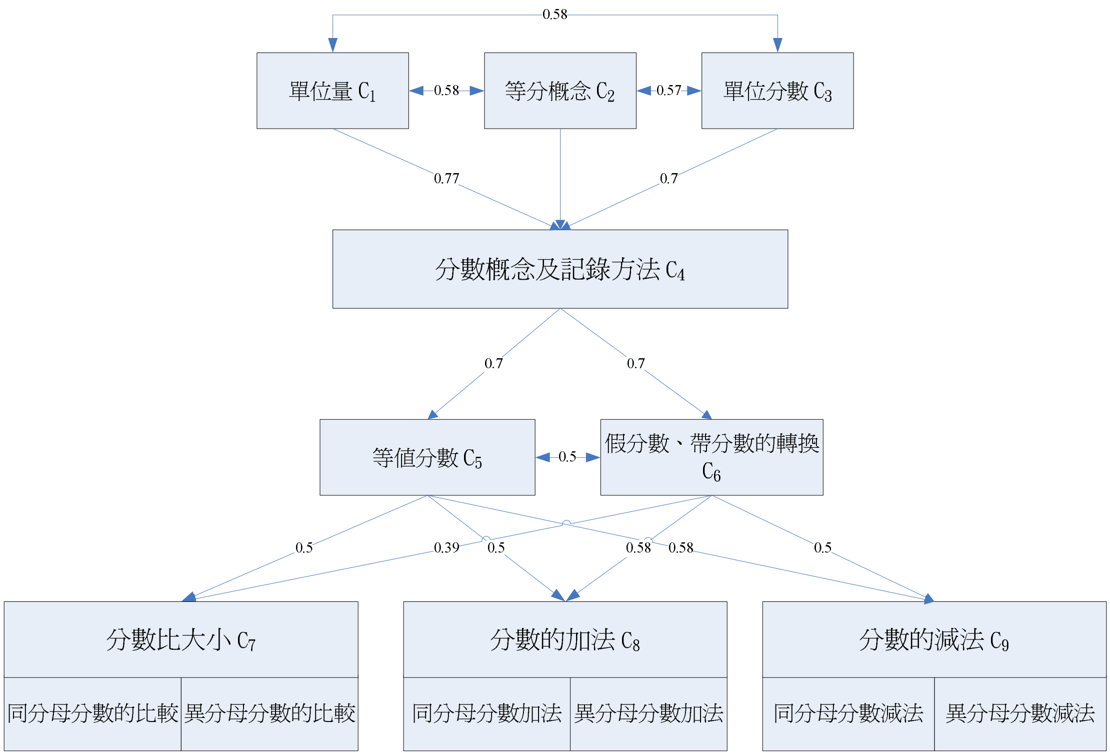


圖8 分數加減學習的概念架構圖

資料來源：本研究整理

二、建立試題測驗工具

我國國小課程教材是根據九年一貫數學課程綱要編定設計，因此以該課程綱要之能力指標為範圍，按照「概念」與「解題技能」兩類，與本研究形成之概念架構圖進行對照，並以國小三到五年級所學習之分數概念及分數加減為範圍，進行診斷工具的建立。根據該題目內容及意涵，以概念架構圖為基礎，賦予各學習捷適當之數值描述。

（一）建立試題概念分配表：試題概念分配表是用來表示出概念在試題中分配的狀況，亦即試題中包含了哪些主題概念以及這些概念是位於試題中的哪個部

分（如表 4）：

表 4 本研究診斷工具之試題概念分配表

概念編號 / 題目編號	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
Q_A_1	0	0	0	1	0	0	0	0	0
Q_A_2a	0	0	0	0	1	0	1	0	0
Q_A_2b	0	0	0	1	1	0	1	1	0
Q_A_3a	0	0	0	0	0	0	0	1	0
Q_A_3b	0	0	0	0	1	0	0	1	0
Q_A_4a	0	0	0	0	0	0	1	0	1
Q_A_4b	0	0	0	0	1	0	1	0	1
Q_A_5a	0	0	0	0	0	1	0	1	0
Q_A_5b	0	0	0	0	1	1	0	1	0
Q_A_6	0	0	0	0	1	1	1	0	1
Q_A_7	0	0	0	0	0	1	0	0	0
Q_A_8	0	0	0	0	0	1	0	0	0
Q_B_1	0	0	0	0	1	1	0	1	0
Q_B_2	1	1	1	1	0	0	0	0	0
Q_B_3	1	1	1	1	0	1	0	1	0
Q_B_4	0	0	0	0	0	1	0	0	0
Q_B_5	0	0	0	0	0	1	1	0	1
Q_B_6	0	0	0	1	0	1	0	0	0
Q_B_7	0	0	0	0	1	1	1	1	1
Q_B_8	1	1	1	1	0	0	0	0	0
Q_B_9	0	0	1	1	1	0	0	0	0
Q_B_10	1	0	1	1	0	0	0	0	0
Q_B_11	1	0	0	1	0	1	1	0	0
Q_B_12	1	1	1	1	0	0	0	0	0
Q_B_13	1	0	0	0	0	0	0	0	0
Q_B_14	1	1	1	1	0	0	0	0	0
Q_B_15	1	0	0	1	0	1	1	0	0
Q_B_16	1	1	1	1	0	0	0	0	0
Q_B_17	0	0	0	0	1	1	1	1	1
總計	10	6	8	13	10	14	10	9	6

※編碼 Q_S_No 中，S 代表類別（A 代表「運算技巧」的錯誤；B 代表「學習概念」錯誤），No 則為在該類別之錯誤模式編號。

（二）建立本研究診斷工具概念分布矩陣表（如表 5）

表 5 本研究診斷工具之概念分布矩陣表

概念編號 / 題目編號	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
Q_A_1	0	0	0	0.077	0	0	0	0	0
Q_A_2a	0	0	0	0	0.1	0	0.1	0	0
Q_A_2b	0	0	0	0.077	0.1	0	0.1	0.11	0
Q_A_3a	0	0	0	0	0	0	0	0.11	0
Q_A_3b	0	0	0	0	0.1	0	0	0.11	0
Q_A_4a	0	0	0	0	0	0	0.1	0	0.167
Q_A_4b	0	0	0	0	0.1	0	0.1	0	0.167
Q_A_5a	0	0	0	0	0	0.071	0	0.11	0
Q_A_5b	0	0	0	0	0.1	0.071	0	0.11	0
Q_A_6	0	0	0	0	0.1	0.071	0.1	0	0.167

Q_A_7	0	0	0	0	0	0.071	0	0	0
Q_A_8	0	0	0	0	0	0.071	0	0	0
Q_B_1	0	0	0	0	0.1	0.071	0	0.11	0
Q_B_2	0.1	0.167	0.125	0.077	0	0	0	0	0
Q_B_3	0.1	0.167	0.125	0.077	0	0.071	0	0.11	0
Q_B_4	0	0	0	0	0	0.071	0	0	0
Q_B_5	0	0	0	0	0	0.071	0.1	0	0.167
Q_B_6	0	0	0	0.077	0	0.071	0	0	0
Q_B_7	0	0	0	0	0.1	0.071	0.1	0.11	0.167
Q_B_8	0.1	0.167	0.125	0.077	0	0	0	0	0
Q_B_9	0	0	0.125	0.077	0.1	0	0	0	0
Q_B_10	0.1	0	0.125	0.077	0	0	0	0	0
Q_B_11	0.1	0	0	0.077	0	0.071	0.1	0	0
Q_B_12	0.1	0.167	0.125	0.077	0	0	0	0	0
Q_B_13	0.1	0	0	0	0	0	0	0	0
Q_B_14	0.1	0.167	0.125	0.077	0	0	0	0	0
Q_B_15	0.1	0	0	1	0	0.071	0.1	0	0
Q_B_16	0.1	0.167	0.125	0.077	0	0	0	0	0
Q_B_17	0	0	0	0	0.1	0.071	0.1	0.11	0.167
總計	10	6	8	13	10	14	10	9	6

三、建立常見錯誤類型比對工具

針對國內、外分數概念相關研究中，分數加減「概念學習」上與「運算技巧」常見錯誤類型綜合整理後總結並建立屬性分析表如表 3：

表 3 分數加減常見錯誤類型之屬性分析表

編號	錯誤類型	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
Err_A_1	整數與分數換算	0	0	0	0	1	1	0	0	0
Err_ A _2	整數借位的問題	0	0	0	0	1	1	1	0	0
Err_ A _3	同分母分數的加法運算錯誤	0	0	0	0	0	0	0	1	0
Err_ A _4	異分母分數的加法運算錯誤	0	0	0	0	0	1	0	1	0
Err_ A _5	同分母分數的減法運算錯誤	0	0	0	0	0	0	0	0	1
Err_ A _6	異分母分數的減法運算錯誤	0	0	0	0	0	1	0	0	1
Err_ A _7	帶分數的加法運算錯誤	0	0	0	0	1	1	1	1	0
Err_ A _8	帶分數的減法運算錯誤	0	0	0	0	1	1	1	0	1
Err_ B _1	等分的觀念並不完整	0	1	0	0	0	0	0	0	0
Err_ B _2	單位量的辨認不清	1	0	0	0	0	0	0	0	0
Err_ B _3	單位分數概念不清	0	0	1	0	0	0	0	0	0
Err_ B _4	受整數基模的影響	1	1	1	0	0	0	0	0	0
Err_ B _5	部份－全部的區分轉	1	1	1	1	0	0	0	0	0

	換									
Err_ B _6	等值分數概念	0	0	0	0	0	1	0	0	0

※編碼 Err_S_No 中，S 代表類別（A 代表「運算技巧」的錯誤；B 代表「學習概念」錯誤），No 則為在該類別之錯誤模式編號。

四、進行診斷性演算

（一）建立學生答題狀況表（如表 6）：

表 6 學生答題狀況表

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
Q_A_1					1			
Q_A_2a		1		1			1	
Q_A_2b	1			1		1		
Q_A_3a		1						
Q_A_3b		1						
Q_A_4a								
Q_A_4b								
Q_A_5a								
Q_A_5b			1					
Q_A_6						1		
Q_A_7	1							
Q_A_8						1		
Q_B_1		1			1	1	1	
Q_B_2								
Q_B_3				1		1	1	
Q_B_4								
Q_B_5				1	1	1		
Q_B_6				1				
Q_B_7				1				1
Q_B_8						1		
Q_B_9	1			1		1		
Q_B_10	1							
Q_B_11		1				1		
Q_B_12			1	1	1		1	
Q_B_13		1		1	1	1		1
Q_B_14	1			1				
Q_B_15		1				1		
Q_B_16	1			1	1		1	
Q_B_17		1						

（二）建立學生答題錯誤概念表（如表 7-a、7-b、7-c）

表 7-a 本研究實作學生答題錯誤概念統計（整體答題狀況）

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
S_1	0.3	0.334	0.5	0.385	0.2	0.071	0.1	0.11	0
S_2	0.3	0	0	1.077	0.4	0.284	0.4	0.44	0.167

S_3	0.1	0.167	0.125	0.077	0.1	0.071	0	0.11	0
S_4	0.5	0.668	0.625	0.539	0.4	0.284	0.4	0.33	0.334
S_5	0.3	0.334	0.25	0.231	0.1	0.142	0.1	0.11	0.167
S_6	0.5	0.334	0.375	1.385	0.4	0.497	0.5	0.33	0.334
S_7	0.3	0.501	0.375	0.231	0.2	0.142	0.1	0.22	0
S_8	0.1	0	0	0	0.1	0.071	0.1	0.11	0.167

表 7-b 本研究實作學生答題錯誤概念統計（解題技能）

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
S_1	0	0	0	0.077	0.1	0.071	0.1	0.11	0
S_2	0	0	0	0	0.2	0	0.1	0.22	0
S_3	0	0	0	0	0.1	0.071	0	0.11	0
S_4	0	0	0	0.077	0.2	0	0.2	0.11	0
S_5	0	0	0	0.077	0	0	0	0	0
S_6	0	0	0	0.077	0.2	0.142	0.2	0.11	0.167
S_7	0	0	0	0	0.1	0	0.1	0	0
S_8	0	0	0	0	0	0	0	0	0

表 7-c 本研究實作學生答題錯誤概念統計（概念學習）

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
S_1	0.3	0.334	0.5	0.308	0.1	0	0	0	0
S_2	0.3	0	0	1.077	0.2	0.284	0.3	0.22	0.167
S_3	0.1	0.167	0.125	0.077	0	0	0	0	0
S_4	0.5	0.668	0.625	0.462	0.2	0.284	0.2	0.22	0.334
S_5	0.3	0.334	0.25	0.154	0.1	0.142	0.1	0.11	0.167
S_6	0.5	0.334	0.375	1.308	0.2	0.355	0.3	0.22	0.167
S_7	0.3	0.501	0.375	0.231	0.1	0.142	0	0.22	0
S_8	0.1	0	0	0	0.1	0.071	0.1	0.11	0.167

（三）錯誤類型的比對

將學習者的「答題錯誤概念表」(表 7-a~c）與「常見錯誤類型屬性分析表」（表 3)，採灰關聯分析之運算（式 7)，進行錯誤類型特徵值的比對，進一步掌握學生的學習狀況，深入其學習歷程的問題，比對結果及分析將於下一段落進行說明。

伍、方法評估

一、本方法之診斷成效

在學習者 S_2 的診斷結果指出，「單位量的辨認不清」、「異分母分數的減法運

算錯誤」、「整數與分數換算」等三錯誤模式，為關聯度較大之錯誤模式。經深入分析學習者之作答，將診斷結果與作答錯誤原因相互對照後，說明如表 8：

表 8 診斷結果與作答錯誤原因對照表

灰關聯排序	診斷結果	作答錯誤題號	作答錯誤分析
1	單位量的辨認不清	Q_B_1	圖形等分成四份後，將四份加上整個圖形（4＋1＝5），當作單位量
		Q_B_11	圈出四個彈珠當作單位量，再從其中圈出一個當分子（學習者自訴其算式）
		Q_B_12	認定整個為單位量，無法將左邊圖形切割獨立視為單位量
		Q_B_13	學生答題說明：因為$\frac{1}{2}=\frac{1}{2}$（缺乏分屬不同單位量之概念）
		Q_B_15	僅以數字比大小，未加入單位量觀念
2	異分母分數的減法運算錯誤	Q_B_17	未完成兩階段問項
3	等值分數概念		
4	整數與分數換算	Q_A_2a	$1\frac{13}{20}$ 中，換算為分數時，忽略分母為 20，誤將整數 1 轉換為分母量 10，$1\frac{13}{20}$ 化為$\frac{23}{20}$
5	同分母分數的減法運算錯誤		
6	帶分數的減法運算錯誤		
7	整數借位的問題		
8	等分的觀念並不完整		
9	單位分數概念不清		
10	同分母分數的加法運算錯誤	Q_A_3a	將「同分母分數的加法」誤算為「同分母分數的減法」
11	帶分數的加法運算錯誤		
12	部份－全部的區分轉換		
13	受整數基模的影響		
14	異分母分數的加法運算錯誤		

說明：試卷中有五題（Q_B_1、Q_B_11、Q_B_12、Q_B_13、Q_B_15），答題錯誤原因與該學習者「單位量的辨認不清」的錯誤類型符合。試題（Q_B_7），答題錯誤原因與該學習者「異分母分數的減法運算錯誤」的錯誤類型符合。試題（Q_A_2a）答題錯誤原因與該學習者「整數與分數換算」的錯誤類型符合。診斷錯誤類型排序為 3 的「等值分數概念」，雖未在答題反應中得到直接對應，但因該錯誤類型在異分母比較大小或加減時使用之「通分」技巧有關，也與整數、分數換算技巧有關，所以推論該錯誤類型為學習者 S2 可能之錯誤類型，需列為教

學者後續觀察、診斷之目標。

二、「解題技能」與「概念學習」兩階段診斷之結果

學生學習過程中，常因為計算技巧不斷的操練，相關題型反覆練習而建立答題模式，答題正確並不一定代表學習者擁有清楚概念。將學生在「解題技能」與「概念學習」兩類別之作答情況進行分析，讓學生的學習訊息得到更完整之呈現，俾利教學者進行更完備之判斷。

整理診斷數據（如表 9)，將各學習者排序前五項錯誤類型，依「解題技能」與「學習概念」分類列出，再將這兩類中的項目進行比對。如果在兩類中均出現則列為「一致項目」，未在另一類別出現的項目則列為「不一致項目」，經由這樣的比對，了解學生在「解題技能」與「學習概念」學習上，是否有一致性，藉此增加了解學生學習狀況的廣度。

表 9 「解題技能」與「概念學習」診斷對照表

	兩類診斷結果一致項目	一致項數	兩類診斷結果不一致項目	不一致項數
S1	單位量的辨認不清	1	等值分數概念、異分母分數的減法運算錯誤、同分母分數的減法運算錯誤、等分的觀念並不完整、單位分數概念不清、等分的觀念並不完整、受整數基模的影響、部份－全部的區分轉換	8
S2	同分母分數的減法運算錯誤、單位量的辨認不清、異分母分數的減法運算錯誤、等值分數概念	4	等分的觀念並不完整、整數與分數換算	2
S3	等值分數概念、單位量的辨認不清、同分母分數的減法運算錯誤、等分的觀念並不完整	4	異分母分數的減法運算錯誤、單位分數概念不清	2
S4	單位分數概念不清　、等分的觀念並不完整、單位量的辨認不清	3	異分母分數的減法運算錯誤、受整數基模的影響、同分母分數的減法運算錯誤、部份－全部的區分轉換	4
S5	單位分數概念不清、同分母分數的減法運算錯誤、等分的觀念並不完整、等值分數概念	4	異分母分數的減法運算錯誤、單位量的辨認不清	2
S6	等值分數概念、異分母分數的減法運算錯誤、單位量的辨認不清	3	同分母分數的減法運算錯誤、等分的觀念並不完整、部份－全部的區分轉換、單位分數概念不清	4
S7	單位量的辨認不清、等分的觀念並不完整	2	同分母分數的減法運算錯誤、異分母分數的減法運算錯誤、等值分數概念、單位分數概念不清、同分母分數的加法運算錯誤、受整數基模的影響	6
S8	異分母分數的減法運算錯誤、同分母分數的減法運算錯誤、等分的觀念並不完整、等值分數概念	4	單位分數概念不清、單位量的辨認不清	2

說明：

學習者 S_2、S_3、S_5與 S_8在兩類別的診斷對照中，呈現錯誤類型歧異度較低，推測四位學習者在該學習主題的「解題技能」與「概念學習」兩方面的錯誤類型較一致。學習者 S_1與 S_7在兩類別的診斷對照中，呈現錯誤類型歧異度較高，推測兩位學習者在該學習主題的「解題技能」與「概念學習」兩方面的錯誤類型並不一致，在後續補救引導的範圍及路徑，教學者都須將其納入考量。

陸、結論及未來工作

一般的測驗著重於以總分來評定受試者的學習狀況，這樣的評量方式很難對受試者的整體學習狀況達到通盤瞭解。如何藉由知識結構的基礎，配合受試者的量化施測資料結果或質性晤談紀錄，呈現出受試者真實、客觀的知識學習狀況，是許多研究者所關心的議題。本研究綜合診斷與分析結果，提出了一個基於知識概念結構，結合灰系統中探求關聯度的方法，提供個別學習診斷報告書，讓學生能夠了解自己的學習狀況，更有利教師根據學生的診斷情形，對於未精熟或漏失的概念進行觀念的建立，更可針對迷思的概念進行轉構教學。

目前本研究成果如下：

一、將學生答題狀況，根據解題技能、概念學習進行分類比對發現學生的解題技能與對該學習主題的概念了解與否，並非一定相關。

二、試題的概念分布和錯誤類型的推論有很大的關係，依循學習路徑結構布題，將能收取更多訊息以供比對。

三、本方法的診斷可客觀呈現出學生答題成果，作為教學模式中「回饋功能」之重要參考。

四、透過學習診斷報告，可以在學習路徑之基礎上，有根據地進行學生錯誤概念之補救，達到事半功倍之效能。

柒、參考文獻

何榮桂（1997）。網路環境題庫與測驗之整合系統。*八十六年度電腦輔助學習及遠距教學專題研究計畫成果討論會摘要*， 44-162。

翁慶昌，陳嘉欉，賴宏仁（2001）。*灰色系統基本方法及其應*用。台北：高立。

張紹勳（1998）。電腦網路遠距教學實施碧況之問題研究。載於中華民國電腦輔助教學學會主辦，*第七屆國際電腦輔助教學研討會論文集*（頁 573-580）。高雄：國立高雄師範大學。

許慶昇、杜淑芬、黃國禎（1998）。概念繼承關係在網路智慧型學習診斷系統之應用。載於中華民國電腦輔助教學學會主辦，*第七屆國際電腦輔助教學研討會論文集*（頁 602-609）。高雄：國立高雄師範大學。

甯自強（1998）。涂景翰的數概念。*科學教育學刊*，6(3), 255-269。

詹婉華、呂玉琴(2004)。國小高年級學童分數概念量表之設計研究。*科學教育學刊*，12(2)， 241-263。

温坤禮、黃宜豐、張偉哲、張廷政、游美利、賴家瑞（2003）。*灰關聯模型方法與應用*。台北：高立。

Clark, C. M., & Peterson, P. L. （1986）. Teachers' thought processes, In M. C. Wittrock (Ed.), *Handbook of research on teaching.* New York: McMillan.

Glaser, R. (1962). Psychology and instructional techonology. In R. Glaser （Ed.）. *Training research and education.* Pittsburgh: University of Pittsburgh.

Ishikawa, A., M. Amagasa, T. Shiga, G. Tomizawa, R. Tatsuta and H. Mieno, (1993). The Max-Min Delphi method and fuzzy Delphi method via fuzzy integration. *Fuzzy Sets and Systems*, 55, 241-253.

Kathlen, T. T. (1986). *Error reduction strategies for whole number operations in grade four.* Doctoral Dissertation, University of Brigham Young.

Nichols, P. D. (1994). A framework for developing cognitively diagnostic assessments. *Review of Educational Research,* 64(4), 575-603.

Piaget, J., & Inhelder, B. （1969）. *The psychology of the child.* London： Routledge & Kegan Paul.

Piaget, J., Inhelder, B., & Szeminska, A. （1960）. *The Child's Concept of Geometry.* New York： Basic Book.

Snow, R. E., & Lohman, D. F. （1989）. Implications of cognitive psychology for educational measurement. In R. L. Linn (Ed.), *Educational measurement* (3rd ed). New York: Macmillan.

Stefanich, G.P. & Rokusek,T.(1992). Analysis of computational error in the use of division algorithms by fourth-grade students. *School Science and Mathematics,* 92(4), 201-205.

Vance, J. H. (1992). Understanding equivalence: A number by any other name. *School Science and Mathematics*, 92, 263-266.

附件三

角色扮演遊戲結合動態評量教學平台架構之研究

董建亨、王昌斌、藍鈺凱
南華大學資訊管理研究所
jhdung@gmail.com

摘要

在教學平台的研究上，目前已有學者研究，將角色扮演遊戲結合動態評量。根據這些的研究發現，角色扮演遊戲結合動態評量，能增進學生的學習興趣，與提升學生學習的動機、增加學生學習的經驗、提升學生整體的知識吸收程度，達到建立長久的學習效果。

基於上述因素，本研究希望藉由角色扮演遊戲結合動態評量的研究，研究設計出最佳化的平台功能架構，且希望藉由學校的課程，將其融入角色扮演遊戲的設計中。在最佳化的平台功能架構設計裡，本研究以達到主要之目的為：一、利用遊戲教學方法提高學習動機。二、利用動態評量方式持續診斷學習能力。三、教師能對學生學習個案分析。四、提供適性化學習內容。五、促進學生的學習成效與興趣。

關鍵詞：網路教學平台、動態評量、遊戲教學法、角色扮演遊戲。

1. 前言

隨著資訊科技與網際網路的進步與普及，電腦應用在教學、評量的學習上，已逐漸形成一種風氣，電腦處理資料的快速、便捷更為大家所喜愛，也是一種節省資源與成本的一種方式。在 E-Learning 和數位學習上，不論是各家企業、教育單位的訓練、教師的出題和學生的學習，已成為在網路應用上的主要工具之一。

處在於 21 世紀知識經濟社會的現今，學校教育的重要功能之一，為能有效地將知識傳遞給學生學習。教育可以協助每一個人的終身學習、學習天賦、才能和潛力地充分發揮，以達成人生學習目標和生活的意義為目的。身為學校教師是有義務、責任，提供所有學生同等品質的學習機會，以達到「因材施教」的目的，讓所有學生都可以學習到知識。不分年齡、社會階層，教育主要以「因材施教」為目的，每個人都有機會接受同等的教育，並使其天賦、才能發揮到最高程度。

2. 主要內容

學校教育的主要功能在培育學生能夠學以致用，能將知識轉化為因應各種生活挑戰和工作所需的關鍵能力(洪明洲，1999)。隨著科技的日新月異，學校利用各種不同的方式來達到教學的目的，從最早的傳統教學方式，老師與學生採面對面的形式進行教學活動，到後來透過一些媒體的傳播，開啟了遠距教學的時代。近年來，為了應用資訊科技之網際網路來進行學習，網路教學的發展已如雨後春筍般地蓬勃發展(孫明照，2002)。

現今的學校教育環境、課程，大多是

為一般的學生設計，但對於弱勢族群、學習障礙和條件不足的學生，無法依學生學習的差異性而修改教材與活動，且很難落實依學生的程度，擬定教學目標和教學進度，此外，依據個別學生的學習狀況，教師也難以針對個別學生，做個別的教學與學習，所以基於上述，本研究擬定「角色扮演遊戲與動態評量教學平台架構」的研究，設計出讓教師能利用此教學平台，從學生的「測驗、評量、再介入測驗」中，知道各個學生已具備的課程知識涵養、欠缺的知識，讓教師能針對個別學生，做個別的教材設計、題庫設計，讓學生能從中學習與獲得課程的知識。另外，教師可依據此教學平台，來設計一套角色扮演遊戲的相關課程之劇情，供學生從遊戲中，能學習到課程所需具備的知識；在學生方面，學生亦可從教師安排的課程與測驗中，從中測驗自己對課程的了解程度，也可從學習歷程與教師設計的教材與知識庫中，學習更多的課程知識，且在教師設計的遊戲中，可從中學到課程的知識，亦能知道自己所具備的課程知識之能力範圍。

此教學平台主要以達到讓教師能設計題庫、教材、動態評量、對學生學習個案分析、達到適性化教學為目的。對學生方面，此教學平台能讓學生從教師的設計題庫、教材編製中，學習到更多的課程知識，此外，學生亦可透過知識地圖的方式，學習到更多的課程知識。

此教學平台架構設計中，主要以「國小」的課程為主，來建構此教學平台，其主要以達到讓教師能設計題庫、管理動態評量、對學生學習個案分析、輔助弱勢學生學習。對學生方面，此教學平台能讓學生從教師的設計題庫、教材編製中，學習到更多的課程知識，此外，學生亦可透過知識地圖的方式，學習到更多的課程知識。基於上述之前言與主要內容，本研究的教學平台設計，主要目的為：一、利用遊戲教學方法提高學習動機。二、利用動態評量方式持續診斷學習能力。三、教師能對學生學習個案分析。四、提供適性化學習內容。五、促進學生的學習成效與興趣。

3. 文獻探討

3.1 網路教學平台

洪明洲(1999)指出，網路教學平台是一種網路應用程式，由兩種結構組成：儲存結構和展現結構。所謂的「儲存結構」是指伺服器上儲存之網路教室教材資料的檔案與目錄之結構，包括HTML網頁、圖片、音樂、電影...等資料；而「展現結構」是網站網頁的架構，到訪者瀏覽網頁時，依網頁上的按鈕或指示去瀏覽網站內容，其結構路徑會影響學生尋找資料的難易度，若展現結構太複雜或尚未建置完全，不管所存放的檔案資料內容如何豐富精彩，學生的學習效果和興致都會受影響。

在教學平台功能部分研究裡，張偉遠(2003)的研究中，分析東西方部份國家網上教學平台，將網上教學平台的功能歸為14項，分別有：電子郵件、電子公告版、在線聊天、電子白版、評分功能、學習跟蹤、學生網頁、技術支援、文件共用、檢索工具、問卷調查、網上教材、補充教材、用戶定制。

根據前文獻 3.1 網路教學平台之探討中，本研究的教學平台架構設計裡，主要是以全球資訊網超連結與多媒體呈現的特性，讓教師與學生能透過簡單的網路使用介面，其主要特性包含有：教師與學生的

互動性、多元教材的呈現、老師與學生的教材資源共享...等；在本教學平台裡，其功能主要有：討論區、學生評量功能、學生學習歷程追蹤、教材資源分享、教師補充教材...等。

3.2 動態評量

動態評量（Dynamic Assessment）是近二十年新興的評量方式，一詞是由Feuerstein(1979)首先使用，主要是相對於傳統評量的靜態測量性質所提出的。其之所以稱為動態的含義有二:(1)著重學習歷程或認知改變的評量;(2)在評量中進行教學，評量者與被評量者的關係是互動的(Haywood, Brown & Ingenfeld, 1990)。Campione and Brown(1987)則從「評量內容」和「評量本身」的角度來闡述「動態」之意，其認為動態評量所著重的評量內容，不在於評量過去既有的知識、技巧或經驗等靜態結果，而在於評量成長、改變的動態歷程以及學習預備度，評量本身即是一種動態的過程。茲將有關動態評量的定義，整理如表1。

表1：動態評量的定義

研究者或研究單位	**動態評量的定義**
莊麗娟（2001）	有別於傳統心理計量施測者保持中立，不提供額外協助的評量立場，強調評量過程中，配合受試者在解題上的實際需求，不斷的給予必要的協助，以引導受試者成功的解題。
邱上真（1996）	動態評量又稱協助式評量，即在測驗進行中，允許給學生提供暗示、線索及協助，以便獲得學生「最大可能操作水準」的資訊，稱之為動態評量。
李坤崇（2002）	動態評量是一種結合教學與診斷的評量模式。
李坤崇（2002）；楊景淵（2002）	動態評量乃是利用「前測—中介—後測」的循環程序，結合教學與評量，在評量的過程中，施測者透過各種符號工具與受試者充分的互動，診斷其認知缺陷，藉以獲得研擬教學處方的訊息；探索其學習潛能，對個體未來表現進行合理的評估；適時給予受試者適當的提示、協助與鼓勵，期望產生「鷹架」的作用，以達成個體的「最佳」表現。

資料來源：本研究整理

林秀娟（1993）的研究中，指出動態評量的主要目的是在：評估學習過程、預測學習潛能、診斷認知缺陷、提供教師瞭解學生的問題癥結及中介訊息以進行補救教學，使學生的潛能發展到最大。所以根據前述，吳國銘、洪碧霞、邱上真（1995）更指出，動態評量是將受試者視為開放的個體，能不斷自外界得到回饋而改變自己的認知結構，並藉由互動的測驗程序，達成瞭解受試者學習歷程及發展脈絡的目的。

綜合上述的文獻資料，本研究結合多位學者的觀點（Feuerstein, 1979；Lidz, 1987；Ferretiti & Butterfield, 1992；Haywood & Wingenfield, 1992；Tzuriel, 1992；Swanson, 1996），動態評量具有六個主要的特性：(1)前測—中介—後測的評量過程，結合評量與教學、(2)兼重鑑定、診斷與處方、(3)探究學習者的認知發展，重視評量過程甚於結果、(4)由評量中發現個體認知改變所需介入的程度和方式、(5)充分的互動、適時的協助、(6)著重於個別學生學習歷程的確認與評量，而非同儕間的能力比較。

歸納上述3.2動態評量的探討中，我們可以知道，動態評量受測者是一個學習的開放個體，透過施測之回饋，而使自己修正認知結構，從互動中得以最大之發展，而電腦動態評量有益於受測者之學習和能力之提昇，透過動態評量來進行補救教學，在教學之實務中可行性很高，也可以避免傳統人工施測不當介入之危機，影響受試者學習和潛能發展之表現。

因此，基於上述文獻探討，本研究的教學平台架構裡，主要設計出讓教師能利用此教學平台，從學生的「測驗、評量、再介入測驗」中，知道各個學生已具備的課程知識涵養、欠缺的知識，讓教師能針對個別學生，做個別的教材設計、題庫設計，讓學生能從中學習與獲得課程的知識。

3.3 遊戲教學法

遊戲教學法最早是由Aufschnaiter, Schwedes, and Helanko在1984年所提出的，主張以開發有趣的單元活動教材來改善教學與學習情境，他們認為影響學生認知推理過程中最重要的因素乃是：學生不斷的透過發展成長中的實際行動與感覺，把事物、行動和實體等各方面建立成一個客觀化的系統，進而形成概念結構，並增進其解決問題的能力。「遊戲教學法」是可以運用於不同領域的教學之中，教學者需將遊戲經過設計、從中誘導學生產生適當的學習能力及反應，讓學生從遊戲中得到有效的學習，達到教育目標。

