

# 植基於 API 之網路應用雛型快速開發平台

## An API-based development platform for efficient web application prototyping

王建堯、曾聖博、楊偉楨、蔡其杭

財團法人資訊工業策進會

前瞻科技研究所

Email: {jimmywang, leovleov, wjyang, brick}@iii.org.tw

### 摘要

近年來，隨著行動裝置的盛行與開放 API (Application programming interfaces) 的快速增長，使得 API 的運用成為許多企業產生新的收益模式與產業價值所不可或缺的經濟模式，API Economy (介面經濟) 日漸成為產業關注的議題。目前多數開發網路應用的開發模式存在著無法快速應用與有效重用 API 的問題，多數的開發者必須經過繁瑣的程式實作建置應用，本論文將介紹一個以流程為基礎的開發平台，使用者能利用既存的 API，快速開發網路應用雛形，以驗證服務概念。

本開發平台希冀利用商業流程管理 (Business Process Management) 能夠快速變動邏輯與易於分析的特性，設計一個可以更快速開發網路應用雛型的開發平台。本平台可以透過既存 API 的輸入與輸出介面快速的產生此 API 的對應資料模型，並且根據資料模型內容自動產生相對應的使用者介面。所以應用開發者只需透過商業流程管理 (Business Process Management) 設計網路應用服務的整體邏輯與選擇需要使用的 API，便可藉此快速產生網路應用服務的基本雛形。此作法簡化了服務整體的設計模式並且加速了網路應用的開發過程，同時以流程來塑模應用邏輯也強化了服務的彈性和再用性，可有效提升服務概念驗證的效率。

本文中以網路商家預定服務為例，利用此服務設計方式實作網路應用雛形如商品查詢、商家預定等功能。透過此類服務實作，實證此開發平台之可行性與功效。

關鍵詞：介面經濟、商業流程管理、資料模型

### Abstract

Recently, with the prevalence of mobile devices and the rapid growth of open API, making use of the API becomes an indispensable business model to generate revenue and industrial value for many companies, which means the API Economy is becoming one of the most important industry issues. However, most web application development model cannot effectively reuse APIs, so many developers must implement their applications through complex implementation steps. We describe a process-based

development platform, which a platform user can take advantage of existing APIs to quickly build a web application prototype to validate the concept and value of the service.

This development platform is a fast web application prototyping platform that leverages the abilities of BPM (Business Process Management) to efficiently change and analyze business logic. The platform can quickly generate corresponding data model for existing API through the input and output interfaces, and automatically generate the user interface based on the data model. To develop a web application, the developers only need to design the application logic and choose the APIs they want through BPM-like development environment, and the platform will do the rest for them. This approach simplifies the overall service design mode and accelerate web application development process, while the application process also reinforces the flexibility and reuse of services, which can effectively improve the efficiency of POC.

We further take e-booking services, for example, the "Restaurant Booking" process, to represent feasibility and efficacy of our development platform.

Keywords: API Economy, Business Process Management, Data Model

### 一、前言

現今網路應用 (web application) 廣泛運用於各個領域，快速且方便的軟體相容性，與使用者的容易使用，已經成為必備的應用設計模式。許多的設計模式相繼提出，其中以嵌入式命令語言 (imperative embedding)[1] 的網路應用設計模式為目前大眾所廣泛使用。利用嵌入式語言，例如 JavaScript、PHP 或者是 HTML 產生與設計網路應用的使用者互動介面與邏輯。但透過此類設計模式，相網路應用開發者則可能必須花費許多的時間來設計與除錯網路應用程式，並且往後欲更新或新增功能與軟體時，又可能再花費相對應的開發時間。

因此，網路應用的設計模式強調的應該為如何有效率且容易的建立一個富有彈性的應用平台。其中以 Leff 等人所提出的 Web Relational Blocks (WebRB)[2] 平台，一個容易檢視的虛擬資料流的設

