

# 整合 Facebook Like 與 Google Plus +1 的第三方混搭機制

## A third-party mashup mechanism for integrating Facebook Like and Google Plus +1

范競之 梅興

萬維運算實驗室

輔仁大學資訊工程學系

Ching-Chih Fan Hsing Mei

Web Computing Lab., Department of Computer Science and Information Engineering,

Fu Jen Catholic University, Taipei, Taiwan, ROC

Email: www1036@weco.net mei@csie.fju.edu.tw

### 摘要

本文主要探討 Facebook Like 與 Google Plus +1 兩社交外掛的服務，混搭於第三方的環境中所面臨的問題。對於一般使用者可能因為資料雜亂而降低使用的程度，第三方服務受限於兩者服務的開放性，造成整合上的困難。第三方服務要統計 Like 及 +1 的資訊，會碰到同一使用者雙重點擊的問題產生。因此本文主要以此為方向，進行解決雙重點擊問題，並提出 MI(Mashup Integration)服務來收集這些點擊的資訊處理上述問題。經由所收集到的資料，進而提出對於使用者點擊量及其行為分析的指標，供第三方服務作為判斷網站價值的方式之一，亦可作為服務的加值應用或廣告投放的依據。

**關鍵詞：**混搭、社交、整合、外掛

### 一、簡介

隨著網路的發展，造就了許多服務興起，自 2005 年 O'Reilly 提出 Web 2.0 概念後，網路應用更著重於人與人之間的互動及分享，因此興起的服務若依賴於這種特性運作，必須要有激起使用者的動機來參與服務的內容；但僅限於單純瀏覽還不夠達成目的，是要讓使用者能夠主動性去貢獻資源或動作，以達 Web2.0 之精神，更讓服務蓬勃發展，因此許多網站紛紛開發出 API(Application Programming Interface)[1]或外掛(Plug-in)提供第三方服務內容進行混搭(Mashup)。在 Web 環境中，開發出屬於自己的個人網站或檔案而不再像過去這麼困難、複雜，但混搭最大的問題是在於整合上的困難；同質性的混搭服務，有時卻因為實際上的需求，而不能從中選一，但又面臨資料整合的困擾。

對於使用者而言，在混搭環境中過度雜亂的資訊下，反而會影響到使用者的操作流程，甚至是使用意願；對於第三方開發者而言，除了推廣自己服務的內容外，獲知自己服務在網路上的價值是非常重要的商業模式之一。在過去判斷網站服務的價值往往透過 SEO 增加瀏覽人次的流量來提升排名；但現今網路當中，第三方服務可以透過買廣告或關鍵字，甚至還有流量銀行的服務，以自我人為的方

式提高曝光數。這種評估方式，有時反而會造成資訊的不正確性，恐影響到使用者獲取到錯誤的資訊。

在 Facebook 的 Like 與 Google+ 的 +1 功能問世後，透過使用者點擊 Like 或 +1 分享到自己的檔案上，是使用者對於網站內容的認同感。藉由這種方式可降低以往造成資料不正確性的問題，因此網站的評估也將這種功能納入其中，如 Google Analytics 將 +1 納入為評估方式之一。

因此本文將以此背景為基礎，討論其整合問題的方法，以及藉由整合後的資訊提供第三發開發者判斷網站價值的指標。

### 二、Open Graph Protocol

2010 年 4 月，Facebook 在 f8 conference(即 Facebook developer conference)正式提出 Open Graph Protocol[2, 3]，這項標準包含整合在第三方服務中混搭 Facebook 的社交外掛(Social Plug-in)——「Like」。Open Graph Protocol 在 meta tag 預先定義了許多格式，以 og:前綴表示，如 og:title、og:type、og:url、og:image...等基本及選用的格式，其中 type 又可分為 website、music、article...等類型，整體的格式如圖 1 所示。