遊戲是一種參與者依循特定規則，達成具有挑戰性目標的活動，遊戲之所以引人入勝並具有趣味性，是因為它本身存在著和現實世界的差異性，參與者可以暫時放下真實世界中邏輯性的規則，而置身在遊戲本身創造出來的世界中挑戰不同的情境、釋放情緒或是扮演不同的角色。根據上述學者的研究中，其遊戲的特色分別為：(1)引起參與者的動機、(2)善於營造生動化的環境、(3)賦予環境的掌控性、(4)容易產生遊戲的快感。

根據上述3.3遊戲教學法之探討，遊戲的活動不但具有趣味性，同時能夠建立多元化的教學環境。另外，遊戲教學具有教育以及娛樂的雙重功能，參與者在遊戲中可以發展個人的認知歷程，以及提高學習的動機，對於接受教育的學生亦有提高學習成效的助益。

3.4 角色扮演遊戲(RPGs)

角色扮演遊戲(Role Play Games，RPGs)主要是以角色為主的遊戲，玩家可以從遊戲中，選擇自己想要扮演的角色，來進行該項遊戲，以從中玩到不同角色所扮演的遊戲之樂趣，並使玩家可以更了解到不同角色所扮演的特色。角色扮演遊戲起初是以單機版遊戲為主，由於網路的無遠弗屆的發展，後來由單機版遊戲，轉向為網路線上遊戲的角色扮演為主。

線上遊戲世界裡的玩家，會因為自己選擇角色的種族、職業等屬性的不同，而在玩法與故事發展上有不同的安排與變化，讓玩家譜出屬於自己特有的冒險故事，此外，也提供了一個情境，讓人們脫離現實限制而有海闊天空地幻想的機會，使個人在其中宣洩生活中的不滿、衝突及挫折，紓解工作或課業壓力，達到個人在生活中情緒的調解，在扮演某個角色裡，可以讓線上遊戲的玩家體驗到更高的遊戲經驗(林培淵，2007)。

從上述3.4角色扮演遊戲的探討中，我們可以知道，線上角色扮演遊戲可以增進玩家對遊戲整個的劇情、事件中，產生興趣、可讓玩家與玩家產生良好的互動與角色之間的學習，因此，基於上述，在本研究裡，主要以「角色扮演遊戲」，結合動態評量教學平台架構，讓教師可依據此教學平台，來設計一套角色扮演遊戲的相關課程之劇情，供學生從遊戲中，能學習到課程所需具備的知識。在遊戲裡，學生可以從教師設計的遊戲劇情中，學到課程的知識，且利用遊戲教學方法，提高學生對課程知識涵養的學習動機、促進其學習成效與興趣。

4. 研究設計與實施

從上述 3.文獻探討中，我們可以知道，在教學平台的功能設計裡，其包含有：HTML 網頁、電子郵件、電子公告版、在線聊天、電子白版、評分功能、學習跟蹤、技術支援、文件共用、檢索工具、補充教材...等。在動態評量的研究裡，我們可以知道，動態評量採「前測—中介—後測」的評量過程，它是著重於個別學生學習歷程的確認與評量，而非同儕間的能力比較，且可以探究學習者的認知發展，審視評量結果...等。在遊戲的研究裡，藉由遊戲教學法，可以開發有趣的單元活動教材，來改善教學與學習情境，且可以運用於不同領域的教學之中，教學者需將遊戲經過設計、從中誘導學生產生適當的學習能力及反應，讓學生從遊戲中得到有效的學習，達到教育目標。在線上角色扮演遊戲中，玩家可以依據角色的選擇、屬性的創建，來進行該項遊戲，玩家不僅可以學習到不同角色的內容與特色，亦可從遊戲中學到該角色扮演的樂趣。

根據教學平台、動態評量、遊戲教學法、角色扮演遊戲的研究裡，本研究擬定「角色扮演遊戲與動態評量教學平台架構」，將教學與角色扮演遊戲相互整合出一個平台架構，其主要設計是希望藉由學校的課程，將其融入角色扮演遊戲的設計中，且在教學平台的設計裡，教師可依據其平台的功能，做教材編製、資源分享、對學生的學習評量、課程的遊戲設計...等；學生亦可從此教學平台裡，學習教師所編製的教材、資源分享、個人學習歷程記錄、線上玩教師所編製的遊戲...等。本研究的教學平台功能，主要先以教師、學生的使用性考量為設計，再根據現有的文獻資料，來做整個平台架構的設計。此外，依據使用者的使用方便性為原則，此教學平台最主要的功能設計，分別依據有：教師平台功能、學生平台功能、使用者管理與系統管理功能等的設計，此部分在「5.平台功能說明」裡有詳細之說明。該教學平台也依據教師使用的方便性，分別設計出的模組有：角色扮演遊戲之設計模組、動態評量之設計模組，以供教師從這些模組中，製作遊戲、教材、題庫的編製，其模組功能設計，說明如下：

一、角色扮演遊戲之設計模組

(1)角色設計：包含遊戲劇本、遊戲戰鬥。遊戲劇本可稱為「故事劇情」，通常是一個角色扮演遊戲是否吸引人,或是賣座的關鍵點,可說是一個遊戲的設計藍圖。遊戲戰鬥是所扮演的虛擬角色與遊戲中,隨機出現的角色採用回合制的戰鬥,取得勝利後,可增加虛擬角色之成長與遊戲中的虛擬金幣。

(2)遊戲物件：包含裝備、地圖...等。裝備主要是讓玩家有新鮮感,在解任務打怪中，裝備的等級越高,所做的防禦力也會提升。地圖即為遊戲探索，在繪製好之地圖中進行探索，地圖可以是迷宮、樹林、村莊、高山、湖泊等，探索活動包括尋寶、與非扮演的主角對談、前往目的地...等。

(3)遊戲任務：藉由探索模式、戰鬥模式或另一事件模式之進行,來滿足遊戲任務中之條件，如：猜謎、完成所交待之任務、搜集情報等，待條件皆滿足後，即可完成此事件,並可往下一階段之故事劇情進行。

二、動態評量之設計模組

(1)出題評量：依據教師所設定的學科別、概念、難易度等資訊，自動從題庫進行選題，並於遊戲進行中進行評量，根據學生作答情形,立即告知作答正確性及正確解答。

(2)概念推論：依據教師所設計的概念選出題目後進行作答,從中判斷學習者對於此概念的能力水準,再推論出學生個人的概念路徑,並採用循序漸進的出題方式進行評量,同時對此概念的作答情況推論出其迷思的概念。

(3)教材庫：可利用概念路徑中的概念依序，找出學習的教材，以供學生自行學習,亦可主動提供其概念迷思的教材進行補救教學。

(4)學習歷程記錄：記錄學生在各概念之答題狀況，並加以統計。

5. 平台功能說明

5.1 教師介面

一、教學設計

(1)教材設定：供教師做教材編製與管理，可讓學生從教學教材裡,學習到課程知識。

(2)學習歷程追蹤：供教師了解各個學生在教學裡的學習狀況,與知道學生的知識概念落在哪個領域,和所欠缺的能力，方便教師在教材的設計中,能針對學生的學習狀況去做設計。

(3)課程評量設定：教師能針對學生的學習狀況和學生在線上的作答情形,針對課程的設計做編製。

(4)學習資源建置：讓教師能針對學生的學習歷程與學習狀況，做「案列學習」的資源建置,且教師能為學生的學習檔案做管理,供教師做「教材設計」的參考。

(5)教學策略設定：讓教師依據不同學生的學習狀況與能力，做「教學準備」、「成效評量」、「知識導引」。在「教學準備」裡，主要讓教師了解學生的學習狀況，供教師日後做教學教材、題庫的編製。「知識指引」，即為根據教師的經驗知識，在教學教材裡，幫助學生做學習的引導,讓學生能充分學習到更多的課程知識。

(6)課程安排：教師可依據學生的學習狀況，在教材設計裡，針對不同的學生，做不同的學習課程安排。

二、遊戲設計

在教師的「遊戲設計」裡，教師可以依據課程的內容，從「遊戲事件」、「RPG元件」和「遊戲參數」裡，設計一套讓學生能從遊戲的劇情、觸發事件中，學習到課程知識。

三、知識擷取

(1)概念知識擷取：讓教師可以做「概念查詢」，以了解各概念的知識內容，供教師為學生做教材的知識指引，且在知識指引中，讓教師在設計教材時，能編製「學習目標」、「學習權重」、「學習時間」，以利教師為學生做不同的教材設計。

(2)教材與題庫分析：讓教師了解教材設計的出題評量中，學科、概念對學生的學習難易度。在「概念推論」裡，能針對學生的學習狀況與歷程中，推論整個的「學習路徑」、「迷失」、「評量試題」、「評量結果」，供後續教材設計。在「教學提示」中，教師能根據學生的學習歷程和記錄，做教材設計的學習提示。

四、學習成效統計

(1)學習歷程與記錄：供教師了解學生在課程教學裡的學習情況，方便教師幫學生做各種的教材設計。

(2)概念知識：讓教師知道學生在自己設計的教材裡，所學習到的知識能力範圍，且了解學生已學會的知識與所欠缺的知識，供教師「教材設計」、「資源建置」、「教學策略」的參考。

(3)能力水準：讓教師知道學生在學習過程中的能力程度，與學習完後的能力程度，供教師做教材設計的考量。

五、討論區

在「討論區」裡，教師可從中與學生進行課程與問題的討論，亦可從中幫助學生解答問題的疑惑、輔助學生對課程學習的指引。

5.2 學生介面

一、線上測驗

依據教師設計的題卷裡，學生可先從「課程」選擇「題卷編號」，以供學生線上作答，帶學生作答完後，可立即產生分數、錯題的解答，供學生了解自己做錯的原因。

二、課程學習

根據教師對各學生的個別課程學習設計中，各個的學生，可從中線上觀看教師給的教材、學習建議和教材下載。

三、歷程與記錄

學生可從中知道過去的測驗、學習的狀況，以供學生學習的參考。

四、遊戲區

依據教師設計的遊戲劇情和觸發事件裡，學生可從遊戲的劇情、觸發事件中，學習到課程知識，及知道自己所具備課程知識的能力範圍。

五、討論區

供學生線上發問問題、討論，亦可從教師、同學的討論與發問中，學習到更多的知識。

5.3 使用者管理介面

此部分包含了：教師、學生、專家。在「教師」的登入裡，主要讓教師可以設計動態評量、教學設計、學習分析。在「學生」的登入裡，可以讓學生從中知道自己學習的歷程、線上學習、使用教學資源、可以進入教師設計的遊戲裡，學習課程知識。在「專家」的登入裡，主要為教師的「知識專家」，知識專家可供教師的知識指引、概念推論。

5.4 系統管理介面

此為「行政管理者」，主要管理使用者的權限、製作平台模組、建置平台的功能。

6. 平台系統設計架構

6.1 平台首頁架構

在系統首頁裡，本研究主要讓使用者能清楚知道此系統的功能、用意，讓使用者能輕易上手使用本教學平台系統，因此，我們設計出：學習目標、學習方法、使用教學，供使用者參考、了解。此外，在「會員中心」裡，包含了：系統管理員、教師、學生、加入會員(如圖 1)。

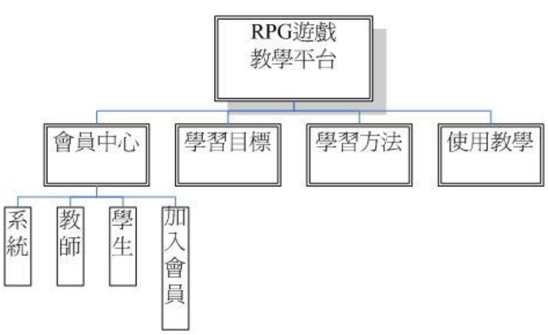


圖 1：平台首頁架構

6.2 教師平台架構

在「教師平台」裡，主要讓教師能針對個別的學生，做個別的教學和教材設計，供學生學習，且教師可從中知道學生的學習狀況，供教師能對學生的學習狀況，做教學的設計。在討論區裡，教師可與學生一同參予討論，且可輔助學生對問題的解答，此外，在遊戲區裡，教師可針對課程內容，設計一套與課程相關的遊戲劇情，供學生從遊戲中，能學習到課程的知識。此「教師平台」系統，分別有：教學設計、遊戲設計、檔案管理、學習歷程、成效統計、討論區(如圖 2)。

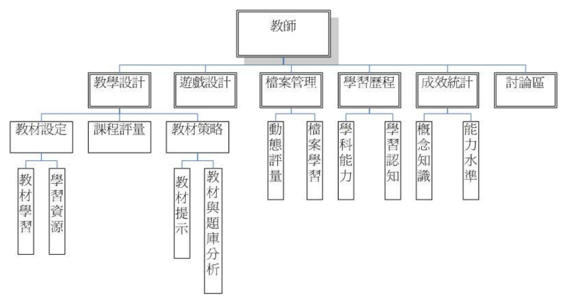


圖 2：教師平台架構

6.3 學生平台架構

在「學生平台」裡，學生可從教師設計的教材、教學裡，學習課程的知識。在學習歷程與記錄裡，學生可以從中知道自己的學習狀況。在遊戲區裡，學生可從遊戲裡的劇情、觸發事件，學習到課程的知識。此「學生平台」系統，分別有：個人資料、遊戲區、線上測驗、課程學習、歷程＆記錄、討論區(如圖 3)。

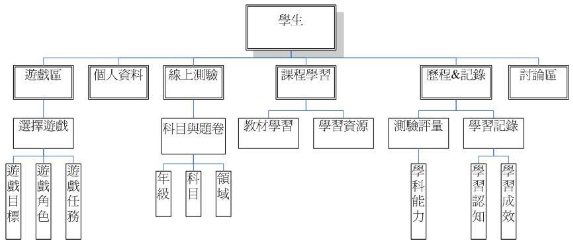


圖 3：學生平台架構

7. 結論

本研究的教學平台架構設計，主要以國小的課程為主，來建構此教學平台，其主要以達到讓教師能設計題庫、管理動態評量、對學生學習個案分析、輔助弱勢學生學習。對學生方面，此教學平台能讓學生從教師的設計題庫、教材編製中，學習到更多的課程知識，此外，學生亦可透過知識地圖的方式，學習到更多的課程知識。本研究具體的貢獻，如下所示：

(1)藉由角色扮演遊戲平台開發,以提高學生對課程的學習動機,並確保能符合輔助教學師生的需求。

(2)開發「評量-教學-再評量」之動態評量機制,並結合診斷機制,以達到適性化教學為目的。

(3)結合角色扮演遊戲與動態評量之設計,並建置教學平台,加強輔導學生,以拉近日漸懸殊的學生素質,使得學生都能成功學習。

(4)利用網路無遠弗屆的特性,建立並助長校際間補救教學的合作關係,以節省教育資源。

(5)協助補救教學知識擷取與知識概念建置,並可重複使用此知識概念應用各領域。

(6)藉由經驗傳承與教育訓練,使教師或有興趣的業界人士,能獨立開發角色扮演遊戲,或增加動態評量之題庫與數位教材,並永續經營成為政府、商業與其他教育機構發展補救教學的主要資源。

參考文獻

1. 吳國銘、洪碧霞、邱上真,「國小學童在動態評量中數學解題學習歷程與遷移效益之探討」,測驗年刊,42,61-84,1995。

2. 李坤崇,「多元化教學評量理念與推動策略」,教育研究月刊,98,24-36,2002。

3. 林秀娟,「動態評量結合試題反應理論在空間視覺學習潛能評量之研究」,國立台灣師範大學教育心理與輔導研究所碩士論文,1993。

4. 林培淵,「線上遊戲之玩家行為初探研究－以《魔獸世界》為例」,國立中正大學電訊傳播研究所碩士論文,2007。

5. 邱上真,「動態評量-教學評量的新嘗試」。載於國立高雄師範大學主編,中小學教學革新研討會論文集(頁33-49),高雄:國立高雄師範大學,1996。

6. 洪明洲,「大學教學理念與技術的演進-網路教學與學習效果之改善實例」,1999 提昇教學品質研習會報告論文,海洋大學,1999。

7. 孫明照,「符合 SCORM 標準之元件化網路教學平台之設計研究」,銘傳大學

資訊管理研究所碩士論文，2002。

8. 張偉遠,「亞洲地區開發大學網上教學的比較研究」,遠距教育國際學術研討會論文，台北：國立空中大學、國立台灣大學，2003。

9. 莊麗娟,「多媒體動態評量低獲益受試者之認知缺陷與協助策略分析」,特殊教育研究學刊，21，109-133，2001。

10. 楊景淵,「給學生多一些學習與表現的機會-動態評量在自然科學概念學習的應用」,翰林文教,28,7-13,2002。

11. Aufschnaiter, V.S., Prum, R., & Schwedes, H., "Play and Play Orientation in Physics Education," in Unterricht-P/C,32,258-263, 1984.

12. Campione, J. C., & Brown, A. L., "Linking dynamic assessment with school achievement," In C.S. Lidz (Ed.), Dynamic assessment: an interaction approach to evaluation learning potential (pp.173-195), New York: Guiford, 1987.

13. Ferretiti, R. P., & Butterfield, E. C., "Intelligence-related differences in the learning, maintenance, and transfer of problem-solving strategies, " Intelligence,16,207-224, 1992.

14. Feuerstein, R., "The dynamic assessment of retarded performers : The learning potential assessment device, theory, instruments, and techniques," Baltimore, Maryland: University Park Press, 1979.

15. Haywood, H.C., & Tzuriel, D., "Interactive assessment," New York：Spring-Verlag,Inc, 1992.

16. Khan, B.H., "Web-based instruction (WBI): What is it and why is it?" Web-based instruction. Englewood Cliffs, NJ:Educational Technology Publications.,5-18, 1997.

17. Lidz, C.S., "Dynamic assessment：An Interactional approach to evaluating learning potential," Historical Perspectives. In C.S.Lidz（Ed.,（pp.3-32）. New York：The Guilford Press, 1987.

18. Swanson, H. L., "Classification and dynamic assessment of children with learning disabilities," Focus on Exceptional children, 28,1-19, 1996.

19. Tzuriel, D., "The dynamic assessment approach：A reply to Frisby and Braden," The Journal of Special Education,26,302-324, 1992.

附件四

A Formal Virtual Enterprise Access Control Model

Tsung-Yi Chen, Yuh-Min Chen, and Chin-Bin Wang

***Abstract*—A virtual enterprise (VE) refers to a cooperative alliance of legally independent enterprises, institutions, or single persons that collaborate with each other by sharing business processes and resources across enterprises in order to raise enterprise competitiveness and reduce production costs. Successful VEs require complete information transparency and suitable resource sharing among coworkers across enterprises. Hence, this investigation proposes a formal flexible integration solution, named the formal VE access control (VEAC) model, based on the role-based AC model, to integrate and share distributed resources owned by VE members. The formal VEAC model comprises a fundamental VEAC model, a project AC policy (PACP) language model, and a model construction methodology. The fundamental VEAC model manages VE resources and the resources of participating enterprises, in which various project relationships are presented to facilitate different degrees of resource sharing across projects and enterprise boundaries, and cooperative modes among VE roles are presented to enable collaboration among coworkers in a VE. This PACP language model features object–subject–action–condition AC policies that jointly determine user access authorizations. In addition, the methodology supplies a systematic method to identify fundamental elements of the VEAC model and to establish assignments between elements and relations.**

***Index Terms*—Access control (AC), resource sharing, role-based access control (RBAC), virtual enterprise (VE).**

I. Introduction

VIRTUAL enterprise (VE) is regarded as one of the most promising business strategies to enhance the global competitiveness of enterprises [1]. VEs integrate the processes, activities, and resources from different enterprises through enterprise alliances to respond quickly to customer expectations. Frenkel *et al.* [2] defined a VE as a collaborative group of existing autonomous enterprises, which selectively share their expertise, skills, and resources to accomplish a common product or service. In practice, a VE is generally implemented with a distributed and collaborative business process, in which individuals from different enterprises cooperate on business-related activities or processes by remote coordination, communication, and control [1]. To attain VE goals and support each other's functionalities, enterprises in a VE must share and exchange information, knowledge, and resources. The features that determine the access level to the local information of every enterprise, when considering the competitive and cooperative relationships among enterprises, include the degree of trust between two enterprises, the function of the enterprises in the VE, and contractual agreements [2], [3]. Lu *et al.* [4] proposed a trust-based privacy preservation method for P2P data sharing.

A collaborative engineering environment allows multiple engineers to work simultaneously with individual assembly parts. Some manufacturing industries, e.g., the automotive sector, use VEs to maintain business relationships with their suppliers and corporate customers, enabling manufacturers to collaborate on the design, production, assembly, and marketing of new products. For instance, designing and developing a new car is a complex and lengthy process; during product R&D, engineering and design drawings can be shared over secure network among the contracting firm, testing facility, marketing firm, and downstream manufacturing and service companies [5]. Information concerning the design for a new product at various segments of the VE has to be visible to all members of the VE at any time. Consequently, the information must be managed properly, with appropriate access control (AC) models, strict policies, discipline, and daily monitoring. Development of a new car model by a VE might involve approximately 20 000 designers and engineers from hundreds of divisions and departments, some of which belong to different enterprises in different countries. One sub-VE in the car-manufacturing VE performs car design, which contains four subprojects, namely, engine design, cool system design, transmission case design, and framework design. Engine designers in the engine design subproject collaboratively develop an engine for the new car model. Information related to the engine, such as drawing and engineering data, is generated and shared in real time to workers in the subproject and other subprojects. Therefore, the success of a VE depends wholly on transparent and effective sharing of information resources, including information, application systems, and knowledge, throughout the business cycle [1]. Not all business partners are equally trusted in today's complex business environment. Today's partners could become tomorrow's competitors. Hence, enterprises do not generally like sharing information. Consequently, a VE or related business strategy, such as allied concurrent engineering or virtual team, is likely to fail. Therefore, VE urgently needs secure and trustworthy AC model, approach, and mechanism that can manage distributed resources across enterprises and share them with collaborative workers. To secure information sharing, competitive and cooperative relationships among enterprises should be considered when using the proposed model to evaluate a user's authorization to access resources.

Secure resource management and sharing across organizational boundaries have seldom been addressed. AC for VEs is difficult for the following reasons: 1) enterprise members in a VE may change frequently; 2) each enterprise member

Manuscript received August 6, 2006; revised May 18, 2007. This work was supported in part by the National Science Council of the Republic of China, Taiwan, under Contract NSC96-2221-E-343-002. This paper was recommended by Associate Editor J. Miller.

T.-Y. Chen and C.-B. Wang are with the Department of Electronic Commerce Management, Nanhua University, Chia-Yi 62248, Taiwan, R.O.C. (e-mail: tsungyi@mail.nhu.edu.tw; cbwang@mail.nhu.edu.tw).

Y.-M. Chen is with the Institute of Manufacturing Engineering, National Cheng Kung University, Tainan 70101, Taiwan, R.O.C. (e-mail: ymchen@mail.ncku.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCA.2008.923090

(EM) typically has many roles and users; 3) a VE often has many EMs with complicated interrelationships—for example, members may cooperate and also compete with one another; 4) organizations within a VE may be dynamic and perform unpredictable activities; and 5) VE resources may be Internet based, distributed, and heterogeneous. Few studies have explored control of access to knowledge, which is one of the most important assets for an enterprise. Therefore, developing an AC mechanism for knowledge protection has been recognized as a vital research topic in knowledge management [6]–[8]. Although role-based models have been adopted successfully for resource management within an enterprise, collaborative VE systems using role authorization management approaches have not been widely investigated. In contrast to conventional AC models, AC for a VE does not specifically assign rights to each role or user in advance because of the dynamic characteristics of VE organizations, such as flexibility and mobile resource sharing. To our knowledge, no studies have developed models for resource sharing management that support collaborative and cooperative business activities across organizational boundaries. Before achieving secure resource sharing in a VE that increases corporate global competitiveness, several requirements for trust management, such as scalability, flexibility, dynamic security, decentralization, and mutual trust, must be addressed [9]. Hence, VEs require an appropriate AC model.

Based on the conceptual AC model in VEs [10], [11], this investigation develops a formal VEAC model to solve the problem of authorization management and to secure AC among organizations within a VE. The formal VEAC model comprises a fundamental VEAC model, a project AC policy (PACP) language model, and a model construction methodology. The proposed fundamental VEAC model comprises a [project-based access control (PBAC)] model for managing public resources within VE and a role-based AC (RBAC) model for managing the sharing of an individual enterprise's private resources with VE members. Public resources are generated, used, modified, and owned by VE activities and are stored or implemented in a VE or its partners. And, private resources are owned by partners and shared with other workers who could be from different partners. This PACP language model features object–subject–action–condition AC policies that jointly determine user access authorizations. Moreover, the methodology supplies a systematical method to identify fundamental elements of the VEAC model and establish assignments between elements and relations. The proposed formal VEAC model provides VE workers with efficient management and easy access to relevant resources and up-to-date information, thus eliminating information delay and enhancing information transparency.

II. Related Works

AC systems and technologies are required to protect such resources and information from illegal access. This section surveys a number of studies related to the aims of this paper, including AC, VE, and AC policy.

A. AC

AC protects the computing system against unauthorized access or modification of information resources [12]. AC determines whether a user has rights to use a given resource; an AC system governs when and how resources can be used by whom. So far, many AC methods had been presented.

Early AC methods for resource management include AC lists (ACLs) and the AC matrix (ACM). A simple ACM is an array containing one row per subject in the system and one column per object. Entries in the matrix specify the operation or access each subject has to each object [13]. These methods are straightforward, intuitive, and only useful for small organizations [14]. ACLs implement the ACM by representing the columns as lists of users attached to a protected object. Each object is associated with an ACL that stores all subjects and the subject's approved operations for a given object. Most AC models, including mandatory AC, discretionary AC, RBAC, task-based AC, and task RBAC [15]–[17], only consider authorization management within a single organization. Furst *et al.* [18] investigated distributed RBAC to delegate administration of resources to individual departments within an enterprise. In RBAC, users are assigned roles that are associated with approved permissions for performing an operation on an enterprise resource (object) [19]. Team-based AC 2004, derived from RBAC, enables users to join team roles within an organization [20].

B. VEs

A VE is defined as a cooperative alliance in which a group of legally independent enterprises, institutions, and individuals cooperate for a particular goal [21]. Ouzounis [22] defined VE as a network of different administrative business domains that cooperate by sharing business processes and resources to provide a value-added service to customers. VE environments (Fig. 1) contain users (subjects/workers) from various enterprises, such as EMs, partners, suppliers, customers, and other VEs. VE-related activities are undertaken by users from different enterprises using collaboration and concurrence. Such a business environment results in complex AC problems. In particular, all VE resources that may be stored on and owned by different enterprises should be managed fully and should be shared as much as possible.

1) Characteristics of VEs: Kanet *et al.* [21] decomposed the life cycle of a VE into five phases, namely, identification, formation, design, operation, and dissolution. Ouzounis [22] found that the life cycle of a VE should include two major phases: establishment and management. Based on the analysis of life cycle and interactions, a VE has the following characteristics [23]–[25].

1) A VE may consist of several distributed VEs or enterprises.
2) A VE's participating members and business processes may be changed during its life cycle.
3) A VE emphasizes professional division and dynamic cooperation among a highly heterogeneous membership.
4) A VE conducts business processes of different stages across enterprises, in which each stage has its own participants, resources, and aims.
5) Various resources in a VE are shared and distributed over all participating enterprises and used by their employees (users).
6) A VE globally specifies members' obligations, responsibilities, and roles.

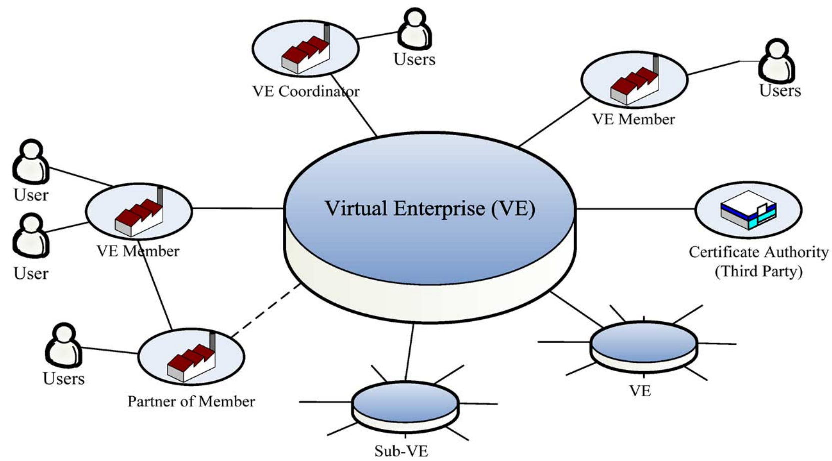


Fig. 1. VE environment.

7) A change in a member's role in a process should not affect the obligations and responsibilities in its other assigned roles.
8) Regulations do not constrain the selection of members in participating enterprises' partners.
9) Each member may own its enterprise resource management policy and AC model.
10) Shared VE resources include private resources owned by a participating enterprise and stored in its own repositories and also public resources belonging to the VE and stored in a public repository.

The levels of resource sharing among partners depend on VE characteristics, including levels of cooperation with partners, degree of trust, distributed tasks, and contractual agreements. When each participant in a VE brings information to the VE, the participant will not want to share more proprietary information than necessary with VE members because of information security issues. Information in VEs can be divided into three areas: 1) information of an individual partner brought to the VE; 2) information generated by the VE; and 3) information assets of the VE [26]. The information must be protected and distributed in a secure manner among all participants.

2) Requirement Analysis for AC in VE: Based on the general requirements of AC expressed in [27] and [28], this paper identifies the following requirements for AC model design: 1) Only the security administrator should be permitted to modify security attributes; 2) roles should be able to inherit authority either fully or partially; 3) positive authorizations and negative authorizations, as well as the principle of strict least privilege, should be supported; 4) the fine-grained authority requirements should be fulfilled; 5) access authority may change with tasks or roles; and 6) the model should be able to manage all users and resource objects in the enterprise [29]–[31].

Aside from the aforementioned requirements, according to the characteristics of VE, additional requirements must be considered when developing a VEAC model, as follows.

1) Since the organization structure of a VE is dynamic, access rights and resource objects can be changed in real time.
2) The model considers all users' access rights because resource administrators cannot predict who will access which resources in a VE.
3) As a VE is formed to achieve a certain goal in a limited time frame, each VE has different goal and business processes. A VE is always conducted as a project. Therefore, project is an essential element of AC in VE.
4) Since each enterprise has a legacy AC system, the VEAC model should be easily integrated with various AC models or policies.
5) The VE manages and shares resources collaboratively.
6) To facilitate trust among enterprises, the access policy in VE is planned and managed together by administrators of all participating enterprises.
7) The VE can maintain the consistency of policies and manage the conflicts between VE access policy and members' own access policies.

C. AC Policy

A significant shortcoming of existing AC systems is that they were developed by using a specific AC policy, which was defined by Lorch *et al.* [32], regarding how services can be utilized. AC policies are typically represented as follows: 1) constrained logic programs that support specific policy options; 2) constrained checks; and 3) administrator queries [33]. AC policy can restrict the use of services to suitably qualified principals and specify constraints that must hold when a service is invoked [19].

