

# 一種以表格驅動為基礎的中醫健保申報資訊系統設計

## The Design of a Traditional Herbal Health Insurance Report

### System with Table Driven Skills

吳致元

東吳大學資訊科學系

ms9013@cis.scu.edu.tw

柯淑津

東吳大學資訊科學系

ksj@cis.scu.edu.tw

#### 摘要

表格驅動是一種非常實用的資訊系統開發技巧。此種技巧係嘗試利用數個表格及其中的資料內容來描述資訊系統的主要機能，而程式部份即以讀取各個資料內容來判斷如何進行既定的相關處理。因此，引用這種技巧所開發資訊系統的穩定度以及可維護度都高。另在政府大力推廣全民健保相關制度下，健保申報相關資訊系統已成為各醫院診所之必備工具；其主要功能即為協助院所順利進行電子媒體申報作業。由於健保申報計算規則常有變動，若資訊系統也必須跟著經常作程式修改、測試，則工作量就顯得非常不經濟。本文針對此一問題提出一種以表格驅動為基礎的設計，以中醫診所之健保申報作業為實例，並以數個申報規則變動的例子來驗證其效益性。

關鍵詞：資訊系統設計、表格驅動、健保媒體申報、中醫、資訊管理系統。

#### 一、前言

在政府大力推廣健保相關制度下，健保申報作業相關資訊系統已成為各中醫院所之必備工具。就目前現有的醫療資訊系統而言，其健保申報部分一般都採用程式基準(code based)的設計方式，也就是依功能予以分析後

建立相對應的模組，再分別撰寫程式。此種設計方式雖可達成需求，但當健保申報相關計算規則有變更時，就要修改程式才能因應處理。也就是說，假使健保規則有所更改時，院所必須等到系統開發者將程式修改完成，並經過測試、更新系統版本後才得進行申報作業。這對於院所來說是相當不方便或不切實際的，因為健保申報是一項具有時效性的作業，一旦超過時限，院所便可能無法進行申報。

鑒於上述之問題，我們提出了以表格驅動(table driven)為基礎的設計方式，亦即同時採用表格基準(table based)與規則基準(rule-based)[14]兩種技巧，來代替一般的程式基準設計方式。此處所謂的表格基準指的是將健保申報所需資料與參數存放在表格中，而規則基準指的則是將原本利用程式描述的健保申報規則改成利用某些敘述存放在表格中。設計目標是日後當健保申報規則有所改變時，只需修改表格中的資料或規則內容即可，而不需要更改程式碼。這也就表示，即使是不會寫程式的院所一般人員，也可以使用者立場自行修改表格中的資料或規則敘述來因應申報規則的改變。

表格驅動設計方式不論對於資訊系統開發者或中醫院所人員，都會有正面的意義。理論上，程式只要修改，相關的測試就必須跟著

進行；若能避免修改程式而僅以變更表格內容就可達成目的，那節省的程式修改以及測試人力、時間資訊都是可觀的，況且程式的原有品質也是可以保證的。以下各節先分別就軟體工程程序說明系統需求概述、系統架構設計、細部設計；接著就實作經驗進行系統設計之分析評量，最後給予結論。

## 二、系統需求概述

健保申報資訊系統主要是針對健保媒體申報作業所設計，而所謂媒體申報是指院所將原有的健保申報紙張文件以健保局所規定的電子格式存放在儲存媒體中，再適時傳送至健保局進行申報。主要的系統功能需求就是要能正確地依格式產生相關申報資料，非功能需求則包括開發及使用實際環境面的考慮，以下分為四項簡要敘述之：

### (一) 健保媒體申報系統需求概述

健保媒體申報是指將當月院所看診病歷記錄資料以電子檔的方式傳送給健保局進行申報以便請領補助款。以中醫院所為例，共有四個電子檔需要傳送，分別為門診費用申請總表[1]、門診處方及治療[2]、門診處方及治療醫令明細[3] [4]、中醫診察費申報表[5]。一般而言，要產生這些電子檔可能必須先具備以下資料庫與相關資訊系統：