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:og="http://ogp.me/ns#"
      xmlns:fb="http://www.facebook.com/2008/fbml">
  <head>
    <title>The Avengers</title>
    <meta property="og:title" content="The Avengers"/>
    <meta property="og:type" content="movie"/>
    <meta property="og:url" content="http://www.example.com/title/1234567890/">
    <meta property="og:image" content="http://www.example.com/avengers.jpg"/>
    <meta property="og:site_name" content="example"/>
    <meta property="fb:admins" content="USER_ID"/>
    <meta property="og:description"
          content="This is a description."/>
    ...
  </head>
  ...
</html>
```

圖 1. Open Graph Protocol 於電影資源的例子

透過在 meta tag 定義的格式，將任何人與資源皆視作為「object」，使用者可將所喜好的內容儲存於個人檔案中。藉由此項設計，對於外部網站開發或經營者而言，網站的內容會在社交網站更加容易的宣傳，對 Facebook 則是可以取得使用者的喜好或活動記錄作為其他服務的延伸。

2011年9月的f8 conference上，對於Open Graph Protocol又有重大變革，更強調於人與資源之間的關係。以往不論是人或資源，皆視作為「object」，但改變後，可以更清楚定義人(user)、資源(object)及人與資源間的關係(action)，甚至可以自行定義資源的type，再延伸出屬於該網站服務自己的格式，讓原本的Open Graph Protocol更具有彈性。

Open Graph Protocol具語義網(Semantic Web)的概念，語意網也被認為是下一個網路世代—Web 3.0的象徵，企圖讓電腦理解人類所要表達的意念。在現今網路環境資源混亂的情況之下，要能馬上正確取得相對應所需的資源，是十分困難的一件事。大多數人每天都會利用到網路服務的搜尋系統，每次搜尋出來都是參次不齊的結果，排名第一個項目還不一定是自己所需要的內容；也有部份搜尋系統已導入語意網的概念，如Wolfram alpha，但實質上是屬於一種知識性導向(Knowledge-based computing)的問答的系統，與一般搜尋引擎仍有較大的差異。

Google+的+1也支援Open Graph Protocol，但現今Google+無法支援至最新版本。本文中的「Like」及「+1」，就是利用語意網的概念，使用者在第三方服務網站按下「Like」及「+1」的按鈕時，同時Facebook或Google+會解析該網頁資源的內容，將使用者所喜好的資料，同步儲存於Facebook或Google+的個人檔案中。

### 三、社交外掛整合

#### 1. 問題描述

在許多網路資源上會同時看到Facebook的Like與Google+的+1的社交外掛，能夠讓使用者在閱讀完該資源後，依據本身認知來決定是否按下這個社交外掛的按鈕(如圖3)，將其記錄至個人檔案中及分享給更多朋友獲知該資源的內容。但以下圖例可知，兩者間的社交外掛現存狀況是各自獨立。對使用者而言，易造成使用者的混淆，而不知該點擊何者(如圖3)；使用者點擊的資訊將會直接傳送回Facebook或Google+，管理者透過Facebook及Google+取回該資源點擊資訊，也將造成統整的困難(如圖4)，清楚點出統整的問題—重複計數的癥結點。