計模式，成為受歡迎的傳統開發平台。WebRB 是由以 Relational Blocks (RB)[3]為基礎衍生的開發模式，他定義了許多預先可供開發者使用的工具來快速的生成後端資料管理與設計使用者介面。此種設計方式，有效的解決嵌入式命令語言網路應用開發時程問題，但此類模式只提供了基本的前端資料管理功能，在後端邏輯變更時卻無法免除大量的變更開發邏輯，故更進一步，Weske 等人也提出了利用商業流程管理(Business Process Management)來開發與設計網路應用邏輯[4]。利用商業流程易於變更邏輯與快速分析的優勢提供更有效且靈活的網路應用設計模式。商業流程中的任務組與關聯可以有效管理整體網路應用的邏輯流程。其中有許多結合商業流程的網路應用開發平台也相繼提出與討論，如 Koch 等人[5]所提出的 UML-based 網路工程方法設計模式。Schmid 等人[6]提出的物件導向超媒體方法結合商業流程的網路應用模組。和 Brambilla 等人利用了商業流程的模組化與 WebML(web modeling language)，成功並更有效率的開發網路應用。

另一方面，因開放 API 的快速增長與盛行，使得 API 的運用成為許多企業產生新的收益模式。API Economy[7]的低成本開發模式日漸成為產業開發網路應用的新模式[8][9]。其中 Janes 等人[10]提出透過了共通的 API 介面的易讀與建立，能夠找出一個共同的 API 模型架構定義。

透過以上所提出的平台驗證，現有網路應用設計模式雖然可以結合商業流程進行開發，並有效使網路應用開發更有效率，然而由於商業流程引擎對於使用者介面的限制，使得使用者介面不容易依照需求客製化，且後端服務串接也需依照不同網路應用需求大量撰寫，造成維護與更新耗時。此外，如何有效利用開放 API 來快速開發新的網路應用並進行驗證，則是另一個值得探索的新議題。故本平台希冀透過開放 API 與使用者介面引擎的加入，設計一網路應用設計平台來解決以上兩個問題。

一般網路應用的最基本元素，即是在頁面上呈現一組欲展示給使用者觀看的資料，並提供使用者回應或其他互動選項，進一步帶出其他功能或給予使用者更多其需要的資訊。基於以上概念，我們設計了一個網路應用雛型開發平台，透過開放 API 自動產生對應的資料模型與前端使用者介面。開發者只需透過商業流程定義網路應用的邏輯與資料模型的使用模式，即可透過此平台有效率的產生網路應用的雛型。圖一為此平台模組架構與資料流示意圖。