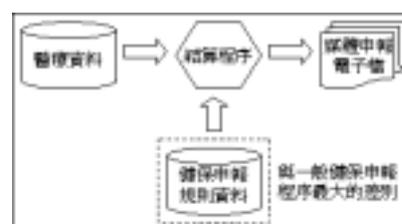
1. 醫療資料庫：記錄院所、醫師、病患、病患就診等資料，依據這些資料健保媒體申報系統才能進行結算產生申報電子檔。
2. 醫療資訊系統：此系統提供院所醫療操作介面，協助院所在進行掛號、看診等過程時，將原先的紙張資料改以電子的方式記錄在醫療資料庫中，以便日後彙集進行媒體申報[6]。
3. 健保媒體申報資料庫：此資料庫目的在於存放健保申報所需的參數與申報規則

等資料。

4. 健保媒體申報系統：此系統提供院所健保媒體申報操作介面，依照健保申報規則將醫療資料庫的資料轉換成媒體申報電子檔。

### (二) 採用表格驅動方式設計

系統架構必須利用表格基準與規則基準兩種技巧，此與一般程式基準設計方式最大的差異在於要將健保申報規則由原本撰寫在程式中改由存放在資料庫中(如圖 1 虛線區域所示)。當健保申報規則有所改變時，使用者只要修改存放在健保申報規則資料庫中的規則，系統即可馬上因應正確運作。



圖表 1、具表格驅動之健保媒體申報系統

### (三) 使用者介面採取網頁操作

在使用者操作介面上採用網頁瀏覽器的方式呈現，需求原因如下：

1. 資訊應用趨勢：在網際網路資訊時代，所有的使用者都會利用瀏覽器上網進行交易、搜尋資料，也就是網頁服務(web service)運作模式。
2. 沒有時間地域範圍的限制：只要透過網路與瀏覽器，使用者不論在何時何地都可進行操作。
3. 使用者端零安裝：與傳統應用系統不同，只要電腦有安裝瀏覽器與網路便可使用系統，使用者端不需再另行安裝其他程式。

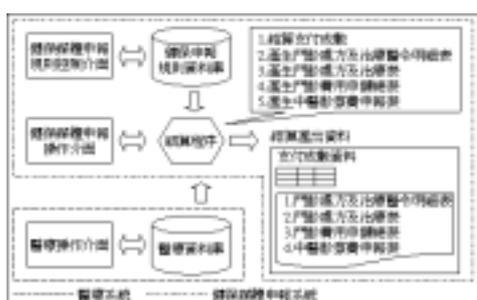
### (四) 使用 JAVA 語言開發

在系統開發上採用 JAVA 語言[8][12]，主

要原因是其具跨平台特性。以此語言開發之資訊系統可以適用於微軟 Windows 作業系統及 Linux 作業系統。

### 三、系統架構設計

整個系統架構分成醫療系統與健保媒體申報系統兩大部分(如圖 2 所示)，此兩部分說明如下：



圖表 2、中醫系統架構圖

1. 醫療系統：包含醫療操作畫面與醫療資料庫兩部分，主要目的在於提供院所整個醫療流程所需的操作介面，並將操作過程中所填寫的資料存入醫療資料庫中。其中醫療流程包含掛號、看診、診療與批價領藥等過程。
2. 健保媒體申報系統：包含了健保媒體申報規則控制與健保媒體申報兩部分。健保媒體申報規則控制部分，提供健保規則資料表格與參數資料的新增、修改與刪除等功能介面給使用者；其目的是讓使用者能簡單快速的建立健保申報規則。而健保媒體申報部分，則提供使用者產生健保媒體申報電子檔、檢驗醫療資料庫資料是否完整與下載健保媒體申報電子檔等功能。其中健保規則資料表格與參數資料總共有七個如下所示：