Recent development of AC policy framework includes languages and graphical approaches that specify different AC policies in a single framework [34]. A graph transformation-based security policy framework was proposed by Koch *et al.* [12] that included negative and positive constraints. The negative constraints specify graphs not contained in any system graph, and positive constraints specify graphs explicitly constructed in a system graph. By combining a formal framework and a logic-based language, Jajodia *et al.* [35] developed the authentication specification language that can be used to identify different AC policies that can coexist within the same system and be enforced by the same security server. Moreover, security assertion markup language is an XML framework identified by OASIS security services to exchange authentication and authorization information. For AC across enterprises, Belokosztolszki and Moody [36] proposed metapolicies. Hada and Kudo [37] proposed XML AC Language, an XML-based language for

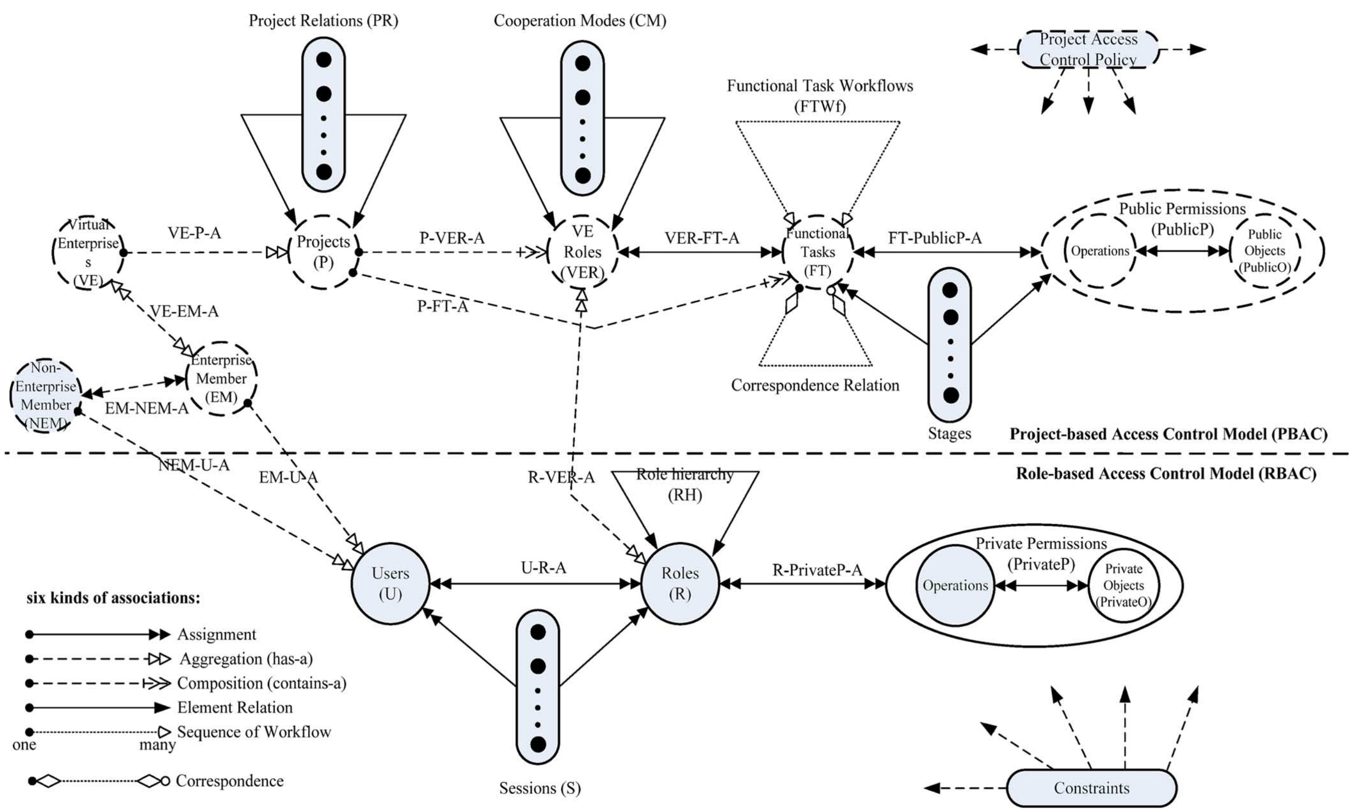


Fig. 2. Fundamental VEAC model.

provisional authorization, which articulates the security policies to be enforced for specific access to XML documents and provides XML with a sophisticated AC mechanism that enables an initiator to securely browse XML documents and securely update each document. Boella and van der Torre [38] studied normative multiagent systems for secure knowledge management based on AC policies.

III. Fundamental VEAC Model

This section introduces the proposed fundamental VEAC model, and its basic elements (Fig. 2), which has been derived from the requirements of AC for VE and characteristics of VE. It includes two submodels: one PBAC model for managing public resources stored on VE and one RBAC model for managing private resources stored on individual EMs [10]. In the model, the solid-line and the dashed-line circles are used to represent the elements in the RBAC and PBAC models, respectively. The six kinds of associations are proposed to indicate the various relationships among elements. Assignment is a well-known relationship in RBAC to continue using in the model. Aggregation is a grouping of other elements, which is also called a has-a association. For example, a VE is a group of EMs. Composition is an inclusion of other elements, which is also called a contains-a association. If the containing element is destroyed, the elements that it contains are also destroyed. Element relation is an interacting mode of other independent elements, which is further decomposed into various relations to facilitate resource sharing. Sequence of workflow is the order in which elements follow one another. Correspondence is a version mapping relation of a functional task (FT) in other project.

A. RBAC Model

This paper slightly adjusts the basic RBAC model [39]–[41] and seamlessly integrates it with the PBAC model. In the adjusted RBAC model, as shown in the bottom layer of Fig. 2, each element is described straightforwardly as follows.

1) User (U) represents a human or agent in an enterprise, which includes direct users, indirect users, and nonmember users.
2) Role (R) represents a functional job or responsibility.
3) Private object (PrivateO) denotes a resource in an enterprise associated with private privileges. Private objects are generally classified into three levels, which are public, proprietary, and protection. The public classification can be provided to the partners in a VE.
4) Private permission (PrivateP) is an approval of a particular mode of access to one or more private objects.
5) Session (S) maps a user to one or more roles.
6) $U{-}R{-}A \subseteq U \times R$ represents a many-to-many user to role assignment relation.
7) $R{-}\text{PrivateP}{-}A \subseteq R \times \text{PrivateP}$ represents a many-to-many role to PrivateP assignment relation.
8) $R_{\text{re}} = \{(x, y) : x, y \in R, x \neq y,$ and x conflicts with $y\}$ signifies that role x conflicts with role y, and x and y cannot be both assigned to the same user.
9) $R_{\text{rh}} = \{(x, y) : x, y \in R, x \neq y,$ and x is a superior of $y\}$ indicates that role x is a senior to role y, and x inherits the PrivatePs of y.
10) $U{-}R{-}A_u(r) : R \to 2^U$, a function mapping a role r to a set of users that can play this role.
11) $U{-}R{-}A_r(u)$: $U \to 2^R$, a function mapping a user u to a set of roles that can be played by this user.

12) $R-\text{PrivateP}-A_r(\text{private\_p}) : \text{PrivateP} \rightarrow 2^R$, a function mapping a PrivateP, private_p, to a set of roles that is authorized to access this PrivateP.
13) $R-\text{PrivateP}-A_{\text{private\_p}}(r) : R \rightarrow 2^{\text{PrivateP}}$, a function mapping a role r to a set of PrivatePs that allows to be accessed by this role.

B. PBAC Model

The top portion of Fig. 2 shows the PBAC model. The core concept of model development, elements, and relations in the PBAC model are introduced and defined in the following sections in order.

1) Core Concept of the PBAC Model: A VE can perform several projects (P) simultaneously, but a project can only be performed by one VE. A project includes management-level and operational-level tasks. The management-level tasks control and manage the project's progress and output according to the project timestamp, whereas the operational level comprises FTs supervised and controlled by the project schedule. Different project relations (PRs), such as subset, exclusion, and reference, exist among projects to facilitate resource sharing (refer to Section IV). Activities within a project can be divided into several FTs, each of which has access to certain public objects (PublicOs), which is public permission (PublicP) of the FT. FTs involved in a project are constructed for performing VE activities in the VE formation stage. The FTs are assigned to VE roles (VERs) that are virtual roles created based on division of labors. It is required to meet certain conditions to start or end an FT. According to the goal and task requirements, an FT can be divided into different stages by timestamp or FT. Users are given different privileges depending on the project stage and FTs. A VE is composed of several real EMs, each of which can participate in more than one VE. Non-EMs (NEMs) are enterprises that do not participate directly in the activities of VE but participate in the activities of an EM which performs directly the activities of the VE. All VE participants, including three user types (direct, indirect, and nonmember users), are generally called users (U) which may play a different role (R) in a different session. Each role has access to private resources, called a PrivateP. A superior role can inherit the privileges of inferior roles through role hierarchy (RH). The EM plays a VER through a user or role to obtain the privilege of sharing public resources in the VE and carry out practically the obligations of a given VER and to achieve the VE goals. PACP is designed to identify the resource sharing rules in a project. Through constructing relations among projects and a PACP, users can share resources among projects. The rules of sharing can be modified at any time.

To simplify the complex assignment and facilitate resource sharing across domains, some relations are gained by exploring the three viewpoints of project, VE, and enterprise. From the project viewpoint, PRs including subset, version, reference, process, and exclusive relations (defined in Section IV) are found out depending on the features of project, facilitating sharing among projects. From the VE viewpoint, cooperative relations including dependent single-task, dependent multitask, and independence (defined in Section V) are found out depending on the information requirements of interaction and cooperation among workers in VE, facilitating sharing among enterprises involved in a VE. From the enterprise viewpoint, relations proposed by RBAC [39], [40], including role hierarchy, static separation of duty, and dynamic separation of duty, are used herein to facilitate sharing among roles in an enterprise.

2) Fundamental Elements: This section concisely introduces the fundamental elements of the PBAC model, each of which is represented as follows.

1) VE $=$ {ve: ve represents a dynamic Internet organization consisting of EMs executing a project to achieve one common business goal}.
2) EM $=$ {em: em can be a substantive enterprise organization, a VE, or an individual, and it is a VE member with at least one worker participating directly in the VE activities}.
3) NEM $=$ {nem: nem can be a substantive enterprise organization, a VE, or an individual, but it is not a VE member; a nem has at least one worker participating directly in the activities of EMs, and the activities have direct relations with the FT of the VE}.
4) Project $(P) = \{p$: p denotes the set of FTs, projects, and subprojects performed by a VE}.
5) FT $=$ {ft: ft is a set of VE activities, which have a common objective and are undertaken by several VERs}.
6) VER $=$ {ver: ver represents a virtual role formed to enable professional division within VE, which is assigned to perform more than one FT}.
7) Object $(O) = \{o$: o denotes an information resource including public and private resources which can be a database, entity, attribute, tuple, document, XML document, application, software component, or knowledge}.
8) PublicO $=$ {public-o: public-o represents a subset of objects owned by a VE, stored in a VE's repository, and implemented in a VE's platform}.
9) Operation $=$ {op: op is a set of access authorities, such as write, read, and execute}.
10) PublicP $=$ {public-p: public-p represents a permitted mode of access to a PublicO}.
11) Permission $= \{x: x \in \text{PublicP} \cup \text{PrivateP}\}$.
12) PACP: PACP identifies which project resources are protected and shared according to the relations among projects and the shared rules and which activities are forbidden in the VE scope. Each project involves a PACP, which can be performed automatically by the VEAC system. The PACP can be dynamically created, enforced, and modified by administrators when the VE environment changes. The main rules described in PACP include the following: 1) rules of resource sharing among projects, describing the resource sharing strategy and relations among projects; 2) rules of resource usage in a project, including constraints on VERs, FTs, PublicPs, and assignments between elements; 3) rules of resource sharing of various cooperation modes, identifying the level of resource sharing according to the cooperation mode between VERs; and 4) rules of exception handling, which can be classified into rules of permitted exception handling and rules of forbidden exception handling. A PACP language model used to construct the PACP is shown in detail in Section VI.

3) Assignments and Relations: The following sections define the concept of assignments and relations between two elements involved in the model based on the concept of a product set (refer to Definitions 1 and 2). Some functions relating to all elements in the VEAC model are defined and then applied to the following sections. These functions are shown in Appendix I.

Definition 1: Given two sets A and B, the product set or Cartesian product of A and B, called the assignment of A and B in AC domain, is $A \times B = \{(a, b) : a \in A, \text{and } b \in B\}$.

Definition 2: Given sets A and B, a binary relation R from A to B is a subset of $A \times B$, i.e., $R \subseteq A \times B$.

4) Foundational Assignments: According to Definitions 1 and 2, the various assignment relations among elements are defined as follows.

1) FT$-S-$Public$-A \subseteq$ FT $\times S \times$ Public$-A$, triple assignment among three elements: FT, S, and PubicP, FT$-S-$Public$-A$ represents the set $R_{\text{ft-s-publicp-a}} = \{$(ft, st, public-p): ft $\in$ FT, st$\in$ Stage, public-p$\in$ PublicP, the public-p is assigned to ft in stage $s\}$.
2) $P-$VER$-A \subseteq P \times$ VER, one-to-many P to VER assignment, is denoted by $R_{\text{p-ver-a}} = \{(p, \text{ ver}): p \in P$, ver $\in$ VER, and p involves ver$\}$. The relation describes which VERs are included in project p.
3) VER$-$FT$-A \subseteq$ VER $\times$ FT, a many-to-many VER to FT assignment, is represented by $R_{\text{ver-ft-a}} = \{$(ver, ft): ver $\in$ VER, ft $\in$ FT, and ver performs ft$\}$. This relation describes which FTs are undertaken by which VERs.
4) VE$-$EM$-A \subseteq VE \times$ EM, a many-to-many VE to EM assignment, is denoted by $R_{\text{ve-em-a}} = \{$(ve, em): ve $\in$ VE, em $\in$ EM, and em is a member of ve$\}$.
5) VE$-P-A \subseteq$ VE $\times P$, one-to-many binary assignment from a VE to P, is represented by $R_{\text{ve-p-a}} = \{$(ve, p): ve $\in$ VE, $p \in P$, and ve performs $p\}$. This relation records which project is performed by a VE.
6) EM$-$NEM$-A \subseteq$ EM $\times$ NEM, many-to-many EM to NEM assignment, is represented by $R_{\text{em-nem-a}} = \{$(em, nem): em $\in$ EM, nem $\in$ NEM, and nem supports em to perform some tasks of the VE$-$EM$-A_{\text{ve}}$(em)$\}$. This relation holds the assignments between EMs and its partners (NEMs) to support the tasks of a VE.
7) FT workflow (FTWf) $\subseteq$ FT $\times$ FT, many-to-many binary relation on FT, is denoted by $R_{\text{FTWf}} = \{(\text{ft}_i, \text{ft}_j) : \text{ft}_i, \text{ft}_j \in \text{FT}, p_i, p_j \in P, \text{ft}_i \subset p_i, \text{ft}_j \subset p_j, i \neq j$, ft$_i$ is an event FT of the action FT ft$_j\}$ that indicates that ft$_j$ is authorized to use the PublicPs of ft$_i$ when ft$_i$ is accomplished.
8) Correspondence $\subseteq$ FT $\times$ FT, one-to-one binary relation on FT, is represented by $R_{\text{correspondence}} = \{(\text{ft}_i, \text{ft}_j) : \text{ft}_i, \text{ft}_j \in \text{FT}, p_i, p_j \in P, \text{ft}_i \subset p_i, \text{ft}_j \subset p_j, i \neq j$, ft$_i$ is the preversion offt$_j$, whereas ft$_j$ is the postversion offt$_i\}$.

5) Assignments Across Models: This section defines the assignment relations across models in order to establish the combination relations of relevant elements among two AC models. These relations are as follows.

1) EM$-U-A \subseteq$ EM $\times U$, one-to-many EM to U assignment, is represented by $R_{\text{em-u-a}} = \{(\text{em}, u)$: em $\in$ EM, $u \in U$, and em have an employee $u\}$. If $\exists$ em$_1 R_{\text{em-u-a}} u_1$, em$_2 R_{\text{em-u-a}} u_2$, em$_1$, em$_2 \in$ EM, and $u_1, u_2 \in U$, then $\neg\exists$ em$_2 R_{\text{em-u-a}} u_1$.
2) NEM$-U-A \subseteq$ NEM$\times U$, one-to-many NEM to U assignment, is denoted by $R_{\text{nem-u-a}} = \{(\text{nem}, u)$: nem $\in$ NEM, $u \in U$, and nem have an employee $u\}$. If $\exists$ nem$_1 R_{\text{nem-u-a}} u_1$, nem$_2 R_{\text{nem-u-a}} u_2$, nem$_1$, nem$_2 \in$ NEM, and u_1, $u_2 \in U$, then $\neg\exists$ nem$_2 R_{\text{nem-u-a}} u_1$.
3) $R-$VER$-A \subseteq R \times$ VER, many-to-many R to VER assignment, is represented by $R_{\text{r-ver-a}} = \{(r, \text{ver})$: $r \in R$, ver $\in$ VER, and r is assigned to play ver$\}$, then VER ver can be assigned to different roles, whereas one role can play different VERs at the same time.

IV. PRs

A PR (R_{p}) indicates the level of information exchange and reuse and also the situation of cooperation between two projects. Various PRs describing the relation between two projects can propagate the authorizations of an FT to other FTs. Different PRs may occur between two projects and may alter with time based on project management and share requirements. While a VEAC-based AC platform is implemented, administrators construct the project resource access strategy in a PACP to indicate the level of resource sharing of each type of PRs. In the project life cycle, the PRs and the PACP can be modified at any time to respond to the demands of resource sharing. Resource sharing or reusing is determined based on five attributes of each FT: 1) FT state (A_{state}) holds the status of the FT being performed; 2) FT stage (A_{stage}) records the current timestamp of an FT for appropriate resource sharing according to its states; 3) allowed reference (A_{ref}) decides whether the FT can be referred by relative FT in a postversion project; 4) allowed subproject ($A_{\text{sub-p}}$) determines whether the FT can be referred by its subprojects; and 5) allowed main project ($A_{\text{main-p}}$) decides whether the FT can be referred by its main project.

To introduce the PRs, given a set Project (P) and $x, y \in P$, a binary relation PR (R_{p}) on P is a subset of $P \times P$, which is distinguished into five subrelations presented in the following sections. For convenience in the following discussion, two inherited functions applied in the following sections are defined to indicate varying degrees of privilege inheritance.

1) Strong-inherited function Inher$_{\text{strong}}$(ft) is defined as all permissions assigned to the ft are inherited, including read (to retrieve data), update (to modify data), insert (to write new data), and create (to create an object).
2) Weak-inherited function Inher$_{\text{weak}}$(ft) is defined as only read permission from the ft is inherited.

A. Subset Relation

Subset relation (R_{ps}) describes the relation between a main project and its subproject. The relation simplifies a large number of assignments. For instance, an FT called announcement shows information about the progress of a project. Through the subset relation, all workers in the main project and subprojects of the project are permitted to look up the progress of the project. The set of pairs of projects between which have subset relation is represented by $xR_{\text{ps}}y = \{(x, y) : x, y \in P, x \neq y$, and x "is a subset of" $y\}$. A main project is permitted to access the resources of its subproject, but an administrator may set or disable the capability by changing the status of the allowed main-project attribute of its each FT. Fig. 3 shows an example of the subset relation to demonstrate these constraints,

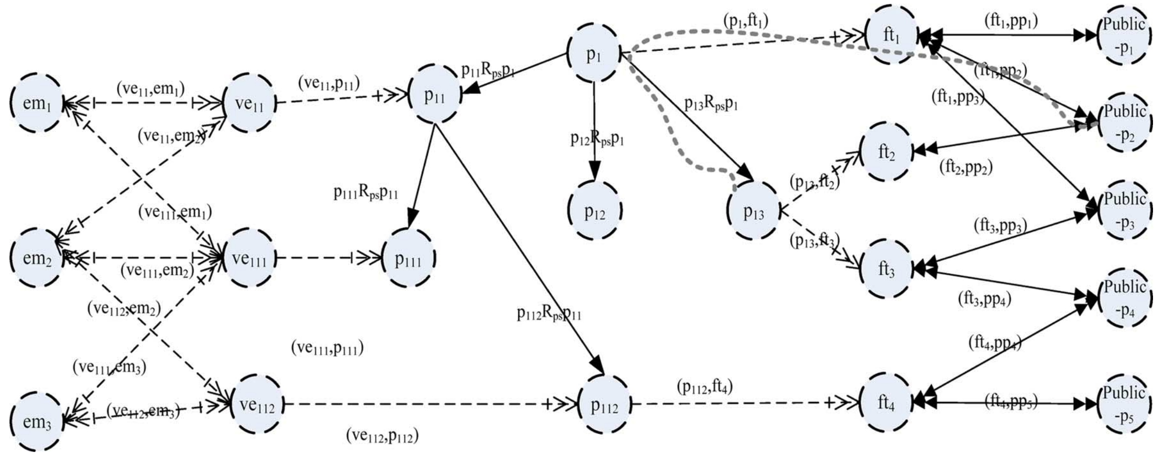


Fig. 3. Example of subset relation.

where Project p_1 involves three subprojects p_{11}, p_{12}, and p_{13}, and project p_{11} is further decomposed into subprojects p_{111} and p_{112}. Owing to $p_{13}R_{\mathrm{ps}}p_1$, PublicPs, such as public-p_2 and public-p_3, are assigned to p_1 and p_{13} via FTs, whereas EMs, such as em_1 and em_2, are permitted to participate in ve_{11} and ve_{111}. Two functions $\mathrm{privilege}_{\mathrm{main-p}}(\mathrm{ft})$ and $\mathrm{privilege}_{\mathrm{sub-p}}(\mathrm{ft})$ are defined, respectively, in (1) and (2), shown at the bottom of the page, for propagating user's privilege from the main project and subproject, respectively, where variables are introduced as follows. Function (1) indicates that the privileges of ft_{1i} involve the PublicP and PrivateP assigned to the ft_{1i} and $\mathrm{ft}_{2j}(1 \leq j \leq n)$ when the conditions shown in the equation hold; otherwise, the privileges of ft_{1i} only have the PublicP and PrivateP from ft_{1i}. Due to the limited space, function (2) shows the propagation of user privileges from subproject, which is similar to function (1) and is not further introduced in detail. p_1 is the main project of p_2 that is the subproject of p_1, ft_{1i}'s are the FTs involved in p_1, $1 \leq i \leq m$, and ft_{2j}'s are the FTs involved in p_2, $1 \leq j \leq n$. Several constraints are applied to use a subset relation: 1) A main project may have more than one subproject; 2) a subproject is only involved in one main project; 3) an EM may participate in the main and subprojects; and 4) a PublicP is only permitted to be assigned to different projects with subset relations.

B. Version Relation

Version relation (R_{pv}) describes a project y called a postversion project that is extended from a project x called preversion project and planned with reference to the preversion project. Therefore, the pre- and postversion projects have similar targets, FTs, and participants. The relation helps support version-dependent authorizations by enabling the reuse of resources for a new product, thus reducing its time to market. Because the pre- and postversion projects have similar targets, activities, and participants, the postversion FT in the postversion project corresponds to the preversion FT in the preversion project. While the postversion FT is performed, the privileges owned by the preversion FT are inherited by the postversion FT using the weak inheritance. The set of pairs of projects between which have version relation is represented by $xR_{\mathrm{pv}}y = \{(x, y) : x, y \in P, x \neq y,$ and x "is the preversion of" $y\}$. Fig. 4 shows an example of the version relation, which demonstrates that project p_1 is the preversion of project p_2. Project p_1 for developing a car engine consists of FTs ft_{11} and ft_{12}, whereas p_2 for developing a new engine based on the engine developed by p_1 comprises ft_{21}, ft_{22}, and ft_{23}. FTs ft_{11} (requirement analysis) and ft_{12} (conceptual design) correspond to ft_{21} (requirement analysis) and ft_{22} (conceptual design), respectively, whereas ft_{23} (primary design) is created for another task, which is not extended from p_1. Therefore, while the ft_{21} performed, workers must refer significantly to information owned by ft_{11}. Due to $p_1R_{\mathrm{pv}}p_2$, each FT in project p_2 is performed by VERs, which are allowed to refer to PublicPs of corresponding FTs in p_1 if the attribute allowed reference of corresponding FT is true. As shown in Fig. 4, a user u_1 is assigned to perform the ft_{21} through (u_1, r_1), (r_1, ver_{21}), and $(\mathrm{ver}_{21}, \mathrm{ft}_{21})$; in addition to the public-p_{21} and public-p_{22}, u_1 may refer to the public-p_{11}, public-p_{12}, and public-p_{13}. Function (3) shown at the bottom of the next page is presented to indicate that the privileges of ft_{2j} involve the PublicP and PrivateP assigned to the ft_{2j}, and partial PublicPs of the corresponded FT ft_{1i} of ft_{2j} through the use of weak inheritance function when the conditions shown in the function hold; otherwise, the privileges of ft_{2j} only have the PublicP and PrivateP from ft_{2j}. p_1 is the preversion project of p_2 that is the postversion project of p_1, ft_{1i}'s are the FTs involved in p_1, and ft_{2j}'s corresponding to ft_{1i}'s are the FTs involved in p_2. Several constraints are applied when using the version relation to support resource sharing between two projects: 1) A postversion project has less than one preversion

$$\mathrm{privilege}_{\mathrm{main-p}}(\mathrm{ft}_{1i}) = \begin{cases} \mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{1i}) \cup \mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{2j}) & \text{if } \exists\, p_1 R_{\mathrm{ps}} p_2 \wedge A_{\mathrm{main-p}} \text{ of } \mathrm{ft}_{2j} = \text{“true”} \\ \mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{1i}) & \text{otherwise} \end{cases} \tag{1}$$

$$\mathrm{privilege}_{\mathrm{sub-p}}(\mathrm{ft}_{2j}) = \begin{cases} \mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{2j}) \cup \mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{1i}) & \text{if } \exists\, p_1 R_{\mathrm{ps}} p_2 \wedge A_{\mathrm{sub-p}} \text{ of } \mathrm{ft}_{1i} = \text{“true”} \\ \mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{2j}) & \text{otherwise} \end{cases} \tag{2}$$

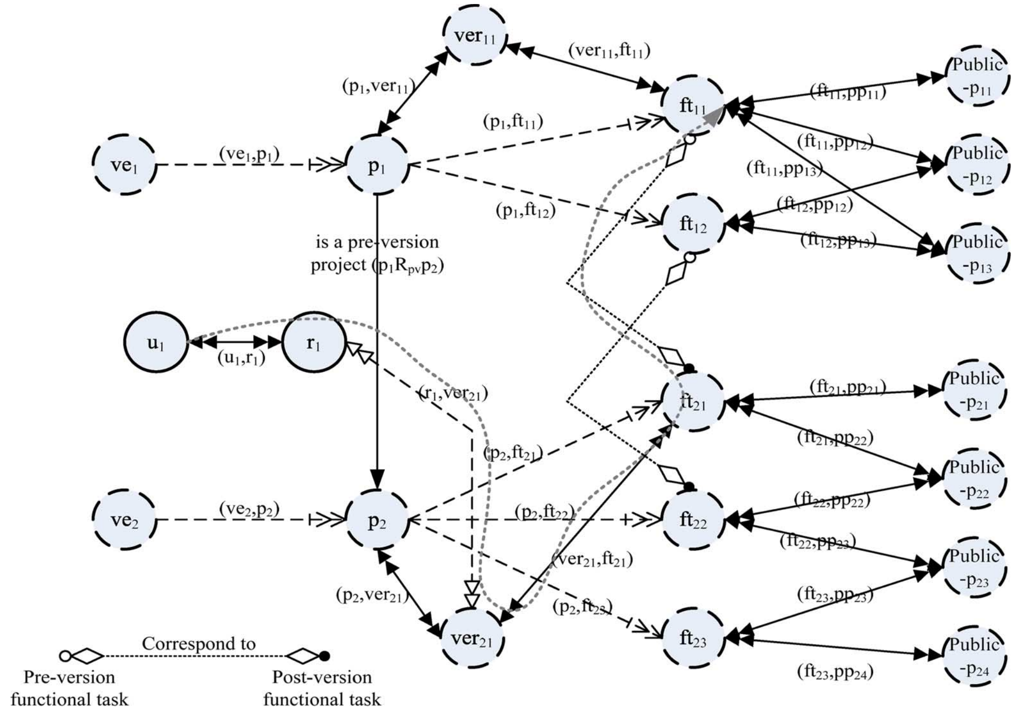


Fig. 4. Example of version relation.

project contrariwise; 2) an FT has less than one corresponding FT; and 3) an EM may participate in pre- and postversion projects simultaneously.

C. Reference Relation

Reference relation (R_{pr}) describes a project x called referring project referring to resources in other project y that is called the referred project. The reference relation indicates that the same users and enterprises can participate in both the referring and referred projects. If two projects have a reference relation, then users in the referring project can refer to the resources of the referred project. While the value of attribute allowed reference of an FT equals true, then the FT can be referred. The set of pairs of projects between which are referred by each other is represented by $xR_{\mathrm{pr}}y = \{(x, y) : x, y \in P, x \neq y$, x refers to resources in y, and $(\neg\exists\, xR_{\mathrm{pe}}y) \wedge (\neg\exists\, yR_{\mathrm{pe}}x)\}$. Project x may refer to y if and only if the following conditions hold: $R_{xi} \cap R_{yj}$, $\mathrm{EM}_{xm} \cap \mathrm{EM}_{yn}$, $\mathrm{FT}_{xk} \cap \mathrm{FT}_{yh}$, $\mathrm{PublicP}_{xv} \cap \mathrm{PublicP}_{yw}$, and $\mathrm{PrivateP}_{xe} \cap \mathrm{PrivateP}_{yf}$ permit unequal ϕ, where R_{xi}, EM_{xm}, FT_{xk}, $\mathrm{PublicP}_{xv}$, and $\mathrm{PrivateP}_{xe}$ are associated with project x, and R_{yj}, EM_{yn}, FT_{yh}, Public-P_{yw}, and Private-P_{yf} are associated with project y. That is, roles, EMs, FTs, PublicP, and PrivateP may be assigned to p_1 and p_2. Fig. 5 shows an example of the reference relation, which indicates that project p_1 can refer to project p_2 through the reference relation $p_1R_{\mathrm{pr}}p_2$. Role r_{31} is assigned to perform VERs ver_{11} and ver_{21}, ver_{11} performs FTs ft_{11} and ft_{12} in project p_1, and ver_{21} performs ft_{21} in project p_2. Therefore, user u_{31} may utilize the public-p_{11}, public-p_{12}, public-p_{13}, public-p_{21}, and public-p_{22} through (u_{31}, r_{31}). The following constraints are applied when using the reference relation: 1) A project may be assigned to more than one project for resource sharing, and 2) a project may refer to more projects simultaneously.

D. Process Relation

Process relation (R_{pp}) describes the executive sequence of two subprojects from the time view and can determine the time for sharing project resources. A process relation can be applied to determine the executive sequence of all subprojects of a project. The set of pairs of projects between which have process relation is represented by using $xR_{\mathrm{pp}}y = \{(x, y) : x, y, z \in P, x \neq y \neq z, (\exists\ xR_{\mathrm{ps}}z) \wedge (\exists\ yR_{\mathrm{ps}}z)$, and x "must be achieved, then start" $y\}$. While the relation is built on two projects, the administrator must specify the sequences of related FTs across the project boundary. This relation can support process-dependent authorization propagation when executing an action FT that can use the resources of the event FTs in event project. Fig. 6 shows an example of a process relation, in which project p_1 denotes the event project of action project p_2; p_1 performs ft_{11} and ft_{12}, and p_2 performs ft_{21}, ft_{22}, and ft_{23}; and ft_{11} denotes an event FT that triggers the ft_{21} and ft_{22} (called action FTs). When ft_{21} is

$$\mathrm{privilege}_{\mathrm{version}}(\mathrm{ft}_{2j}) = \begin{cases} \mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{2j}) \cup \mathrm{Inher}_{\mathrm{weak}}\left(\mathrm{FT{-}PublicP{-}}A_{\mathrm{public\_p}}(\mathrm{ft}_{1i})\right) & \text{if } \exists\, p_1 R_{\mathrm{pv}} p_2 \wedge A_{\mathrm{ref}} \text{ of } \mathrm{ft}_{1i} = \text{“true”} \\ \mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{2j}) \text{ otherwise} & \end{cases} \quad (3)$$

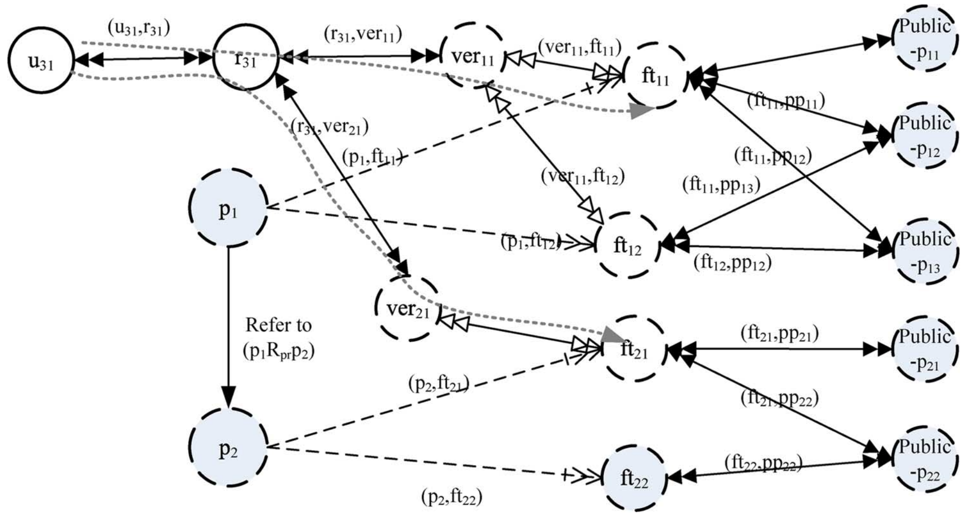


Fig. 5. Example of reference relation.

triggered and performed, user u_1 obtains authorizations public-p_{11}, public-p_{12}, and public-p_{13} from ft_{11}, and authorizations public-p_{21} and public-p_{22} from ft_{21}.

Function (4) shown at the bottom of the page, showing the propagation of user privilege by using process relation, is presented to indicate that the privileges of $\mathrm{ft}_{\mathrm{action}}$ involve the PublicP and PrivateP assigned to the $\mathrm{ft}_{\mathrm{action}}$ and also partial permissions of the event FT $\mathrm{ft}_{\mathrm{event}}$ of the $\mathrm{ft}_{\mathrm{action}}$ through the use of the weak inheritance function when the conditions shown in the function hold; otherwise, the privileges of $\mathrm{ft}_{\mathrm{action}}$ only have the PublicP and PrivateP from $\mathrm{ft}_{\mathrm{action}}$. p_1 is the action project of p_2 that is the event project of p_1, $\mathrm{ft}_{\mathrm{action}}$ is the action FT included in p_1, and $\mathrm{ft}_{\mathrm{event}}$ is the event FT included in p_2. Using the process relation must obey the following constraints: 1) A process relation exists between two projects which must have the subset relation; 2) an event project may trigger more than one action project simultaneously; 3) an event FT may trigger more than one action FT simultaneously; and 4) an action project may be triggered if all of its event projects are accomplished.

