

國家原子能科技研究院
委託研究計畫研究報告

配電資訊模型系統傳輸最佳化之研究

Study of Data Exchange Optimization for Common Information Model

計畫編號：112A012

受委託機關(構)：國立高雄科技大學 電機系

計畫主持人：辜德典教授

研究期程：中華民國 112 年 5 月至 112 年 12 月

研究經費：新台幣 59 萬 5 仟元

國原院聯絡人員：曾興嘉

報告日期：112 年 11 月 21 日

目錄

目錄.....	1
圖目錄.....	3
表目錄.....	6
中文摘要.....	7
英文摘要.....	8
壹、計畫緣起與目的	9
貳、研究方法與結果	10
一、國內外資訊模型相關文獻與技術資料蒐集與彙整.....	10
1.1 國內外相關技術彙整.....	10
1.2 國際相關文獻收集.....	16
二、分析配電資訊模型之資料性質及傳輸與儲存分類方式.....	17
2.1 可延伸標記式語言(Extensible Markup Language, XML)	17
2.2 關聯性資料庫.....	22
2.3 非關聯性資料庫.....	25
2.4 關聯性與非關聯性資料庫之比較.....	28
三、運用商用資訊模型轉換器分析配電系統管理運轉資訊傳輸協定.....	29
3.1 商用資訊模型轉換器.....	30
3.2 簡式物件存取協定(Simple Object Access Protocol, SOAP).....	31
3.3 表現層狀態轉換(Representational State Transfer, REST).....	34
3.4 REST 與 SOAP 的比較	37
四、分析配電變壓器視覺化之關鍵參數.....	39
4.1 影響配電變壓器壽命之因素.....	39
4.2 絕緣對配電變壓器壽命之影響.....	40
4.3 負載對於絕緣壽命影響分析.....	40
4.4 諧波對於絕緣壽命影響分析.....	45
五、建構資訊模型傳輸系統架構並提出多系統間介接模式.....	51
5.1 傳輸系統之資料庫系統.....	52
5.2 傳輸系統之存取模組.....	54
5.3 多系統介接模式.....	58
六、分析資訊模型系統傳輸與存取效能及提出最佳化方法.....	59
6.1 測試架構	59
6.2 實際測試與結果分析.....	61
七、變壓器運維狀態與圖形化介面.....	74

參、結論.....	75
肆、參考文獻	76
伍、附錄.....	78
一、國際期刊論文.....	78
二、國內電力研討會論文.....	79

圖目錄

圖 1 RESTFUL WS 建構步驟	10
圖 2 變電所之 CIM RDF XML 文本	11
圖 3 基底 URI 範例	11
圖 4 引用參考範例	11
圖 5 絕對及相對連結參考範例	11
圖 6 變電所包含之電壓級別範例	11
圖 7 IEC 61970-301 之 CIM 頂層套件	12
圖 8 資料庫建立所使用之應用程式流程	13
圖 9 ENTERPRISE ARCHITECT© 軟體中的類別圖	13
圖 10 資料庫的可視化	14
圖 11 轉換 CIM RDF XML 文件的 SQL 命令範例	14
圖 12 透過 SQL 命令生成的 CIM RDF XML 文件	15
圖 13 基本 RDF 數據模型結構	18
圖 14 基本 RDF 數據模型結構	19
圖 15 變壓器 CIM 模型之 RDFS	20
圖 16 CIM 之 XSD 語法範例	22
圖 17 關聯性資料庫資料表示意圖	22
圖 18 NOSQL 以 JSON 模型儲存書籍資料的結構	25
圖 19 文件導向資料庫之範例	26
圖 20 圖形資料庫之範例	27
圖 21 鍵值資料庫之範例	27
圖 22 UDDI 之運行機制	33
圖 23 SOAP 訊息結構	34
圖 24 REST 三角形	37
圖 25 變壓器典型一日負載曲線圖	43
圖 26 變壓器典型一日負載曲線圖	44
圖 27 太陽能發電系統架構圖	45

圖 28 諧波失真示意圖	46
圖 29 (A)電流基波與協波 (B)環境溫度.....	50
圖 30 (A)頂部油溫曲線 (B)繞組最熱點溫度曲線 (C)老化因子FAA曲線	50
圖 31 (A)台灣夏季氣溫一日曲線 (B)頂部油溫曲線.....	50
圖 32 (A)繞組最熱點曲線 (B) 老化因子曲線.....	50
圖 33 配電系統資訊模型傳輸系統架構圖	51
圖 34 SQL SERVER 操作畫面.....	53
圖 35 BASEX 資料庫圖形化介面	54
圖 36 存取模組示意圖	55
圖 37 PYTHON 於 TIOBE 之排名	56
圖 38 資料轉換程式工作示意圖	56
圖 39 網路服務概念圖	57
圖 40 系統介面之逐漸更換與支持 IEC CIM 示意圖	58
圖 41 CIM XML 之資料交換與存取效能之測試概念圖	59
圖 42 資料庫測試架構圖	60
圖 43 資料傳輸測試之架構圖	60
圖 44 POSTMAN 測試頁面.....	61
圖 45 TTU 之 UML 結構模型	62
圖 46 TTU 之 CIM XML 格式	63
圖 47 SQL SERVER 資料庫之資料表及屬性參數.....	64
圖 48 BASEX 資料庫介面與 TTU 之 CIM XML 格式.....	64
圖 49 SQL SERVER 資料庫提供之資料庫介接 API.....	65
圖 50 SQL SERVER 資料庫資料轉換程式部分節錄.....	65
圖 51 SQL SERVER 資料庫存取模組資料轉換後之 CIM XML 格式	66
圖 52 BASEX 資料庫提供之資料庫介接 API	66
圖 53 BASEX 資料庫存取模組獲取之 CIM XML 格式.....	66
圖 54 資料庫存取測試時間差距圖	67
圖 55 RESTFUL 服務之 URI 示意圖	68
圖 56 使用 HTTP 對 RESTFUL 服務之 URI 操作示意圖	68

圖 57 RESTFUL 服務之 URI 具體實現部分節錄.....	68
圖 58 SOAP 服務之 URI 與操作概念圖.....	69
圖 59 SOAP 服務功能定義部分節錄.....	69
圖 60 對 RESTFUL 服務之 URI 進行 GET 操作之資料獲取測試畫面.....	70
圖 61 RESTFUL 服務回覆之 CIM XML 格式.....	70
圖 62 對 RESTFUL 服務之 URI 延伸進行資料查詢之測試畫面.....	70
圖 63 RESTFUL 服務回覆之查詢資料內容.....	71
圖 64 對 SOAP 服務之 URI POST 資料獲取測試封包之實際畫面.....	71
圖 65 SOAP 服務回覆之 CIM XML 格式.....	71
圖 66 對 SOAP 服務之 URI POST 資料查詢測試封包之實際畫面.....	72
圖 67 SOAP 服務回覆之 CIM XML 格式.....	72
圖 68 資料傳輸測試時間差距圖.....	73
圖 69 資料查詢測試時間差距圖.....	73
圖 70 安全性登入.....	74
圖 71 運維狀態圖形化介面.....	74

表目錄

表 1 RDFS 之基礎詞彙表-類別(CLASSES)	18
表 2 RDFS 之基礎詞彙表-特性(PROPERTIES)	19
表 3 XSD 基本單元說明.....	21
表 4 關聯性與非關聯性之比較	29
表 5 SOAP 訊息元素之結構說明	33
表 6 SOAP 訊息之錯誤代碼.....	34
表 7 為 RESTFUL API 與一般 API 在 HTTP 上實現的比較.....	37
表 8 SOAP 與 REST 之比較.....	37
表 9 變壓器絕緣殘餘壽命	43
表 10 變壓器絕緣殘餘壽命	44
表 11 電壓失真限制.....	46
表 12 最大奇次諧波電流失真限制	47
表 13 最大偶次諧波電壓失真限制	47
表 14 每項總電壓諧波失真值的負載損耗	47
表 15 每項總電流諧波失真值的負載損耗	47
表 16 每項總電流諧波失真值造成的生命損失	47

中文摘要

本計畫針對共同資訊模型(Common Information Model, CIM)之資料傳輸，分析簡式物件存取通訊協定(Simple Object Access Protocol, SOAP)與表現層狀態轉換(Representational State Transfer, RESTful)傳輸差異，建立其 CIM 之傳輸系統架構，並以 SOAP 與 RESTful 所提供之應用程式介面(Application Programming Interface, API)完成不同系統之資料介接，並針對其資料存取分析關聯性(Structured Query Language, SQL)與非關聯性(Not Only SQL, NoSQL)資料庫(Database)之架構及差異性，以提出提升 CIM 之資料交換及存取效能之方法。另一方面，針對配電變壓器之運轉維護，本計畫亦應用 CIM 進行設備管理，同時根據提供之即時數據資料，分析大量再生能源對配電變壓器之影響，並以視覺化方式進行呈現，有效達到配網端設備之資訊收集與交換，進一步達到配電設備資產管理之應用功能。

英文摘要

The aim of this project is to analyze the differences between simple object access protocol (SOAP) and the representational state transfer (RESTful) for the common information model (CIM). The SOAP and RESTful are used to exchange data and verify their performance in the proposed data exchange architecture. The application programming interface (API) of SOAP or RESTful is used to exchange data for different systems and record data sets in the database. The structured query language (SQL) and not only SQL (NoSQL) databases are selected for the comparisons of the access performance. Besides, the CIM is also used for the operation and maintenance of the distribution transformers. The real-time information of CIM is used to analyze the impact of distribution transformers and display the status of the distribution transformers on the graphic user interface (GUI) when high-penetration renewable energy is installed in the distribution system. Finally, the proposed data exchange architecture with database method collects and exchanges data between facilities to achieve the asset management of the distribution systems.

壹、計畫緣起與目的

台電公司針對配電系統已建構數個資訊系統進行資料收集與分工處理，其系統可執行對應之工作流程，而為使配電系統達到智慧電網(Smart Grid)之應用功能，台電公司已開始推展先進配電管理系統(Advanced Distribution Management System, ADMS)之建置工程，其 ADMS 主要目的為整合不同資訊系統，進行配電系統之運轉與管理，同時可藉由應用程式(Applications, APPs)之功能開發，進一步提升配電系統之運轉效率與供電品質。為使 ADMS 可以順利達成預期功能與發揮功效，其資訊系統之資料交換顯得相當重要，目前國際與電力有關之資訊系統，其國際電工委員會(International Electrotechnical Commission, IEC)提出 CIM 作為各資訊系統之資料交換方式，藉以降低整體 ADMS 之複雜度外，亦透過標準資料轉換模型降低整體系統之轉換程式數量與其開發成本。

對於 CIM 之資料傳輸而言，目前除使用 SOAP 之外，亦使用 RESTful 以 API 方式進行資料介接，這兩種形式在資料傳輸上其性能及架構對於 ADMS 而言為其重要，其資料傳輸之即時性與安全性將會對 ADMS 之應用程式功能有著顯著之影響，同時對於所收到之資料在資料儲存上也必須進行探討，因此對於關聯性與非關聯性之資料庫在資料存取性能也必須深入了解，對此本計畫針對台電未來建構之 ADMS 系統，應用在雲林區處已佈建之變壓器端末單元(Transformer Terminal Unit, TTU)，以 CIM 為基礎規劃其 TTU 之資訊模型，並執行饋線電壓監測，同時提出具可靠性之 SOAP 與 RESTful 訊息架構，以確保 CIM 於資訊交換時之可靠性，進一步降低 ADMS 之資料交換複雜度，同時評估關聯性與非關聯性資料庫在資料存取上之性能，評估並建構合適於 ADMS 使用之資料庫，並以 TTU 所收集之資料發展配電變壓器運維應用程式功能，以圖形化方式進行運轉狀態呈現，以達到配電系統之智慧電網應用功能，進一步支援配電設備之資產管理之功能。

貳、研究方法與結果

一、國內外資訊模型相關文獻與技術資料蒐集與彙整

1.1 國內外相關技術彙整

1.1.1 CIM 的傳輸技術: Web 服務

共同資訊模型(Common Information Model, CIM)的目的是提供不同系統間更簡單地實現資料交換和整合，提高系統間的互通性和互操作性，而國際電工委員會(International Electrotechnical Commission, IEC)已對 CIM 各層次標準進行了定義，如 IEC 61970、IEC 61968、IEC 62325 等。CIM 為一種資料模型，可以透過多種方式進行傳輸，而在機器間之傳輸方式，最常見的為 Web 服務，如簡式物件存取協定(Simple Object Access Protocol, SOAP)及表現層狀態轉換(Representational State Transfer, REST)。

IEC 已在 IEC 61968-100 標準定義了一個包含名詞(例如 MeterReading)和動詞(例如 GET、CREATE 或 DELETE)的 Header 元素，以針對使用 SOAP 作為兩個系統之間的訊息交換。而 REST 是一種將每個對象視為具有統一資源標識符(UniformResourceIdentifier,URI)的資源，並基於操作超文字傳輸協定(HyperTextTransferProtocol,HTTP)方法(最常見的是 GET、PUT、POST 和 DELETE)的應用程式架構風格，具有輕量級、可擴展、簡單等特點。

在參考文獻中，已提出一個可以透過基於資源導向架構(Resource-Oriented Architecture, ROA)作為實現 Restful Web Service 的方法，它是由 URI、HTTP、可延伸標記式語言(Extensible Markup Language, XML)所組成，藉由操作資源、資源的名字、資源的描述(Representations)及資源之間的連結實現。ROA 將 RestFul WS 之架構創建分為以下步驟，如圖 1 所示。

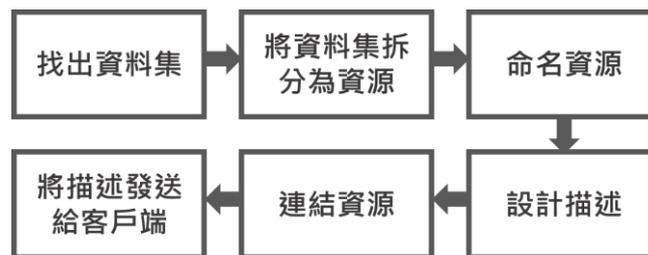


圖 1 RestFul WS 建構步驟

- 設計描述

CIM 透過 XML 映射出的 CIM RDF XML 已完成前面 3 個步驟，CIM RDF XML 為一個 XML 化的資源描述框架(Resource Description Framework, RDF)，RDF 描述了約束但不禁止擴展，它定義了如何將電力系統資源完成分類、命名並設計了相關的描述再映射為 XML，再透過實現步驟 5、6 之動作，便可架構出完整的 RestFulWS。圖 2 為 CIM RDF XML 的文本摘錄，通過唯一標示、別名、敘述、名稱完成了變電所的描述。

```

<cim:Substation rdf:ID="_4139122910736445088">
<cim:IdentifiedObject.aliasName>BBNTP</cim:IdentifiedObject
.aliasName>
<cim:IdentifiedObject.description>BBNTP</cim:IdentifiedObject.description>
<cim:IdentifiedObject.name>JDM_17071</cim:IdentifiedObject.name>
</cim:Substation>

```

圖 2 變電所之 CIM RDF XML 文本

- 連結資源

為了連結描述，首先需要為 RESTful WS 定義基本的 URI，如圖 3 所示。其餘的 URI 都將根據此基底做延伸，如有一變電所為 JDM_17071，該 URI 就應設為 http://www.utility.com/EMS/v1/Substation/JDM_17071。

