

國立臺灣海洋大學

資訊工程學系

碩士學位論文

指導教授：張雅惠 博士

基於服務導向架構之洪氾預警系統

(初稿)

A Flood Forecasting System Based on
Service-Oriented Architecture

研究生：吳佩珊 撰

中華民國 101 年 7 月



基於服務導向架構之洪氾預警系統

A Flood Forecasting System Based on Service-Oriented Architecture

研 究 生：吳佩珊

Student：Pei-Shan Wu

指導教授：張雅惠

Advisor：Ya-Hui Chang

國 立 臺 灣 海 洋 大 學

資 訊 工 程 學 系

碩 士 論 文

A Thesis

Submitted to Department of Computer Science and Engineering

College of Electrical Engineering and Computer Science

National Taiwan Ocean University

In Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Computer Science and Engineering

July 2012

Keelung, Taiwan, Republic of China

中華民國 101 年 7 月



摘要

由於洪氾預警系統的建置，時常需要藉由各式各樣的地理水文資料進行傳遞交換，再經過異質系統中的各水理演算模組進行通訊合作，因此模組彼此之間的溝通橋梁成為重要的研究議題。在本論文中，我們基於 SOA（Service-Oriented Architecture）的概念，提出一個良好的分散式系統架構，並利用 WCF（Windows Communication Foundation）與 WF（Windows Workflow Foundation）的技術，實作有效的資訊整合平台，讓異質系統中的各水理演算模組能夠順利地取得與傳遞資料，且能夠直接進行溝通與合作。除此之外，我們將各水理演算模組的執行結果利用 Google Maps API 與 Google Chart API 的技術，搭配地圖與圖表適當地呈現於前端介面，提供使用者直接透過網路瀏覽器查詢瀏覽，以達成有效預警的目的。

Abstract

In a flood forecasting system, we need to handle a variety of geographic data and hydraulic data. Since many types of data need to be transmitted between hydraulic modules which operate in heterogeneous operating systems, the coordination and data exchange between these modules has become an important research issue. In this thesis, we propose to build a distributed system based on SOA (Service-Oriented Architecture), so that each module can successfully acquire the data it needs, and the programs of each module executed in heterogeneous operating systems can properly cooperate. This system is implemented by the techniques of WCF (Windows Communication Foundation) and WF (Windows Workflow Foundation). In addition, we apply the techniques of Google Maps API and Google Chart API to construct a graphical user interface, so that we can clearly show the warning message and the calculated result.

誌謝

首先，感謝指導教授張雅惠博士，對於本論文給予相當鼎力的協助，且在論文研究期間不時地共同討論，以及悉心指正與釋疑，並彌補學生於專業學識上的不足。由於老師的耐心督導，使本論文得以順利地完成，同時也讓學生獲益匪淺，在此向敬愛的老師致上衷心的謝忱。

除此之外，感謝口試審查委員郭忠義博士與馬尚彬博士，細心審稿以及論文修正，並對於本論文不吝提供寶貴的建議，使本論文更趨於嚴謹完善，在此深表謝意。

最後，感謝實驗室的學長姊，帶領我快速地適應實驗室的環境。感謝實驗室的同學與學弟妹，陪伴我渡過多采多姿的實驗室生活。感謝大學部的專題生與國科會其餘子計畫的助理們，於本系統整合建置階段給予許多幫助。感謝親愛的家人，於學習成長階段給予許多支持與鼓勵。在此一併致上心中無限的感激，謝謝你們。

目錄

摘要	i
Abstract	ii
誌謝	iii
圖目錄	v
表目錄	vi
第 1 章 序論	1
1.1 研究動機與目的	1
1.2 研究方法與貢獻	2
1.3 相關研究	4
1.4 論文架構	7
第 2 章 技術背景	9
2.1 水理模組介紹	9
2.2 水理模組資料介面	11
第 3 章 系統架構	15
第 4 章 水理服務與偵測站服務	19
4.1 網路服務規劃	19
4.2 工作流程控制	23
4.3 實驗	29

第 5 章	資料庫服務	32
5.1	關聯式資料庫綱要	32
5.2	資料庫服務介紹	35
第 6 章	系統預警呈現	42
6.1	系統網站架構	42
6.2	河川溢堤預警	44
6.3	二維淹水預警	45
6.4	沿海越波預警	46
6.5	流量資訊顯示	47
6.6	雨量資訊顯示	48
第 7 章	結論與未來方向	50
參考文獻		51
附錄		54
A.	Windows Communication Foundation	54
B.	Windows Workflow Foundation	55
C.	工作流程控制之 WF 實作範例	58

圖目錄

圖 2-1：水理模組關係圖	9
圖 2-2：林邊溪流域之地理環境圖	11
圖 3-1：系統架構圖	15
圖 3-2：雲端Web GIS架構圖	18
圖 4-1：水理服務內部運作流程圖	20
圖 4-2：網路服務資料介面－規範一	22
圖 4-3：網路服務資料介面－規範二	23
圖 4-4：工作流程控制圖－策略一	25
圖 4-5：工作流程控制圖－策略二	26
圖 4-6：工作流程控制圖－策略三	28
圖 4-7：輸出結果之總批次數	30
圖 4-8：輸入資料之平均時間差	31
圖 5-1：網路服務資料介面－規範三	36
圖 6-1：洪氾預警系統網站首頁	43
圖 6-2：河川溢堤預警圖	44
圖 6-3：二維淹水預警圖	45
圖 6-4：沿海越波預警圖	46
圖 6-5：流量資訊顯示圖	48
圖 6-6：雨量資訊顯示圖	49

表目錄

表 2-1：各水理模組輸出資料表.....	13
表 2-2：各偵測站輸出資料表.....	14
表 4-1：水理服務執行時間對照表.....	29
表 5-1：關聯式資料庫綱要－地理資料.....	33
表 5-2：關聯式資料庫綱要－Hydraulics Services 執行結果.....	34
表 5-3：關聯式資料庫綱要－Monitor Services 執行結果.....	35
表 5-4：關聯式資料庫 SQL 指令－History Service 範例.....	38
表 5-5：資料庫服務之 JSON 字串格式範例.....	38
表 5-6：關聯式資料庫 SQL 指令－Location Service 範例.....	39
表 5-7：關聯式資料庫 SQL 指令－RealTime_Chart Service 範例.....	40
表 5-8：關聯式資料庫 SQL 指令－RealTime_Map Service 範例.....	41

第 1 章序論

在此章，我們先敘述本論文的研究動機與目的，同時提出本論文的研究方法與貢獻，並針對相關研究進行討論，最後說明本論文各章節的內容規劃。

1.1 研究動機與目的

台灣地區雨量豐沛，年平均降雨量高達 2500 公釐，為世界平均值的 2.5 倍。常年因颱風與豪雨事件所帶來的降雨，而釀成嚴重的淹水災害，造成人民生命及經濟上的重大損失。近年來，由於全球氣候變遷的加劇影響，使得降雨狀況更加趨於極端而超過預期，導致災害範圍與程度相較於以往更加慘重。有鑑於淹水發生頻繁，突顯出堤防構造物並不能完全杜絕水患，因此我們希望建置一個洪氾預警系統，藉由即時的偵測資料有效模擬推估未來的水情預報，並利用合適的淹水警戒值研判預警區域，以掌握黃金時間疏散避災。

然而，建置一個完善的洪氾預警系統，需要結合數個具有不同功能的水理演算模式。例如，海岸越波演算模式能夠預測河川下游的水位高度，以及波浪越過防波堤而產生的越波量，而降雨逕流演算模式能夠預測河川上游的入流量，至於河道斷面演算模式能夠預測河川的溢岸流量，最後二維淹水演算模式能夠預測河川下游低窪地區的水深高度。因此，在整合與規劃各水理演算模式的運作過程中，我們考量到以下數個研究議題：

1. 如何建立一個良好的分散式系統架構，讓處於不同作業平台的各水理演算模式能夠彼此進行合作，使各模式順利地取得所需要的輸入資料，以及傳遞所獲得的輸出結果，並各自針對特定的地理目標，進行接連的模擬演算過程。
- 在不同作業平台方面，例如，海岸越波演算模式與二維淹水演算模式必須於 Linux 系統下執行，而降雨逕流演算模式與河道斷面演算模式則必須於 Windows 環境下執行。至於在資料傳遞方面，例如，海岸越波演算模式執行

所輸出的結果，其中河川下游水位必須傳遞給降雨逕流演算模式使用，而海岸越波量則必須傳遞給二維演水演算模式使用。

2. 如何設計一個合適的工作運作流程，可以考量到各水理演算模式落差懸殊的執行時間，使各模式不僅能夠依循正確的順序進行有效的執行，還能夠讓系統於單位時間內整體輸出的資料量（throughput）明顯提升。例如，海岸越波演算模式與二維淹水演算模式所耗費的時間是以小時為單位，而降雨逕流演算模式與河道斷面演算模式則是以分鐘為單位，因此針對不同的工作運作流程會讓系統輸出不同數量的預測值。
3. 如何建立一個親善的使用者介面，讓各水理演算模式所回傳的運算結果以及預警資訊能夠適當地呈現，並提供民眾與相關單位直接進行查詢瀏覽，以輔助於短時間內做出決策判斷，有效減少災害帶來的損失。

1.2 研究方法與貢獻

綜合以上討論，整個洪氾預警系統的建置，除了各水理演算模式的研發之外，更需要資訊相關技術的協助，才能夠使其成功地運行。首先，針對分散式系統架構方面，我們基於服務導向架構（Service-Oriented Architecture，簡稱 SOA）的系統設計方法進行規劃，以達成資料傳遞與作業整合的目的。服務導向架構是網際網路興起以來所發展的一種系統設計方法，其主要基礎是依據使用者的需求所組成的一組軟體模型，而組成的元素通常包含軟體元件、服務與流程三個部分。至於其主要概念是以「服務」做為導向出發，透過功能物件化的想法，將系統內部的「軟體元件」封裝為服務的形式對外提供，接續再以「流程」定義服務的相關處理步驟，以設計並建構整個系統。由於網路服務可以透過網路通訊協定（HTTP）以及 XML 標準格式的底層協定，進行合作溝通與資料傳遞，且其擁有不受限於特定技術與作業平台的中立特性，使異質系統的整合變得更加容易，而應用程式的再利用度，以及系統的彈性與擴充性亦可大為提升，甚至可以有效

降低系統管理層面的複雜度。因此在整個洪氾預警系統中，我們依據各水理演算模式的輸入與輸出需求，來設計各模式彼此間資料傳遞的介面規範，並將處於不同作業平台的各模式封裝成網路服務的形式，使其能夠順利地進行合作。除此之外，我們額外撰寫數個小型的網路服務，其中包含監控偵測站的網路服務、處理資料庫的網路服務等，以輔助我們進行整個系統的整合建置。

至於在實作的技術選擇方面，由於服務導向架構的發展與應用越來越廣，服務的建立方式亦趨於多元化，因此我們採用微軟公司於 .Net Framework 中所支援的 WCF (Windows Communication Foundation) 技術，而該技術的優點是其整合多種通訊協定的不同溝通模式，例如 .Net Remoting、Web Service、WSE、Microsoft Message Queuing (簡稱 MSMQ) 等，可方便系統日後的維護與擴充。不過，由於微軟公司的技術僅適用於 Windows 環境，並沒有直接提供其它作業平台 .Net Framework 的運行環境，因此我們必須額外針對 Linux 系統安裝開放式資源 Mono 套件，使其亦能夠支援 .Net Framework 所開發的應用程式，以滿足我們於此分散式異質系統中，統一使用 WCF 技術建立各網路服務的需求。

其次，針對工作運作流程方面，我們依據各網路服務的功能特性與執行時間差異，進行數種工作流程策略的規劃與討論。至於在實作的技術選擇方面，我們同樣採用微軟公司於 .Net Framework 中所提供的 WF (Windows Workflow Foundation) 技術，而該技術的優點是其能夠協助以圖形化的方式，設計使用者所需要的工作運作流程，且流程中所控制的活動物件，包含一般的應用程式專案與 WCF 技術所建立的網路服務。同時，此技術亦支援多種流程中常用的基礎活動，例如條件判斷 (ifElse)、迴圈 (while)、循序執行 (sequence)、平行執行 (parallel) 等，可方便我們進行不同工作流程策略的設計。基於上述優點，我們利用 WF 技術所提供的支援，協調控管洪氾預警系統內部各網路服務的工作流程邏輯。

最後，針對使用者介面方面，我們基於網路地理資訊系統 (Web Geographic

Information System，簡稱 Web GIS）的概念進行規劃，並利用 Google Maps API 與 Google Chart API 的技術，將各水理演算模式所回傳的運算結果以及預警資訊，搭配地圖與圖表的方式適當地呈現於網頁上，提供民眾與相關單位可以直接透過網路瀏覽器進行查詢瀏覽，以達成系統有效預警的目的。

本論文的貢獻總結如下所示：

1. 我們基於服務導向架構的系統設計方法，規劃洪氾預警系統的架構與工作運作流程的控管，以達成資料傳遞與作業整合的目的。
2. 針對此架構的實作，我們利用 .Net Framework 內部所提供的 WCF 技術，將各水理演算模式封裝成網路服務，並設計其資料傳遞的介面規範，使其順利地傳遞交換所需要的資料。再利用 .Net Framework 內部所提供的 WF 技術，協調控管洪氾預警系統內部各網路服務的工作運作流程，使其依循正確的順序進行有效的執行，且提升系統於單位時間內的資料輸出量。
3. 我們利用 Google Maps API 與 Google Chart API 的技術，成功地實作洪氾預警系統的使用者介面，以達成系統有效預警的目的。

1.3 相關研究

首先，我們討論洪水預警系統建置的相關研究。由於其時常需要藉由各式各樣的地理水文資料進行傳遞交換，再經過異質系統中的水理演算模式進行合作才能夠獲得結果。然而，獨立的電腦程式因缺乏資料的通訊能力而產生問題，因此有研究者提出建構於全球資訊網路的即時洪水系統[AMB03]。而論文[Li+06]提出的 Web-based Flood Forecasting System(簡稱 WFFS)，則是進一步利用多階層系統架構的優點，將系統架構劃分為三個分別負責不同工作的階層，其中「presentation tier」包含與使用者互動的介面，「application tier」負責執行水理演算模式，「data tier」則是用以儲存整個系統所需要的資料。由於工作經過良好的

分割，因此即使修改其中一階層，也不會使得另一階層需要進行大幅修改。除此之外，荷蘭 Delft Hydraulics 所發展的洪水早期預警系統，稱之為 FEWS(Flood Early Warning System)，該系統採用 XML 作為資料的表示方式，能夠解決各單位測量站資料格式不一的問題，因此臺灣水利署與其合作開發出適用於台灣水理環境的 FEWS-Taiwan 系統。而論文[黃+09]則是進一步利用 FEWS-Taiwan 作為資料的接收與運算平台，並將其資料庫結合 Google Maps API 的技術，因此可以直接透過網頁於網路瀏覽器上做即時的線上查詢與警戒測站的標示等。

其次，我們討論 SOA 系統規劃的相關研究。由於基於 SOA 的系統規劃方式，其主要概念是能夠提供網路服務單位，以建構具有彈性、可重複使用的整合性介面，加速達到網路服務提供的目標，因此近期被廣泛地應用[GDG10, Zheng+10, Zhu+11, 何+09]。其中論文[GDG10]即是利用網路服務技術進行複雜的演算應用程式，並建立一個整合性介面，提供特定的地理專家進行演算的參數調整。不同於該篇論文，我們主要是藉由網路服務技術協助整個洪氾預警系統的建置，並將演算結果與預警資訊以親善的介面呈現給一般民眾。而論文[Zheng+10]與論文[Zhu+11]則皆是基於 SOA 進行系統建置，前者是針對上海沿海區域，建置一個風暴災害風險的決策支援系統，後者是搭配 Web GIS 的使用，建置一個病蟲害地理區塊的預警系統。至於論文[何+09]則提及經濟部水利署建置的水資源資訊服務平台（Water Resources Information Services Platform，簡稱 WRISP）亦是基於 SOA 所發展，用以做為水利署署內及跨機關資訊交換必須的基礎建設，透過 WRISP 整合署內既有系統及資料庫，提供署內系統間彼此資訊整合的橋梁。

接續，我們討論網路服務流程管理的相關研究。其主要目標是希望藉由明確的流程定義，將一群獨立且耦合鬆散的活動，以網路服務的方式串聯成一個滿足需求的流程，並讓不同的流程彼此之間能夠相互溝通整合，以執行流程的運作與提供完善的服務。目前有許多關於流程管理的相關標準，例如 OASIS 組織制定的網路服務商業流程執行語言（Web Service Business Process Execution Language，

簡稱 WSBPEL) [WSBPEL]，其是以 XML 語法所撰寫而成，且可以定義操作變數以及多種結構化活動，而支援此 WSBPEL 程序的執行軟體，包含微軟公司的 BizTalk Server、IBM 公司的 BPWS4J、Oracle 公司的 BPEL Process Manager 等。除此之外，微軟公司亦提供 Windows Workflow Foundation (簡稱 WF) [WF]，支援一般應用程式專案與網路服務的工作流程控管，相較於上述 WSBPEL，其設計較為單純且容易實作，但是所支援的延展能力與容錯能力較為低。主要是因為 WF 協調控管的應用程式，其所傳遞資料的結構並非侷限於固定格式，而可以為開發者熟悉的基礎型別或自訂類別，且其通訊方式可以包羅萬象，例如 RPC、Web Service 等，至於 WSBPEL 的主要基礎則是偏重在遵循標準的資料結構與溝通方式，所以適合較為大型且複雜的系統整合。然而，針對本論文所建置的洪氾預警系統，由於其所涉及的系統整合情況較為單純，因此 WF 流程管理已足以滿足本系統整合建置的需求。

最後，在建置洪水預警系統的過程中，各水理演算模式間時常需要傳遞包含時間與空間的資料，因此我們討論地理資料處理的相關研究。為了讓不同 GIS 在進行地理資料的流通交換時，皆能採用共同標準規範的資料格式，以便在不同系統環境與應用程式下延用資料的建置及服務，所以有許多組織在研究地理資料共通的開放介面標準。例如開放式地理資訊系統協會 (Open Geospatial Consortium，簡稱 OGC) 所制定的 OpenGIS[OpenGIS]，於該標準中 GML (Geography Markup Language) 被提出來作為地理物件的表示格式，其是以 XML 為編碼基礎的語言，主要是對於地理資料中，地理圖徵的空間與非空間屬性進行模式化、傳輸與存取。除此之外，Google 旗下的子公司 Keyhole 亦研究開發 KML (Keyhole Markup Language) [KML]，其同樣是以 XML 為編碼基礎的語言，相較於上述 GML，其針對地理物件的定義涵蓋範圍較少且較為單純易懂，而主要是透過 Google Earth、Google Map 進行展示地理資料的空間位置，由於其應用逐漸受到重視，所以亦被納入 OpenGIS 標準規範內。然而，針對本論文所建置的