- ◆ 結算表目錄 (Target Form List Table)：其記錄需要被結算的結算表清單與每個結算表所存放的位置，並記載是否需篩選產生「原始資料

主鍵陣列」。

- ◆ 結算表欄位定義表 (Target Form Field Definition Table)：此表格定義各個結算表中各欄位的資料型態、資料取得方式與資料格式等資訊。也就是說，結算表所得到的結果與呈現型態都是由此定義表所訂定的。也因此，此定義表必須參照其他定義表與陣列來進行結算。
- ◆ 資料篩選定義表 (DB SQL Table)：此表格目的在於輔助「結算表欄位定義表」對原始資料庫內的資料進行查詢篩選，並找出所需的資料。
- ◆ 邏輯條件定義表 (Condition Table)：輔助「結算欄位定義表」進行邏輯判斷處理。經由判斷找出符合的執行指令，並執行求得所要的資料。
- ◆ 原始資料主鍵陣列 (PK Array)：存放每筆結算記錄重要對應指標資訊。以健保申報來說，一筆結算記錄即代表一筆病歷記錄。而此病歷相關資訊就是我們所要的結算資料。
- ◆ 共用常數資料陣列 (Constant Variable Array)：存放不會改變的常數資料，其中包含使用者設定的系統預設值 (User profile)，如院所名稱與使用者在介面上所輸入項目暫存表格 (User input)，如申報年月。
- ◆ 共用變數資料陣列 (Global variables Array)：當「結算表目錄」未指定「結算表欄位定義表」處理後，要將結果放置至哪個暫存表格時，則系統會將這些處理的資料存放在「共用變數陣列」中以供使用。

我們所提出的系統架構之所以會區分成

醫療系統與健保媒體申報系統兩部分，是基於假使院所已有醫療系統的考量，那我們要導入具表格驅動的健保媒體申報系統的話，只需將院所既有的醫療資料庫轉換成我們所需的醫療資料庫結構即可馬上導入使用。這也就是說，不管院所原先醫療資料庫結構為何，我們都可透過醫療資料庫的轉換而將我們所提出系統架構導入。

## 四、 細部設計

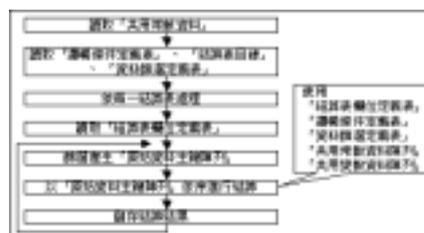
### (一) 健保媒體申報之結算程序說明

在圖 2 中所提到的結算程序，以中醫為例有以下七個步驟(如圖 3 所示)：

1. 讀取「共用常數資料」：讀取「共用常數資料」並存放在記憶體中以供結算使用。
2. 讀取「結算表目錄」、「資料篩選定義表」與「邏輯條件定義表」：這些是健保申報資料取得與規則定義表格。同樣的我們將這些資料讀取出並存放在記憶體中。
3. 依每一結算表處理：系統依照步驟 2 中所下載的「結算表目錄」一一進行結算處理，每一次執行的結果便會產生一張健保媒體申報結算表。
4. 讀取「結算表欄位定義表」：讀取「結算表欄位定義表」並將這些資料存放在記憶體中以供結算使用。
5. 篩選產生「原始資料主鍵陣列」：此步驟我們依使用者輸入的結算條件篩選出這些重要的指標資訊，並將這些資訊存放到記憶體。如病歷編號等資料，對健保申報而言一筆申報記錄即代表一次就診的病歷記錄，而這筆病歷的相關資訊就是我們所要結算處理的資料，故我們可以藉由病歷編號得到病歷等相關資料。
6. 以「原始資料主鍵陣列」依序進行結算：在此步驟，我們依照步驟 5 所篩選出來的「原始資料主鍵陣列」進行結算處理。