基底URI : <http://www.utility.com/EMS/v1/>
協議標識 伺服器網域名 應用程式名稱及版本

圖 3 基底 URI 範例

CIM RDF XML 實現通常根據使用文本內的相對 URI 以互相關聯，圖 4 為一台設備如何引用其設備容器。為實現 ROA 的主要原則之一:連結性，該 RESTful WS 提供的每一個片段都必須為獨立且有效的描述文本。這意味著每個片段資源可以連結成為一個有效的大型描述文本，這可以透過圖 5 所示之相對連結完成，並且我們可以藉由新增一個由 rdf:about 屬性構成的絕對連結來連結到該資源的單獨描述，如圖 6 所示。圖 6 為當單個資源包含了多個資源，可以透過上述之連結，相對連結用於多個資源關聯某個資源，而絕對連結，可以連結該資源的有效描述。

```

<cim:Equipment.MemberOf_EquipmentContainer
rdf:resource="#_8146570711573811664"/>

```

圖 4 引用參考範例

```

<cim:Equipment.MemberOf_EquipmentContainer
rdf:resource="#_8146570711573811664"
rdf:about="/VoltageLevel/_8146570711573811664"/>

```

圖 5 絕對及相對連結參考範例

```

<cim:Substation.Contains_VoltageLevels
rdf:parseType="Collection">
<cim:VoltageLevel rdf:resource="#_8146570711573811664"
rdf:about="/VoltageLevel/_8146570711573811664" />
<cim:VoltageLevel rdf:resource="#_8146570711573811665"
rdf:about="/VoltageLevel/_8146570711573811665" />
...
</cim: Substation.Contains_VoltageLevels>

```

圖 6 變電所包含之電壓級別範例

- 將描述發送給客戶端

RestFul WS 之特點為透過 HTTP 方法(如 POST、GET、PUT 等)對資源 URI 進行操作，以便更簡單直觀得使用。如對 JDM_17071 變電所的 URI:http://www.utility.com/EMS/v1/Substation/JDM_17071 進行 GET 操作，便能獲得該變電所的描述文本。

1.1.2 CIM 的儲存技術

目前主流的資料庫分為:關聯性資料庫與非關聯性資料庫，兩種資料庫皆有各自的優缺點及應用場景。CIM 之結構可分為兩個部分，先以統一塑模語言(Unified Modeling Language, UML)對資料進行定義，再將所建立之資料模型透過 XML 映射成一套標準訊息結構進行傳輸。故可以理解為欲儲存 CIM，並先得對其 UML 之結構進行儲存，在透過該定義之模型，取出資料並透過 UML 之繼承、連結等關係進行 XML 之映射在進行傳輸。因為電力系統包含了許多設備及組件，多種模型及架構會導致儲存上的困難與結構混亂。

CIM 實際應用上較常使用之形式為透過類別(Class(es))、屬性(Attributes)、套件(Packages)、繼承(Inheritance)及連結(Associations)等物件(Objects)相關概念展示系統內部之結構基礎與其交互關係。而在 UML 語法中，可將數個相關類別及其交互關係歸納成一個套件(Packages)，以此來簡化模型之複雜度，增加設計過程中之可讀性。套件之使用並無明確限制，可包含整個模型或只包含單一類別，甚至套件內包含數個套件，皆取決於使用者對系統之結構設計，而模型中若套件間有相互依賴之關係則可用虛線箭頭表示。對於 CIM 之套件規範，IEC 已定義其細節，紀錄於 IEC61970-301、IEC61968-11 及 IEC62325-301 等規範內，圖 7 為 IEC61970-301 規範中所定義 CIM 之頂層套件結構。

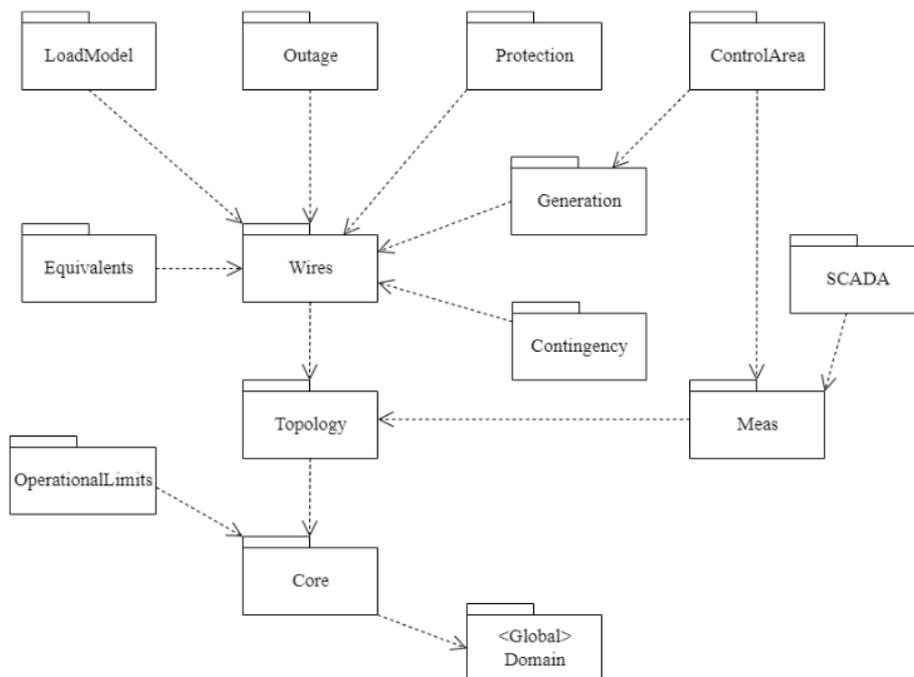


圖 7 IEC 61970-301 之 CIM 頂層套件

接下來，透過參考文獻中的其中一種實現說明常見的 CIM 儲存方式，圖 8 為該實現的流程圖。

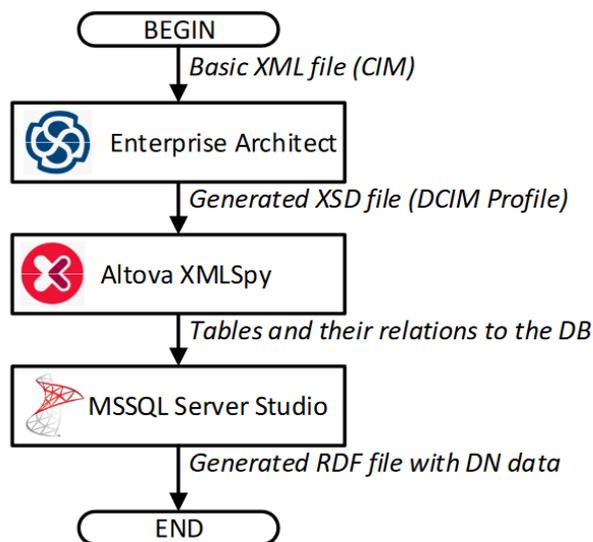


圖 8 資料庫建立所使用之應用程式流程

- 第一步，將 CIM 之輸入檔案透過 Enterprise Architect©軟體，生成一個對 XML 進行約束的 XML 概要定義(XML Schema Definition, XSD)檔案，這個檔案描述了輸入檔案中所有的類別和其子類別的屬性及類別之間的繼承、連結關係，這套軟體支援 IEC 61970-301 和 IEC 61968-11 標準所定義之 CIM 的所有類別。圖 9 有包含了兩部份，展示了使用 Enterprise Architect©軟體獲得的類別圖，左邊的部分為資料包，包含了該配置文件的所有類別，右邊的部分顯示了配電網路的模型。

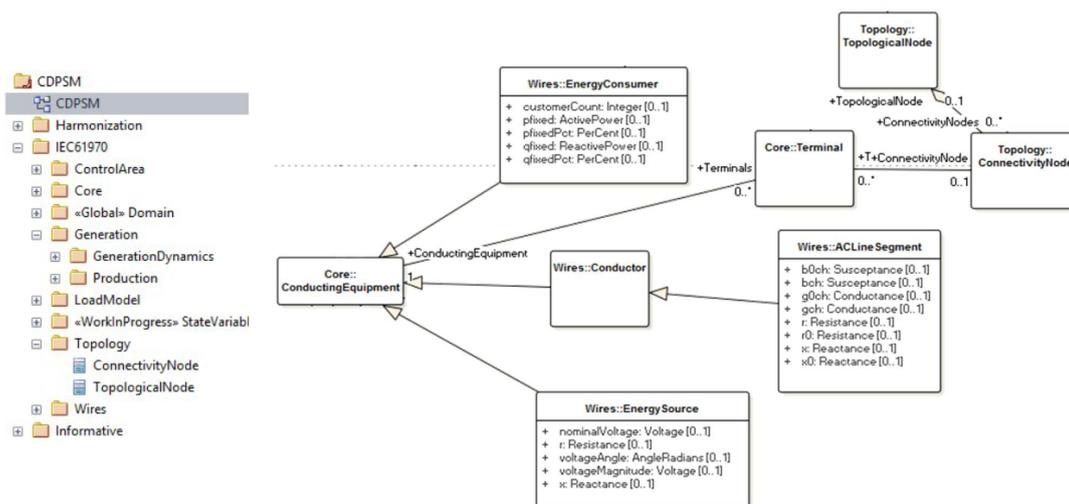


圖 9 Enterprise Architect© 軟體中的類別圖

- 第二步，使用 Altova XMLSpy©這套軟體處理該 XSD 檔案，該軟體會將該檔案轉換為結構化查詢語言(Structured Query Language, SQL)命令，即可以根據輸入的檔案將標準化資料表插入資料庫，並在定義資料表後，會根據

Enterprise Architect©軟件的類別圖自動指定資料列。指定完資料列後，必須要手動定義每個資料列將操作的變數及其類型以及主索引鍵與外部索引鍵，因為從 Enterprise Architect©生成的 XSD 檔案不包含表格和資料列的變數之正確類型(如:布林值、整數、字串等)。最後，通過將 Altova XMLSpy©與 MSSQL server 連接來插入這些資料表，就可以透過 SQL Server Management Studio©軟體來管理及維護。藉由這種方式，所創建的資料庫就可以儲存來自現實配電網路的資料。圖 10 中展示的是資料庫的可視化，有兩個部分，在左邊的部分中，顯示了資料庫中的大部分資料表，在右邊部分中，顯示了主索引鍵和外部索引鍵的關係。

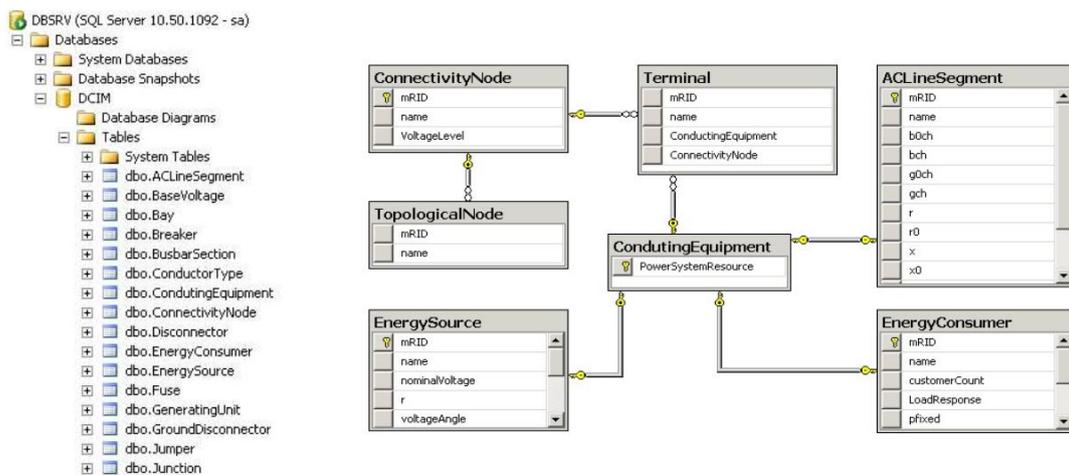


圖 10 資料庫的可視化

- 第三步，利用資料庫中所儲存之資料創建通用於系統之間訊息交換的 CIM RDF XML 文件。圖 11 顯示了 Microsoft SQL Server Management Studio©中的 SQL 命令，用於生成資料庫資料的 XML 文件，輸出文件如圖 12 所示。

```

WITH XMLNAMESPACES (
    'http://iec.ch/TC57/CIM100#' as cim,
    'http://www.w3.org/1999/02/22-rdf-syntax-ns#' as rdf)
SELECT
    (SELECT mRID AS "@rdf:ID",
     mRID, name, nominalVoltage, r, x, PSRType, VoltageLevel
     FROM dbo.EnergySource FOR XML PATH ('cim:EnergySource'), TYPE),
    (SELECT mRID AS "@rdf:ID",
     mRID, b0ch, r, x, BaseVoltage, Line, ConductorType
     FROM dbo.ACLineSegment FOR XML PATH ('cim:ACLineSegment'), TYPE),
    (SELECT mRID AS "@rdf:ID",
     mRID, customerCount, pfixed, qfixed, VoltageLevel
     FROM dbo.EnergyConsumer FOR XML PATH ('cim:EnergyConsumer'), TYPE)
FOR XML PATH (''), root ('rdf:RDF');
GO
    
```

