

以 ORM 為基礎的快速 GIS 詮釋資料查詢系統開發方法

Rapid Implementation Approach of GIS Metadata Searching System based on ORM

林嘉成

逢甲大學地理資訊系統研究中心管理部
系統工程師
Email: ricky@gis.tw

郭靜怡

逢甲大學地理資訊系統研究中心管理部
經理
Email: julia@gis.tw

摘要

GIS 詮釋資料因實際運用需求而擁有不同的資料格式，在實際系統開發上，詮釋資料的屬性多寡及複雜性將影響開發的複雜度。本文介紹以 ORM 為基礎的詮釋資料塑模方式，達到跨資料庫、跨資料來源的統一塑模，並結合如何動態組合 Entity Object 查詢條件的系統開發方法，以加速複雜的 GIS 詮釋資料查詢系統的開發。

一、前言

因地理資訊本身的複雜性，詮釋資料包含之元素特性(或屬性)通常較為多、且複雜，以 ISO 19115 的地理資訊詮釋資料為例，其資料元素即高達 4 百餘項 (Geospatial metadata, 2010)。因此，如何表達、儲存、搜尋詮釋資料，一直是 GIS 系統實作上，開發人員棘手的問題。

二、何謂詮釋資料

詮釋資料 (Metadata) 之定義為「描述資料的資料 (Data about data)」即針對已存在資料或服務之特性，建立描述性之資料，以供可能之使用者解讀，避免錯誤之資料或服務應用 (內政部資訊中心, 2009)。詮釋資料的基本角色應包含資料搜尋、資料管理與流通、資料說明等。

資料搜尋係指資料需求者可針對空間、時間或其他詮釋資料所定義的條件進行搜索，以找到符合的資料；資料管理與流通係指地理資料需由特定權責單位進行管理與流通，相關權責及流通資訊需予以紀錄；資料說明係指地理資料的品質、摘要等說明，以避免無謂的下載與交換。

在臺灣，亦有國家級詮釋資料標準的描述內容與架構，其名稱為國土資訊系統地理資料詮釋資料標準，英文名稱為 TWSMP (Taiwan Spatial Metadata Profile)，適用於我國各類空間資源之描述(內政部資訊中心, 2009)。

三、面臨的困境

GIS 詮釋資料通常以關聯式資料庫、XML 檔案或自訂格式純文字檔的型式，存放於儲存媒體上(以下稱這樣的儲存媒體為 Repository)。然而，在詮釋資料查詢系統的實作上，卻面臨 2 個主要的困境：

- 跨資料來源整合不易
- 屬性資料龐大時，系統複雜度提高

詮釋資料的資料來源可能橫跨各種不同的關聯式資料庫，或不同的檔案格式，跨資料來源的結果，使得資料整合不易，為了不同資料來源可能需要實作不同的商務邏輯，進而使系統邏輯的重用性降低。

詮釋資料的屬性資料的多寡，也會影響到系統的複雜度，複雜度的提昇，直接影響了開發的時間成本，並降低維護的可靠性。

四、解決方案簡介

為了解決上述問題，我們將資料統一以關聯式資料庫進行儲存，並以 ORM(Object Relational Mapping)為基礎的概念，提供快速塑模的方案。

ORM 提供了以 object 作為資料儲存體的概念，對應於關聯式資料庫或 XML 的 repository，object 相當於資料表的「row」或 XML 的「element」；object 的屬性相當於資料表的「column」或 XML 的「attribute」。這麼做的好處，在於提供一個快速的對應方法，將 repository 快速塑形為商務模型，系統開發時，僅需關注在商務模型的操作上。若資料來源有所變更，亦不影響系統操作商務模型的這一層關聯。

然而，使用 ORM 之後帶來了新的問題。傳統的查詢概念，是透過 SQL 指令進行資料庫查詢，或以 XPath 來進行 XML 的尋覽。上述兩種方式在撰寫複雜的查詢指令時擁有極高的彈性，但以 object 為基礎的物件查詢上，卻相較下缺乏彈性。

為了解決這個問題，.Net Framework 3.5 所推出的 Language-Integrated Query (LINQ) 可作為解決方案，它可針對強型別的物件集合撰寫物件化查詢。