圖 2. Facebook Like 與 Google Plus +1 的社交外掛

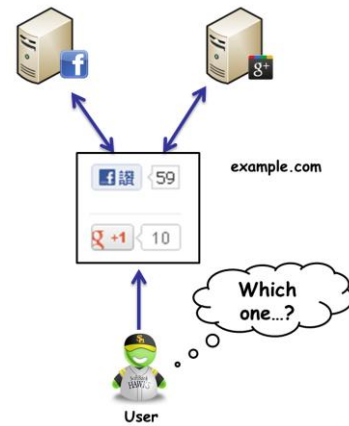


圖 3. Like 與 +1 的社交外掛於使用者面臨之問題

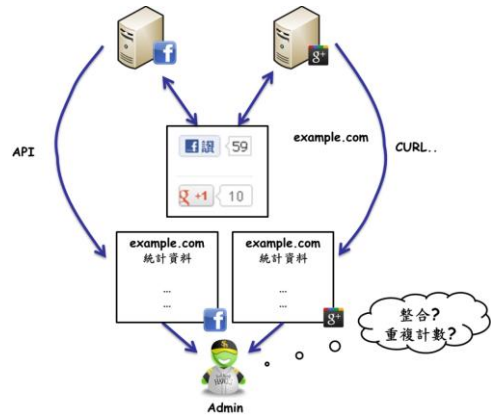


圖 4. Like 與 +1 的社交外掛於管理者面臨之問題

#### 2. 整合模型

##### 2.1. 使用者介面

在前一節曾提過，因為兩者的Facebook的Like與Google+的+1的社交外掛是各自獨立分開。除了易發生使用者的困擾(如圖3)，點擊資訊也會分別回傳至不同的服務提供者進行統計(如圖4)。以上敘述更進一步說明，Like與+1回傳資料的流向如圖5，將使第三方服務開發者取得點擊資訊的困難，更造成重複計數的問題。

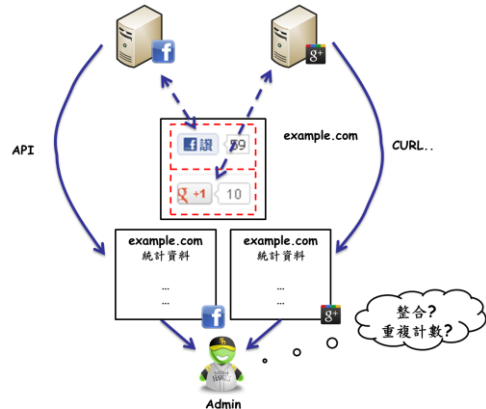


圖 5. Like 與 +1 的社交外掛資料回傳流向及對於管理者面臨之問題

在本文中，第一個欲解決的內容是解決使用者介面的整合，同時將回傳的資料進行統一的處理再回傳給各自的服務提供者。所以在使用者介面上，必須先將 Like 及+1 進行整合成一個社交外掛，以降低使用者的混淆，設計如圖 6。



圖 6. 整合之社交外掛

雖然概念設計已將兩者的社交外掛整合，但仍須考量到使用者的狀況。假設使用者均有 Facebook 及 Google+ 帳號，但使用者只想使用其一的服務的情形。因此在圖 7 至圖 10 將進行各種的情景說明，表 1 將會統整各種情景的狀況。

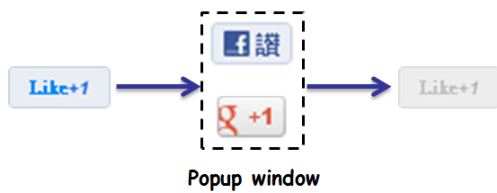


圖 7. 尚未 Like+1

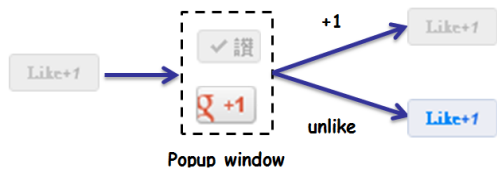


圖 8. 已用 Facebook 帳號 Like+1 過

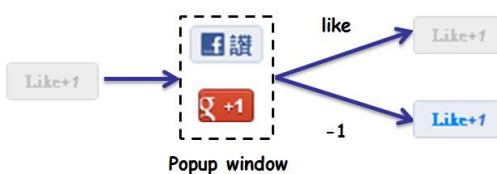


圖 9. 已用 Google+ 帳號 Like+1 過

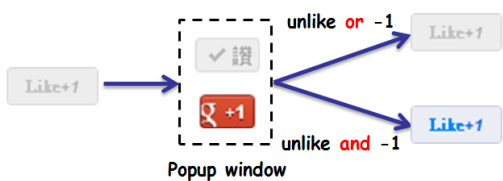


圖 10. 已用 Facebook 及 Google+ 帳號 Like+1 過

圖 7 代表為均未 Like+1 過的初始狀態；圖 8 代表為僅已使用 Facebook 帳號 like，其中又分為兩種情況，分別為再繼續使用 Google+ 帳號+1 及 Facebook 帳號 unlike；圖 9，同理；圖 10 則是已使用兩者的帳號 like 及+1，但也可分為兩種情況，一是僅使用 unlike 或-1，二是 unlike 及-1 均使用的狀況。

表 1. Facebook Like 與 Google Plus +1 的統整使用情景

當前狀態 條件	未 Like+1	已 Like+1		
		fb	G+	fb and G+
fb	Like	Unlike	Like	Unlike
G+	+1	+1	-1	-1
fb and G+	Like and +1	-	-	Unlike and -1

統整後的情況如表 1 所示，當前狀態可分為以下兩種情形：(1)未 Like+1 未受過任何來源的點擊；(2)已 Like+1 受過任一來源的點擊，其中來源又可分為 fb、G+ 以及兩者同時的情形。條件代表使用者在當前狀態可能採取下一步點擊來源的動作，其中網底為灰色的代表為已 Like+1 過的狀態。

