引用 XML 儲存系統建構以內容為基礎之訊息撮合及通報系統

Building a Content-Based Event Brokering Service with an XML Storage System

陳志達　蔡尚榮　杜勇進　陸志恆
成功大學電機工程研究所
{andypony, tu, lok}@turtle.ee.ncku.edu.tw

摘要

本文研究如何運用發佈/訂閱之非同步通訊模型及 XML 儲存系統來實現一個以內容為基礎之訊息撮合及通報系統。此系統為運作於網際網路中提供多用途的訊息撮合及主動通報之應用服務，能針對不同的應用需求快速的建立不同的應用服務而不需要撰寫程式。對資訊提供者而言，可以透過方便的介面輕易的提供資訊；對資訊消費者而言，能夠準確的訂閱想要的訊息，並且方便的取得。系統以 XML Object 來作為訊息的載體，使得訊息具有彈性的內容定義、自給自足、高可攜性的特點。並引用 XML 儲存系統來管理訊息，對長期性訊息的維護提供一個良好的解決方案。

關鍵字：非同步通訊模型，XML 儲存系統，訊息撮合，XML Object

Abstract

In this thesis, we apply an asynchronous publish/subscribe communication model and the XML technology to build a content-based event brokering service with an XML Storage System. Our system is called XEBS. The XEBS provides a general-purpose event brokering and notification service on the Internet. Varieties of event notification services can be built by the XEBS according various application requirements without programming efforts. The information providers can easily publish event messages via convenient user interfaces. On the other hand, information consumers can subscribe interested event channels accurately. We use XML as the event messages format to encapsulate the message as an XML object to achieve the advantages of flexible content definition, self-contained data description, and high portability. The XEBS uses the XML Storage System as the message exchange center to provide effective management for persistent event messages.

Keyword: publish/subscribe communication, event brokering service, XML Storage System, XEBS, XML object

1. 研究動機

由於 Internet 的普及，廉價，快速，互動性，很多的商業資料通訊也利用 Internet 來進行網路上的新聞發佈，網路拍賣，交友相親，求職求才，電子報刊，網路書店，股票報價···等等的應用，資訊提供者(information provider)透過網路，可以很輕易、快速、即時的發佈他們的訊息給資訊消費者(information consumer)。但長久以來，資

訊提供者與資訊消費者之間的通訊模式是很被動的，是以“瀏覽”為主要的運作模型。以 Web 系統來講，資訊提供者把資訊以 Web page 的方式放在 Web server 上，資訊消費者必須連上 Web server 查看。以後是否有新的資訊被發佈，資訊是否有所更新改變，資訊消費者無從得知。

比如說，一個新聞發佈的網站，提供讀者很多各式各樣的新聞。但是對讀者來說，真正有用的資訊卻是極少，反倒是充斥許多的雜訊。讀者若想要看自己感興趣的新聞，則必須要靠自己去搜尋。搜尋時，只能按照新聞的分類（Channel）和標題（Subject）來尋找或透過站內全文搜尋引擎來作全文檢索（Full-text searching）。這種資訊提供者與資訊消費者通訊方式主要的缺點是沒有效率，網站上可能充斥許多雜訊，是讀者沒有興趣的，讀者必須花時間跟精神過濾。所以資訊消費者每次都得重復相同的動作，去尋找他感興趣的訊息，資訊消費者若想獲得最新的資訊，必須經常的檢查是否有新的訊息被發佈，浪費寶貴的時間。這種資訊取得的方式是屬於拉入式(pull)的方法，若能以推出(push)的方式，由資訊系統主動將有用的資訊送到資訊消費者手中應是較有效率的方法。

訊息通報服務（Event Notification Service）是近年來漸受重視之新技術，其中 Publish/Subscribe 通訊模型是最常用來解決這一類問題的模型。其理念是要把資訊消費者感興趣的資訊，主動的推播(push)給使用者。本論文所要探討的，即如何利用 Publish/Subscribe 通訊模型與 XML 技術[1]，來建立一套資訊通報系統，作為資訊提供者與資訊消費者之間的橋樑，幫助他們發佈、撮合、訂閱及通報訊息，以達到更有效率、更自動化以及更便利的通訊方式。

