

Viewdb：一個可應用於WWW與資料庫整合之視覺化系統
Viewdb：A Visualization System for WWW and Database Integration

竇其仁 葉良偉 范大偉 洪世勳
Chyi-Ren Dow Liang-Wei Yeh Da-Wei Fan Shi-Xun Hong

逢甲大學資訊工程學系

Department of Information Engineering, Feng Chia University
{crdow,lweyh,dwfan,pitos}@iecs.fcu.edu.tw

摘要

在這篇論文中，我們介紹一個 WWW 與資料庫整合的視覺化系統，本系統除了可將使用者文字型態問卷自動轉換成 HTML Form 問卷，並且對於簡單的 HTML Form 以及整合資料庫的 Form 自動產生相對應的 CGI 程式。我們的系統也提供了一個以視覺化圖形的方式代替傳統撰寫 SQL 命令的 Visual Query Editor 來自動產生 SQL 命令，並將其 SQL 部份整合至自動產生的 CGI 程式中。

Abstract

This paper presents a visualization system for WWW and database integration. The system supports the automatic translation of queries from plain text to HTML/FORM format and the automatic generation of CGI programs for not only simple HTML/FORMs but also FORMs that integrate with databases. Our system provides a visual query editor that is easy to use and the corresponding SQL query embedded in the CGI program can be generated automatically.

一、緒論

網際網路(Internet)的迅速發展，改變了人們處理資訊以及追求知識的觀念與方法。透過網際網路工具，如 ftp、gopher、mail、news、bbs 以及目前最熱門的 WWW(World-Wide Web)等，人們可以比以前更有效率的和遠端電腦進行資料的傳遞接收。另一方面，我們也可以透過網際網路吸收世界各地的最新訊息。對一般的使用者而言，這幾年 WWW 迅速的整合原本的 gopher、ftp 等系統，快的讓許多仍在純文字模式的使用者有點措手不及；再加上 HTML 語法迅速更新版本，且兩大瀏覽器(Netscape 的 Navigator 與 Microsoft 的 Internet Explorer)彼此在 HTML 語法上的解釋有一些出入。因為這些情形，讓許多使

用者對 HTML 語法有一些學習上的障礙。因此我們特別針對一般最常用的 HTML 表格格式做整合，讓使用者可以很方便的利用一般文字編輯的常見問卷，經過我們的轉換程式立刻成為 WWW 上 HTML Form 問卷型式，使您的問卷廣泛的應用在 Internet 領域，而完全不需要懂得 HTML 語法。

此外資料庫在電腦資料處理上長久以來一直扮演著不可或缺的角色。從 1980 年代後期，主從式(client/server)架構成為網路系統的主流之後，資料庫便扮演著伺服器的角色，負責提供資料給各個客戶(client)使用。儘管 WWW 具備了許多的優點，不過美中不足的是 WWW 並未提供資料庫做為資料儲存的設備。所以在面對 WWW 如此快速發展的情形下，WWW 勢必要結合資料庫來處理日益龐大的各種資料。況且，重新開發專為 WWW 所設計的資料庫似乎有點本末倒置，而最佳的方案當然是設法將 WWW 連上目前現有的資料庫系統，這也比較能發揮 WWW 以及資料庫都是主從式架構的特點了。

要延伸 WWW 的功能，CGI 可以稱的上是最普遍的選擇了。尤其是在 WWW 整合資料庫系統方面，目前可以採行的方法有許多種 [14]。雖然利用 ODBC(Open DataBase Connectivity) 或是 JDBC(Java DataBase Connectivity) API 的方式是目前較新的技術，不過 ODBC 目前只能在 Windows 平台上執行，加上 ODBC 在連接資料庫的效率上一直為人詬病。而 JDBC 因為是配合 Java 所發展出來類似 ODBC 的 API，所以才一出現便引起一陣鑽研的熱潮，不過採用 JDBC 的方法只能利用 Java 語言來撰寫程式，再加上目前雖然許多資料庫的廠商都聲稱將支援 JDBC，但是這畢竟還是需要一段時間來發展的。因此整體說來，CGI 具有簡單(simplicity)以及開放(openness)優點，目前利用 CGI 來整合 WWW 以及資料庫系統，仍然是最為普遍的選擇。