## 2.2. 重複點擊分析

除了使用者介面的整合，最重要的部份還是在於資料的彙整。在前一小節已稍微提到，由於 Like 及+1 的資訊會分別回傳各自的服務；在第三方服務取得這些資料時，若有同一人針對同一資源同時使用 Like 及+1，將會造成重複計數之問題產生。因此在本節當中，將要進行探討這個問題的源由及解決方式，整體的問題以於前一小節說明

為了能有效統整這些計數，第三方服務可自行解決衝突與整合問題；但考慮到未來可將統計資料擴充到更多的第三方服務使用，因此在本文提出一個能夠在第三方服務與在 Facebook 與 Google+ 間的整合處理計數的伺服器 — MI (Mashup Integration) 服務。MI 服務會先將第三方服務受到使用者點擊的資訊進行來源分析與處理計數的整合，再分別送自各自的服務提供者。

在解決重複計數的過程當中，將點擊資訊儲存於 MI 服務中，也可便於計算各來自的服務計數率，作更多的應用，如廣告投放的選擇等等。因此除了解決重複計數以及相關的應用，本文中更整理出以下的數學公式。

**Formula 1: 特定網路資源，不重複使用者的總點擊數**

$$\begin{aligned}
 n(\text{URI}) &= \sum_{i=1}^n \text{Count}(\text{URI}, \text{Source}_i) \\
 &\quad - \sum_{i,j: i < j} \text{Count}((\text{URI}, \text{URI: UserID}, \text{Source}_i) \\
 &\quad \cap (\text{URI}, \text{URI: UserID}, \text{Source}_j)) \\
 &\quad + \sum_{i,j,k: i < j < k} \text{Count}((\text{URI}, \text{URI: UserID}, \text{Source}_i) \\
 &\quad \cap (\text{URI}, \text{URI: UserID}, \text{Source}_j) \cap (\text{URI}, \text{URI: UserID}, \text{Source}_k) \dots \\
 &\quad + (-1)^{n-1} \text{Count}(\bigcap_{i=1}^n (\text{URI}, \text{URI: UserID}, \text{Source}_i))
 \end{aligned}$$

**Formula 2: 特定使用者，不雙重擊的總點擊數**

$$\begin{aligned}
 n(\text{UserID}) = & \sum_{i=1}^n \text{Count}(\text{UserID}, \text{Source}_i) \\
 & - \sum_{i,j:i < j} \text{Count}((\text{UserID}, \text{UserID:URI}, \text{Source}_i) \\
 & \cap (\text{UserID}, \text{UserID:URI}, \text{Source}_j)) \\
 & + \sum_{i,j,k:i < j < k} \text{Count}((\text{UserID}, \text{UserID:URI}, \text{Source}_i) \\
 & \cap (\text{UserID}, \text{UserID:URI}, \text{Source}_j) \\
 & \cap (\text{UserID}, \text{UserID:URI}, \text{Source}_k)) - \dots \\
 & + (-1)^{n-1} \text{Count}\left(\bigcap_{i=1}^n (\text{UserID}, \text{UserID:URI}, \text{Source}_i)\right)