2. 系統架構與規劃

2.1 Publish/Scribe 事件撮合系統

圖 1 說明了 Publish/Subscribe System 的基本架構[2][3]。它由一個或多個 Event Sources (ES)、一個 Event Brokering System (EBS)以及一個或多個 Event Displayers (ED)所組成。Event Sources 可能是人或某些特殊的監控裝置。Event Source 產生 Events，Events 可以是 Event Source 監測真實世界的某些變量(如：股票價格、病人的心跳)、也可以是對真實世界的描述(如：新聞報告、電子期刊)。Event 被發佈到 Event Brokering System 以後，EBS 把它們與 subscribers 所定義的 subscription 集合作撮合比對。如果系統找到符合使用者 subscription 的 Event，這一 Event 將會被轉送到使用者的 Event Displayer 上，Event Displayer 負責把 Event 呈現給使用者。Event Displayer 可能是 PC 也可能是行動式的 device，如 PDA、手機等。

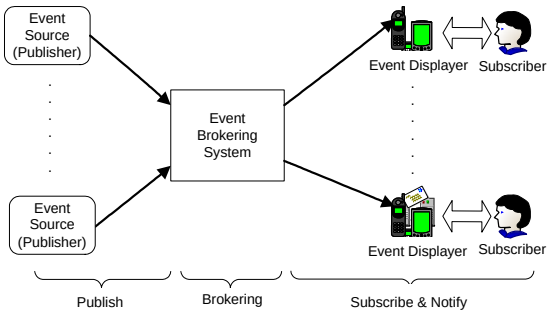


圖 1. Publish/Subscribe 系統基本架構

如上所述，Publish/Subscribe 系統中存在兩種角色的使用者：

1. **Publisher (Event Source)**：Event 的提供可能是人，也可能是某種監測裝置。
2. **Subscriber**：Event 的消費者，消費的方式是以記錄在 subscription 的一組規則去定義他們感興趣的 Event。

以及四種基本功能：

1. **Publish**：讓 Publisher 傳送他們的 Event

到系統中。

2. **Subscribe**：讓使用者註冊他們感興趣的 Event，常用的註冊方法為，使用者用系統所規定的 subscription language 把 rule 記錄在 subscription 中。
3. **Brokering**：Publish/Subscribe 系統最重要的功能[4]。根據使用者的 subscription 為使用者過濾出他們有興趣的 Event。
4. **Notify**：為 Event Service 重要的功能，把過濾出來的 Event 主動的推播到使用者指定的 Event Displayer 上。

2.2 系統架構規劃

根據 2.1 節所述 Publish/Subscribe 事件撮合系統的四個基本功能，我們知道我們的系統 XEBS (XML Event Brokering System)必須具備下面幾個部份：

1. **Publish/Subscribe Engine & Interface**：
   Publish/Subscribe Engine 負責處理 publish 及 subscribe 的動作。前端使用者介面則獨立成 Publish/Subscribe Interface，這個介面提供使用者一個便利的方式來向後端 Publish/Subscribe Engine 發出請求。
2. **Subscription Management**：
   使用者的 Subscription 經常要用來撮合比對，必須要把它們放到長期性的儲存裡面。在我們的系統中應有一 Subscription database 來負責 subscription 的儲存與管理的任務。
3. **撮合比對(Brokering)的機制**：
   Brokering 是本系統最重要的功能，它是負責過濾出使用者感興趣的 Event。在我們的系統中，我們規劃了 Matching Daemon 這個背景執行的比對精靈來負責此一重要的任務。
4. **Notification 的機制[5]**：
   我們希望 XEBS 系統能支援多樣化的接收裝置，包含有線及無線通訊裝置，並且可以根據 Event Displayer 的不同展現不同的效果[13]。因此規劃了 Event Presentation 元件及數個針對不同通訊協定的 Event Delivery Gateway，負責把訊息轉換成各種 Event Displayer 能解釋的格式，發送到訂閱者手中[14]。目前的 Event Delivery Gateway 包括：SMTP Gateway、HTTP Gateway、WAP Gateway(SMS/EMS/MMS/WML)、Instant Messaging Gateway、GPS Gateway。

