

第3章 研究設計

本研究在於建立一套服務導向架構的系統平台，整合金融資產帳戶服務，將使用者來自各個不同金融機構的帳戶資料整合到客戶端，使用者能夠輕易取得整合性的資產帳戶。此平台架構涉及 UDDI 建立、服務註冊與服務搜尋以及建立仲介服務提供者的共通平台，將不同金融機構的 XML 欄位，匯整成標準欄位格式，客戶端根據標準欄位格式，將各個帳戶彙總成帳戶報表。

本章探討開發整合資產帳戶網路服務平台所採用的發展方法與架構設計。第一節首先說明系統設計的概念。第二節將從需求與技術層面說明詳細的系統架構。第三節為系統流程，陳述整合金融資產帳戶服務的形成流程。第四節將說明帳戶欄位名稱對應與實際值資料呈現。

第1節 系統概念

3.1.1 系統概念說明

以往資料整合動作大多是由單一入口網站向外整合其他的網站資料，使用者將個人資料、帳號及密碼記錄到該網站，當使用者要用到某一網站服務時，由該網站自動登入其他網站，如圖 3-1 所示。

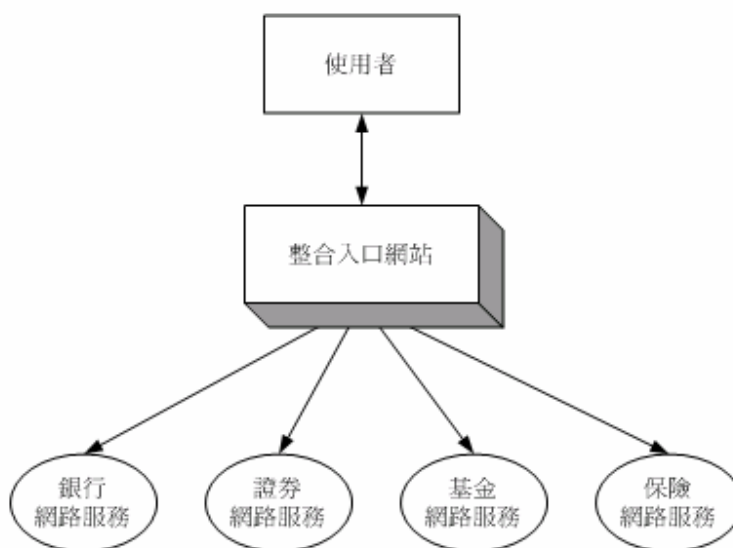


圖 3-1 單一網站整合網路服務

使用者透過整合入口網站，獲得各個網站所提供的服務資訊，然而，對於提供資產帳戶服務而言，使用者有著不同的需求，由於客戶帳戶交易資料屬於高度隱密性，若採取上述之整合入口網站模式，使用者個人帳戶資料(如：帳號、密碼、信用卡號等)要儲存一份在該網站中。本研究考量個人金融資料的安全性及私密性問題，以避免資料備份在入口網站，故提出一分散式整合系統架構，以適用於金融資產帳戶整合之用，如圖 3-2 所示。

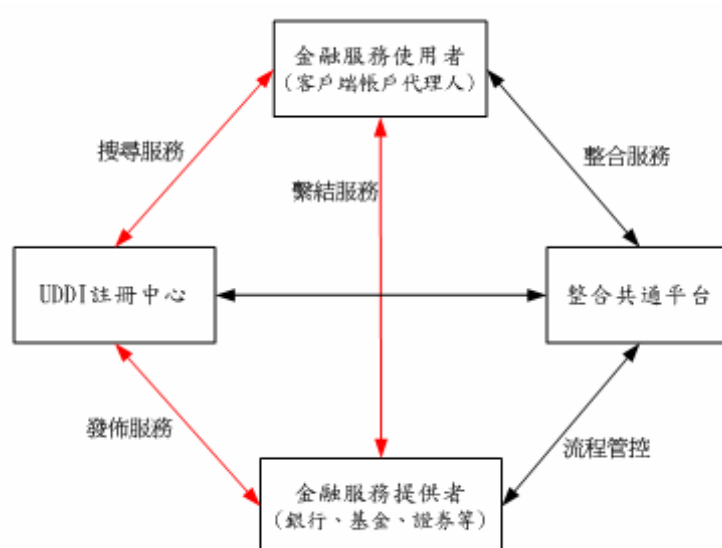


圖 3-2 系統概念圖

【資料來源：本研究】

由於服務導向架構網路服務特性，能解決資料放在外部網站整合的問題，本研究透過網路服務技術將帳戶資料直接從服務提供者傳至服務使用者，傳遞間不經過任何仲介者的介入，能確保資料保密。

