

SAWA Framework

(School Administrative-affair Web Application Framework)

支援單一登入及 MVC 樣式的校務行政系統應用程式框架

國立政治大學電子計算機中心

SAWA Framework 建置目的

短時間內發展出功能完整、風格一致且容易維護的應用程式

國內多數大學院校的校務行政系統都是由電算中心人員自行建置或以委外方式開發。系統的需求則由相關行政業務負責人提出，不同的開發人員再依需求量身訂做。時間長久之後，整個校務行政系統變得小系統林立，造成系統維護上的嚴重問題。隨著校務決策的改變，分散在各系統相關的程式碼都要一一取出並加以修改，不但浪費寶貴人力，也容易造成錯誤。

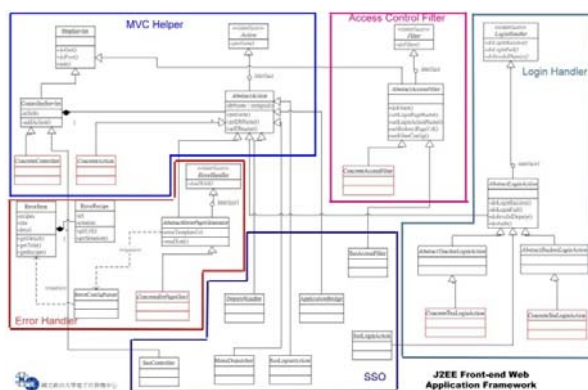


圖 1. SAWA Framework

利用設計樣式(Design Patterns)及框架(Framework)的概念，結合過去建置 Web 校務行政系統的經驗，我們發展出一個以 MVC(Model-View-Controller)為基礎，支援單一登入(Single Sign-on)的校務行政系統 Web 框架。在這個框架基礎之上，開發人員可以輕易地在短時間內發展出具有功能完整、風格一致且容易維護的應用程式。我們稱這個框架為 SAWA Framework(School Administrative-affair Web Application Framework)。

以 J2EE™ 技術為基礎，提供功能完整的校務系統開發平台

SAWA Framework 是一個以 J2EE 前端技術(JSP 1.2/JSTL 1.0 與 Servlet 2.3)為基礎，支援單一登入及 MVC(Model-View-Controller)樣式的校務行政系統 Web 框架。基於這個框架，開發人員上可以在短時間內開發出具有 MVC、單一登入、一致外觀、有效資源管制及結構化錯誤處理機制的應用程式。

SAWA Framework 系統特色

風格一致的使用者界面(User Interface)

校務行政系統會不斷的成長及修改，如果每支系統都有各有一種界面及操作方法，不但造成使用者操作上的困擾，也使整個系統看起來沒有整體感。若校務系統屬於外包形式，這種情況可能更加嚴重。SAWA 使校務系統具有風格一致的使用者界面。新增的應用程式以元件方式嵌入整個系統，其使用者界面便會與既有系統一致，嵌入的過程完全不會造成程式開發人員額外的限制及負擔。



圖 2. 風格一致的使用者界面

透過 Custom Tag Library，我們將系統的美工與程式部份清楚地切開，一旦校務系統的外觀需要修改，除 Custom Tag Library 外，其它程式完全不需要更動。

SAWA 提供的 Custom Tag Library 同時可完整支援 JSTL 1.0 的 Expression Language(如圖 3 中之 `#{sessionScope.chname}` 等敘述)。因此對開發人員來說，除了 namespace 不同，使用 Server Control 和使用 JSTL 的其它標籤感覺沒有差別，不必額外花時間學習。

```
<nccu:html>
  <nccu:header pageTitle="網頁標題" contentType="text/html"
    encoding="big5">
    <%-- html 標頭區，有其它標頭設定或java script可放在這 --%>
  </nccu:header>
  <nccu:body apname="行政系統"
    userName="#{sessionScope.chname}"
    message="#{sessionScope.apname}" >
    <%-- 在這裏開始寫JSP --%>
  </nccu:body>
</nccu:html>
```

