

第2章 文獻探討

本章將介紹本研究所應用到的相關知識及技術背景與功能，第一節介紹網路服務現況與發展趨勢，第二節介紹網路服務核心技術探討，第三節介紹 XML 與關連資料庫之間資料轉換，第四節介紹銀行公會訊息標準。

第1節 網路服務現況與發展趨勢

網路服務是一種動態的電子商業模式，企業透過網際網路向外擴展基礎建設，使用多元化交易方式，將商業流程自動化，創造出更有彈性且有效的商業模式。使用網路服務，企業得以根據業務需求，動態發現並引入商業功能，而不需大幅更動現有業務流程整合體系。根據 Tidwell (2001)的定義：網路服務為一個以標準網際網路技術建立的程式介面，允許使用者可透過網路來取某應用程式的功能。網路服務是以 XML 為描述語言，並以 XML 檔做為被呼叫的訊息格式，不論在何種平台或是以何種語言開發的，只要能夠透過網路被另一個程式呼叫，稱為網路服務 [McCright, 2001]。

2.1.1 網路服務類型

綜合當今的網際網路應用以及網路服務的特點，網路服務實施領域可以分為四類 [柴曉路，民 89]：

1. 商業導向網路服務(Business-Oriented Web Services)

商業導向網路服務針對的是企業導向應用服務包括企業內部的企業資源規畫(Enterprise Resource Planning, ERP)系統，企業間的 SCM/CRM 等系統。當這些系統以網路服務的形式在網路(Internet 和 Intranet)中出現時，企業內的應用整合將更加容易，而在企業間的眾多合作夥伴的系統聯接也將不再是無法完成的工作。目前現有的解決方案和產品的提供商有 Bowstreet、Epicentric 等。

2. 顧客導向網路服務(Consumer-Oriented Web Services)

顧客導向網路服務針對的是原先的企業對顧客(Business to Customer, B2C)

的網站的改造，為這些瀏覽導向網路應用(Browser-Oriented Web Application)增加了網路服務的應用介面，使得第三方的桌面工具或其自身提供的增值的桌面工具能夠利用更優秀的使用者介面提供跨越多個 B2C 服務的桌面服務。使用者將更為方便運用網路資源，獲得更加便捷的服務。比如我們完全就可以在個人理財桌面系統中整合(呼叫)Internet 上的股票價格查詢的網路服務、機票預訂的網路服務等，使得個人理財應用的自動化程度更高。

3. 設備導向網路服務(Device-Oriented Web Services)

設備導向網路服務的使用終端設備一般是手持設備和日用家電，對於前者而言，可以在不用修改網路服務的體系架構的前提下，令先前的網路服務支援除 PC 以外的各種終端設備，比如個人數位助理(Personal Digital Assistant, PDA)、手機(Smart Phone)等。如此，那些天氣預報服務、Email 服務、主動訊息服務等將更為有效和便捷。而後者對於日用家電，則可能是一個市場的啟動期，有了網路服務作為基礎框架，智慧型的日用家電將真正獲得標準的支援，從而有了廣泛使用的可能。

4. 系統導向網路服務(System-Oriented Web Services)

傳統意義上的系統服務，比如使用者權限認證，系統監控等，如果被遷移到全球範圍的網際網路上，或者企業內部網路上，其作用範圍將從單個系統或局部網路拓展到整個企業網路或整個網際網路。如此，基於同一系統服務的不同應用將得以在整個網際網路環境中部署，譬如跨國企業的所有線上服務可以使用同一個使用者權限認證網路服務。

2.1.2 網路服務之商業模式

網路服務要能夠實際應用於企業系統中，需發展完整且具效益的網路服務付費機制(Payment Models)，其付費機制大致可分為四種模式〔Deitel, 2003〕。

1. 免費使用(Free of charge)

免費提供給使用者使用，目的是在於推廣網路服務的應用，服務無關於

商業交易行爲，服務多半是功能限制性的服務，例如：應用於新聞標題報導或個人行事曆的網路服務。這類網路服務沒有提供服務水準協定 (Service-Level Agreements¹，SLA)及服務品質保證(Quality of Services²，QoS)等功能，大部分設計的目的，只在於推展網路服務的應用。

2. 以次計費(Pay-per-use)

根據服務要求(Service Request)次數計費，以一個完整交易(Integrity Transaction)爲收費單位，顧客使用服務後，依使用次數收費。此模式對於服務提供者有部分的風險，提供網路服務的業者無法確認顧客是否真的付費。

3. 會員入會(Flat-fee subscription)

顧客申請成爲網路服務的會員，固定繳交年費、會員費或訂閱費。在一定訂閱期內，無限制使用服務，當顧客超過訂閱期時，服務提供者可通知顧客重新訂閱服務。目前提供網路服務業者大都採取此模式運作。

4. 終身服務(One-time charge)

顧客只要購買一次，就能夠永遠擁有此服務的使用權。在實際現實生活中，此模式無法運作的，因爲服務提供者必須永遠保持網路服務的正常運作，即使服務已經過時或對使用者無任何效益，但基於永遠保持服務的運作，則維護成本將隨時間愈久愈大，幾乎趨近無窮大。

2.1.3 網路服務之應用發展

目前 Web Services 的發展，主要有下列三種應用形式：服務對顧客 (Service-to-Consumer，S2C)、服務對企業(Service-to-Business，S2B)、服務對企業員工(Service-to-Employee，S2E)的網路服務 [Deitel，2003]。

1. 服務對顧客的網路服務(S2C Web Services)

服務對顧客(的網路服務屬於免費使用的 Web Services，目的推廣其應

¹ 服務水準協定(Service-Level Agreements)是服務提供者與服務使用者明確簽署的服務合約，雙方須遵守合約之規定。

² 服務品質(Quality of Services)包含一般作業回應時間、尖峰作業回應時間及錯誤容忍度等。

用，使 Web Services 成爲新商業銷售管道，成功案例爲 XMethods³，提供顧客追蹤聯邦快遞(FedEx)包裹運送狀況的 Web Services，並與顧客建立良好關係，提高其忠誠度。另一成功案例爲 Microsoft.Net Passport⁴，允許顧客使用單一簽入(Single-Sign-On)服務，只須使用一個共用帳號，就能登入到 Hotmail、MSN、Passport 網站及其它相關 Microsoft 網站，減少記憶多個帳號的麻煩，避免資料不一致的情況發生。

2. 服務對企業的網路服務(S2B Web Services)

透過服務對企業的網路服務，減化企業與企業間商業交易的流程，連結彼此資料庫及應用系統，同步更新訂單、庫存等資料，有效整合上下游供應鏈。

3. 服務對企業員工(S2E Web Services)