過程中會使用到「結算表欄位定義表」、「邏輯條件定義表」、「資料篩選定義表」、「共用常數資料」與「共用變數資料」等資料。

7. 儲存結算結果：最後將步驟 6 所求得的結算結果儲存到指定的位置。



圖表 3、結算程序流程圖

### (一) 表格驅動之規則定義表格說明

接著我們針對上述結算程序中所提到的「結算表目錄」、「結算表欄位定義表」、「資料篩選定義表」、「邏輯條件定義表」四個表格作結構欄位的說明。

GroupID	SaveTable	PKArray	TableOrder
1	Null	Null	1
2	1	512	4

GroupID = 1  
 = 不需要PK Array Data  
 = 最後處理之資料存於Global Variable中  
 GroupID = 4  
 = 需要PK Array Data  
 = 最後處理之資料存於table中

FieldID	FieldName	FieldType	FieldOrder
1	Field1	Integer	1
2	Field2	String	2
3	Field3	Boolean	3
4	Field4	Integer	4

圖表 4、結算表目錄 圖表 5、結算表欄位定義表

1. 結算表目錄(Target Form List Table): 如圖 4 所示，其欄位定義如下：
  - GroupID：結算表之檔案編號，此編號與「結算表欄位定義表」的檔案編號是一致的。
  - SaveTable：結算表所存放的位置。如果此欄位的值為 Null，則表示結算後會將結算資料存放在共用變數資料陣列(Global Variable Array)中。
  - PKArray：各結算表進行結算前，事先需篩選產生的「原始資料主鍵陣列」。如果此欄位的值為 Null，則表示不用事先篩選產生。

- DfdtOrder :「結算表目錄」中各結算表之處理順序。
2. 結算表欄位定義表(Target Form Field Definition Table) : 如圖 5 所示,其欄位定義如下 :
- Group ID : 結算表檔案編號與「結算表目錄」的檔案編號是一致的。
  - ID : 結算欄位識別號。
  - Length : 結算欄位之資料長度。
  - Format : 結算欄位之資料格式(如靠左補空白、靠右補空白等)。
  - Type : 欄位資料處理型態。結算程序會依照此欄位所定義的處理型態,進行不同的處理方式。處理方式共有十種如下所示 :
    - i. U : 為共用常數資料處理型態。此種資料型態系統會到「共用常數陣列」找出所對應的值。
    - ii. G : 為共用變數資料處理型態。此種資料型態系統會到「共用變數陣列」找出所對應的值。
    - iii. A : 為流水號資料型態。此種為自動編號的資料會根據上次之流水號自動加 1。
    - iv. X : 為補空白資料處理型態。
    - v. C : 為邏輯判斷資料處理型態。此種資料型態必須參照「邏輯條件定義表」找出符合的執行指令。之後執行指令求得結果。
    - vi. M : 為數學運算資料處理型態。
    - vii. S : 為從原始資料庫擷取資料處理型態。此資料型態必須參照「資料篩選定義表」找出符合的資料篩選條件,並將其解析成所需的 SQL 指令,再依照此 SQL 指令到原始資料庫搜尋所要的資料。
    - viii. L : 為執行迴圈(Loop)的標示符號。此欄位目的在於標示需要透過迴圈處理的資料。以健保申報來說,針灸治療可有六次療程,所以對於同一筆病歷記錄可能會有六筆治療記錄,所以我們必須將這六筆記錄結合成一筆資料。迴圈標示符號其記載方式為 X. X. X 三部份,而這三部份分別代表迴圈開始欄位、結束欄位與每次處理筆數。(例如 9.18.5 則表示迴圈處理由欄位 9 開始到欄位 18 結束,每次處理以 5 筆記錄為基準)
    - ix. W : 為連結字串處理型態,表示欄位由多個字串所組成的情況。
    - x. D : 為日期資料處理型態,處理日期類型的資料欄位。
  - Operation : 欄位資料取得方式。上述 Type 主要是在定義欄位的資料型態,功用為告知系統該如何處理此欄位。至於真正處理過程是由此欄位定義的,此欄位會告訴系統資料的來源處。
  - Order :「結算表欄位定義表」中各欄位的處理順序。
  - 接下來我們以圖 5 為例,來對「結算表欄位定義表」內所記錄的資訊進行說明 :
    - i. ID = D1 : Type 為 U 表示為共用常數資料處理型態,此時系統便會從「共用變數陣列」中尋找陣列識別號為 P2 的資料。
    - ii. ID = D2 : Type 為 S,表示從原始資料庫擷取的處理型態,系統根據 Operation 欄位內的索引到「資料篩選定義表」中搜尋資料篩選識別碼為 S1 的篩選條件,接著分析此條件並組合成可用的 SQL 指令。最後根據此 SQL 指令到原始資料庫中找出所需的資料。