洪氾預警系統，由於各水理演算模式所輸出的結果皆為未來一至三小時的預警資訊，牽涉到許多時間同時搭配空間的資料，而上述標準規範內所定義的地理物件，並未完全符合本系統的需求，因此目前無法直接於本系統中套用，但若未來期望僅呈現特定時間的地理資料，則可以將標準規範納入系統設計的考量。

另一方面，除了針對廣義的地理資訊系統所制定的標準規範之外，水理方面的專家組織亦有提出水理模組整合的介面標準 OpenMI (Open Modeling Interface) [OpenMI]，目前亦被廣泛地應用[Bu+10, BS11, 林+09]。該標準的目的是希望水理模組於執行時，能夠在時間疊代的基礎下動態地交換資訊。而針對資料交換介面部分，其提出如何定義空間(Where)、時間(When)、數值的物理意義(What)與數值本身。至於針對水理模組串接部分，其提出採用回拉機制(Pull Mechanism)，由需求方透過 GetValues()方法向來源方要求所需要的資料。然而，針對本論文所建置的洪氾預警系統，除了需要處理各水理模組的串接外，亦需要將各模組輸出的演算結果自動匯入資料庫中，以方便系統網站做預警資訊的呈現，因此我們希望各水理模組盡可能增加執行次數，以提升預警資訊的可信度。但是由於考量各水理模組的執行時間落差懸殊，所以本系統較不適合採用該標準的回拉機制(Pull Mechanism)，而是選擇採用推行機制(Push Mechanism)的概念來控管各模組的運作，讓執行時間較短的模組不受執行時間較長的模組影響，而能夠持續不斷地執行並將演算結果輸出，以提升系統於單位時間內輸出的預測值。

1.4 論文架構

本論文其餘各章節的架構如下：於第二章敘述本論文的相關技術背景，其中包含既有水理模組的主要功能與資料介面，藉以對本論文的研究有基礎的認識。於第三章提出基於 SOA 的整體系統架構，並介紹系統架構的多階層設計。於第四章說明水理演算與偵測站相關的網路服務規劃，以及網路服務的工作流程控制，並以實驗進行流程策略的分析討論。於第五章針對資料庫服務的相關細節做介紹，

其中包含關聯式資料庫綱要的設計與 SQL 指令。最後於第六章展示系統介面的預警呈現，並於第七章提出本論文的結論與未來方向。

第 2 章技術背景

在此章，我們先介紹既有水理模組的主要功能以及與其它模組的相關性，接續再說明各水理模組的資料介面，藉以對本論文的研究有基礎的認識。

2.1 水理模組介紹

本論文建置的洪氾預警系統包含數個既有的水理模組，各模組所對應的水理演算模式與之間所傳遞的資料，如圖 2-1 所示。由於水理演算模式的細節研究非屬本論文著重的焦點，所以我們僅介紹各水理模組的主要功能以及與其它模組的相關性，藉以了解各水理模組彼此之間的資料流向與複雜的相互作用。

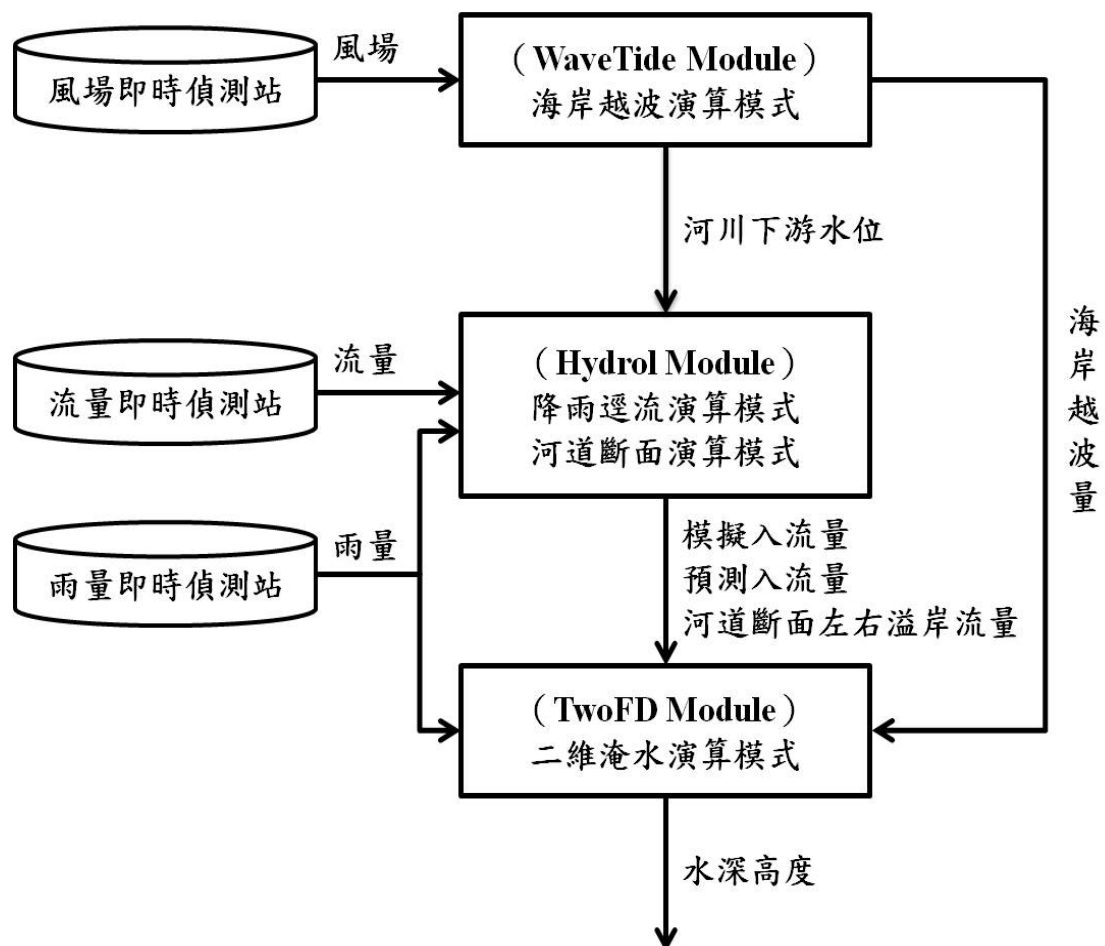


圖 2-1：水理模組關係圖

如圖 2-1 所示，首先，整個水理模組關係圖的起始點為「WaveTide Module」，其將執行海岸越波演算模式。由於颱風期間的沿海水域時常同時出現暴潮與巨浪，而當河川大量的洪水抵達出海口位置時，若受到潮汐漲落與巨浪影響，將導致河川的迴水效應。如此一來，海岸區域不僅會受到原本河川洪水引致的淹水災害，亦會因高水位與巨浪聯合引致的越波量影響，而加重沿海區域淹水的嚴重程度。因此，此模組的目的在於利用即時偵測的風場資料，結合暴潮水位的模擬分析，演算河海口未來一至三小時的暴潮水位以及海岸保護結構物的越波量，進而依其演算結果模擬沿海低窪地區的海水溢淹情形。此模組輸出的資料，將做為「Hydrol Module」與「TwoFD Module」執行演算時的下游邊界條件。

其次，「Hydrol Module」獲得「WaveTide Module」的輸出結果後，其將先執行降雨逕流演算模式。該模式將研究流域劃分為上游次集水區、支流次集水區，以及殘流域等多個演算區塊，以考慮空間降雨的非均勻特性。而後進行降雨逕流演算，並結合氣象單位偵測站的即時雨量與流量資料，發展即時逕流校正方式，以提高預測的精準度與效能。該模式所模擬預測的輸出資料為未來一至三小時的上游入流量、河道支流入流量與殘流域側入流量，以提供後續模式執行演算使用。其中所謂的模擬入流量是依據偵測站的即時雨量資料，模擬其流入演算區塊的流量數值，至於預測入流量則是依據上述的模擬入流量，進一步演算並預測未來可能的流量數值。接續，「Hydrol Module」將再執行河道斷面演算模式。該模式的目的在於考慮不確定土石量影響，建立集水區河道斷面的洪水模擬演算。在執行演算之前，該模式將先依據河道各斷面量測的地形資料，繪製河道各斷面圖。而當颱風來臨時，該模式則即時考量上游陡坡河川側坡崩塌、土石流由支流河谷進入主流河川，以及土石隨洪水進入下游平原河川段等不同河川淤積型態，進而預測河道各斷面未來一至三小時的水位高度、左溢岸與右溢岸的流量，並依其演算結果研判河川可能於何段發生溢堤情形。此模組輸出的資料，將做為「TwoFD Module」執行演算時的上游邊界條件。

最後，「TwoFD Module」集結之前兩個模組所提供的演算結果，執行二維淹水演算模式。此模組將利用多核心伺服器進行平行運算數值模擬，並配合空間與時間的離散處理方式，以發展高效能的二維淹水演算，縮短傳統二維淹水演算計算耗時的缺失，進而預測演算河川下游低窪地區未來一至三小時的水深高度，以做為即時淹水預警區域的研判依據。

2.2 水理模組資料介面

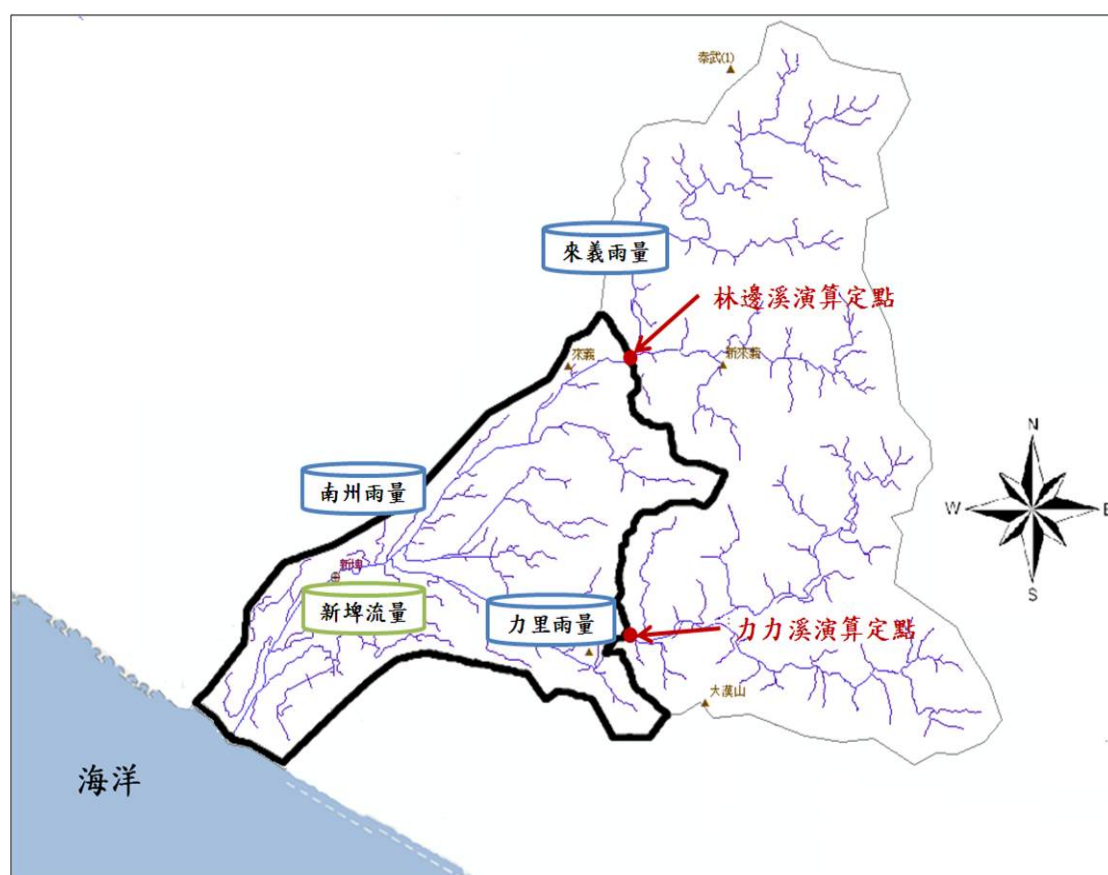




圖 2-2：林邊河流域之地理環境圖

本論文建置的洪氾預警系統，是以台灣南部林邊河流域為研究案例，進行洪氾預警的模擬與演算。而該演算區塊的地理環境大致如圖 2-2 所示，其中「WaveTide Module」是以林邊河流域附近的海洋進行演算，而「Hydrol Module」則是將林邊河流域劃分為主幹流域林邊溪與分支流域力力溪分別進行演算，至於

「TwoFD Module」則是以林邊溪流域附近的陸地進行演算，主要為圖 2-2 中黑色粗體框的區域，後既有水理模組執行演算所需要的雨量與流量資料，則是以南州雨量偵測站、來義雨量偵測站、力里雨量偵測站以及新埤流量偵測站為主。

承接前述章節的水理模組介紹，為了方便解決既有水理模組的資料交換問題，以下我們針對既有水理模組實際的輸出資料進一步做說明，其輸出資料的格式與範例內容統整如表 2-1 所示。首先，說明既有水理模組執行演算皆會處理到的資料格式，其中時間資料的表示方式為“西元年-月-日 時:分:秒”(YYYY-MM-DD Hr:Mi:Se)，而空間資料的表示方式則是基於 TWD97 (Taiwan Datum 97, 1997 台灣大地基準)的二度分帶座標。其次，依據既有水理模組各自的演算需求，說明其餘輸出結果的資料格式，其中水位的單位為公尺 (m)，越波量的單位為每秒平方公尺 (m^2/sec)，至於雨量的單位為公釐 (mm)，流量與左右溢岸的單位均為每秒立方公尺 (m^3/sec)，而水深的單位則為公分 (cm)，並以數值 1 與 0 分別表示警示的「是」與「否」。

水理模組	輸出檔名	資料格式與範例內容
WaveTide	HydrolInput.dat (資料數：6 筆)	資料格式： (時間，二度分帶座標 X，二度分帶座標 Y，水位) 範例： (2012-06-06 12:00:00, 198738.643, 2480062.259, 0.0098)
	TwoFDInput.dat (資料數：33 筆)	資料格式： (時間，二度分帶座標 X，二度分帶座標 Y，越波量，警示) 範例： (2012-06-06 12:00:00, 200588.092, 2478948.852, 6.8, 1)
Hydrol	K_Real00.dat K_Pred00.dat K_Real02.dat K_Pred02.dat (各檔資料數：19 筆)	資料格式： (時間，雨量，流量) 範例： (2012-06-06 12:00:00, 8.63, 2.1)
	SecData_LinPien +編號.dat (編號 00 至 43) SecData_Lili +編號.dat (編號 00 至 18) (各檔資料數：19 筆)	資料格式： (時間，水位，流量，左溢岸，右溢岸) 範例： (2012-06-06 12:00:00, 0.098, 0.012, 729.971, 109.793)

TwoFD	Output.txt (資料數：0~10 ⁴ 筆)	資料格式： (時間，二度分帶座標 X，二度分帶座標 Y，水深) 範例： (2012-06-06 12:00:00，201944.681，2485393.236，38.5)
--------------	---	--

表 2-1：各水理模組輸出資料表

由於既有水理模組皆是利用 FORTRAN 程式語言所撰寫，其習慣於處理文字檔案，因此所有模組的輸出皆是以單純的文字檔案呈現。如表 2-1 所示，其一，就「WaveTide Module」而言，其會依據其它模組的不同需求輸出兩個檔案，其中「HydrolInput.dat」內是存放林邊溪流域下游各演算定點的水位高度，提供「Hydrol Module」執行演算使用，而「TwoFDInput.dat」內則是存放沿海區域各演算定點的海岸越波量以及警示資訊，提供「TwoFD Module」執行演算使用。

其二，就「Hydrol Module」而言，其會針對主幹流域林邊溪與分支流域力力溪分別進行演算。而由於河川上游並未架設流量偵測站，因此該模組會利用雨量偵測站的資料執行模擬演算產生模擬入流量，接續再利用該模擬入流量執行預測演算產生預測入流量，最後依據河川下游流量偵測站的資料校正演算的準確度。以主幹流域為例，該模組執行模擬演算產生的模擬入流量儲存於「K_Real00.dat」中，而至於「K_Pred00.dat」內儲存的資料，則是該模組執行預測演算產生的預測入流量。同理，分支流域亦是進行相同的演算過程，差異僅在於其演算結果是分別儲存於「K_Real02.dat」與「K_Pred02.dat」中。

除此之外，「Hydrol Module」還會針對主幹流域林邊溪的 44 個橫斷面，以及分支流域力力溪的 18 個橫斷面，進行河道橫斷面的預測演算，其所得到的水位高度、左右溢岸的流量等，將分別儲存於對應編號的檔案「SecData_LinPien+編號.dat」與「SecData_Lili+編號.dat」中。其三，就「TwoFD Module」而言，其會將演算區域劃分為數個網格演算定點，並依據上述水理模組的輸出資料，進行二維淹水的預測演算，而各演算定點的水深結果則是存放至「Output.txt」中。

另一方面，我們針對既有水理模組執行演算所需要的雨量、流量與風場資料進一步做說明。由於為了符合既有水理模組執行演算的輸入需求，因此各偵測站實際偵測到的檔案，其必須先透過資料格式轉換的應用程式，進行檔案內容處理後才能夠使用，至於處理後的資料格式與範例內容統整如表 2-2 所示。首先，時間資料的表示方式同樣為“西元年-月-日 時:分:秒”(YYYY-MM-DD Hr:Mi:Se)，且雨量的單位亦為公釐(mm)，而流量的單位亦為每秒立方公尺(m^3/sec)。其次，關於各雨量偵測站的站號，其中南州雨量偵測站為 541，來義雨量偵測站為 2401、力里雨量偵測站為 2402。除此之外，新埤流量偵測站的站號則為 1760H004。最後，風場資料內各網格點的表示方式是基於 TWD97 (Taiwan Datum 97, 1997 台灣大地基準) 的經緯度座標，於各網格點 X 方向與 Y 方向的風場速度，其單位則皆為每秒公尺(m/s)。