 \end{aligned}$$

**Formula 3: 特定使用者，對於 Source 的點擊百分比**

$$\text{Click Rate}(\text{UserID}, \text{Source}) = \frac{\text{Count}(\text{UserID}, \text{Source})}{\sum_{i=1}^n \text{Count}(\text{UserID}, \text{Source}_i)} \times 100\%$$

**Formula 4: 特定網路資源，對於 Source 的點擊百分比**

$$\text{Click Rate}(\text{URI}, \text{Source}) = \frac{\text{Count}(\text{URI}, \text{Source})}{\sum_{i=1}^n \text{Count}(\text{URI}, \text{Source}_i)} \times 100\%$$

**Formula 4-1: 特定第三方服務，對於 Source 的點擊百分比**

$$\text{Click Rate}(\text{Service}, \text{Source}) = \frac{\sum_{\text{URIs}} \text{Click Rate}(\text{Service:URI}, \text{Source})}{\text{Count}(\text{Service:URI})}$$

**Formula 5: 特定網路資源，對於 Source 的點擊百分比(含該資源點擊使用者之平均習慣)**

$$\begin{aligned}
 \text{Click Rate}'(\text{URI}, \text{Source}) & \\
 & = \frac{\text{Count}(\text{URI}, \text{Source})}{\sum_{i=1}^n \text{Count}(\text{URI}, \text{Source}_i)} \\
 & \times \frac{\sum_{\text{Users}} \text{Click Rate}(\text{URI:UserID}, \text{Source})}{n(\text{URI})}
 \end{aligned}$$

**Formula 5-1: 特定第三方服務，對於 Source 的點擊百分比(含該資源點擊使用者之平均習慣)**

$$\begin{aligned}
 \text{Click Rate}'(\text{Service}, \text{Source}) & \\
 & = \frac{\sum_{\text{URIs}} \text{Click Rate}(\text{Service:URI}, \text{Source})}{\text{Count}_{\text{Service}}(\text{URI}) \times 100} \\
 & \times \frac{\sum_{\text{Users}} \text{Click Rate}(\text{Service:UserID}, \text{Source})}{n(\text{Service})}
 \end{aligned}$$

參數定義：

URI：代表網路資源之識別

UserID：代表使用者之識別

Source：代表來自於服務提供者，如

Facebook、Google+

Service：代表第三方服務之識別

URI:UserID：代表網路資源對應之使用者

UserID:URI、Service:URI、Service:UserID：

承上，以此類推