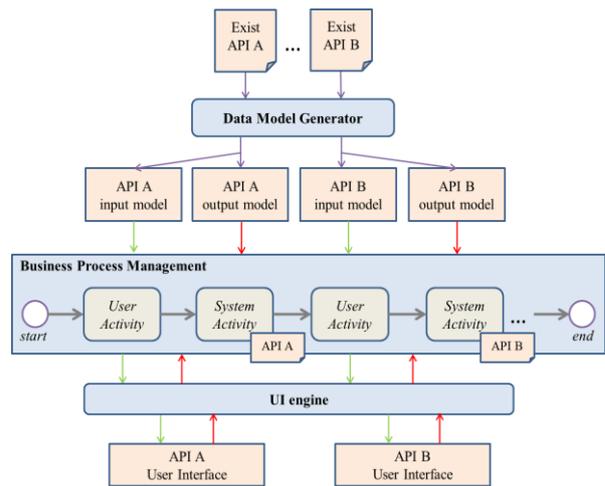


圖 1、平台模組架構與資料流

## 二、平台設計

平台依據功能可分為三個模組，如圖 1 藍色區塊所示，分別為商業流程管理、資料模型產生器、使用者介面引擎，以下分段詳述。

### 1、商業流程管理 (Business Process Management)

商業流程定義整體網路應用服務的商業流程與使用者互動邏輯，不同的應用服務依據商業模型設計與實作出所需的任務組(activities)與各任務之間的關聯(transitions)。其中系統所執行的邏輯在商業流程中定義為系統任務(system activity)，而使用者互動邏輯定義為使用者任務(user activity)。實作系統任務常以 APIs 形式完成，而本平台設計了一個資料模型產生器來將 APIs 的描述轉換為流程中的輸入輸出資料模型(詳見 2、資料模型產生器)。當系統運行到使用者任務時，商業流程將輸入輸出資料模型傳送至使用者介面引擎，來產生對應的使用者互動頁面，而運行至系統任務時，所產生的資料模型也能透過商業流程的中介，快速傳入此任務的 APIs 處理資料，如圖 2 所示。如此一來，在本平台上的應用服務透過商業流程來定義整體邏輯和資料，每當應用服務的邏輯需要改變時，開發者就能夠利用商業流程設計的優勢，快速且靈活的修改流程來達成邏輯改變，而不需進行繁瑣的程式邏輯修改。

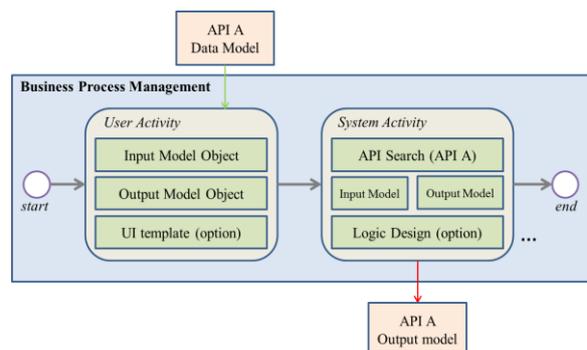


圖 2、商業流程內部模組設計

## 2、資料模型產生器(Data Model Generator)

資料模型(Data Model)為平台所定義的一組傳輸格式, APIs 透過 swagger 標準化後, 能夠透過 Swagger-Jersey-JAXRS 官方套件導出標準格式的 APIs 輸入與輸出參數, 定義為兩組 key 與 value 對應的 json 標準資料格式。經由資料模型產生器的包裝與格式轉換, 產生輸入模型(Input Model)與輸出模型(Output Model), 輸入模型定義為 APIs 的輸入參數, 而輸出模型則定義為輸出結果。此資料模型格式透過 APIs 對應產生, 故可以完全符合 APIs 輸入與輸出參數格式, 透過使用者介面引擎與資料模型的溝通, 能夠有效將 APIs 介面呈現。

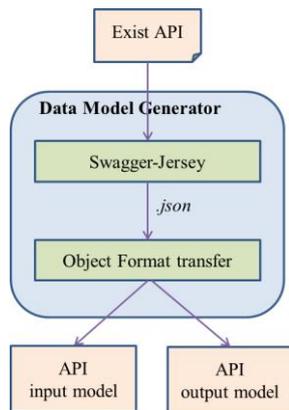


圖 3、資料模型產生器內部模組設計

## 3、使用者介面引擎(UI Engine)

使用者介面引擎的功能為使用者介面的自動生成, 並連結商業流程與前端網頁。

當使用者介面引擎接受到客戶端的請求, 可能是一個流程初始化亦或者是繼續一個流程的請求, 此時使用者介面引擎會依據前端所傳入之請求, 連結商業流程引擎取得流程中對應的資料模型, 並利用此資料模型生成相關介面組件(UI components), 有效率的自動產生對應使用者頁面雛形。在使用者完成網頁的填寫或操作後, 使用者介面引擎會再將使用者前端操作的結果與資訊包裝組合後傳送至商業流程中。

開發者更可以依據需求更進一步修正自動產生之使用者介面, 依據需求與應用程式風格匯入介面樣板(UI template), 可能是一組外部定義的 Cascading Style Sheet (CSS) 或者一組網頁, 就可以提供更美觀與彈性的頁面設計。

## 三、實例展示

此章節描述與設計一個網路應用實例設計開發, 以驗證此雛形開發平台之實際運用與可行性。此平台運作模組運行於 Intel Core i7-2640M CPU @2.8GHz 和 8GB RAM 硬體環境, 而透過 Zend server 6.2.0 的網路服務與 Activiti Business Process server 5.16 商業流程運行環境實作。其中開發流程