整合共通平台可視為第三加值服務提供者(Third Party Service Providers)，提供附加整合性的諮詢服務，處理金融機構資料格式不一致之對應轉換，以目前各金控產業下的銀行業為例，雖然銀行公會已定義好銀行間交換標準，但銀行內部即有系統仍有特殊欄位是銀行公會所沒有定義，本研究希望透過共通平台，能夠有效處理 XML 欄位對應與合併，讓使用者容易取得彙整性的帳戶欄位。同時，在客戶端方面，由於系統回傳資料是 XML 檔，使用者無法透過瀏覽器直接閱讀，

故本研究在客戶端提供一套帳戶代理人，能夠取得共通平台的對應合併資料及各金融帳戶明細，將帳戶明細依合併規則一一對應，呈現在客戶端畫面上。

3.1.2 系統環境需求

表 3-1 各模組發展工具

項目 角色	應用實體角色	發展環境及工具	儲存設備
服務提供者	金融機構 (銀行、基金等)	Visual Studio.NET 和 C#	MS SQL Server
UDDI 註冊中心	公開第三者 (財團法人等)	AUDDI Server	LDAP Server
整合共通平台	加值服務公司	BEA WebLogic Application Server	MS SQL Server
服務使用者	一般使用者	J2SE、JBuilder	MS Access

1. Visual Studio.NET 和 C#

Visual Studio.NET 是一整合開發環境(Integrated Development Environment)的發展工具，而 C#是.NET 中具備物件導向的程式語言，兩者皆為 Microsoft 所發展，本系統服務提供者將採用兩項技術實作相關金融之網路服務。

2. AUDDI Server

AUDDI Server 是由 Acumen Advanced Technologies 公司所發展出來的，符合 GASIS 中 UDDI 2.0 規範標準之系統，故服務使用者在向 UDDI Registry 查詢網路服務時，使用 UDDI API(Application Program Interface)與 AUDDI Server 溝通，取得適合的網路服務。這類 UDDI Server 大多可建立在大型的應用程式伺服器(Application Server)軟體上，提供完整的網路服務註冊與搜尋，這類開發工具有 BEA WebLogic 的 AUDDI Server 和 SUN ONE 的 JAXR Registry

等。

3. BEA WebLogic Application Server

WebLogic Application Server 是一套符合 J2EE 設計架構的應用程式伺服器 (Application Server) 軟體，同時，支援網路服務各項技術標準的開發，其中 WebLogic Workshop 更是撰寫網路服務的開發工具，友善的使用者介面，輕易建置複雜的網路服務，縮短系統發展時間。

4. J2SE 與 Borland JBuilder

由於 JAVA 具有平台獨立特性、可攜性及可於任何設備及平台上執行，適合行動代理人程式開發之要求，而 Borland JBuilder 是一套撰寫 JAVA 的開發環境，能夠輔助程式開發，減少執行程式的錯誤，提昇程式人員的競爭力。

第2節 系統架構

系統架構分成三個部分說明，UDDI 註冊與搜尋的服務仲介者、共通平台整合 XML 文件與關連資料庫的資料模式、服務使用者行動代理程式處理相關事宜，詳見圖 3-3。