除此之外，本文亦針對複雜的詮釋資料，介紹了如何動態的產生 LINQ 查詢語法的方式，以簡化複雜的物件查詢語法的撰寫，並提高查詢邏輯的重用性。

五、實際作法

為了進行資料物件化，並希望提高商務邏輯的重用率，本文以 3-tier 架構為基礎，以區分資料層、商務層的個別責任。

3-tier 架構是一種設計模式(design pattern)，分

為展示層(Presentation tier)、邏輯層(Logic/Business tier)、及資料層(Data tier)。其主要目的在於界定明確的介面，使得各層擁有各自的責任。各層的實作技術可被抽換，由另一種實作方式實作該層的責任。

以此架構進行說明，若將各式 repository 作為 Data access layer 的基礎資料來源，其整體結構可以下圖表示：

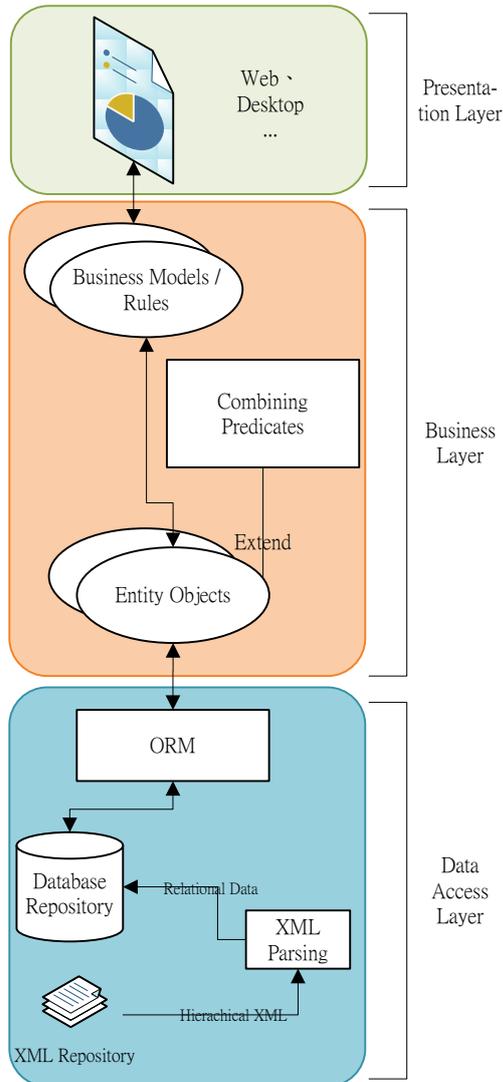


圖 1 ORM 為基礎的詮釋資料查詢系統架構

關聯式資料庫為基礎的資料整合

本方案以 ORM 為基礎，因 ORM 以關聯式資料庫為核心，所以相關的詮釋資料以整合至資料庫儲存較佳。

從圖 2 由下而上來看，若資料來源為 XML 資料檔，將先進行一個 XML Parsing 的步驟，將階層式的 XML 結構，轉換為關聯式資料的結構。若資料原本即為關聯式的資料結構，則無需此 parsing 的步驟。

Object-Relational Mapping

ORM 是一個高度程式語言相關的機制，將關聯式資料庫的資料，轉換為物件導向程式語言的物件類別。

關聯式資料庫的選擇則可以很多元，不同的資料來源可以透過不同的資料提供者(provider)實現與 ORM 串接的可能，所以只要該資料庫有相對應的 provider，就可以進行整合。因 ORM 與程式語言相關，這裡我們以 .Net Framework C# 為物件導向語言基礎，採用的 ORM 技術，為微軟的 ADO.NET Entity Framework (Microsoft, 2006)。

透過 ORM 的自動化 mapping 機制，可以得到物件化的強型別物件類型- Entity Object。此時，複雜的物件屬性已經透過 ORM 自動產生，也已可供商業邏輯進行使用。

擴充 Entity Object

一般而言，Business Model 可以直接操作 Entity Objects 進行資料邏輯的應用，並使用 LINQ(Microsoft, 2007)進行查詢語法的撰寫。

但直接操作的結果，通常將「資料的操作邏輯」撰寫於個別分散的 method 中，可能會使得操作邏輯難以重用，在複雜的 GIS 詮釋資料中更是如此。

為了解決這個問題，本文建議了 Combining Predicates (Meek, 2008)的方案。

Combining Predicates 的主要目的在於希望可以動態組成集合物件的 LINQ 查詢語法(即 Predicate, 述詞)的查詢邏輯，以增加重用性；並且維持強型別的特性，在編譯時期即可發現語法上的錯誤。而實作的重點，則是利用 .Net Framework 的泛型及 Extension Method 特性，將 Entity Object 本身擴充之，以增加物件的 Predicates 串接。