圖 11 轉換 CIM RDF XML 文件的 SQL 命令範例

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cim="http://iec.ch/TC57/CIM100#">
  <cim:EnergySource rdf:ID="EquivalentSource_01">
    <mRID>EquivalentSource_01</mRID>
    <name>AIGUEBHT01</name>
    <nominalVoltage>9.0000000e+001</nominalVoltage>
    <r>0.0000000e+000</r>
    <x>1.7052631e+001</x>
    <PSRType>PSRType_1</PSRType>
    <VoltageLevel>VL_01</VoltageLevel>
  </cim:EnergySource>
  <cim:ACLineSegment rdf:ID="ACLS_0216_1">
    <mRID>ACLS_0216_1</mRID>
    <b0ch>4.4900000e+002</b0ch>
    <r>2.0000000e-001</r>
    <x>1.0000000e-001</x>
    <BaseVoltage>BaseVoltage_02</BaseVoltage>
    <Line>Line_0216</Line>
    <ConductorType>ConductorType_1</ConductorType>
  </cim:ACLineSegment>
  <cim:EnergyConsumer rdf:ID="EC_205">
    <mRID>EC_205</mRID>
    <customerCount>1.7300000e+002</customerCount>
    <pfixed>8.1000000e-002</pfixed>
    <qfixed>2.6930000e-003</qfixed>
    <VoltageLevel>VL_05</VoltageLevel>
  </cim:EnergyConsumer>
</rdf:RDF>

```

圖 12 透過 SQL 命令生成的 CIM RDF XML 文件

1.2 國際相關文獻收集

- A. W. McMorran, R. W. Lincoln, G. A. Taylor, and E. M. Stewart, "Addressing misconceptions about the Common Information Model (CIM)," in *2011 IEEE Power and Energy Society General Meeting*, Jul. 2011, pp. 1–4. doi: [10.1109/PES.2011.6039391](https://doi.org/10.1109/PES.2011.6039391).
- E. Varga, I. Lendak, M. Gavrić, and A. Erdeljan, "Applicability of RESTful web services in control center software integrations," in *2011 International Conference on Innovations in Information Technology*, Apr. 2011, pp. 282–286. doi: [10.1109/INNOVATIONS.2011.5893833](https://doi.org/10.1109/INNOVATIONS.2011.5893833).
- G. Ravikumar, S. A. Khaparde, and Y. Pradeep, "CIM oriented database for topology processing and integration of power system applications," presented at the *IEEE Power and Energy Society General Meeting*, Jan. 2013, pp. 1–5. doi: [10.1109/PESMG.2013.6672164](https://doi.org/10.1109/PESMG.2013.6672164).
- S. Devic, B. Atlagic, and Z. Gorecan, "Database Modelling and Development of Code Generator for Handling Power Grid CIM Models".
- J. V. De Barros and J. B. Leite, "Development of a Relational Database Oriented on the Common Information Model for Power Distribution Networks," in *2021 IEEE URUCON*, Jan. 2021, pp. 63–66. doi: [10.1109/URUCON53396.2021.9647155](https://doi.org/10.1109/URUCON53396.2021.9647155).
- C. Zolotas, T. Diamantopoulos, K. C. Chatzidimitriou, and A. L. Symeonidis, "From requirements to source code: a Model-Driven Engineering approach for RESTful web services," *Autom Softw Eng*, vol. 24, no. 4, pp. 791–838, Dec. 2017, doi: [10.1007/s10515-016-0206-x](https://doi.org/10.1007/s10515-016-0206-x).
- J. Wu and N. N. Schulz, "Overview of CIM-oriented database design and data exchanging in power system applications," in *Proceedings of the 37th Annual North American Power Symposium*, 2005., Oct. 2005, pp. 16–20. doi: [10.1109/NAPS.2005.1560495](https://doi.org/10.1109/NAPS.2005.1560495).
- I. Lendak, E. Varga, A. Erdeljan, and M. Gavric, "RESTful web services and the Common Information Model (CIM)," in *2010 IEEE International Energy Conference*, Manama, Bahrain: IEEE, Dec. 2010, pp. 716–721. doi: [10.1109/ENERGYCON.2010.5771774](https://doi.org/10.1109/ENERGYCON.2010.5771774).

二、分析配電資訊模型之資料性質及傳輸與儲存分類方式

為推動節能減碳政策，智慧電網已納入「國家綠能低碳總行動方案」標竿計畫，而 IEC 訂定之 IEC 61970 能源管理系統應用程式介面 (Energy Management System-Application Program Interface, EMS-API) 則為重要發展目標之一。其中，CIM 是 EMS-API 標準模型的主要組成部分，可用於提高不同應用程式之間的互操作性和數據共享。

CIM 是基於 UML 定義的電力系統模型，再將其透過具有可讀性、可擴展性、跨平台性和多功能性的資料描述語言等特性的 XML，映射成一套標準訊息結構進行傳輸，實現不同的系統和應用程式間可以共享和交換電力系統資訊。IEC 也已透過 IEC 61970-501、IEC 61970-552 標準對 CIM 之 XML 標籤結構及映射格式進行定義。

為達到智慧電網推動之目標，本計畫對網路傳輸協定與技術進行比較，對其資訊傳輸速率及穩定度、安全性進行研究。而因應智慧電網推動，導致系統內之感測器、量測設備數據的遽增，數據的存取與維護必為重要之工作，本計畫亦對各式資料庫之類型與技術進行研究，比較 CIM 映射之 XML 檔案存儲於資料庫中其存取之效率與性能。

2.1 可延伸標記式語言(Extensible Markup Language, XML)

XML 是一種通用的標記式語言，在 1998 年 2 月成為全球資訊網協會(World Wide Web Consortium, W3C) 推薦標準，XML 的設計目標是提供一個靈活且可擴展的標記語言，使得不同系統和平台之間可以有效地傳遞和解析結構化資料。相較於同為 W3C 推薦之開放標準的超文本標記語言(Hyper Text Markup Language, HTML)同為全球資訊網協會(World Wide Web Consortium, W3C)，XML 被設計用於儲存和傳輸資料數據，著重於描述所攜帶的資訊內容，而非對其表現和展示。

XML 使用標籤來標記資料元素，每個元素由起始標籤和結束標籤組成，中間可以包含子元素、屬性和文本內容。這種結構化的格式使得 XML 非常適合在不同系統和平台之間傳遞和解析資料。

XML 的主要特點如下：

- 可擴展性:XML 允許用戶定義自己的標籤和結構，以適應特定的需求和領域。這使得 XML 非常靈活，可以應對各種資料結構和格式。加上本身並無預定義之標籤，使用者可自行針對訊息內容進行標籤創建
- 可讀性: XML 使用自然語言而非計算機語言，且使用有意義且具有描述性的標籤和結構，使得資料易於閱讀和理解。由於每個 XML 標籤都緊跟在相關數據之前，使得信息結構容易被人類和計算機識別，具有可讀性和易於理解的特點。
- 跨平台和語言:XML 以純文本格式存儲數據，提供了一種獨立於軟體和硬體的方式來存儲、傳輸和共享數據。更提供了一個通用且強大的標記語言，能夠應對不同類型的資料和需求。無論是用於數據庫交換、網路通訊、文件表

示還是配置文件，可以在不同的系統和程式語言之間共享和解析資料。

- 支援層次結構:XML 允許創建多層次的結構，使得資料可以按照層次組織和表示。這種結構化的特性使得 XML 非常適合於描述和交換複雜的資料結構，並且可以輕鬆地定義父子關係、屬性和元素之間的關聯。

2.1.1 資源描述框架(Resource Description Framework, RDF)

資源描述框架(Resource Description Framework, RDF)是 W3C 所提出基於標記式語言所衍伸出的架構規範，設計初衷是為解決元數據(Metadata)在資源描述上混亂無序之問題，RDF 提供多種用於編碼、資料交換及讀取之序列化標準，透過通用且制式之格式提升設備或程序間資源使用上之精確性，以此建構出中立且獨立於操作系統之元數據模型，並於 1999 年發布第一個 RDF 規範標準，如今版本為 RDF1.1，於 2014 年發行。RDF 主要用於呈現網際網路上之各項資源與其對應連結性，並載明了特定領域的綱要(Schema)，宣告該領域的資源描述語彙，以表達領域內對元素之定義、概念及層次結構等，最後透過語法敘述出其邏輯關係，建立出使機器能夠理解及處理而不喪失語意的通用框架。

RDF 數據模型以主語-謂語-賓語之形式對資料數據進行敘述(Statements)，又稱為三元組(Triples)，主語表示資源(Resource)，謂語表示資源之特徵或屬性(Property)，賓語表示主客體間連結關係所描述的值(Value)或資源，形成了一個由主語和賓語組成的有向圖，基本 RDF 數據模型結構如圖 13 所示，其中資源可以是一串字元、數字與 URI 等，又或者是以空白節點表示匿名資源，在 Web 相關應用程序中資源以 URI 表示十分普遍，用於引用唯一識別之網路或非網路資源。而在 RDF 訊息敘述中所遵循之編碼機制與基礎詞彙則稱為 RDF 概要(Resource Description Framework Schema, RDFS)，RDFS 定義了 RDF 中資源與屬性間的基礎語言，並對規範進行擴展，提供了組織元數據的基本元素與格式標準，來詮釋模型所描述之層次結構體系，使得 RDF 數據具備更豐富的語意描述及推論能力，而 RDF1.1 規範中定義之數據建模相關詞彙表如表 1 與 2 所示。

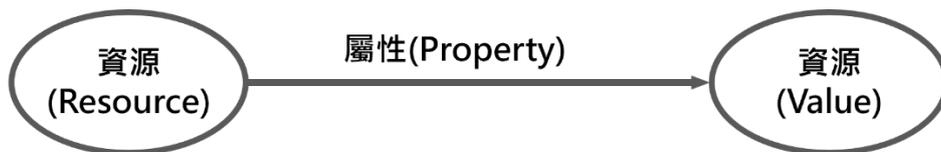


圖 13 基本 RDF 數據模型結構

表 1 RDFS 之基礎詞彙表-類別(Classes)[4]

類別名稱		
rdfs:Resource	rdfs:Class	rdfs:Literal
rdfs:Datatype	rdf:langString	rdf:HTML
rdf:XMLLiteral	rdf:Property	rdf:Statement
rdf:Bag	rdf:Seq	rdf:Alt
rdfs:Container	rdfs:ContainerMembershipProperty	rdf>List

表 2 RDFS 之基礎詞彙表-特性(Properties) [4]

特性名詞		
rdf:type	rdfs:subClassOf	rdfs:subPropertyOf
rdfs:domain	rdfs:range	rdfs:label
rdfs:comment	rdfs:member	rdf:first
rdf:rest	rdfs:seeAlso	rdfs:isDefinedBy
rdf:value	rdf:subject	rdf:predicate
rdf:object		

上述提到 RDF 本質上是為建立出中立且獨立於操作系統之元數據模型，故對應不同格式或語言之系統存在多種序列化格式，將數據模型之節點關係圖映射成機器可讀信息，以便計算機應用程序可以讀取、理解並推論信息，而此種序列化之語義敘述規則稱為 RDF 語法(Syntax)，本節將針對基於 XML 所衍伸出之 RDF/XML 語法進行概要說明：

• RDF/XML

RDF/XML 是由 W3C 所定義，也是歷史上第一個定義 RDF 語法之規範標準，使用 XML 所撰寫，用於將 RDF 數據模型圖轉換 XML 文本，透過 XML 來表示 RDF 圖形中的主體、屬性和值等概念，使數據能在不同類型的操作系統與應用程序間達到跨平台之資料交換，然而並非所有 RDF 模型皆選擇 RDF/XML 進行格式表達，這僅僅只是眾多 RDF 序列化格式其中之一。下述以 CIM 之變壓器模型為例，使用 RDF 轉換 UML 模型圖至 XML 文本格式進行範例說明：

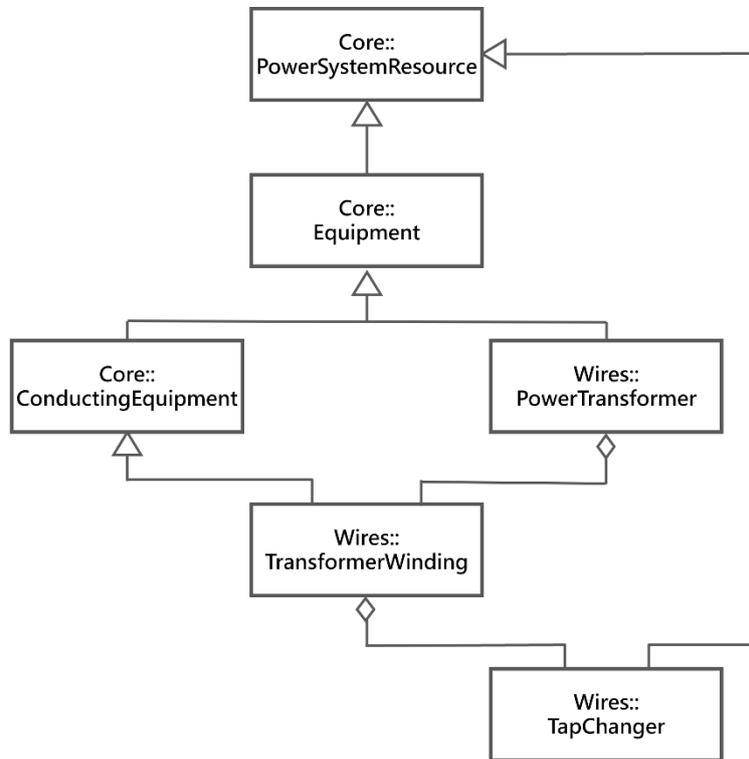


圖 14 基本 RDF 數據模型結構[5]