偵測站	輸出檔名	資料格式與範例內容
雨量偵測站	NowRainFall.txt (資料數：3 筆)	資料格式： (偵測站站號，時間，雨量) 範例： (541, 2012-06-06 12:00:00, 5.36)
流量偵測站	NowFlow.txt (資料數：1 筆)	資料格式： (偵測站站號，時間，流量) 範例： (1760H004, 2012-06-06 12:00:00, 9.12)
風場偵測站	NowWind.txt (資料數： 10^4 筆)	資料格式： (時間，經度 X，緯度 Y，風場速度 X，風場速度 Y) 範例： (2012-06-06 12:00:00, 120.516, 22.432, -3.23, -1.99)

表 2-2：各偵測站輸出資料表

第 3 章系統架構

如前述章節所介紹，本論文的目的以分散式異質系統為計算基礎，整合既有的水理模組建構完善的洪氾預警系統。由於整個洪氾預警的模擬與演算，不僅需要透過各式各樣的地理水文資料進行傳遞交換，亦牽涉到分散式異質系統中，既有水理模組的連結計算，同時必須將其演算結果於短時間內進行預警資訊呈現。因此，考量到未來系統整合的彈性與功能的擴充性，我們提出一個基於 SOA 的資料傳遞與作業整合平台，讓既有水理模組順利地傳遞交換所需要的資料，並協調管既有水理模組的執行作業，使其不僅能夠進行跨平台的通訊合作，亦能夠達成資源共享與洪氾預警的功能，至於整個系統架構如圖 3-1 所示。

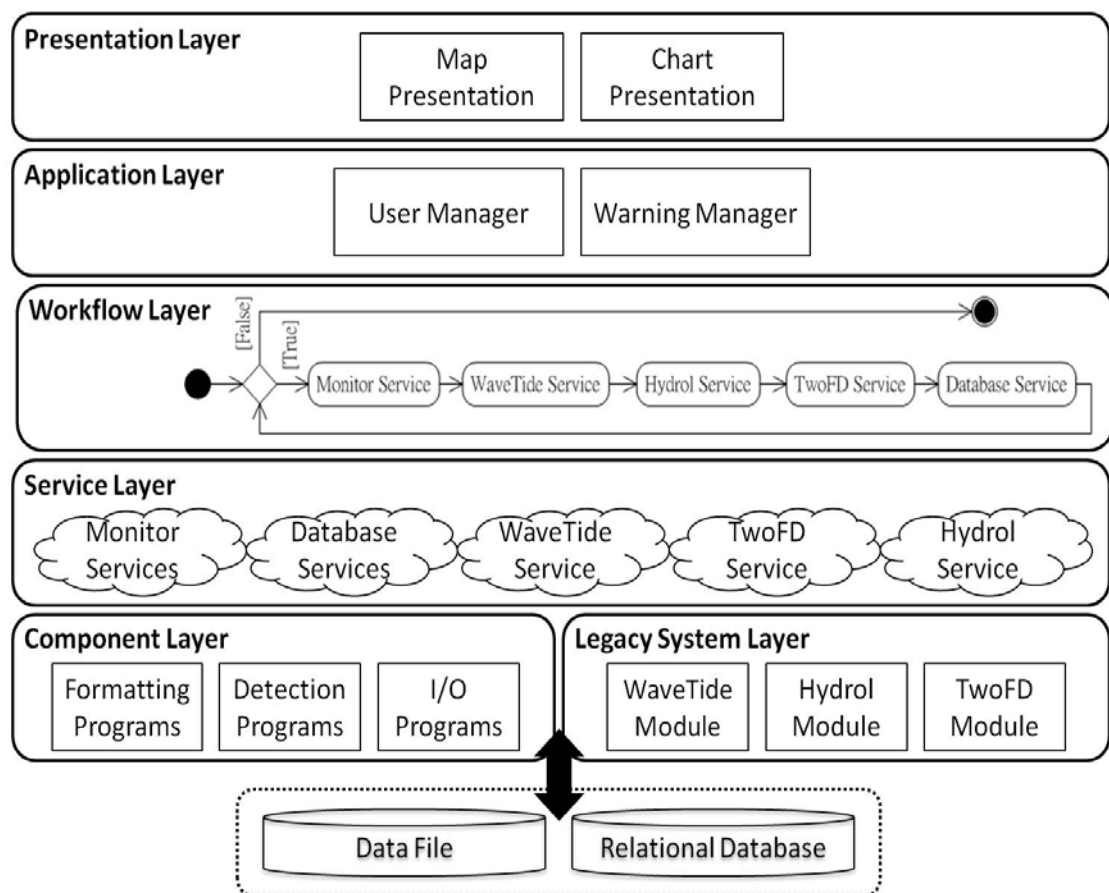


圖 3-1：系統架構圖

首先，在最底層的虛線方框中，我們將系統內部的資料規劃為兩大類型儲存。其一為「Data File」，包含各應用程式的執行輸出結果。其二為「Relational Database」，其是為了方便未來系統的資料維護，以及使用者介面的預警資訊呈現，所以將各應用程式的執行輸出結果經過處理後，匯入我們事先設計好的關聯式資料庫中。

其次，Legacy System Layer 是涵蓋既有水理模組的應用程式，包含執行海岸越波演算模式的「WaveTide Module」、執行降雨逕流演算模式與河道斷面演算模式的「Hydrol Module」，以及執行二維淹水演算模式的「TwoFD Module」，這些應用程式皆是利用 FORTRAN 程式語言所撰寫。至於 Component Layer 則是概括為了整合系統所額外撰寫的數個小型應用程式。其中，「Formatting Programs」負責資料格式的轉換，例如將來自雨量偵測站的累計雨量資料，轉換為每十分鐘的降雨量，以滿足既有水理模組執行演算的輸入需求。而「Detection Programs」是負責監控各偵測站的同步目錄是否獲取新的偵測資料，包含風場偵測站、流量偵測站以及雨量偵測站等。其主要是由於本系統目前與各偵測站的聯繫方式，是透過在 FTP 伺服器上建立同步目錄的功能，以達成即時接獲偵測站資料的目的。除此之外，「I/O Programs」則是負責將各應用程式的執行輸出結果自動匯入關聯式資料庫中，或是從關聯式資料庫中查詢所需的資料。

接續，根據 Legacy System Layer 中的既有水理模組與 Component Layer 中的輔助應用程式，我們建立數個網路服務於 Service Layer 中，例如水理模組的網路服務，分別為「WaveTide Service」、「Hydrol Service」與「TwoFD Service」，以及監控偵測站的網路服務「Monitor Services」等，以解決分散式異質系統的溝通整合問題，並增加既有水理模組的再利用價值。除此之外，還有處理資料庫的網路服務「Database Services」，其功能是將上述各網路服務的輸出結果自動匯入關聯式資料庫中，以及協助查詢於使用者介面呈現的洪氾預警資訊，讓系統能夠統一控管與資料庫的相關聯繫。至於 Workflow Layer 則是再依據上述網路服務各

自的特定功能，並額外搭配一些工作流程控管的基礎活動，例如條件判斷活動、迴圈活動、循序執行活動、平行執行活動等，以協調各網路服務的執行運作，讓整個洪氾預警的模擬與演算能夠順利完成。

除此之外，Application Layer 負責實作整個洪氾預警系統的網站建置，其大致可以劃分為兩大類型管理應用程式。其中「User Manager」是依據使用者身分的不同，給予不同的系統瀏覽與操作權限，例如刪除過往歷史資料的操作權限僅配置給系統管理者。至於「Warning Manager」則是負責從關聯式資料庫中，抓取各水理模組的執行結果，並經過特定的警戒判定條件過濾後，告知 Presentation Layer 應警戒的區域。

最後，Presentation Layer 負責前端使用者介面的洪氾預警呈現。一方面，由於 Google 提供許多免費的地圖服務與 Google Maps API，允許開發者在不需要建立 GIS 地圖伺服器的情況下，便可以將 Google Maps 地圖資料嵌入到網路應用中。另一方面，由於 Google 亦提供許多免費的圖表服務與 Google Chart API，允許開發者直接利用內建的物件參數，於網路應用中實現各種統計圖表的繪製。綜合上述 Google 所提供的功能，不僅足以滿足本系統對地理資訊的獲取、儲存、查詢、展示等需求，亦足以支援本系統動態繪製統計圖表方面的應用。同時，基於 Google 龐大的雲端基礎建設，能夠在資源有效利用的條件下，於網路上進行動態延展計算與儲存的能力，提供使用者快速的回應速度。因此，Presentation Layer 將利用開放網路資源 Google Maps API 與 Google Chart API 的技術建立 Web GIS，協助「Map Presentation」以地圖上的警戒圖示標記演算的預警所在地點，同時輔助「Chart Presentation」以時間序列的統計圖表展示演算的數值結果，以便使用者可以直接透過網路瀏覽器掌握洪氾預警資訊，至於其對應的架構如圖 3-2 所示。

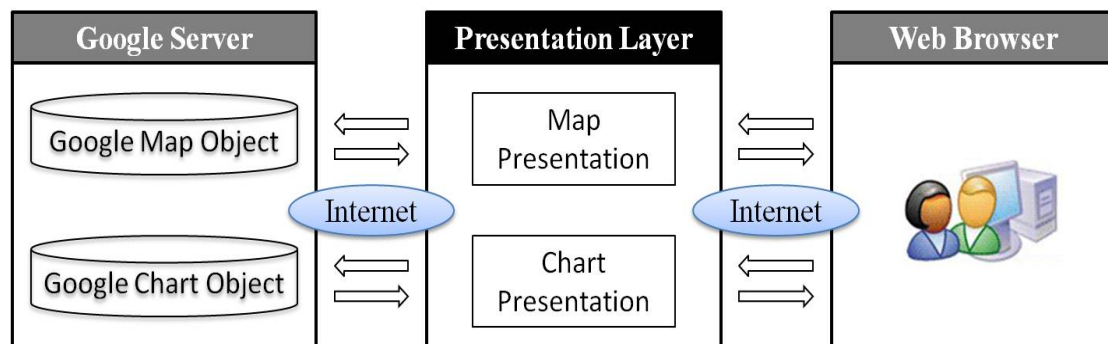


圖 3-2：雲端 Web GIS 架構圖

第 4 章水理服務與偵測站服務

在此章，我們先介紹系統內部所規劃的水理服務與偵測站服務，接續再針對協調控管各網路服務的工作流程控制做策略分析，並進行實驗討論。

4.1 網路服務規劃

關於洪氾預警系統中網路服務的建立，在 Windows 環境下，我們使用 Microsoft Visual Studio 2010 軟體，在 Linux 系統下，我們則是使用 MonoDevelop 2.4 軟體。透過這兩個軟體可以協助使上述兩種作業平台，其各自所架設的網站伺服器，皆能支援 .Net Framework 應用程式，以便我們於此分散式異質系統中，順利地進行網路服務的封裝與共享。而網路服務實作的底層技術，可以參考附錄 A. Windows Communication Foundation (簡稱 WCF) 的介紹。至於系統中所建立的網路服務可以劃分為三大類型，分別為水理服務 (Hydraulics Services)、偵測站服務 (Monitor Services) 與資料庫服務 (Database Services)，在此節中將介紹其設計理念與資料介面規範。

首先，第一類型 Hydraulics Services，主要是將既有水理模組的應用程式封裝成可以對外提供模擬演算的網路服務。透過前述章節的圖 2-1 與表 2-1，我們可以知道既有水理模組皆是利用文字檔案做為其資料輸入與輸出的主要形式。因此，我們考量在不變動任何水理模組應用程式的情況下，選擇採用文字檔案內容直接轉換字串型態的方式，做為網路服務的資料輸入與輸出介面，協助其彼此間順利地進行資料的取得與傳遞，此類型網路服務內部運作流程大致如圖 4-1 所示。在圖 4-1 中，一開始我們會將網路服務的每個字串型態的輸入參數，分別寫入對應檔案名稱的輸入文字檔案中，以提供水理模組的應用程式使用。接續，我們會利用程序呼叫的方式執行水理模組的應用程式，直至其演算完畢後，再針對水理模組應用程式的輸出文字檔案，進行讀取動作並分別儲存至各別字串型態的輸出

參數，以做為網路服務的執行結果。如此一來，透過上述相似的運作流程，我們便可以將既有水理模組分別封裝成對應的網路服務，使其可以順利地取得所需要的輸入資料，以及傳遞所獲得的輸出結果。

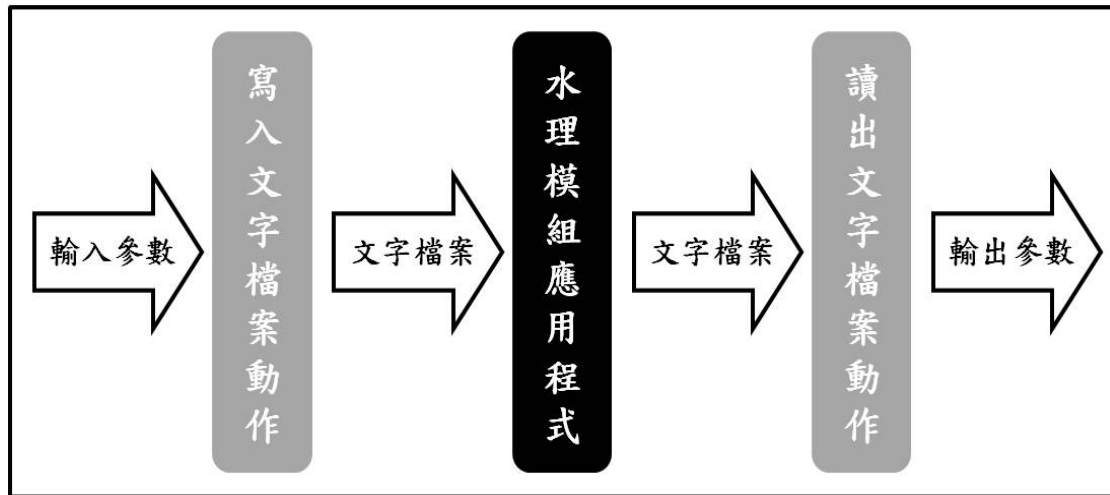



圖 4-1：水理服務內部運作流程圖

然而，針對圖 4-1 水理服務的封裝方式，雖然其設計理念與實作方法較為單純明瞭，且可以於短時間內完成文字檔案與字串型態的轉換動作，以提供既有水理模組直接執行演算之用。不過，在實際使用此類型網路服務的運作過程中，我們發現透過網路服務所能傳遞的輸入與輸出參數，其內容大小有配額的限制。而參照前述第 2.2 章節所介紹，部分既有水理模組的資料輸出檔案，若將其轉換成對應的網路服務輸出參數，其內容大小將存在超過配額限制的疑慮，例如「TwoFD Services」便為此疑慮的典型代表。因此，針對存在此疑慮的網路服務，我們選擇將其輸出參數額外進行字串的壓縮動作，且其它與其相關的網路服務，於使用輸入參數前需額外進行字串的解壓縮動作，以順利地達成傳遞演算結果的目的。舉例來說，一般 WCF 網路服務傳遞的訊息大小預設為 65,536 個位元組，我們假設「TwoFD Services」輸出一萬筆預警資料，則其對應的文字檔案大小會是 258,478 個位元組，至於經過壓縮處理後則會是 62,936 個位元組，亦即字串壓縮率約為 25%，因此該作法足以解決本系統的疑慮。

至於 Hydraulics Services 的相關資料介面規範，我們將其統整於圖 4-2 右側，而輸入與輸出參數的命名方式，基本上配合既有水理模組的輸入與輸出檔案名稱。我們可以藉由將圖 4-2 與前述章節的圖 2-1 進行相互參照，以更清楚掌握其中的變化。其一為既有「WaveTide Module」封裝而成的「WaveTide Service」，輸入參數是風場偵測站的即時風場（NowWind），而輸出參數則是河川下游水位（HydrolInput）與海岸越波量（TwoFDInput）。其二為既有「Hydrol Module」封裝而成的「Hydrol Service」，輸入參數包含流量偵測站的即時流量（NowFlow）、雨量偵測站的即時雨量（NowRainFall），以及「WaveTide Service」輸出的河川下游水位（HydrolInput），而輸出參數則包含控制點的模擬入流量（K_Real）與預測入流量（K_Pred），以及力力溪（SecData_Lili）與林邊溪（SecData_LinPien）的河道斷面左右溢岸流量  其三為既有「TwoFD Module」封裝而成的「TwoFD Service」，輸入參數包含雨量偵測站的即時雨量（NowRainFall）、「WaveTide Service」輸出的海岸越波量（TwoFDInput），以及上述「Hydrol Service」輸出的所有資料，而輸出參數則是二維水深高度（TwoFDResult）。

其次，第二類型 Monitor Services，主要是提供各偵測站的監控服務，並回傳既有水理模組執行演算所需要的即時偵測資料。由於目前與各偵測站的聯繫方式，我們是透過在 FTP 伺服器上建立共享目錄，並利用 FTP 軟體支援的同步目錄功能，設定自動化排程與相關需求，以達成即時接獲偵測站資料的目的。因此，此類型網路服務內部的運作流程，則是在不需要輸入任何資料的情況下，我們直接利用監聽物件持續監聽指定的同步目錄下是否有新檔案產生，直至監聽到有新檔案產生後，先針對該檔案進行資料格式的轉換處理，以便符合既有水理模組的輸入需求，接續再進行檔案內容讀取動作，並將其以字串型態儲存至輸出參數，以做為網路服務的執行結果。

至於 Monitor Services 的相關資料介面規範，我們將其統整於圖 4-2 左側。而參照前述章節的圖 2-1 與表 2-2，依據既有水理模組執行演算的輸入需求，各

偵測站將會分別封裝成三個網路服務，其一為風場偵測站的「Detect_Wind Service」，輸出參數為即時風場資料（NowWind），以提供「WaveTide Service」執行演算使用。其二為流量偵測站的「Detect_Flow Service」，輸出參數為即時流量資料（NowFlow），以提供「Hydrol Service」執行演算使用。其三為雨量偵測站的「Detect_RainFall Service」，輸出參數為即時雨量資料（NowRainFall），以提供「Hydrol Service」與「TwoFD Service」執行演算使用。