圖 3. 使用 Custom Tag Library 為應用程式提供一致的外觀

動態提供線上說明

SAWA 提供動態的線上說明機制，系統會視使用者目前使用的頁面，動態地提供合適的線上使用說明。隨著每支新系統的加入，新增加的線上說明文件也可以直接嵌入系統。



圖 4. 動態提供線上使用說明

協助開發人員有效率地以 MVC 樣式來開發 Web 應用程式

Web 的應用程式中實作 MVC 要處理的細節很多，若無特殊的機制協助，可能造成開發人員無法專注於真正要處理的邏輯上，對開發過程反而造成負面影響。SAWA 中的「MVC Helper」提供容易學習的機制來協助開發人員處理 MVC 重覆的細節部份。

支援單一登入 (Single Sign-on)

單一登入機制指的是使用者在某支 Web 應用程式登入後，便不需重覆登入即可使用他有權使用的應用程式。應用程式之間如缺乏單一登入的機制，會讓使用者每使用單獨一個系統，皆需重複做登出、登入的動作。



圖 5. 提供單一登入之窗口

實作單一登入面臨到最大的考驗是如何跨 Web 應用程式分享使用者的登入資訊。登入資訊一般利用 Session 儲存，但在 Servlet 2.3 的規格書中明訂 Session 不可跨 Web 應用程式分享。另外，整個單一登入機制必須是可擴充的，以便新加入的應用程式隨都可以單一登入的功能。

我們以生命週期和 Session 一致的加密 cookie 來解決這個問題。使用者登入後，一個記錄登入資訊的加密 cookie 會送到 Client 端，其生命週期與 Session 一致；關閉瀏覽器時，cookie 也同時失效。另外，ApplicationBridge 的機制使單一登入系統具有可擴充的彈性。ApplicationBridge 是一個 Servlet，它會讀出 cookie 並加以解密，再將資訊橋接到框架中的 LoginHandler 中，自動為使用者完成登入程序。而單一登入系統本身就是一個建構在 SAWA 框架上的 Web 應用程式。

登入機制的抽象化

幾乎進入所有的系統的第一步都是登入並取得使用者的資訊，應該將登入機制加以分析，將共同部份適當地抽象化，降低開發時的負擔。SAWA 中的「LoginHandler」主要功能是將登入功能抽象化，並整合至「單一登入」及即存的「職務代理人機制」。

管制資源的不當存取

許多設計不良的應用程式可能會忽略資源存取的管制。程式必須確保除了登入的合法使用者之外，系統資源沒有其它任何的進入點。但若在開發時期就加上管制，常造成除錯的困難。所以需要一種經由設定檔就能動態加上或解除管制的機制。

傳統防止使用者跳過認證頁直接指到某頁的方法都是在處理 http 要求時，一開始便檢查 session 中是否有認證資料。但只要檢查權限的邏輯一改，則每處都必須跟著修改。另外在程式在開發時期，有時候不想加這個限制，比較容易除錯。

我們使用 J2EE Pattern 建議的 Filter 來做 Access Control，藉由改變 web.xml 的設定，可以隨時將 filter 裝上或卸下。在 web.xml 中可以事先設定，那些要求會被 FilterServlet 攔截。攔截後加以驗證，如果不合法，就導到登入頁要求登入。

結構化的錯誤處理

當應用程式發生錯誤時，程式必須向使用者說明錯誤的原因及解決方法。如果沒有一定的規範，每個開發人員所寫的錯誤處理網頁都不同，容易造成使用者的困擾。

依照過去開發的經驗，我們將錯誤資訊分成四個部份：

- **錯誤標題**：以一句話簡述錯誤情況。
- **錯誤 ID**：用來識別這個錯誤。
- **錯誤描述**：向使用者簡述錯誤發生的原因及復元的可能性。

解決方案：以超連結方式引導使用者回報錯誤或重試。

```
<ErrorItem>
  <id>500</id>
  <title>系統內部目前發生錯誤，暫時無法提供服務</title>
  <detail>.....</detail>
  <recipes>
    <recipeItem>
      <solution>回前一頁</solution>
      <url>javascript:history.back()</url>
    </recipeItem>
    <recipeItem>
      <solution>登出</solution>
      <url>http://moltke.cc.nccu.edu.tw/SSO/doLogout</url>
    </recipeItem>
  </recipes>
</ErrorItem>
```