從以上的敘述，我們已經了解到 CGI 處理 Form 以及 WWW 整合資料庫系統的重要

性了。不過撰寫 CGI 程式這項工作，即使對於一個有經驗的程式設計師而言也不是一件十分容易的工作，更何況是對於一些並沒有具備相當程式設計背景知識的人。因為如果你要設計一個單純處理表單的 CGI 程式，你就得學會像 Perl 這類語言，或是如果你要處理更複雜的整合資料庫的 CGI，那你就必須要同時了解撰寫 Perl 語言，並且你也必須要知道如何下命令去存取資料庫的資料了(例如：SQL 命令)，這就更為的不容易了。你不可能光寫一個 CGI 程式就想處理所有的表單，它必須針對不同的需求來寫，它也可能花費你不少的時間去學習寫一個 Perl 程式。

這篇文章介紹一個可應用於 WWW 與資料庫整合之視覺化系統，Viewdb(a Visualization Environment for WWW and Database Integration)。Viewdb 提供了自動化與視覺化的功能，可用來加速許多需要整合 WWW 與資料庫之應用程式的開發。本文在第二節首先介紹與我們研究相關的一些背景知識。在第三節與第四節介紹我們的系統架構以及系統實作。第五節介紹系統的離型，而最後做一結論以及說明未來的工作。

二、背景

我們這個系統是一個視覺化的程式自動產生器，有許多的研究領域和我們系統有關聯，而主要幾個比較重要的相關研究與技術包括：CGI/Form、WWW 與資料庫整合、程式自動產生器(automatic program generator)以及視覺化語言(visual language)等等。下面便分別對這幾個相關研究與技術做一個簡單介紹：

共通開道介面(Common Gateway Interface, 簡稱 CGI) 乃是指伺服器與伺服器的相關應用資訊溝通管道。CGI 的作用是讓我們在編寫以 HTML 為標準的超文件時，使靜態的文件可以結合外部的程式，而達到使用者要求的相關服務，讓整個服務系統與使用者產生互動的關係。CGI 最為人知，就是表格(Form)的處理。Form 其實是為了提供使用者輸入訊息，藉由 Form 的互動介面，使得使用者與伺服器有良好溝通管道。所以我們可以在 HTML 的文件中，加上一個新的標籤(tag)，使伺服器知道使用者是在呼叫 CGI 程式，而非一般文件的鏈結。

Web 的技術在大型資料庫服務應用程式方面有著一些優點，例如：Web 瀏覽器是跨平台的，並且具備親和力極佳的使用者介面。WWW 的核心技術，HTML、URL 以及 HTTP 都是對外公開的、使用者可以用 HTML Form 文件，以及 Web 瀏覽器做為介面輸入資料來

查詢資料庫的內容、以及從 Web 剛開始發展到現在，這方面技術不僅被廣泛採用並且也有許多功能上的擴充等等。一般來說，WWW 以及資料庫系統整合的基本做法是使用者在客戶端利用 Web 的技術向伺服器發出一存取資料庫的要求，伺服器接到這個要求之後便對它做處理並透過 database gateway 存取資料庫，最後再將查詢結果以 HTML 型式呈現在瀏覽器上。

軟體自動化指的是基於軟體自動處理為目的之工具，一般方法是建立一套使用規格語言(specification language)來做為自動轉換依據的軟體系統。使用這種方式來做為程式碼轉換的發展是發展一轉換產生器(transformation generator)，這種產生器分析轉換規格並產生程式碼〔3〕。