以上是 Event Service 的基本需求。根據對訊息的儲存與維護的需求，如對長期性、參考價值高、重用性高的訊息，我們希望訊息被撮合轉送以後，系統仍然可以保留一份訊息的記錄，作長期的維護。所以 XEBS 系統底層需要有一穩定的儲存系統(Storage System)來應付此一需求，所以系統裡又增加了下面部份：

5. **Event 的儲存管理子系統**：
   Event Container 是規劃用來儲存管理 Event 的子系統。由於本系統的訊息是以 XML 作為儲存格式，在儲存系統方面，選擇了由本實驗室所開發的 XML Storage System - XDS (XML Document Storage)[6]。XDS 是一套適用於 XML 文件的儲存、管理與查詢的系統。它以 XML Object 的觀念，能對資料作有效的維護管理及遷移(Migration)。引用 XDS 來設計 Event Container，主要是利用它對 XML 文件儲存管理、遷移及查詢的能力。

6. **Event Reader：**

Online Event Reader 是一個以 JSP 和 Servlet 實作的網頁介面，幫助使用者以瀏覽器來瀏覽系統所收集的訊息。以及提供查詢的介面供使用者搜尋過去曾發佈過的訊息。

為了系統運作的需要，必須要維護一些必要的資料，包括：系統設定參數、系統所掛載之應用資訊、使用者相關資料、使用者訂閱條件、撮合記錄。我們系統規劃了三個管理元件：System Information Manager、Service Manager 及 Account Manager，System Information Manager 負責系統運作所需的設定資訊；另外系統具有產生不同應用服務能力，系統應用服務的產生、刪除、設定資訊都交由 Service Manager 元件來負責。由於我們的系統是直接面對 end-user，因此必須要具有儲存及管理使用者帳戶資料的能力，**Account Manager** 便是負責這一任務。

綜合上述對系統元件及其所負責之功能劃分，把它們組合成一個完整的系統，我們可以將其元件關係畫成如圖 2 之階層架構圖：

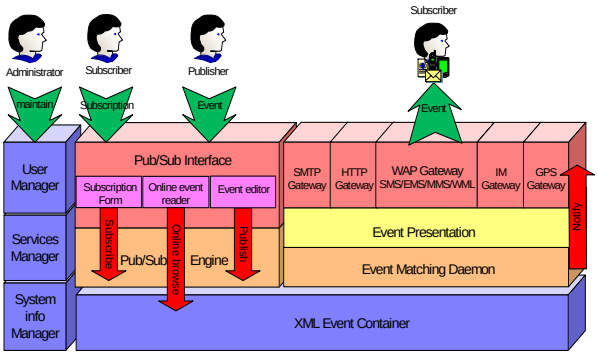


圖. 2 XEBS 系統階層架構圖

從圖 2 可以看到，除了 Publish/Subscribe 系統原有的兩個使用者角色：Publisher、Subscriber 以外，系統還多了一個管理者的角色。管理者的工作是有三項：1. 系統參數的設定、2. 使用者帳號的管理、3. 應用服務的管理。這三項工作分別對應到三個系統管理的元件。

如果以系統元件的功能性來劃分，可以分成七個功能模組，如圖 3 所示：

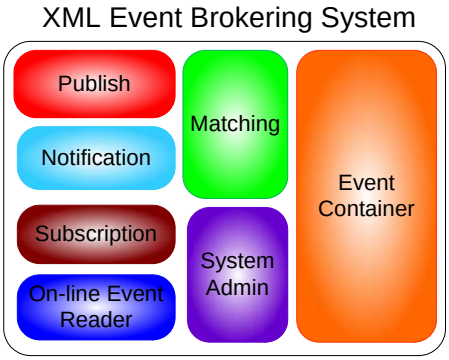


圖. 3 XEBS 系統功能模組

七大功能模組的所負責的任務定義如下：

Name	Function Description
Publish	➢ 提供一個方便的 user interface 以幫助 publisher 編輯 event ➢ 產生符合 XEBS 格式規範的 XML event ➢ 把 XML event 儲存在 Event Container 中
Notification	➢ 把符合 subscriber 的 event，根據 subscriber 的設定套上不同的 presentation template[12][13] ➢ 把 notification 透過 Delivery Gateway，push 到 subscriber 所設定的 notification device (Event Displayer)中
Subscription	➢ 提供一個方便的 user interface 以幫助