圖 6. 錯誤網頁的設定檔

SAWA 框架的開發人員在開發時期不需要寫作錯誤網頁，他們只要事先在 xml 檔案中設定(如圖 6)，給予 ErrorPage Builder 錯誤標題、錯誤描述及解決方案的資訊，系統會自行建立錯誤的網頁(如圖 7)。ErrorPage Builder 在 Web 應用程式初始化時會讀入 error.xml 設定檔，並使用 SAX API 解析此設定檔後，將之轉換成錯誤資訊的網頁。

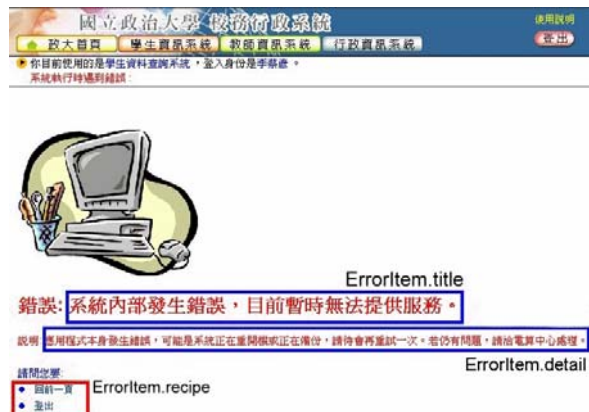


圖 7. 系統依據 xml 檔建立的錯誤網頁

SAWA 框架的效益

使用 SAWA 開發校務系統方式具有以下優點：

- 由於每支應用程式均基於相同的框架開發，所以程式碼整體的品質提高。
- 不同開發人員開發之應用程式網頁美工及錯誤處理風格均能保持一致。
- 框架中幫助開發人員處理許多細節，因此提高了系統開發的效率。
- 系統以 MVC 方式清楚地切割，所以程式除錯及維護較容易。
- 資源管制功能由框架統一處理，因此不用擔心開發人員寫程式時造成管制漏洞。
- 一致且清楚的錯誤及解決方案說明，幫助使用者了解系統失效的原因及初步解決方案。有效降低了使用者諮詢服務的負擔。
- 框架扮演了設計規範的角色，降低外包風險。

結語

目前政治大學 Web 上的校務行政系統均以 J2EE 技術為基礎，建構在單一的校務行政系統框架上。

新增之專案在完成系統分析，進入設計階段後，開發人員藉由校務行政系統框架的幫助，以其提供之抽象類別為基礎，訂製出該系統的 **Controller** 及 **Action** 類別，建立整個系統的基礎結構。

使用者界面則以 **JSTL 1.0(Java Standard Tag Library)** 為主，利用校務行政系統框架所提供的 **Server Controls** 提供一致的外觀。待整個系統完成之後，再修改設定檔，設定 **Access Filter** 之管制範圍，將整支子系統嵌入校務系統中。

針對校務行政系統所面臨的問題與需求，我們建立了一個以 **MVC** 為基礎，支援單一登入(**Single Sign-on**)的應用程式框架，並將此框架實際應用在政治大學的校務行政系統。這個框架的建立讓開發人員在短時間內可發展出具有完整功能、風格一致且容易維護的應用程式。未來我們希望在行政業務部份也都能加以抽象化，並加入支援目錄服務之功能，使 **SAWA Framework** 的功能更完整。

聯絡方式

國立政治大學電子計算機中心應用系統組 廖峻鋒先生，try@nccu.edu.tw