下列範例，即為實作 Combining Predicates 後的應用，藉以串接查詢條件，得到類似於 SQL 語法的彈性：

```

/// <summary>
/// 以DocumentMetadata 為被查詢物件，
/// 建立Predicate的查詢語法串接，
/// 使得DocumentMetadata的查詢邏輯可被重用。
/// </summary>
/// <param name="criteria">查詢條件</param>
/// <returns>查詢邏輯語法</returns>
internal static Expression<Func<DocumentMetadata, bool>>
    GetPredicate(DocumentSearchCriteria criteria)
{
    //建立Predicate
    Expression<Func<DocumentMetadata, bool>> predicate = o => true;

    //判斷某條件是否為空值，如果否，則進行查詢語法的串接
    if (!string.IsNullOrEmpty(criteria.DocClassID))
        predicate = predicate.
            And(o => o.DocClassID.Equals(
                criteria.DocClassID,
                StringComparison.CurrentCultureIgnoreCase));

    //進行第二個條件的判斷
    if (!string.IsNullOrEmpty(criteria.ProjectID))
        predicate = predicate.
            And(o => o.ProjectID.Equals(
                criteria.ProjectID,
                StringComparison.CurrentCultureIgnoreCase));

    return predicate;
}

```

圖 3 Combining Predicates 的應用範例

上圖範例中，我們假設詮釋資料的型別為 DocumentMetadata，並希望透過名為 GetPredicate 的方法來取得針對 DocumentMetadata 作為查詢對象的動態查詢條件的組合。GetPredicate 接受一個參數，型態為 DocumentSearchCriteria，內含相關的搜尋條件，並回傳型態為 Expression 的述詞結果。

接下來，針對 DocumentSearchCriteria 的搜尋條件，逐一的將查詢邏輯組合至述詞當中，這裡允許以 AND 或 OR 的邏輯來串接。

撰寫商務邏輯並展示成果

Business Model 透過已擴充的 Entity Objects，可針對其任何屬性自由撰寫查詢條件，以滿足查詢系統的商業邏輯，配合各種使用者介面的展示技術，將結果回傳至前端展示層進行展示。

圖 4 實作成果展示 - 詮釋資料查詢條件

完成日期	比例尺	坐標系統	部門	專業	人員	檔案名稱	檔案格式	關鍵字
2009/07/22	25000分之1							群組影像
2009/08/18		TWD67	監測管理 部	專業		99年現地 資料蒐集 暨觀測站 維護管理 計畫	SHP	觀測站

圖 5 實作成果展示 - 詮釋資料查詢結果

五、結論

GIS 詮釋資料所擁有的龐大屬性結構，在 ORM 機制下，可跨資料來源，並快速產生強型別的 Entity Object 以供商業邏輯使用。透過 Combining Predicates 方式，可產生具有彈性、且可重用的物件化查詢語法。

以上方法可有效降低詮釋資料查詢系統在開發上的時程及複雜性，並提高未來維護的可靠性。

參考文獻

- [1] Geospatial metadata. (2010年7月29日). 2010年8月16日 擷取自 WIKIPEDIA: http://en.wikipedia.org/wiki/Geospatial_metadata
- [2] Meek. (2008年5月2日). LINQ to Entities: Combining Predicates. 2010年8月16日 擷取自 MSDN Blogs: <http://blogs.msdn.com/b/meek/archive/2008/05/02/linq-to-entities-combining-predicates.aspx>
- [3] Microsoft. (2007年2月). LINQ: .NET Language-Integrated Query. 2010年8月16日 擷取自 MSDN: <http://msdn.microsoft.com/en-us/library/bb308959.aspx>
- [4] Microsoft. (2006年6月). The ADO.NET Entity Framework Overview. 2010年8月16日 擷取自 MSDN: Microsoft Developer Network: [http://msdn.microsoft.com/zh-tw/library/aa697427\(v=VS.80\).aspx#ado.netenfrmovw_topic4](http://msdn.microsoft.com/zh-tw/library/aa697427(v=VS.80).aspx#ado.netenfrmovw_topic4)
- [5] 內政部資訊中心. (2009年9月). 詮釋資料標準. 2010年8月16日 擷取自 國土資訊系統標準制度入口網站: <http://standards.moi.gov.tw/giss/front.progress.htm>