	subscriber 下達他們的訂閱條件(subscription rules) ➢ 對 user subscription 的儲存與管理(add, update, delete)
On-line Event Reader	➢ 提供 user 線上瀏覽與查詢 Event Container 內訊息的 user interface
Matching	➢ 根據 user subscription 比對 Event Container 內的訊息，找出 subscriber 所訂閱的 Event ➢ 採用週期性(periodic)的方式，每隔一段時間始動 Matching 程序
System administration	➢ User account 與 profile 的管理 ➢ Service 的管理(create, delete) ➢ 系統參數的設定
Event Container	➢ 提供對 Event 的儲存 ➢ 提供對 Event 的管理 (add, delete, update) ➢ 提供對 Event 的查詢

表 1. XEBS 系統功能模組定義

三、設計研討

3.1 Event 格式的設計

為了實現 Content-based Subscription 的目的，訊息本身必須存在某種能讓使用者訂閱的結構[3]。本系統是採用 XML 格式來描述訊息。 XML 是以樹狀結構來表達資料之間的關係，並利用標籤 (Tag) 把內容的意義標記 (Mark Up) 起來。XML 能表達文件的結構和內容的意義，因此使用者在訂閱的時候，能下達較精準的 Subscription Rule。在訊息維護管理的問題上，我們引用 XDS 做為訊息維護管理之用。我們的 Event 是以 XML Object 作為儲存單位，一個 Event 就是一份 XML Object。所謂 XML Object[6]，它包含了對自身完整的維護管理資訊，使得儲存系統能對它作有效的管理維護(存取控制、索引、查詢、遷移)。

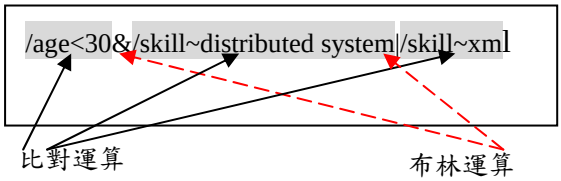
3.2 Subscription Lnaguage

一般討論 Content-based 模型大多只採用簡單的 Subscription language 來描述訂閱條件(Subscription rule)[7]。利用 XPath[11]來當作 Subscription language，使用 XPath 能指定 XML 文件樹狀結構的任一部份。引用 XPath 來作 Subscription Language[8]的主要觀念在於利用 XPath 加上比對運算來限制(Qualify)訊息中某一部份內容出現的 Pattern，也就是用 XPath 所指定路徑的內容要“符合”使用者所設定的 Pattern。

假設有一求職者履歷表的內容如下：

```
<name>Simon</name>
<sex>Male</sex>
<age>27</age>
<skill>Distributed System</skill>
<skill>xml</skill>
```

若要針對此一訊息進行訂閱，只要下達如下 Subscription Rule：



上面的訂閱條件分成三個比對運算式用兩個布林運算子連接在一起，意思是說：要找

尋年齡小於 30 歲且會分散式系統或 XML 技術的人。對於 XML 格式的訊息，使用 XPath 加上前面所說的幾種運算來當作訂閱語言，已能滿足 Content-based Subscription 的需求。

3.3 Publish/subcribe 介面的考量

Event Sources 可能人或某些特殊的監控裝置，因此有必要提供不同的呼叫的介面給 End-user 和 User-agent 使用。我們可以把這一層要提供給使用者的介面分成兩大類：

1. 提供給 End-user 使用的介面：
   針對 End-user 我們提供 Web-based 介面讓使用者可以透過瀏覽器來完成 Publish 及 Subscribe 的動作。
2. 提供給 User-agent 使用的介面：
   本系統擬提供一套以 SOAP(Simple Object Access Protocol)[9]作為通訊協定的 Language Independent API，不管開發的語言為何，都可以透過 SOAP 的協定達到異質環境下的 RPC(Remote Procedure Call)功能，未來要將本系統的功能整合到 Web Services 中，也毫無問題。