依序為服務邏輯定義、擇定 APIs 並導出資料模型與商業流程實作, 以下分段詳述。

## 1、服務邏輯定義

本實例設計開發一個網路商家預定服務, 內容包含商家資訊查詢與商家瀏覽, 並可進一步針對選擇的商家做預訂的動作。對於開發此網路服務, 開發者可能需要的 APIs 有商家查詢服務與預定服務。故可定義出整體服務邏輯設計為圖 4 所示, 使用者可經由查詢頁面與預定頁面與此網路服務產生互動。

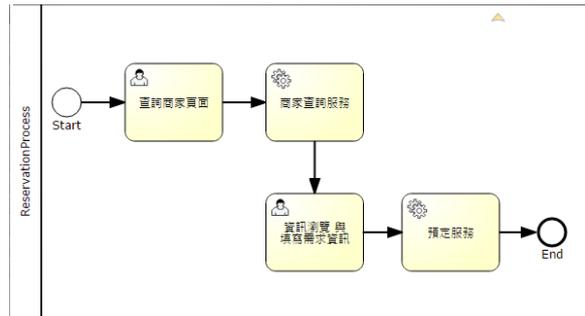


圖 4、商家預定服務商業流程設計

```
GET /foodanddrink/findByCounty
Get the foodanddrinks information by county id

Response Class
Model Model Schema
FoodAndDrink (
  id (string, optional),
  themes (array(string), optional),
  description (string, optional),
  pictures (array(Picture), optional),
  videos (array(string), optional),
  location (Location, optional),
  tels (array(string), optional),
  address (string, optional),
  opentime (string, optional),
  avgrank (number, optional),
  parking (string, optional),
  price (integer, optional),
  note (string, optional),
  website (string, optional),
  transport (string, optional),
  county (string, optional),
  collection (integer, optional),
  shareUrl (string, optional),
  name (string, optional)
)
Picture (
  url (string, optional),
  metas (array(PictureMeta), optional),
  description (string, optional)
)
PictureMeta (
  url (string, optional),
  width (integer, optional),
  height (integer, optional)
)
Location (
  latitude (number, optional),
  longitude (number, optional)
)
Response Content Type application/json
Parameters
```

圖 5、商家查詢服務 API 相關資訊

## 2、擇定 APIs 並導出資料模型

在此網路服務中由前段論述, 必須有商家查詢服務與預定服務兩個 APIs。透過既有的 APIs 中搜尋符合此需求之 API, 如圖 5 所示, 或實作一個符合需求的新 API。將 APIs 設定至商業流程中的系統任務中, 平台將自動程式碼建構(code generation) 出符合該 API 的連結程式, 讓執行階段時可透過商業流程呼叫此 API 並存取。另一方面, 透過資料模型產生器快速產生此 API 的輸入模型與輸出

模型，其中輸入模型的內容節錄如以下所示。

```
{ "title": "預定頁面",
  "type": "object",
  "id": "Booking",
  "required": false,
  "properties": {
    "name": {
      "title": "姓名",
      "type": "string",
      "id": "UserName",
      "required": true },
    "count": {
      "title": "預定人數",
      "type": "integer",
      "id": "HeadCount",
      "required": true },
    "tele": {
      "title": "聯絡電話",
      "type": "string",
      "id": "Tele",
      "required": true },
    "subtype": {
      "title": "時段",
      "type": "string",
      "id": "Timetype",
      "required": true,
      "enum": ["01001", "01002"],
      "options": {
        "enum_titles": ["中餐", "晚餐"] },
    "mail": {
      "title": "電子郵件",
      "type": "string",
      "id": "Email",
      "required": true },
    "description": {
      "title": "備註",
      "type": "string",
      "format": "textarea",
      "id": "Description",
      "required": false }, }
```