```

<rdfs:Class rdf:ID="PowerSystemResource">
  <rdfs:label xml:lang="en">PowerSystemResource</rdfs:label>
  <rdfs:subClassOf rdf:resource="#IdentifiedObject"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Equipment">
  <rdfs:label xml:lang="en">Equipment</rdfs:label>
  <rdfs:subClassOf rdf:resource="#PowerSystemResource"/>
</rdfs:Class>
<rdfs:Class rdf:ID="ConductingEquipment">
  <rdfs:label xml:lang="en">ConductingEquipment</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Equipment"/>
</rdfs:Class>
<rdfs:Class rdf:ID="PowerTransformer">
  <rdfs:label xml:lang="en">PowerTransformer</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Equipment"/>
</rdfs:Class>
<rdfs:Class rdf:ID="TransformerWinding">
  <rdfs:label xml:lang="en">TransformerWinding</rdfs:label>
  <rdfs:subClassOf rdf:resource="#ConductingEquipment"/>
</rdfs:Class>
<rdfs:Class rdf:ID="TapChanger">
  <rdfs:label xml:lang="en">TapChanger</rdfs:label>
  <rdfs:subClassOf rdf:resource="#PowerSystemResource"/>
</rdfs:Class>
<rdfs:Property rdf:ID="TransformerWinding.PowerTransformer">
  <rdfs:label xml:lang="en">PowerTransformer</rdfs:label>
  <rdfs:domain rdf:resource="#TransformerWinding"/>
  <rdfs:range rdf:resource="#PowerTransformer"/>
</rdfs:Property>
<rdfs:Property rdf:ID="TapChanger.TransformerWinding">
  <rdfs:label xml:lang="en">TransformerWinding</rdfs:label>
  <rdfs:domain rdf:resource="#TapChanger"/>
  <rdfs:range rdf:resource="#TransformerWinding"/>
</rdfs:Property>

```

繼承

聚合連結

圖 15 變壓器 CIM 模型之 RDFS[1]

根據上述範例可得知，RDF 針對 CIM 之類別層次結構提供一套完整之序列化格式及標準語法，將 CIM 之 UML 結構模型透過映射轉換成 XML 之形式，供電力系統相關程序應用，且語法上是人機皆可讀的。此外，語義敘述上不只表明格式轉換所引用之規章標準及版本，並使用唯一的識別碼(ID)定義該資源之來源與連結關係，如 URI、IRI 或設備序號等標識符，除了使該 XML 文本具備階層式概念外，亦能從連結與集合關係推論出其模組化結構，使文本內容更加便於管理與解析，而其轉換格式定義紀錄於 IEC 61970-552 規範中。

2.1.2 XML 概要(XML Schema)

由於 XML 在資料定義上有極大的彈性，所以衍伸出許多的 XML 格式，稱為 XML 概要(XML Schema)。XML Schema 用於定義了 XML 文本的元素、屬性、資料類型以及它們之間的關係。它使用 XML 語法來描述文本結構，並定義了每個元素和屬性的有效值範圍、約束條件和關聯性。通過定義 XML Schema，可以確保 XML 文本符合特定的結構和規範，並且可以在解析和處理文本時進行驗證，這對於資料交換、驗證和整合非常有用。XML Schema 被應用於各種領域，包括網際網路、電子商務、資料庫管理和 Web 服務等，針對不同情境與應用，XML Schema 存在多種概要語言(Schema Language)標準，如 XSD、RELAX NG、DTD 等，其中又以 XSD 在應用上最為廣泛。

XML 概要定義(XML Schema Definition, XSD)是由 W3C 所制定和管理，內容支持複雜的數據類型、約束和層次結構等特性，也是 Java 和 .NET 平台中 XML 數據綁定的規範基礎。XSD 能夠對 XML 文本結構進行約束，只有發送端與接收端雙方均遵循該語法規則且解讀方式相同時，方能視為有效通訊，通常可以在 XML 開發工具中編輯和檢查，以 .xsd 為副檔名，XSD 的基本組成單元及用法如表 3 所示。而對於 CIM 設定檔(Profile)中所使用之 XSD 語法 IEC62361-100 已對其定義，規範中敘述了如何將 CIM 中 UML 之語義結構能有效且一致性地映射至 XML Schema 訊息中，確保用戶與供應商間不會因語法標準不同，而導致映射後 XML 格式上產生分歧與不兼容，也同時為文檔映射過程提供了溯源依據，CIM 之 XSD 結構如圖 16 所示，因文檔內容繁多，故僅擷取部分內容。

表 3 XSD 基本單元說明[6]

XSD 基本單元	功能說明
元素 (Element)	元素是 XML 文本中的基本結構單位，可以定義其名稱、數據類型、出現次數、順序、默認值與必要性等屬性信息。
屬性 (Attribute)	屬性是 XML 文本中對元素描述的一個關聯值，從而提高文檔的可讀性及可擴展性。
資料型別 (Data type)	XSD 規範中定義了許多基本的資料型別，如字符串、整數、日期和時間等，也可以定義複合的數據類型，如結構體、列舉和聯合等。
命名空間 (Namespace)	命名空間使用一個識別碼定義 XML 元素所處的區域範圍，以區分不同的 XML 資源，不同命名空間相互獨立且含義不同，通過劃分不同區域，宣告識別碼之邏輯群組，使整個系統更加模組化。
繼承 (Inheritance)	用於描述 XML 元素和屬性之間的繼承與集合關係，以實現 XML 文本的層次結構，使其便於管理與解析。
約束 (Constraint)	可以定義數據的約束和限制，如範圍、長度、格式與資料型別等，以確保 XML 文件的合法性和完整性，防止文本中出現無效數據。

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:a="http://langdale.com.au/2005/Message#" xmlns:sawSDL="http://www.w3.org/2003/05/xml-schema-annotation/">
  <xs:element name="TransformerData" type="m:TransformerData"/>
  <xs:complexType name="TransformerData">
    <xs:sequence>
      <xs:element name="BaseVoltage" type="m:BaseVoltage" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="PowerTransformer" type="m:PowerTransformer" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="PowerTransformerEnd" type="m:PowerTransformerEnd" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="BaseVoltage" sawSDL:modelReference="http://iec.ch/TC57/2010/CIM-schema-cim15#BaseVoltage">
    <xs:annotation>
      <xs:documentation>Defines a system base voltage which is referenced.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="nominalVoltage" minOccurs="1" maxOccurs="1" type="m:Voltage" sawSDL:modelReference="http://iec.ch/TC57/2010/CIM-schema-cim15#Voltage">
        <xs:annotation>
          <xs:documentation>The PowerSystemResource's base voltage.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="PowerTransformer" sawSDL:modelReference="http://iec.ch/TC57/2010/CIM-schema-cim15#PowerTransformer">
    <xs:annotation>
      <xs:documentation>An electrical device consisting of two or more coupled windings, with or without a magnetic core, for introducing or changing the voltage and current in an AC power system. A power transformer may be composed of separate transformer tanks that need not be identical.</xs:documentation>
      <xs:documentation>A power transformer can be modelled with or without tanks and is intended for use in both balanced and unbalanced systems.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="mRID" minOccurs="1" maxOccurs="1" type="xs:string" sawSDL:modelReference="http://iec.ch/TC57/2010/CIM-schema-cim15#mRID">
        <xs:annotation>
          <xs:documentation>A Model Authority issues mRIDs. Given that each Model Authority has a unique id and this id is part of the mRID, Global uniqueness is easily achieved by using a UUID for the mRID. It is strongly recommended to do this.</xs:documentation>
          <xs:documentation>For CIMXML data files the mRID is mapped to rdf:ID or rdf:about attributes that identifies CIM object elements.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="name" minOccurs="1" maxOccurs="1" type="xs:string" sawSDL:modelReference="http://iec.ch/TC57/2010/CIM-schema-cim15#name">
        <xs:annotation>
          <xs:documentation>The name is any free human readable and possibly non unique text naming the object.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="PowerTransformerEnd" minOccurs="2" maxOccurs="2" type="m:PowerTransformerEnd" sawSDL:modelReference="http://iec.ch/TC57/2010/CIM-schema-cim15#PowerTransformerEnd">
        <xs:annotation>
          <xs:documentation>The other end of the transformer, which may be the same as the first end if the transformer is self-coupled.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

```

XSD基本元素

XSD元素之屬性內容

圖 16 CIM 之 XSD 語法範例[1]

2.2 關聯性資料庫

關聯性資料庫是一種廣泛應用於組織和管理結構化資料的資料管理系統。它是一種基於關聯模型的資料庫，透過建立資料表之間的邏輯連結，有效地組織和存儲資料。在此架構下，資料以表格的形式表示，稱為資料表，資料表被視為結構化的資料單元，每個資料表都具有一個獨特的名稱，例如客戶表或訂單表。而資料表由多個屬性(亦稱為資料欄)組成，每個屬性具有特定的資料類型和名稱，例如姓名或地址。這些屬性集合組成資料表中的紀錄(亦稱為值組或資料行)，每個紀錄包含相關屬性的實際資料。透過定義值域，確定了每個屬性所允許的合法值範圍。關聯性資料庫資料表的範例如圖 17 所示。

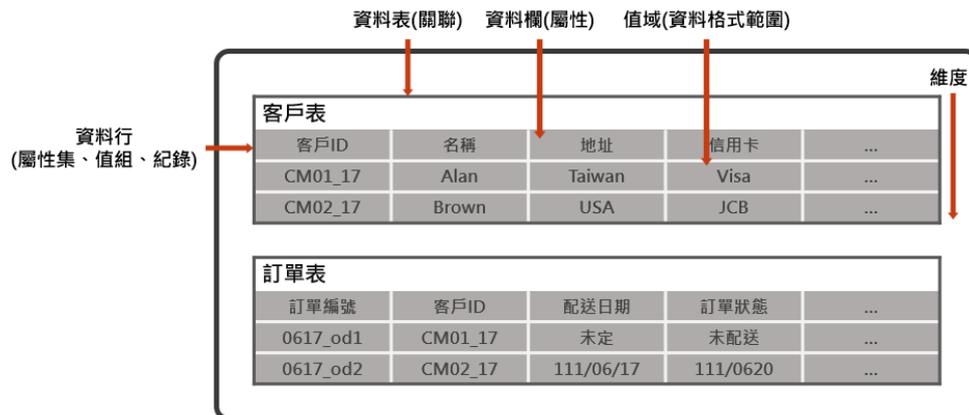


圖 17 關聯性資料庫資料表示意圖

- **關聯(Relation):**
關聯是不同資料表格之間的邏輯連接，它基於這些資料表格的互動關係建立，以實現資料之間的聯繫和關聯。在關聯性資料庫中，每個資料表格被稱為「關聯」或「資料表(Table)」，並被賦予一個名稱，例如客戶表、訂單表等。
- **屬性(Attribute):**
資料表由多個屬性構成，也被稱為「資料欄(Field)」。屬性定義了資料的類型，每個屬性都有一個名稱，例如名稱、地址等。
- **屬性集(Attribute Set):**
屬性集是資料表中的資料行，也被稱為「值組(Tuple)」或「紀錄(Record)」。每個屬性集包含該資料行的值。例如，在訂單表中，每個屬性集代表一個訂單的資料。
- **值域(Domain):**
值域定義了每個資料格式的範圍。它確定了資料可以包含的合法值範圍。例如，在客戶表中，姓名屬性的值域可以被定義為 CHAR。
- **維度(Degree):**維度表示資料表中有多少個屬性。
- **主索引鍵(Primary Key):**
主鍵具有唯一性和最小性兩個特點，它的值確保每個屬性集都是唯一的。主鍵在資料表中標識唯一的紀錄。例如，在客戶表中，客戶 ID 可以被指定為主索引鍵。
- **外部索引鍵(Foreign Key):**
外部索引鍵的主索引鍵建立關聯，用於連結不同資料表之間的資料。例如，在訂單表中，我們可以使用客戶 ID 作為外部索引鍵，來關聯並檢索兩個資料表中的相關資料。

2.2.1 關聯性資料庫之特點

本段為介紹關聯性資料庫的特點，作為一個重要的資料管理系統，在現代商業和組織中扮演著關鍵角色。關聯性資料庫是一種基於關聯模型的資料庫，它以結構化查詢語言作為主要通訊介面，提供了強大而靈活的功能，以滿足不同應用場景的需求。

- **結構化查詢語言 (Structured Query Language, SQL):**
關聯性資料庫使用 SQL 作為主要的通訊介面，以執行資料庫操作。所有常見的關聯性資料庫引擎都支援標準 ANSI SQL，並提供引擎專屬的延伸功能。透過 SQL，可以進行資料的新增、更新、刪除以及擷取，並支援交易處理和分析應用程式的需求，以及資料庫管理的各種操作。使用 SQL 進行資料操作和檢索，使得查詢和檢索資料變得高效和方便。您可以使用複雜的查詢語句來篩選和排序資料，以滿足各種需求和應用場景。
- **跨平台和跨系統的相容性:**
關聯性資料庫是一個通用的資料管理系統，可以在不同的平台和操作系

統上運行。這意味著可以將資料庫部署在不同的環境中，並實現跨系統的資料共享和整合。且關聯性資料庫支援多個使用者同時訪問和使用資料庫，並提供存取控制機制來確保資料的安全性和隱私保護。您可以設定使用者權限和角色，限制其對資料庫的操作和存取。

- 資料完整性:

關聯性資料庫通過一套限制條件確保資料的完整性，這包括主索引鍵、外部索引鍵、非空白值限制、唯一性限制、預設值限制和檢查限制。這些完整性限制條件實施商業規則，確保資料的準確性和可靠性。此外，關聯性資料庫還提供觸發器(Triggers)功能，允許在特定操作發生時執行自訂程式碼，進一步確保資料的完整性。關聯性資料庫更具有良好的資料持久性，確保資料在故障或災難發生時能夠保存和恢復。資料庫管理系統通常提供備份和還原功能，以及故障恢復機制，以最大程度地減少資料損失和服務中斷。

- 交易(transaction)處理:

關聯性資料庫支援交易處理，它是由一系列操作組成的邏輯工作單位。交易原則是「全有或全無」，表示整個交易必須作為單一單位完成並寫入資料庫，否則交易的任何個別元件都不應通過。在關聯性資料庫的術語中，交易的結果包括「認可(Commit)」或「回滾(Rollback)」，每個交易以一致可靠的方式進行處理，與其他交易相互獨立，確保資料的正確性和一致性。

- ACID 特性:

所有關聯性資料庫交易必須符合 ACID 特性，即不可分割性、一致性、獨立性和持久性，以確保資料的完整性。

- (1) 不可分割性(Atomicity):

交易會作為一個最小單位進行，即不可分割、不可精簡。一個交易的操作要求執行全部成功，或者全部不成功，完全回滾到初始狀態，不會留下部分完成的狀態。

- (2) 一致性(Consistency):

要求交易在執行過程中和執行後保持資料庫的一致狀態。這意味著交易必須遵守所有定義的規則和約束條件，包括資料完整性規則、級聯操作和觸發器(Triggers)等。

- (3) 獨立性(Isolation):

確保在多個並發交易執行時，每個交易都獨立且不受其他交易的影響。以防止並發交易之間的相互干擾和衝突，保證每個交易都能獨立執行，不會受到交叉執行帶來的未預期結果。

- (4) 持久性(Durability):

確保在交易成功完成後，對資料庫做的變更將永久保存，不會因為系統故障或電源中斷而丟失。資料庫管理系統使用日誌(Log)和備份機制來實現持久性，以確保在任何情況下都能夠恢復資料庫到正確的狀態。

2.3 非關聯性資料庫

非關聯性資料庫，又稱 NoSQL，「最初表示為 Non-SQL，後也稱為 Not only SQL」，意指與關聯性資料庫不同的資料管理系統。與關聯性資料庫有所不同，關聯性資料庫是將資料正規化再以直行橫列的表格結構組織和存儲資料，而 NoSQL 資料庫的設計目的是快速處理大規模、非結構化或半結構化的資料，所以非關聯性資料庫採用了不同的資料模型，例如鍵值(Key-Value)、文件(Document)、列(Column)或圖形(Graph)等，這樣的模型更適用於彈性和多變的資料，不需要預先定義固定的資料架構。圖 18 為 NoSQL 以 JSON 模型儲存書籍資料的結構簡易範例：

- 在關聯性資料庫中，通常會正規化書籍記錄，並將記錄存放於個別表格中，同時以主索引鍵和外部索引鍵限制來定義關係。在此範例中，客戶表有客戶 ID、名稱、地址和信用卡種類，訂單表有客戶 ID 和訂單的相關資料。關聯性模型的設計是使資料庫強化資料庫表格間的參考完整性，並經過正規化以減少重複，而且針對儲存加以整體優化。
- 在此 NoSQL 資料庫範例中，客戶資料通常儲存為 JSON 文件。就每一本書，將客戶相關資料與其訂單資訊存成單一文件中的屬性。在非關聯性模型中，資料針對直覺開發和橫向擴充進行優化。

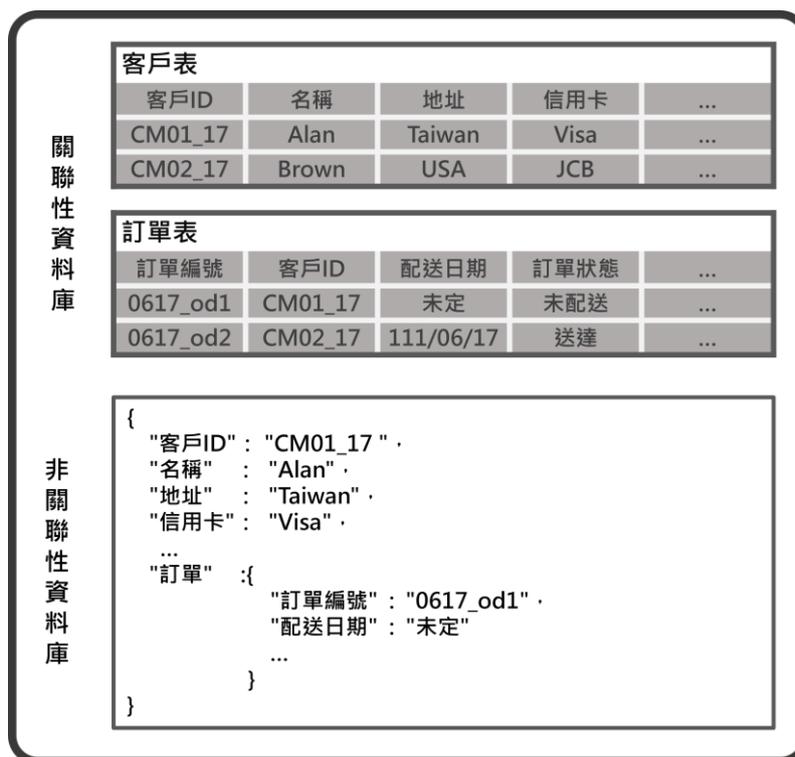


圖 18 NoSQL 以 JSON 模型儲存書籍資料的結構

2.3.1 非關聯性資料庫之種類

非關聯性資料庫是一類具有多種資料模型特點的資料庫系統，目的為解決傳統關聯性資料庫所面臨的某些限制和挑戰。傳統的關聯性資料庫以表格形式組織和存儲資料，在處理結構化數據方面表現出色，但對於處理非結構化和半