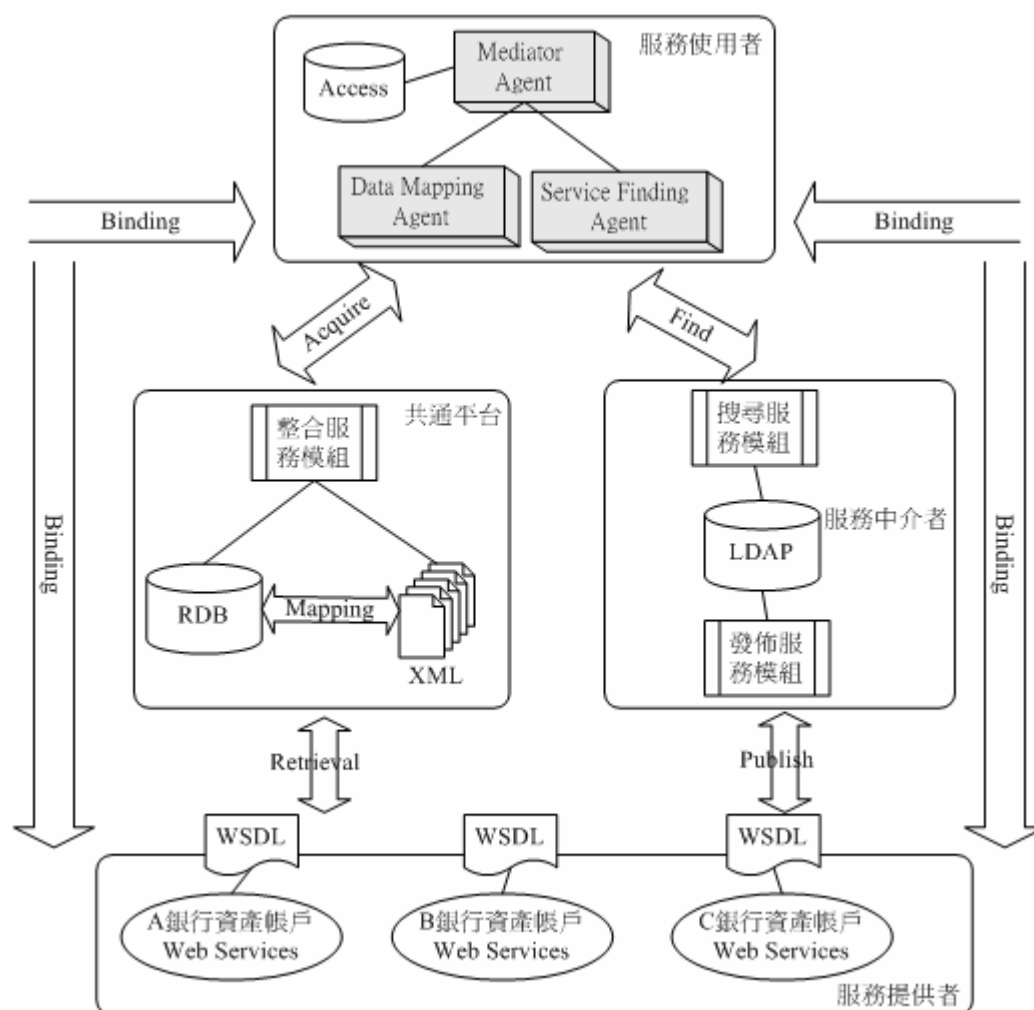


圖 3-3 系統架構圖

【資料來源：本研究】

1. 服務仲介者(UDDI)：具有搜尋服務、發佈服務模組以及 LDAP Server(Lightweight Directory Access Protocol Server)。任何服務要註冊到 UDDI，須使用到發佈服務模組，而服務使用者查詢服務則透過搜尋服務模組找到適合的服務。所有的網路服務資料皆儲存在 LDAP Server。

2. 共通平台：處理 XML 檔與關連資料轉換，將不同服務提供者所提供的 XML 檔合併成單一關聯資料表格。(第四節將詳細說明)
3. 服務使用者：擁有 Mediator Agent、Service Finding Agent 和 Data Mapping Agent 三項代理程式。Mediator Agent 處理使用者與系統之互動事項，如：畫面呈現、Agents 間訊息傳遞等。Service Finding Agent 則是實際呼叫 UDDI 的搜尋服務，所採用 Inquiry 模組為 UDDI4J⁶，符合 UDDI 查詢服務的規範。Data Mapping Agent 則是呼叫共通平台之整合服務模組，取得對應後的關聯資料表格，並向各個銀行服務提供者取得實際的帳戶明細資料，依關聯資料表格的規則合併出彙整資料，呈現在使用者畫面上。

第3節 系統流程

本研究之系統流程可分為六個步驟，下述說明之。

1. Publish：服務提供者(金融機構)將網路服務 WSDL 註冊到 UDDI。
2. Find：服務使用者透過 Service Finding Agent 啟動 UDDI 搜尋網路服務，並回傳網路服務的位置。
3. Integrate：Data Mapping Agent 向共通平台發出請求，回傳步驟 4、5 所產生的合併規則。
4. Retrieval：共通平台向服務提供者索取 XML 檔。
5. 轉換 XML 檔和關連資料表格：
 - 5a：將所有 XML 檔儲存至關連資料表格。
 - 5b：產生 XML 文件的合併規則。
6. Binding：實際透過網路服務，取得各銀行的帳戶資料，並依關連表格的規則，將帳戶資料儲存至行動代理程式的 Access 資料庫。