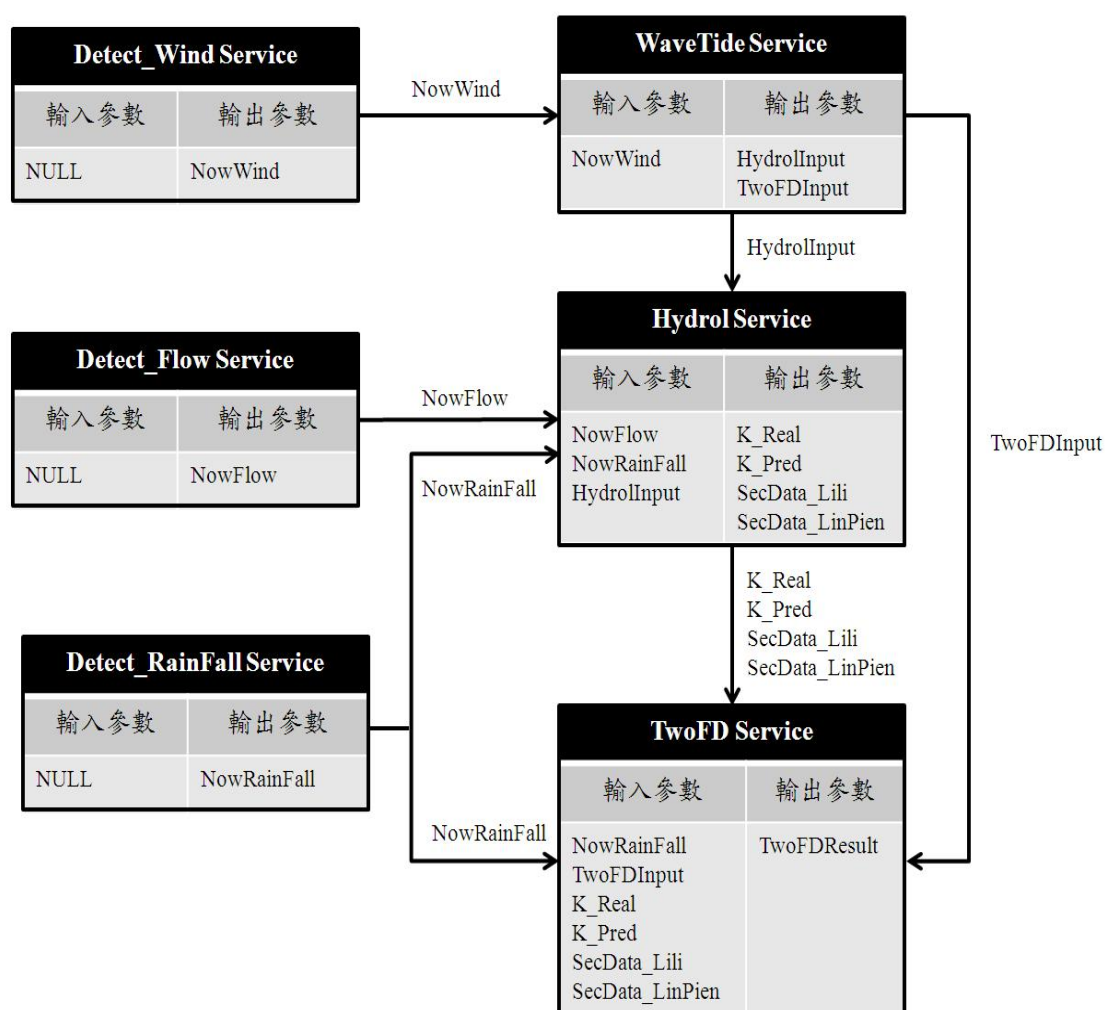


圖 4-2：網路服務資料介面 - 規範一

最後，第三類型 Database Services，主要是提供與關聯式資料庫進行互動的服務，該類網路型服務又可細分為兩部分進行介紹。第一部分是由於上述 Hydraulics Services 與 Monitor Services 的執行結果，其均需要自動匯入關聯式資

料庫中，進行資料的維護與管理。因此，我們利用上述兩類型網路服務的輸出參數，做為此類型網路服務的輸入參數，且依據後續章節會介紹的關聯式資料庫網要，將輸入參數進行適當地處理並匯入關聯式資料庫中，直至其完成動作後，皆分別以一個布林型態的輸出參數做為結束的標記。至於此類型網路服務的相關資料介面規範，我們將其統整於圖 4-3 中。除此之外，第二部分是由於使用者介面所呈現的洪氾預警資訊，其需要藉由此類型網路服務，協助進行關聯式資料庫的資料查詢。然而，因為該部分與後續章節會介紹的關聯式資料庫網要與 SQL 指令有密切的相關性，所以我們統一於後續章節一併做介紹。

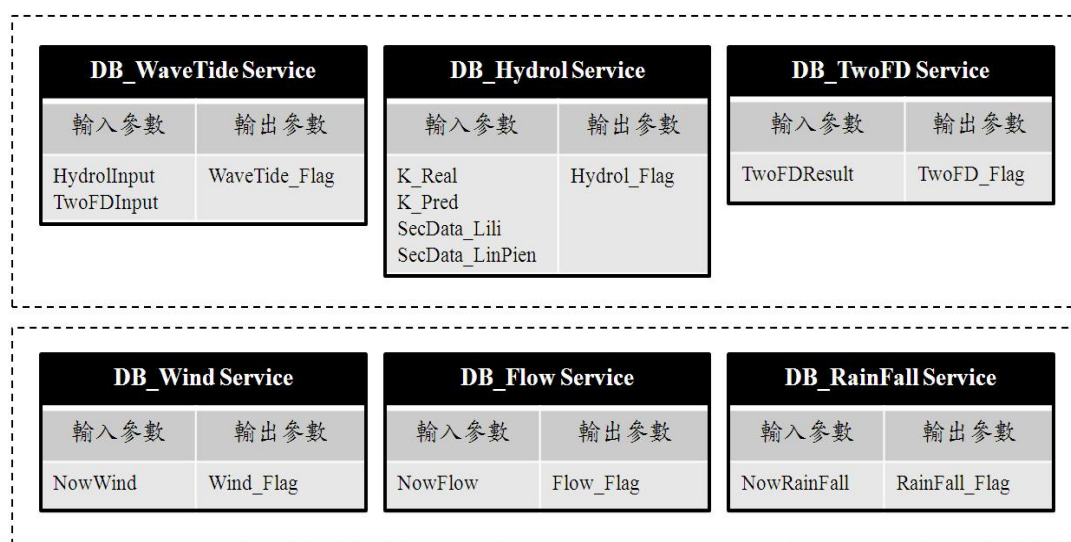


圖 4-3：網路服務資料介面 - 規範二

4.2 工作流程控制

關於洪氾預警系統中工作流程的控制，我們使用 Microsoft Visual Studio 2010 軟體，統一於 Windows 環境下以圖形化的方式協調各網路服務的執行運作，並額外搭配一些工作流程控管的基礎活動，以便我們於此分散式異質系統中，順利進行網路服務的溝通與合作。而工作流程實作的底層技術，可以參考附錄 B. Windows Workflow Foundation（簡稱 WF）的介紹。至於系統中所規劃的工作流程控制，在此節中將介紹其編排方式的設計理念。

承接前述章節網路服務規劃的介紹，我們將依據各網路服務的功能特性與執行時間差異，針對不同工作流程的編排方式進行分析比較，而經由工作流程所控管的網路服務，其中包含Hydraulics Services、Monitor Services，以及Database Services中負責將演算結果匯入關聯式資料庫的服務¹。以下我們分別提出三個工作流程的控制策略，並藉由UML（Unified Modeling Language）中的活動圖（Activity Diagram）表示之。其中每一個圓角矩形方框為一個網路服務，其命名方式與前述章節的網路服務規劃相同，另外標記「Assign」的圓角矩形方框表示指派特定值到工作流程中的變數，而菱形方框表示條件判斷的節點，至於每上下兩個黑色長條所包覆的區塊表示平行執行處理，而箭頭表示循序執行處理。另一方面，每個工作流程內皆會存在多個額外變數，其主要目的是用以儲存各網路服務的輸出參數，並再利用其做為相關網路服務所需要的輸入參數，以順利達成網路服務彼此間資料傳遞的需求。在圖中我們省略表示這些額外變數，但是為了對照起見，等同於圖 4-4 的WF實作介面可以參考附錄C。除此之外，針對工作流程的控制策略，我們主要是基於推行機制（Push Mechanism）的概念進行設計，由於考量各網路服務落差懸殊的執行時間，我們將上述WF流程裡的額外變數視為推行機制的緩衝器（Buffer）作用，以儲存網路服務的輸出資料，不過該額外變數會一直被目前最新的資料所更新覆蓋，以滿足本系統即時預警的需求。

策略一如圖 4-4 所示，每一個 Hydraulics Services 必須等待其它相關網路服務輸出當前的執行結果，整個工作流程才可繼續向下進行。意即每獲取一次當前的資料，網路服務才會因而執行一次演算，因此該策略的優點是工作流程相當簡單明瞭。同時，彼此沒有相關性的網路服務則可以同時執行，以提升工作流程整體的效能。在圖 4-4 中，首先，工作流程的起始點會先經由一個「Active」的條

¹在 Database Service 中，其中 DB_Wind Service 的即時風場資料，由於相關人員日後沒有查閱的需求，因此在工作控制流程中並沒有將其納入考量。

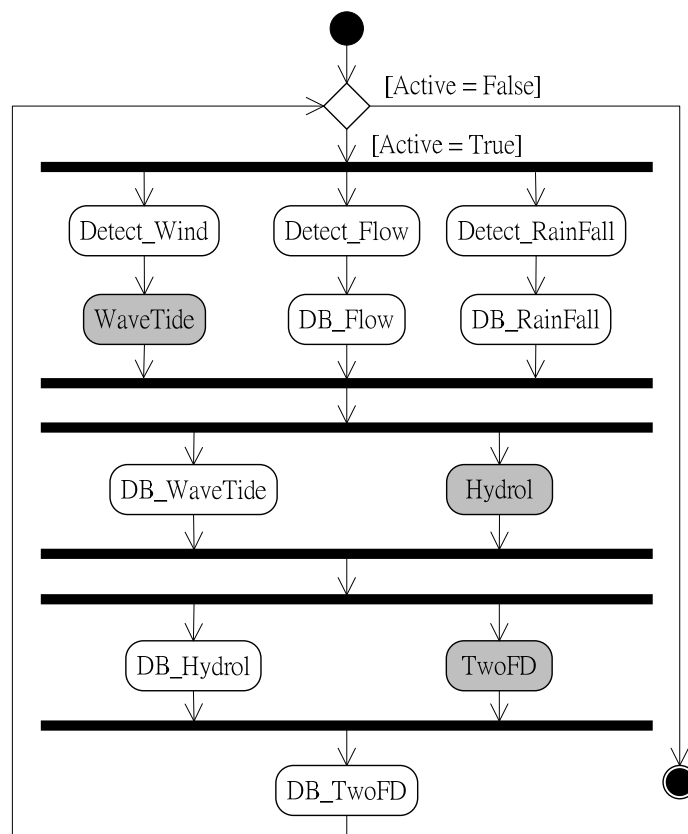


圖 4-4：工作流程控制圖－策略一

件判斷，以確認工作流程於啟動狀態，而後會同時循序執行三部分活動。其一是透過「Detect_Wind Service」偵測風場資料，直至獲取即時偵測資料後，便可執行「WaveTide Service」進行演算。其二與其三則是分別藉由「Detect_Flow Service」與「Detect_RainFall Service」偵測流量與雨量資料，直至獲取即時偵測資料後，將其分別利用「DB_Flow Service」與「DB_RainFall Service」自動匯入關聯式資料庫。當此三部分活動均執行完畢後，亦表示「Hydrol Service」所需要的輸入資料皆已產生，其才可接續進行演算，同時亦會利用「DB_WaveTide Service」將前一步驟「WaveTide Service」的演算結果匯入資料庫。依循類似的工作流程規則，「TwoFD Service」必須等待所有需要的輸入資料皆產生才能進行演算，同時前一步驟「Hydrol Service」的演算結果，則會利用「DB_Hydrol Service」進行匯入資料庫的動作。最後，當「TwoFD Service」的演算結果亦利用「DB_TwoFD Service」處理完畢後，便會導向一開始「Active」的條件判斷，再一次重新執行

整個工作流程，直至「Active」的條件判斷為否，則表示工作流程於停止狀態並結束離開。

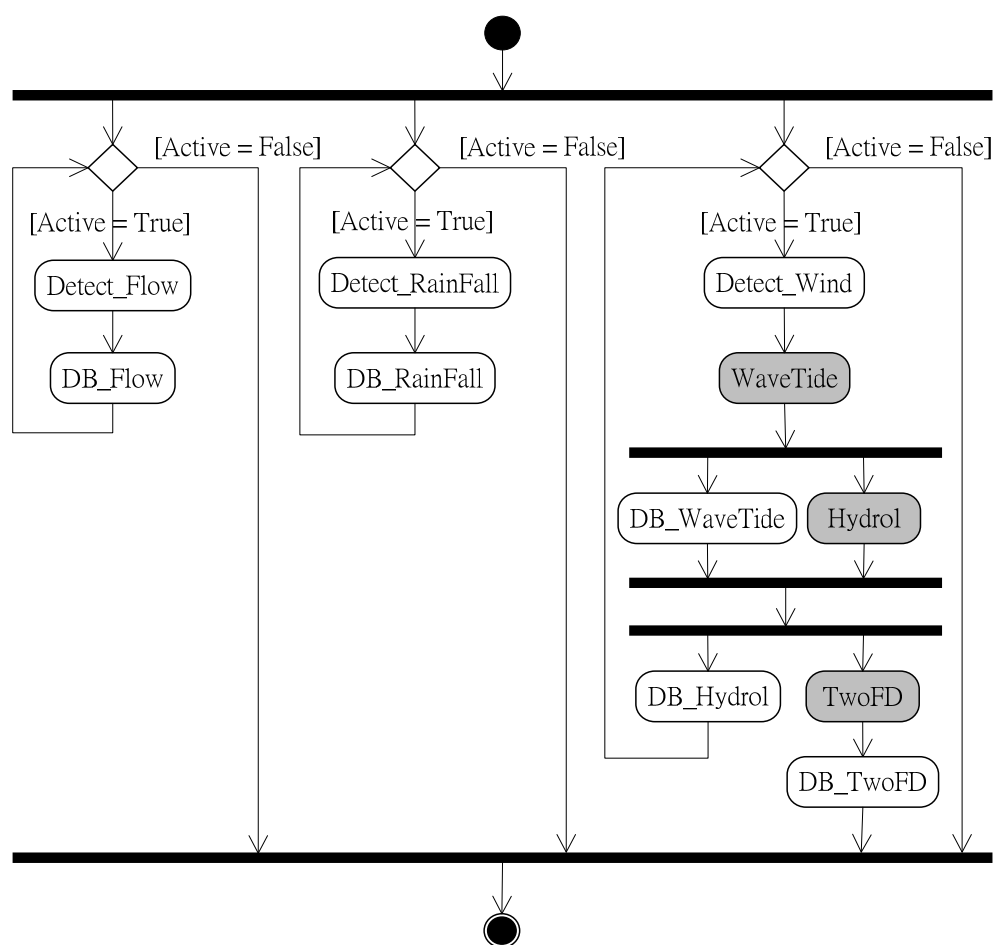


圖 4-5：工作流程控制圖 - 策略二

然而，由於流量與雨量偵測站於實際正常情況下，每十分鐘便會偵測到一筆即時資料，所以我們希望不停斷地持續偵測，並將即時偵測資料匯入資料庫，以便我們於使用者介面繪製有意義且精準的統計圖表。另外，因為「WaveTide Service」與「TwoFD Service」，均需要耗費以小時為單位的時間執行演算，因此我們希望能夠善用等待「TwoFD Service」執行演算的閒置時間，直接同時重新導向下一階段的工作流程，以提升工作流程整體的效能。基於上述兩項原因，我們將策略一進行改良，並提出策略二如圖 4-5 所示。一方面，我們將「Detect_Flow Service」與「Detect_RainFall Service」另行獨立出來，其會各自不停斷地持續偵

測，並分別利用「DB_Flow Service」與「DB_RainFall Service」將即時偵測資料匯入資料庫，直至工作流程於「Active」的條件判斷為停止狀態，才會因而結束離開。另一方面，當「DB_Hydrol Service」處理完畢後，其會直接導向一開始「Active」的條件判斷，重新開始下一次的工作流程，亦即此應用程式同時會有多個執行個體在運作。如此一來，不僅「WaveTide Service」不需要額外耗費一小時以上的閒置時間，且下一次執行「TwoFD Service」的前置等待時間亦會因而縮短。

除此之外，我們考量到「Hydrol Service」本身僅需要耗費以分鐘為單位的時間執行演算，相較於其它兩個 Hydraulics Services 落差懸殊，若能夠使其在不顧慮「WaveTide Service」是否結束當前演算的情況下，直接利用暫存變數內現有的演算結果做為輸入參數，並搭配平均每十分鐘偵測到的流量與雨量資料，進行多次「Hydrol Service」的執行，便能夠有效校正「Hydrol Service」的演算結果，以提升洪氾預警系統的可信度。依循相同的概念，為了讓 Hydraulics Services 均能盡可能不處於漫長的閒置等待狀態，因此我們依據策略二再次進行改良，並提出策略三如圖 4-6 所示。首先，由於在首次執行工作流程時，所有暫存變數內均未儲存任何輸出結果，因此整個工作流程可以藉由一個「First」的條件判斷大致區分為兩大部分。其一為首次執行工作流程的情況，該部分活動大致如同上述介紹的策略二，差異僅在於我們額外增添「First」的條件判斷，並於首次的「DB_TwoFD Service」處理完畢後，藉由指定 First 變數為 False，以得知所有暫存變數內均已儲存輸出結果，因而得以結束該部分活動，準備進入下一階段活動。其二為非首次執行工作流程的情況，該部分活動主要是以洪氾預警系統裡的三個 Hydraulics Services 為主軸劃分，各自將其獨立出來進行演算，且各網路服務會直接利用暫存變數內現有的演算結果做為輸入參數，並於執行完畢後將輸出參數更新至對應的暫存變數中，依循此方式持續執行，直至「Active」的條件判斷為停止狀態，才會因而結束離開。