Formula 1 為解決重複計數的模型，也是本文最重要的部份，Formula 2 至 5 為進一步之延伸，分析出使用者的點擊次數、對於 Like 及+1 的常用點擊行為、網路資源對於特定來源的點擊百分比以及加上使用者慣用來源的因素。透過以上的數學公式，除了可以解決本文最重要的核心外，也提供第三方服務管理者更多參酌的指標，以加值第三方的應用。

## 四、混搭整合服務

### 4.1. 認證授權及協議

針對於 MI 服務收集使用者點擊資訊，一般而言，透過非官方的服務是無法直接取得使用者的資料。在收集之前，一定必須先向使用者取得同意才有辦法進行其目的；但若直接向使用者要求憑證，恐有安全及隱私上的疑慮。因此該機制以 OAuth 2.0 的標準為基礎，可不將憑證交付給該服務，而直接透過官方服務去授權給 MI 服務，同時可將權限分割，讓使用者去選擇該授予那些的權限，以達到保護使用者的安全。

除此之外，利用於 MI 服務的第三方應用服務，也必須遵守此認證授權的模式；但因考慮第三方服務與 MI 服務的關係，在第三方服務要混搭這個服務前，必須先向第三方服務簽訂協議。服務與服務間的協議除了「雙向配合」，尚有「單向配合」的協議模式；大多數而言，「雙向配合」模式須兩者服務間必須規模相當，才比較有機會互相討論交換的格式等細節，但缺點在於參與的服務商必須完整配合其細節的內容，實施運作較有困難。因此目前的網路服務大多以「單向配合」模式為主，如 Facebook、Google 等較大規模的服務，要提供服務給第三方服務混搭，都是由第三方服務主動去配合這些服務商。在使用其服務前，先閱讀開發或使用協議，再透過註冊等等的流程來進行使用；再者，考慮第三方服務內容為若為隱私資料的問題，要讓 MI 服務去存取該內容的資料以及使用者行為，必定要先透過認證授權及協議上的保護。因此，不但第三方服務要混搭開發前，甚至連使用者要註冊服務前，都必須先簽訂該服務內容的使用條款。在使用過程中，存取資料則受到認證授權的機制保障，以保證整個服務在使用上的不可否認性，提高使用者隱私資料的安全性，同時降低第三方應用服務不必要的問題產生。

至於 MI 服務與 Facebook 及 Google+ 之間的關係，考量到 MI 服務在本文的目的在於可整合使用者所有的 Like 及+1 資訊，以供第三方網站服務混搭所使用。兩者協議模式相較之下，採用雙向配合協議模式較能達到此種目的的效果。

### 4.2. 環境假設

本文的實驗，主要是進行解決前文所提及的解決 Like 及+1 在同一網路資源重複計數的問題。但由於現實環境的限制，因此在實驗實作前必須先行假設部份的條件，以達實驗的目的。以下為本實驗

的假設、初始及系統架構內容：

1. Facebook 及 Google+ 可提供 Like 及 +1 存取資料之 API
2. Google Plus +1 允許自行定義 meta tag 格式
3. 經使用者同意授權後，MI 服務可接收該使用者在 Facebook 及 Google+ 的所有 Like 及 +1 資料
4. 承 1~3，由於 Facebook 及 Google+ 目前無可能使用本文的提及的部份內容機制，因此該兩服務平台在本實驗中為實作的仿造服務
5. MI 服務與 Facebook 及 Google+ 間採用「雙向配合」協議，與第三方網路應用服務採用「單向配合」協議
6. 使用者須使用 OpenID 作為識別帳號，並遵循 OAuth 2.0 認證授權機制。

#### 4.3. 系統架構

藉由前兩小節的協議模式及環境假設後，建立出圖 11 的系統架構。

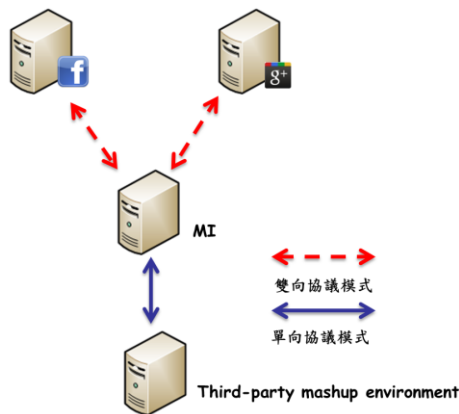


圖 11. 系統架構

在實驗中，建立出 Like 及 +1 的介面、架構及資料流的整合環境下，測試同一使用者在單一網路資源對 Like 及 +1 點擊。透過 MI 服務處理重複問題後，同時提供本文所提出之相關點擊資訊的指標，藉以證明本文所提及的整合機制確實可行。