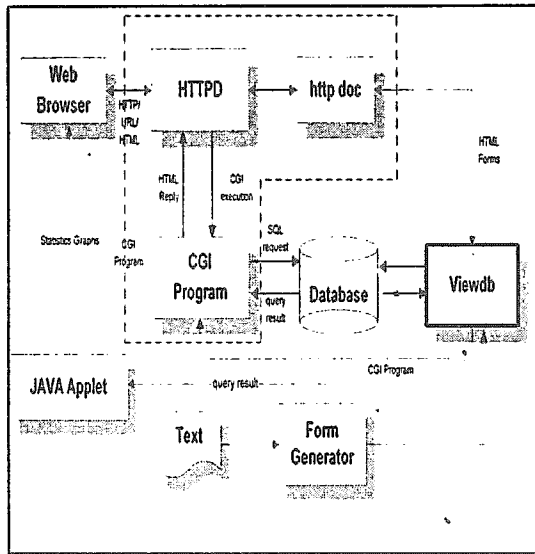
在視覺化語言(visual languages)方面〔2,3,5,8〕，主要是針對視覺化查詢語言。大部份視覺化語言皆以圖形化方式表現，視覺化的目的就是希望幫助程式發展者，以一種不同於文字型式撰寫程式的方式來更有效的開發相關程式。和資料庫有關的視覺化查詢語言〔2,5,9,12〕的例子像是：SIL-icon〔10〕，它提供了一種以 icons 型式來做為資料庫存取方式；Graphical Filter/Flow Representation〔5〕，它是針對資料庫的布林查詢(boolean queries)，提供一種控制流程的圖形方式來做資料庫的布林運算(and、or 以及 not)；另外還有一些像是以圖形(diagram-based)或是表單(form-based)的方式來做為查詢的型態。而這些視覺化語言為了達到以圖形方式運作，它們也必須提供所謂的基本的文法模組(grammar model)來配合圖形化的運作以產生相對應的程式碼。

三、系統架構

我們的系統架構如圖一所示。在圖中我們可看到 Viewdb 在配合了 WWW 與資料庫系統之整體架構，在整體架構中可分成五個部份。第一個部份是客戶的瀏覽器，負責 HTML Form 文件以及查詢結果的展示。第二個部份是資料庫系統，Viewdb 可以處理位於本地(local)或者遠端(remote)的資料庫系統。第三個部份是 Web 伺服器，伺服器設定兩個目錄，分別來處理 HTML 文件以及 CGI 外部執行程式。第四個部份是 Viewdb 系統本身。第五個部份是 Viewdb 的前處理工具 Form Generator。

Viewdb 可以直接輸入 HTML 表格或透過 Form Generator 來將一般書面問卷(各類回函卡、產品調查問卷)，再經由電腦登錄成文字檔問卷(可採用 Word 或一般文字編輯器)，

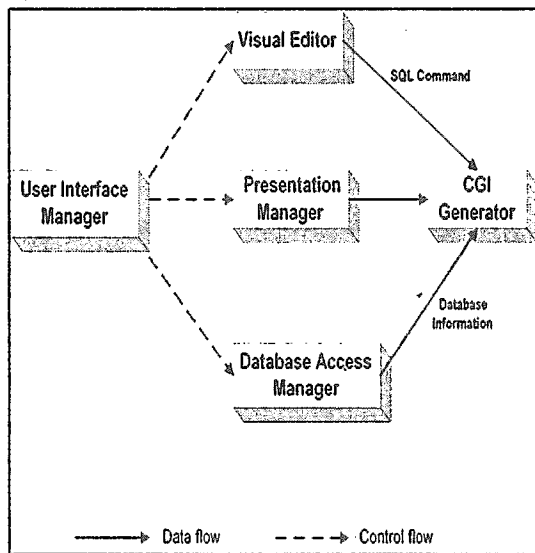
或利用影像處理檔做掃描及處理後，再經由我們的問卷轉換程式轉換成 HTML Form 型



圖一、系統架構圖

式的問卷。如果使用者已經自行設計了 HTML Form 型式的問卷，同樣可以透過我們另一支轉換程式來處理，使得使用者所設計的 HTML Form 問卷符合我們的要求型式。

Viewdb 主要包含了五個模組，如圖二所示。



圖二、Viewdb 系統組成模組

- (1) Visual Editor：根據所定義的 grammar，以視覺化圖形的編輯方式產生資料庫查詢 (SQL) 命令。
- (2) DB Server Access Manager：負責與資料庫 (local 或 remote) 的連結與存取功能的管理。
- (3) CGI Generator：負責 CGI 程式之建立，如果是簡單的 HTML Form 檔案，沒有

關係到資料庫存取，則直接產生 CGI 程式；如果屬於 DB-Form 型式，則必須從 DB Server Access Manager 獲得所開始的 tables 資訊，以及從 Visual Editor 獲得所要執行的 SQL 命令，加以整理最後產生完整的 CGI 程式。