- iii. ID = D6:Type 為 A 表示是流水號資料型態，系統會根據上一筆流水號加 1 作為此筆資料之流水號
- iv. ID = D11 : Type 為 C 表示是邏輯判斷處理型態，系統會根據 Operation 欄位到「邏輯條件定義表」中找尋邏輯條件識別號為 C1 的邏輯判斷條件，並對此條件進行分析並找出符合的執行指令，最後執行此指令求出所要的值。
- v. ID = D18 : Type 為 M 表示是數學運算處理型態，系統會直接執行 Operation 欄位中所記載的指令，此經解析為  $D18=D16*D17$ 。
- vi. ID 為空白: Type 為 L 表示是執行迴圈的標示符號。其 Operation 為 ( 9.18.5 ) 代表整個迴圈從欄位 ID 為 D9 開始到欄位 ID 為 D18 為止，每次執行 5 次。故在此結算表中，每一筆結算記錄會記載欄位 D1 至 D8 各 1 次，而欄位 D9 到 D18 各 5 次。

圖表 6、資料篩選定義表 圖表 7、邏輯條件定義表

3. 資料篩選定義表(DB SQL Table)：如圖 6 所示，其欄位定義如下：
  - ID：資料篩選識別號。
  - DB：資料來源之原始資料庫名稱。
  - SQL Table：資料來源之原始資料表格名稱。
  - ResultSet：搜尋的資料欄位名稱。
  - RetrunField：回傳的資料欄位名稱。
  - Relation：資料篩選條件。
4. 邏輯條件定義表(Condition Table): 如圖 7

所示，其欄位定義如下：

- ID：邏輯判斷識別號。
- Condition：邏輯判斷條件。
- conThen：成立時所要執行的指令。
- conEles：不成立時所執行的指令。

我們設計的健保申報規則資料庫結構、規則資料與參數資料，是依據中醫健保媒體申報所需的規定進行分析，並搭配一些資料結構[10]之概念所制定的。經過我們的分析發現中醫健保媒體申報中的規定所需資料取得方式有從申報介面取得、從醫療庫取得、從常數資料取得、經過四則運算取得、經過判斷式判別後取得等方式，而我們所設計的表格驅動架構足以應付這些資料取得之方式。另外，我們所設計的系統採用的資料庫為 MYSQL[9][13]，之所以採用 MYSQL 最主要的原因是考量其為免費軟體可以減少醫療院所之成本。

## 五、系統設計之分析評量

我們針對一般程式基準設計架構與我們提出表格驅動設計架構分別開發的系統進行以下四項分析比較[11]：

1. 正確性(Validity)：此項為最基本的評估。在運算與資料處理規則建立無誤下，表格驅動設計架構所結算得到的結果與一般程式基準設計架構所得到的相同。
2. 效能(Performance)：在結算速度上，以院所每個月 1000 筆資料來做比較，表格驅動設計架構平均需要 1 分 28 秒，其速度比一般程式基準設計架構的 1 分 30 秒還要快上 2 秒。(硬體及軟體基台相同，均以一般 X86 級電腦)
3. 靈活度(Flexibility)：我們針對健保局所提出的十項修改申報規則進行因應健保規則修改之靈活度測試，修改規則包含了「中醫醫療院所專任醫師每月看診日平均針灸傷科及脫臼整復合計申報量三十