### 3、商業流程實作

商業流程必須定義資料模型的流向與設定系統任務串接的 API，因此平台開發者要定義每個系統任務和使用者任務的輸入與輸出等資訊，但由於上一步驟中平台已自動為使用者導出了資料模型於流程中，使用者可以很容易的將其設定完成。藉由商業流程的實作定義，本平台的使用者介面引擎能夠利用資料模型快速生成使用者頁面雛形。如圖 6 所示為商業流程中第二個使用者任務所產生的資訊瀏覽與預訂資料輸入頁面，頁面上所顯示的表單資訊則直接由商家查詢 API 的輸出參數與商家預定 API 的輸入參數所產生。故當使用者送出資訊時，使用者介面引擎能夠快速地包裝使用者填寫資訊送回商業流程，並直接藉由下一個系統任務(預定服務)將資料傳遞至 API 中。透過此流程的簡單實作，便能達成透過 APIs 快速產生商家預定服務應用。



圖 6、使用者介面雛型示意圖

更進一步，假如往後需求變更，必須將查詢內容改變或預定其他類型服務時，只須更新 API 和重新產生資料模型即可快速完成流程調整，商務邏輯不受影響，前端介面也會因更換的 API 自動產生對應頁面，大大提升服務的再用性。

### 四、結論

本論文提出與驗證一網路應用雛型開發平台。透過商業流程引擎與標準格式 APIs 的串接應用，配合平台特有的資料模型產生器與使用者介面引擎，應用開發者只需利用商業流程設計網路應用服務的整體邏輯與選擇所要使用的 API，便可藉此快速產生網路應用服務的基本雛形。同時，當邏輯更動亦或者是新增其他功能時，只需重新擇定 API 產生資料模型與變更商業流程即可，實現簡單、高效率且富有靈活度的網路應用雛型開發平台。

### 致 謝

本研究依經濟部補助財團法人資訊工業策進會「103 年度系統服務與事業轉型計畫(2/4)」辦理。感謝經濟部技術處支持資訊工業策進會前瞻科技研究所進行網路應用雛型快速開發平台發展之研發。

### 參考文獻

[1] Avraham Leff, James T. Rayfield, "WebRB: a different way to write web applications," IEEE Internet Computing, vol. 12, no. 3, 2008, pp. 52-61.  
[2] Avraham Leff, James T. Rayfield, "WebRB:

- evaluating a visual domainspecific language for building relational web-applications,” ACM SIGPLAN Notices, vol. 42, no. 10, 2007, pp. 281–300
- [3] Daniel D. Hils, “Visual languages and computing survey: dataflow visual programming languages,” J. of Visual Languages and Computing, vol. 3, no. 1, 1992, pp. 69–101.
  - [4] Mathias Weske, Business Process Management: Concepts, Languages, Architectures, 2nd ed., Springer, 2012.
  - [5] Nora Koch, Andreas Kraus, Cristina Cachero, and Santiago Melia, “Integration of business processes in web application models,” J. of Web Engineering, vol. 3, no.1, 2004, pp. 22–49.
  - [6] Hans Albrecha Schmid and Gustavo Rossi, “Modeling and designing processes in e-commerce applications,” IEEE Internet Computing, vol. 8, no. 1, 2004, pp. 19–27.
  - [7] Israel Gat ,Agile, Giancarlo Succi “A Survey of the API Economy,” Agile Product & Project Management , Executive Update Vol. 14, No. 6., 2013
  - [8] Gat, Israel, Remencius, Tadas, Sillitti, Alberto, Succi, Giancarlo, Vlasenko, Jelena, “API Economy: Playing the Devil’s Advocate,” Cutter IT Journal 26(9), 6–11, 2013
  - [9] Remencius Tadas, Succi Giancarlo, “Tailoring ITIL for the Management of APIs,” Cutter IT Journal 26 (9), 22-29, 2013
  - [10] Andrea Janes, Tadas Remencius, Alberto Sillitti, Giancarlo Succi “Towards Understanding of Structural Attributes of Web APIs Using Metrics Based on API Call Responses,” OSS 2014: 83-92.