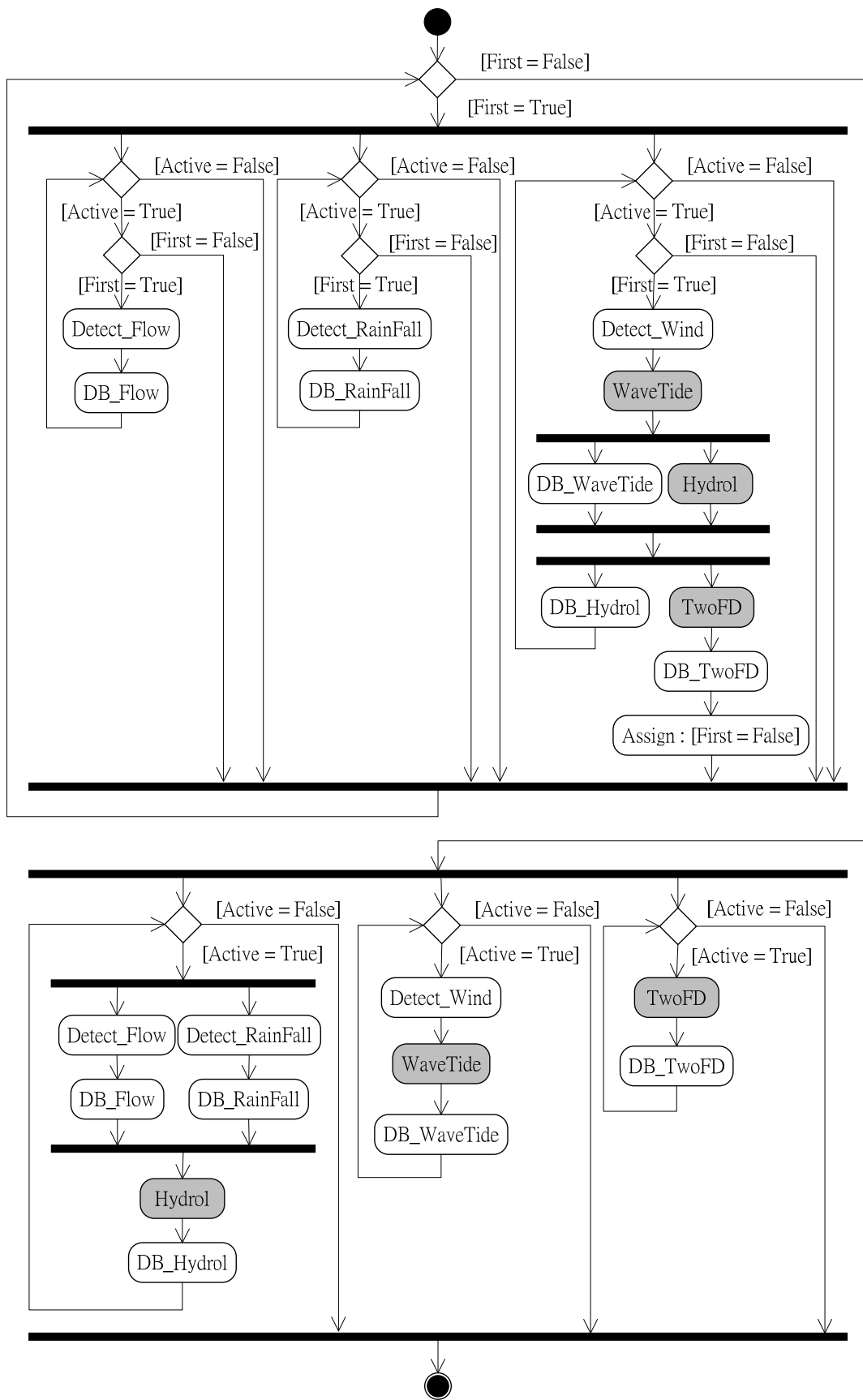


圖 4-6：工作流程控制圖 - 策略三

4.3 實驗

承接前述章節工作流程控制的介紹，以下我們將針對各工作流程策略進行實驗與分析討論，其中包含各水理服務的總執行次數比較，以及輸入資料的平均延遲時間比較，藉以利用數值化的實驗結果說明工作流程策略的合適性。至於洪氾預警系統的實作環境，在 Windows 伺服器方面，其 CPU 為四核心且核心時脈為 2.5GHz，而記憶體為 2GB，所採用的作業系統是 Windows Server 2008 版本。在 Linux 伺服器方面，其 CPU 為八核心且核心時脈為 2.4GHz，而記憶體為 16GB，所採用的作業系統是 Ubuntu10.10 版本。

由於考慮水理服務先後執行的優先順序與彼此間的相關性，我們控制其執行時間如表 4-1 所示。在表格最後一欄的水理服務執行時間之最大差距值，主要是利用水理服務中執行最長的時間與執行最短的時間相減而得。例如表格最後一列所示，「WaveTide Service」假設執行 1 分鐘，而「Hydrol Service」假設執行 31 分鐘，另外「TwoFD Service」假設執行 61 分鐘，則其執行時間的最大差距值便為 $(61 - 1) = 60$ 分鐘。我們將利用表 4-1 假設的執行時間做為實驗的調控變數，並統一於相同的系統環境下，針對前述章節所介紹的工作流程策略分別連續執行 5 個小時，以進行實驗結果的分析討論。

水理服務執行時間（分鐘）			水理服務執行時間 之最大差距值
WaveTide Service	Hydrol Service	TwoFD Service	
1	6	11	10 分鐘
1	11	21	20 分鐘
1	16	31	30 分鐘
1	21	41	40 分鐘
1	26	51	50 分鐘
1	31	61	60 分鐘

表 4-1：水理服務執行時間對照表

首先，在實驗中，我們將各水理服務於實驗期間內，其個別執行的次數進行加總，並依據該實驗結果繪製出圖 4-7。而由於一次執行便會有一次演算結果輸出，因此執行次數若越多，則表示輸出的演算結果則越多。在圖 4-7 中，其中橫軸座標為水理服務中執行時間的最大差距值，而其單位為分鐘，至於縱軸座標則為各水理服務的總執行次數。我們可以清楚發現三個工作流程策略，均隨著時間差距的增加，總執行次數有逐漸削減的趨勢，但是策略三的總執行次數始終大於策略一與策略二，足以說明不同工作流程策略於相同執行時間內，策略三裡各水理服務所輸出的演算結果相對較多，因此其為較佳的工作流程編排方式。

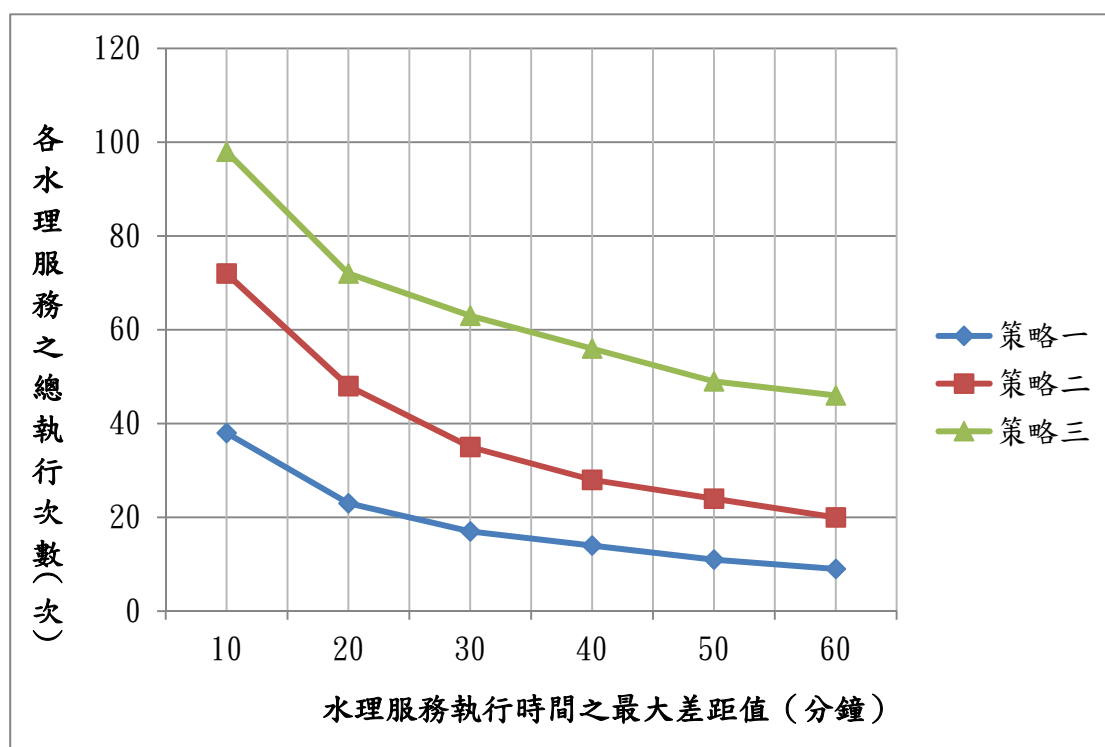


圖 4-7：輸出結果之總批次數

除此之外，在本實驗中，我們針對水理服務中最後執行演算的「TwoFD Service」進一步做討論。由於其需要以「WaveTide Service」與「Hydrol Service」的演算結果做為輸入參數，才能夠進行接續的演算過程，因此其接收到輸入資料的平均延遲時間將會影響其演算結果的可信度。其中所謂的平均延遲時間是指在實驗期間內，利用每一次「TwoFD Service」開始執行演算的時間與該次執行演

算所接收到輸入資料的時間相減，而後再依據這些數值取得的平均值。換而言之，若「TwoFD Service」接收到輸入資料的時間與其本身開始執行演算的時間越吻合，則表示「TwoFD Service」獲取越近似實際況狀的輸入資料進行演算，而演算結果的可信度亦會因而相對提高。因此我們針對「TwoFD Service」於實驗期間內，計算其輸入資料的平均延遲時間，並依據該實驗結果繪製出圖 4-8。其中橫軸座標為水理服務中執行時間的最大差距值，而縱軸則為「TwoFD Service」輸入資料的平均延遲時間，且兩者的單位均為分鐘。在圖 4-8 中，我們可以清楚發現三個工作流程策略，均隨著時間差距的增加，輸入資料的平均延遲時間有逐漸增加的趨勢，但是策略三輸入資料的平均延遲時間始終小於策略一與策略二，足以說明不同工作流程策略於相同執行時間內，策略三裡「TwoFD Service」演算結果的可信度相對較高，因此其為較佳的工作流程編排方式。

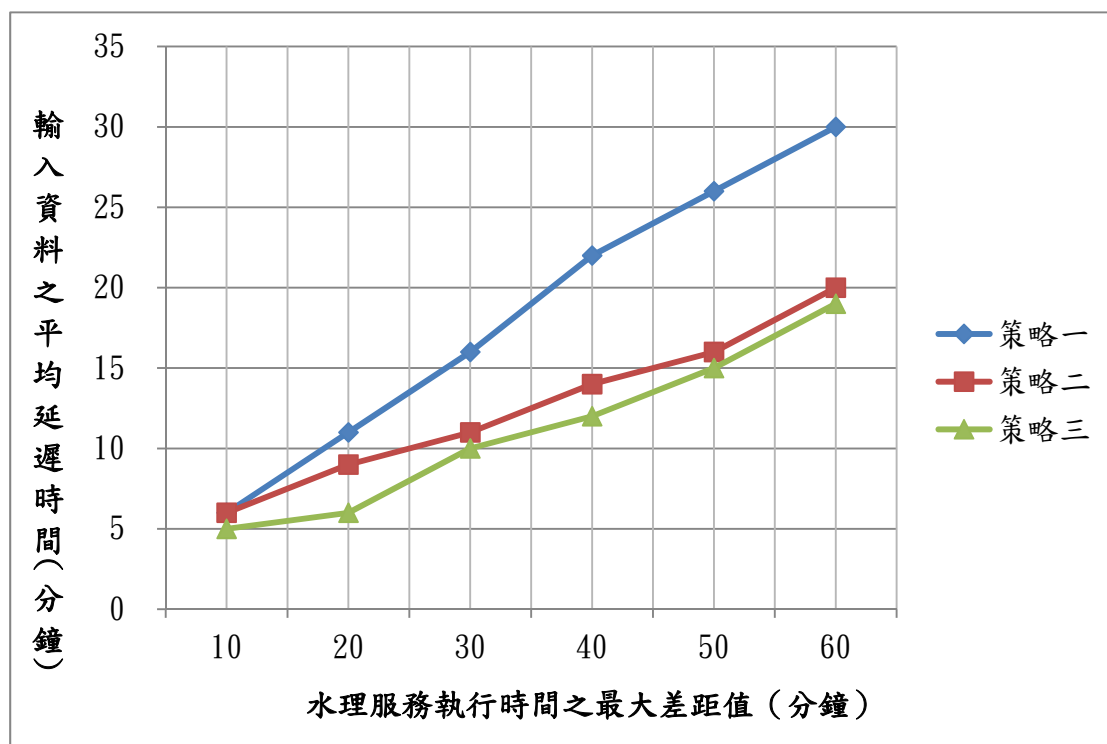


圖 4-8：輸入資料之平均時間差

第 5 章 資料庫服務

在此章，我們先介紹系統內部所設計的關聯式資料庫綱要，接續再依據其說明關聯式資料庫 SQL 指令的查詢使用，以及相對應的資料庫服務規劃。

5.1 關聯式資料庫綱要

針對前述章節 Hydraulics Services 與 Monitor Services 的執行結果，我們將額外進行資料的儲存與維護，其目的是為了提供豐富的使用者介面呈現，以及做為未來研究的查詢與分析之用。目前，關聯式資料庫（Relational Database）是工商業上最被廣為使用的資料維護軟體，主要原因是其可以搭配 SQL（Structured Query Language）進行簡易的查詢方式，且提供快速的查詢處理，以及方便的資料維護介面。因此，在本洪氾預警系統中，我們有另行規劃一個關聯式資料庫。以下我們簡介關聯式資料庫綱要（Database Schema）的設計，其中所有以底線標示的屬性為該表格的主鍵（Primary Key），表示該屬性值在表格中不會重複，以做為辨識之用。

我們維護的第一類型資料，為一些特定定點的地理資料，以便我們於網路地理資訊系統上明顯標示，並提供使用者點擊連結，進一步獲取該定點的相關資訊，如表 5-1 所示。首先，表格 Station_info 是依據洪氾預警系統的地理特性與演算需求，儲存雨量與流量偵測站的地理資料，而記錄內容包含其空間 ID、名稱以及座標。其次，表格 Section_info 是儲存「Hydrol Service」研究流域內，各河道橫斷面的地理資料，而記錄內容包含其空間 ID、左岸與右岸的座標以及高程，其中高程資料表示堤防頂端至河床底層的距離，我們將利用此做為判斷水位是否達河川溢堤的警戒標準。最後，表格 Place_info 則是儲存「TwoFD Service」演算範圍內，重要地名與其座標的對應，以便於網路地理資訊系統上清楚告知淹水預警的地名。

Station_info		偵測站基本地理資料
<u>SID</u>	varchar(10)	記錄偵測站的空間 ID
name	varchar(20)	記錄偵測站的名稱
x	decimal(9, 3)	記錄偵測站的二度分帶座標 x
y	decimal(10, 3)	記錄偵測站的二度分帶座標 y
Section_info		河道橫斷面基本地理資料
<u>SID</u>	varchar(10)	記錄河道橫斷面的空間 ID
Lx	decimal(9, 3)	記錄河道橫斷面左岸的二度分帶座標 x
Ly	decimal(10, 3)	記錄河道橫斷面左岸的二度分帶座標 y
L_height	float	記錄河道橫斷面左岸的高程
Rx	decimal(9, 3)	記錄河道橫斷面右岸的二度分帶座標 x
Ry	decimal(10, 3)	記錄河道橫斷面右岸的二度分帶座標 y
R_height	float	記錄河道橫斷面右岸的高程
Place_info		二維地點基本地理資料
<u>x</u>	decimal(9, 3)	記錄二維地點的二度分帶座標 x
<u>y</u>	decimal(10, 3)	記錄二維地點的二度分帶座標 y
name	varchar(40)	記錄二維地點的名稱

表 5-1：關聯式資料庫綱要－地理資料

我們維護的第二類型資料，為既有水理模組的演算結果，如表 5-2 所示。首先，「WaveTide Service」的演算結果會儲存至表格 OverTop 中，而記錄內容是未來一至三小時海岸各演算定點的座標、越波量數值以及預警狀況。其次，「Hydrol Service」的演算結果會分別整合至 Real_FQ、Pred_FQ 以及 RiverSection 三個表格中（FQ 為 flow quantity 欄位名稱的縮寫），其中表格 Real_FQ 是記錄未來一至三小時降雨逕流演算定點的模擬入流量，而預測入流量則是記錄在表格 Pred_FQ。至於表格 RiverSection 是記錄未來一至三小時研究流域，其各橫斷面的水位高度、流量數值以及左右溢岸流量數值。最後，「TwoFD Service」的演算結果則是儲存至表格 TwoFD 中，而記錄內容是未來一至三小時二維淹水各演算定點的座標以及水深高度。除此之外，由於既有水理模組演算輸出的結果會不斷累積至各表格中，但我們實際於網路地理資訊系統上僅需顯示最新的即時預警資訊。因此，對於每一次水理模組執行演算所輸出的結果，我們均會額外給予一個遞增的批次編號標記，其將以 batch 欄位名稱表示，以便直接依據最大的批次編號，獲取所需

呈現的即時預警資訊。同時，由於每一個批次皆會包含未來一至三小時的輸出結果，而同一時間與同一地理位置的資料又可能在連續的批次中被重複估算，所以表 5-2 中各表格的主鍵為批次、時間以及地理位置三者所構成。

OverTop		WaveTide Service 演算結果之海岸越波量
<u>batch</u>	int	記錄演算結果的批次
<u>time</u>	Datetime	記錄海岸越波量的時間
<u>x</u>	decimal(9, 3)	記錄演算定點的二度分帶座標 x
<u>y</u>	decimal(10, 3)	記錄演算定點的二度分帶座標 y
overtopping	float	記錄海岸越波量的數值
warning	int	記錄演算定點的預警狀況
Real_FQ		Hydrol Service 演算結果之模擬入流量
<u>batch</u>	int	記錄演算結果的批次
<u>time</u>	datetime	記錄模擬入流量的時間
<u>SID</u>	varchar(10)	記錄模擬入流量的空間 ID
flow_quantity	float	記錄模擬入流量的數值
Pred_FQ		Hydrol Service 演算結果之預測入流量
<u>batch</u>	int	記錄演算結果的批次
<u>time</u>	datetime	記錄預測入流量的時間
<u>SID</u>	varchar(10)	記錄預測入流量的空間 ID
flow_quantity	float	記錄預測入流量的數值
RiverSection		Hydrol Service 演算結果之左右溢岸流量
<u>batch</u>	int	記錄演算結果的批次
<u>time</u>	datetime	記錄左右溢岸的時間
<u>SID</u>	varchar(10)	記錄河道橫斷面的空間 ID
water_level	float	記錄河道橫斷面的水位
flow_quantity	float	記錄河道橫斷面的流量
L_overflow	float	記錄河道橫斷面的左溢岸流量
R_overflow	float	記錄河道橫斷面的右溢岸流量
TwoFD		TwoFD Service 演算結果之水深高度
<u>batch</u>	int	記錄演算結果的批次
<u>time</u>	datetime	記錄水深的時間
<u>x</u>	decimal(9, 3)	記錄演算定點的二度分帶座標 x
<u>y</u>	decimal(10, 3)	記錄演算定點的二度分帶座標 y
depth	float	記錄水深高度的數值