結構化數據，如文字文件、日記文件、媒體內容等，需要有更多的步驟及處理程序。為了應對大數據時代和資料多樣性的挑戰，NoSQL 資料庫則根據不同的應用需求開發出多種資料模型，以更靈活和有效地處理不同類型的數據。以下針對各種 NoSQL 資料庫進行介紹：

- 文件導向資料庫(Document Oriented Database):

一種以文件，如 JSON 或 XML，作為基本儲存單位的資料庫模型。它提供開發者可以依據在其應用程式中使用的語言進行資料庫的存取。它適合存儲和處理半結構化數據，例如 Web HTML、網頁資料和社交媒體貼文等。文件導向資料庫是基於集合 (Collection)的方式儲存資料，而每筆資料則被稱為文件 (Document)，每筆文件都可以具有不同的結構。文件導向資料庫著名的引擎有:MongoDB、CouchDB、RavenDB 等。



圖 19 文件導向資料庫之範例

- 圖形資料庫(Graph Oriented Database):

一種專注於處理具有豐富關係和連接性的數據的資料模型。圖形資料庫適合存儲和分析具有複雜關係的數據，如人際關係、社交網路、推薦系統和地理圖資系統，它們能夠輕鬆地表示實體之間的多對多關係、層次結構、循環關係等。圖形資料庫可以有效地表示和查詢實體之間的關係 (Relationship)，並支援複雜的圖形分析算法。它使用節點(Node)和邊的形式來表示實體和它們之間的關係，邊必定含開始節點、結束節點、類型和方向。邊可以用來描述父項與子項的關係、動作、所有權等。較著名的引擎為:Neo4j、Hyper GraphDB、FlockDB(Twitter 使用)。

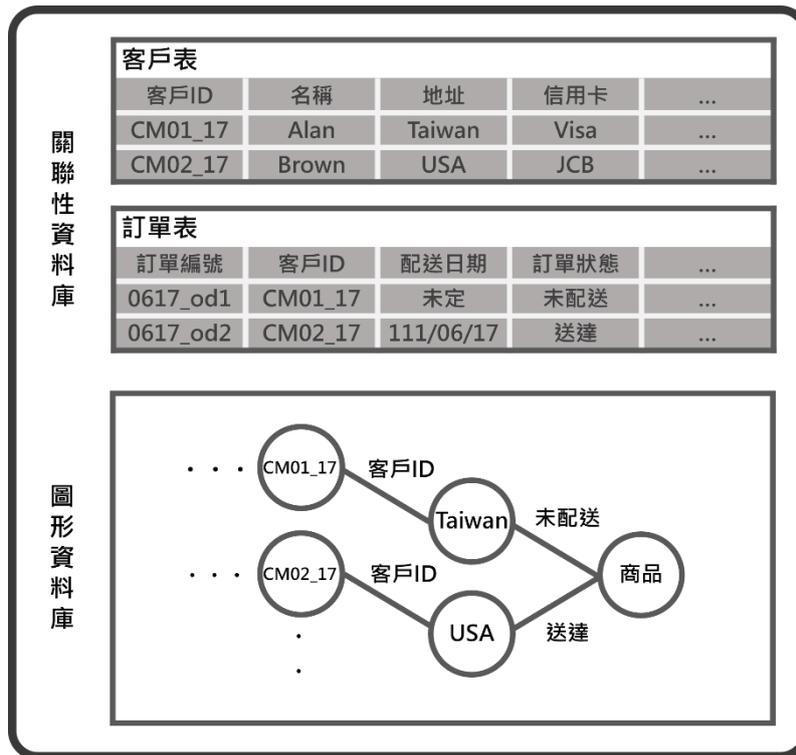


圖 20 圖形資料庫之範例

- 鍵值資料庫(Key-value Oriented Database):

一種簡單的鍵值存儲系統，每個鍵都對應著一個值，其中鍵和值不限種類，從簡單物件到複雜的複合物件都可以被定義，而我們可以通過這個鍵快速查找和存取相應的數據，最後這些鍵值以鍵值組集合的結構儲存於資料庫。它廣泛應用於緩存系統、對話儲存、分佈式儲存等場景。鍵值資料庫以桶(Bucket)的方式儲存資料，資料被稱為鍵值 (key-value)。較有名的引擎為: Redis(Flick 使用)、Memcached、Dynamo。

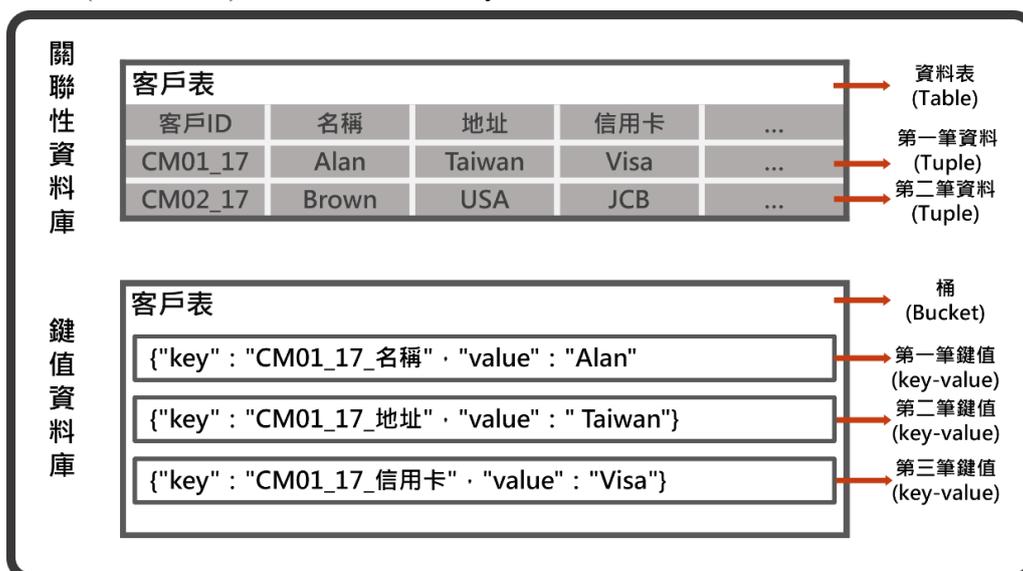


圖 21 鍵值資料庫之範例

2.3.2 非關聯性資料庫之特點

NoSQL 資料庫以其彈性、高擴充性、高效能和高功能性等特點，成為應對不同類型資料和變化需求的理想解決方案。開發人員可以根據具體的應用場景和資料需求，選擇最適合的 NoSQL 資料庫，以實現卓越的性能和靈活性。這些特點使得 NoSQL 資料庫在當今大量數據及訊息交換的環境中扮演著重要角色，推動著應用程式和系統的創新和發展。以下為 NoSQL 資料庫特點之詳細介紹：

- 彈性：

彈性結構使得資料模型更容易修改和擴充，適應半結構化和非結構化資料的處理需求。提供開發者可以更直接、貼近自己開發的應用程式語言儲存資料，不僅可以減少轉換的需求，也更易於理解開發架構，使得開發者能夠在不受固定模型限制的情況下，更自由地設計和調整資料模型，以滿足不同的應用場景。

- 擴展性：

藉由分散式叢集架構，NoSQL 資料庫能夠實現水平擴充，而無需與傳統關聯性資料庫一樣增加昂貴且複雜的伺服器進行垂直擴充。這使得資料庫能夠輕鬆處理大量數據和高流量的工作負載，確保系統的可靠性和性能表現。

- 效能：

每個 NoSQL 資料模型都專注於其架構及特定需求，並且使用比傳統關聯性資料庫更高效的存取模式。這使得 NoSQL 資料庫能夠快速處理大量的查詢和操作，無須再進行語言與需求間進行轉換，提供更高的效能和即時回應能力。此外，有些 NoSQL 資料庫更能充分運用雲端技術，提供零停機時間。

- 功能性：

不同類型的 NoSQL 資料庫，如文件導向、圖形或鍵值，都提供了針對特定資料模型的專門功能和 API。這些功能和 API 使得開發者能夠更輕鬆地操作和管理資料，並為開發者提供更靈活的應用空間。

2.4 關聯性與非關聯性資料庫之比較

非關聯性資料庫是近年來興起並廣泛運用的技術，它針對不同類型的數據提供了多樣化的資料模型，如文件導向、鍵值、圖形和列等。這種多樣性資料模型使得非關聯性資料庫能夠彈性處理半結構化和非結構化數據，為開發人員提供了更靈活的開發空間。另一方面，關聯性資料庫已經存在多年，相關技術也變得非常成熟和穩定。它以表格的結構化模型為基礎，適用於處理結構化數據。關聯性資料庫通常具有較嚴格的結構和規範，並且使用成熟的優化和索引技術，能夠處理複雜的查詢和關聯操作。兩者在不同面向和應用中各有所長。開發人員需要根據具體的應用需求和資料特性來選擇合適的資料庫技術，以實現最佳的性能、可靠性和開發效率。下表為兩種類型之資料庫比較。

表 4 關聯性與非關聯性之比較

	關聯性	非關聯性
資料模型	基於表格、列和欄組成的結構化模型。資料表格間的關係由結構描述嚴格定義，包括索引、關聯和其他資料庫元素，強調參考完整性。	多種資料模型，包括鍵值、文件、列式和圖形等。這些模型能夠儲存半結構化和非結構化的資料，根據應用程式的需求靈活地調整資料結構。
彈性	基於關聯的資料模型具有較嚴格的結構和規範，適用於結構化數據。	資料模型多樣且靈活，能處理半結構化和非結構化數據。
擴展性	透過增加硬體運算能力向上擴展，即垂直擴展。意指增加更多的處理器、記憶體和儲存容量。受到成本高昂，物理硬體的極限後，較受限制。	設計為分散式系統，可以透過增加節點和資源來擴展資料庫的容量和處理能力，即水平擴展，具有高擴展性。
效能	通常取決硬體運算能力，例如：磁碟子系統的速度和性能。	通常會受到基礎硬體叢集大小、網路延遲，以及呼叫應用程式的影響。
可用性	具有事務處理和 ACID 特性，提供資料完整性和數據驗證機制，可以保證資料的正確性和一致性。在處理交易性應用和需要高度一致性的場景時非常可靠。	通過放寬了 ACID 原則的限制，及分散式架構提供了更高的可用性和分區容錯性。在節點故障或網絡中斷等情況，仍然保持系統的正常運作。
查詢	成熟的技術和 SQL 索引，能處理複雜的查詢和關聯操作，並具有豐富的查詢功能和優化技術。	引擎多專注於特定的資料格式，使用專門的查詢語言，如 MongoDB 的查詢語言，以便處理特定資料模型的操作。
API	通常使用 SQL API，需透過 SQL 語言進行操作。	多樣化的 API，如 RESTful、NoSQL-specific 等。
開發成本	需要預先定義和規劃資料表結構，較難變更和調整。需要在高性能硬體上運行，以確保良好的效能和可靠性，因此專業的維護、管理和硬體設備，有較高的實施和運維成本。	資料庫的資料結構設計更加彈性和簡單、靈活。分散式的設計，可以橫向擴展，不需要昂貴的單一伺服器，降低了硬體和運維的成本。開源解決方案也很多，不用高額軟體授權。
運用負載	其 ACID 的交易性及一致性，適合於線上交易處理應用程式，如：銀行、商店。而 SQL 可支持處理複雜查詢，也適合用於線上分析處理應用程式。	其高效能、能處理大量數據、高擴展性並且具有靈活的資料模型等特點，適用於大數據、大型社交網絡、電子商務和即時分析等場景。

三、運用商用資訊模型轉換器分析配電系統管理運轉資訊傳輸協定

在本文前段有說明，CIM 是以 UML 進行結構之定義，再將所建立之數據模

型透過 XML 映射成一套標準的訊息結構進行傳輸。而在系統及設備之間的資料傳輸和交換，最為廣泛且常見的傳輸方式就是網路服務(Web Services)，Web Services 是指透過標準的 Web 協議向用戶提供服務，Web Services 的主要目標是促進不同系統間的互操作性，使其能夠輕鬆地共享資源和功能，透過 Web Services，應用程式可以通過標準的 Web 技術，如 HTTP、XML、SOAP 等，進行通訊資料交換。在本章節將通過多個面向是去探討兩個具有不同的特點和優勢的主流網路服務:REST 和 SOAP。

3.1 商用資訊模型轉換器

商用資訊模型是企業運作的重要工具，它能夠以結構化和系統化的方式描述企業的組織架構、資源配置、流程設計等，為了提高效率、降低成本、提供更優質的產品和服務，企業需要深入了解自身的運作方式。商用資訊模型作為一種描述企業業務的工具，可以幫助企業從整體和系統的角度來分析和優化業務流程。這些模型可以涵蓋多個領域和方面，以下是一些常見的商用資訊模型:

- 企業架構模型:描述企業的組織架構、職能和職責分配，以及各個部門之間的關係和溝通。
- 流程模型:描述企業的業務流程和工作流程，包括各個步驟、活動、角色和資源的訊息。
- 資料系統模型:描述企業的資料系統結構和組成，包括應用程式、資料庫、通訊接口等。
- 產品模型:描述企業的產品和服務，包括產品屬性、特徵和規格等。
- 決策支持模型:描述企業的決策制定過程和決策支持系統，包括大數據、資料分析、模型構建等

CIM 即涵蓋在商用資訊模型的範圍中，CIM 是將現代電力系統之結構，透過 UML 定義並映射為一套標準的訊息結構傳輸。目的是為了實現跨系統的無縫連接，減少設備之間複雜的資料轉換及更易於進行資料的理解及辨別。

而商用資訊模型轉換器是指將商用資訊模型轉換成其他形式或結構的過程或形式。這種轉換可以是從高層次的抽象模型轉換成低層次的實現模型，也可以是從一種表示形式轉換成另一種表示形式。在資訊模型轉換的過程中，需要注意資料的轉換必須準確且完整，任何不正確或缺失的資料都可能導致轉換結果誤差或丟失，進而影響到後續的分析和應用。因此，在轉換過程中，需要對資料進行驗證與校驗，或使用特定的工具或技術，如:網路服務、傳輸協定、文本結構等，以確保轉換的準確及有效。

上述有提到 CIM 主要是以 XML 映射為一套訊息結構進行傳輸，而資料傳輸較為常見的方式為 Web Services，本文將透過對兩種主流的 Web Services:REST、SOAP 進行比較，兩者在不同方面的特點，如:安全性、機制、速度等，分析何者最適合用於 CIM 之傳輸及交換。

3.2 簡式物件存取協定(Simple Object Access Protocol, SOAP)