企業內部員工可透過 PDA 或手機，使用企業內部的 Web Services，取得最新的顧客資料和相關工作知識，發展企業知識管理及顧客關係管理。成功案例爲 Hewitt Associates LLC⁵，每位員工有一個 PDA，透過企業內的 Web Services 隨時通知會議時間，自動安排開會時間表，定期回報個人的健康狀況及工作績效等資料，達到最佳的人力資源管理。

2.1.4 服務導向架構

網路服務在服務導向架構有三個重要角色，分別使用 UDDI、簡單物件存取協定(Simple Object Access Protocol, SOAP)、網路服務描述語言(Web Services Description Language, WSDL)等技術，進行服務發佈、找尋、繫結操作呼叫，取得網路服務，如圖 2-1 所示。

³ XMethods (www.xmethods.net)

⁴ Microsoft.NET Passport (www.passport.com)

⁵ Hewitt Associates LLC (www.hewitt.com)

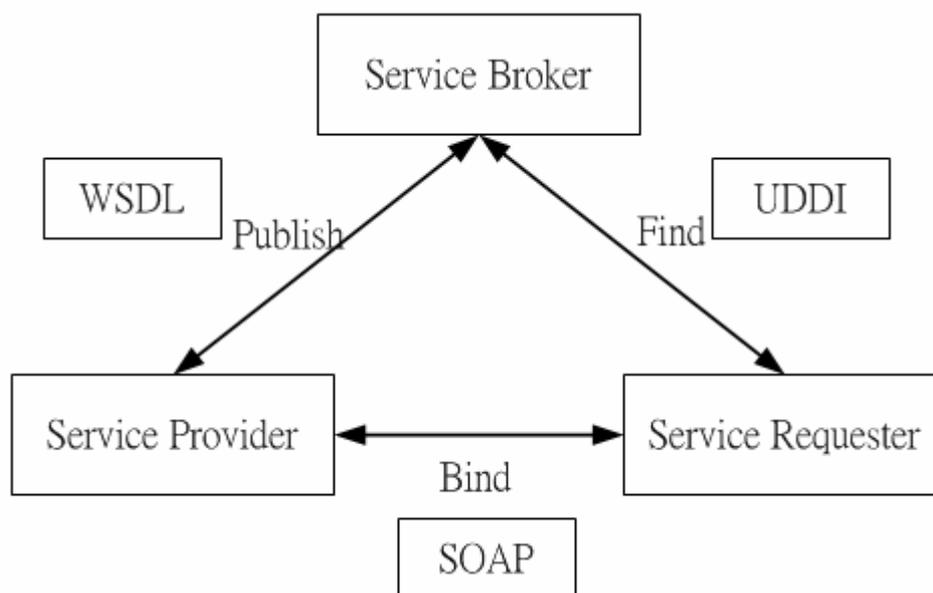


圖 2- 1 服務導向架構圖

【資料來源：張思源，民 91】

服務導向架構的主要角色可分成下列三類：

1. 服務提供者(Service Provider)

提供服務及服務本身的執行環境，包含服務的介面建置與發佈及服務的實作與服務需求回應。服務提供者提供完整網路服務，包含建立、發佈、推廣及銷售四項工作活動。

- 建立(Creation)：設計、發展、測試、建立文件等工作。
- 發佈(Publication)：將網路服務外包給應用服務提供者(ASPs)，委託 ASP 業者開發維護或是公司自行發展。
- 推廣(Promotion)：將服務註冊到服務仲介者。
- 銷售(Sale)：服務如何計價收費與使用稽核等。

2. 服務要求者(Service Requestor)

一種客戶端應用程式或相關應用，透過服務仲介者搜尋適合的服務，與實際服務提供者互動溝通、以存取或使用網路服務。

3. 服務仲介者(Service Broker)

以 XML 為基礎的規範，提供服務註冊、分類與搜尋機制，服務提供者可對其服務做適當的描述與分類並註冊公佈相關資訊。而服務要求者經由搜尋機制找出所需的服務，並取得和網路服務溝通的相關資訊。一般說來，服

務仲介者可分為公開註冊中心與私有註冊中心兩類。

— 公開註冊中心(Public Registries)：

類似於一本公開電話簿，記載所有服務資訊的環境，稱為 UDDI 商業儲存庫(UDDI Business Registry, UBR)。UBR 規範出三種頁面，分別為白頁、黃頁、綠頁。白頁記錄服務提供的基本資料，例如：公司名稱、位址及聯絡方式等。黃頁記錄公司分類資訊及服務詳細資料。綠頁記錄服務的技術資料以及商業流程等。目前已有數家資訊大廠提供公開註冊，例如：IBM、Microsoft、HP、SAP、NTT-com 等。

— 私有註冊中心(Private Registries)

私有註冊中心為私人建立的，能夠自行管理的註冊中心，可用於建立企業或組織內部的 UDDI。也可能是其他網站所發展的註冊中心，提供增值服務、進階搜尋、服務監控等附加價值的服務支援。成功案例的公司有 Salcentral、Allesta 等公司提供網路服務的品質測試及服務監控等功能。

2.1.5 網路服務在企業系統整合的優勢

網路服務的出現對現存資訊系統帶來眾多衝擊。就企業系統整合而言，網路服務的採用具有以下幾項優勢，如表 2-1 所示：

表 2-1 網路服務的優勢

簡單化	典型系統整合大多採用分散式元件物件模式 (Distributed Component Object Model, DCOM)、簡單物件呼用仲介架構 (Common Object Request Broker Architecture, CORBA)或是透過電子資料交換(Electronic Data Interchange, EDI)來達到整合。就網路服務而言，用來發展及使用網路服務的架構已建置完成，即可輕易建立一橫跨多個應用系統的新企業流程，故網路服務在設計、發展、維護及使用上皆較為簡單化。
開放標準	相較於傳統式的系統整合，網路服務所採用的標準為開放式標準，包含：UDDI、SOAP 及 HTTP。此為目前已存在的通訊協定，企業無需再投資更多的成本來支援網路協定。此優點為網路服務可被廣泛使用的原因之一。