表 5-2：關聯式資料庫綱要 - Hydraulics Services 執行結果

我們維護的第三類型資料，為氣象單位偵測站量測的即時資料，以便我們於網路地理資訊系統上繪製統計圖，提供相關人員即時分析數值的變化，亦可做為檢討水理演算模式是否需要修正的依據。在表 5-3 中，表格 Measure_RF 是儲存來自「Detect_RainFall Service」各雨量偵測站量測的資料，而記錄內容包含該雨量偵測站的空間 ID、量測的時間以及雨量數值，至於表格 Measure_FQ 則是儲存來自「Detect_Flow Service」各流量偵測站量測的對應資料。

Measure_RF		雨量偵測站即時雨量
<u>time</u>	datetime	記錄即時雨量的時間
<u>SID</u>	varchar(10)	記錄雨量偵測站的空間 ID
rainfall	float	記錄即時雨量的數值
Measure_FQ		流量偵測站即時流量
<u>time</u>	datetime	記錄即時流量的時間
<u>SID</u>	varchar(10)	記錄流量偵測站的空間 ID
flow_quantity	float	記錄即時流量的數值

表 5-3：關聯式資料庫綱要 – Monitor Services 執行結果

5.2 資料庫服務介紹

承接前述第 4.1 章節所介紹，我們接續針對 Database Services 的第二部分進行說明。該部分是由於使用者介面所呈現的洪氾預警資訊，其均需要經由各網頁應用程式進行資料庫的資料查詢。然而，為了讓資料庫的資料查詢更加模組化，以方便日後系統進行維護與管理，因此我們依據使用者介面所呈現的資料性質進行統整分類，並依據分類結果將其封裝成各資料庫服務。至於該部分於系統中所規劃的資料庫服務與相關資料介面規範，我們將其統整於圖 5-1 中。在圖 5-1 中，各資料庫服務的輸入與輸出參數皆為字串型態，由於其輸出參數名稱均為 Result，所以我們僅列出輸入參數名稱並加以說明。注意到，Result 裡所存放的字串為 JSON（Javascript Object Notation）格式，我們會於後續一併詳細解釋。

History Service	
輸入參數	附註說明
Keyword	南州雨量站：541 來義雨量站：2401 力里雨量站：2402 新埤流量站：1760H004 林邊溪控制點：LinPien 力力溪控制點：Lili 河川溢堤預警：Hydrol_Section 二維淹水預警：TwoFD 沿海越波預警：WaveTide
StartTime	開始時間 (YYYY-MM-DD Hr:Mi:Se)
EndTime	結束時間 (YYYY-MM-DD Hr:Mi:Se)

Location Service	
輸入參數	附註說明
SID	南州雨量站：541 來義雨量站：2401 力里雨量站：2402 新埤流量站：1760H004 林邊溪控制點：LinPien 力力溪控制點：Lili 河川溢堤演算定點：Hydrol_Section 沿海越波演算定點：WaveTide

RealTime_ChartService	
輸入參數	附註說明
SID	南州雨量站：541 來義雨量站：2401 力里雨量站：2402 新埤流量站：1760H004 林邊溪控制點：LinPien 力力溪控制點：Lili

RealTime_Map Service	
輸入參數	附註說明
Keyword	河川溢堤預警：Hydrol_Section 二維淹水預警：TwoFD 沿海越波預警：WaveTide
WarningTime	未來一小時預警：1H 未來二小時預警：2H 未來三小時預警：3H
OutputType	地圖顯示：Diagram 表格顯示：Table

圖 5-1：網路服務資料介面－規範三

首先，「History Service」主要是負責過往歷史資料的查詢，其將對應到資料庫綱要中表 5-2 與表 5-3 的資料表格查詢，而該服務的輸入參數是關鍵字 (Keyword)、開始時間 (StartTime) 以及結束時間 (EndTime)。其次，「Location Service」主要是負責使用者介面中特定定點的地理位置查詢，包含雨量偵測站與流量偵測站，以及 Hydraulics Services 執行演算的演算定點，其將對應到資料庫綱要中表 5-1 的資料表格查詢，而該服務的輸入參數是空間 ID (SID)。除此之外，「RealTime_Chart Service」主要是負責使用者介面中繪製統計圖的資料查詢，包含雨量偵測站與流量偵測站的即時偵測資料，以及「Hydrol Service」執行演算所輸出的模擬與預測入流量，其將對應到資料庫綱要中表 5-2 (Real_FQ、Pred_FQ) 與表 5-3 的資料表格查詢，而該服務的輸入參數是空間 ID (SID)。最後，「RealTime_Map Service」主要是負責使用者介面中標記地圖預警圖示的資料查詢，其將對應到資料庫綱要中表 5-2 (OverTop、RiverSection、TwoFD) 的資料表格查詢，而該服務的輸入參數是關鍵字 (Keyword)、預警時間 (WarningTime)

以及資料顯示型態 (OutputType)。其中預警時間是由於 Hydraulics Services 執行演算的輸出結果皆包含未來一至三小時的預警資訊，至於資料顯示型態則是由於我們希望在使用者介面中，藉由地圖標記的預警資訊，皆能搭配與其相互對應的數據表做呈現，以供參照分析之用。

接續，我們針對上述所介紹的資料庫服務，將各別以其內部的 SQL 指令進行範例說明。首先，關於「History Service」的範例，假設輸入參數的關鍵字為“541”，開始時間為“2012-04-05 12:00:00”，結束時間為“2012-04-05 15:00:00”，則其所對應的 SQL 指令如表 5-4 的範例 1 所示。該服務會協助從 Measure_RF 表格中，查詢與關鍵字符合的偵測站空間 ID，且時間介於開始時間與結束時間的區間，並從符合條件的資料中，輸出其批次編號、時間與雨量數值。另一方面，假設輸入參數的關鍵字為“TwoFD”，開始時間為“2012-05-25 15:00:00”，結束時間為“2012-05-25 16:00:00”，則其所對應的 SQL 指令如表 5-4 的範例 2 所示。該服務會協助從 TwoFD 表格中，查詢時間介於開始時間與結束時間區間的資料，並輸出符合條件的資料其所有欄位(SQL 指令以符號 “*” 表示之)，而參照資料庫綱要表 5-2 裡的表格 TwoFD，其輸出欄位則包含批次編號、時間、二度分帶座標 x 與 y，以及水深高度的數值。因此，各資料庫服務內部的 SQL 指令會依據不同的輸入參數，決定不同的查詢表格與查詢條件，且該查詢表格則會間接影響 SQL 指令的輸出欄位。

除此之外，由於 SQL 指令所輸出的結果主要為 Data Table 型態，因此在資料庫服務內，我們會額外利用微軟公司內建的 Json.Net 套件，協助將 Data Table 型態的輸出結果轉換成 JSON 字串型態，以便於網頁應用程式進行介面呈現的處理。舉例來說，表 5-5 便為表 5-4 裡範例 2 所回傳的片段結果，而 JSON 用於描述資料的結構為「{name:value}」，以「{」與「}」區分 Data Table 型態裡的每一筆資料，並以「:」分隔「name」與「value」，其中「name」表示 Data Table 型態裡的欄位名稱，至於該欄位所對應的資料內容則以「value」表示。

範例 1	SELECT batch, time, rainfall FROM Measure_RF WHERE (SID = '541') AND (time BETWEEN '2012-04-05 12:00:00' AND '2012-04-05 15:00:00')
範例 2	SELECT * FROM TwoFD WHERE (time BETWEEN '2012-05-25 15:00:00' AND '2012-05-25 16:00:00')

表 5-4：關聯式資料庫 SQL 指令 – History Service 範例

```
"Table": [
  {
    "batch": 2,
    "time": "2012-05-25T15:37:41",
    "x": 200835.140,
    "y": 2480353.700,
    "depth": 106.1
  },
  . . .
]
```

表 5-5：資料庫服務之 JSON 字串格式範例

其次，關於「Location Service」的範例有三，首先，假設輸入參數的 SID 為“541”，則其所對應的 SQL 指令如表 5-6 的範例 1 所示。該服務會協助從 Sation_info 表格中，查詢偵測站空間 ID 符合 SID 的資料，並將其二度分帶座標 x 與 y 輸出。其次，假設輸入參數的 SID 為“Hydrol_Section”，則其所對應的 SQL 指令如表 5-6 的範例 2 所示。該服務會協助從 Section_info 表格中，查詢各河道橫斷面的空間 ID，以及左岸與右岸的二度分帶座標 x 與 y，並將上述欄位一併輸出。最後，假設輸入參數的 SID 為“WaveTide”，則其所對應的 SQL 指令如表 5-6 的範例 3 所示。該服務會協助從 OverTop 表格中，查詢符合最大批次編號且既為其中最大時間的資料，並將其二度分帶座標 x 與 y 輸出，該地理座標即為「WaveTide Service」執行演算的演算定點，而採用此做法的原因主要是該水理服務的演算定點可能會擴充或改變。

範例 1	SELECT x, y FROM Station_info WHERE (SID = '541')
範例 2	SELECT SID, Lx, Ly, Rx, Ry FROM Section_info
範例 3	SELECT x, y FROM OverTop WHERE (batch IN (SELECT MAX(batch) FROM OverTop)) AND (time IN (SELECT MAX(time) FROM OverTop))

表 5-6：關聯式資料庫 SQL 指令 – Location Service 範例

接下來，關於「RealTime_Chart Service」的範例，假設輸入參數的 SID 為“541”，則其所對應的 SQL 指令如表 5-7 的範例 1 所示。該服務會協助從 Measure_RF 表格中，查詢與 SID 符合的偵測站空間 ID，且批次編號介於最大值減去 431 與最大值的區間，並從符合條件的資料中，輸出其時間與雨量數值。其中數字 431 是由於使用者介面的統計圖需要呈現過往 3 天的偵測資料，而偵測站平均每十分鐘可輸出 1 個批次資料，即每一小時可輸出 6 個批次資料，至於 3 天便總共為 432 個批次資料（ $6 \times 24 \times 3 = 432$ ），因此繪製統計圖所需資料的批次編號會介於最大值減去 431 與最大值的區間。

另一方面，假設輸入參數的 SID 為“LinPien”，則其所對應的 SQL 指令如表 5-7 的範例 2 所示。該服務會協助從 First_Real_FQ 表格中，查詢與 SID 符合的空間 ID，且批次編號同樣介於最大值減去 431 與最大值的區間，並從符合條件的資料中，輸出其時間與流量數值，而此輸出資料即為「Hydrol Service」過往 3 天的模擬入流量。其中表格 First_Real_FQ 主要是儲存表格 Real_FQ 裡，每一個批次的第二筆資料，而我們以該筆資料做為該次演算當下時間的模擬入流量，因此若將其額外統整於 First_Real_FQ 表格中，可以方便系統快速地獲取過往 3 天的模擬入流量。除此之外，該服務會協助從 Pred_FQ 表格中，查詢與 SID 符合的空間 ID，且批次編號為最大值的資料，並將其時間與流量數值輸出，而此輸出資料即為「Hydrol Service」未來一至三小時的預測入流量。然而，如同上述

所介紹的範例 2，該資料庫服務於此種情況下，必須輸出兩個 SQL 指令的執行結果，因此該資料庫服務的輸出參數便會增添為兩個，其分別為 Result1 與 Result2，亦即 Result1 回傳「Hydrol Service」過往 3 天的模擬入流量，而 Result2 回傳「Hydrol Service」未來一至三小時的預測入流量。

範例 1	SELECT time, rainfall FROM Measure_RF WHERE (SID = '541') AND (batch BETWEEN ((SELECT MAX(batch) FROM Measure_RF) - 431) AND (SELECT MAX(batch) FROM Measure_RF))
範例 2	SELECT time, flow_quantity FROM First_Real_FQ WHERE (SID = 'LinPien') AND (batch BETWEEN ((SELECT MAX(batch) FROM First_Real_FQ) - 431) AND (SELECT MAX(batch) FROM First_Real_FQ)) SELECT time, flow_quantity FROM Pred_FQ WHERE (SID = 'LinPien') AND (batch IN (SELECT MAX(batch) FROM Pred_FQ))

表 5-7：關聯式資料庫 SQL 指令 – RealTime_Chart Service 範例

最後，關於「RealTime_Map Service」的範例，假設輸入參數的關鍵字為“WaveTide”，預警時間為“1H”，資料顯示型態為“Diagram”，則其所對應的 SQL 指令如表 5-8 的範例 1 所示。由於「WaveTide Service」輸出的每一個批次皆包含未來一至三小時的預警資訊，因此該服務會協助從 OverTop 表格中，查詢預警狀況為「是」，且符合最大批次編號，而時間為最大值（即未來第三小時）往前減去兩小時的資料，亦即未來第一小時的資料，並從符合條件的資料中，輸出其二度分帶座標 x 與 y。其中 SQL 指令所使用到的 dateadd(interval, number, date) 函數，其主要是用以回傳已添加指定時間間隔的日期。舉例來說，dateadd(hour, -2, MAX(time))便是將時間間隔單位設定為小時，時間間隔數值設定為-2，所以該函數所回傳的日期則為最大時間往前推估兩個小時。另一方面，假設輸入參數除了資料顯示型態改為“Table”之外，其餘輸入參數皆與範例 1 相同，則其所對應的 SQL 指令如表 5-8 的範例 2 所示，而相較於範例 1 的 SQL 指令，差亦僅在於輸出資料還需要定點座標所對應的海岸越波量。因此，該資料庫服務的輸入參數

OutputType 會直接影響到查詢表格的輸出欄位，假設 OutputType 為 “Diagram” ，則輸出欄位僅會包含預警資訊的地理座標，另外假設 OutputType 為 “Table” ，則輸出欄位則會一併包含該預警地理座標的相關數值結果。

範例 1	SELECT x, y FROM OverTop WHERE (warning = '1') AND (batch IN (SELECT MAX(batch) FROM OverTop)) AND (time IN (SELECT dateadd(hour, -2, MAX(time)) FROM OverTop))
範例 2	SELECT x, y, overtopping FROM OverTop WHERE (warning = '1') AND (batch IN (SELECT MAX(batch) FROM OverTop)) AND (time IN (SELECT dateadd(hour, -2, MAX(time)) FROM OverTop))

表 5-8：關聯式資料庫 SOL 指令 – RealTime_Map Service 範例

第 6 章系統預警呈現

在此章，我們介紹洪氾預警系統的使用者介面，並說明利用 Google Maps API 與 Google Chart API 所呈現的預警資訊。至於本系統網站的實作工具，在資料庫應用方面，我們使用 SQL Server 2008 軟體，而在網站程式撰寫方面，我們使用 ASP.Net 以及 JavaScript 程式語言，最後在網站伺服器管理方面，我們則是使用 IIS 伺服器管理員。

6.1 系統網站架構

本論文建置的系統網站是以台灣南部林邊溪流域為研究案例，進行洪氾預警的介面呈現，而系統網站的首頁如圖 6-1 所示。在視窗的左上角，我們將系統的功能選單規劃為四大類型，分別為「預警資訊」、「即時資料」、「歷史資料」與「關於」，且其皆對應到圖 3-1 系統架構裡 Application Layer 的「Warning Manager」。首先，功能選單中第一類型「預警資訊」，主要是依據 Hydraulics Services 執行演算所輸出的結果進行預警呈現，其中涵蓋「Hydrol Service」的河川溢堤預警、「TwoFD Service」的二維淹水預警，以及「WaveTide Service」的沿海越波預警。我們會先利用資料庫服務中的「Location Service」，於地圖上標記演算定點的地理位置，隨後再利用資料庫服務中的「RealTime_Map Service」，於地圖上標記各水理服務的預警圖示。除此之外，於該類型功能選單中，我們會藉由即時資訊視窗的網站頁面，顯示系統中特定定點的地理位置與相關資訊的連結，而特定定點包含雨量偵測站、流量偵測站，以及「Hydrol Service」執行降雨逕流演算模式的兩個上游控制點。其次，功能選單中第二類型「即時資料」，主要是依據系統中特定定點的即時資料繪製統計圖，其中流量資料是以折線圖繪製，而雨量資料則是以長條圖繪製，至於該網站頁面所使用的資料庫服務則是「RealTime_Chart Service」。接續，功能選單中第三類型「歷史資料」，主要是提供使用者查詢 Hydraulics Services 與 Monitor Services 過往某段時間的歷史資料，以供未來模擬

演算的研究分析之用，而該網站頁面所使用的資料庫服務則是「History Service」。

最後，功能選單中第四類型「關於」，主要是說明系統介面的使用方式與注意事項。同時，在視窗的右上角，我們支援系統管理者利用帳號與密碼進行登入，而該部分是對應到圖 3-1 系統架構裡 Application Layer 的「User Manager」，以便系統管理者直接透過網站頁面，進行過往歷史資料的管理。上述相關預警資訊的網站頁面呈現，我們將於後續章節做更詳細的介紹。