一至四十五人次區間者申報編號 B42(針灸治療，未開內服藥)，B54(傷科治療，未開內服藥)，B63(脫臼整復，同療程複診未開內服藥)者九折支付，四十六至六十人次區間者申報編號 B42，B54，B63者五折支付。」等十項[7]。

針對這十項健保規則更改，表格驅動設計架構之因應處理狀況如表 1 所示，一般程式基準設計架構之因應處理狀況如表 2 所示，由此兩個統計表我們可以整理得到處理效能差異比較表如表 3 所示，其說明如下：

表格 1、表格驅動處理統計 表格 2、程式基準處理統計

項目	修改時間	測試時間	修改行數
1	0.5小時	0.5小時	0行
2	1小時	0.5小時	0行
3	0.5小時	0.5小時	0行
4	0.5小時	0.5小時	0行
5	0.5小時	0.5小時	0行
6	0.5小時	0.5小時	0行
7	0.5小時	0.5小時	0行
8	0.5小時	0.5小時	0行
9	0.5小時	0.5小時	0行
10	0.5小時	0.5小時	0行
總計	1.5小時	1小時	0行
平均	0.55小時	0.5小時	0行

項目	修改時間	測試時間	修改行數
1	4.5小時	2小時	26行
2	8小時	3小時	13行
3	3小時	1.5小時	16行
4	3小時	1.5小時	15行
5	3小時	1.5小時	16行
6	4小時	2小時	20行
7	4小時	2小時	20行
8	4小時	2小時	20行
9	3.5小時	1.5小時	19行
10	3.5小時	1.5小時	19行
總計	42.5小時	18.5小時	222行
平均	4.25小時	1.85小時	22.2行

- [1] 修改時間：表格驅動設計架構平均每項修改比一般程式基準設計架構快 3.7 個小時，所以上述十項修改總共快了 37 個小時，也就是 4.6 個工作天。
- [2] 測試時間：表格驅動設計架構平均每項測試比一般程式基準設計架構快 1.35 個小時，所以上述十項測試總共快了 13.5 個小時，也就是 1.7 個工作天。
- [3] 修改行數：表格驅動設計架構不需修改任何程式，而一般程式基準設計架構平均每個規則需修改 22.2 行程式，所以上述十項修改總共需修改 220 行程式。

由上述的比較數據我們得知，表格驅動設計架構與一般程式基準設計架構的維護時間成本花費(修改時間 + 測試時間)為 1 : 6.3，也就是說，健保局修改的規則越多，則所需的維護時間成本也會相差越大。

表格 3 表格驅動與程式基準設計架構之處理效能差異比較

	修改時間	測試時間	修改行數
表格驅動	0.55小時/1個	0.5小時/1個	0行/1個
程式基準	4.25小時/1個	1.85小時/1個	22.2行/1個

4. 重覆使用性(reusable)[15][16]：就系統開發者而言，在程式碼不需修改的情況下系統可重覆使用為最有利。我們的系統在面對不同醫療院所時，只需修改表格中的資料即可，而一般的程式基準設計架構則需修改程式才足以應付。

就上述的評量比較，正確性與效能上的比較兩種設計架構差異不大，但在靈活度比較上，表格驅動設計架構遠比一般程式基準設計架構要來的好，從比較數據上我們可以發現當健保規則變更次數越多的話，則對一般程式基準設計架構越不利。另外在重覆使用性上，表格驅動設計架構也比一般程式基準設計架構更加便利。