有彈性	傳統的系統整合多採用點對點(Point-to-Point)的整合方法，系統與系統會緊密結合在一起，一但一端有所異動則會影響另一端，造成維護上的不便性；相較於此，網路服務所採用的整合方式為透過使用被公佈出來的服務來與其他應用系統做結合，充份保留系統與系統間整合的彈性。
整合範圍廣	就傳統式的系統整合而言，一個應用系統僅提供一個整合入口；然就網路服務而言，其可將應用系統區分成多個獨立的邏輯單位，以提供多個整合的連結點。以訂單管理系統為例，網路服務的整合方式可提供接單、訂單狀態查詢、訂單確認、收款及付款等服務做為系統整合的入口。
有效性高	鑑於網路服務的整合方式可為單一應用系統提供多種整合入口的優勢，網路服務的整合方法較傳統方式具有效性。
具動態性	企業系統整合的過程中，若採用網路服務的方式來進行，則只需從服務提供者所公佈出來的服務中選擇適用的商業服務，並納入企業流程中，即可達到系統整合的目的。相較於傳統式的整合方法，網路服務是以動態整合介面來整合系統，較具動態性。
封裝性	網路服務既然是一種部署在網路上的物件，自然具備物件的良好封裝性，對於使用者而言，他只能看到該物件提供的功能清單。
低耦合度	這一特性也是源於物件/元件技術，當一個網路服務的實作發生變更的時候，呼叫者是不會感到這一點的，對於呼叫者來說，祇要網路服務的呼叫介面不變，網路服務的實作任何變更對他們來說都是透明的，甚至當網路服務的實作平台從 J2EE 遷移到了.NET 或者是相反的遷移流程，使用者都可以對此一無所知。對於低耦合度而言，尤其是在 Internet 環境下的網路服務而言，需要有一種適合 Internet 環境的訊息交換協定。而 XML/SOAP 正是目前最為適合的訊息交換協定。

<p>使用協約的規範性</p>	<p>這一特性從物件而來，但相比一般物件其介面規範更加規範化和易於機器理解。首先，作為網路服務，物件介面所提供的功能應當使用標準的 WSDL 語言來描述；其次，由標準描述語言描述的服務介面應當是能夠被發現的，因此 這一描述檔需要被儲存在私有的或公共的註冊庫裡面。同時，使用標準描述語言描述的使用協約將不只有是服務介面，它將被延伸到 Web 服務的聚合、跨 網路服務的交易、工作流程等，而這些又都需要服務品質保障。其次，我們知道安全機制對於低耦合度的物件環境的重要性，因此我們需要對諸如授權認證、資料完整性(比如簽名機制)、訊息源認證以及交易的不可否認性等運用規範的方法來描述、傳輸和交換。最後，在所有層次的處理都應當是可管理的，因此需要對管理協約運用同樣的機制。</p>
<p>使用標準協定</p>	<p>作為網路服務，其所有公共的協約完全需要使用開放的標準協定進行描述、傳輸和交換。這些標準協定具有完全免費的規範，以便由任意方進行實作。一般而言，絕大多數規範將最終有 W3C 或 OASIS 作為最終版本的發佈方和維護方。</p>
<p>高度整合</p>	<p>由於網路服務採取簡單的、易理解的標準網路協定作為元件介面描述和協同描述規範，完全遮罩了不同軟體平台的差異，無論是 CORBA、DCOM 或 EJB(Enterprise JavaBean)都可以透過這一種標準的協定進行相互作業，實作在目前環境下最高的可整合性。</p>

資料來源：【劉遠威，黃雯汝，民 91】、【柴曉路，民 91】、【張思源，民 91】

第2節 網路服務核心技術探討

目前網路服務技術標準的發展依功能分為下列幾類，如表 2-2 所示：

表 2- 2 網路服務技術標準

功能	標準	說明
傳輸 (Transport)	HTTP,HTTPS	低層傳輸
服務呼叫(Invocation) 服務結構化(Marshalling)	SOAP	程式訊息與程式類別間格式轉換
描述 (Description)	WSDL	服務描述之標準
繫結 (Binding)	UDDI	查詢服務功能，類似 DNS
安全 (Security)	WS-Security,SAML ,XKMS	認證與授權
舞步/路由 (Orchestration/Routing)	WS-Routing,WSCL	長時間處理程式與多角色互動
商業流程 (Business Process)	WSFL, XLANG, BPML, BPSS	描述商業流程
計價收費 (Metering/Billing)	未定義	服務使用收費

資料來源：【張思源，民 91】

而簡單物件存取協定、網路服務描述語言及統一描述搜尋及整合三者為網路服務最主要的技術，下節將詳細介紹。

2.2.1 簡單物件存取協定(SOAP)

SOAP 在網路服務架構中所扮演的底層角色。SOAP 是一種架構簡單的輕量級(Light Weight)資料傳輸協定，用於分散式網路環境下做資料訊息交換，只要訊息收送雙方支援 SOAP，彼此就能夠交談，這也正是網路服務想要突破平台、語言疆界的最佳利器。

SOAP 是以 XML 作為資料傳輸格式，搭配網路標準的傳輸協定 HTTP、SMTP、TCP 等來傳送訊息。其中 HTTP Message 是網路伺服器所接聽的，所以一般企業的防火牆都會開放 HTTP 埠口，若網路服務的訊息透過 HTTP 來傳送，網路服務提供者就不必再多開一條對外的埠口通路，還可以直接繼承 HTTP 支援 SSL 的優點，因此大多數 SOAP 相關應用工具都會先支援 HTTP 格式。SOAP 包括四個部分：

1. SOAP 封裝(Envelop)：定義了一個描述訊息中的內容是什麼，是誰傳送的，誰應當接受並處理它以及如何處理它們的框架。
2. SOAP 編碼規則(Encoding rules)：定義序列化的機制可以用來交換實體應用程式之間的資料型態。
3. SOAP RPC 表示(RPC representation)：表示遠端過程呼叫和回應的協定。
4. SOAP 繫結 (Binding)：使用底層協定交換訊息。

雖然這四個部分都作為 SOAP 的一部分，作為一個整體定義的，但他們在功能上是相交的、彼此獨立的。信封和編碼規則是被定義在不同的 XML 命名空間(namespace)中，這樣使得定義更加簡單。

SOAP 主要特性是簡單性和可擴充性的設計，透過 SOAP 訊息由四個元素所組成，達到簡單和可擴充的目的，其設計說明如下：

1. Protocol Header 元素：以 HTTP 為例，在 HTTP Header 上指名使用那種方法進行要求、訊息會傳送到那個服務上以及採用通訊協定的版本。
2. Envelope 元素：SOAP 訊息的根元素是 Envelope 元素。由於 Envelope 元素的命名空間(Namespace)是唯一識別，他允許處理工具馬上決定是否該 XML 檔是 SOAP 訊息。SOAP 封包可以包含一個選擇性的(可有可無)Header 元素

與一個必要的 **Body** 元素。在 **Body** 元素後可接任意的 **XML** 元素。此項的擴充性對於 **SOAP** 內訊息的編碼很有幫助。

3. **Header** 元素：一般來說定義了 **SOAP** 資料型態、**SOAP** 編碼及以” **SOAP Action**”來指出訊息的處理者及呼叫方法。
4. **Body** 元素：為 **SOAP** 訊息的主要元素，一般稱為主體。可以包含任意的 **XML** 檔。主要傳送的是客戶端的要求訊息以及伺服器端回應的訊息。