E. Exclusive Relation

Exclusive relation (R_{pe}) identifies mutual conflict between two projects, signifying that the resources of the two projects cannot refer to each other. The exclusive relation is default. That is, two projects are preset as exclusive relation if no other relation exists between them. The set of pairs of projects that conflict with each other is represented by $xR_{\mathrm{pe}}y = \{(x, y) : x, y \in P, x \neq y, x$ "conflicts with" y, and $(\neg\exists\ xR_{\mathrm{pr}}y) \wedge (\neg\exists\, yR_{\mathrm{pr}}x)\}$. If two projects are exclusive, then all users, EMs, FTs, and permissions in a project are exclusive with the other project. That is, an enterprise is disallowed from participating simultaneously in two projects with exclusive relation; attempts by users of the exclusive projects to use the same resources are rejected. Using the process relation must obey the following constraints: 1) A project may conflict with more than one simultaneously; 2) a PublicP may not be assigned to two exclusive projects; and 3) an EM is not allowed to be assigned to two mutual exclusive projects.

V. Cooperation Modes Among Two VERs

This section introduces three cooperation modes among VERs based on the resource sharing requirements of collaborative operations in the VE.

Cooperation mode (R_{c}) describes interactions among VERs according to the dependent level of their duties. Given a set VER, x and $y \in$ VER, a binary relation cooperation relation (R_{c}) on VER is a subset of VER $\times$ VER, which is distinguished into three cooperation relations. For convenience in the following discussion, two items are first defined in terms of authority inheritance. A VER in cooperative mode can inherit strongly or weakly the privileges from the other VER. Strong inheritance means that the privilege of a VER can be fully inherited by the other VER, whereas weak inheritance means that the privilege can only be partially inherited, such as only inheriting read privilege.

1) Dependent single-task mode (R_{cds}) is the most seamless cooperative relationship between two VERs, working together to perform FTs, that have dependencies and share resources with each other. The two VERs' permissions are inherited from each other via strong inheritance (defined in Section IV). When two VERs collaboratively perform different FTs, the users playing the two VERs obtain the same permissions from the FTs. The set of pairs of VERs with R_{cds} is represented by using $xR_{\mathrm{cds}}y = \{(x, y) : x, y \in \mathrm{VER}, x \neq y, \exists(x, \mathrm{ft}_1), (y, \mathrm{ft}_1) \in$

$$\mathrm{privilege}_{\mathrm{process}}(\mathrm{ft}_{\mathrm{action}}) = \begin{cases} \mathrm{FT}{-}\mathrm{Permission}{-}A(\mathrm{ft}_{\mathrm{action}}) \cup \mathrm{Inher}_{\mathrm{weak}}\,(\mathrm{FT}{-}\mathrm{Permission}{-}A(\mathrm{ft}_{\mathrm{event}})) \\ \text{if } \exists(p_1 R_{\mathrm{pr}} p_2) \wedge (A_{\mathrm{state}} \text{ of } \mathrm{ft}_{\mathrm{event}} = \text{"achieved"}) \wedge (A_{\mathrm{ref}} \text{ of } \mathrm{ft}_{\mathrm{event}} = \text{"true"}) \\ \mathrm{FT}{-}\mathrm{Permission}{-}A(\mathrm{ft}_{\mathrm{action}}) \text{ otherwise} \end{cases} \quad (4)$$

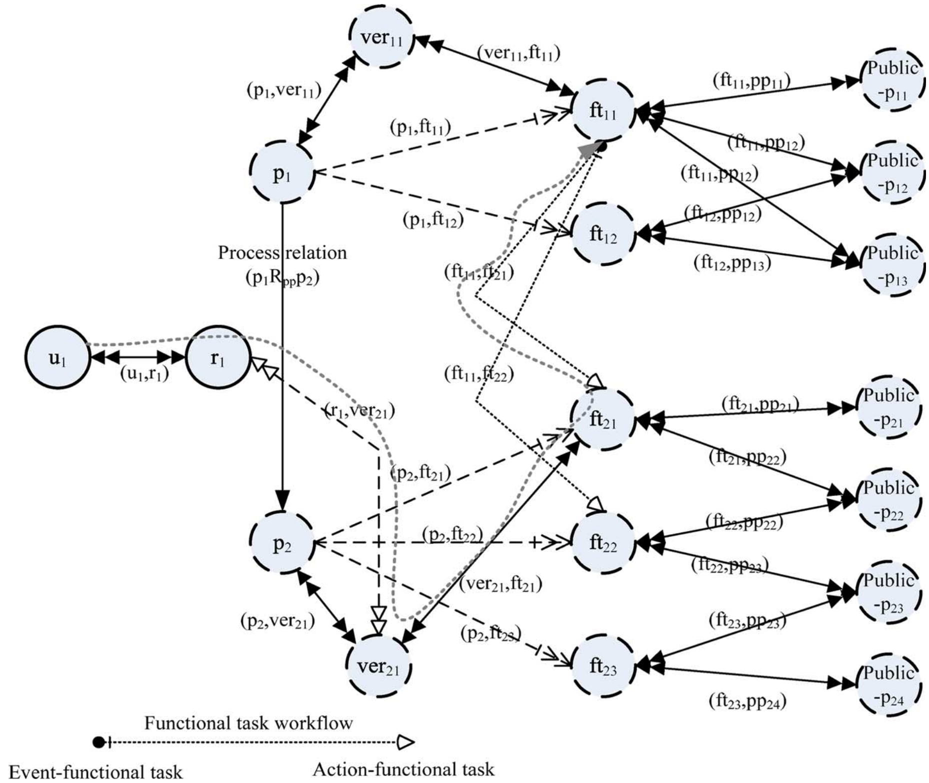


Fig. 6. Example of process relation.

$\text{VER}-\text{FT}-A \rightarrow \text{FT}-\text{PublicP}-A_{\text{public\_p}}(\{\text{VER}-\text{FT}-A\_\text{ft}(x) : (x, \text{ft}) \in \text{VER}-\text{FT}-A\})$ are inherited strongly by VER y, and $\text{FT}-\text{PublicP}-A_{\text{public\_p}}(\{\text{VER}-\text{FT}-A\_\text{ft}(y) : (y, \text{ft}) \in \text{VER}-\text{FT}-A\})$ are inherited strongly by VER x, and $(\neg\exists\ xR_{\text{cdm}}y) \wedge (\neg\exists\ yR_{\text{cdm}}x) \wedge (\neg\exists\ xR_{\text{ci}}y) \wedge (\neg\exists\ yR_{\text{ci}}x)\}$ means that VERs x and y cooperate to perform an FT ft_1 and have the same access privilege to all its resources.

2) Dependent multitask mode (R_{cdm}) indicates that two VERs interact when performing different FTs. For instance, the results of an FT performed by a VER affect those of an FT performed by another VER. The two VERs inherit each other's permissions via weak inheritance. The set of pairs of VERs with R_{cdm} is represented by using $xR_{\text{cdm}}y = \{(x, y) : x, y \in \text{VER}, x \neq y, \forall(x, \text{ft}_x)$, $(y, \text{ft}_y) \in \text{VER}-\text{FT}-A \rightarrow \text{FT}-\text{PublicP}-A_{\text{public\_p}}$ $(\{\text{VER}-\text{FT}-A_{\text{ft}}(x) : (x, \text{ft}_x) \in \text{VER}-\text{FT}-A\})$ are inherited weakly by VER y, and $\text{FT}-\text{PublicP}-A_{\text{public\_p}}$ $(\{\text{VER}-\text{FT}-A_{\text{ft}}(y) : (y, \text{ft}_y) \in \text{VER}-\text{FT}-A\})$ are inherited weakly by VER x, and $(\neg\exists\ xR_{\text{cds}}y) \wedge (\neg\exists\ yR_{\text{cds}}x) \wedge (\neg\exists\ xR_{\text{ci}}y) \wedge (\neg\exists\ yR_{\text{ci}}x)\}$. Hence, VERs x and y perform related FTs separately, and that outputs of the FTs are referred to each other.

3) Independent mode (R_{ci}) indicates that two VERs independently perform their FTs, disregarding the outputs generated by other FTs. The relation is applied to protect business secrets when companies that compete with each other perform VERs. If the two VERs work independently, then they are not permitted to perform the same FTs and have each other's access privileges for FTs performed by them. The set of pairs of VERs between which have R_{ci} is represented by $xR_{\text{ci}}y = \{(x, y) :$ $x, y \in \text{VER}, x \neq y, T-\text{PublicP}-A_{\text{public\_p}}$ $(\{\text{VER}-\text{FT}-A_{\text{ft}}(x) : (x, \text{ft}_x) \in \text{VER}-\text{FT}-A\})$ are not inherited by VER y, and $\text{FT}-\text{PublicP}-A_{\text{public\_p}}(\{\text{VER}-\text{FT}-A_{\text{ft}}(y) : (y, \text{ft}_y) \in \text{VER}-\text{FT}-A\})$ are not inherited by VER x, and $(\neg\exists\ xR_{\text{cds}}y) \wedge (\neg\exists\ yR_{\text{cds}}x) \wedge (\neg\exists\ xR_{\text{cdm}}y) \wedge (\neg\exists\ yR_{\text{cdm}}x)\}$.

The use of cooperative relations is constrained by the following rules.

1) $\#(\{y : (x_1, y) \in R_c, x_1, y \in \text{VER}\}) \geq 0$ means that a VER x_1 is permitted to have different cooperation modes with other VERs.
2) $\#(\{(x_1, y_1) : (x_1, y_1) \in R_c, x_1, y_1 \in \text{VER}\}) \leq 1$ signifies that only one cooperation mode is permitted between two VERs.

VI. PACP Language Model

Based on the VEAC model, the PACP language model for VEs designed in this paper, as Fig. 7 shows, is represented in class model of Unified Modeling Language (UML) and mainly targets contents of information text. This model features an object–subject–action–condition AC policy consisting of multiple sets of authorization rules that jointly determine user access permissions. Therefore, regarding specific resource (object), authorization (action) to execute certain resource is granted to certain users (subject) under certain restrictions (conditions).

The PACP language model for VEs has been proposed in this section for the following reasons: 1) to provide a method that effectively describes resource AC policy for VEs; 2) to reduce costs and complexity in resource AC; 3) to improve flexibility in managing access permission; and 4) to make the management of resource access permission adaptive to changing needs in a

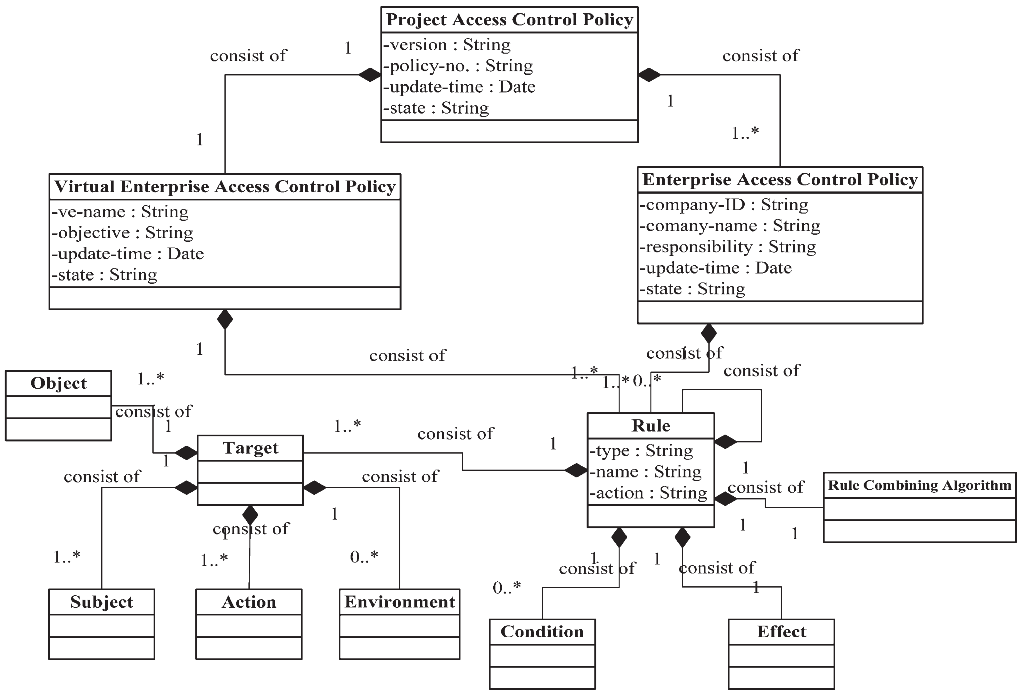


Fig. 7. PACP language model.

business environment in a timely manner. The PACP language model has the following main components.

1) PACP. A PACP consists of one VEAC policy (VEACP) and many enterprise AC policies (EACPs), which are sets of rules.
2) VEACP, a set of rules, describes the regulation and constraint on resource AC and sharing in a VE to manage the VE's resource.
3) EACP, consisting of a series of rules, describes rules and conditions for enterprise resource AC for each EM. Its rules shall not be in conflict with the VEACP it belongs to and must comply with the sharing rules agreed upon by VE so to make available resource in need of sharing.
4) Rule element is the most basic unit of policy and corresponds to the conventional concept of authorization. The principal components of rule have a target, effect, condition, and rule combining algorithm. Each rule permits or denies one or more subjects to performing actions on one or more objects under some conditions.
5) A target element involved in a rule defines the set of objects, subjects, and actions to which the rule or policy applies.
6) Object may be data, information, and knowledge owned by the VE or one of its EMs.
7) A subject is an actor whose attributes may be referenced by a predicate. Actor may be a user, role, enterprise, or VER.
8) An action is an operation on resource.
9) A condition element represents additional constraints that further refine rule applicability.
10) Rule combining algorithm compresses the output from the embraced rules. The PACP language model has four rule combining algorithms: deny overrides, permit overrides, first applicable, and only-one-applicable. Based on the selected combining algorithm, an authorization decision can be permit, deny, not applicable, or indeterminate.
11) Effect is the intended consequence of a satisfied rule—either Permit or Deny.

VII. VEAC Model Construction Methodology

The proposed formal VEAC model can efficiently manage and share information resources in the VE life cycle. To assist the administrators of VEs and their EMs to successfully implement the proposed fundamental VEAC model and to use the PACP language model appropriately for VE information resource security and sharing, this section develops a VEAC model construction methodology based on the five phases of VE life cycle, namely, identification, formation, design, operation, and dissolution phases. The methodology provides the security administrators of the leader and partners of VEs with a systematic method for the following reasons: 1) to identify the fundamental elements of VEAC model, such as P, VER, FT, U, R, PublicP, and PrivateP; and 2) to establish assignments between elements, PRs between projects, and cooperation modes between VERs. The VEAC model applied for certain VE is initially planned at the formation phase, all elements and assignments of the VEAC model are designed at the design phase, and the constructed VEAC model is implemented at the operation phase. Thus, information resources are managed at the operation and dissolution phases. The goal, procedure, inputs, outputs, and related method and technologies of each phase of the methodology are separately introduced in the following sections.

A. *Identification and Formation Phases*

Fig. 8 shows the first two phases in the proposed methodology, namely, identification and formation phases, which are

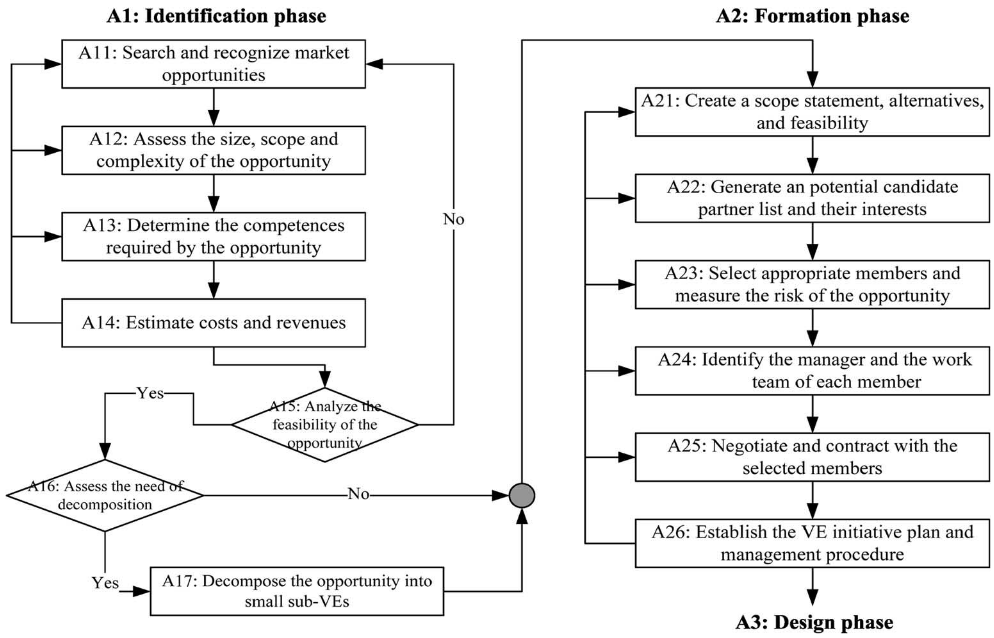


Fig. 8. Identification and formation procedures of a VE.

introduced simply as follows.

1) Identification phase numbered A1 defines the boundaries of a VE to analyze whether the goals, technologies, and cost of the VE are acceptable; to evaluate the complexity of the VE; and to establish procedures for supporting later VE activities. The leader of a VE generally analyzes historical transaction data or carries out market research to find out a valuable and feasible market opportunity and then to form a VE. To achieve the aims of the identification phase, the following seven numbered actions (the left of Fig. 8) should be undertaken in order or repetitively: (A11) searching and recognizing market opportunities; (A12) assessing the size, scope, and complexity of the opportunity; (A13) determining the competences required by the opportunity; (A14) estimating costs and revenues; (A15) analyzing the feasibility of the opportunity; (A16) assessing the need of opportunity decomposition; and (A17) decomposing the opportunity into small sub-VEs to perform the decomposed opportunities, thus establishing R_{ps} between the main and subprojects. The final output of the phase is a practical and valuable opportunity.
2) Formation phase numbered A2 selects suitable partners against alignment factors for their skills, experiences, and capabilities; identifies each member's responsibilities explicitly; ensures that every member of the VE understands his own individual roles and responsibilities; and allocates project resources, including people, service, facilities and equipment, supplies and materials, and money. To accomplish this process at the formation phase, the following six numbered actions (the right of Fig. 8) should be executed in order or repetitively: (A21) creating the scope statement, alternatives, and feasibility of a VE; (A22) generating a potential candidate partner list and their interests; (A23) selecting appropriate partners for the VE and its sub-VEs and measuring the possible risk from the partners; (A24) identifying the VE manager and work team of each partner; (A25) negotiating and contracting with the selected partners for sharing and using resources; and (A26) establishing the initiative plan and management procedure of a VE and its sub-VEs. The final outputs of achieving the six actions include a certain VE organizational structure model and contracts for cooperation among all EMs. The design phase is then executed based on this model.

B. Design Phase

The design phase in the proposed methodology is a significant phase for constructing a real VE based on the proposed VEAC model, since it is relative mostly to the plan and design, and resource use and assignment of VEs (Fig. 9). The actions involved at the phase are achieved collaboratively by the security administrators of the VE leader and all partners for managing public and private resources and VE user authorizations. The design phase numbered A3, which includes three subprocedures A31, A32, and A33, is described as follows.

1) Subprocedure A31—Plan and Design VE: The subprocedure models a VE in terms of organization, business, process, and activity perspectives. The detailed organizational structure model, business and resource sharing regulations, VE process model, and activity models of each EM are produced at the end of subprocedure A31. The subprocedure involving six actions is numbered and described below.

(A311) Identify all participators of each partner. Each partner in the VE is assigned certain tasks or responsibilities at the formation phase. The subprocedure starts with action A311 from the organizational view, in which each partner has to choose suitable employees or teams to perform enterprise-assigned tasks, according to employees' skills, experiences, and capabilities. At the time, partners must offer a list of employees who participate directly or indirectly in the VE and are permitted to access the VE resources. The employees involved in the list become user elements in the VEAC model.

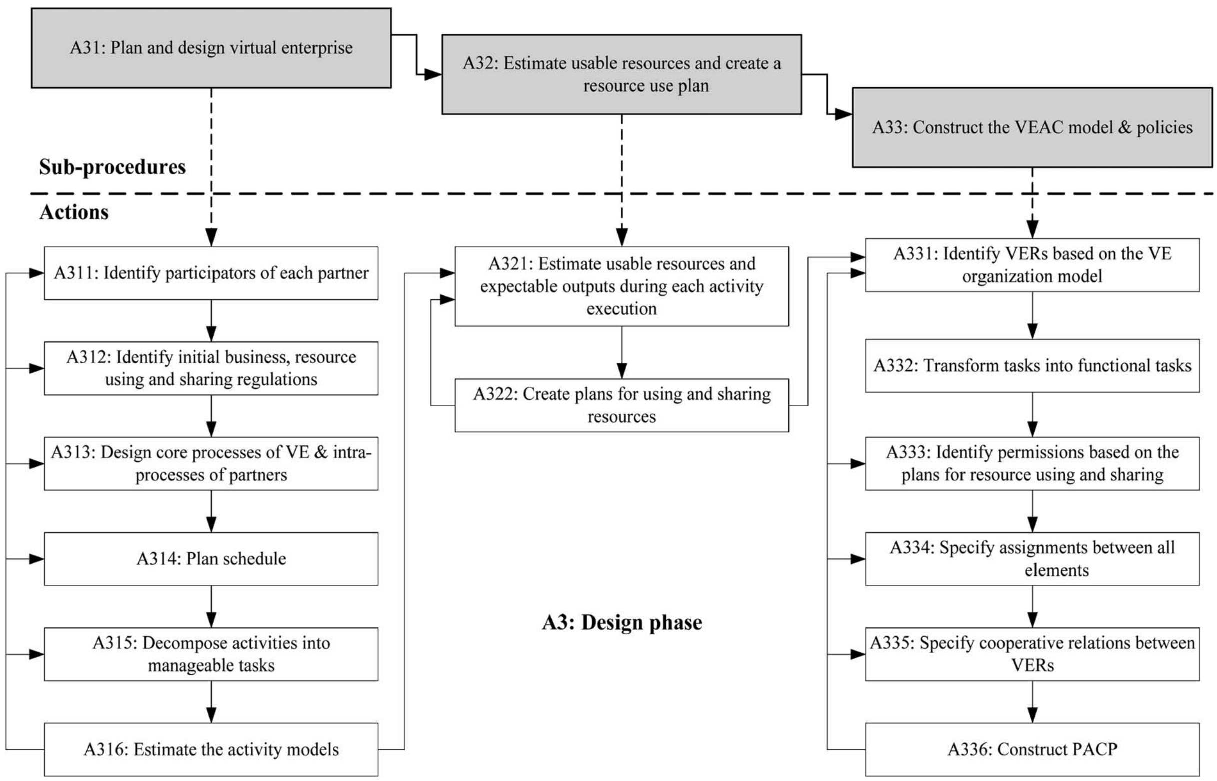


Fig. 9. Design procedure of a VE based on the VEAC model.

(A312) Identify initial business and resource using and sharing regulations. From a business perspective, action A312 identifies regulations regarding usage and sharing of resources to restrict the behavior of partners and specify each participator's responsibility and obligations. In the VE organizational structure model, every participator in a VE is assigned certain tasks, which are performed and restricted by the regulations. The regulations are then converted into VEACP and EACPs by A336.

(A313) Design the core processes of VE and the intraenterprise processes of partners. Based on the planned VE process at A2, a VE leader at the phase designs the core processes of the VE project represented by a project evaluation and review technique chart. The core processes are composed of many activities to accomplish VE's goal. Each activity in the core process is assigned to certain partners to perform. Each partner must then spread up and perform its assigned activities and integrate them into its intraenterprise processes. Finally, PRs R_{pv}, R_{pr}, R_{pp}, and R_{pe} can be established at action A313 if they are needed.

(A314) Plan schedule. According to the core VE processes designed by A313, the VE leader at the action negotiates and communicates with partners to plan the start and end times of each activity in the core VE and intraenterprise processes, and the activity prerequisites.

(A315) Decompose activities into manageable tasks. The activities involved in the core VE and intraenterprise processes are further decomposed into tasks until every task can represent a manageable amount of work that can be planned, scheduled, and assigned. A work breakdown structure, comprising a hierarchical decomposition of project, activities, and tasks, is planned at this point. The decomposed tasks are then further decomposed or combined into manageable tasks in terms of resource AC. The priority of every manageable task is determined from the start and end times of the original tasks, the information flow between tasks and task outputs.

(A316) Estimate the activity models. An activity model is composed of some partially ordered tasks that are conducted to achieve the actions to be performed within a VE. Action A316 estimates the duration of every task and changes the baseline based on reasonable estimations. The following factors should be addressed: 1) the resources that should be used; 2) the amount of time required; 3) how many people are needed; 4) the skills that are necessary; and 5) the tasks that need to be completed before other tasks are started. Subprocedure 2 is executed after all tasks are estimated.

2) Subprocedure A32—Estimate Usable Resources and Create a Resource Use Plan: Subprocedure A32 estimates the usable VE resources and builds a resource use plan for the entire life cycle of a VE. The plan is adopted to restrict assignments between elements and to build the PACP.

(A321) Estimate usable resources and expectable outputs. The first action of this subprocedure estimates usable VE resources according to the activity models outputted by A316. These resources include public and private resources, which are supplied or shared with partners to facilitate the execution of VE tasks. In addition, the administrator has to expect possible outputs during the execution of each task and know whom the outputs will be shared with. Some shared outputs should be specified by specific data containers, which are then converted into permissions and assigned operations permitted on them at A333.

(A322) Create plans for using and sharing resources. Based on the regulations created by A312, the result of A321, and

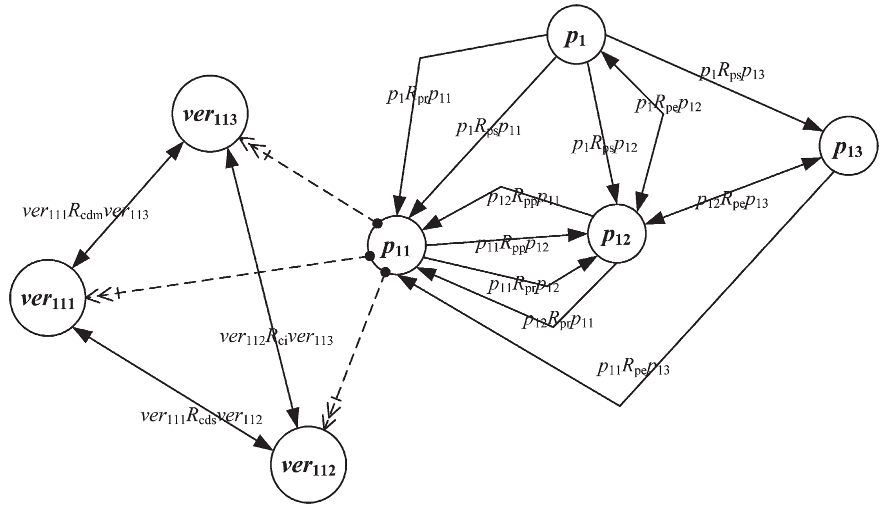


Fig. 10. Engine R&D project.

the activity models of A316, the action forms a resource using and sharing plan, which describes which users, activities, and partners can use or share which resource.

3) Subprocedure A33—Construct the VEAC Model and Policies: Subprocedure A33 identifies the elements, assignments, relations, and policies involved in the VEAC model according to the outputs at subprocedures A31 and A32.

(A331) Identify VERs based on the VE organizational model. VERs can be identified by using two different methods: decomposing the VE's goal or decomposing the VE's organizational structure. In the first method, the aim of a VE is decomposed into little goals that can be completed by a single individual or team. Each of these little goals is then transformed into a VER. In the second method, the organizational structure of a VE is decomposed hierarchically into different levels of element, namely, EM, department, team, role, and user. Several elements in the same level are then chosen to form a VER if they can be assigned to different workers; they have the same authorizations, and they do not have resource security problems resulting from sharing or collaboration. Finally, those single elements that cannot be assigned to different workers are converted into VERs.

(A332) Transform tasks into FTs. Action A332 transforms the manageable tasks in activity models into FTs, whose properties must be filled in. If the resources of a manageable task simultaneously allow and disallow sharing, then the manageable task must be decomposed into two or more FTs.

(A333) Identify permissions based on the plans for resource using and sharing. Action A333 combines resources and assigns operations on the resources to form PublicP and PrivateP. The PublicPs are identified by a VE leader administrator, and the PrivatePs are identified by every partner's administrator.

(A334) Specify assignments between all elements. All elements involved in the VEAC model have been identified at previous actions. This action specifies all assignments between two elements, such as VE$-P-A$, $P-$VER$-A$, VER$-$FT$-A$, FT$-$PublicP$-A$, and $R-$VER$-A$.

(A335) Specify cooperative modes between VERs. The cooperative modes between two VERs are specified here according to the resource usage and sharing plan.

(A336) Construct PACP. Based on the proposed PACP language model, the action utilizes the business regulations and the resource using and sharing plan to build the PACP of the VE, including a VEACP and several EACPs. The VEACP is built by the administrator of the VE leader, and the EACPs of partners are built by their administrators.

C. Operation and Dissolution Phases

1) Operation phase first sets up the real VEAC model modeled in the third phase. The VEAC system can then manage VE information resources and generate user authorizations, and monitor, control, and report progress against goals, schedule, and milestone of the VE.
2) Dissolution phase assesses successes or failure at the conclusion of the VE, and its results pave the experience for the next new VE. The PACP established at the design phase of a real VE must be modified to comply with the resource sharing rules after the VE dissolution.

VIII. Example of Practical VE Applying the VEAC Model

This section utilizes the automobile industry as an example to verify the feasibility of the proposed fundamental VEAC model and the PACP language model. Fig. 10 shows a new car engine R&D project (p_1) performed by VE ve$_1$. In Fig. 10, only parts of the projects are shown; some elements and assignments regarding the project and its three subprojects are shown in detail in the following tables. The engine R&D project has three subprojects: cylinder head design (p_{11}), cylinder block design (p_{12}), and crankshaft design (p_{13}). p_{11} is associated with p_{12} by using process and reference relations $(p_{11}R_{\rm pp}p_{12}$ and $p_{11}R_{\rm pr}p_{12})$; p_{12} is associated with p_{11} via process and reference relations $(p_{12}R_{\rm pp}p_{11}$ and $p_{12}R_{\rm pr}p_{11})$; p_{13} is exclusive to p_{11} and p_{12} via $p_{11}R_{\rm pe}p_{13}$ and $p_{12}R_{\rm pe}p_{13}$, and p_1 is exclusive to p_{12} via $p_1R_{\rm pe}p_{12}$. According to these definitions, p_{11} and p_{12} have stronger requirement for resource sharing, whereas p_{12} and p_{13} are independent. This example focuses on trust evaluation between the four projects and trust evaluation between three VERs (ve$_{11}$, ve$_{12}$, and ve$_{13}$) involved

TABLE I
VE$-P-A$ AND $P-$VER$-A$ LISTS

VE Name	Performed Project	Involved VERs	Objectives of the Project
ve_1: engine R&D	p_1	ver_{11}, ver_{12}	Designing a car engine (displacement: 2000cc., and horsepower >140 Hp)
ve_{11}: cylinder head design	p_{11}	ver_{111}, ver_{112}, ver_{113}	Designing the cylinder head of the engine
ve_{12}: cylinder block design	p_{12}	ver_{121}, ver_{122}, ver_{123}, ver_{124}	Designing the cylinder block and the timing gear cover of the engine
ve_{13}: crankshaft design	p_{13}	ver_{131}, ver_{132}, ver_{133}, ver_{134}, ver_{135}, ver_{136}	Designing the crankshaft and connecting rod of the engine

TABLE II
COMPANY LIST

Company No.	Company Name	Number of Employees	Company Address	Core Capacities
em_1	Company-A	100	Tainan Taiwan	Block, Internal Combustion Engine
em_2	Company-B	20	Taipei Taiwan	Cooling System
em_3	Company-C	200	Beijing China	Cylinder
em_4	Company-D	1200	Detroit USA	Cylinder
em_5	Company-E	5	TaichungT aiwan	Internal and External Combustion Engines
em_6	Company-F	13	Tokyo Japan	Main Bearing, Vibration Damper
em_7	Company-G	100	Shangha China	Flywheel, Crankshaft, Cam

in p_{11}; hence, some elements or assignments are ignored in the following tables.

Table I, the VE$-P-A$ and $P-$VER$-A$ lists, shows the VE name, project performed by the VE, the VERs involved in the VE, and the project objectives. For example, ve_1 involves two VERs, ver_{11} and ver_{12}, and performs project p_1 whose aim is to develop a 2000 cc car engine with at least 140 hp.