#### 4.4. 結果分析

本文實作在於解決雙重點擊的機制，並提供能給予第三方服務判斷其網站價值之指標。本節將針對同一主題在特定期間內，不同的兩個第三方服務所點擊的狀況，進行使用本文所提出的部份整合機制；但該二案例使用知名部落格的點擊數據，無法正確判定雙重點擊以及使用者平均慣用狀況，因此在雙重點擊部份取其最小值作為範圍並以隨機產生，使用者平均習慣部份將 (fb, G+) 分為 (10%, 90%)、(20%, 80%)……、(90%, 10%) 等情形進行驗證，在本文將以 (40%, 60%) 及 (80%, 20%) 兩種情形的結果為例；但因每個資源平均習慣並非相

同，所以上述比例也將以 ±10% 的範圍隨機產生。

表 2. 第三方服務的點擊狀況(案例一)

URI	fb 點擊量	G+ 點擊量	雙重點擊量
1	3	2	2
2	6	1	0
3	1	3	0
4	0	1	0
5	0	1	0
6	2	1	0
7	10	1	0
8	16	3	2
9	3	2	2
10	30	1	1
11	66	2	1
12	13	4	1
13	17	5	4
14	15	1	1
15	1	1	1
16	5	1	1
17	0	2	0
18	0	2	0
19	17	8	4

表 3. 第三方服務的點擊狀況(案例二)

URI	fb 點擊量	G+ 點擊量	雙重點擊量
1	1	2	1
2	2	4	0
3	4	4	3
4	3	3	3
5	4	1	0
6	3	2	2
7	5	9	2
8	1	1	1
9	7	4	2
10	2	2	1
11	3	4	0
12	4	1	0

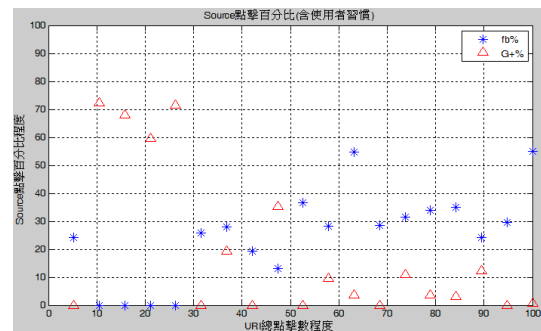


圖 12. Source 點擊百分比 — 案例一(平均習慣 40-60%)

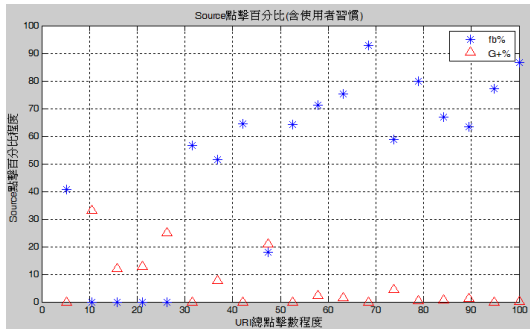


圖 13. Source 點擊百分比 — 案例一(平均習慣 80-20%)

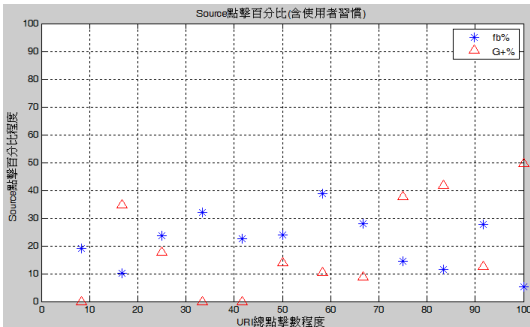


圖 14. Source 點擊百分比 — 案例二(平均習慣 40-60%)

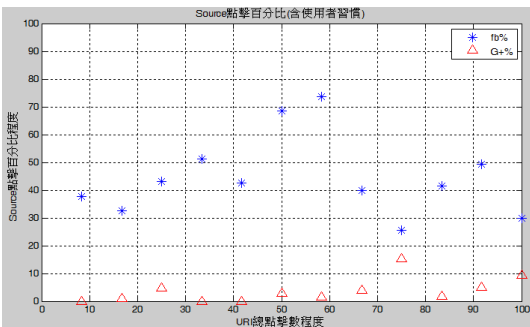


圖 15. Source 點擊百分比 — 案例二(平均習慣 80-20%)

使用該整合機制後，結果如表 2、3 及圖 12 至 15 的內容所示。分別加入使用者平均習慣後，在案例二中，發現到(40%，60%)的 fb 優於 G+ 的資源約有 8 個，(80%，20%)約有 12 個；案例一也有類似的情況，過程看出使用者習慣對於最後結果所產生的影響。針對同一主題，也可以評估何者的服務較受歡迎，可作為廣告投放等應用的依據。因此利用本文提出之方法，確實可有效助於第三方服務管理的評估及發展。

## 五、結論

在網路混搭環境盛行下，造成資料過度雜亂的問題產生。本文將以社交外掛作為方向，分析同時混搭 Facebook 的 Like 及 Google Plus +1 對於使用者及第三方服務管理者的角度所面臨到的問題。在該問題的情境下：(1) 使用者可能降低使用社交外掛的意願度；(2) 第三方服務管理者缺乏整合後的資訊。因此藉以上問題，本文遂提出整合社交外掛的機制，進而提出可提供第三方服務評估本身網站的價值或加值的服務。

在整合機制中，除了提出整合性的社交外掛，解決使用者在使用上的操作流程，也解決後端資料可能面臨到使用者雙重點擊、社交服務資料流的問題。因此後端資料都會記錄於 MI 服務當中，解決第三方服務管理者整合的困擾。

未來若有類似社交服務興起，MI 服務也可繼續擴充，亦或者產出更多整合後的資訊。雖本文所提出之方法確實可解決在使用或應用上碰到的困難，但未針對整體機制的資料傳遞或網路服務功能無法正常運作的狀況，因此未來必須得在針對可靠度的方向進行探討，以提升整體服務的安全性及可靠度，讓本文所提出的方法能夠更加完全。同時繼續朝向使用者行為(如興趣、年齡、性別等)分析的方向繼續延伸，甚至落實於應用當中。