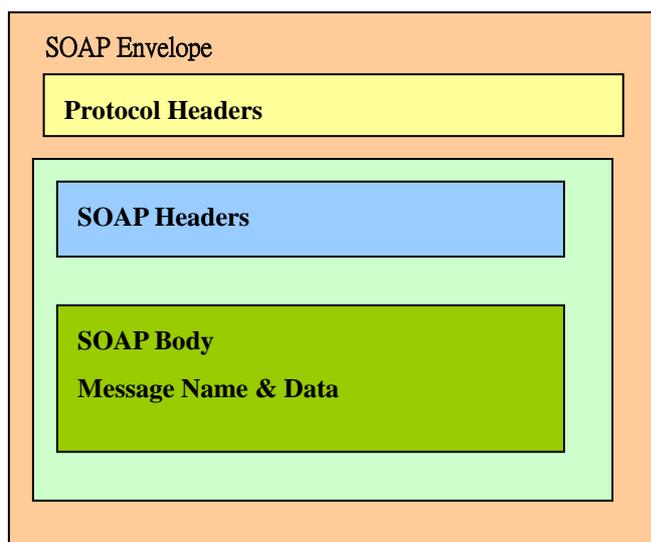


圖 2- 2 SOAP 訊息格式

【資料來源：Gudgin，2003】

在買賣雙方之間傳遞 **SOAP** 訊息，訊息可分為 **SOAP** 要求和 **SOAP** 回應兩種，使用者送出 **SOAP** 要求查詢股價，如圖 2-3 所示。**Header** 元素包括兩個參數：**priority** 和 **expires**，說明 **SOAP** 訊息的優先權及有效使用時間。**Body** 元素是實際傳送的訊息，查詢股票 2303 的股價，其程式碼見圖 2-3。

```

<SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <SOAP-ENV:Header>
    <n:altercontrol xmlns:n="http://example.org/altercontrol">
      <n:priority>1</n:priority>
      <n:expires>2003-11-20T11:50:00</n:expires>
    </n:altercontrol>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <BankQuery xmlns="http://www.openuri.org">
      <stock>2303</stock>
    </BankQuery>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

圖 2- 3 SOAP 要求訊息範例

銀行端接收到使用者發出的 SOAP 要求，透過內部股票交易系統，查詢報價資訊，產生 SOAP 回應訊息，傳送回使用者。SOAP 回應的 Body 元素表示公司 2303 的股價為 29.8，其程式碼見圖 2-4。

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <ns:BankResponse xmlns:ns="http://www.openuri.org">
      <BankResult>
        <dealPrice>29.8</dealPrice>
      </BankResult>
    </ns:BankResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

圖 2- 4 SOAP 回應訊息範例

2.2.2 網路服務描述語言(WSDL)

WSDL 是符合 XML 格式的檔。由 Microsoft、IBM、Ariba 等軟體廠商共同開發，WSDL1.1 規範也在 2001 年三月正式列為 W3C 標準。以 XML 格式來描述網路服務的語言，也就是網路服務的描述語言。當服務提供者欲對外公佈其提

供之網路服務，就必須以 WSDL 來建置描述檔案，主要分成服務實作定義及服務介面定義兩個部分，其格式如圖 2-5 所示。

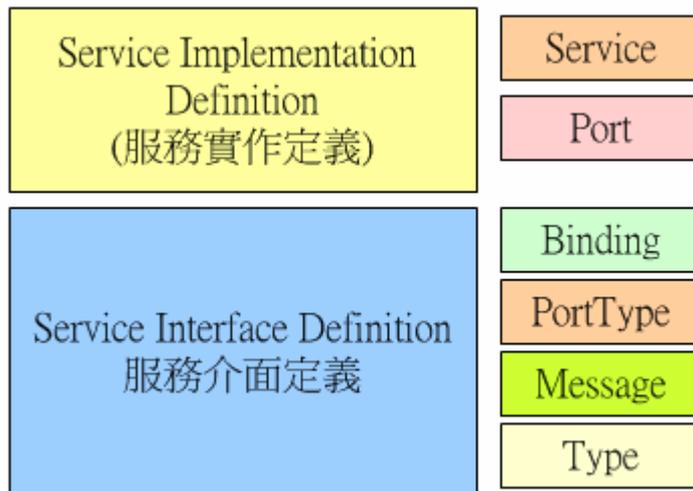


圖 2- 5 WSDL 文件

整個 WSDL 檔會從定義(definitions)開始，其包含了型別(Type)、訊息(Message)、埠口類別(PortType)、繫結(Binding)、服務(Service)、操作(Operations)以及埠口(Port)。圖 2-6 為完整 WSDL 檔的範例。

1. 定義元素(definitions)

定義元素(definitions)包括一個或多個服務的定義。大多數情況下，一個 WSDL 文件定義一個獨立服務。definitions 通常緊跟著以下屬性的聲明：

- ◆ 名稱(name)：這個屬性是選擇性的，用來說明服務的主要目的。
- ◆ 目標空間名稱(targetNamespace)：這個屬性定義了關於服務訊息的邏輯命名空間，並且各服務的屬性值通常是不同的。這個屬性在稍後會作更進一步的討論。
- ◆ xmlns:tns：在許多的 WSDL 檔中，這個命名空間並不出現（包括我們的範例），但是很快就會流行起來的。如果出現，則被設定成 argetNamespace 的值。這個屬性在稍後會作更進一步的討論。
- ◆ xmlns:soap 和 xmlns:xsd：它們是標準命名空間的定義，在以後的 WSDL 檔中被用作指定特定的 SOAP 的訊息和資料類型。
- ◆ xmlns：預設的 WSDL 檔的命名空間，被設定到 <http://schemas.xmlsoap.org/wsdl/>。所有的 WSDL 標示，就像 definitions、

message 和 service 都記錄在這個命名空間之內。

2. 訊息元素(Message)

訊息元素(message)對應在呼叫者和服務之間傳遞的一條訊息。Web Service 往返的遠端方法呼叫兩條訊息，一條負責請求，一條負責回應。回應訊息可以是空訊息或特定格式的 XML 檔。

3. 埠口類別(portType)

埠口類別元素(portType)對應一組作業或多組操作元素(operation)，而操作元素定義一組特定的輸入/輸出(input/output)訊息序列。每組輸入/輸出的訊息屬性必須對應前面定義過的訊息元素的名稱。如果一個操作只指定了輸入，則祇是單向作業。當 WSDL 描述一個物件時，每個操作元素對應一個方法，並且每個埠口類別元素對應一個 Java 介面或類別。

在這個範例中，s0:ConvertTemperatureSoapIn 操作接受一個訊息元素作為它的輸入，並回應一個 s0:ConvertTemperatureSoapOut 訊息作為它的輸出。

4. 繫結(Binding)

繫結元素(binding)對應於用特定的協定 -- 如 SOAP 或者 CORBA -- 來實作的埠口類別元素。繫結的類型屬性必須對應定義過的埠口類別元素的名稱。因為 WSDL 是中性的協定，所以您可以指定 SOAP、CORBA、DCOM 和其他的標準協定的繫結。如果服務支援不止一個協定，WSDL 應該對每個它支援的協定都包括一個繫結元素。

5. 服務(Service)、操作(Operations)與埠口(Port)

服務元素(service)是一個連接埠口，埠口元素(port)代表了在特定端點進行特定繫結的可用性。埠口的繫結屬性必須對應於前面定義過的繫結元素的名稱。例如，透過 <http://localhost/myEfpc/Service1.asmx> 網站的 s0:Service1Soap 繫結可存取服務元素，並呼叫 ConvertTemperature，執行操作。