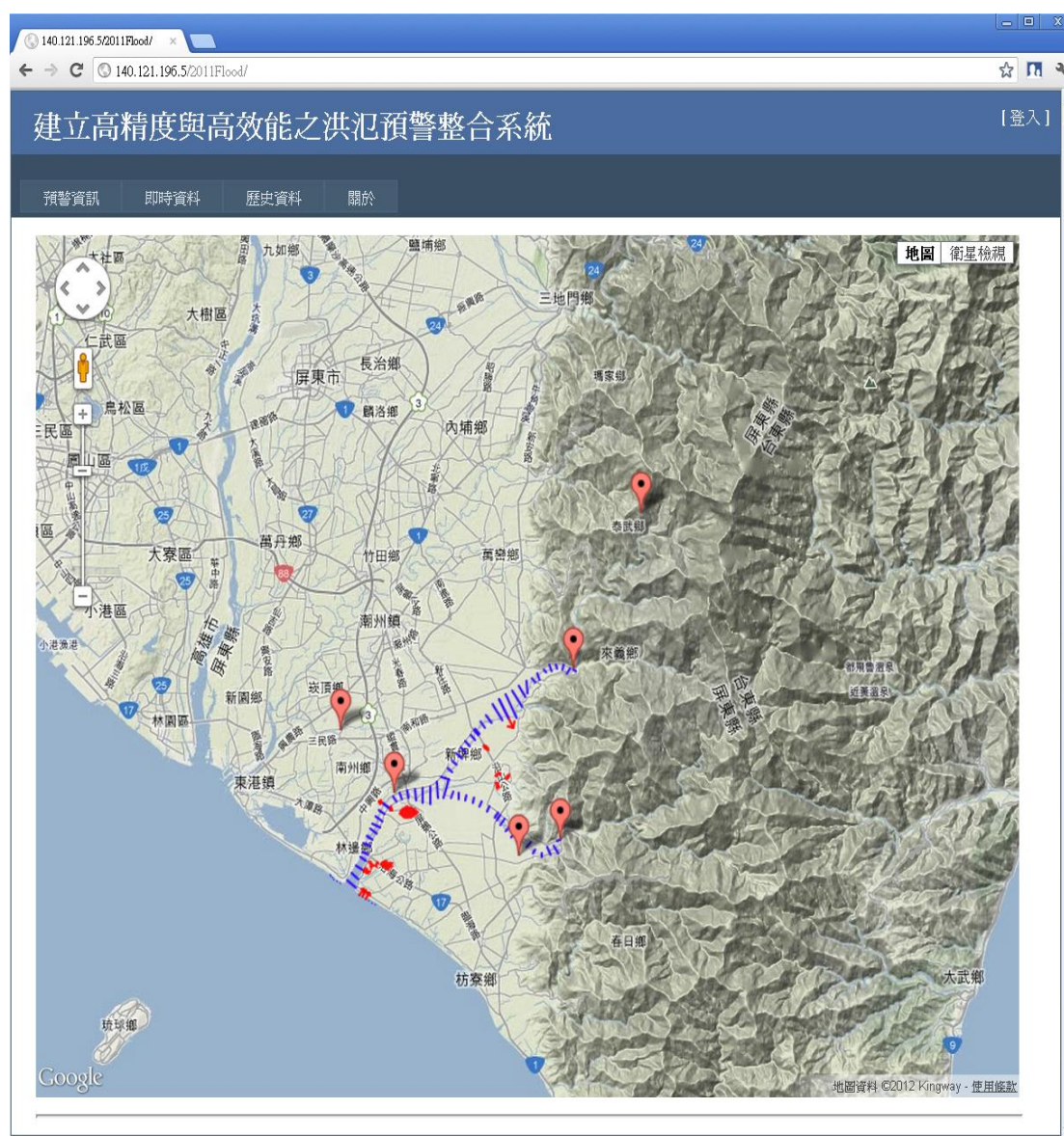


圖 6-1：洪氾預警系統網站首頁

6.2 河川溢堤預警

此網站頁面主要是依據「Hydrol Service」中，其河道斷面演算模式的執行結果做預警呈現。如圖 6-2 所示，我們會先利用資料庫服務中的「Location Service」進行資料查詢，以便獲取林邊溪流域實際量測的各橫斷面地理資料，並依據此資料於地圖上將各橫斷面的左岸與右岸座標相連接成紫色線段。接續，我們會再利用資料庫服務中的「RealTime_Map Service」進行資料查詢，而若該斷面輸出的左溢岸流量值已達預警標準，則會於該斷面的左岸向外延伸一個紅色箭頭，用以顯示此地點可能發生河川溢堤的情形。同理，若河川溢堤的情形可能發生在該斷面的右岸，則亦是以相同的方式表示。至於未來一至三小時的河川溢堤預警呈現，目前則是依據預警優先權並搭配箭頭顏色的深淺做區分，亦即未來第一小時預警以深紅色表示，未來第二小時預警以桃紅色表示，未來第三小時預警以粉紅色表示，因此箭頭顏色越深則可說明預警越迫切。



圖 6-2：河川溢堤預警圖

6.3 二維淹水預警

此網站頁面主要是依據「TwoFD Service」中，其二維淹水演算模式的執行結果做預警呈現。如圖 6-3 所示，該演算模式會針對林邊溪流域附近的鄉鎮進行空間離散的處理方式，推估出二維淹水的各個演算定點，而我們再利用資料庫服務中的「RealTime_Map Service」進行資料查詢，以便將這些演算定點輸出的水深高度進行預警判斷，若該演算定點已達預警標準，則利用其二度分帶座標於地圖上以紅色小圓點標示。因此，我們可以在圖 6-3 中，明顯發現當這些紅色小圓點越密集，則表示該區域淹水預警程度越強烈，且這些紅色小圓點普遍會群聚落在一些地勢較為低窪的區域。至於未來一至三小時的二維淹水預警呈現，目前則是劃分為三個網站頁面，分別為未來第一小時預警頁面、未來第二小時預警頁面以及未來第三小時預警頁面。

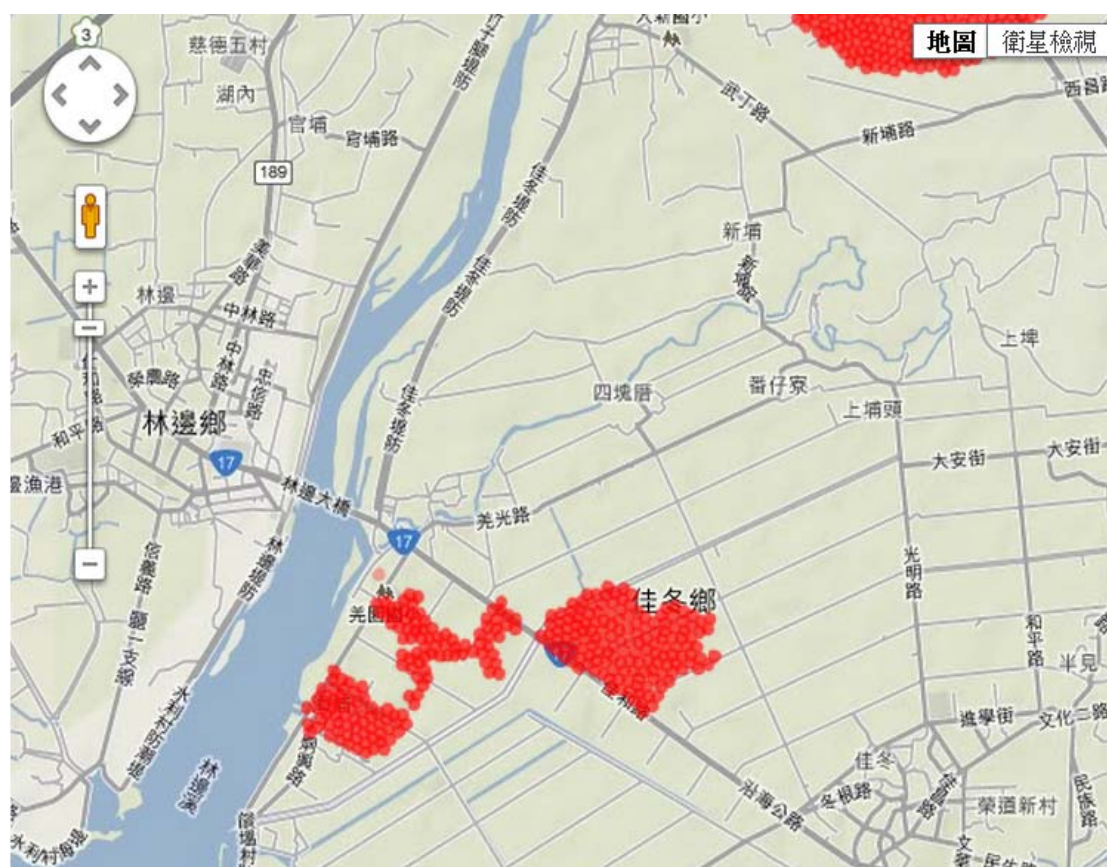


圖 6-3：二維淹水預警圖

6.4 沿海越波預警

此網站頁面主要是依據「WaveTide Service」中，其海岸越波演算模式的執行結果做預警呈現。如圖 6-4 所示，該演算模式會針對林邊溪流域沿海進行網格計算，推估出數個越波流量的演算定點，而我們會先利用資料庫服務中的「Location Service」進行資料查詢，以便依據這些演算定點的二度分帶座標，於地圖上以藍色圓點標示。接續，我們再利用資料庫服務中的「RealTime_Map Service」進行資料查詢，而若該演算定點輸出的越波流量值已達預警標準，則會於該定點以一個紅色箭頭指向內陸，以顯示此地點可能發生海浪越波的情形。至於未來一至三小時的海岸越波預警呈現，則是仿照二維淹水預警呈現的方式，分別劃分為三個預警頁面做呈現。



圖 6-4：沿海越波預警圖

6.5 流量資訊顯示

此網站頁面主要目的是讓使用者便於觀察特定定點的流量資訊。如圖 6-5 所示，我們會先利用資料庫服務中的「Location Service」進行資料查詢，並針對特定定點的二度分帶座標，於地圖上做特殊的地標標記，標記的定點除了「Detect_Flow Service」的新碑流量偵測站之外，還有在「Hydrol Service」中，其執行降雨逕流演算模式的兩個上游控制點，分別位於主幹流域林邊溪與分支流域力力溪。而當使用者點擊該地標標記時，便會根據此地標標記顯示一個相關資訊的對話方塊，導引使用者進一步連結預覽統計圖，以及觀看詳細的數據資料表。若使用者直接選擇預覽模式，則將於地圖下方動態出現其對應的流量統計圖。反之，若使用者選擇觀看詳細的數據資料表，則將會另行連結至一個有統計圖與數據資料表相互對照的網站頁面。

至於流量統計圖的設計，以降雨逕流演算模式的執行結果為例，我們是利用資料庫服務中的「RealTime_Chart Service」進行資料查詢，並依據查詢結果搭配折線圖進行繪製，其中以前段藍線顯示該控制點過去三天的模擬入流量，並以後段紅線顯示該控制點未來三小時的預測入流量。同時，我們會以一條橘色橫線顯示預警標準，若流量數值超過該線段，則表示使用者需要多加警戒。除此之外，若使用者點擊新碑流量站的地標標記，我們則是顯示該偵測站於過去三天內，實際量測的即時流量資料，以提供對照參考之用。

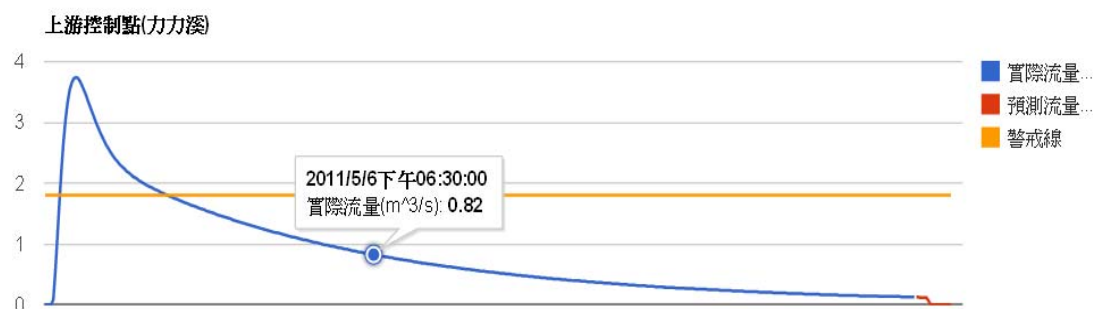


圖 6-5：流量資訊顯示圖

6.6 雨量資訊顯示

此網站頁面主要目的是讓使用者便於觀察特定定點的雨量資訊。如同上述流量資訊顯示的規劃方式與其使用的資料庫服務，在圖 6-6 中，我們會先依據林邊鄉雨量偵測站的二度分帶座標，於地圖上做特殊的地標標記，標記的定點包含「Detect_RainFall Service」的南州雨量偵測站、來義雨量偵測站，以及力里雨量偵測站。不過，相較於圖 6-5，我們則是利用長條圖來進行繪製，而顯示的資料為該偵測站於過去三天內，實際量測的即時雨量資料。

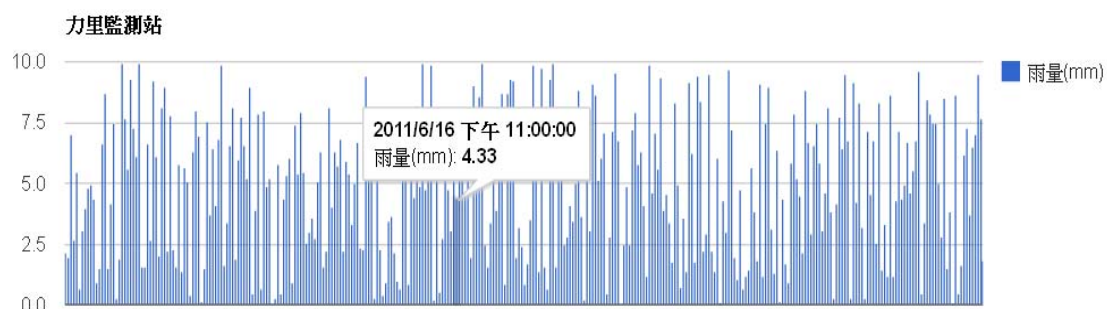


圖 6-6：雨量資訊顯示圖

第 7 章結論與未來方向

在本論文中，我們基於服務導向架構（Service-Oriented Architecture，簡稱 SOA）的系統規劃方式，建構一個洪氾預警系統的資料交換與作業整合平台，並藉由 .Net Framework 內所支援的 WCF（Windows Communication Foundation）與 WF（Windows Workflow Foundation）技術的協助，讓既有水理模組順利地傳遞交換所需的資料，並協調控管既有水理模組的執行作業，使其不僅能夠進行跨平台的通訊合作，亦能夠達成資源共享與洪氾預警的功能。

同時，我們善用網路地理資訊系統（Web Geographic Information System，簡稱 Web GIS）的概念，以及 Google 的雲端基礎建設，利用 Google Maps API 與 Google Chart API 的技術，將既有水理模組所回傳的演算結果與預警資訊，搭配地圖與圖表的方式適當地呈現於網頁上，提供民眾與相關單位可以直接透過網路瀏覽器進行查詢瀏覽，以達到系統有效預警的目的。

關於本論文未來的研究方向，由於目前各水理模組封裝而成的網路服務，其輸入與輸出的資料介面規範，尚未符合地理資料共通的標準規範，因此我們期望能夠為此研究合適的改良方法。同時，我們期望能夠提升工作流程控制的容錯能力，將多種例外狀況的錯誤處理一併納入考量。除此之外，針對二維淹水預警的介面呈現，由於其有機會需要於地圖上繪製上萬個預警標示，導致網路瀏覽器的介面呈現有所延遲，因此我們期望能夠為此找尋合適的解決方法。最後，我們希望系統網站能夠提供使用者即時的洪氾預警資訊，亦即各水理服務若有新的演算結果輸出，則系統網站能夠協助使用者自動刷新網站頁面，且該刷新動作必須為必要網頁更新，以避免造成網站系統龐大的負擔。

參考文獻

- [AMB03] W. Al-Sabhan, M. Mulligan, and G.A. Blackburn, “A real-time hydrological model for flood prediction using GIS and the WWW”, Computers, Environment and Urban Systems, vol. 27, no. 1, pp. 9-32, 2003.
- [Bu+10] T. Bulatewicz, X. Yang, J. M. Peterson, S. Staggenborg, S. M. Welch, and D. R. Steward, “Accessible integration of agriculture, groundwater, and economic models using the Open Modeling Interface (OpenMI) : methodology and initial results”, Hydrology and Earth System Sciences, vol. 14, pp. 521-534, 2010.
- [BS11] Bernhard P. Becker, and Holger Schuttrumpf, “An OpenMI module for the groundwater flow simulation programme Feflow”, Journal of Hydroinformatics, vol. 13, no. 1, pp. 1-12, 2011.
- [GDG10] Carlos Granell, Laura Diaz, and Michael Gould, “Service-oriented applications for environmental models: Reusable geospatial services”, Environmental Modelling & Software, vol. 25, no. 2, pp. 182-198, 2010.
- [FEWS] Delft Hydraulics, “Delft-FEWS, an open shell flood forecasting system”, <http://www.wldelft.nl/soft/fews/int/index.html>
- [Li+06] Xiang-Yang Li, Kwok-wing Chau, Chuntian Cheng, and Y. S. Li, “A Web-based flood forecasting system for Shuangpai region”, Advances in Engineering Software, vol. 37, no. 3, pp. 146-158, 2006.
- [KML] Keyhole Markup Language, <http://kml-samples.googlecode.com/svn/>

trunk/interactive/index.html

- [OpenGIS] OpenGIS Standards, <http://www.opengeospatial.org/standards>
- [OpenMI] Open Modeling Interface, <http://www.openmi.org>
- [WCF] Windows Communication Foundation, <http://msdn.microsoft.com/zh-tw/library/dd456779>
- [WF] Windows Workflow Foundation, [http://msdn.microsoft.com/zh-tw/library/bb628617\(v=vs.90\).aspx](http://msdn.microsoft.com/zh-tw/library/bb628617(v=vs.90).aspx)
- [WSBPEL] Web Service Business Process Execution Language, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
- [Zheng +10] Zongsheng Zheng , Dongmei Huang, Jianxin Zhang, Shengqi He, and Zhiguo Liu, “A SOA-based Decision Support Geographic Information System for Storm Disaster Assessment”, International Conference on Geoinformatics, 2010.
- [Zhu+11] Zhengwei Zhu, Shanqin Wang, Jiaying Chen, Hongyu Zhang, and Wenfeng Dun, “Design and Implementation of the Geospatial Early Warning System of the Insect Pest Based on SOA”, International Conference on Geoinformatics, 2011.
- [何+09] 何丁武, 馬尚彬, 黃富馴, 王躍強, 黃貴麟, “e – 河川計畫的應用推廣與服務”, 第18屆水利工程研討會論文集, 2009。
- [林+09] 林旭信, “都市積淹水即時預報系統研發 – 子計畫：分散式都市淹水模擬與支援決策系統框架之研發(三)”, 國科會研究成果報告NSC