## 參考文獻

- [1] Application Programming Interface [http://en.wikipedia.org/wiki/Application\\_programming\\_interface](http://en.wikipedia.org/wiki/Application_programming_interface)
- [2] Open Graph Protocol <http://ogp.me/>
- [3] Facebook Developer <http://developers.facebook.com/>
- [4] Google+ Developer <https://developers.google.com/+/>
- [5] J. Meng and J. Chen, "A Mashup Model for Distributed Data Integration" Proceedings of the 2009 International Conference on Management of e-Commerce and e-Government, IEEE Computer Society, pp. 168-171, 2009.
- [6] P. Austel, S. Bhola, S. Chari, L. Koved, M. McIntosh, M. Steiner, S. Weber, "Secure Delegation for Web 2.0 and Mashups," Proceedings. of the Workshop on Web 2.0 Security and Privacy, 2008.
- [7] Chunying Zhou, Huajun Chen, Tong Yu, "Social network mashup: Ontology-based social network integration for statistic learning," in Proceedings of the 2008 International Conference on Information Reuse and Integration, IEEE Computer Society, pp. 168-171, 2008
- [8] Mata, F., Pimentel, A., Zepeda, S., "Integration of Heterogeneous Data Models: A Mashup for Electronic Commerce," in Electronics, Robotics and Automotive Mechanics Conference (CERMA), 2010
- [9] Hao Han, Tokuda, T., "A Method for Integration of Web Applications Based on Information Extraction," Proceedings of the 2008 International Conference on Web Engineering, IEEE Computer Society, pp. 189-195, 2008
- [10] K. Seamons, M. Winslett, T. Yu, L. Yu and R. Jarvis, "Protecting Privacy during Online Trust Negotiation," In 2nd Workshop on Privacy Enhancing Technologies, 2002.