```

<definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:= "http://www.w3.org/2001/XMLSchema" xmlns:s0="http://redice.mis.nccu.edu.tw"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" targetNamespace="http://redice.mis.nccu.edu.tw"
xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://redice.mis.nccu.edu.tw">
      </s:schema>
    </types>
    <message name="ConvertTemperatureSoapIn"><part name="parameters" element="s0:ConvertTemperature"/>
  </message>
    <message name="ConvertTemperatureSoapOut"><part name="parameters"
element="s0:ConvertTemperatureResponse"/></message>
    <portType name="Service1Soap">
      <operation name="ConvertTemperature">
        <documentation>這個method目的是將華氏測量的溫度轉換為攝氏溫度公式為 ((dFahrenheit - 32) *
5) / 9</documentation>
        <input message="s0:ConvertTemperatureSoapIn"/>
        <output message="s0:ConvertTemperatureSoapOut"/>
      </operation>
    </portType>
    <binding name="Service1Soap" type="s0:Service1Soap">
      <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
      <operation name="ConvertTemperature">
        <soap:operation soapAction="http://redice.mis.nccu.edu.tw/ConvertTemperature" style="document"/>
        <input><soap:body use="literal"/></input>
        <output><soap:body use="literal"/></output>
      </operation>
    </binding>
    <service name="Service1">
      <port name="Service1Soap" binding="s0:Service1Soap"><soap:address
location="http://localhost/myEfpc/Service1.asmx"/></port>
    </service>
  </definitions>

```

圖 2- 6 WSDL 文件範例

2.2.3 統一描述搜尋及整合(UDDI)

UDDI 定義發佈與發現 Web Service 的方法。UDDI 中心最主要功能是 UDDI 商業註冊，當中使用 XML 檔案來描述商業實體與其所提供的網路服務，UDDI 為 B2B 電子交易提供服務描述的功能，發佈描述服務的檔案，並將同產業的不同供應商的供應整合在一起的註冊機制。透過 UDDI 的描述與發現機制，將自己所從事的商業行為發佈出去並且可以讓別人搜尋與使用。也可以將從事同產業商業行為的廠商整合在一起，可以取得同產業相關供應者的目錄【王之怡，民 91】。

UDDI 中的每一個資料結構，都用來表示一種特殊資料型態的資料。例如，商業實體想要發佈其提供的網路服務程式，不論這些程式是用來與會計程式互動的進入點，或是允許顧客查詢訂單處理的狀況，每個商業實體，或是相關的服務描述，皆以結構化的資料型態存在於 UDDI Registry 中。

UDDI 的註冊資訊由 XML 來表示，其資料型態由五個元件組成，分別為：businessEntity、businessService、bindingTemplate、tModel 與 publicAssertion。businessEntity 為有關發佈服務群的公司團體資訊，每一個 businessEntity 當中可以包含一個或多個 businessService。

businessService 為描述特定服務的資訊，每一個 businessService 當中可以有一個或多個 bindingTemplate。bindingTemplate 為對服務的連結點與規格技術上的資訊。對於每一個 bindingTemplate 資料參考對應到一個 tModel，而 tModel 派了每一個服務的介面規格，tModel 中還包含了基於技術上的特點，對服務與專有名詞的描述。有些商業實體是由多家公司共同公佈，在 publicAssertion 中表示有關兩個公司團體間的資訊，由其中一方或兩者共同公佈。圖 2-7 表示 UDDI 的資料結構與關係。

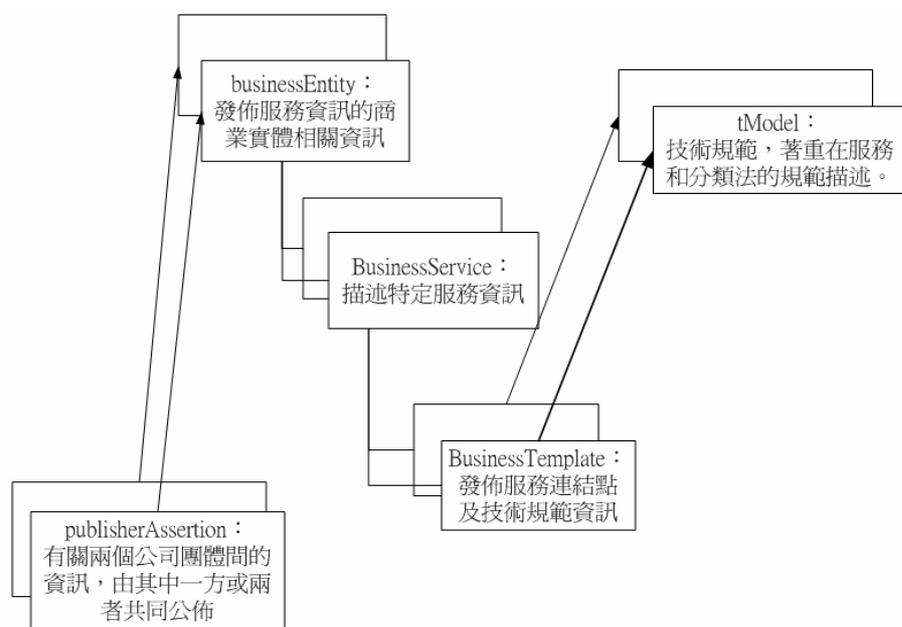


圖 2- 7 UDDI 資料結構

2.2.4 網路服務運作流程

網路服務是一個以標準化 XML 訊息描述網路上操作集合的介面，與應用程式的互通性有關。網路服務描述一個架構，使在分散式計算的環境中，可以描述、發佈、發現，動態呼叫的商業服務。網路服務流程如圖 2-8 所示，其說明如下：

1. 服務提供者向 UDDI 發佈網路服務，發佈 WSDL 描述檔的網路位址，並建立 SOAP 伺服器端程式。WSDL 描述檔與伺服器端的程式所在的位址可由服務提供者決定。
2. 服務使用者至 UDDI 找尋所需求的網路服務，從 UDDI 中所指向的網路位址中，下載描述網路服務的 WSDL 檔。
3. 服務使用者根據 WSDL 檔中描述的介面規格，實作出 SOAP 客戶端程式，藉由使用自己實作的客戶端程式，呼叫遠端的網路服務。

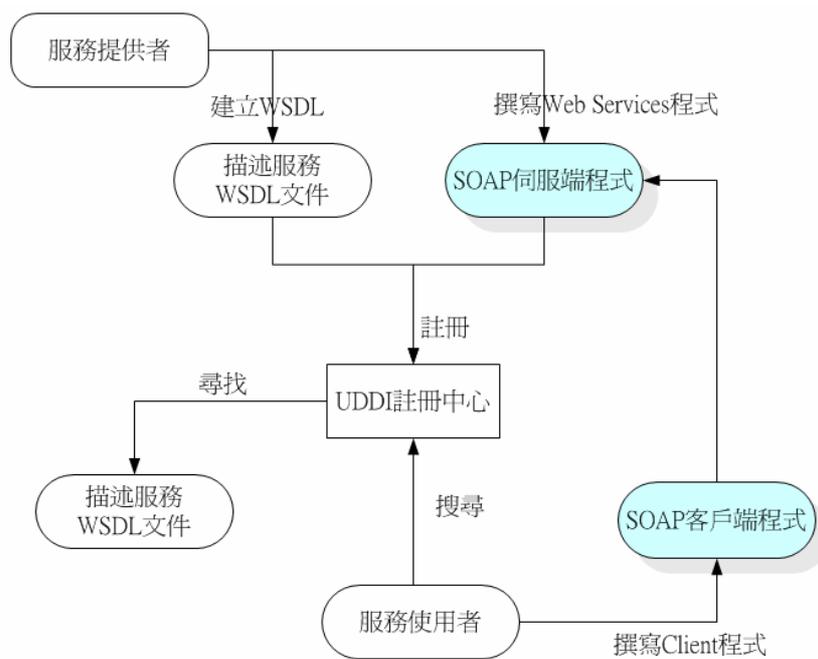


圖 2- 8 網路服務運作流程圖

第3節 XML 與關連資料表格間的轉換

Florescu(1999)所提出 XML 文件儲存至關連資料庫，為針對一簡單 XML 訊息範例，如圖 2-9 所示，並建構出 XML 訊息的有向圖，如圖 2-10 所示，其中節點代表物件(object)，用以編號(oid)區別，連結線(edges)代表屬性(attribute)。每個物件可能有一連串的屬性，皆以連接線表示。葉子(leaves)代表資料值(value)，可能為整數、字串或日期等型態。若連結的目標物件為葉子，稱為資料屬性，反之，稱為參考屬性(例如：圖 2-10 物件 2 號的 child 屬性指向物件 4 號，即為一參考屬性)。此模式的限制為無法區分子元素與屬性，皆將其視為屬性。