- (4) User Interface Manager：這個部分主要負責整個系統圖形界面的管理，提供系統主要畫面以及各項功能的使用者介面控制管理。對於使用者最關心的查詢資料庫之後的結果，本系統在資料輸出方面，除了有一般文字或 table 的展現回應給 client 端使用者瀏覽外，最特殊的應該是我們另外整合了一個 Java Applet 的統計圖，例如：長條圖、曲線圖、圓餅圖等再配合縱向、橫向與 3D 的立體呈現所組成，使這些數據讓人更一目了然。
- (5) Presentation Manager：對於沒有關係到資料庫的查詢，我們提供 text 與 table 型式展示；若牽涉到資料庫統計結果，我們以 Java Applet 的統計圖方式呈現。

四、系統實作

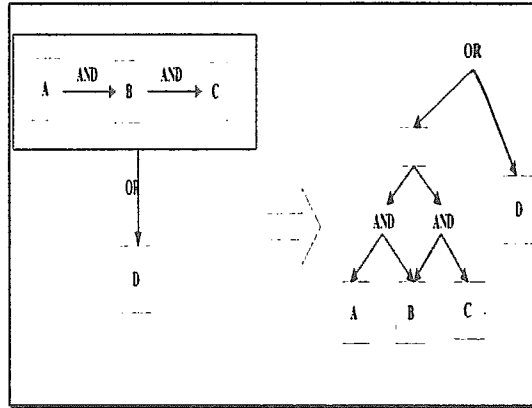
首先在本系統的問卷轉換實作部分，也就是將純文字的問卷轉換成 HTML Form 的型式問卷，我們採用 Perl language 當做我們的開發語言。最主要的原因是 Perl 在 pattern matching 這方面表現的比任何語言還要出色，特別是在我們問卷轉換過程中，常常需要辨識文字中的樣式 (pattern)；或是萃取出一段文字而以新的字串來代替。正因為在 Perl 之中，只要妥善運用這些正規運算 (regular expression)，就可以簡單的表達出所要的樣式。我們在問卷轉換方面就是利用 Perl 的 regular expression 與其組合符號交相並用下，來將一些純文字檔的問卷中關鍵特殊字串轉換成我們所需要表達的 HTML Form 型式的問卷。例如：我們會將文字檔中的問卷屬於 “” 轉換成 HTML Form 的 Checkbox or Radio Button；將 “{ }” 轉換成 HTML Form 的 Select；將屬於 “_____” 轉換成 HTML Form 的 Text or TextArea 的型式。

我們在發展 Viewdb 系統的整個發展環境，本地端為我們發展 Viewdb 系統的環境，發展平台為以 FreeBSD 為作業系統的 Pentium 機器，採用的 Web Server 為 NCSA Server，採用的 local 資料庫為 MiniSQL。遠端平台以 SunOS 為作業系統的 SPAC 5 機器，採用的 Web Server 為 Apache Server，至於遠端所採用的資料庫為 Sybase System 10。使用的程式語言是以 Tcl/Tk 做為整個使用者介面 (user interface) 發展的語言以及自動產生 SQL 命令

的視覺化編輯器(visual editor)的工具，Tcl/Tk 的版本是 Tcl7.5 以及 Tk4.1，另外也使用 Tcl/Tk 的 extensions，包括：TIX、BLT、TclX 以及 Expect [6,7]。

Viewdb 在連結資料庫方面所採取的方法是透過我們所修改撰寫的使用者界面來存取資料庫。而在資料庫的使用方面，分為本地以及遠端的資料庫，在本地的資料庫是 MiniSQL，在遠端的資料庫是 Sybase。至於連結資料庫的使用者界面，針對 Sybase 所採用的是利用 Tcl/Tk 所寫成的 Sybtcl，針對 MiniSQL 所採用的也是利用 Tcl/Tk 所寫成的 msqtlcl 軟體。Sybtcl 以及 msqtlcl 都是可以從網路上自由下載的軟體，我們也對它們做了部份的修改以符合我們系統的需要。透過 Viewdb 功能列上的 Database 功能，使用者可以選擇所要連結的資料庫是本地或遠端的資料庫，系統便有不同的交談畫面回應處理。