Table II lists the detailed information for each company participating in the four VEs.

Table III, the VE$-$EM$-A$ list, shows all EMs in each VE; for instance, the companies participating in ve_{11} are em_1, em_2, and em_3.

Table IV lists the attributes of FTs that are associated with the four projects, including the number, name, allowed reference, allowed subproject, and allowed main-project attributes.

Table V lists the $P-$FT$-A$ with project names, the number of FTs assigned to the projects, and the FTs involved in the projects.

Table VI lists the executed sequence of FTs involved in the two projects (p_{11} and p_{12}) between which a process relation is held. Consequently, when the event FT ft_{111} is achieved, the action FT ft_{121} is triggered. According to the process relation definition, ft_{121} will hierarchy all or part of the privileges assigned to ft_{111} when ft_{121} is executed.

Table VII, FT$-$PublicP$-A$, lists each FT and PublicPs assigned to each FT.

Table VIII shows the VER$-$FT$-A$ list, in which only VERs involved in ve_{11} are considered and listed.

In the aforementioned example, ve_{11} (cylinder head design) is used as an example to construct PACP for managing re-

TABLE III
VE$-$EM$-A$ LIST

VE Name	Enterprise Members
ve_1	$em_1, em_2, em_3, em_4, em_5, em_6, em_7$
ve_{11}	em_1, em_2, em_3
ve_{12}	em_3, em_4, em_5
ve_{13}	em_6, em_7

sources that belong to ve_{11}, as shown in the Appendix II. With the objective of cylinder head design of a new car engine, this VE consists of three EMs, i.e., Company-A (em_1), Company-B (em_2), and Company-C (em_3), responsible for oil filler cap design, cylinder head design, and stopper design, respectively.

In this PACP (see Appendix II), only part of the rules in the VEACP and part of the rules in the EACP of Company-A are listed. According to VEACP rule-ve_{11}-001, when two tasks ft_{111} and ft_{112} are being executed from May 20, 2007 to October 20, 2008, all Company-A, Company-B, and Company-C personnel may read knowledge of know-what about cylinder head design, car engine, and cylinder. The EACP rule-em_1-001 for Company-A dictates that, from November 20, 2007 to October 20, 2008, all Company-B and Company-C personnel may read R&D knowledge related to oil filler cap design.

IX. CONCLUSION AND FUTURE WORK

The results and contributions of this paper are as follows.

1) The formal VEAC model, including the fundamental VEAC model, PACP language model, and construction

TABLE IV
ATTRIBUTE LIST OF FTs

FT No.	FT Name	Attributes		
		Allowed-reference	Allowed-sub-project	Allowed-main-project
ft_{11}	Sub-project progress management	T	T	F
ft_{12}	Sub-project progress management	T	T	F
ft_{13}	Sub-project progress management	T	T	F
ft_{14}	Bulletin	T	T	T
ft_{111}	Oil filler cap design	T	T	F
ft_{112}	Cylinder head design	T	T	F
ft_{113}	Stopper design	T	T	F
ft_{121}	Cylinder liner design	T	T	F
ft_{122}	Cylinder head knock pin design	T	T	F
ft_{123}	Clutch housing design	T	T	F
ft_{124}	Engine rear bracket design	T	T	F
ft_{131}	Crankshaft design	F	F	F
ft_{132}	Crankshaft bearing upper metal design	F	F	F
ft_{133}	Lower oil ring design	F	F	F

TABLE V
$P-\mathrm{FT}-A$ LIST

Project Name	Number of FTs	Functional Tasks
p_1	4	$ft_{11}, ft_{12}, ft_{13}, ft_{14}$
p_{11}	4	$ft_{11}, ft_{111}, ft_{112}, ft_{113}$
p_{12}	5	$ft_{12}, ft_{121}, ft_{122}, ft_{123}, ft_{124}$
p_{13}	4	$ft_{13}, ft_{131}, ft_{132}, ft_{133}$

TABLE VI
SEQUENCE LIST

Event-Functional Task	Action-Functional Task
ft_{111}	ft_{121}
ft_{121}	ft_{122}
ft_{122}	ft_{112}
ft_{112}	ft_{123}

TABLE VII
FT$-$PublicP$-A$ LIST

FT	Public Permissions
ft_{11}	$public\text{-}p_1$
ft_{12}	$public\text{-}p_2$
ft_{13}	$public\text{-}p_3$
ft_{14}	$public\text{-}p_4$
ft_{111}	$public\text{-}p_5$
ft_{112}	$public\text{-}p_6$
ft_{113}	$public\text{-}p_7$
ft_{121}	$public\text{-}p_7, public\text{-}p_8$
ft_{122}	$public\text{-}p_6, public\text{-}p_8$
ft_{123}	$public\text{-}p_9$
ft_{124}	$public\text{-}p_{10}$
ft_{131}	$public\text{-}p_{11}, public\text{-}p_{13}$
ft_{132}	$public\text{-}p_{14}$
ft_{133}	$public\text{-}p_{15}$

TABLE VIII
VER$-$FT$-A$ LIST

VER	Performed Functional Tasks
ver_{111}	$ft_{11}, ft_{111}, ft_{112}$
ver_{112}	$ft_{11}, ft_{112}, ft_{113}$
ver_{113}	ft_{12}, ft_{112}

methodology, is proposed to facilitate VE resource management and sharing across organizations.

2) The fundamental VEAC model is designed to adapt to changes in VE members, both individuals and organizations, without affecting authorities of VERs, and eliminates the need to reset users' access authorities due to changes in cooperation targets.
3) Participation or withdrawal of an enterprise does not change the existing management model of resource access, thus significantly reducing administrative cost and complexity.

The results of this paper may help VEs solve the challenges of resource management and sharing among enterprises. Resource management and sharing will become increasingly complicated in the future owing to the requirement of strong information transparency. The proposed formal VEAC model solves AC and VE resource sharing challenges.

However, this paper has some deficiencies. For instance, the non-RBAC model, and integration of its access policies, has not been explored. An enterprise that adopts non-RBAC models and other access policies must perform additional model-transferring process to transform the models to RBAC to integrate them into the proposed PBAC model. This paper does not consider the possibility that the user might share a resource with unauthorized users, for example, by copying it, after legally acquiring the resource. The works in future are listed as follows.

1) An enterprise might adopt a non-RBAC-based scheme. Therefore, integrating different AC schemes or policies should be a focus for future works.
2) An enterprise should ideally retain its original AC model when joining a VE. Hence, a "plug-and-play" AC integration mechanism should be developed.
3) Because an enterprise might participate in several competing VEs, preventing the leaking of key technology or data should be considered.
4) Distributed security infrastructure including distributed heterogonous security architecture and collaborative VE policy management approaches should be completely designed for implementing the VEAC system.

APPENDIX I

TABLE IX
LIST OF FUNCTIONS RELATED TO THE VEAC MODEL

Function	Domain	Co-domain	Description
$VE\text{-}EM\text{-}A_{em}(ve)$	*VE*	2^{EM}	a *ve* to a set of *EM*s that participate in this *ve*
$VE\text{-}EM\text{-}A_{ve}(em)$	*EM*	2^{VE}	an *em* to a set of *VE*s that involve this *em*
$VE\text{-}P\text{-}A_{ve}(p)$	*P*	*VE*	a project *p* to a *VE* that performs this *p*
$VE\text{-}P\text{-}A_{p}(ve)$	*VE*	2^{P}	a VE *ve* to a set of *Project*s that are performed by this *ve*
$P\text{-}VER\text{-}A_{p}(ver)$	*VER*	*P*	a *ver* to a project *p* that involves this *ver*
$P\text{-}VER\text{-}A_{ver}(p)$	*P*	2^{VER}	a project *p* to a set of *VER*s that are assigned to this *p*
$P\text{-}FT\text{-}A_{ft}(p)$	*P*	2^{FT}	a project *p* to a set of *FT*s that are involved in this *p*
$P\text{-}FT\text{-}A_{p}(ft)$	*FT*	*P*	a *ft* to a project that involves this *ft*
$VER\text{-}FT\text{-}A_{ver}(ft)$	*FT*	2^{VER}	a *ft* to a set of *VER*s that perform this *ft*
$VER\text{-}FT\text{-}A_{ft}(ver)$	*VER*	2^{FT}	a *ver* to a set of *FT*s that are performed by this *ver*
$EM\text{-}U\text{-}A_{u}(em)$	*EM*	2^{U}	an *em* to a set of *U*s that are employees of this *em*
$EM\text{-}U\text{-}A_{em}(u)$	*U*	*EM*	a user *u* to an *EM* that involves this *u*
$NEM\text{-}U\text{-}A_{u}(nem)$	*NEM*	2^{U}	a *nem* to a set of *U*s that are employees of this *nem*
$NEM\text{-}U\text{-}A_{nem}(u)$	*U*	*NEM*	a user *u* to a *NEM* that involves this *u*
$EM\text{-}NEM\text{-}A_{em}(nem)$	*NEM*	2^{EM}	a *nem* to a set of *EM*s with tasks are supported by this *nem*
$EM\text{-}NEM\text{-}A_{nem}(em)$	*EM*	2^{NEM}	an *em* to a set of *NEM*s that support some tasks of this *em*
$R\text{-}VER\text{-}A_{r}(ver)$	*VER*	2^{R}	a *ver* to a set of *R*s that play this *ver*
$R\text{-}VER\text{-}A_{ver}(r)$	*R*	2^{VER}	a role *r* to a set of *VER*s that this *r* plays
$FT\text{-}PublicP\text{-}A_{public\_p}(ft)$	*FT*	$2^{PublicP}$	a *ft* to a set of *PublicP*s over all stages
$FT\text{-}PublicP\text{-}A_{ft}(public\_p)$	*PublicP*	2^{FT}	a *public_p* to a set of *FT*s over all stages
$FT\text{-}PrivateP\text{-}A_{private\_p}(ft)$	*FT*	$2^{PrivateP}$	a *ft* to a set of *PrivateP*s over all stages
$FT\text{-}PrivateP\text{-}A_{ft}(private\_p)$	*PrivateP*	2^{FT}	a *private_p* to a set of *FT*s over all stages
$FT\text{-}Permission\text{-}A(ft)$	*FT*	$2^{PublicP \quad PrivateP}$	a *ft* to a set of *Permission*s (including private and public permissions) over all stages
$FT\text{-}PublicP\text{-}A_{ft}(st)$	*Stage*	2^{FT}	a stage *st* to a set of *FT*s
$Stage_{public\_p}(st)$	*Stage*	$2^{PublicP}$	a stage *st* to a set of *Public-P*s, $Public\_Permission(st) \subseteq \{public\_p: (FT\text{-}PublicP\text{-}A_{ft}(st), public\_p) \in FT\text{-}PublicP\text{-}A\}$, which can change with *st*
$Stage_{ft}(st)$	*Stage*	2^{FT}	a stage *st* to a set of *FT*s, $Functional\_Task(st) \subseteq \{ft: (FT\text{-}PublicP\text{-}A_{ft}(st), public\_p) \in FT\text{-}PublicP\text{-}A\}$, which can alter with *st*
$Correspondence_{post}(ft)$	*FT*	*FT*	a pre-version FT *ft* to its post-version FT
$Correspondence_{pre}(ft)$	*FT*	*FT*	a post-version FT *ft* to its pre-version FT
$FTWf_{event}(ft)$	*FT*	2^{FT}	an action FT *ft* to a set of its event FTs
$FTWf_{action}(ft)$	*FT*	2^{FT}	an event FT *ft* to a set of its action FTs
$RH_{senior}(r)$	*R*	2^{R}	a role *r* to a set of *R*s, which are the senior roles of the *r*
$RH_{junior}(r)$	*R*	2^{R}	a role *r* to a set of *R*s, which are the junior roles of the *r*
$PR_{subset}(p)$	*P*	2^{P}	a project *p* to a set of *P*s with which the *p* has a subset relation
$PR_{version}(p)$	*P*	2^{P}	a project *p* to a set of *P*s with which the *p* has a version relation
$PR_{reference}(p)$	*P*	2^{P}	a project *p* to a set of *P*s with which the *p* has a reference relation
$PR_{process}(p)$	*P*	2^{P}	a project *p* to a set of *P*s with which the *p* has a process relation
$PR_{exclusive}(p)$	*P*	2^{P}	a project *p* to a set of *P*s with which the *p* has a exclusive relation
$CM_{cds}(ver)$	*VER*	2^{VER}	a *ver* to a set of *VER*s that cooperate with the *ver* by using dependent single-task mode
$CM_{cdm}(ver)$	*VER*	2^{VER}	a *ver* to a set of *VER*s that cooperate with the *ver* by using multi-task mode

Appendix II

TABLE X

Example of PACP for the Development of Cylinder Head of a Car Engine

```
<PACP Version= "version 1.1.1" Policy-no.= "N00233" Update-time= "5/15/2007" State= "active">
    <VEACPVE-name= "ve11" objective= "cylinder head design" Update-time= "5/15/2007" State= "active">
        <RuleSet>
            <RuleCombiningAlgorithm>permit-overrides</RuleCombiningAlgorithm>
            <Rule Type= "rule-kind" Name= "rule-ve11-001" Action= "active">
                <Target>
                    <SubjectSet>
                        <Subject>Company-A</Subject>
                        <Subject>Company-B</Subject>
                        <Subject>Company-C</Subject>
                    </SubjectSet>
                    <ActionSet>
                        <Action>read</Action>
                    </ActionSet>
                    <ObjectSet>
                        <Object>know-what to cylinder head design</Object>
                        <Object>know-what to car engine</Object>
                        <Object>know-what to cylinder</Object>
                    </ObjectSet>
                    <Environment> date>=5/20/2007 and date<=10/20/2008 </Environment>
                </Target>
                <Condition>ft111(oil filler cap design) and ft112 (cylinder head design) are being
                          executed</Condition>
                <Effect>permit</Effect>
            </Rule>
            <Rule Type= "rule-kind" Name= "rule-ve11-002" Action= "active">
                <Target>
                    <SubjectSet>
                        <Subject>Company-A</Subject>
                        <Subject>Company-B</Subject>
                        <Subject>Company-C</Subject>
                    </SubjectSet>
                    <ActionSet>
                        <Action>write</Action>
                        <Action>read</Action>
                    </ActionSet>
                    <ObjectSet>
                        <Object>all resources assigned to ft11</Object>
                    </ObjectSet>
                    <Environment> date>=5/20/2007 and date<=10/20/2008</Environment>
                </Target>
                <Condition> anyone of ft111(oil filler cap design), ft112 (cylinder head design) and ft113 (stopper
                          design) are being executed</Condition>
                <Effect>permit</Effect>
            </Rule>
        </RuleSet>
    </VEACP>

    <EACP Company-ID= "em1" Company-name= "Company-A" Responsibility= "oil filler cap design"
        Update-time= "5/16/2007" State= "active">
        <RuleSet>
            <RuleCombiningAlgorithm>permit-overrides</RuleCombiningAlgorithm>
            <Rule Type= "rule-kind" Name= "rule-em1-001" Action= "active">
                <Target>
                    <SubjectSet>
                        <Subject>Company-B</Subject>
                        <Subject>Company-C</Subject>
                    </SubjectSet>
                    <ActionSet>
                        <Action>read</Action>
                    </ActionSet>
                    <ObjectSet>
                        <Object>R&D knowledge related to oil filler cap design</Object>
                    </ObjectSet>
                    <Environment> date>=11/20/2007 and date<=10/20/2008</Environment>
                </Target>
                <Condition>f111 is completed</Condition>
                <Effect>permit</Effect>
            </Rule>
        </RuleSet>
    </EACP>
    . . .

</PACP>
```

References

[1] Y.-M. Chen and M.-W. Liang, "Design and implementation of a collaborative engineering information system for allied concurrent engineering," *Int. J. Comput. Integr. Manuf.*, vol. 13, no. 1, pp. 11–30, Jan. 2000.

[2] A. Frenkel, H. Afsarmanesh, C. Garita, and L. O. Hertzberger, "Supporting information access rights and visibility levels in virtual enterprises," in *Proc. 2nd IFIP Work. Conf. Infrastructure Virtual Enterprise*, 2000, pp. 177–192.

[3] J. Ma and M. A. Orgun, "Trust management and trust theory revision," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 3, pp. 451–460, May 2006.

[4] Y. Lu, W. Wang, B. Bhargava, and D. Xu, "Trust-based privacy preservation for peer-to-peer data sharing," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 3, pp. 498–502, May 2006.

[5] E. Turban, D. King, D. Viehland, and J. Lee, *Electronic Commerce: A Managerial Perspective*. Upper Saddle River, NJ: Pearson Educ. Int., 2006.

[6] H. R. Rao and S. J. Upadhyaya, "Special issue on secure knowledge management," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 35, no. 1, p. 185, Jan. 2005.

[7] E. Bertino, L. R. Khan, R. Sandhu, and B. Thuraisingham, "Secure knowledge management: Confidentiality, trust, and privacy," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 3, pp. 429–438, May 2006.

[8] R. Singh and A. F. Salam, "Semantic information assurance for secure distributed knowledge management: A business process perspective," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 3, pp. 472–486, May 2006.

[9] R. Au, M. Looi, and P. Ashley, "Automated cross-organizational trust establishment on extranets," in *Proc. Workshop Inf. Technol. Virtual Enterprises*, 2001, pp. 3–11.

[10] T.-Y. Chen, Y.-M. Chen, C.-B. Wang, and H.-C. Chu, "Development of an access control model, system architecture and approaches for information sharing in virtual enterprise," *Comput. Ind.*, vol. 58, no. 1, pp. 57–73, Jan. 2007.

[11] T.-Y. Chen, Y.-M. Chen, H.-C. Chu, C.-B. Wang, and H. Yang, "Secure resource sharing on cross-organization collaboration using a novel trust method," *Robot. Comput.-Integr. Manuf.*, vol. 23, no. 4, pp. 421–435, Aug. 2007.

[12] M. Koch, L. V. Mancini, and F. Parisi-Presicce, *Graph Transformations for the Specification of Access Control Policies*. Amsterdam, The Netherlands: Elsevier science B. V, 2002.

[13] D. F. Ferraiolo, D. R. Kuhn, and R. Chandramouli, *Role-Based Access Control*. Norwood, MA: Artech House, 2003.

[14] S. Oh and S. Park, "Task-role-based access control model," *Inf. Syst.*, vol. 28, no. 6, pp. 533–562, Sep. 2003.

[15] A. Kern, "Advanced features for enterprise-wide role-based access control," in *Proc. Comput. Security Appl. Conf.*, 2002, pp. 333–342.

[16] C. J. Moon, D. H. Park, S. J. Park, and D. K. Baik, "Symmetric RBAC model that takes the separation of duty and role hierarchies into consideration," *Comput. Security*, vol. 23, no. 2, pp. 126–136, Mar. 2004.

[17] D. Shin, G. J. Ahn, and J. S. Park, "An application of directory service markup language (DSML) for role-based access control (RBAC)," in *Proc. Comput. Softw. Appl. Conf.*, 2002, pp. 934–939.

[18] K. Furst, T. Schmidt, and G. Wippel, "Managing access in extended enterprise networks," *IEEE Internet Comput.*, vol. 6, no. 5, pp. 67–74, Sep./Oct. 2002.

[19] J. Bacon, K. Moody, and W. Yao, "A model of OASIS role-based access control and its support for active security," *ACM Trans. Inf. Syst. Security*, vol. 5, no. 4, pp. 492–540, Nov. 2002.

[20] F. T. Alotaiby and J. X. Chen, "A model for team-based access control (TMAC)," in *Proc. Inf. Technol.: Coding Comput.*, 2004, vol. 1, pp. 450–454.

[21] J. J. Kanet, W. Faisst, and P. Mertens, "Application of information technology to a virtual enterprise broker: The case of Bill Epstein," *Int. J. Prod. Econ.*, vol. 62, no. 1, pp. 23–32, May 1999.

[22] E. K. Ouzounis, "An agent-based platform for the management of dynamic virtual enterprises," Ph.D. dissertation, Tech. Univ. Berlin, Berlin, Germany, 2001.

[23] J. S. Park and J. Hwang, "RBAC for collaborative environments: Role-based access control for collaborative enterprise in peer-to-peer computing environments," in *Proc. 8th ACM Symp. Access Control Models Technol.*, 2003, pp. 93–99.

[24] N. Mezzetti, "Towards a model for trust relationships in virtual enterprises," in *Proc. 14th Int. Workshop Database Expert Syst. Appl.*, 2003, pp. 420–424.

[25] T. J. Smith and L. Ramakrishnan, "Joint policy management and auditing in virtual organizations," in *Proc. 4th Int. Workshop Grid Comput.*, 2003, pp. 117–124.

[26] G. Steinke and R. Leamon, "Information security issues facing virtual enterprises," in *Proc. Int. Conf. Eng. Technol. Manage.*, 1996, pp. 641–644.

[27] H. Zhu, "Some issues of role-based collaboration," in *Proc. Can. Conf. Elect. Comput. Eng.*, 2003, vol. 2, pp. 687–690.

[28] G. Kolaczek, "Specification and verification of constraints in role based access control," in *Proc. 12th IEEE Int. Workshops Enabling Technol.: Infrastructure Collaborative Enterprise*, 2003, pp. 190–195.

[29] J. Luo and D. He, "Research on object-oriented role-based access control model," in *Proc. 4th Int. Conf. Parallel Distrib. Comput., Appl. Technol.*, 2003, pp. 132–135.

[30] J. D. Moffett, "Control principles and role hierarchies," in *Proc. 3rd ACM Workshop Role-Based Access Control*, 1998, pp. 63–69.

[31] S. Osborn, "Integrating role graphs: A tool for security integration," *Data Knowl. Eng.*, vol. 43, no. 3, pp. 317–333, Dec. 2002.

[32] M. Lorch, S. Proctor, R. Lepro, D. Kafura, and S. Shah, "First experiences using XACML for access control in distributed systems," in *Proc. ACM Workshop XML Security*, 2003, pp. 25–37.

[33] S. Barker and P. J. Stuckey, "Flexible access control policy specification with constraint logic programming," *ACM Trans. Inf. Syst. Security*, vol. 6, no. 4, pp. 501–546, Nov. 2003.

[34] M. Coetzee and J. H. P. Eloff, "Virtual enterprise access control requirements," in *Proc. Annu. Res. Conf. South Afr. Inst. Comput. Scientists Inf. Technologists Enablement Through Technol.*, 2003, pp. 285–294.

[35] S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian, "Flexible support for multiple access control policies," *ACM Trans. Database Syst.*, vol. 26, no. 2, pp. 214–260, Jun. 2001.

[36] A. Belokosztolszki and K. Moody, "Meta-policies for distributed role-based access control systems," in *Proc. 3rd Int. Workshop Policies Distrib. Syst. Netw.*, 2002, pp. 106–115.

[37] S. Hada and M. Kudo, "XML document security based on provisional authorization," in *Proc. 7th ACM Conf. Comput. Commun. Security*, 2000, pp. 87–96.

[38] G. Boella and L. van der Torre, "Security policies for sharing knowledge in virtual communities," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 3, pp. 439–450, May 2006.

[39] A. Kern, A. Schaad, and J. Moffett, "Enterprise role administration: An administration concept for the enterprise role-based access control model," in *Proc. 8th ACM Symp. Access Control Models Technol.*, 2003, pp. 3–11.

[40] R. A. Botha and J. H. P. Eloff, "Designing role hierarchies for access control in workflow systems," in *Proc. 25th Annu. Int. Comput. Softw. Appl. Conf.*, 2001, pp. 117–122.

[41] F. Dridi, B. Muschall, and G. Pernul, "Administration of an RBAC system," in *Proc. 37th Annu. Hawaii Int. Conf. Syst. Sci.*, 2004, pp. 187–192.

Tsung-Yi Chen received the B.S. degree from Providence University, Taichung, Taiwan, R.O.C., in 1996, and the M.S. and Ph.D. degrees from the Institute of Manufacturing Engineering, National Cheng Kung University, Tainan, Taiwan, in 2001 and 2006, respectively.

He is currently an Assistant Professor with the Department of Electronic Commerce Management, Nanhua University, Chia-Yi, Taiwan. His research interests include virtual enterprise, e-commerce and knowledge commerce, enterprise and information integration, access control, and knowledge sharing.

Yuh-Min Chen received the B.S. and M.S. degrees from National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1981 and 1983, respectively, and the Ph.D. degree in industrial and systems engineering from Ohio State University, Columbus, in 1991.

He is currently a Professor and the Director of the Institute of Manufacturing Engineering, National Cheng Kung University, Tainan, Taiwan. Before joining the faculty of the Institute of Manufacturing Engineering in 1994, he was a Research Engineer with the Structural Dynamics Research Corporation, Plano, TX, for three years. His current research interests include enterprise integration, engineering data and knowledge management, computer-aided concurrent engineering, and manufacturing information systems.

Chin-Bin Wang received the B.S. degree from National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1981, the M.S. degree from the University of Southern California, Los Angeles, in 1985, and the Ph.D. degree in computer science from the City University of New York, New York, in 1995.

He is currently a Professor and the Chairman of the Department of Electronic Commerce Management, Nanhua University, Chia-Yi, Taiwan. His research interests include data mining, network management, engineering data and knowledge management, and system integration.

A Formal Virtual Enterprise Access Control Model

Tsung-Yi Chen, Yuh-Min Chen, and Chin-Bin Wang

***Abstract*—A virtual enterprise (VE) refers to a cooperative alliance of legally independent enterprises, institutions, or single persons that collaborate with each other by sharing business processes and resources across enterprises in order to raise enterprise competitiveness and reduce production costs. Successful VEs require complete information transparency and suitable resource sharing among coworkers across enterprises. Hence, this investigation proposes a formal flexible integration solution, named the formal VE access control (VEAC) model, based on the role-based AC model, to integrate and share distributed resources owned by VE members. The formal VEAC model comprises a fundamental VEAC model, a project AC policy (PACP) language model, and a model construction methodology. The fundamental VEAC model manages VE resources and the resources of participating enterprises, in which various project relationships are presented to facilitate different degrees of resource sharing across projects and enterprise boundaries, and cooperative modes among VE roles are presented to enable collaboration among coworkers in a VE. This PACP language model features object–subject–action–condition AC policies that jointly determine user access authorizations. In addition, the methodology supplies a systematic method to identify fundamental elements of the VEAC model and to establish assignments between elements and relations.**

***Index Terms*—Access control (AC), resource sharing, role-based access control (RBAC), virtual enterprise (VE).**

I. Introduction

Virtual enterprise (VE) is regarded as one of the most promising business strategies to enhance the global competitiveness of enterprises [1]. VEs integrate the processes, activities, and resources from different enterprises through enterprise alliances to respond quickly to customer expectations. Frenkel *et al.* [2] defined a VE as a collaborative group of existing autonomous enterprises, which selectively share their expertise, skills, and resources to accomplish a common product or service. In practice, a VE is generally implemented with a distributed and collaborative business process, in which individuals from different enterprises cooperate on business-related activities or processes by remote coordination, communication, and control [1]. To attain VE goals and support each other's functionalities, enterprises in a VE must share and exchange information, knowledge, and resources. The features that determine the access level to the local information of every enterprise, when considering the competitive and cooperative relationships among enterprises, include the degree of trust between two enterprises, the function of the enterprises in the VE, and contractual agreements [2], [3]. Lu *et al.* [4] proposed a trust-based privacy preservation method for P2P data sharing.

A collaborative engineering environment allows multiple engineers to work simultaneously with individual assembly parts. Some manufacturing industries, e.g., the automotive sector, use VEs to maintain business relationships with their suppliers and corporate customers, enabling manufacturers to collaborate on the design, production, assembly, and marketing of new products. For instance, designing and developing a new car is a complex and lengthy process; during product R&D, engineering and design drawings can be shared over secure network among the contracting firm, testing facility, marketing firm, and downstream manufacturing and service companies [5]. Information concerning the design for a new product at various segments of the VE has to be visible to all members of the VE at any time. Consequently, the information must be managed properly, with appropriate access control (AC) models, strict policies, discipline, and daily monitoring. Development of a new car model by a VE might involve approximately 20 000 designers and engineers from hundreds of divisions and departments, some of which belong to different enterprises in different countries. One sub-VE in the car-manufacturing VE performs car design, which contains four subprojects, namely, engine design, cool system design, transmission case design, and framework design. Engine designers in the engine design subproject collaboratively develop an engine for the new car model. Information related to the engine, such as drawing and engineering data, is generated and shared in real time to workers in the subproject and other subprojects. Therefore, the success of a VE depends wholly on transparent and effective sharing of information resources, including information, application systems, and knowledge, throughout the business cycle [1]. Not all business partners are equally trusted in today's complex business environment. Today's partners could become tomorrow's competitors. Hence, enterprises do not generally like sharing information. Consequently, a VE or related business strategy, such as allied concurrent engineering or virtual team, is likely to fail. Therefore, VE urgently needs secure and trustworthy AC model, approach, and mechanism that can manage distributed resources across enterprises and share them with collaborative workers. To secure information sharing, competitive and cooperative relationships among enterprises should be considered when using the proposed model to evaluate a user's authorization to access resources.

Secure resource management and sharing across organizational boundaries have seldom been addressed. AC for VEs is difficult for the following reasons: 1) enterprise members in a VE may change frequently; 2) each enterprise member

Manuscript received August 6, 2006; revised May 18, 2007. This work was supported in part by the National Science Council of the Republic of China, Taiwan, under Contract NSC96-2221-E-343-002. This paper was recommended by Associate Editor J. Miller.

T.-Y. Chen and C.-B. Wang are with the Department of Electronic Commerce Management, Nanhua University, Chia-Yi 62248, Taiwan, R.O.C. (e-mail: tsungyi@mail.nhu.edu.tw; cbwang@mail.nhu.edu.tw).

Y.-M. Chen is with the Institute of Manufacturing Engineering, National Cheng Kung University, Tainan 70101, Taiwan, R.O.C. (e-mail: ymchen@mail.ncku.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCA.2008.923090

(EM) typically has many roles and users; 3) a VE often has many EMs with complicated interrelationships—for example, members may cooperate and also compete with one another; 4) organizations within a VE may be dynamic and perform unpredictable activities; and 5) VE resources may be Internet based, distributed, and heterogeneous. Few studies have explored control of access to knowledge, which is one of the most important assets for an enterprise. Therefore, developing an AC mechanism for knowledge protection has been recognized as a vital research topic in knowledge management [6]–[8]. Although role-based models have been adopted successfully for resource management within an enterprise, collaborative VE systems using role authorization management approaches have not been widely investigated. In contrast to conventional AC models, AC for a VE does not specifically assign rights to each role or user in advance because of the dynamic characteristics of VE organizations, such as flexibility and mobile resource sharing. To our knowledge, no studies have developed models for resource sharing management that support collaborative and cooperative business activities across organizational boundaries. Before achieving secure resource sharing in a VE that increases corporate global competitiveness, several requirements for trust management, such as scalability, flexibility, dynamic security, decentralization, and mutual trust, must be addressed [9]. Hence, VEs require an appropriate AC model.

Based on the conceptual AC model in VEs [10], [11], this investigation develops a formal VEAC model to solve the problem of authorization management and to secure AC among organizations within a VE. The formal VEAC model comprises a fundamental VEAC model, a project AC policy (PACP) language model, and a model construction methodology. The proposed fundamental VEAC model comprises a [project-based access control (PBAC)] model for managing public resources within VE and a role-based AC (RBAC) model for managing the sharing of an individual enterprise's private resources with VE members. Public resources are generated, used, modified, and owned by VE activities and are stored or implemented in a VE or its partners. And, private resources are owned by partners and shared with other workers who could be from different partners. This PACP language model features object–subject–action–condition AC policies that jointly determine user access authorizations. Moreover, the methodology supplies a systematical method to identify fundamental elements of the VEAC model and establish assignments between elements and relations. The proposed formal VEAC model provides VE workers with efficient management and easy access to relevant resources and up-to-date information, thus eliminating information delay and enhancing information transparency.

II. Related Works

AC systems and technologies are required to protect such resources and information from illegal access. This section surveys a number of studies related to the aims of this paper, including AC, VE, and AC policy.

A. AC

AC protects the computing system against unauthorized access or modification of information resources [12]. AC determines whether a user has rights to use a given resource; an AC system governs when and how resources can be used by whom. So far, many AC methods had been presented.

Early AC methods for resource management include AC lists (ACLs) and the AC matrix (ACM). A simple ACM is an array containing one row per subject in the system and one column per object. Entries in the matrix specify the operation or access each subject has to each object [13]. These methods are straightforward, intuitive, and only useful for small organizations [14]. ACLs implement the ACM by representing the columns as lists of users attached to a protected object. Each object is associated with an ACL that stores all subjects and the subject's approved operations for a given object. Most AC models, including mandatory AC, discretionary AC, RBAC, task-based AC, and task RBAC [15]–[17], only consider authorization management within a single organization. Furst *et al.* [18] investigated distributed RBAC to delegate administration of resources to individual departments within an enterprise. In RBAC, users are assigned roles that are associated with approved permissions for performing an operation on an enterprise resource (object) [19]. Team-based AC 2004, derived from RBAC, enables users to join team roles within an organization [20].