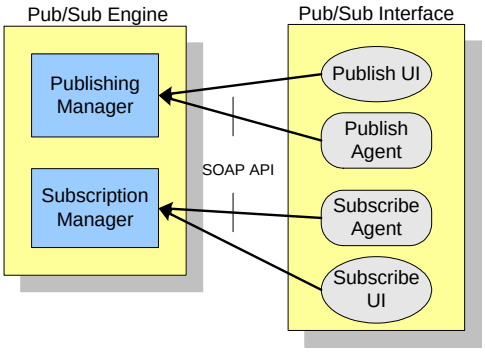


圖.4 Publish/Subscribe 介面

3.4 撮合機制設計

撮合的機制是 Event Service 一個重要的部份，主要的任務是為使用者過濾訊息。通常 Content-based Event Service 把這一功能交由一獨立的元件 — Event Filter 來負責[10]。Event Filter 能根據使用者用 Subscription language 寫成的 Subscription rule 來為使用者過濾出他們感興趣的 event。本系統利用 XPath 來當作 Subscription language，在我們引用的 XDS 已經具有以 XPath 來對資料進行查詢能力，而且有索引的功能，對資料的查詢有極高的效率。因此我們直接引用 XDS 的查詢功能來為我們比對訊息，便能馬上擁有高效率的撮合能力，不用重新實作 Filter 元件。

在我們的目標中，我們對“即時性”的要求沒有那麼高。對一般的應用來說，訊息在被發佈以後的數十秒甚至數分鐘以內被撮合轉送是可以被接受的，所以我們決定採用週期(Periodic)的方法，先把訊息收集起來，然後再一起進行撮合。為了保持彈性，我們的系統能讓系統管理者去設定撮合的週期間隔(Matching Interval)，以應付不同應用服務的需求。

四、系統的應用範圍

本系統欲建立一個應用服務的運作模型，適合之應用皆可以利用本系統提供訊息發佈、撮合及通報的服務。本系統支援同時掛載多種應用服務於系統之中，如圖 5.所示，新聞發佈、求職求才、物品拍賣、技術報告分享等系統同時透過本系統來提供服務。

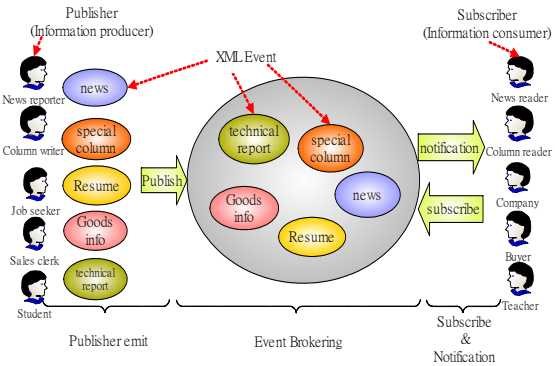


圖 5. 多個應用同時透過 XEBS 提供服務

歸納現今常用的應用服務，適合套用 XEBS 系

統模型的應用範例有：

- **自動撮合系統：**

  適用各種需要訊息撮合功能的應用（在電子商務中常稱做自動撮合系統），包含相親、求職求才、個人化的新聞、報刊訂閱、找商品、開會協調系統等等。

- **資訊共享系統：**

  提供訊息或文件的存取介面，並且具備良好的搜尋能力，讓資訊能分享給許多使用者。例如：旅遊資訊共享、論文共享系統、公文交換系統。

- **即時通報系統：**

  透過本系統所提供訊息撮合及通報的功能，可將應用所需的即時訊息不斷的發佈到系統之中，依照設定的條件來決定什麼時機該通知什麼人，例如：醫療看護系統、網管系統、股票資訊通報。

五.結論

目前網路上資訊消費者取得資訊的方式大多採取 Pull-based 的被動方式。訊息仲介服務能以 Push-based 的主動方式把資訊轉送到訂閱者手中，以達到更有效率、更方便的資訊傳播方式。本論文延續了本實驗室對 Event Brokering System 的研究並針對了之前 EBS 系統對長期性訊息的儲備及撮合效能等問題提出改善的方法。本系統成功的引用了 XML 儲存系統及相關技術來建立以內容為基礎之訊息撮合及通報系統。