```
<people>
  <person id="1" age="55">
    <name>Peter</name>
    <address>4711 Fruitdale Ave.</address>
    <child>
      <person id="3" age="22">
        <name>John</name>
        <address>5361 Columbia Ave.</address>
        <hobby>swimming</hobby>
        <hobby>cycling</hobby>
      </person>
    </child>
    <child>
      <person id="4" age="7">
        <name>David</name>
        <address>4711 Fruitdale Ave.</address>
      </person>
    </child>
  </person>
  <person id="2" age="38" child="4">
    <name>Mary</name>
    <address>4711 Fruitdale Ave.</address>
    <hobby>painting</hobby>
  </person>
</people>
```

圖 2- 9 Florescu XML 文件

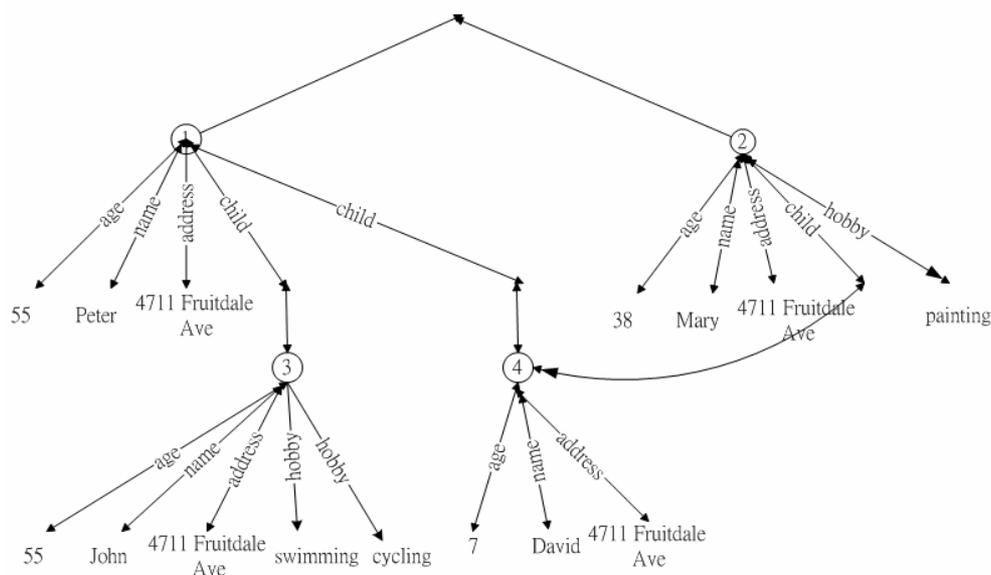


圖 2- 10 XML 資料有向圖

建立 XML 資料有向圖後，接下來我們說明將其轉換為資料表格儲存於關連資料庫中。Florescu 整理出屬性資料有四種對應方法：連結線法(Edge Approach)、屬性法(Attribute Approach)、統一表格法(Universal Table)與正規統一表格法(Normalized Universal Approach)，下述分別說明之。

2.3.1 連結線資料模式

第一種方法為連結線資料模式，將所有 XML 連結屬性依元素的來源(source)、順序(ordinal)、名稱(name)、旗標(flag)及目的(target)屬性、儲存於同一連結(Edge)表格中。其結構表示式為 Edge(source, ordinal, name, flag, target)，旗標屬性為元素型態，如：int、string、date 等基本型態。目的屬性為元素值(value)。其範例依 Edge Approach 表示，如表 2-3 所示。

表 2- 3 Edge Approach 資料模式

Source	ordinal	name	target
1	1	age	55
1	2	name	Peter
1	3	address	4711 Fruitdale
1	4	child	3
1	5	child	4
2	1	age	38
...

2.3.2 屬性資料模式

第二種方法為屬性資料模式，將相同的 XML 屬性儲存在同一表格。一種屬性就有一個表格，故使用此法會產生多個表格，其結構表示式為 $A_{name}(source, ordinal, flag, target)$ 。其範例依 Attribute Approach 表示，如表 2-4 所示。

表 2- 4 Attribute Approach 資料模式

A_{hobby}		
source	ordinal	target
2	5	painting
3	4	swimming
4	5	cycling

A_{child}		
source	ordinal	target
1	4	3
1	5	4
2	4	4

2.3.3 統一表格資料模式

第三種方法為統一表格資料模式，將所有 XML 屬性儲存在單一表格。概念上，Universal Table 類似使用 Outer Join 所有表格欄位元的結果，其結構表示式為 $Universal(source, ordinal_{n1}, flag_{n1}, target_{n1}, ordinal_{n2}, flag_{n2}, target_{n2}, \dots, ordinal_{nk}, flag_{nk}, target_{nk})$ 。其範例依 Universal Table 表示，如表 2-5 所示。