B. VEs

A VE is defined as a cooperative alliance in which a group of legally independent enterprises, institutions, and individuals cooperate for a particular goal [21]. Ouzounis [22] defined VE as a network of different administrative business domains that cooperate by sharing business processes and resources to provide a value-added service to customers. VE environments (Fig. 1) contain users (subjects/workers) from various enterprises, such as EMs, partners, suppliers, customers, and other VEs. VE-related activities are undertaken by users from different enterprises using collaboration and concurrence. Such a business environment results in complex AC problems. In particular, all VE resources that may be stored on and owned by different enterprises should be managed fully and should be shared as much as possible.

1) Characteristics of VEs: Kanet *et al.* [21] decomposed the life cycle of a VE into five phases, namely, identification, formation, design, operation, and dissolution. Ouzounis [22] found that the life cycle of a VE should include two major phases: establishment and management. Based on the analysis of life cycle and interactions, a VE has the following characteristics [23]–[25].

1) A VE may consist of several distributed VEs or enterprises.
2) A VE's participating members and business processes may be changed during its life cycle.
3) A VE emphasizes professional division and dynamic cooperation among a highly heterogeneous membership.
4) A VE conducts business processes of different stages across enterprises, in which each stage has its own participants, resources, and aims.
5) Various resources in a VE are shared and distributed over all participating enterprises and used by their employees (users).
6) A VE globally specifies members' obligations, responsibilities, and roles.

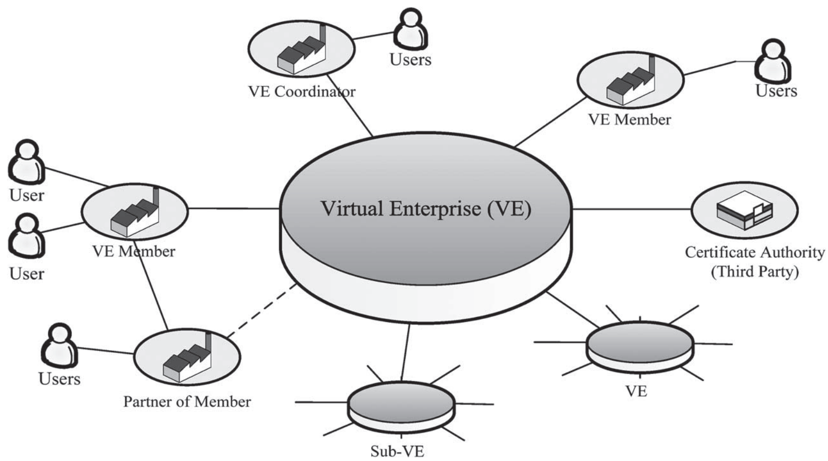


Fig. 1. VE environment.

7) A change in a member's role in a process should not affect the obligations and responsibilities in its other assigned roles.
8) Regulations do not constrain the selection of members in participating enterprises' partners.
9) Each member may own its enterprise resource management policy and AC model.
10) Shared VE resources include private resources owned by a participating enterprise and stored in its own repositories and also public resources belonging to the VE and stored in a public repository.

The levels of resource sharing among partners depend on VE characteristics, including levels of cooperation with partners, degree of trust, distributed tasks, and contractual agreements. When each participant in a VE brings information to the VE, the participant will not want to share more proprietary information than necessary with VE members because of information security issues. Information in VEs can be divided into three areas: 1) information of an individual partner brought to the VE; 2) information generated by the VE; and 3) information assets of the VE [26]. The information must be protected and distributed in a secure manner among all participants.

2) Requirement Analysis for AC in VE: Based on the general requirements of AC expressed in [27] and [28], this paper identifies the following requirements for AC model design: 1) Only the security administrator should be permitted to modify security attributes; 2) roles should be able to inherit authority either fully or partially; 3) positive authorizations and negative authorizations, as well as the principle of strict least privilege, should be supported; 4) the fine-grained authority requirements should be fulfilled; 5) access authority may change with tasks or roles; and 6) the model should be able to manage all users and resource objects in the enterprise [29]–[31].

Aside from the aforementioned requirements, according to the characteristics of VE, additional requirements must be considered when developing a VEAC model, as follows.

1) Since the organization structure of a VE is dynamic, access rights and resource objects can be changed in real time.
2) The model considers all users' access rights because resource administrators cannot predict who will access which resources in a VE.
3) As a VE is formed to achieve a certain goal in a limited time frame, each VE has different goal and business processes. A VE is always conducted as a project. Therefore, project is an essential element of AC in VE.
4) Since each enterprise has a legacy AC system, the VEAC model should be easily integrated with various AC models or policies.
5) The VE manages and shares resources collaboratively.
6) To facilitate trust among enterprises, the access policy in VE is planned and managed together by administrators of all participating enterprises.
7) The VE can maintain the consistency of policies and manage the conflicts between VE access policy and members' own access policies.

C. AC Policy

A significant shortcoming of existing AC systems is that they were developed by using a specific AC policy, which was defined by Lorch *et al.* [32], regarding how services can be utilized. AC policies are typically represented as follows: 1) constrained logic programs that support specific policy options; 2) constrained checks; and 3) administrator queries [33]. AC policy can restrict the use of services to suitably qualified principals and specify constraints that must hold when a service is invoked [19].

Recent development of AC policy framework includes languages and graphical approaches that specify different AC policies in a single framework [34]. A graph transformation-based security policy framework was proposed by Koch *et al.* [12] that included negative and positive constraints. The negative constraints specify graphs not contained in any system graph, and positive constraints specify graphs explicitly constructed in a system graph. By combining a formal framework and a logic-based language, Jajodia *et al.* [35] developed the authentication specification language that can be used to identify different AC policies that can coexist within the same system and be enforced by the same security server. Moreover, security assertion markup language is an XML framework identified by OASIS security services to exchange authentication and authorization information. For AC across enterprises, Belokosztolszki and Moody [36] proposed metapolicies. Hada and Kudo [37] proposed XML AC Language, an XML-based language for

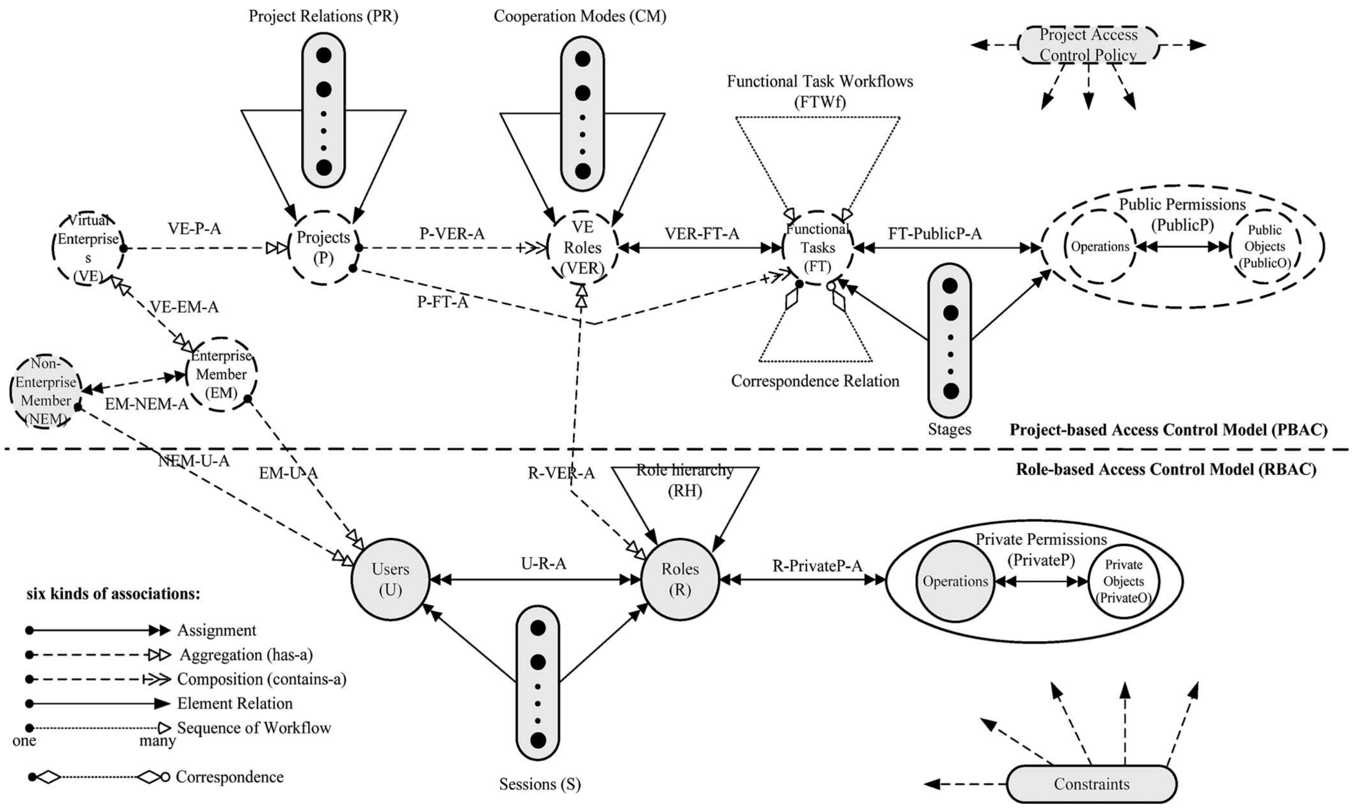


Fig. 2. Fundamental VEAC model.

provisional authorization, which articulates the security policies to be enforced for specific access to XML documents and provides XML with a sophisticated AC mechanism that enables an initiator to securely browse XML documents and securely update each document. Boella and van der Torre [38] studied normative multiagent systems for secure knowledge management based on AC policies.

III. Fundamental VEAC Model

This section introduces the proposed fundamental VEAC model, and its basic elements (Fig. 2), which has been derived from the requirements of AC for VE and characteristics of VE. It includes two submodels: one PBAC model for managing public resources stored on VE and one RBAC model for managing private resources stored on individual EMs [10]. In the model, the solid-line and the dashed-line circles are used to represent the elements in the RBAC and PBAC models, respectively. The six kinds of associations are proposed to indicate the various relationships among elements. Assignment is a well-known relationship in RBAC to continue using in the model. Aggregation is a grouping of other elements, which is also called a has-a association. For example, a VE is a group of EMs. Composition is an inclusion of other elements, which is also called a contains-a association. If the containing element is destroyed, the elements that it contains are also destroyed. Element relation is an interacting mode of other independent elements, which is further decomposed into various relations to facilitate resource sharing. Sequence of workflow is the order in which elements follow one another. Correspondence is a version mapping relation of a functional task (FT) in other project.

A. *RBAC Model*

This paper slightly adjusts the basic RBAC model [39]–[41] and seamlessly integrates it with the PBAC model. In the adjusted RBAC model, as shown in the bottom layer of Fig. 2, each element is described straightforwardly as follows.

1) User (U) represents a human or agent in an enterprise, which includes direct users, indirect users, and nonmember users.
2) Role (R) represents a functional job or responsibility.
3) Private object (PrivateO) denotes a resource in an enterprise associated with private privileges. Private objects are generally classified into three levels, which are public, proprietary, and protection. The public classification can be provided to the partners in a VE.
4) Private permission (PrivateP) is an approval of a particular mode of access to one or more private objects.
5) Session (S) maps a user to one or more roles.
6) $U{-}R{-}A \subseteq U \times R$ represents a many-to-many user to role assignment relation.
7) $R{-}\text{PrivateP}{-}A \subseteq R \times \text{PrivateP}$ represents a many-to-many role to PrivateP assignment relation.
8) $R_{\text{re}} = \{(x, y) : x, y \in R, x \neq y, \text{ and } x \text{ conflicts with } y\}$ signifies that role x conflicts with role y, and x and y cannot be both assigned to the same user.
9) $R_{\text{rh}} = \{(x, y) : x, y \in R, x \neq y, \text{ and } x \text{ is a superior of } y\}$ indicates that role x is a senior to role y, and x inherits the PrivatePs of y.
10) $U{-}R{-}A_u(r) : R \rightarrow 2^U$, a function mapping a role r to a set of users that can play this role.
11) $U{-}R{-}A_r(u)$: $U \rightarrow 2^R$, a function mapping a user u to a set of roles that can be played by this user.

12) R−PrivateP−A_r(private_p) : PrivateP $\rightarrow 2^R$, a function mapping a PrivateP, private_p, to a set of roles that is authorized to access this PrivateP.
13) R−PrivateP−$A_{\text{private\_p}}(r) : R \rightarrow 2^{\text{PrivateP}}$, a function mapping a role r to a set of PrivatePs that allows to be accessed by this role.

B. PBAC Model

The top portion of Fig. 2 shows the PBAC model. The core concept of model development, elements, and relations in the PBAC model are introduced and defined in the following sections in order.

1) Core Concept of the PBAC Model: A VE can perform several projects (P) simultaneously, but a project can only be performed by one VE. A project includes management-level and operational-level tasks. The management-level tasks control and manage the project's progress and output according to the project timestamp, whereas the operational level comprises FTs supervised and controlled by the project schedule. Different project relations (PRs), such as subset, exclusion, and reference, exist among projects to facilitate resource sharing (refer to Section IV). Activities within a project can be divided into several FTs, each of which has access to certain public objects (PublicOs), which is public permission (PublicP) of the FT. FTs involved in a project are constructed for performing VE activities in the VE formation stage. The FTs are assigned to VE roles (VERs) that are virtual roles created based on division of labors. It is required to meet certain conditions to start or end an FT. According to the goal and task requirements, an FT can be divided into different stages by timestamp or FT. Users are given different privileges depending on the project stage and FTs. A VE is composed of several real EMs, each of which can participate in more than one VE. Non-EMs (NEMs) are enterprises that do not participate directly in the activities of VE but participate in the activities of an EM which performs directly the activities of the VE. All VE participants, including three user types (direct, indirect, and nonmember users), are generally called users (U) which may play a different role (R) in a different session. Each role has access to private resources, called a PrivateP. A superior role can inherit the privileges of inferior roles through role hierarchy (RH). The EM plays a VER through a user or role to obtain the privilege of sharing public resources in the VE and carry out practically the obligations of a given VER and to achieve the VE goals. PACP is designed to identify the resource sharing rules in a project. Through constructing relations among projects and a PACP, users can share resources among projects. The rules of sharing can be modified at any time.

To simplify the complex assignment and facilitate resource sharing across domains, some relations are gained by exploring the three viewpoints of project, VE, and enterprise. From the project viewpoint, PRs including subset, version, reference, process, and exclusive relations (defined in Section IV) are found out depending on the features of project, facilitating sharing among projects. From the VE viewpoint, cooperative relations including dependent single-task, dependent multitask, and independence (defined in Section V) are found out depending on the information requirements of interaction and cooperation among workers in VE, facilitating sharing among enterprises involved in a VE. From the enterprise viewpoint, relations proposed by RBAC [39], [40], including role hierarchy, static separation of duty, and dynamic separation of duty, are used herein to facilitate sharing among roles in an enterprise.

2) Fundamental Elements: This section concisely introduces the fundamental elements of the PBAC model, each of which is represented as follows.

1) VE = {ve: ve represents a dynamic Internet organization consisting of EMs executing a project to achieve one common business goal}.
2) EM = {em: em can be a substantive enterprise organization, a VE, or an individual, and it is a VE member with at least one worker participating directly in the VE activities}.
3) NEM = {nem: nem can be a substantive enterprise organization, a VE, or an individual, but it is not a VE member; a nem has at least one worker participating directly in the activities of EMs, and the activities have direct relations with the FT of the VE}.
4) Project $(P) = \{p$: p denotes the set of FTs, projects, and subprojects performed by a VE$\}$.
5) FT = {ft: ft is a set of VE activities, which have a common objective and are undertaken by several VERs}.
6) VER = {ver: ver represents a virtual role formed to enable professional division within VE, which is assigned to perform more than one FT}.
7) Object $(O) = \{o$: o denotes an information resource including public and private resources which can be a database, entity, attribute, tuple, document, XML document, application, software component, or knowledge$\}$.
8) PublicO = {public-o: public-o represents a subset of objects owned by a VE, stored in a VE's repository, and implemented in a VE's platform}.
9) Operation = {op: op is a set of access authorities, such as write, read, and execute}.
10) PublicP = {public-p: public-p represents a permitted mode of access to a PublicO}.
11) Permission = $\{x$: $x \in$ PublicP $\cup$ PrivateP$\}$.
12) PACP: PACP identifies which project resources are protected and shared according to the relations among projects and the shared rules and which activities are forbidden in the VE scope. Each project involves a PACP, which can be performed automatically by the VEAC system. The PACP can be dynamically created, enforced, and modified by administrators when the VE environment changes. The main rules described in PACP include the following: 1) rules of resource sharing among projects, describing the resource sharing strategy and relations among projects; 2) rules of resource usage in a project, including constraints on VERs, FTs, PublicPs, and assignments between elements; 3) rules of resource sharing of various cooperation modes, identifying the level of resource sharing according to the cooperation mode between VERs; and 4) rules of exception handling, which can be classified into rules of permitted exception handling and rules of forbidden exception handling. A PACP language model used to construct the PACP is shown in detail in Section VI.

3) Assignments and Relations: The following sections define the concept of assignments and relations between two elements involved in the model based on the concept of a product set (refer to Definitions 1 and 2). Some functions relating to all elements in the VEAC model are defined and then applied to the following sections. These functions are shown in Appendix I.

Definition 1: Given two sets A and B, the product set or Cartesian product of A and B, called the assignment of A and B in AC domain, is $A \times B = \{(a, b) : a \in A, \text{ and } b \in B\}$.

Definition 2: Given sets A and B, a binary relation R from A to B is a subset of $A \times B$, i.e., $R \subseteq A \times B$.

4) Foundational Assignments: According to Definitions 1 and 2, the various assignment relations among elements are defined as follows.

1) FT$-S-$Public$-A \subseteq$ FT $\times\, S \times$ Public$-A$, triple assignment among three elements: FT, S, and PubicP, FT$-S-$Public$-A$ represents the set $R_{\text{ft-s-publicp-a}} = \{$(ft, st, public-p): ft $\in$ FT, st$\in$ Stage, public-p$\in$ PublicP, the public-p is assigned to ft in stage $s\}$.
2) $P-$VER$-A \subseteq P \times$ VER, one-to-many P to VER assignment, is denoted by $R_{\text{p-ver-a}} = \{(p, \text{ ver})$:$p \in P$, ver $\in$ VER, and p involves ver$\}$. The relation describes which VERs are included in project p.
3) VER$-$FT$-A \subseteq$ VER $\times$ FT, a many-to-many VER to FT assignment, is represented by $R_{\text{ver-ft-a}} = \{$(ver, ft): ver $\in$ VER, ft $\in$ FT, and ver performs ft$\}$. This relation describes which FTs are undertaken by which VERs.
4) VE$-$EM$-A \subseteq VE \times$ EM, a many-to-many VE to EM assignment, is denoted by $R_{\text{ve-em-a}} = \{$(ve, em): ve $\in$ VE, em $\in$ EM, and em is a member of ve$\}$.
5) VE$-P-A \subseteq$ VE $\times\, P$, one-to-many binary assignment from a VE to P, is represented by $R_{\text{ve-p-a}} = \{$(ve, p): ve $\in$ VE, $p \in P$, and ve performs $p\}$. This relation records which project is performed by a VE.
6) EM$-$NEM$-A \subseteq$ EM $\times$ NEM, many-to-many EM to NEM assignment, is represented by $R_{\text{em-nem-a}} = \{$(em, nem): em $\in$ EM, nem $\in$ NEM, and nem supports em to perform some tasks of the VE$-$EM$-A_{\text{ve}}$(em)$\}$. This relation holds the assignments between EMs and its partners (NEMs) to support the tasks of a VE.
7) FT workflow (FTWf) $\subseteq$ FT $\times$ FT, many-to-many binary relation on FT, is denoted by $R_{\text{FTWf}} = \{(\text{ft}_i, \text{ft}_j) : \text{ft}_i, \text{ft}_j \in \text{FT}, p_i, p_j \in P, \text{ft}_i \subset p_i, \text{ft}_j \subset p_j, i \neq j$, ft$_i$ is an event FT of the action FT ft$_j\}$ that indicates that ft$_j$ is authorized to use the PublicPs of ft$_i$ when ft$_i$ is accomplished.
8) Correspondence $\subseteq$ FT $\times$ FT, one-to-one binary relation on FT, is represented by $R_{\text{correspondence}} = \{(\text{ft}_i, \text{ft}_j) : \text{ft}_i, \text{ft}_j \in \text{FT}, p_i, p_j \in P, \text{ft}_i \subset p_i, \text{ft}_j \subset p_j, i \neq j$, ft$_i$ is the preversion offt$_j$, whereas ft$_j$ is the postversion offt$_i\}$.

5) Assignments Across Models: This section defines the assignment relations across models in order to establish the combination relations of relevant elements among two AC models. These relations are as follows.

1) EM$-U-A \subseteq$ EM $\times\, U$, one-to-many EM to U assignment, is represented by $R_{\text{em-u-a}} = \{(\text{em}, u)$: em $\in$ EM, $u \in U$, and em have an employee $u\}$. If $\exists$ em$_1 R_{\text{em-u-a}} u_1$, em$_2 R_{\text{em-u-a}} u_2$, em$_1$, em$_2 \in$ EM, and $u_1, u_2 \in U$, then $\neg\exists$ em$_2 R_{\text{em-u-a}} u_1$.
2) NEM$-U-A \subseteq$ NEM$\times U$, one-to-many NEM to U assignment, is denoted by $R_{\text{nem-u-a}} = \{(\text{nem}, u)$: nem $\in$ NEM, $u \in U$, and nem have an employee $u\}$. If $\exists$ nem$_1 R_{\text{nem-u-a}} u_1$, nem$_2 R_{\text{nem-u-a}} u_2$, nem$_1$, nem$_2 \in$ NEM, and u_1, $u_2 \in U$, then $\neg\exists$ nem$_2 R_{\text{nem-u-a}} u_1$.
3) $R-$VER$-A \subseteq R \times$ VER, many-to-many R to VER assignment, is represented by $R_{\text{r-ver-a}} = \{(r, \text{ver})$: $r \in R$, ver $\in$ VER, and r is assigned to play ver$\}$, then VER ver can be assigned to different roles, whereas one role can play different VERs at the same time.

IV. PRs

A PR (R_{p}) indicates the level of information exchange and reuse and also the situation of cooperation between two projects. Various PRs describing the relation between two projects can propagate the authorizations of an FT to other FTs. Different PRs may occur between two projects and may alter with time based on project management and share requirements. While a VEAC-based AC platform is implemented, administrators construct the project resource access strategy in a PACP to indicate the level of resource sharing of each type of PRs. In the project life cycle, the PRs and the PACP can be modified at any time to respond to the demands of resource sharing. Resource sharing or reusing is determined based on five attributes of each FT: 1) FT state (A_{state}) holds the status of the FT being performed; 2) FT stage (A_{stage}) records the current timestamp of an FT for appropriate resource sharing according to its states; 3) allowed reference (A_{ref}) decides whether the FT can be referred by relative FT in a postversion project; 4) allowed subproject ($A_{\text{sub-p}}$) determines whether the FT can be referred by its subprojects; and 5) allowed main project ($A_{\text{main-p}}$) decides whether the FT can be referred by its main project.

To introduce the PRs, given a set Project (P) and $x, y \in P$, a binary relation PR (R_{p}) on P is a subset of $P \times P$, which is distinguished into five subrelations presented in the following sections. For convenience in the following discussion, two inherited functions applied in the following sections are defined to indicate varying degrees of privilege inheritance.

1) Strong-inherited function Inher$_{\text{strong}}$(ft) is defined as all permissions assigned to the ft are inherited, including read (to retrieve data), update (to modify data), insert (to write new data), and create (to create an object).
2) Weak-inherited function Inher$_{\text{weak}}$(ft) is defined as only read permission from the ft is inherited.

A. Subset Relation

Subset relation (R_{ps}) describes the relation between a main project and its subproject. The relation simplifies a large number of assignments. For instance, an FT called announcement shows information about the progress of a project. Through the subset relation, all workers in the main project and subprojects of the project are permitted to look up the progress of the project. The set of pairs of projects between which have subset relation is represented by $xR_{\text{ps}}y = \{(x, y) : x, y \in P, x \neq y$, and x "is a subset of" $y\}$. A main project is permitted to access the resources of its subproject, but an administrator may set or disable the capability by changing the status of the allowed main-project attribute of its each FT. Fig. 3 shows an example of the subset relation to demonstrate these constraints,

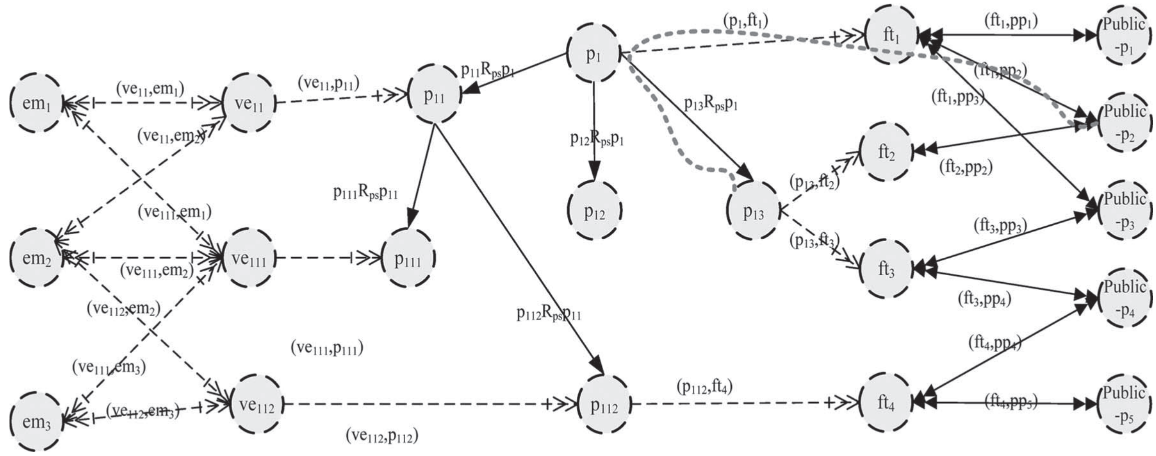


Fig. 3. Example of subset relation.

where Project p_1 involves three subprojects p_{11}, p_{12}, and p_{13}, and project p_{11} is further decomposed into subprojects p_{111} and p_{112}. Owing to $p_{13}R_{\rm ps}p_1$, PublicPs, such as public-p_2 and public-p_3, are assigned to p_1 and p_{13} via FTs, whereas EMs, such as $\rm em_1$ and $\rm em_2$, are permitted to participate in $\rm ve_{11}$ and $\rm ve_{111}$. Two functions $\text{privilege}_{\rm main-p}(\rm ft)$ and $\text{privilege}_{\rm sub-p}(\rm ft)$ are defined, respectively, in (1) and (2), shown at the bottom of the page, for propagating user's privilege from the main project and subproject, respectively, where variables are introduced as follows. Function (1) indicates that the privileges of $\rm ft_{1\it i}$ involve the PublicP and PrivateP assigned to the $\rm ft_{1\it i}$ and $\rm ft_{2\it j}(1 \le j \le n)$ when the conditions shown in the equation hold; otherwise, the privileges of $\rm ft_{1\it i}$ only have the PublicP and PrivateP from $\rm ft_{1\it i}$. Due to the limited space, function (2) shows the propagation of user privileges from subproject, which is similar to function (1) and is not further introduced in detail. p_1 is the main project of p_2 that is the subproject of p_1, $\rm ft_{1\it i}$'s are the FTs involved in p_1, $1 \le i \le m$, and $\rm ft_{2\it j}$'s are the FTs involved in p_2, $1 \le j \le n$. Several constraints are applied to use a subset relation: 1) A main project may have more than one subproject; 2) a subproject is only involved in one main project; 3) an EM may participate in the main and subprojects; and 4) a PublicP is only permitted to be assigned to different projects with subset relations.

B. Version Relation

Version relation $(R_{\rm pv})$ describes a project y called a postversion project that is extended from a project x called preversion project and planned with reference to the preversion project. Therefore, the pre- and postversion projects have similar targets, FTs, and participants. The relation helps support version-dependent authorizations by enabling the reuse of resources for a new product, thus reducing its time to market. Because the pre- and postversion projects have similar targets, activities, and participants, the postversion FT in the postversion project corresponds to the preversion FT in the preversion project. While the postversion FT is performed, the privileges owned by the preversion FT are inherited by the postversion FT using the weak inheritance. The set of pairs of projects between which have version relation is represented by $xR_{\rm pv}y = \{(x, y) : x, y \in P, x \ne y,$ and x "is the preversion of" $y\}$. Fig. 4 shows an example of the version relation, which demonstrates that project p_1 is the preversion of project p_2. Project p_1 for developing a car engine consists of FTs $\rm ft_{11}$ and $\rm ft_{12}$, whereas p_2 for developing a new engine based on the engine developed by p_1 comprises $\rm ft_{21}$, $\rm ft_{22}$, and $\rm ft_{23}$. FTs $\rm ft_{11}$ (requirement analysis) and $\rm ft_{12}$ (conceptual design) correspond to $\rm ft_{21}$ (requirement analysis) and $\rm ft_{22}$ (conceptual design), respectively, whereas $\rm ft_{23}$ (primary design) is created for another task, which is not extended from p_1. Therefore, while the $\rm ft_{21}$ performed, workers must refer significantly to information owned by $\rm ft_{11}$. Due to $p_1R_{\rm pv}p_2$, each FT in project p_2 is performed by VERs, which are allowed to refer to PublicPs of corresponding FTs in p_1 if the attribute allowed reference of corresponding FT is true. As shown in Fig. 4, a user u_1 is assigned to perform the $\rm ft_{21}$ through (u_1, r_1), $(r_1, \rm ver_{21})$, and $(\rm ver_{21}, ft_{21})$; in addition to the public-p_{21} and public-p_{22}, u_1 may refer to the public-p_{11}, public-p_{12}, and public-p_{13}. Function (3) shown at the bottom of the next page is presented to indicate that the privileges of $\rm ft_{2\it j}$ involve the PublicP and PrivateP assigned to the $\rm ft_{2\it j}$, and partial PublicPs of the corresponded FT $\rm ft_{1\it i}$ of $\rm ft_{2\it j}$ through the use of weak inheritance function when the conditions shown in the function hold; otherwise, the privileges of $\rm ft_{2\it j}$ only have the PublicP and PrivateP from $\rm ft_{2\it j}$. p_1 is the preversion project of p_2 that is the postversion project of p_1, $\rm ft_{1\it i}$'s are the FTs involved in p_1, and $\rm ft_{2\it j}$'s corresponding to $\rm ft_{1\it i}$'s are the FTs involved in p_2. Several constraints are applied when using the version relation to support resource sharing between two projects: 1) A postversion project has less than one preversion

$$\text{privilege}_{\rm main-p}(\mathrm{ft}_{1i}) = \begin{cases} \mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{1i}) \cup \mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{2j}) & \text{if } \exists\, p_1R_{\rm ps}p_2 \wedge A_{\rm main-p} \text{ of } \mathrm{ft}_{2j} = \text{“true”} \\ \mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{1i}) & \text{otherwise} \end{cases} \quad (1)$$

$$\text{privilege}_{\rm sub-p}(\mathrm{ft}_{2j}) = \begin{cases} \mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{2j}) \cup \mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{1i}) & \text{if } \exists\, p_1R_{\rm ps}p_2 \wedge A_{\rm sub-p} \text{ of } \mathrm{ft}_{1i} = \text{“true”} \\ \mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{2j}) & \text{otherwise} \end{cases} \quad (2)$$

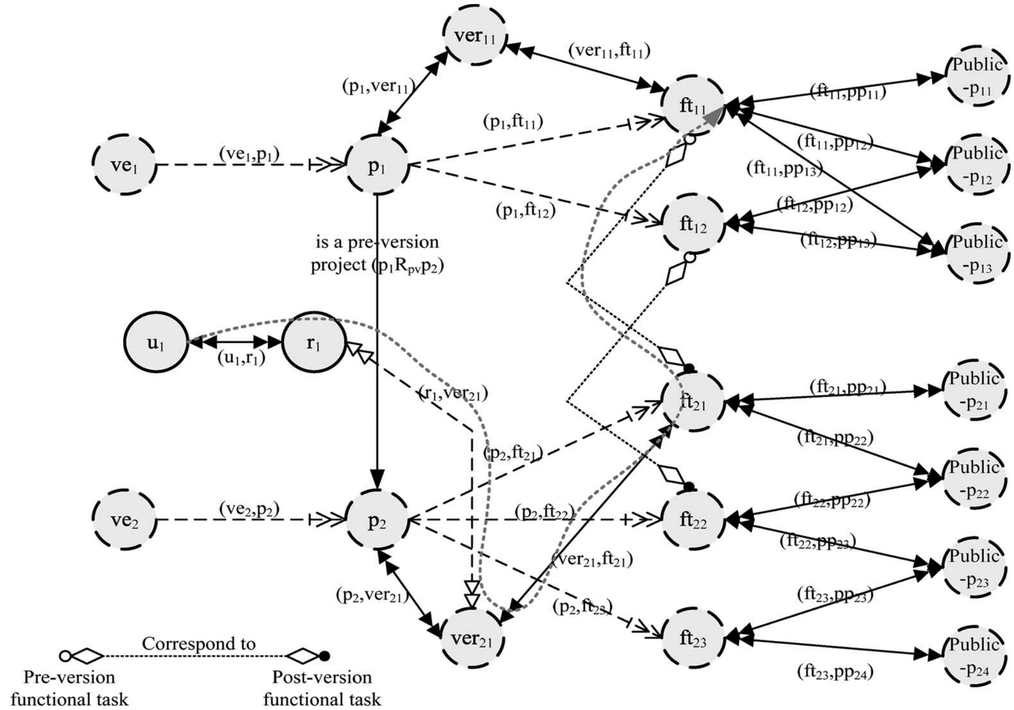


Fig. 4. Example of version relation.

project contrariwise; 2) an FT has less than one corresponding FT; and 3) an EM may participate in pre- and postversion projects simultaneously.