視覺化查詢編輯器在 Viewdb 系統中是個重要的部份，整個視覺化查詢編輯器是以 Tcl/Tk 語言所實作而成。在處理 WWW 與資料庫整合的 CGI 程式中有一部份的程式碼是用來處理資料庫的查詢命令，因此 Viewdb 提供一個視覺化查詢編輯器，幫助使用者可以利用圖形物件排列的方式下達查詢命令。在實作視覺化查詢編輯器之初，我們在處理代表資料庫邏輯運算的 edge(代表 AND 或 OR)的做法是先利用按鍵選擇第一個圖形物件，然後再選擇了第二個圖形物件後，系統自動將兩個圖形物件連接起來。不過這種做法在後來處理優先順序的運算時出現了問題，因為在我們的編輯器中優先順序是利用一外框線來表示，而框線可能包圍住了多個不同的圖形物件或 edges，因此如果又有一個圖形物件和框線之間有 edge 相連接，則無法將框線以及包圍住的所有物件當成一個單一的圖形物件，所以採取這種方法在處理運算的優先順序上變的有所限制。最後我們透過 Visual Query Editor 以圖形化的方式產生 Query 圖形，再經由 DAG(Directed Acyclic Graph)轉換的觀念轉換成相對應的 syntax tree [1]，最後產生文字型式的 SQL 命令，如圖三。

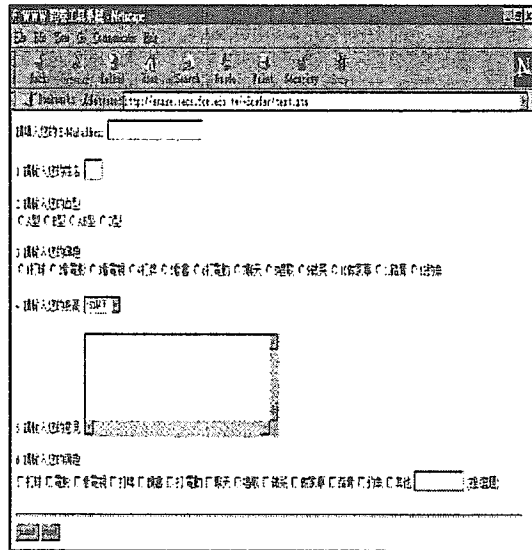


圖三、Visual Query 以 DAG 表示

五、系統雛型

我們的 Form Generator 最主要的工作就是將原本使用者輸入的文字問卷也就是我們一再提到的書面問卷轉換成為 HTML Form 的 WWW 問卷。以下舉一個系統的實例來說明：

我們一般所看到的原始文字問卷，經由我們的問卷轉換程式後所呈現的 HTML Form 如圖四。圖中的 E-Mail Address 是針對使用者在資料庫中的 primary key。



圖四、轉換成 HTML Form 問卷

接下來將介紹我們所設計實作的 Viewdb 系統雛型，並利用例子說明 Viewdb 系統運作的方式。圖五所示，為啟動系統之後所產生的一個主視窗。首先，如果我們選擇 File 下開啟檔案功能，並開啟一個 HTML Form 的檔案，檔案的內容便會顯示在工具列下方的文字區域。這時如果開啟的 Form 檔案是一個純粹處理表單中欄位資料的簡單型式表單(例

如:只是接收使用者所輸入的資料並將輸入資料顯示在瀏覽器上),並沒有牽涉到資料庫的存取時,則可以選擇 Run→Generate CGI 直接產生對應的 CGI 程式。如果沒有錯誤發生的話,則系統會出現一個提示視窗,告知使用者系統已經順利的產生 CGI 程式。接下來使用者可以選擇 View → View CGI 來觀看所產生的 CGI 程式。