簡式物件存取協定(Simple Object Access Protocol, SOAP)為基於 XML 格式所發展的一種輕量級通訊協定，主要應用在分散式環境下執行運算，為環境中計算資源提供一個用於交換之消息格式與協議標準，以實現整個系統的交互通訊及互操作性，加上 SOAP 本身獨立於語言、平台與傳輸，時常運用在全球資訊網服務(Web Service)中執行跨平台資料互換，傳輸過程可綁定 HTTP、SMTP、FTP 等通訊協定進行傳遞，但由於 HTTP 在網際網路上應用十分常見，並且早期大多數網路服務都是基於 Web 所發展，故 SOAP 使用 HTTP 通訊協定作為傳輸協定的案例相當普遍，目前版本為 SOAP 1.2，修訂版於 2007 年 4 月發布。

此外，SOAP 提供網絡服務描述語言(Web Service Description Language, WSDL)文件為客戶端描述該 Web 服務所發布之功能與信息細節，包含通訊上的訊息格式、輸入與出參數等操作集合，使客戶端能根據其信息更正確地理解該服務內容，以及如何進行請求服務的所有操作，而欲於網際網路上查詢特定 Web 服務，客戶端可透過統一描述、發現和集成(Universal Description, Discovery, and Integration, UDDI)對其功能、位置與元數據等相關資訊進行搜尋。

SOAP 作為一套正式且嚴謹之通信協定，提供了內置錯誤處理機制，在傳輸過程中發生錯誤時，如網絡中斷、傳輸超時或伺服器連線失敗等，SOAP 協定可以自動檢測到並重新傳送消息，直到它成功被傳送為止。這個過程是透明的，用戶不需要擔心丟失或重複傳輸的問題，一定程度地保障了訊息的可靠性，並且如果客戶端傳送之請求有問題，伺服器響應端會提供可用於解決問題之錯誤訊息，該回報訊息甚至提供了標準化的代碼，以便客戶後續在代碼中自行執行一些除錯動作。以下章節將針對 SOAP 中之各項名詞與結構進行更詳細之說明。

3.2.1 Web 服務描述語言

WSDL 是一套針對 Web 服務所發展之描述語言，使用 XML 作為基礎語言，用於描述 Web 服務所發布之功能與信息細節，內容定義如何操作其介面端點之相關文件導向或程序導向信息，如通訊上的訊息格式、傳輸參數與響應值等操作集合，而這些端點則代表 Web 服務可以被訪問的位置及使用方式，WSDL 之設計目的是讓使用者易於理解該 Web 之請求格式，並根據其信息執行訪問，而不必考慮該服務底層如何運行，同時透過標準化的描述方式，使得不同平台上的開發人員都可以使用相同形式來描述 Web 服務，除了可以提高服務的互操作性和可靠性外，亦能簡化了 Web 服務的開發與使用過程。

使用上 WSDL 通常與 SOAP 和 XML Schema 搭配使用，確保訪問該 Web 服務的相關程序有標準語法能夠讀取其 WSDL，而標準之 WSDL 文件內通常因具備該 Web 服務進行資料交換時之訪問位置、需要綁定之協議及信息格式、接受之公開操作與相關請求或響應訊息等端點信息，而根據 W3C 1.1 規範 WSDL 元素可由以下幾個部分組成[10]:

- 定義(definition):元素必須是所有 WSDL 文檔的根元素，它定義了 Web 服務中的容器名稱與命名空間。
- 類型(Types):定義 Web 服務中輸入及輸出參數之資料型別，以及如何映射至其他服務，WSDL 會使用 XSD 規範作為其預設定義之數據類型，並在發送方和接收方之間共享。
- 訊息(Messages):描述 Web 服務中可用於交換之輸入或輸出參數。
- 端口類型(PortTypes):描述如何將元素與端點組合成一個完整的請求/響應操作，操作過程可是單向或雙向。
- 綁定(Binding):定義 Web 服務所使用之傳輸協定及其格式細節。
- 端點位址(Port):定義端點綁定之位址信息，且一個端點僅能指定一個地址。
- 服務(Services):描述該 Web 服務所支持的端口，是所有位址之集合。

3.2.2 統一描述、發現和集成

UDDI 是一個基於 XML 且獨立於平台的開放框架標準，最初由 Microsoft、IBM、Ariba 等公司於 2000 年共同推動，定義了標準化程序來規範如何描述、發佈及探索 Web 服務，旨在幫助用戶端在分散式環境中自由地查找並使用所需之 Web 服務，簡化不同系統間應用程序的開發及部署過程，加速企業間交互合作與系統介接，以下為 UDDI 幾項主要特性：

- 使用 SOAP 作為其通訊協定對服務進行遠程調用，亦遵循 SOAP 訊息相應的 XML Schema 格式來描述所發布之 Web 服務的信息內容，有了制式結構能幫助用戶快速該服務之功能及使用方式，如服務的名稱、操作說明、接口與位址等資訊，即獲取該服務之 WSDL 描述文件，總的來說，UDDI、SOAP 及 WSDL 分別 Web 服務的三個不同層面上協同工作，達成伺服器到用戶端對服務使用上之一系列操作。
- 提供多組應用程式介面(Application Programming Interface, API)組件來實現企業對 Web 服務之註冊、發布及查詢等操作，其中包含全球通用之 UDDI 註冊中心(UDDI Registry)，供企業註冊並發布自家 Web 服務至公開之 UDDI 目錄列表，使用戶端根據不同條件來篩選所需之服務內容，或運用相關應用程序對 UDDI 目錄進行自動化訪問或介接。
- UDDI 本身並不負責執行與處理具體的服務請求和響應，僅僅提供一個標準化的註冊和發現機制，讓企業能夠註冊其服務並提供相關的描述信息，而用戶則可以通過 UDDI 進行查詢和發現所需的服務。

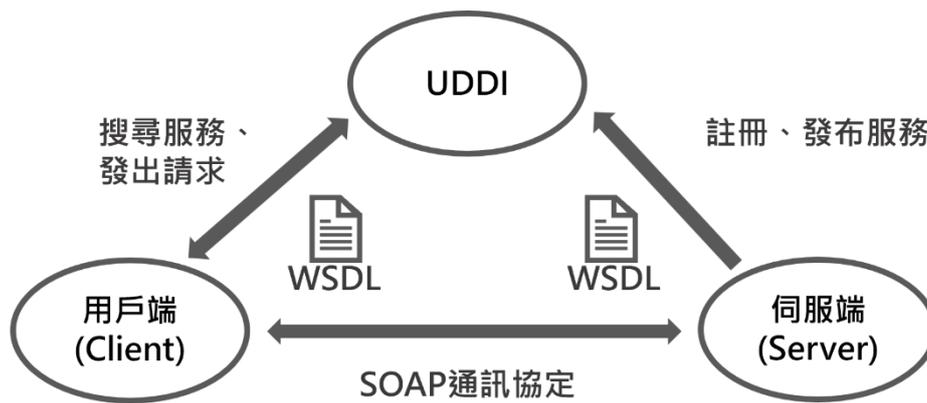


圖 22 UDDI 之運行機制

3.2.3 SOAP 之訊息結構

SOAP 訊息是透過 XML 所撰寫，由封套(Envelope)、標頭(Header)、主體(Body)及其他可選元素所組成[9]，各元素在訊息中會按照階層結構排列，形成了一種樹狀結構，並依其出現順序被依次處理，在語意表達上提供了靈活性與可擴展性，同時也讓應用程序能夠以結構化的方式解析和處理訊息，SOAP 訊息結構之內容說明如下表 5 所述，結構圖如圖 23 所示。

表 5 SOAP 訊息元素之結構說明

SOAP 訊息元素	功能說明
封套(Envelope)	SOAP 訊息的根元素，用於封裝整個訊息，定義了訊息的開始與結束，同時提供該訊息內元素引用之命名空間、訊息位址及協定版本等聲明。
標頭(Header)	一則 SOAP 訊息中可具備零或多個標頭，為可選的元素，用於描述訊息的額外附加資訊，如驗證信息等自定義元素，來實現更高階層的功能與可擴展性。
主體(Body)	主體包含訊息中的實際正文資料，可以使用任一格式表示，如 XML 文本、JSON、圖片或二進制數據等，通常包含用於執行操作之所需信息。
錯誤(Fault)	提供了有關 SOAP 訊息處理失敗時的詳細資訊，包含錯誤代碼與錯誤訊息等，如無效的輸入參數、伺服器內部錯誤及 SOAP 版本不匹配等，如表 4-2 所示。

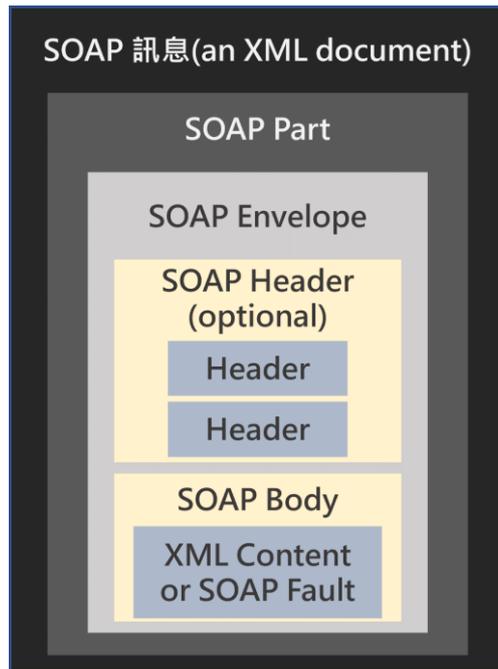


圖 23 SOAP 訊息結構[9]

表 6 SOAP 訊息之錯誤代碼

SOAP 錯誤代碼	代碼說明
VersionMismatch	故障節點出現無效元素訊息，命名空間及版本不匹配
MustUnderstand	SOAP 標頭中包含了一個或多個不能被處理之元素。
DataEncodingUnknown	指定的資料編碼方式未知或不支援。
Sender	無效的請求內容。
Receiver	無法解析訊息或上個節點處理訊息時出現異常。

3.3 表現層狀態轉換(Representational State Transfer, REST)

REST 是一種全球資訊網軟體架構風格，而不是一種標準或協定，目的是用於設計和建立分散式系統中的網路服務，以幫助不同軟體與程式便於在網路中互相傳遞資訊。REST 通常是基於 HTTP、URI、XML 以及超文本標記語言(Hyper Text Markup Language, HTML)這些常見且被廣泛應用的協定和標準。REST 的設計原則旨在實現輕量、簡單、可擴展、可靠且可互操作的系統。核心思想為資源表徵狀態的改變。

3.3.1 REST 的原則(constraints)

REST 在風格架構上定義了幾項架構約束，當設計的系統符合這些約束的時候，我們可稱這個系統為 RESTful。REST 提及的原則如下：

- 主從式架構 (Client-Server):

應用在實現時，需明確區分客戶端和伺服器的職責。客戶端為主動發出

請求並處理使用者介面，而伺服器負責處理請求、執行業務邏輯和發送回應。這種分離式設計使得系統的演化和擴展更加靈活，能夠支持多種不同的客戶端都向同一服務端發出請求。

- 無狀態 (Stateless):

伺服器在處理每個請求時不負責不保留客戶端的狀態，意味著每次客戶端發送請求都需要包含所有必需的資訊，使伺服器能夠理解和處理該請求。因此對話雙方的資訊由客戶端負責儲存維護，而不需要依賴之前的請求或狀態，這種特性使得伺服器能夠更好的擴展和適應負載增加的情況。

- 可緩存 (Cacheability) :

伺服器響應可被緩存的回應，以減輕伺服器的負載並提高效能。客戶端可自我決定是否緩存該請求響應的內容，而不需要再次向伺服器發送請求，這有助於減少網路流量和提高系統的效率。緩存機制可建立在伺服器、客戶端或是兩者並行。

- 統一接口 (Uniform Interface) :

要求系統使用一致且統一的接口來訪問和操作資源，主要目標是確保不同的系統和軟體能夠以一致的方式進行通訊和操作，並且能夠相互理解和交換資訊，簡化系統架構，減少耦合性。為了實現統一接口，REST 提供了以下幾個關鍵要素:

- (1)資源的標識(Identification of resources):

每個資源在 REST 架構中都有一個唯一的標識以便客戶端可以準確地定位和訪問該資源。這個標識符通常使用統一資源標識符(URI)來表示，並且應該是持久且穩定的，以確保長期可用性，如:在 HTTP 中的是 URL。

- (2)通過表示形式來操作資源(Manipulation of resources through representations):

客戶端通過發送請求並使用資源的表示形式來操作資源。這些表示形式可以是各種格式，如 JSON、XML 或其他自定義格式，根據應用的需求進行選擇。這種基於表示形式的設計允許客戶端和服務器之間的互操作性，使得不同的軟體/程式能夠在網絡中互相傳遞資訊。

- (3)訊息的自我描述性(Self-descriptive messages):

每個請求和響應的消息都應該包含足夠的信息，以使其自我描述。這些信息可以包括內容型態(例如 Content-Type)、編碼方式(例如 Content-Encoding)以及其他相關元數據。這樣的自我描述性使得消息能夠被獨立地理解和處理，減少了對外部元數據的依賴，從而提高了系統的互操作性。

- (4)用超媒體驅動應用狀態(Hypermedia as the engine of application state, HATEOAS):

當服務器在響應中提供超媒體連結，這些連結指示可用的操作和下一步的應用狀態。客戶端通過解析超媒體連結，動態地獲取並執行相應的操作，從而實現應用程序的狀態轉換。這種基於超媒體的設計使得系統更具

彈性和可擴展性，並且減少了客戶端對特定操作和路徑的依賴。

- 分層系統 (Layered system):

要求系統被分為多個層次，每個層次都具有特定的功能和責任，每個層次只能與其相鄰的層次進行通信，使得系統的結構更清晰。分層系統還提供了更好的安全性和可靠性。每個層次可以實施相應的安全措施，例如身份驗證和授權機制，以保護系統的資源和數據。根據此原則，當特定層次需要進行修改和擴展或是某個層次出現故障時，不會對其他層次產生影響，提高了系統的模組化和可維護性。

- 按需代碼(可選的) (Code on demand- optional):

一般而言，伺服器回覆的表現層是靜態形式的，但伺服器可以在需要時將可執行的程式碼傳送給客戶端，客戶端可以在執行應用程式時，動態地獲取、下載並執行伺服器提供的程式碼進行客製化功能，以擴展客戶端的功能。按需代碼是可選的，意味著客戶端可以選擇是否接受和執行伺服器端提供的程式碼。這是考慮到安全性和可移植性的需求。客戶端可以根據自身的安全政策和運行環境，選擇是否允許執行來自伺服器端的程式碼。同時，按需代碼的可選性確保客戶端和伺服器端的獨立性，使得它們可以獨立演化和部署。

3.3.2 RESTful API 的架構

前節已說明，REST 是一種設計架構風格，而符合 REST Constraints 的系統稱為 RESTful。可以用「REST 三角形」的模型描述 RESTful 架構中的三個核心概念，即資源(Resources)、行為(Actions)和表徵(Representations)。這三個概念相互結合，形成了 RESTful 架構中資源的定位、操作和表達的基礎。表 7 為簡單的 RESTful API 與一般的 API 比較。

- 資源:

系統中的實體或服務，可以是一個具體的物件、數據庫中的一個表格、網頁上的一個頁面或任何可識別和存取的內容。每個資源都有一個唯一的標識符(URI)，用於定位和訪問該資源。

- 行為:

對資源進行的操作或動作，通常使用 HTTP 協定中的方法(GET、POST、PUT、DELETE)來表示。例如，使用 GET 方法可以從資源中檢索資料，使用 POST 方法可以創建新的資源，使用 PUT 方法可以更新現有的資源，使用 DELETE 方法可以刪除資源。

- 表徵:

資源的表達形式，即資源的內容表示方式。在 RESTful 架構中，常見的表徵格式包括 JSON(JavaScript Object Notation)和 XML，使用者可以根據需要選擇最適合的表徵格式來處理和解析資源。

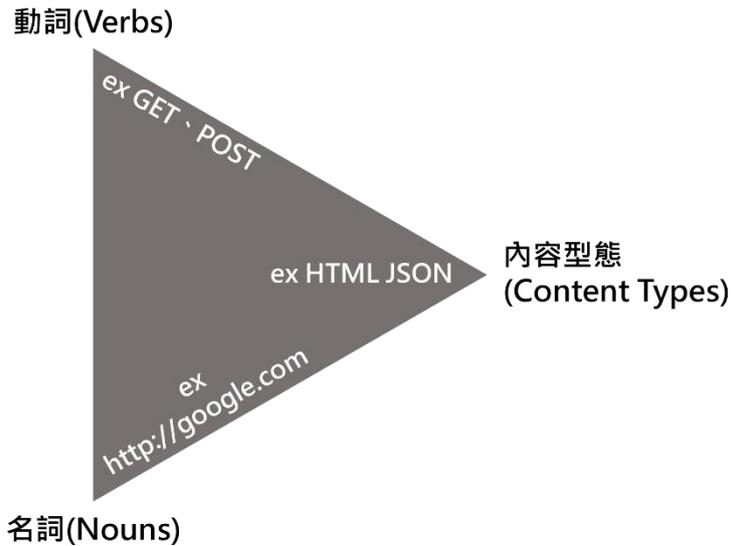


圖 24 REST 三角形

表 7 為 RESTful API 與一般 API 在 HTTP 上實現的比較

	Normal	RESTful
獲取所有使用者資料	POST:/getAllUsers	GET:/users
獲取特定使用者資料	POST:/getUser?id=1	GET:/users/1
新增特定使用者資料	POST:/createUser	POST:/user
更新特定使用者資料	POST:/updateUser?id=1	PUT:/users/1
刪除特定使用者資料	POST:/deleteUser?id=1	DELETE:/users/1

3.4 REST 與 SOAP 的比較

REST 與 SOAP 兩種資料傳輸方式為現今 WEB 服務的主流，各自用於不同的應用程式和系統之間的資訊交換。它們在設計理念、傳輸協定、資料格式、操作風格、訊息傳輸量、路由和安全性等方面存在著明顯的差異。下表為一個簡單的表格，比較了 SOAP 和 REST 在不同方面的差異：

表 8 SOAP 與 REST 之比較

	SOAP	REST
設計理念	基於服務導向架構(Service-Oriented Architecture, SOA)的一種協定，強調將應用程式設計為可重複使用的服務。	基於 ROA 的一種架構風格，著重於資源的標識和操作。
傳輸協定	可以使用多種協定進行訊息傳輸，如 HTTP、SMTP、TCP 等。	REST 主要基於 HTTP 協定，利用 HTTP 之 GET、POST、PUT 等方法進行資源的操作。
資料格式	使用 XML 作為主要的資料交換格式，並支援多種輔助格式，如 JSON、HTML 等。	支援多種資料格式，如 JSON、XML、HTML 等，具有更大的彈性，可以根據需求選擇最適合的資料格式。

	SOAP	REST
操作風格	使用 WSDL(Web Services Description Language)來描述應用程式的操作和服務，並使用 SOAP 消息來進行通訊。	使用 HTTP 方法進行資源的操作，如 GET 用於獲取資源，POST 用於新增資源，PUT 用於更新資源，DELETE 用於刪除資源等
訊息傳輸量	較大，需要在消息中包含大量的標記和描述資料，增加了訊息的大小。	傳輸量較小，使用輕量的資料格式，如 JSON，並避免了冗長的標記和描述，提升了訊息的效率。
安全性	支援多種安全性機制，如 XML 加密、數位簽章等，可以提供較高的安全性保護。	基於 HTTP 的安全性機制，如 SSL/TLS，並且可以使用額外的安全層(如 OAuth)來增強安全性。
路由和中介機制	支援複雜的路由和中介機制，可以在多層網路架構中傳遞訊息和處理中介服務。	簡單的 URI 進行路由，並避免了複雜的路由和中介機制，提高了系統的簡潔性
拓展性	相對較低，需要遵循嚴格的規範和標準，並依賴於複雜的描述和消息交換	較高，容易進行擴展和定制化，可以根據需求添加新的資源類型或操作方式，並且不影響現有的系統架構
互操作性	較高的互操作性，因為它擁有明確的標準和規範，各種平台和程式語言都可以實現 SOAP 的通訊。	取決於開發者的實作方式，需要確保遵從 REST 的原則和標準，以便各個系統能夠互相溝通和交換資源。
應用場景	適用於複雜的企業應用程式集成，需求較高的安全性和可靠性，如金融系統、醫療系統等。	適用於輕量且快速開發的網路應用程式，如移動應用程式的後端 API、Web 服務的開發等。

四、分析配電變壓器視覺化之關鍵參數

現今台灣的電力系統由發電系統、輸變電系統和配電系統構成，而配電系統中，變壓器擔任著電力轉換與聯絡的重要角色，用於電力的轉換即升壓和降壓，無論是發電廠、變電所還是最終用戶的家庭或工廠，都必須藉由各種不同類型之變壓器的操作，才能確保電力能順利地傳輸到目的地。台灣的配電變壓器形式多樣且廠牌型式眾多，根據設置地點則可以分為桿上變壓器、亭置式變壓器和地下設置變壓器，若依外型可分為分為一般型、改良型、密封型、直立式、小型、薄型、不鏽鋼橫式與沉水式等 8 類，實際應用上為一般型、改良型、密封型與直立式為主。根據不同的運行需求，設置最適合的變壓器。

隨著科技產業的進步帶來的負載增加及再生能源的積極發展造成的併網問題，都會對配電變壓器產生影響。本研究將透過 CIM 之便於跨系統間服務及透過 Web Services 即時收集數據等優點，將相關資料擷取並進行分析和處理，最後將配電變壓器的運行狀況進行視覺化的呈現，以達到便於進行配電設備維護及管理之目的。本節將探討變壓器的壽命以及預測其使用壽命的方法，並將其作為有效管理電力資源、提高系統運行效率以及便於維護運轉之視覺化關鍵參數。

4.1 影響配電變壓器壽命之因素

影響配電變壓器的壽命有許多因素，下列針對幾項重要因素做說明：

- 設置環境：

- (1) 鹽害：台灣四周環海，且氣候溫暖潮溼，變壓器外殼易受生鏽及鹽害的侵蝕。因為變壓器的外殼更換成本高，所以外殼的破損或鏽蝕成為決定變壓器壽命的重要因素，而變壓器外殼嚴重鏽蝕或破損一般出現在鹽害地區，鹽離子會附著在瓷套管上，造成絕緣電阻降低，產生表面放電，損害絕緣材料，從而造成危害。
- (2) 溫度：變壓器都有其適合的工作溫度，以達到最長的使用年限及效率，而當周遭環境溫度的影響，也會造成變壓器的工作溫度變化，高溫環境會使變壓器的內部電氣元件和絕緣材料受熱損壞，而低溫則可能導致材料的脆化，影響變壓器的壽命

- 雷擊、突波：

雷擊和突波是瞬時的高電壓事件，對配電變壓器的絕緣系統造成嚴重衝擊，會導致絕緣損傷、放電和短路故障，縮短變壓器的使用壽命。

- 不當操作和維護：

缺乏正確的操作和維護可能導致變壓器的故障和損壞。例如，忽略絕緣油的檢查和更換、不適當的冷卻系統操作等都可能導致變壓器的性能下降和壽命縮短。

- 絕緣：

絕緣在配電變壓器中扮演關鍵角色。良好的絕緣強度和材料選擇是確保變壓器安全可靠運行的重要因素。外界因素如上述所提到之設置環境、雷擊

等都會對絕緣造成損害，降低絕緣強度，縮短壽命。而其他因素如電壓波動、過載運行、不當操作和維護也會對絕緣產生負面影響。大部分的事故都來自於絕緣不良而引發的故障(短路)。

4.2 絕緣對配電變壓器壽命之影響

配電變壓器能否正常運轉，絕緣材料是關鍵要素之一。根據事故案例統計，絕多數變壓器的故障或損壞原因是來自絕緣材料的劣化所引起，大約占變壓器事故 85%以上。而變壓器開始運轉後，隨著時間推移或其他因素，如：溫度、環境、負載等，會導致其絕緣性能下降，進而影響變壓器的使用壽命，且絕緣材料的更換需耗費較高的經費。綜上所述，藉由評估和監測絕緣的劣化狀況，可以做為變壓器的使用年限之判定條件。

現行台灣地區之配電變壓器主要為油浸式以提供變壓器絕緣保護，而絕緣材料在油浸式變壓器中扮演關鍵的角色，用於隔離和保護導體和鐵心。變壓器使用之絕緣材料可分為兩種：液體的絕緣油及固體的，如壓紙板、絕緣紙、皺紋紙等。絕緣油具有卓越的絕緣性能和隔熱性能。而絕緣紙和壓紙板則用於包覆導體並提供絕緣支撐。絕緣油大概可分為一般絕緣油、高燃點絕緣油和乾式樹脂等三類，根據使用環境及需求選擇適合之絕緣油，如於桿上變壓器使用一般絕緣油型變壓器，亭置式配電變壓器則使用高燃點絕緣油型變壓器。

4.3 負載對於絕緣壽命影響分析

IEEE Std C57.91 針對油浸式變壓器進行了規範和壽命評估。該指南說明絕緣老化或劣化是一個由溫度、溼度和氧含量組合而成的時間函數。在現代的油浸式維護系統中，可以將濕度及含氧量的影響最大程度的降低，將溫度做為控制參數。由於大多數設備內部的溫度分布是不固定的，因此運轉在最高溫度下的組件通常會受到最大的損害，因此老化研究著重於考慮運行在最熱點溫度所產生的老化影響。本計畫運用該指南提供之溫度對於絕緣壽命的影響，作為變壓器使用年限及健康度之參考值。

根據實驗數據證明，絕緣劣化與時間、溫度之間的關係遵循一個以阿瑞尼斯方程式為基底所改編的方程式，其形式為：

$$\text{Per Unit Life} = A * e^{\left[\frac{B}{\theta_H + 273}\right]} \quad (4-1)[12]$$

其中

- θ_H 為繞組最熱點溫度，單位為 $^{\circ}\text{C}$
- A 為常數，選定 110°C 做為基準的每單位修改常數
- B 為常數，代表老化率
- e 為自然對數的底數

此公式將單位變壓器絕緣壽命與繞組最熱點溫度進行關聯。此公式適用於配電變壓器和電力變壓器，因為兩者都是使用相同的纖維素絕緣材料。在此公式中，將溫度作為影響變壓器壽命的主要變量並進行後續的分析。

- 變壓器絕緣老化加速因子

$$F_{AA} = e^{\left[\frac{15000}{110+273} - \frac{15000}{\theta_H+273}\right]} \quad (4-2)[12]$$

其中

F_{AA} 為老化加速因子
 θ_H 為繞組最熱點溫度，單位為 $^{\circ}\text{C}$

從 1957 年開始，就有許多科學家開始進行配電變壓器的功能壽命測試。多項功能測試和使用經驗的結果都表明，配電變壓器在 110°C 的運行溫度下，合理的使用壽命為 180000 小時約 20.55 年，並以此為標準並使用多年。根據公式(4-2)，當繞組最高溫度 θ_H 為 110°C 時， F_{AA} 為 1 標么值，當溫度高於 110°C 時，老化速率會超出正常值，絕緣老化會加速，而當溫度低於 110°C 時，老化速率降低到正常值以下，對絕緣老化幾乎沒有造成影響。

- 等效老化因子

變壓器之一日負載為時間函數，故等效老化因子為計算某一時間區間內配電變壓器運轉於不同溫度下之絕緣老化程度。

$$F_{EQA} = \frac{\sum_{n=1}^N F_{AA,n} \times \Delta t_n}{\sum_{n=1}^N \Delta t_n} \quad (4-3)[12]$$

其中

F_{EQA} 為總時間段的等效老化因子
 $F_{AA,n}$ 為時間區間內間隔的老化加速因子
 n 為時間區間內的間隔索引
 N 為時間間隔總數
 Δt_n 為時間區間，單位為小時

- 絕緣壽命損失百分比

計算某時間段內的絕緣壽命損失百分比，需要將公式(3-2)中求得之等效老化因子乘上以小時為單位的運轉時間段(t)來確定該時間段中損失的壽命小時數。再將消耗的壽命小時數除以總正常絕緣壽命(小時)並乘以 100，即可得出。計算公式如下：

$$\%Life_{lose} = \frac{F_{EQA} \times t \times 100}{Normal\ insulation\ life} \quad (4-4)[12]$$

Normal insulation life 為正常絕緣壽命基準值，IEEE 標準為運行在 110°C 時，標準絕緣壽命為 20.55 年

- 負載量與溫度的關係

最熱點溫度公式如下：

$$\theta_H = \theta_A + \Delta\theta_{TO} + \Delta\theta_H \quad (4-5)[12]$$

其中

θ_H 為繞組最熱點溫度，單位為 $^{\circ}\text{C}$
 θ_A 為研究之負載期間平均環境溫度，單位為 $^{\circ}\text{C}$
 $\Delta\theta_{TO}$ 為油面相對於環境之溫升(即油面溫度減環境溫度)，單位為 $^{\circ}\text{C}$
 $\Delta\theta_H$ 為繞組瞬間最熱點相對油面溫度之溫升

(1) $\Delta\theta_{TO}$ 為油面溫度相對於環境溫度之溫升，其公式為：

$$\Delta\theta_{TO} = \Delta\theta_{TO,rate} \times \left[\frac{K^2 \times R + 1}{R + 1} \right]^n \quad (4-6)[12]$$

其中

- R 為損失率，額定負載下總損失與無載損失之比值
- $\Delta\theta_{TO,rate}$ 為額定負載下油面溫度相對於環境溫度之溫升，單位為 $^{\circ}\text{C}$
- n 為經驗指數，這裡採用油浸自冷式變壓器，故為 0.8
- K 為負載率，每單位負載與額訂負載之比值

(2) $\Delta\theta_H$ 為繞組瞬間最熱點相對於油面溫度之溫升，其公式為：

$$\Delta\theta_H = \Delta\theta_{H,rate} \times K^{2n} \quad (4-7)[12]$$

其中

- $\Delta\theta_{H,rate}$ 為額定負載下繞組最熱點相對於油面溫度之溫升，單位為 $^{\circ}\text{C}$
- n 為經驗指數，這裡採用油浸自冷式變壓器，故為 0.8
- K 為負載率，每單位負載與額訂負載之比值

4.3.1 負載對於絕緣壽命影響計算範例[13]

- 上節為討論影響變壓器壽命的因素為溫度及負載，變壓器長時間運行在超過其額定負載的狀態下，會使變壓器的溫度升高，加速絕緣材料的老化和劣化，進而影響其壽命。評估變壓器的絕緣壽命時，需要以每小時負載為單位進行計算，以精確地反映溫度與負載變化對絕緣壽命之影響。國內目前無提供每小時平均負載資料，無法透過有效的計算得出負載與溫度對於變壓器絕緣壽命之影響。