表 2- 5 Universal Table 資料模式

source	ord _{name}	targ _{name}	ord _{child}	targ _{child}	Ord _{hobby}	Targ _{hobby}	...
1	2	Peter	4	3	Null	Null	...
1	2	Peter	5	4	Null	Null	...
2	2	Mary	4	4	5	painting	
3	2	John	Null	Null	4	swimming	
3	2	John	Null	Null	5	cycling	
4	2	David	Null	Null	Null	Null	

Universal Table 處理多重屬性值時，容易產生空值(Null)和資料重複(redundancy)問題，在表 2-5 中，為了表示 Peter 的 child=4 和 child=5，則需建立兩筆紀錄，同時 Ord_{hobby} 和 Targ_{hobby} 為空值，造成資料庫空間的浪費。

2.3.4 正規統一表格資料模式

第四種方法為正規統一表格資料模式，類似於第三種方法，不同的是正規統一表格處理多重屬性值，會儲存在另一額外(Overflow)表格中，其結構表示式為 Universal(source, ordinal_{n1}, flag_{n1}, target_{n1}, ordinal_{n2}, flag_{n2}, target_{n2}, ..., ordinal_{nk}, flag_{nk}, target_{nk})、Overflow_{n1}(source, ordinal, flag, target), ..., Overflow_{nk}(source, ordinal, flag, target)，旗標值等於”m”表示屬性為多重屬性，須對應到 Overflow 表格，如此一來，就能解決資料重複問題。其範例依 Normalized Universal Approach 表示，如表 2-6 所示。

表 2- 6 Normalized Universal 和 Overflow 資料模式範例

source	ord _{name}	targ _{name}	...	Ord _{hobby}	flag _{hobby}	Targ _{hobby}
1	2	Peter	...	Null	Null	Null
2	2	Mary	...	5	-	painting
3	2	John	...	4	m	Null
4	2	David	...	Null	Null	Null

Overflowhobby		
source	ord	target
3	4	swimming
3	5	cycling

2.3.5 DTD 檔關聯資料模式

J. Shanmugasundaram(2001)和 Feng Tian(2002)兩位學者分別提出以 XML DTD 文件為基礎，將 XML 檔轉換到關聯式資料表格。由於 DTD 檔清楚定義了節點之間的關聯規則，藉由 DTD 文件將 XML 實際資料儲存於關聯式資料表格。

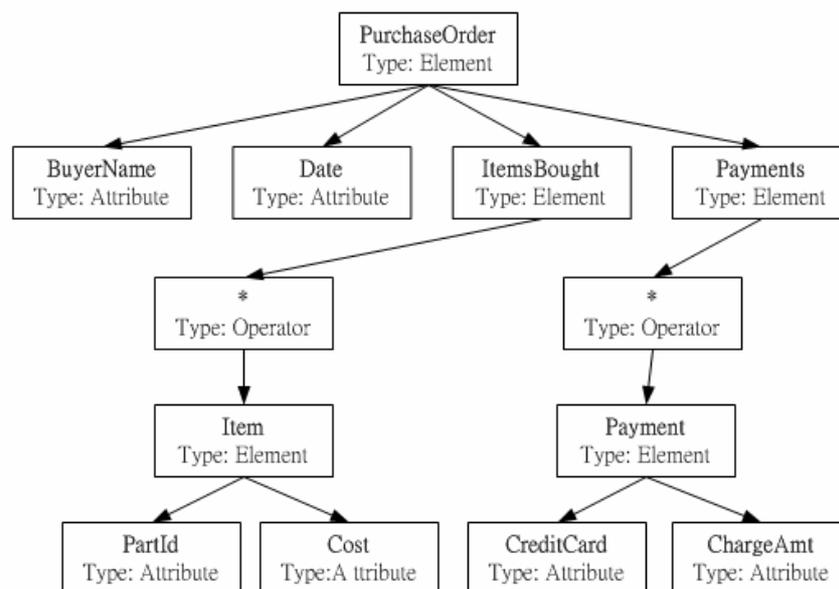
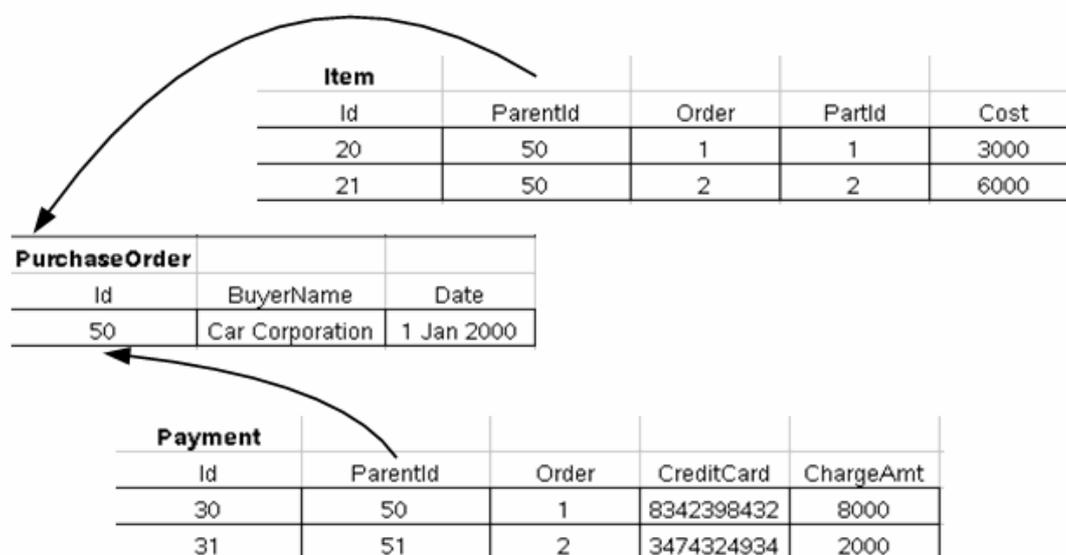


圖 2- 11 XML DTD 有向圖

XML DTD Approach 的轉換結果，如圖 2-12 所示。PurchaseOrder、Payment 和 Item 為元素(Element)節點，轉換為單獨的資料表格。BuyerName、Date、PartId、Cost、CreditCard 和 ChargeAmt 為屬性(Attribute)節點，則轉換為資料表格中的屬性。而表格與表格之間的外國鍵關係(Foreign Key)，則是依靠 ParentId 屬性與主表格作關聯外國鍵。Order 表示原始資料的順序。

表 2- 7 Relational DTD Approach 資料模式



第4節 銀行公會訊息標準