C. Reference Relation

Reference relation (R_{pr}) describes a project x called referring project referring to resources in other project y that is called the referred project. The reference relation indicates that the same users and enterprises can participate in both the referring and referred projects. If two projects have a reference relation, then users in the referring project can refer to the resources of the referred project. While the value of attribute allowed reference of an FT equals true, then the FT can be referred. The set of pairs of projects between which are referred by each other is represented by $xR_{\mathrm{pr}}y = \{(x, y) : x, y \in P, x \neq y$, x refers to resources in y, and $(\neg\exists\, xR_{\mathrm{pe}}y) \wedge (\neg\exists\, yR_{\mathrm{pe}}x)\}$. Project x may refer to y if and only if the following conditions hold: $R_{xi} \cap R_{yj}$, $\mathrm{EM}_{xm} \cap \mathrm{EM}_{yn}$, $\mathrm{FT}_{xk} \cap \mathrm{FT}_{yh}$, $\mathrm{PublicP}_{xv} \cap \mathrm{PublicP}_{yw}$, and $\mathrm{PrivateP}_{xe} \cap \mathrm{PrivateP}_{yf}$ permit unequal ϕ, where R_{xi}, EM_{xm}, FT_{xk}, $\mathrm{PublicP}_{xv}$, and $\mathrm{PrivateP}_{xe}$ are associated with project x, and R_{yj}, EM_{yn}, FT_{yh}, Public-P_{yw}, and Private-P_{yf} are associated with project y. That is, roles, EMs, FTs, PublicP, and PrivateP may be assigned to p_1 and p_2. Fig. 5 shows an example of the reference relation, which indicates that project p_1 can refer to project p_2 through the reference relation $p_1R_{\mathrm{pr}}p_2$. Role r_{31} is assigned to perform VERs ver_{11} and ver_{21}, ver_{11} performs FTs ft_{11} and ft_{12} in project p_1, and ver_{21} performs ft_{21} in project p_2. Therefore, user u_{31} may utilize the public-p_{11}, public-p_{12}, public-p_{13}, public-p_{21}, and public-p_{22} through (u_{31}, r_{31}). The following constraints are applied when using the reference relation: 1) A project may be assigned to more than one project for resource sharing, and 2) a project may refer to more projects simultaneously.

D. Process Relation

Process relation (R_{pp}) describes the executive sequence of two subprojects from the time view and can determine the time for sharing project resources. A process relation can be applied to determine the executive sequence of all subprojects of a project. The set of pairs of projects between which have process relation is represented by using $xR_{\mathrm{pp}}y = \{(x, y) : x, y, z \in P, x \neq y \neq z, (\exists\, xR_{\mathrm{ps}}z) \wedge (\exists\, yR_{\mathrm{ps}}z)$, and x "must be achieved, then start" $y\}$. While the relation is built on two projects, the administrator must specify the sequences of related FTs across the project boundary. This relation can support process-dependent authorization propagation when executing an action FT that can use the resources of the event FTs in event project. Fig. 6 shows an example of a process relation, in which project p_1 denotes the event project of action project p_2; p_1 performs ft_{11} and ft_{12}, and p_2 performs ft_{21}, ft_{22}, and ft_{23}; and ft_{11} denotes an event FT that triggers the ft_{21} and ft_{22} (called action FTs). When ft_{21} is

$$\mathrm{privilege}_{\mathrm{version}}(\mathrm{ft}_{2j}) = \begin{cases} \mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{2j}) \cup \mathrm{Inher}_{\mathrm{weak}}\left(\mathrm{FT{-}PublicP{-}}A_{\mathrm{public\_p}}(\mathrm{ft}_{1i})\right) & \text{if } \exists\, p_1R_{\mathrm{pv}}p_2 \wedge A_{\mathrm{ref}} \text{ of } \mathrm{ft}_{1i} = \text{“true”} \\ \mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{2j}) \text{ otherwise} & \end{cases} \tag{3}$$

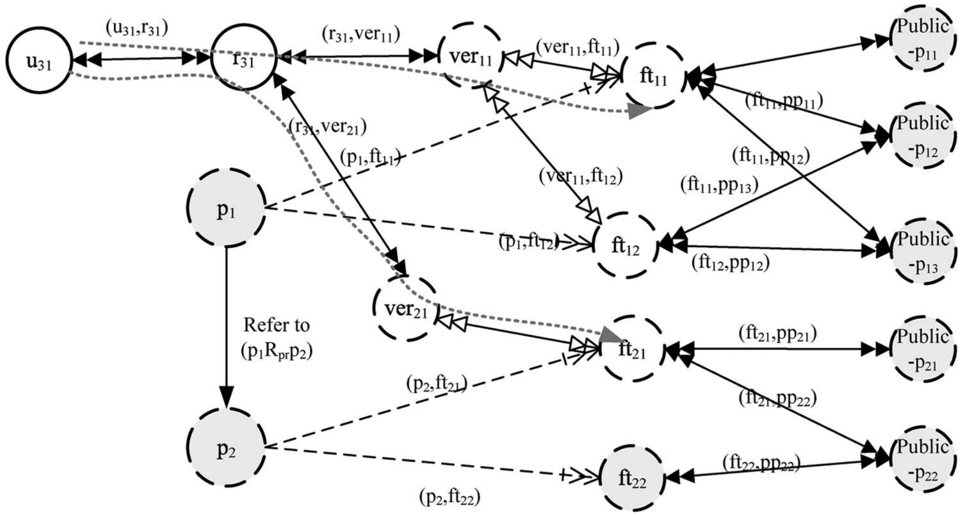


Fig. 5. Example of reference relation.

triggered and performed, user u_1 obtains authorizations public-p_{11}, public-p_{12}, and public-p_{13} from ft_{11}, and authorizations public-p_{21} and public-p_{22} from ft_{21}.

Function (4) shown at the bottom of the page, showing the propagation of user privilege by using process relation, is presented to indicate that the privileges of $\mathrm{ft}_{\mathrm{action}}$ involve the PublicP and PrivateP assigned to the $\mathrm{ft}_{\mathrm{action}}$ and also partial permissions of the event FT $\mathrm{ft}_{\mathrm{event}}$ of the $\mathrm{ft}_{\mathrm{action}}$ through the use of the weak inheritance function when the conditions shown in the function hold; otherwise, the privileges of $\mathrm{ft}_{\mathrm{action}}$ only have the PublicP and PrivateP from $\mathrm{ft}_{\mathrm{action}}$. p_1 is the action project of p_2 that is the event project of p_1, $\mathrm{ft}_{\mathrm{action}}$ is the action FT included in p_1, and $\mathrm{ft}_{\mathrm{event}}$ is the event FT included in p_2. Using the process relation must obey the following constraints: 1) A process relation exists between two projects which must have the subset relation; 2) an event project may trigger more than one action project simultaneously; 3) an event FT may trigger more than one action FT simultaneously; and 4) an action project may be triggered if all of its event projects are accomplished.

E. Exclusive Relation

Exclusive relation (R_{pe}) identifies mutual conflict between two projects, signifying that the resources of the two projects cannot refer to each other. The exclusive relation is default. That is, two projects are preset as exclusive relation if no other relation exists between them. The set of pairs of projects that conflict with each other is represented by $xR_{\mathrm{pe}}y = \{(x,y) : x, y \in P, x \neq y, x$ "conflicts with" y, and $(\neg\exists\ xR_{\mathrm{pr}}y) \wedge (\neg\exists\, yR_{\mathrm{pr}}x)\}$. If two projects are exclusive, then all users, EMs, FTs, and permissions in a project are exclusive with the other project. That is, an enterprise is disallowed from participating simultaneously in two projects with exclusive relation; attempts by users of the exclusive projects to use the same resources are rejected. Using the process relation must obey the following constraints: 1) A project may conflict with more than one simultaneously; 2) a PublicP may not be assigned to two exclusive projects; and 3) an EM is not allowed to be assigned to two mutual exclusive projects.

V. Cooperation Modes Among Two VERs

This section introduces three cooperation modes among VERs based on the resource sharing requirements of collaborative operations in the VE.

Cooperation mode (R_{c}) describes interactions among VERs according to the dependent level of their duties. Given a set VER, x and $y \in$ VER, a binary relation cooperation relation (R_{c}) on VER is a subset of VER $\times$ VER, which is distinguished into three cooperation relations. For convenience in the following discussion, two items are first defined in terms of authority inheritance. A VER in cooperative mode can inherit strongly or weakly the privileges from the other VER. Strong inheritance means that the privilege of a VER can be fully inherited by the other VER, whereas weak inheritance means that the privilege can only be partially inherited, such as only inheriting read privilege.

1) Dependent single-task mode (R_{cds}) is the most seamless cooperative relationship between two VERs, working together to perform FTs, that have dependencies and share resources with each other. The two VERs' permissions are inherited from each other via strong inheritance (defined in Section IV). When two VERs collaboratively perform different FTs, the users playing the two VERs obtain the same permissions from the FTs. The set of pairs of VERs with R_{cds} is represented by using $xR_{\mathrm{cds}}y = \{(x,y) : x, y \in \mathrm{VER}, x \neq y, \exists(x, \mathrm{ft}_1), (y, \mathrm{ft}_1) \in$

$$\mathrm{privilege}_{\mathrm{process}}(\mathrm{ft}_{\mathrm{action}}) = \begin{cases} \mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{\mathrm{action}}) \cup \mathrm{Inher}_{\mathrm{weak}}\,(\mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{\mathrm{event}})) \\ \text{if } \exists(p_1 R_{\mathrm{pr}} p_2) \wedge (A_{\mathrm{state}} \text{ of } \mathrm{ft}_{\mathrm{event}} = \text{“achieved”}) \wedge (A_{\mathrm{ref}} \text{ of } \mathrm{ft}_{\mathrm{event}} = \text{“true”}) \\ \mathrm{FT{-}Permission{-}}A(\mathrm{ft}_{\mathrm{action}}) \text{ otherwise} \end{cases} \quad (4)$$

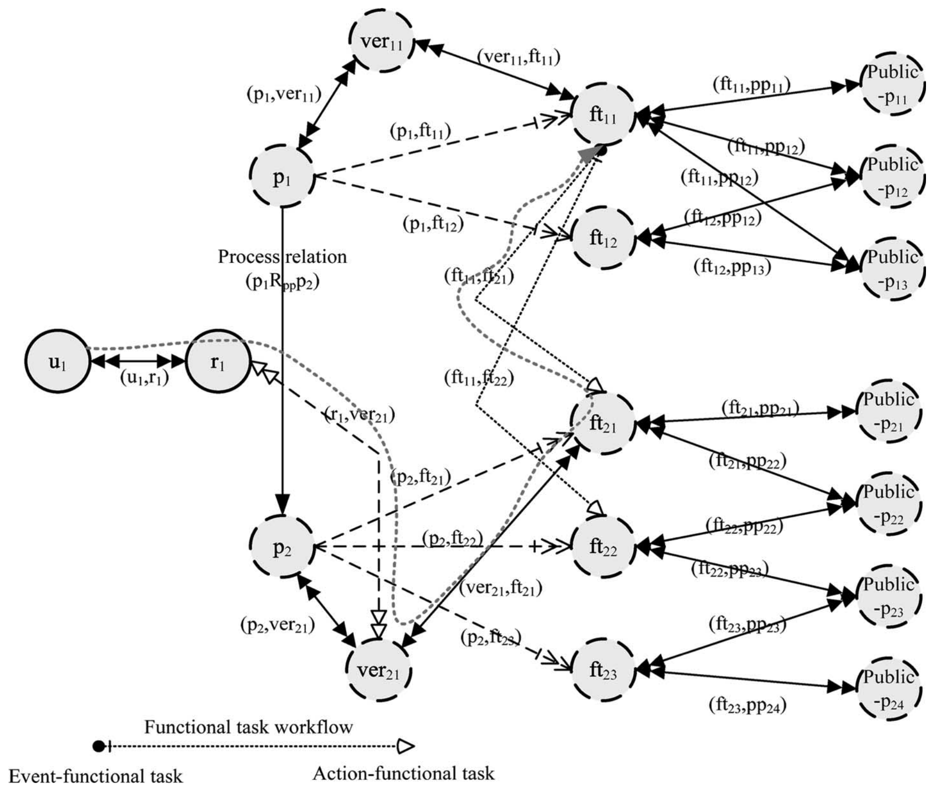


Fig. 6. Example of process relation.

VER−FT−A → FT−PublicP−$A_{\text{public\_p}}$($\{$VER − FT − A_ft$(x) : (x, \text{ft}) \in$ VER−FT−$A\}$) are inherited strongly by VER y, and FT−PublicP−$A_{\text{public\_p}}$($\{$VER−FT−A_ ft$(y) : (y, \text{ft}) \in$ VER−FT−$A\}$) are inherited strongly by VER x, and $(\neg\exists\ xR_{\text{cdm}}y) \wedge (\neg\exists\ yR_{\text{cdm}}x) \wedge (\neg\exists\ xR_{\text{ci}}y) \wedge (\neg\exists\ yR_{\text{ci}}x)\}$ means that VERs x and y cooperate to perform an FT ft$_1$ and have the same access privilege to all its resources.

2) Dependent multitask mode (R_{cdm}) indicates that two VERs interact when performing different FTs. For instance, the results of an FT performed by a VER affect those of an FT performed by another VER. The two VERs inherit each other's permissions via weak inheritance. The set of pairs of VERs with R_{cdm} is represented by using $xR_{\text{cdm}}y = \{(x, y) : x, y \in$ VER, $x \neq y, \forall(x, \text{ft}_x)$, $(y, \text{ft}_y)\in$VER−FT−A→FT−PublicP−$A_{\text{public\_p}}$ ($\{$VER−FT−$A_{\text{ft}}(x) : (x, \text{ft}_x) \in$ VER−FT−$A\}$) are inherited weakly by VER y, and FT−PublicP−$A_{\text{public\_p}}$ ($\{$VER−FT−$A_{\text{ft}}(y) : (y, \text{ft}_y) \in$ VER−FT−$A\}$) are inherited weakly by VER x, and $(\neg\exists\ xR_{\text{cds}}y) \wedge (\neg\exists\ yR_{\text{cds}}x) \wedge (\neg\exists\ xR_{\text{ci}}y) \wedge (\neg\exists\ yR_{\text{ci}}x)\}$. Hence, VERs x and y perform related FTs separately, and that outputs of the FTs are referred to each other.

3) Independent mode (R_{ci}) indicates that two VERs independently perform their FTs, disregarding the outputs generated by other FTs. The relation is applied to protect business secrets when companies that compete with each other perform VERs. If the two VERs work independently, then they are not permitted to perform the same FTs and have each other's access privileges for FTs performed by them. The set of pairs of VERs between which have R_{ci} is represented by $xR_{\text{ci}}y = \{(x, y) :$ $x, y \in$VER, $x \neq y$, T−PublicP−$A_{\text{public\_p}}$ ($\{$VER−FT−$A_{\text{ft}}(x) : (x, \text{ft}_x) \in$ VER−FT−$A\}$) are not inherited by VER y, and FT−PublicP−$A_{\text{public\_p}}$($\{$VER−FT−$A_{\text{ft}}(y)$: $(y, \text{ft}_y) \in$ VER−FT−$A\}$) are not inherited by VER x, and $(\neg\exists\ xR_{\text{cds}}y) \wedge (\neg\exists\ yR_{\text{cds}}x) \wedge (\neg\exists\ xR_{\text{cdm}}y) \wedge (\neg\exists\ yR_{\text{cdm}}x)\}$.

The use of cooperative relations is constrained by the following rules.

1) $\#(\{y : (x_1, y) \in R_c, x_1, y \in \text{VER}\}) \geq 0$ means that a VER x_1 is permitted to have different cooperation modes with other VERs.
2) $\#(\{(x_1, y_1) : (x_1, y_1) \in R_c, x_1, y_1 \in \text{VER}\}) \leq 1$ signifies that only one cooperation mode is permitted between two VERs.

VI. PACP Language Model

Based on the VEAC model, the PACP language model for VEs designed in this paper, as Fig. 7 shows, is represented in class model of Unified Modeling Language (UML) and mainly targets contents of information text. This model features an object–subject–action–condition AC policy consisting of multiple sets of authorization rules that jointly determine user access permissions. Therefore, regarding specific resource (object), authorization (action) to execute certain resource is granted to certain users (subject) under certain restrictions (conditions).

The PACP language model for VEs has been proposed in this section for the following reasons: 1) to provide a method that effectively describes resource AC policy for VEs; 2) to reduce costs and complexity in resource AC; 3) to improve flexibility in managing access permission; and 4) to make the management of resource access permission adaptive to changing needs in a

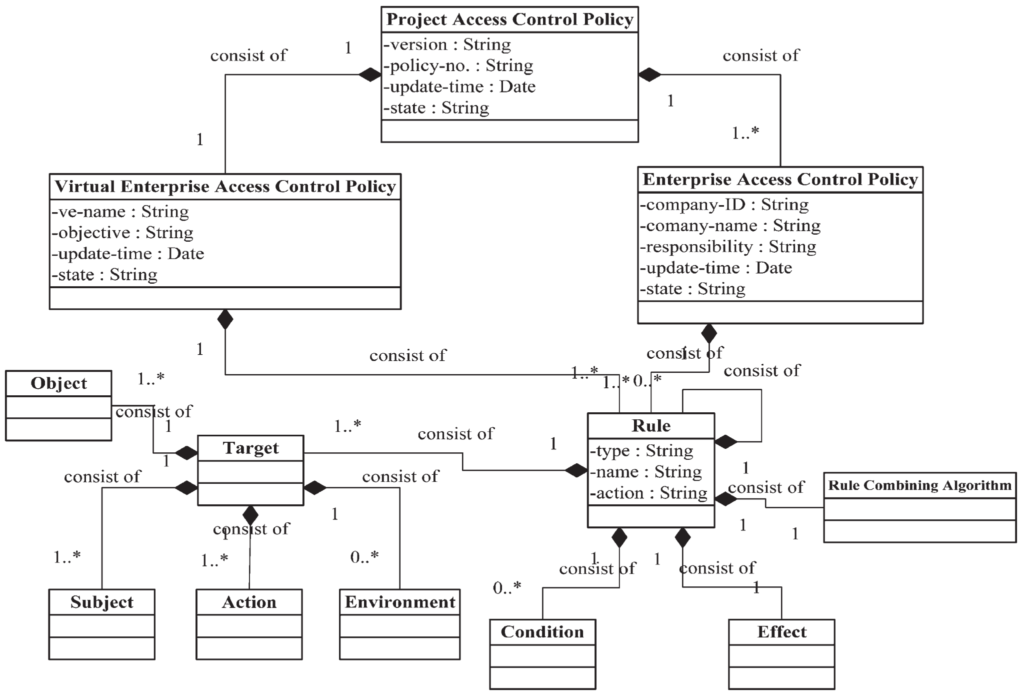


Fig. 7. PACP language model.

business environment in a timely manner. The PACP language model has the following main components.

1) PACP. A PACP consists of one VEAC policy (VEACP) and many enterprise AC policies (EACPs), which are sets of rules.
2) VEACP, a set of rules, describes the regulation and constraint on resource AC and sharing in a VE to manage the VE's resource.
3) EACP, consisting of a series of rules, describes rules and conditions for enterprise resource AC for each EM. Its rules shall not be in conflict with the VEACP it belongs to and must comply with the sharing rules agreed upon by VE so to make available resource in need of sharing.
4) Rule element is the most basic unit of policy and corresponds to the conventional concept of authorization. The principal components of rule have a target, effect, condition, and rule combining algorithm. Each rule permits or denies one or more subjects to performing actions on one or more objects under some conditions.
5) A target element involved in a rule defines the set of objects, subjects, and actions to which the rule or policy applies.
6) Object may be data, information, and knowledge owned by the VE or one of its EMs.
7) A subject is an actor whose attributes may be referenced by a predicate. Actor may be a user, role, enterprise, or VER.
8) An action is an operation on resource.
9) A condition element represents additional constraints that further refine rule applicability.
10) Rule combining algorithm compresses the output from the embraced rules. The PACP language model has four rule combining algorithms: deny overrides, permit overrides, first applicable, and only-one-applicable. Based on the selected combining algorithm, an authorization decision can be permit, deny, not applicable, or indeterminate.
11) Effect is the intended consequence of a satisfied rule—either Permit or Deny.

VII. VEAC Model Construction Methodology

The proposed formal VEAC model can efficiently manage and share information resources in the VE life cycle. To assist the administrators of VEs and their EMs to successfully implement the proposed fundamental VEAC model and to use the PACP language model appropriately for VE information resource security and sharing, this section develops a VEAC model construction methodology based on the five phases of VE life cycle, namely, identification, formation, design, operation, and dissolution phases. The methodology provides the security administrators of the leader and partners of VEs with a systematic method for the following reasons: 1) to identify the fundamental elements of VEAC model, such as P, VER, FT, U, R, PublicP, and PrivateP; and 2) to establish assignments between elements, PRs between projects, and cooperation modes between VERs. The VEAC model applied for certain VE is initially planned at the formation phase, all elements and assignments of the VEAC model are designed at the design phase, and the constructed VEAC model is implemented at the operation phase. Thus, information resources are managed at the operation and dissolution phases. The goal, procedure, inputs, outputs, and related method and technologies of each phase of the methodology are separately introduced in the following sections.

A. *Identification and Formation Phases*

Fig. 8 shows the first two phases in the proposed methodology, namely, identification and formation phases, which are

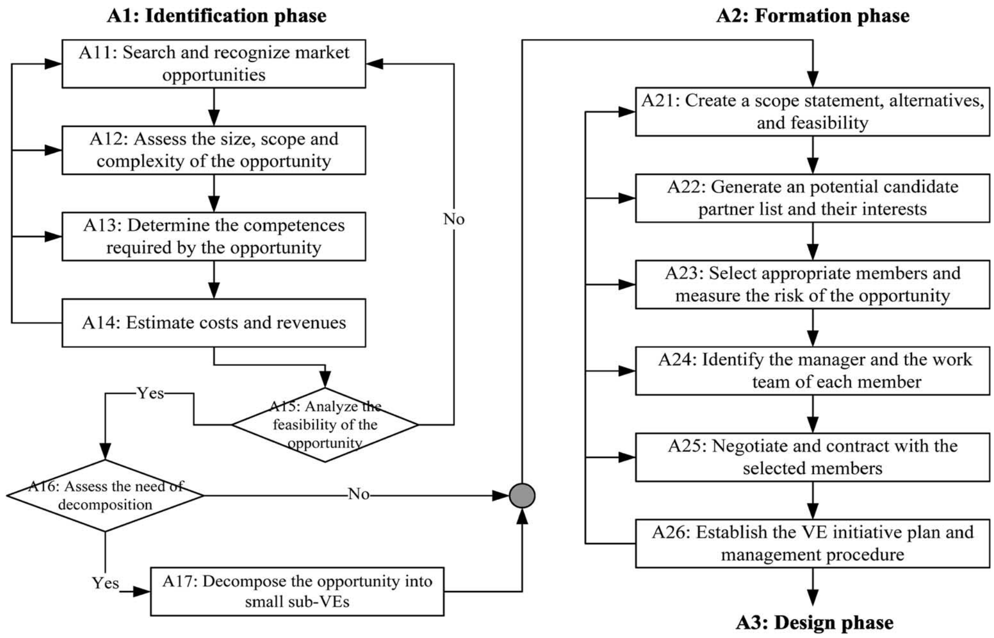


Fig. 8. Identification and formation procedures of a VE.

introduced simply as follows.

1) Identification phase numbered A1 defines the boundaries of a VE to analyze whether the goals, technologies, and cost of the VE are acceptable; to evaluate the complexity of the VE; and to establish procedures for supporting later VE activities. The leader of a VE generally analyzes historical transaction data or carries out market research to find out a valuable and feasible market opportunity and then to form a VE. To achieve the aims of the identification phase, the following seven numbered actions (the left of Fig. 8) should be undertaken in order or repetitively: (A11) searching and recognizing market opportunities; (A12) assessing the size, scope, and complexity of the opportunity; (A13) determining the competences required by the opportunity; (A14) estimating costs and revenues; (A15) analyzing the feasibility of the opportunity; (A16) assessing the need of opportunity decomposition; and (A17) decomposing the opportunity into small sub-VEs to perform the decomposed opportunities, thus establishing R_{ps} between the main and subprojects. The final output of the phase is a practical and valuable opportunity.
2) Formation phase numbered A2 selects suitable partners against alignment factors for their skills, experiences, and capabilities; identifies each member's responsibilities explicitly; ensures that every member of the VE understands his own individual roles and responsibilities; and allocates project resources, including people, service, facilities and equipment, supplies and materials, and money. To accomplish this process at the formation phase, the following six numbered actions (the right of Fig. 8) should be executed in order or repetitively: (A21) creating the scope statement, alternatives, and feasibility of a VE; (A22) generating a potential candidate partner list and their interests; (A23) selecting appropriate partners for the VE and its sub-VEs and measuring the possible risk from the partners; (A24) identifying the VE manager and work team of each partner; (A25) negotiating and contracting with the selected partners for sharing and using resources; and (A26) establishing the initiative plan and management procedure of a VE and its sub-VEs. The final outputs of achieving the six actions include a certain VE organizational structure model and contracts for cooperation among all EMs. The design phase is then executed based on this model.

B. *Design Phase*

The design phase in the proposed methodology is a significant phase for constructing a real VE based on the proposed VEAC model, since it is relative mostly to the plan and design, and resource use and assignment of VEs (Fig. 9). The actions involved at the phase are achieved collaboratively by the security administrators of the VE leader and all partners for managing public and private resources and VE user authorizations. The design phase numbered A3, which includes three subprocedures A31, A32, and A33, is described as follows.

1) Subprocedure A31—Plan and Design VE: The subprocedure models a VE in terms of organization, business, process, and activity perspectives. The detailed organizational structure model, business and resource sharing regulations, VE process model, and activity models of each EM are produced at the end of subprocedure A31. The subprocedure involving six actions is numbered and described below.

(A311) Identify all participators of each partner. Each partner in the VE is assigned certain tasks or responsibilities at the formation phase. The subprocedure starts with action A311 from the organizational view, in which each partner has to choose suitable employees or teams to perform enterprise-assigned tasks, according to employees' skills, experiences, and capabilities. At the time, partners must offer a list of employees who participate directly or indirectly in the VE and are permitted to access the VE resources. The employees involved in the list become user elements in the VEAC model.

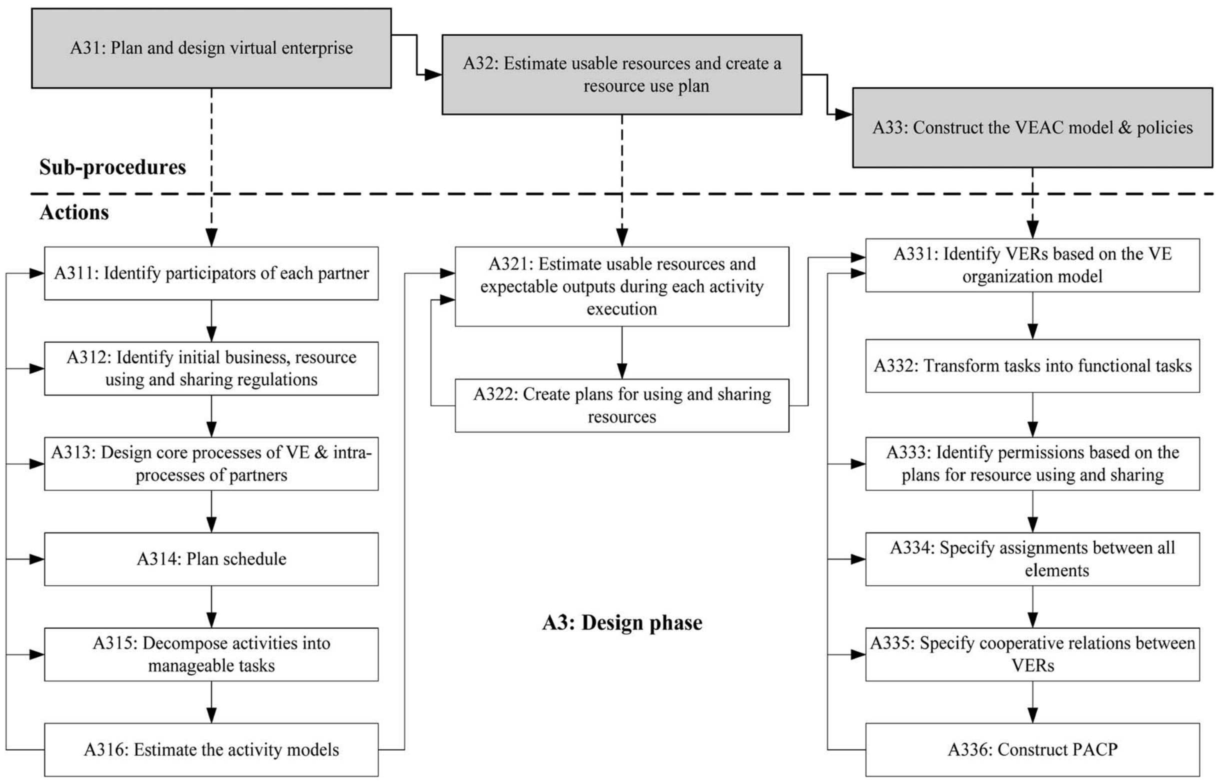


Fig. 9. Design procedure of a VE based on the VEAC model.

(A312) Identify initial business and resource using and sharing regulations. From a business perspective, action A312 identifies regulations regarding usage and sharing of resources to restrict the behavior of partners and specify each participator's responsibility and obligations. In the VE organizational structure model, every participator in a VE is assigned certain tasks, which are performed and restricted by the regulations. The regulations are then converted into VEACP and EACPs by A336.

(A313) Design the core processes of VE and the intraenterprise processes of partners. Based on the planned VE process at A2, a VE leader at the phase designs the core processes of the VE project represented by a project evaluation and review technique chart. The core processes are composed of many activities to accomplish VE's goal. Each activity in the core process is assigned to certain partners to perform. Each partner must then spread up and perform its assigned activities and integrate them into its intraenterprise processes. Finally, PRs $R_{\rm pv}$, $R_{\rm pr}$, $R_{\rm pp}$, and $R_{\rm pe}$ can be established at action A313 if they are needed.

(A314) Plan schedule. According to the core VE processes designed by A313, the VE leader at the action negotiates and communicates with partners to plan the start and end times of each activity in the core VE and intraenterprise processes, and the activity prerequisites.

(A315) Decompose activities into manageable tasks. The activities involved in the core VE and intraenterprise processes are further decomposed into tasks until every task can represent a manageable amount of work that can be planned, scheduled, and assigned. A work breakdown structure, comprising a hierarchical decomposition of project, activities, and tasks, is planned at this point. The decomposed tasks are then further decomposed or combined into manageable tasks in terms of resource AC. The priority of every manageable task is determined from the start and end times of the original tasks, the information flow between tasks and task outputs.

(A316) Estimate the activity models. An activity model is composed of some partially ordered tasks that are conducted to achieve the actions to be performed within a VE. Action A316 estimates the duration of every task and changes the baseline based on reasonable estimations. The following factors should be addressed: 1) the resources that should be used; 2) the amount of time required; 3) how many people are needed; 4) the skills that are necessary; and 5) the tasks that need to be completed before other tasks are started. Subprocedure 2 is executed after all tasks are estimated.

2) Subprocedure A32—Estimate Usable Resources and Create a Resource Use Plan: Subprocedure A32 estimates the usable VE resources and builds a resource use plan for the entire life cycle of a VE. The plan is adopted to restrict assignments between elements and to build the PACP.

(A321) Estimate usable resources and expectable outputs. The first action of this subprocedure estimates usable VE resources according to the activity models outputted by A316. These resources include public and private resources, which are supplied or shared with partners to facilitate the execution of VE tasks. In addition, the administrator has to expect possible outputs during the execution of each task and know whom the outputs will be shared with. Some shared outputs should be specified by specific data containers, which are then converted into permissions and assigned operations permitted on them at A333.