⁶ UDDI4J 是由 IBM、HP 與 SAP 共同開發。UDDI4J 的內涵其實是遵循 UDDI 規範的 API Sets。UDDI4J 提供以 Java 開發程式時與 UDDI 註冊中心互動所需的 API。

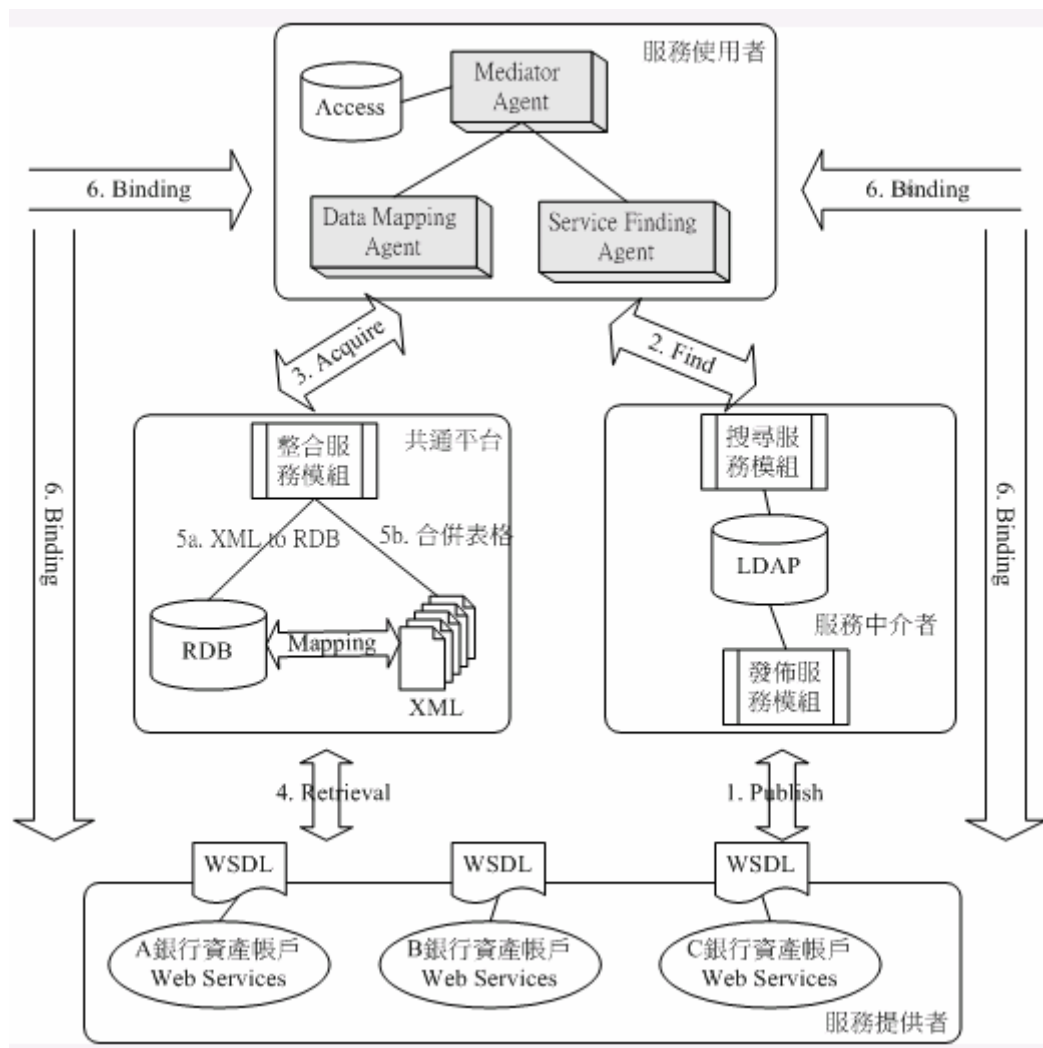


圖 3-4 系統流程圖

【資料來源：本研究】

第4節 帳戶欄位名稱對應與整合報表呈現方式

各銀行網路服務提供的資料分為帳戶欄位及帳戶欄位實際值，分別以 XML 格式表示。帳戶欄位只記錄欄位型態及欄位說明，由共通平台將帳戶欄位 XML 檔轉換成關聯式資料表格，再進行各銀行帳戶欄位對應及合併成標準欄位。帳戶代理人再向共通平台索取標準欄位後，依照標準欄位定義，一一將帳戶欄位實際值彙總，呈現整合報表，本節分成兩個部分說明銀行帳戶欄位合併成標準欄位，以及整合報表呈現方式。