## 六、 結論

我們所提出的具表格驅動之健保媒體申報系統架構，在使用者介面上採用一般使用者最熟悉的網頁瀏覽器操作方式，對於使用者來說得以迅速適應系統操作，而且院所只要伺服器與網路架設完成後，使用者便可在院所的各個定點直接作業，不需再額外安裝任何應用程式。在開發語言上採用 JAVA 來開發，所以不論是何種作業系統都可使用。最重要的是我們所採用的是表格驅動設計架構，此架構不論在正確性或是效能比較上都不比一般程式基準設計架構來的遜色，尤其在靈活度比較上，其因應健保申報規則更改時的能力遠遠超過一般程式基準設計架構。

在政府大力推行全民健保的情形下，院所醫療流程勢必一定要資訊化，如此才得以進

行健保媒體申報作業。因此在因應健保媒體申報規則更改就顯得格外的重要，而我們所提出以表格驅動為基礎的中醫健保申報資訊系統設計，便可幫助院所減少許多此類的困擾。

### 誌謝

在本系統設計開發期間，感謝翔威國際股份有限公司劉龍龍博士與游俊樺博士，在表格驅動架構設計與系統開發上給予相當多的觀念與指導。

### 參考文獻

- [1] 吳梅芬, 門診醫療服務點數申報總表媒體申報格式及填表說明.doc, 中央健康保險局, 五月, 2002.
- [2] 吳梅芬, 門診醫療服務點數清單媒體申報格式及填表說明.doc, 中央健康保險局, 五月, 2002.
- [3] 吳梅芬, 門診醫療服務醫令清單媒體申報格式及填表說明.doc, 中央健康保險局, 五月, 2002.
- [4] 吳梅芬, 全民健康保險特約中醫院所申報針灸、傷科、脫臼整復合理量計算範例.doc, 中央健康保險局, 五月, 2002.
- [5] 吳梅芬, 全民健康保險醫療費用支付標準(第四部中醫).doc, 中央健康保險局, 五月, 2002.
- [6] 林介文, “The Implementation of a Web-Based From Management System”, 東吳大學商學院資訊科學系碩士論文, 2001.
- [7] 健保醫療費用申報規則參考文件, [http://www.nhi.gov.tw/02hospital/hospital\\_4.htm](http://www.nhi.gov.tw/02hospital/hospital_4.htm)
- [8] 楊洸, 沈見男, *Java Server Page 程式設計實務*, 學貫行銷股份有限公司,

- 10月, 2000.
- [9] Randy Jay Yarger, George Reese, Tim King, *MySQL & mSQL 管理與設計*, 美商歐萊禮股份有限公司台灣分公司, 九月, 2000.
- [10] Ellis Horowitz, Staraj Sahni, and Susan Anderson-Freed, *Fundamental of Data Structure in C*, Computer Science Press, 1993.
- [11] Eric J. Braude, *Software engineering an object-oriented perspective*, John Wiley & Sons, Inc. 2001.
- [12] Java Rule Engine API 之技術文件說明, <http://jcp.org/en/jsr/detail?id=94>
- [13] Mysql Database SQL Command Online Support, [http://www.mysql.com/doc/en/Function\\_Index.html](http://www.mysql.com/doc/en/Function_Index.html)
- [14] P.G. Chander, T. Redhakrishnan, R.Shinghal, “Using paths to detect redundancy in rule base”, *11th Conference on Artificial Intelligence for Applications*, pp133-139, February 1995.
- [15] Rine, D. C, Nada, N., “An empirical study of software reuse reference model”, *Information and Software Technology*, Volume:42, Issue:1, pp.47-65, January 1, 2000.
- [16] Rine David C., Sonnemann, Robert M., “Investment in reusable software. A study of software reuse investment success factor”, *Journal of System and Software* Volume:59, Issue:3, pp343-354, December 15, 2001.