96-2625-M-033-001-, 2009。

[張+10] 張書源, “Visual C# 2010 與 UML 開發實戰”, 悅知文化出版公司, 2010。

[黃+09] 黃博炫, 謝龍生, 朱子偉, 傅金城, “洪水災害早期預警系統建置之研究”, 第 18 屆水利工程研討會論文集, 2009。

附錄

A. Windows Communication Foundation

Windows Communication Foundation (簡稱 WCF) [WCF, 張+10] 內建於 .NET Framework 中，用來建置 SOA 應用程式的統一程式設計模型。由於 SOA 的發展與應用越來越廣，服務元件的建立方式亦趨於多元化，為了滿足 SOA 發展的需求，同時簡化維護與存取服務元件所需的程式碼，WCF 將多種通訊協定的不同溝通模式進行技術整合，其中包含 .Net Remoting、Web Service、WSE、Microsoft Message Queuing (簡稱 MSMQ) 等。因此，透過 WCF 所提供的支援，使用者可以使用簡單的方式建立分享給其它應用程式的服務，或是存取已存在的服務。

在 WCF 的架構下，當使用者需要建立服務或是存取服務時，透過提供關於服務的存取位置 (Address)、繫結方式 (Binding) 以及合約內容 (Contract) 等三種資訊，WCF 便能協助發佈或是繫結服務。因此，使用者不需額外針對發佈服務元件或是繫結服務元件而開發程式碼，僅需藉由設定伺服器端與用戶端的服務相關資訊，則 WCF 便會自動進行溝通，其參考架構如圖 A-1 所示，至於其範例則參考附錄 C。

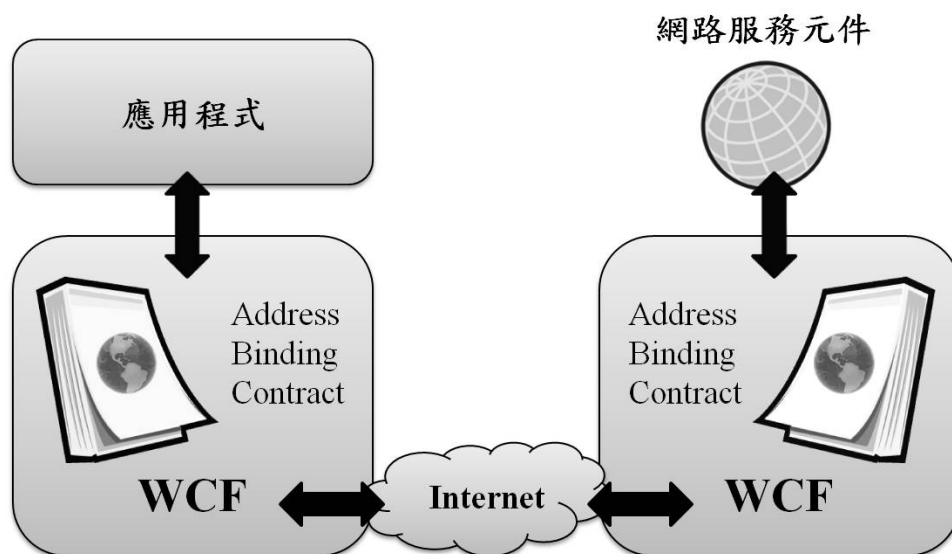


圖 A-1：使用 WCF 架構

B. Windows Workflow Foundation

Windows Workflow Foundation (簡稱 WF) [WF, 張+10] 內建於 .NET Framework 中，用來執行環境的應用程式流程服務。透過 WF 所提供的應用程式流程引擎，再搭配 Visual Studio 2010 開發工具所提供的流程設計工具，便可使用圖形化的方式設計應用程式所需要的流程。而 WF 主要包含基礎活動程式庫與架構，以及執行階段流程引擎兩個部分，以下分別說明其內容。

1. 基礎活動程式庫與架構

在 WF 工作流程中的每一個物件，稱為一個活動 (Activity)。 .NET Framework 提供許多基礎活動的定義，例如條件判斷 (ifElse)、迴圈 (while) 等，這些基礎活動均可直接使用在 WF 工作流程中。除此之外， .NET Framework 亦提供「活動應用程式架構 (Activity Framework)」，所以，若有需要的話，使用者亦可自行開發程式中所需要的 WF 工作流程活動。最後，WF 工作流程常用的基礎活動整理於表 B-1 中。

活動	描述
 DoWhile	執行所包含的活動一次，並繼續執行直到條件為 True 為止。
 ForEach<T>	針對集合中的每個活動項目循序執行。
 If	如果條件為 True 時，執行所包含的活動，如果條件為 False 時，則執行包含在 Else 中的活動。
 Parallel	平行執行所包含的活動。
 ParallelForEach<T>	平行執行集合中的活動項目。
 Pick	提供以事件為基礎的流程控制模型。
 PickBranch	表示 Pick 活動中的部分執行路徑。
 Sequence	循序執行所包含的活動。
 Switch<T>	根據運算式的值，選取不同的活動執行。
 While	當條件運算式的值為 True 時，執行所包含的活動。
 Flowchart	使用一般的流程圖執行所包含的活動。
 FlowDecision	建立兩種結果的條件判斷流程節點。
 FlowSwitch<T>	建立一個條件運算式對應一組流程的條件判斷節點。
 Receive	接收訊息的節點。
 Send	傳送訊息的節點。

	ReceiveAndSendReply	接收訊息並傳送訊息的節點。
	SendAndReceiveReply	傳送訊息並接收回應的節點。
	Persist	儲存流程的狀態。
	TerminateWorkflow	中止工作流程。
	Assign	指派值到工作流程的變數中。
	Delay	延遲工作流程一段時間。
	InvokeMethod	觸發工作流程中的方法。
	WriteLine	將文字寫入至指定的 TextWriter 物件。
	AddToCollection<T>	將指定項目加入至集合。
	ClearCollection<T>	清除指定集合中的所有項目。
	ExistsInCollection<T>	當指定項目存在於集合中時，則回傳 True。
	RemoveFromCollection<T>	將指定項目從特定集合中移除，如果移除成功則回傳 True。
	Rethrow	重新擲出 try...catch 活動中最後擲出的例外狀況。
	Throw	擲出例外狀況。
	TryCatch	在工作流程中實作例外狀況處理的活動。

表 B-1：WF 常用基礎活動

2. 執行階段流程引擎

在建立好 WF 工作流程定義後，還必須透過「工作流程執行階段引擎 (Workflow Runtime Engine)」建立「工作流程實體 (Workflow Instance)」，才能夠開始執行。而一個應用程式內可以支援多個工作流程執行個體同時執行，且 WF 工作流程可以在任何 .NET 應用程式內部執行，其中包含主控台應用程式、表單架構應用程式、ASP.NET 網站與網路服務等。除此之外，由於工作流程執行階段引擎，以及其所建立的工作流程執行個體，都是建立在應用程式執行環境當中，因此，工作流程亦可以與主應用程式 (Host Application) 溝通產生互動。如圖 B-1 所示，便是於主應用程式中載入 WF 定義，並建立工作流程執行個體的執行環境。

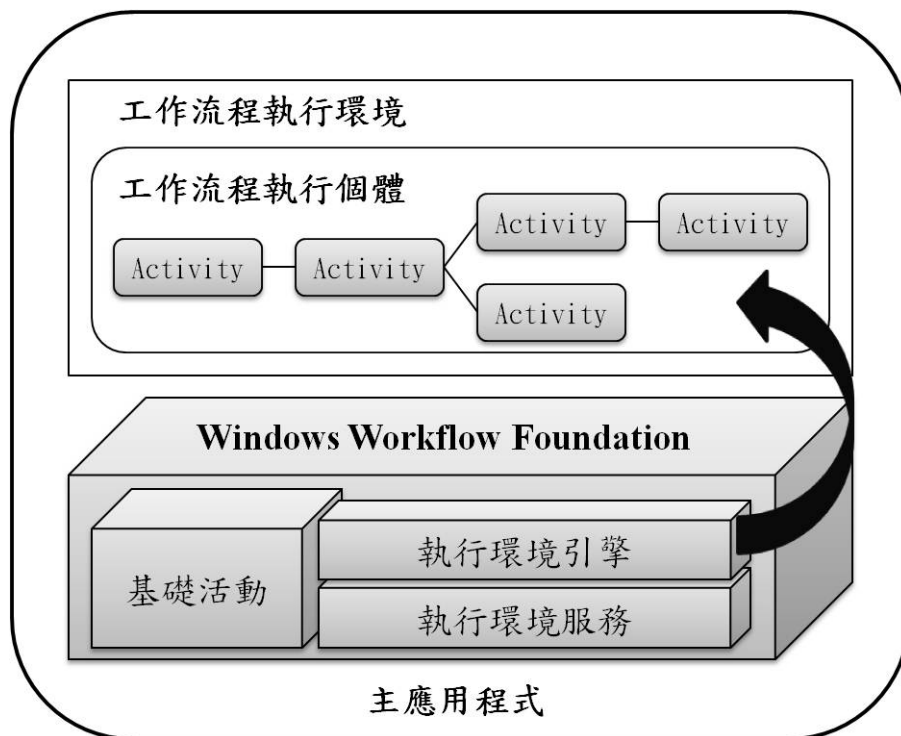



圖 B-1：載入 WF 之應用程式執行環境

C. 工作流程控制之WF實作範例

依據前述第 4.1 與 4.2 章節所介紹，我們使用 .Net Framework 內所支援的 WCF 與 WF 技術，同時搭配 Microsoft Visual Studio 2010 軟體的協助，成功地進行各網路服務的封裝與共享，以及協調各網路服務的執行運作流程。以下我們依據前述章節圖 4-4 策略一的工作流程控制為實作範例進行說明，而該軟體實際控管此工作流程的圖形化介面如圖 C-1 所示。

首先，我們利用該軟體協助將前述第 4.1 章節所建立各網路服務，皆透過「服務參考」的方式將其引用進來。假設以「WaveTide Service」為範例說明，其對應的服務參考介面如圖 C-2 所示。在圖 C-2 中，該水理服務的實作是以 Wcf_WaveTide 為命名方式，而我們僅需要藉由在黑色粗體方框中，將該水理服務實際的存取位置 (Address) 輸入並點選「移至」，便能夠順利地將其引用進來。同時，圖 C-2 中「IWcf_WaveTide」的相關內容可以對應到圖 C-3，以便我們可以清楚了解該水理服務的繫結方式 (Binding)。舉例來說，CloseTimeOut 與 OpenTimeOut 表示取得或設定針對連線所提供的時間間隔、MaxBufferSize 表示取得或設定自通道接收訊息的緩衝區大小上限、MaxReceivedMessageSize 表示取得或設定可在使用此繫結所設定之通道上接收的訊息大小上限。至於圖 C-2 中的「WaveTide_DoWork」，則可以協助我們得知該水理服務的合約內容 (Contract)，亦即其所提供的功能函式以及輸入輸出參數。接續，上述引用進來的網路服務在工作運作流程中，便可視為可以直接進行控管的活動，而依據事先設計好的工作流程圖，並搭配該軟體內提供的基礎活動，便能夠以圖形化的方式將工作流程實作出來。最後，我們利用圖 C-4 的應用程式執行上述圖 C-1 所建立好的工作流程，由於該工作流程的起始點存在一個受外部應用程式影響的「Active」條件判斷，因此我們新增一個名為 Active 的引數  Dictionary<string, object> 型別集合物件中，藉以設定該引數為“True”讓工作流程能夠於起始狀態，並利用 Run() 方法

命令該工作流程開始執行。

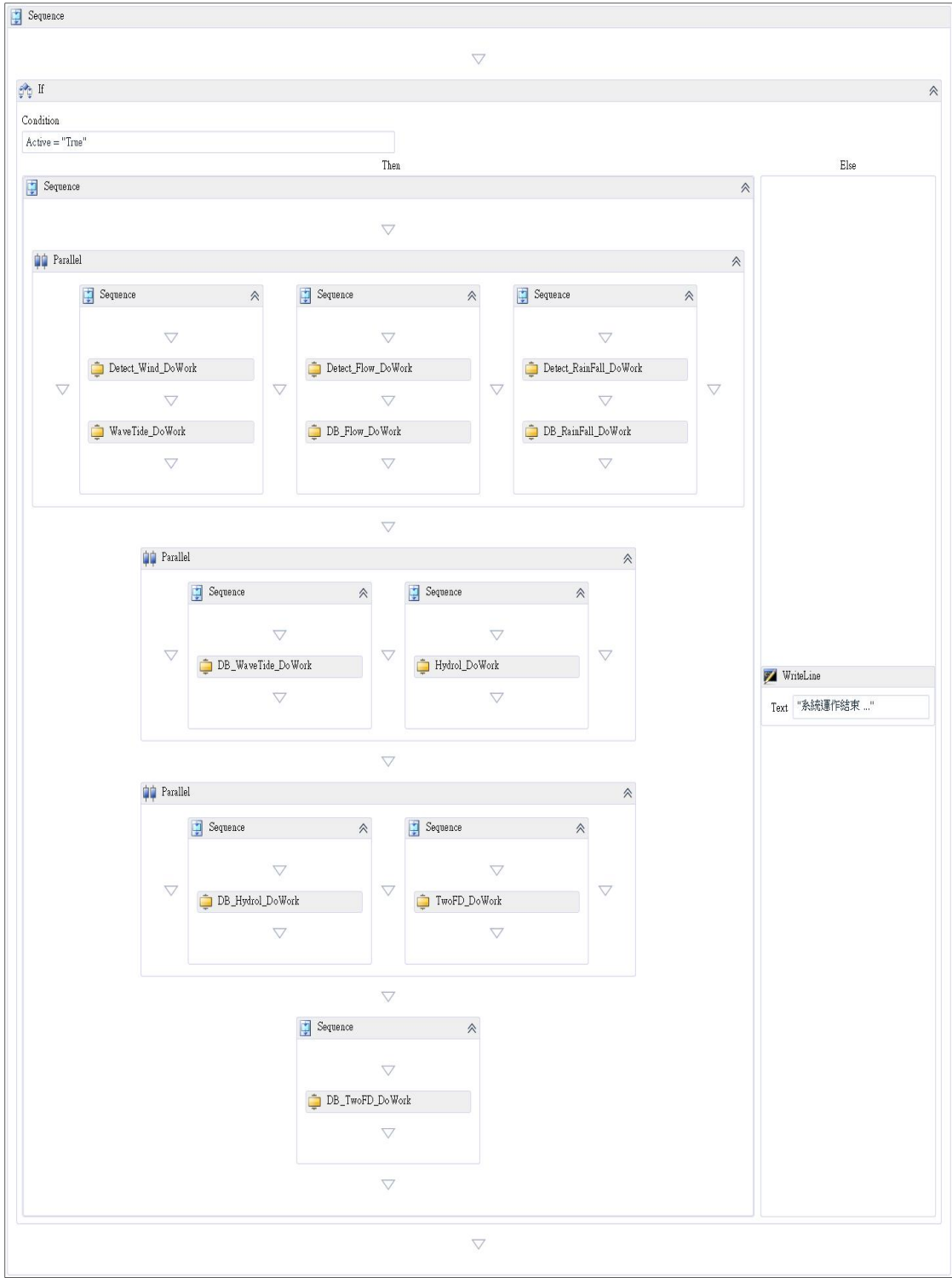


圖 C-1：工作流程控制之 WF 實作範例

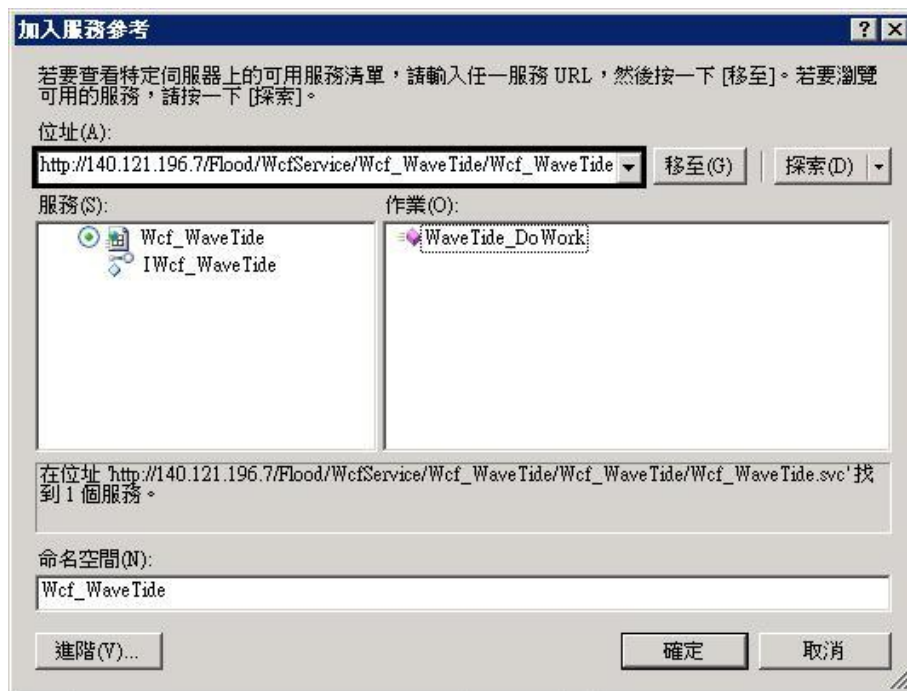


圖 C-2：服務參考實作範例

```
<binding name="BasicHttpBinding_IWcf_WaveTide" closeTimeout="00:01:00"
  openTimeout="00:01:00" receiveTimeout="00:10:00" sendTimeout="24.00:00:00"
  allowCookies="false" bypassProxyOnLocal="false" hostNameComparisonMode="StrongWildcard"
  maxBufferSize="65536" maxBufferPoolSize="524288" maxReceivedMessageSize="65536"
  messageEncoding="Text" textEncoding="utf-8" transferMode="Buffered"
  useDefaultWebProxy="true">
  <readerQuotas maxDepth="32" maxStringContentLength="8192" maxArrayLength="16384"
    maxBytesPerRead="4096" maxNameTableCharCount="16384" />
  <security mode="None">
    <transport clientCredentialType="None" proxyCredentialType="None"
      realm="" />
    <message clientCredentialType="UserName" algorithmSuite="Default" />
  </security>
</binding>
```

圖 C-3：繫結方式 (Binding) 內容範例

```
static void Main(string[] args)
{
    Dictionary<string, object> parameters = new Dictionary<string, object>();
    parameters.Add("Active", "True");

    WorkflowApplication app = new WorkflowApplication(new Workflow1(), parameters);
    app.Run();
}
```

圖 C-4：載入 WF 之應用程式執行範例 