3.4.1 銀行帳戶欄位合併成標準欄位

共通平台須將一家銀行帳戶欄位資料 XML 檔，轉換成關聯式資料表格，其原始銀行帳戶欄位檔如圖 3-5 所示，經過共通平台對應及合併帳戶欄位之後，將所有欄位元存入資料庫表格中，其結果如表 3-2 所示。

```
<BankAcctTmRec>
  <TmType flag="string">借貸別</TmType>
  <PostedDt flag="int">交易日期</PostedDt>
  <PostedTm flag="int">交易時間</PostedTm>
  <TmDesc flag="string">交易摘要</TmDesc>
  <AcctBal>
    <BalType flag="string">餘額類別</BalType>
    <CurAmt>
      <Amt flag="double">金額</Amt>
      <ExpireDate flag="double">逾期日期</ExpireDate>
    </CurAmt>
    <DeliveryMethod flag="string">通知方式</DeliveryMethod>
  </AcctBal>
</BankAcctTmRec>
```

圖 3-5 帳戶欄位 XML 文件

表 3-2 中 source 和 ordinal 指的是標準帳戶欄位的來源和順序，name 是帳戶欄位代號，flag 是帳戶欄位的資料型態，target 是帳戶欄位的說明定義，表 3-2 表示共通平台已整合了 A、B 銀行兩家的帳戶欄位，未來有其他的銀行帳戶欄位整合的話，則表格續加 sourceXX、ordinalXX，即可達到擴充的效果。

表 3-2 標準欄位關聯表格

來源	順序	名稱	型別	值	子節點	來源A	順序A	來源B	順序B
source	ordinal	name	flag	target	tokey	sourceA	ordinalA	sourceB	ordinalB
0	1	DepAcctTmRec	string	x	1	0	1
1	1	BankAcctTmRec	string	帳戶交易明細	2	1	1
2	1	TmType	string	借貸別	0	2	1
2	2	PostedDt	double	交易日期	0	2	2
2	3	PostedTm	double	交易時間	0	2	3
2	4	TmDesc	string	交易摘要	0	2	4
2	5	AcctBal	string	帳戶餘額	3	2	5
3	1	BalType	string	餘額類別	0	2	6
3	2	CurAmt	string	交易金額	4	3	1
4	1	Amt	double	金額	0	3	2
4	2	ExpireDate	double	逾期日期	0	4	1
3	3	DeliveryMethod	string	通知方式	0	4	2

3.4.2 整合報表呈現方式

帳戶代理人進行整合報表需進行兩個步驟，第一步驟向共通平台索取標準欄位定義，也就是取得表 3-2 的關聯表格資料。第二步驟再向各銀行網路服務取得帳戶欄位實際值，並依據標準欄位的定義，將欄位實際值放入整合報表。此時，向銀行網路服務要求的實際值需要有顧客在該銀行的帳號和密碼，方能取得帳戶實際交易明細，見圖 3-6 所示，不同於圖 3-5 XML 檔的是有實際交易資訊在 XML 檔。

```

<BankAcctTmRec>
  <TmType>Credit</TmType>
  <PostedDt>2003/09/20</PostedDt>
  <PostedTm>11:26:50</PostedTm>
  <TmDesc>存簿存款</TmDesc>
  <AcctBal>
    <BalType>現金</BalType>
    <CurAmt>
      <Amt>10000</Amt>
      <ExpireDate/>
    </CurAmt>
    <DeliveryMethod/>
  </AcctBal>
</BankAcctTmRec>

```

圖 3-6 帳戶欄位實際值 XML 文件

之後，帳戶代理人將各銀行帳戶欄位實際值取回後，以標準欄位為整合規則，彙總所有銀行帳戶欄位實際值，呈現帳戶整合報表。結果示意如表 3-3 所示。

表 3-3 整合報表示意圖

銀行別	借貸別	交易日期	交易時間	交易摘要	帳戶餘額		逾期日期	通知方式
					餘額類別	交易金額		
					金額			
A銀行	Credit	37884	0.4769676	存簿存款	現金	10000		
B銀行	Credit	37905	0.3961806	轉帳	現金	20000		電匯
C銀行	Debit	37989	0.6322917	股票入股	現金	30000	2004/1/2	
...

逾期日期為 C 銀行特有的資料欄位元、通知方式為 B 銀行特有的資料欄位，透過共通平台合併各銀行的帳戶欄位成標準欄位，由帳戶代理人將實際值對應到適當欄位中，合併成帳戶整合報表。