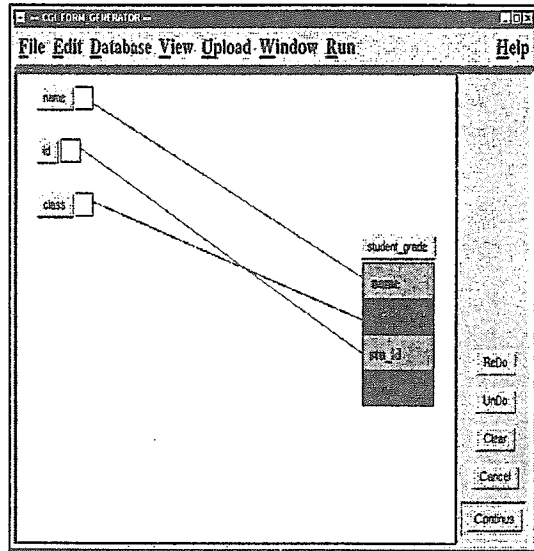
圖五、Viewdb 系統主畫面

CGI 程式產生之後,便是要決定放置的目錄位置,一般都是放置在 WWW server 下的 CGI-BIN 目錄,不過這也可能依 server 管理者當初的設定有關,而置於不同的目錄,我們可以擇 Upload → Local 或是 Remote 來決定 CGI 程式放置的目錄及名稱。另一方面,如果開啟的 Form 是屬於 DB-Form 文件,則首先必須先和遠端或在本地的資料庫系統連結。使用者可以選擇 Database → Connect DB 來選擇所要存取的資料庫所在的位置,如果資料庫和伺服器位於不同機器上,則必須先登錄遠端機器系統,成功登錄之後便出現了要求進入資料庫所須登錄的視窗。

取得了所必須用到的資料庫的表格(tables)以及欄位(fields)之後,接下來便須將表格的欄位名稱與 DB-Form 中的欄位名稱做對應,選擇 Window → Form_DB Mapping,如圖六所示,在圖六中我們以學生成績系統查詢為例來說明 HTML Form 上欄位和資料庫欄位的對應,因為接下來我們在視覺化查詢編輯器中使用的圖形物件是利用 Form 上的欄位名稱來運作的,圖中左邊的是 Form 的欄位名稱,右邊部份是資料庫表格以及欄位的名稱。

完成對應的工作之後,選擇畫面右邊視窗的 Continue,接下來便是利用系統所提供的一個視覺化編輯器,以圖形的方式來產生

Query 命令,在編輯器中使用者可以完全利用滑鼠拖曳的方式組合欄位的圖形物件以及運算物件來產生 SQL 命令。另外,我們也提供另一種以文字命令的型式,讓使用者可以直接下 SQL 命令,產生了 SQL 命令之後,接下來便要將這些 SQL 命令結合產生完整的 CGI。



圖六、Form 欄位與資料庫欄位的對應

在輸出的展示部份,我們可以將查詢的數據結果透過 Java applet 展示出來。如圖七所示,透過瀏覽器我們可以看到資料庫查詢的結果以表格型式展示,在其下方部份便是將數據資料以統計圖形方式展現。在這個 Java applet 中提供了七種不同的統計圖形式,使用者可以選擇一種來做展示。

六、結論

在這篇文章中,我們設計並實作一個可應用於 WWW 與資料庫整合之視覺化系統。本系統的目的主要是希望能減少發展 Web 以及資料庫整合應用程式所需花費撰寫程式的時間。在發展 WWW 與資料庫系統整合的應用程式,除了 CGI 的方法之外,現在也可以利用 Java 或是 API(例如: NSAPI, ISAPI)等先進的技術來發展。不過 CGI 的方法目前仍然是最為普遍的方法,所以我們也是針對 CGI 的方法來發展,希望能對於一些專業或非專業的程式發展者有所助益。至於在未來的研究方向部份,本系統未來仍然有幾點是可以再加以改進以及加強的地方。

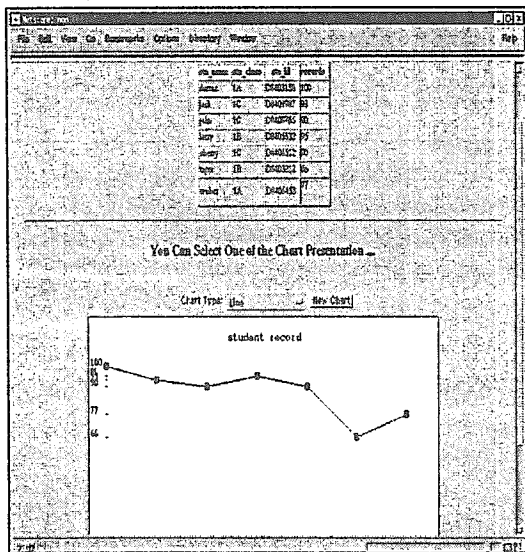
(1) 在資料庫的擴充方面:

我們目前所針對的資料庫系統是以 Sybase 以及 Mini SQL 為主,不過將來也可以擴展到其它的資料庫系統(像 Oracle、Informix、MS

SQL、FoxPro 等)。例如也可以採用 ODBC 或 JDBC 做為整合各種異質性資料庫的界面，希望能夠達到 database independent 的目標。並且可以搭配不同資料庫系統所提供的 API，使得 Web 與資料庫間的溝通更為簡易。並加上 Tcl/Tk 語言本身可在 UNIX-based 以及 PC-based 的平臺上執行，系統可以跨平臺的使用。

(2) 在視覺化編輯器部份：

目前系統雛形在以視覺化方式處理資料庫的查詢方面並不能處理非常複雜的查詢情形，我們設計的文法未來可以延伸至較複雜的文法(grammar)，例如:IN、GROUP、ANY 以及 ALL 等等的查詢運算。並且在圖形的表示方面，也可以再加以改善，希望能對於一些較複雜型式的資料庫查詢動作也可以利用我們所發展的視覺化編輯器來表示。在視覺化查詢編輯器上的圖形物件，未來可以做到利用滑鼠在圖形物件上點選時，出現像 Pop-Up Windows、Fill-in Form 以及 Pull-Down Menu 等等交談視窗，使用者可以直接在這些交談式視窗中輸入像布林運算(>、<、>=、<=、...)或其他的限制條件。



圖七、查詢結果以統計圖形展示

參考文獻

[1] Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, "Compilers Principles, Techniques, and Tools," Addison Wesley, 1985.
 [2] Ben Shneiderman, "Synamic Queries for Visual Information Seeking," CS-TR-3022, January 1994.
 [3] Chyi-Ren Dow, "PIVOT: A Program Parallelization and Visualization Environment," University of Pittsburgh, TR

94-22, 1994.
 [4] CGI*Star, Available <http://www.webgenie.com/Software/Cgistar/>.
 [5] Degi Young, Ben Shneiderman, "A Graphical Filter/Flow Representation of Boolean Queries: A Prototype Implementation and Evaluation," February 1993.
 [6] Don Libes, "Exploring Expect: A Tcl-based Toolkit for Automating Interactive Programs," O'Reilly & Associates, Inc., 1994.
 [7] Don Libes, "Using expect to Automate System Administration Tasks," January 21, 1992.
 [8] Gennaro Costagliola and Genoveffa Tortora, Sergio Orefice, Andrea De Lucia, "Automatic Generation of Visual Programming Environments," IEEE Computer, pp.56-66, March 1995.
 [9] L. Mohan and R.L.Kashyap, "A Visual Query Language for Graphical Interaction with Schema-Intensive Database," IEEE Trans. on Knowledge and Data Engineering, Vol.5, October 1993.
 [10] MASAHIRO HIRAKAWA, MINORU TANAKA, TADAO ICHIKAWA, "An Iconic Programming System, HI-VISUAL," IEEE Trans. Software Eng., Vol.16, No.10, pp.1178-1184, Oct 1990.
 [11] Microsoft Corporation, "Microsoft dbWeb1.1 Tutorial," August 20, 1996.
 [12] M. Angelaccio, T. Catarci, and G. Santucci, "QBD*: A Graphical Query Language with Recursion," IEEE Trans. on Software Engineering, Vol. 16, No. 10, pp.1150-1163, October 1996.
 [13] Shishir Gundavaram, "CGI Programming on the World Wide Web," O'& Associates, Inc. 1996.
 [14] Pyung-Chul Kim, "A Taxonomy on the Architecture of Database Gateways for the Web," 1996.
 [15] Tiziana Catarci, Shi-Kuo Chang, Maria F. Costabile, Stefano Levialdi, and Giuseppe Santucci, "A Graph-Based Framework for Multiparadigmatic Visual Access to Databases," IEEE Trans. on Knowledge and Data Engineering, Vol. 8, No 3, June 1996.