XEBS 系統針對不同的應用需求，在不需要寫程式的情況之下，能快速的建立起不同的應用服務。對資訊的提供者而言，可以透過方便的介面輕易的提供資訊；對資訊的消費者而言，能夠準確的訂閱想要的訊息，並且方便的取得。實作之系統具有以下的特點：

1. 系統採用 XML Object 來作為訊息的載體，使得訊息具有彈性的內容定義與較高的互通性(Interoperability)。並且有足夠的自我描述資訊，使之成為一個獨立的資訊個體。提高了訊息的可攜性(Portability)。
2. 引用了 XML Document Storage，對長期性訊息的維護提供了一個良好的解決方案。
3. 相較過去 EBS 系統的撮合效率，系統直接採用 XDS 高速的查詢能力，有較高的撮合效能。
4. 使用者能利用 XPath 表達式達到準確的訊息訂閱。
5. 訊息有包含多媒體資訊的能力。並且訊息能針對不同的顯示裝置搭配不同的 XSL 把資訊呈現給使用者。

系統在設計實作上，成功的結合理論與實務。並從本實驗室過去的對 Event Brokering System 的研究經驗，針對 EBS 系統的缺點作出了改進，使得本系統有更高的實用價值。但是分散式系統方面的研究與應用，還有許多值得探究的地方，尤其是在現今無線通訊相當發達的情況下，未來應努力吸收新知，使系統功能日趨完善。

六、Reference

[1] World Wide Web Consortium. Extensible Markup Language (XML). http://www.w3.org/XML/.

[2] Y. Huang and H. Garcia-Molina. Publish/Subscribe in a Mobile Environment. MobiDE 01, 2001.

[3] P.Th. Eugster, P.Felber, R.Guerraoui, and A.-M. Kermarrec. The Many Faces of Publish/Subscribe.

[4] R. Strom, G. Banavar, T. Chandra, M. Kaplan, K. Miller, B. Mukherjee, D. Sturman and

M. Ward. Gryphon: An Information Flow Based Approach to Message Brokering, International Symposium on Software Reliability Engineering ’98 Fast Abstract.

[5] Antonio Carzaniga, David S.Rosenblum, and Alexander L.Wolf. Achieving scalability and expressiveness in an internet-scale event notification service. In Proceedings of the 19th ACM Symposium on Principles of Distributed Computing, Portland OR, USA, 2000.

[6] 魏朝信、蔡尚榮, A Storage System for XML Data Objects, Master thesis, Dept. Of EE, NCKU, July 2003.

[7] M. K. Aguilera, R. E. Storm, D. C. Sturman, M. Astley and T. D. Chandra. Matching events in a content-based subscription system. In Eighteenth ACM Symposium on Principles of Distributed Computing (PODC’99)(Atlanta, GA, May 4—6, 1999).

[8] Shrideep Pallickara, Geoffrey Fox and Marlon Pierce. Community Grid Labs, Indiana University. Incorporating an XML Matching Engine in Distributed Brokering Systems.

[9] W3C Note 08 May 2000. Simple Object Access Protocol (SOAP). http://www.w3.org/TR/SOAP/.

[10] Alexis Campailla, Sagar Chaki, Edmund Clarke. International Conference on Software Engineering. Efficient Filtering in Publish-Subscribe Systems using Binary Decision Diagrams. 2001

[11] World Wide Web Consortium. XML Path Language. http://www.w3.org/TR/xpath.html

[12] J. Clark. XSL Transformations (XSLT) Version 1.0. http://www.w3.org/TR/xslt. Nov 1999.

[13] S. Adler. World Wide Web Consortium. Extensible Stylesheet Language (XSL) Version 1.0. http://www.w3.org/TR/xsl/. Oct 2001.

[14] A. Carzaniga, D. S. Rosenblum and A. L. Wolf. Challenges for Distributed Event Services: Scalability vs. Expressiveness. In Engineering Distributed Objects '99, Los Angeles CA, USA, May 1999.