(A322) Create plans for using and sharing resources. Based on the regulations created by A312, the result of A321, and

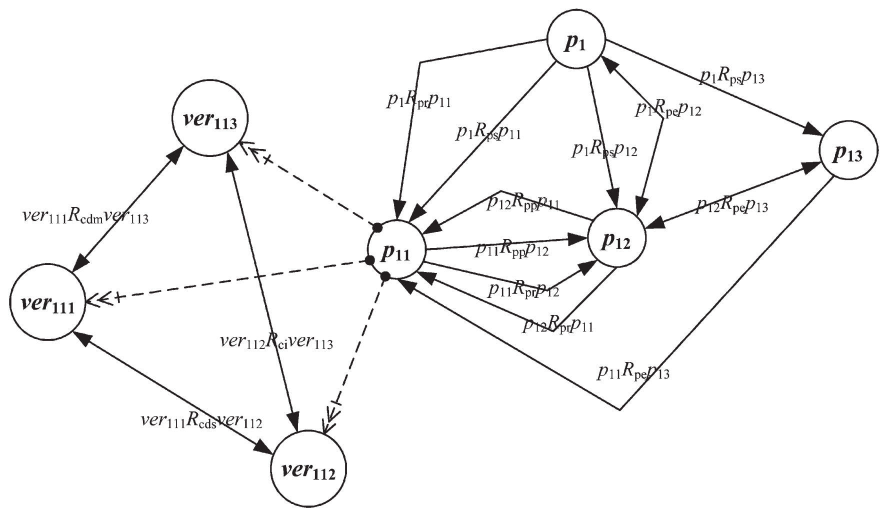


Fig. 10. Engine R&D project.

the activity models of A316, the action forms a resource using and sharing plan, which describes which users, activities, and partners can use or share which resource.

3) Subprocedure A33—Construct the VEAC Model and Policies: Subprocedure A33 identifies the elements, assignments, relations, and policies involved in the VEAC model according to the outputs at subprocedures A31 and A32.

(A331) Identify VERs based on the VE organizational model. VERs can be identified by using two different methods: decomposing the VE's goal or decomposing the VE's organizational structure. In the first method, the aim of a VE is decomposed into little goals that can be completed by a single individual or team. Each of these little goals is then transformed into a VER. In the second method, the organizational structure of a VE is decomposed hierarchically into different levels of element, namely, EM, department, team, role, and user. Several elements in the same level are then chosen to form a VER if they can be assigned to different workers; they have the same authorizations, and they do not have resource security problems resulting from sharing or collaboration. Finally, those single elements that cannot be assigned to different workers are converted into VERs.

(A332) Transform tasks into FTs. Action A332 transforms the manageable tasks in activity models into FTs, whose properties must be filled in. If the resources of a manageable task simultaneously allow and disallow sharing, then the manageable task must be decomposed into two or more FTs.

(A333) Identify permissions based on the plans for resource using and sharing. Action A333 combines resources and assigns operations on the resources to form PublicP and PrivateP. The PublicPs are identified by a VE leader administrator, and the PrivatePs are identified by every partner's administrator.

(A334) Specify assignments between all elements. All elements involved in the VEAC model have been identified at previous actions. This action specifies all assignments between two elements, such as VE$-P-A$, $P-$VER$-A$, VER$-$FT$-A$, FT$-$PublicP$-A$, and $R-$VER$-A$.

(A335) Specify cooperative modes between VERs. The cooperative modes between two VERs are specified here according to the resource usage and sharing plan.

(A336) Construct PACP. Based on the proposed PACP language model, the action utilizes the business regulations and the resource using and sharing plan to build the PACP of the VE, including a VEACP and several EACPs. The VEACP is built by the administrator of the VE leader, and the EACPs of partners are built by their administrators.

C. Operation and Dissolution Phases

1) Operation phase first sets up the real VEAC model modeled in the third phase. The VEAC system can then manage VE information resources and generate user authorizations, and monitor, control, and report progress against goals, schedule, and milestone of the VE.
2) Dissolution phase assesses successes or failure at the conclusion of the VE, and its results pave the experience for the next new VE. The PACP established at the design phase of a real VE must be modified to comply with the resource sharing rules after the VE dissolution.

VIII. Example of Practical VE Applying the VEAC Model

This section utilizes the automobile industry as an example to verify the feasibility of the proposed fundamental VEAC model and the PACP language model. Fig. 10 shows a new car engine R&D project (p_1) performed by VE ve_1. In Fig. 10, only parts of the projects are shown; some elements and assignments regarding the project and its three subprojects are shown in detail in the following tables. The engine R&D project has three subprojects: cylinder head design (p_{11}), cylinder block design (p_{12}), and crankshaft design (p_{13}). p_{11} is associated with p_{12} by using process and reference relations $(p_{11}R_{\mathrm{pp}}p_{12}$ and $p_{11}R_{\mathrm{pr}}p_{12})$; p_{12} is associated with p_{11} via process and reference relations $(p_{12}R_{\mathrm{pp}}p_{11}$ and $p_{12}R_{\mathrm{pr}}p_{11})$; p_{13} is exclusive to p_{11} and p_{12} via $p_{11}R_{\mathrm{pe}}p_{13}$ and $p_{12}R_{\mathrm{pe}}p_{13}$, and p_1 is exclusive to p_{12} via $p_1R_{\mathrm{pe}}p_{12}$. According to these definitions, p_{11} and p_{12} have stronger requirement for resource sharing, whereas p_{12} and p_{13} are independent. This example focuses on trust evaluation between the four projects and trust evaluation between three VERs (ve_{11}, ve_{12}, and ve_{13}) involved

TABLE I
VE−P−A AND P−VER−A LISTS

VE Name	Performed Project	Involved VERs	Objectives of the Project
ve_1: engine R&D	p_1	ver_{11}, ver_{12}	Designing a car engine (displacement: 2000cc., and horsepower >140 Hp)
ve_{11}: cylinder head design	p_{11}	ver_{111}, ver_{112}, ver_{113}	Designing the cylinder head of the engine
ve_{12}: cylinder block design	p_{12}	ver_{121}, ver_{122}, ver_{123}, ver_{124}	Designing the cylinder block and the timing gear cover of the engine
ve_{13}: crankshaft design	p_{13}	ver_{131}, ver_{132}, ver_{133}, ver_{134}, ver_{135}, ver_{136}	Designing the crankshaft and connecting rod of the engine

TABLE II
COMPANY LIST

Company No.	Company Name	Number of Employees	Company Address	Core Capacities
em_1	Company-A	100	Tainan Taiwan	Block, Internal Combustion Engine
em_2	Company-B	20	Taipei Taiwan	Cooling System
em_3	Company-C	200	Beijing China	Cylinder
em_4	Company-D	1200	Detroit USA	Cylinder
em_5	Company-E	5	TaichungT aiwan	Internal and External Combustion Engines
em_6	Company-F	13	Tokyo Japan	Main Bearing, Vibration Damper
em_7	Company-G	100	Shangha China	Flywheel, Crankshaft, Cam

in p_{11}; hence, some elements or assignments are ignored in the following tables.

Table I, the VE−P−A and P−VER−A lists, shows the VE name, project performed by the VE, the VERs involved in the VE, and the project objectives. For example, ve_1 involves two VERs, ver_{11} and ver_{12}, and performs project p_1 whose aim is to develop a 2000 cc car engine with at least 140 hp.

Table II lists the detailed information for each company participating in the four VEs.

Table III, the VE−EM−A list, shows all EMs in each VE; for instance, the companies participating in ve_{11} are em_1, em_2, and em_3.

Table IV lists the attributes of FTs that are associated with the four projects, including the number, name, allowed reference, allowed subproject, and allowed main-project attributes.

Table V lists the P−FT−A with project names, the number of FTs assigned to the projects, and the FTs involved in the projects.

Table VI lists the executed sequence of FTs involved in the two projects (p_{11} and p_{12}) between which a process relation is held. Consequently, when the event FT ft_{111} is achieved, the action FT ft_{121} is triggered. According to the process relation definition, ft_{121} will hierarchy all or part of the privileges assigned to ft_{111} when ft_{121} is executed.

Table VII, FT−PublicP−A, lists each FT and PublicPs assigned to each FT.

Table VIII shows the VER−FT−A list, in which only VERs involved in ve_{11} are considered and listed.

In the aforementioned example, ve_{11} (cylinder head design) is used as an example to construct PACP for managing resources that belong to ve_{11}, as shown in the Appendix II. With the objective of cylinder head design of a new car engine, this VE consists of three EMs, i.e., Company-A (em_1), Company-B (em_2), and Company-C (em_3), responsible for oil filler cap design, cylinder head design, and stopper design, respectively.

TABLE III
VE−EM−A LIST

VE Name	Enterprise Members
ve_1	$em_1, em_2, em_3, em_4, em_5, em_6, em_7$
ve_{11}	em_1, em_2, em_3
ve_{12}	em_3, em_4, em_5
ve_{13}	em_6, em_7

In this PACP (see Appendix II), only part of the rules in the VEACP and part of the rules in the EACP of Company-A are listed. According to VEACP rule-ve_{11}-001, when two tasks ft_{111} and ft_{112} are being executed from May 20, 2007 to October 20, 2008, all Company-A, Company-B, and Company-C personnel may read knowledge of know-what about cylinder head design, car engine, and cylinder. The EACP rule-em_1-001 for Company-A dictates that, from November 20, 2007 to October 20, 2008, all Company-B and Company-C personnel may read R&D knowledge related to oil filler cap design.

IX. CONCLUSION AND FUTURE WORK

The results and contributions of this paper are as follows.

1) The formal VEAC model, including the fundamental VEAC model, PACP language model, and construction

TABLE IV
ATTRIBUTE LIST OF FTs

FT No.	FT Name	Attributes		
		Allowed-reference	Allowed-sub-project	Allowed-main-project
ft_{11}	Sub-project progress management	T	T	F
ft_{12}	Sub-project progress management	T	T	F
ft_{13}	Sub-project progress management	T	T	F
ft_{14}	Bulletin	T	T	T
ft_{111}	Oil filler cap design	T	T	F
ft_{112}	Cylinder head design	T	T	F
ft_{113}	Stopper design	T	T	F
ft_{121}	Cylinder liner design	T	T	F
ft_{122}	Cylinder head knock pin design	T	T	F
ft_{123}	Clutch housing design	T	T	F
ft_{124}	Engine rear bracket design	T	T	F
ft_{131}	Crankshaft design	F	F	F
ft_{132}	Crankshaft bearing upper metal design	F	F	F
ft_{133}	Lower oil ring design	F	F	F

TABLE V
P−FT−A LIST

Project Name	Number of FTs	Functional Tasks
p_1	4	$ft_{11}, ft_{12}, ft_{13}, ft_{14}$
p_{11}	4	$ft_{11}, ft_{111}, ft_{112}, ft_{113}$
p_{12}	5	$ft_{12}, ft_{121}, ft_{122}, ft_{123}, ft_{124}$
p_{13}	4	$ft_{13}, ft_{131}, ft_{132}, ft_{133}$

TABLE VI
SEQUENCE LIST

Event-Functional Task	Action-Functional Task
ft_{111}	ft_{121}
ft_{121}	ft_{122}
ft_{122}	ft_{112}
ft_{112}	ft_{123}

TABLE VII
FT−PublicP−A LIST

FT	Public Permissions
ft_{11}	$public\text{-}p_1$
ft_{12}	$public\text{-}p_2$
ft_{13}	$public\text{-}p_3$
ft_{14}	$public\text{-}p_4$
ft_{111}	$public\text{-}p_5$
ft_{112}	$public\text{-}p_6$
ft_{113}	$public\text{-}p_7$
ft_{121}	$public\text{-}p_7, public\text{-}p_8$
ft_{122}	$public\text{-}p_6, public\text{-}p_8$
ft_{123}	$public\text{-}p_9$
ft_{124}	$public\text{-}p_{10}$
ft_{131}	$public\text{-}p_{11}, public\text{-}p_{13}$
ft_{132}	$public\text{-}p_{14}$
ft_{133}	$public\text{-}p_{15}$

TABLE VIII
VER−FT−A LIST

VER	Performed Functional Tasks
ver_{111}	$ft_{11}, ft_{111}, ft_{112}$
ver_{112}	$ft_{11}, ft_{112}, ft_{113}$
ver_{113}	ft_{12}, ft_{112}

methodology, is proposed to facilitate VE resource management and sharing across organizations.

2) The fundamental VEAC model is designed to adapt to changes in VE members, both individuals and organizations, without affecting authorities of VERs, and eliminates the need to reset users' access authorities due to changes in cooperation targets.
3) Participation or withdrawal of an enterprise does not change the existing management model of resource access, thus significantly reducing administrative cost and complexity.

The results of this paper may help VEs solve the challenges of resource management and sharing among enterprises. Resource management and sharing will become increasingly complicated in the future owing to the requirement of strong information transparency. The proposed formal VEAC model solves AC and VE resource sharing challenges.

However, this paper has some deficiencies. For instance, the non-RBAC model, and integration of its access policies, has not been explored. An enterprise that adopts non-RBAC models and other access policies must perform additional model-transferring process to transform the models to RBAC to integrate them into the proposed PBAC model. This paper does not consider the possibility that the user might share a resource with unauthorized users, for example, by copying it, after legally acquiring the resource. The works in future are listed as follows.

1) An enterprise might adopt a non-RBAC-based scheme. Therefore, integrating different AC schemes or policies should be a focus for future works.
2) An enterprise should ideally retain its original AC model when joining a VE. Hence, a "plug-and-play" AC integration mechanism should be developed.
3) Because an enterprise might participate in several competing VEs, preventing the leaking of key technology or data should be considered.
4) Distributed security infrastructure including distributed heterogonous security architecture and collaborative VE policy management approaches should be completely designed for implementing the VEAC system.

APPENDIX I

TABLE IX
LIST OF FUNCTIONS RELATED TO THE VEAC MODEL

Function	Domain	Co-domain	Description
$VE\text{-}EM\text{-}A_{\mathrm{em}}(ve)$	VE	2^{EM}	a *ve* to a set of *EM*s that participate in this *ve*
$VE\text{-}EM\text{-}A_{\mathrm{ve}}(em)$	EM	2^{VE}	an *em* to a set of *VE*s that involve this *em*
$VE\text{-}P\text{-}A_{\mathrm{ve}}(p)$	P	VE	a project *p* to a *VE* that performs this *p*
$VE\text{-}P\text{-}A_{\mathrm{p}}(ve)$	VE	2^{P}	a VE *ve* to a set of *Project*s that are performed by this *ve*
$P\text{-}VER\text{-}A_{\mathrm{p}}(ver)$	VER	P	a *ver* to a project *p* that involves this *ver*
$P\text{-}VER\text{-}A_{\mathrm{ver}}(p)$	P	2^{VER}	a project *p* to a set of *VER*s that are assigned to this *p*
$P\text{-}FT\text{-}A_{\mathrm{ft}}(p)$	P	2^{FT}	a project *p* to a set of *FT*s that are involved in this *p*
$P\text{-}FT\text{-}A_{\mathrm{p}}(ft)$	FT	P	a *ft* to a project that involves this *ft*
$VER\text{-}FT\text{-}A_{\mathrm{ver}}(ft)$	FT	2^{VER}	a *ft* to a set of *VER*s that perform this *ft*
$VER\text{-}FT\text{-}A_{\mathrm{ft}}(ver)$	VER	2^{FT}	a *ver* to a set of *FT*s that are performed by this *ver*
$EM\text{-}U\text{-}A_{\mathrm{u}}(em)$	EM	2^{U}	an *em* to a set of *U*s that are employees of this *em*
$EM\text{-}U\text{-}A_{\mathrm{em}}(u)$	U	EM	a user *u* to an *EM* that involves this *u*
$NEM\text{-}U\text{-}A_{\mathrm{u}}(nem)$	NEM	2^{U}	a *nem* to a set of *U*s that are employees of this *nem*
$NEM\text{-}U\text{-}A_{\mathrm{nem}}(u)$	U	NEM	a user *u* to a *NEM* that involves this *u*
$EM\text{-}NEM\text{-}A_{\mathrm{em}}(nem)$	NEM	2^{EM}	a *nem* to a set of *EM*s with tasks are supported by this *nem*
$EM\text{-}NEM\text{-}A_{\mathrm{nem}}(em)$	EM	2^{NEM}	an *em* to a set of *NEM*s that support some tasks of this *em*
$R\text{-}VER\text{-}A_{\mathrm{r}}(ver)$	VER	2^{R}	a *ver* to a set of *R*s that play this *ver*
$R\text{-}VER\text{-}A_{\mathrm{ver}}(r)$	R	2^{VER}	a role *r* to a set of *VER*s that this *r* plays
$FT\text{-}PublicP\text{-}A_{\mathrm{public\_p}}(ft)$	FT	$2^{PublicP}$	a *ft* to a set of *PublicP*s over all stages
$FT\text{-}PublicP\text{-}A_{\mathrm{ft}}(public\_p)$	$PublicP$	2^{FT}	a *public_p* to a set of *FT*s over all stages
$FT\text{-}PrivateP\text{-}A_{\mathrm{private\_p}}(ft)$	FT	$2^{PrivateP}$	a *ft* to a set of *PrivateP*s over all stages
$FT\text{-}PrivateP\text{-}A_{\mathrm{ft}}(private\_p)$	$PrivateP$	2^{FT}	a *private_p* to a set of *FT*s over all stages
$FT\text{-}Permission\text{-}A(ft)$	FT	$2^{PublicP \quad PrivateP}$	a *ft* to a set of *Permission*s (including private and public permissions) over all stages
$FT\text{-}PublicP\text{-}A_{\mathrm{ft}}(st)$	$Stage$	2^{FT}	a stage *st* to a set of *FT*s
$Stage_{\mathrm{public\_p}}(st)$	$Stage$	$2^{PublicP}$	a stage *st* to a set of *Public-P*s, $Public\_Permission(st) \subseteq \{public\_p: (FT\text{-}PublicP\text{-}A_{ft}(st), public\_p) \in FT\text{-}PublicP\text{-}A\}$, which can change with *st*
$Stage_{\mathrm{ft}}(st)$	$Stage$	2^{FT}	a stage *st* to a set of *FT*s, $Functional\_Task(st) \subseteq \{ft: (FT\text{-}PublicP\text{-}A_{ft}(st), public\_p) \in FT\text{-}PublicP\text{-}A\}$, which can alter with *st*
$Correspondence_{\mathrm{post}}(ft)$	FT	FT	a pre-version FT *ft* to its post-version FT
$Correspondence_{\mathrm{pre}}(ft)$	FT	FT	a post-version FT *ft* to its pre-version FT
$FTWf_{\mathrm{event}}(ft)$	FT	2^{FT}	an action FT *ft* to a set of its event FTs
$FTWf_{\mathrm{action}}(ft)$	FT	2^{FT}	an event FT *ft* to a set of its action FTs
$RH_{\mathrm{senior}}(r)$	R	2^{R}	a role *r* to a set of *R*s, which are the senior roles of the *r*
$RH_{\mathrm{junior}}(r)$	R	2^{R}	a role *r* to a set of *R*s, which are the junior roles of the *r*
$PR_{\mathrm{subset}}(p)$	P	2^{P}	a project *p* to a set of *P*s with which the *p* has a subset relation
$PR_{\mathrm{version}}(p)$	P	2^{P}	a project *p* to a set of *P*s with which the *p* has a version relation
$PR_{\mathrm{reference}}(p)$	P	2^{P}	a project *p* to a set of *P*s with which the *p* has a reference relation
$PR_{\mathrm{process}}(p)$	P	2^{P}	a project *p* to a set of *P*s with which the *p* has a process relation
$PR_{\mathrm{exclusive}}(p)$	P	2^{P}	a project *p* to a set of *P*s with which the *p* has a exclusive relation
$CM_{\mathrm{cds}}(ver)$	VER	2^{VER}	a *ver* to a set of *VER*s that cooperate with the *ver* by using dependent single-task mode
$CM_{\mathrm{cdm}}(ver)$	VER	2^{VER}	a *ver* to a set of *VER*s that cooperate with the *ver* by using multi-task mode

APPENDIX II

TABLE X

EXAMPLE OF PACP FOR THE DEVELOPMENT OF CYLINDER HEAD OF A CAR ENGINE

```
<PACP Version= "version 1.1.1" Policy-no.= "N00233" Update-time= "5/15/2007" State= "active">
    <VEACPVE-name= "ve11" objective= "cylinder head design" Update-time= "5/15/2007" State= "active">
        <RuleSet>
            <RuleCombiningAlgorithm>permit-overrides</RuleCombiningAlgorithm>
            <Rule Type= "rule-kind" Name= "rule-ve11-001" Action= "active">
                <Target>
                    <SubjectSet>
                        <Subject>Company-A</Subject>
                        <Subject>Company-B</Subject>
                        <Subject>Company-C</Subject>
                    </SubjectSet>
                    <ActionSet>
                        <Action>read</Action>
                    </ActionSet>
                    <ObjectSet>
                        <Object>know-what to cylinder head design</Object>
                        <Object>know-what to car engine</Object>
                        <Object>know-what to cylinder</Object>
                    </ObjectSet>
                    <Environment> date>=5/20/2007 and date<=10/20/2008 </Environment>
                </Target>
                <Condition>ft111(oil filler cap design) and ft112 (cylinder head design) are being
                          executed</Condition>
                <Effect>permit</Effect>
            </Rule>
            <Rule Type= "rule-kind" Name= "rule-ve11-002" Action= "active">
                <Target>
                    <SubjectSet>
                        <Subject>Company-A</Subject>
                        <Subject>Company-B</Subject>
                        <Subject>Company-C</Subject>
                    </SubjectSet>
                    <ActionSet>
                        <Action>write</Action>
                        <Action>read</Action>
                    </ActionSet>
                    <ObjectSet>
                        <Object>all resources assigned to ft11</Object>
                    </ObjectSet>
                    <Environment> date>=5/20/2007 and date<=10/20/2008</Environment>
                </Target>
                <Condition> anyone of ft111(oil filler cap design), ft112 (cylinder head design) and ft113 (stopper
                          design) are being executed</Condition>
                <Effect>permit</Effect>
            </Rule>
        </RuleSet>
    </VEACP>

    <EACP Company-ID= "em1" Company-name= "Company-A" Responsibility= "oil filler cap design"
        Update-time= "5/16/2007" State= "active">
        <RuleSet>
            <RuleCombiningAlgorithm>permit-overrides</RuleCombiningAlgorithm>
            <Rule Type= "rule-kind" Name= "rule-em1-001" Action= "active">
                <Target>
                    <SubjectSet>
                        <Subject>Company-B</Subject>
                        <Subject>Company-C</Subject>
                    </SubjectSet>
                    <ActionSet>
                        <Action>read</Action>
                    </ActionSet>
                    <ObjectSet>
                        <Object>R&D knowledge related to oil filler cap design</Object>
                    </ObjectSet>
                    <Environment> date>=11/20/2007 and date<=10/20/2008</Environment>
                </Target>
                <Condition>f111 is completed</Condition>
                <Effect>permit</Effect>
            </Rule>
        </RuleSet>
    </EACP>
    . . .

</PACP>
```

REFERENCES

[1] Y.-M. Chen and M.-W. Liang, "Design and implementation of a collaborative engineering information system for allied concurrent engineering," *Int. J. Comput. Integr. Manuf.*, vol. 13, no. 1, pp. 11–30, Jan. 2000.

[2] A. Frenkel, H. Afsarmanesh, C. Garita, and L. O. Hertzberger, "Supporting information access rights and visibility levels in virtual enterprises," in *Proc. 2nd IFIP Work. Conf. Infrastructure Virtual Enterprise*, 2000, pp. 177–192.

[3] J. Ma and M. A. Orgun, "Trust management and trust theory revision," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 3, pp. 451–460, May 2006.

[4] Y. Lu, W. Wang, B. Bhargava, and D. Xu, "Trust-based privacy preservation for peer-to-peer data sharing," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 3, pp. 498–502, May 2006.

[5] E. Turban, D. King, D. Viehland, and J. Lee, *Electronic Commerce: A Managerial Perspective*. Upper Saddle River, NJ: Pearson Educ. Int., 2006.

[6] H. R. Rao and S. J. Upadhyaya, "Special issue on secure knowledge management," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 35, no. 1, p. 185, Jan. 2005.

[7] E. Bertino, L. R. Khan, R. Sandhu, and B. Thuraisingham, "Secure knowledge management: Confidentiality, trust, and privacy," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 3, pp. 429–438, May 2006.

[8] R. Singh and A. F. Salam, "Semantic information assurance for secure distributed knowledge management: A business process perspective," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 3, pp. 472–486, May 2006.

[9] R. Au, M. Looi, and P. Ashley, "Automated cross-organizational trust establishment on extranets," in *Proc. Workshop Inf. Technol. Virtual Enterprises*, 2001, pp. 3–11.

[10] T.-Y. Chen, Y.-M. Chen, C.-B. Wang, and H.-C. Chu, "Development of an access control model, system architecture and approaches for information sharing in virtual enterprise," *Comput. Ind.*, vol. 58, no. 1, pp. 57–73, Jan. 2007.

[11] T.-Y. Chen, Y.-M. Chen, H.-C. Chu, C.-B. Wang, and H. Yang, "Secure resource sharing on cross-organization collaboration using a novel trust method," *Robot. Comput.-Integr. Manuf.*, vol. 23, no. 4, pp. 421–435, Aug. 2007.

[12] M. Koch, L. V. Mancini, and F. Parisi-Presicce, *Graph Transformations for the Specification of Access Control Policies*. Amsterdam, The Netherlands: Elsevier science B. V, 2002.

[13] D. F. Ferraiolo, D. R. Kuhn, and R. Chandramouli, *Role-Based Access Control*. Norwood, MA: Artech House, 2003.

[14] S. Oh and S. Park, "Task-role-based access control model," *Inf. Syst.*, vol. 28, no. 6, pp. 533–562, Sep. 2003.

[15] A. Kern, "Advanced features for enterprise-wide role-based access control," in *Proc. Comput. Security Appl. Conf.*, 2002, pp. 333–342.

[16] C. J. Moon, D. H. Park, S. J. Park, and D. K. Baik, "Symmetric RBAC model that takes the separation of duty and role hierarchies into consideration," *Comput. Security*, vol. 23, no. 2, pp. 126–136, Mar. 2004.

[17] D. Shin, G. J. Ahn, and J. S. Park, "An application of directory service markup language (DSML) for role-based access control (RBAC)," in *Proc. Comput. Softw. Appl. Conf.*, 2002, pp. 934–939.

[18] K. Furst, T. Schmidt, and G. Wippel, "Managing access in extended enterprise networks," *IEEE Internet Comput.*, vol. 6, no. 5, pp. 67–74, Sep./Oct. 2002.

[19] J. Bacon, K. Moody, and W. Yao, "A model of OASIS role-based access control and its support for active security," *ACM Trans. Inf. Syst. Security*, vol. 5, no. 4, pp. 492–540, Nov. 2002.

[20] F. T. Alotaiby and J. X. Chen, "A model for team-based access control (TMAC)," in *Proc. Inf. Technol.: Coding Comput.*, 2004, vol. 1, pp. 450–454.

[21] J. J. Kanet, W. Faisst, and P. Mertens, "Application of information technology to a virtual enterprise broker: The case of Bill Epstein," *Int. J. Prod. Econ.*, vol. 62, no. 1, pp. 23–32, May 1999.

[22] E. K. Ouzounis, "An agent-based platform for the management of dynamic virtual enterprises," Ph.D. dissertation, Tech. Univ. Berlin, Berlin, Germany, 2001.

[23] J. S. Park and J. Hwang, "RBAC for collaborative environments: Role-based access control for collaborative enterprise in peer-to-peer computing environments," in *Proc. 8th ACM Symp. Access Control Models Technol.*, 2003, pp. 93–99.

[24] N. Mezzetti, "Towards a model for trust relationships in virtual enterprises," in *Proc. 14th Int. Workshop Database Expert Syst. Appl.*, 2003, pp. 420–424.

[25] T. J. Smith and L. Ramakrishnan, "Joint policy management and auditing in virtual organizations," in *Proc. 4th Int. Workshop Grid Comput.*, 2003, pp. 117–124.

[26] G. Steinke and R. Leamon, "Information security issues facing virtual enterprises," in *Proc. Int. Conf. Eng. Technol. Manage.*, 1996, pp. 641–644.

[27] H. Zhu, "Some issues of role-based collaboration," in *Proc. Can. Conf. Elect. Comput. Eng.*, 2003, vol. 2, pp. 687–690.

[28] G. Kolaczek, "Specification and verification of constraints in role based access control," in *Proc. 12th IEEE Int. Workshops Enabling Technol.: Infrastructure Collaborative Enterprise*, 2003, pp. 190–195.

[29] J. Luo and D. He, "Research on object-oriented role-based access control model," in *Proc. 4th Int. Conf. Parallel Distrib. Comput., Appl. Technol.*, 2003, pp. 132–135.

[30] J. D. Moffett, "Control principles and role hierarchies," in *Proc. 3rd ACM Workshop Role-Based Access Control*, 1998, pp. 63–69.

[31] S. Osborn, "Integrating role graphs: A tool for security integration," *Data Knowl. Eng.*, vol. 43, no. 3, pp. 317–333, Dec. 2002.

[32] M. Lorch, S. Proctor, R. Lepro, D. Kafura, and S. Shah, "First experiences using XACML for access control in distributed systems," in *Proc. ACM Workshop XML Security*, 2003, pp. 25–37.

[33] S. Barker and P. J. Stuckey, "Flexible access control policy specification with constraint logic programming," *ACM Trans. Inf. Syst. Security*, vol. 6, no. 4, pp. 501–546, Nov. 2003.

[34] M. Coetzee and J. H. P. Eloff, "Virtual enterprise access control requirements," in *Proc. Annu. Res. Conf. South Afr. Inst. Comput. Scientists Inf. Technologists Enablement Through Technol.*, 2003, pp. 285–294.

[35] S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian, "Flexible support for multiple access control policies," *ACM Trans. Database Syst.*, vol. 26, no. 2, pp. 214–260, Jun. 2001.

[36] A. Belokosztolszki and K. Moody, "Meta-policies for distributed role-based access control systems," in *Proc. 3rd Int. Workshop Policies Distrib. Syst. Netw.*, 2002, pp. 106–115.

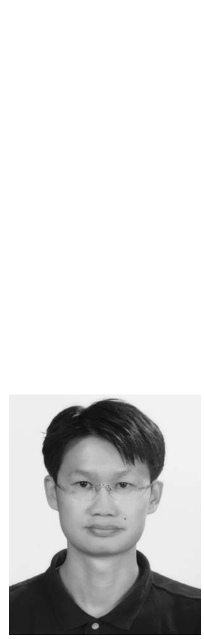
[37] S. Hada and M. Kudo, "XML document security based on provisional authorization," in *Proc. 7th ACM Conf. Comput. Commun. Security*, 2000, pp. 87–96.

[38] G. Boella and L. van der Torre, "Security policies for sharing knowledge in virtual communities," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 3, pp. 439–450, May 2006.

[39] A. Kern, A. Schaad, and J. Moffett, "Enterprise role administration: An administration concept for the enterprise role-based access control model," in *Proc. 8th ACM Symp. Access Control Models Technol.*, 2003, pp. 3–11.

[40] R. A. Botha and J. H. P. Eloff, "Designing role hierarchies for access control in workflow systems," in *Proc. 25th Annu. Int. Comput. Softw. Appl. Conf.*, 2001, pp. 117–122.

[41] F. Dridi, B. Muschall, and G. Pernul, "Administration of an RBAC system," in *Proc. 37th Annu. Hawaii Int. Conf. Syst. Sci.*, 2004, pp. 187–192.



Tsung-Yi Chen received the B.S. degree from Providence University, Taichung, Taiwan, R.O.C., in 1996, and the M.S. and Ph.D. degrees from the Institute of Manufacturing Engineering, National Cheng Kung University, Tainan, Taiwan, in 2001 and 2006, respectively.

He is currently an Assistant Professor with the Department of Electronic Commerce Management, Nanhua University, Chia-Yi, Taiwan. His research interests include virtual enterprise, e-commerce and knowledge commerce, enterprise and information integration, access control, and knowledge sharing.

Yuh-Min Chen received the B.S. and M.S. degrees from National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1981 and 1983, respectively, and the Ph.D. degree in industrial and systems engineering from Ohio State University, Columbus, in 1991.

He is currently a Professor and the Director of the Institute of Manufacturing Engineering, National Cheng Kung University, Tainan, Taiwan. Before joining the faculty of the Institute of Manufacturing Engineering in 1994, he was a Research Engineer with the Structural Dynamics Research Corporation, Plano, TX, for three years. His current research interests include enterprise integration, engineering data and knowledge management, computer-aided concurrent engineering, and manufacturing information systems.

Chin-Bin Wang received the B.S. degree from National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1981, the M.S. degree from the University of Southern California, Los Angeles, in 1985, and the Ph.D. degree in computer science from the City University of New York, New York, in 1995.

He is currently a Professor and the Chairman of the Department of Electronic Commerce Management, Nanhua University, Chia-Yi, Taiwan. His research interests include data mining, network management, engineering data and knowledge management, and system integration.