近年來我國銀行在業務自動化已有相當的成效，但目前金融電子資料交換(FEDI)範圍侷限在專屬封閉性網路，為考慮企業對網際網路運用已日趨成熟，且基於網際網路開放性、全球性之特性，網際網路上電子資料交換作業必將成為未來電子交易趨勢，因而財政部在 2000 年即督促銀行公會成立 XML(eXtensible Markup Language)訊息訂定小組，負責金融 XML 訊息標準訂定，開發適用於網際網路之金融訊息。同時為配合行政院推動全國知識經濟發展方案，以完成網際網路跨行作業平台為推動目標。XML 訊息訂定小組成立時，國際間尚未有明確統一之金融 XML 訊息標準，小組面臨第一個議題為--選擇標準訂定參考依據及第一階段訊息訂定範圍。經小組討論，並進行企業及資訊服務業者之間卷調查後，決定第一階段訊息設計範圍為企業間(B2B)的台幣付款機制，在訊息訂定範圍討論的同時也開始蒐集研讀國際相關 XML 標準資訊，經比較分析後決定採用金融產業相關訊息最完整、且為最多國際性銀行參與標準訂定之 IFX(Interactive Financial Exchange)為 XML 訊息標準訂定參考依據。

金融訊息訂定從 2000 年 7 月開始，歷經 XML 國際標準組織資訊蒐集、訊息訂定範圍及方向評估、問卷調查表設計、問卷調查、問卷結果統計分析、訊息內容及流程討論，在 2001 年 11 月將跨行、整批付款及通知訊息納入，完成系統建置指引 1.1 版、訊息建置指引 1.1 版增修及跨行訊息建置指引 1.0 版撰寫，提供企業或個人藉由網際網路進行自行、跨行之付款、轉帳及帳戶查詢功能。

2002 年 8 月除依 IFXv1.3 改版配合並將總數檢核訊息納入，完成系統建置指引 v1.2 版、系統建置指引附冊一文件資料項目說明表、訊息建置指引 v1.2 版增修及跨行訊息建置指引 v1.1 版增修，提供參加單位、企業或個人藉由網際網路進行收付款業務系統建置之重要參考依據。2002 年 12 月除依一般業務委員會電子商務組之"多行歸戶餘額查詢"之需求新增相關訊息外，並參酌會員銀行陸續提供之 DMR 審慎評估後，完成跨行訊息與系統建置。

銀行公會定義金融 XML 訊息、金融 XML 跨行訊息、金融 XML 融資訊息、金融 XML 外幣訊息等標準規範，其中以金融 XML 訊息標準最為完整。目前已有的銀行服務訊息，包含餘額查詢訊息、多行歸戶餘額查詢訊息、交易明細查詢回應訊息、付款處理狀況查詢訊息、預約付款處理訊息等標準。

圖 2-12 和表 2-8 分別表示銀行服務類的交易明細訊息結構以及各項資料元所代表的意義。

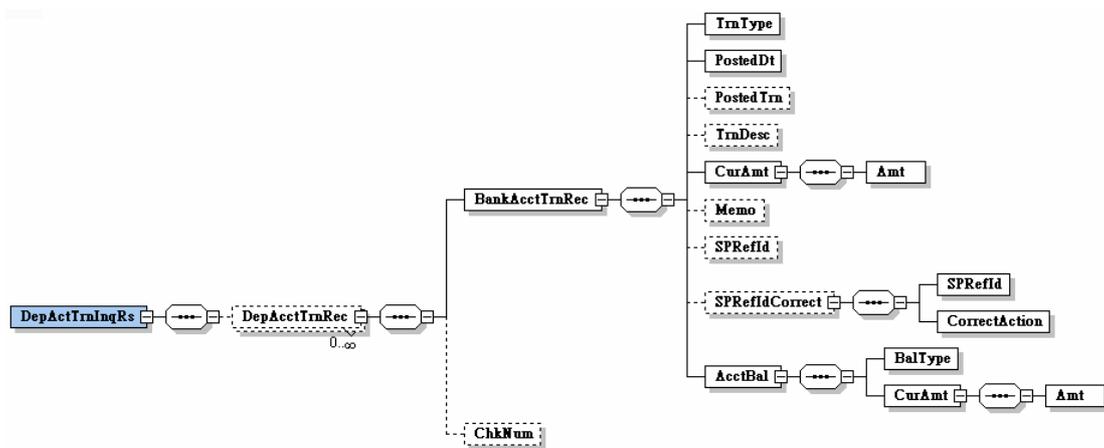


圖 2-12 銀行公會交易明細訊息結構圖

表 2- 8 各項資料元所代表的意義

資料元名稱	內容說明
DepAcctTrnRec	存款帳戶交易明細，本組合資料元可重複使用，係用以列出所有符合查詢條件之交易明細。
BankAcctTrnRec	銀行帳戶交易明細，包含交易之借貸別、交易日期、交易時間、交易金額、備註、更正資料等。
TrnType	借貸別，表示交易之借貸別，「Debit」為提出。「Credit」為存入。
PostedDt	交易日期。
PostedTm	交易時間。
TrnDesc	交易摘要，說明交易的類別或方式，由各銀行依業務需求自行訂定。
CurAmt	交易金額，若交易為更正交易時，金額仍為正值，且借貸方不變。
Memo	備註，由各銀行依業務需求自行訂定。
SPRefId	交易參考序號。
SPRefIdCorrect	更正資料，描述交易之相關更正資料。
SPRefId	原交易參考序號，本交易所更正的原交易參考序號。此資料元若於逐筆付款類交易已提供<SPRefId>(交易參考序號)者，得由銀行端斟酌提供，如無法提供則預設為「0」。
CorrectAction	更正記號，表示本交易為更正交易，固定值為「deleted」。
AcctBal	帳戶餘額，用以表示交易後的帳戶餘額。
BalType	餘額類別，係用以說明客戶之帳戶餘額類別，在「交易明細查詢回應」訊息中必須回應下列帳戶相關餘額，「Current」為帳面餘額。
CurAmt	金額，係用以說明客戶之帳戶餘額。
ChkNum	支票號碼