故本節藉由 IEEE 提供的變壓器典型一日負載曲線來進行分析，並以一日平均負載 0.6 至 1.2 標么畫出六條曲線，如圖 25 所示。假設有一具容量為 25kVA 的變壓器，無載損失是 93.5W，於額定負載下運轉時總損失為 266.9W，額定負載下油面溫度相對於環境溫度之溫升 $\Delta\theta_{TO,rate}$ 為 37.5 度，額定負載下繞組最熱點相對於油面溫度之溫升 $\Delta\theta_{H,rate}$ 為 17 度，環境溫度 θ_A 採用台灣夏月日均溫 30 度。根據負載資料計算負載率 K 及損失率 R ，再將 $\Delta\theta_{TO,rate}$ 、 $\Delta\theta_{H,rate}$ 及參數 θ_A 帶入公式(4-5)~(4-7)後，便可得出各時間點最熱點溫度 θ_H 。將 θ_H 帶入公式(4-2)可計算出個時間點的老化加速因子 F_{AA} ，再由公式(4-3)可得一日等效老化因子 F_{EQA} ，最後將 F_{EQA} 帶入公式(4-4)即可算出變壓器運行一年後之絕緣壽命損失。再以標準壽命扣除運轉一年後之壽命損失，即可得變壓器絕緣殘餘壽命。

透過表 9 可以看出，當變壓器運轉於平均負載為 1 標么以下時，縱使在某些時段有運轉在超過 1 標么的情況，絕緣壽命幾乎沒有太大的減少；但當變壓器運轉於平均負載為 1 標么以上時，即過載情況下，雖然有些時間段為正常運轉，但隨著過載量越高與時間段越長，絕緣壽命的減少越加劇烈。

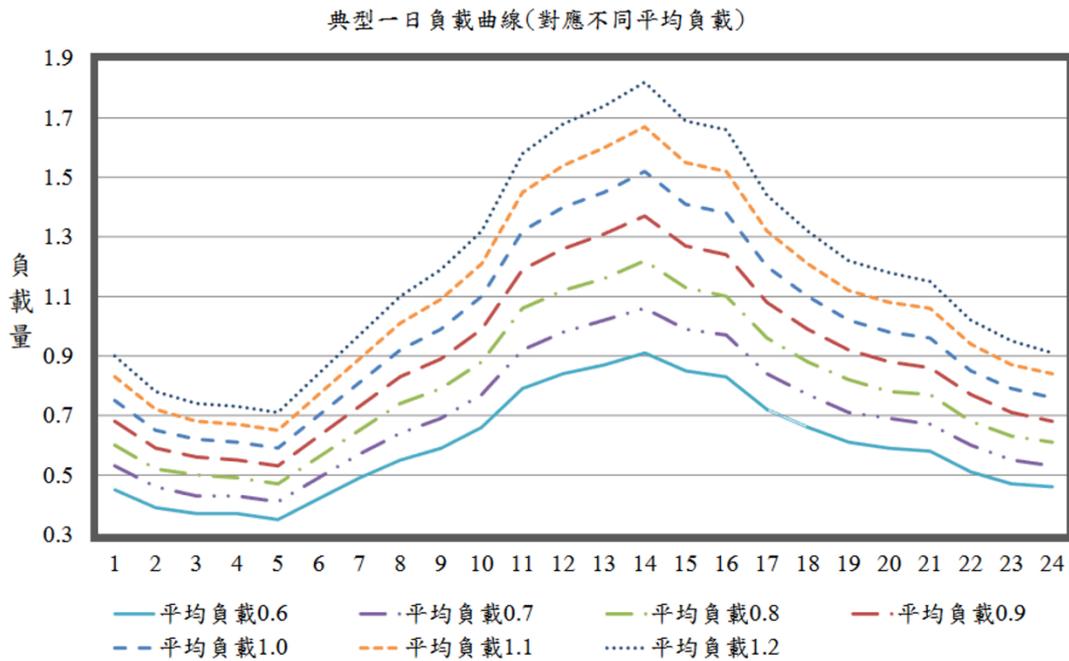


圖 25 變壓器典型一日負載曲線圖

表 9 變壓器絕緣殘餘壽命

平均負載量(標么)	殘餘壽命(年)
0.6	20.53
0.7	20.51
0.8	20.44
0.9	20.21
1.0	19.44
1.1	16.78
1.2	8.03

- 除了典型一日負載曲線外，IEEE 還提供了短時間緊急過載型一日負載曲線，如圖 26 所示。這種曲線代表了一天內出現極短時間內的極端高負載情況。當平均負載為 1 標么時，一日最大負載可達 1.7 標么。一樣以一日平均負載 0.6 至 1.2 標么畫出六條曲線，並進行計算及分析變壓器的絕緣殘餘壽命。

根據表 10 可以看出，當變壓器運轉於這種負載曲線時，其絕緣壽命損失相較於典型負載曲線更加顯著。例如運轉於短時間緊急過載型曲線時，當平均負載為 1 標么時，絕緣壽命損失為 1.94 年；而當平均負載為 1.1 標么時，絕緣壽命損失為 7.74 年。相對於典型一日負載曲線下的平均負載為 1 標么和 1.1 標么時，絕緣壽命損失分別為 1.11 年及 3.77 年，運轉於短時間緊急過載型曲線下的絕緣壽命損失明顯高出許多。根據這兩種結果也能進一步表明，在相同的平均負載下，變壓器的絕緣壽命損失存在著巨大的差異。因此，判斷變壓器的老化程度不能僅僅依賴於平均負載，而必須考慮超載時數和超載量的影響。

短時間緊急過載型負載曲線(對應不同平均負載)

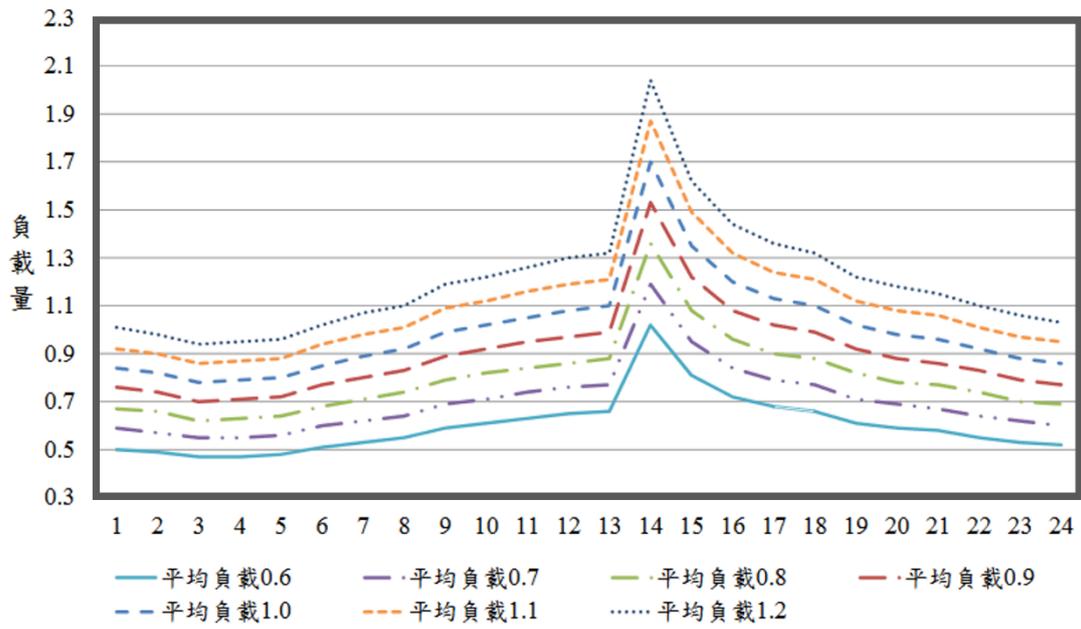


圖 26 變壓器典型一日負載曲線圖

表 10 變壓器絕緣殘餘壽命

平均負載量(標么)	殘餘壽命(年)
0.6	20.53
0.7	20.51
0.8	20.42
0.9	20.06
1.0	18.61
1.1	12.81
1.2	不到一年

4.4 諧波對於絕緣壽命影響分析

近年來，全球暖化日益嚴重，促使環保意識的興起，加上空氣汙染等健康議題，社會大眾對於燃煤和燃油等傳統發電方式逐漸產生排斥。雖然核能發電具有成本較為低廉且效率又高等優點，但台灣地處於環太平洋地震帶上，相同於地震帶上之日本的 311 福島事件，導致核能發電的安全問題被各界抱有餘慮及不信任，更何況台灣的核廢料存放問題以爭執多年，因此，以太陽能、風力發電和水力等再生能源替代傳統化石能源與核能發電，已成為我國政府明確的政策趨勢及發展目標。

由於再生能源所發出之電能多為直流電，必須透過電壓逆變器轉為交流電才可以併入輸配電網路或轉為家庭用，如圖 27 所示。由於是透過逆變器進行交直流轉換，逆變器所產生的諧波(Harmonic)可能會導致負載與設備產生多餘損耗及發熱，進而影響設備之安全性與系統運行效率。因此各國對於再生能源的併網訂定了對其諧波產生與抑制的相關規定，如：UL1741、IEEE 1547、台電的電力系統諧波管制暫行標準等。以下將針對諧波對於變壓器溫度上升進而導致絕緣壽命之影響進行討論。

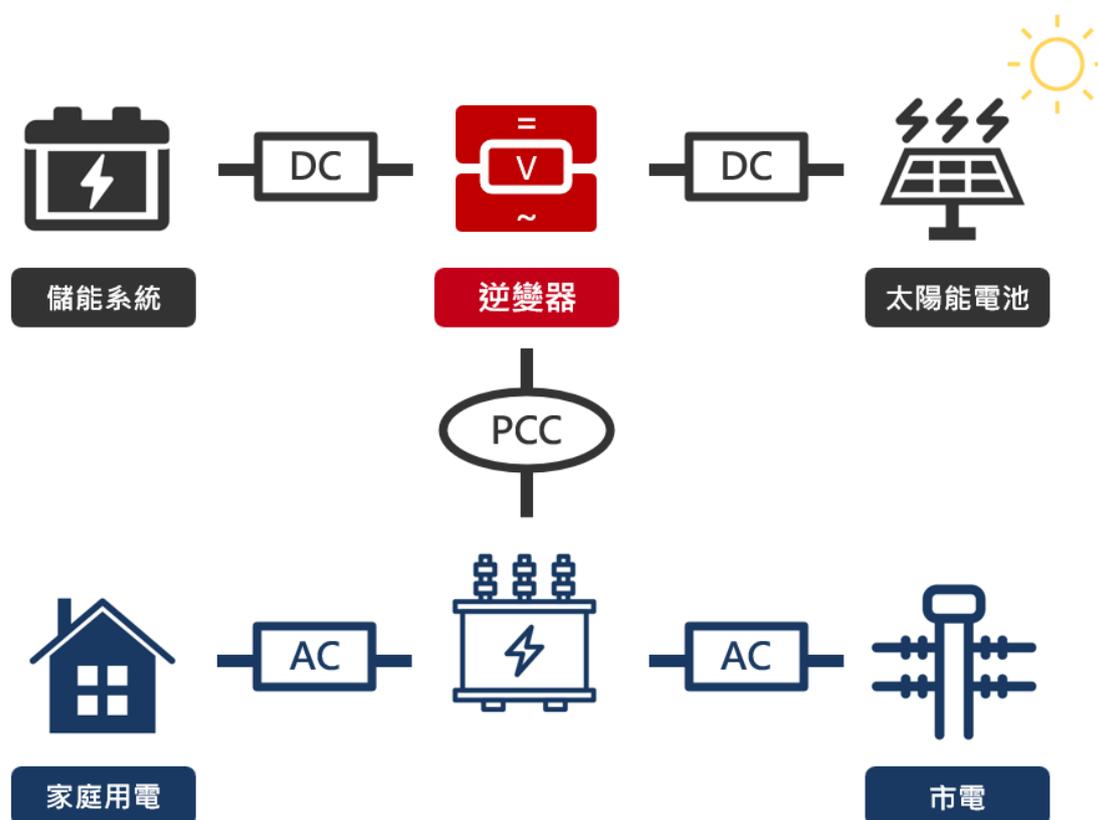


圖 27 太陽能發電系統架構圖

4.4.1 諧波相關定義與限制

由於過大的諧波會造成電壓電流失真與上升，進而導致設備與負載額外的損失與發熱風險。因此，在電力系統中諧波管理應為用戶、電力公司與營運商共同維護之責任，必須透過對電壓與電流之諧波進行限制，將電壓與電流失真限制在可接受之範圍。圖 28 為諧波造成電壓電流失真之示意圖。各項標準對於諧波之限制通常使用總諧波失真(Total Harmonic Distortion, THD)做為衡量單位，THD 值之定義如公式 4-8 所示。關於電壓之諧波限制，在 IEC 519 標準中針對於電壓之諧波限制進行了定義，如表 11 所示，該標準定義了在公共耦合點 (Point of Common Coupling, PCC) 中，電壓應在表 11 所描述之限制範圍中，PCC 之範例如圖 27 所示。

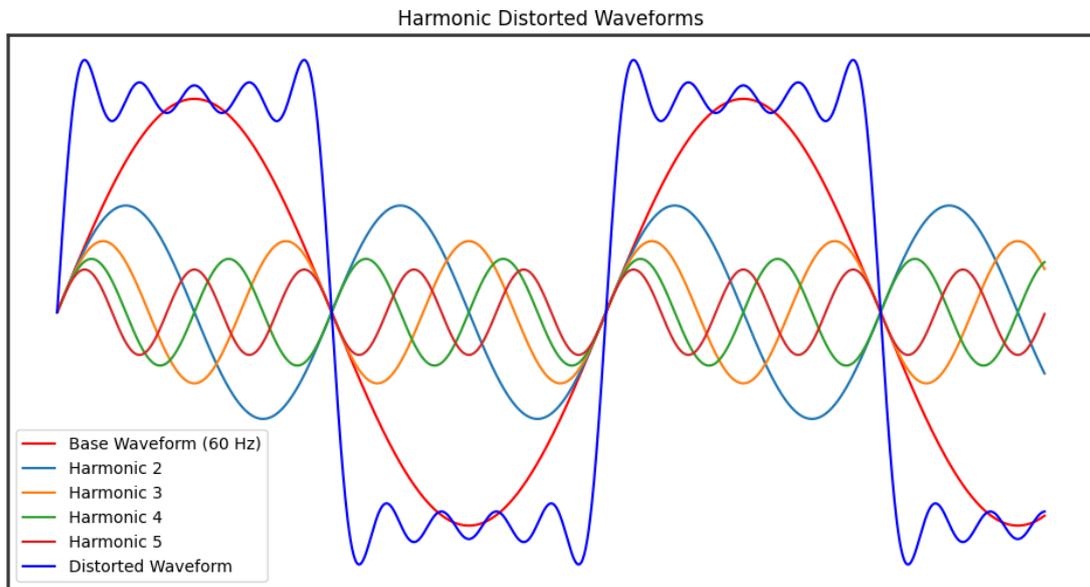


圖 28 諧波失真示意圖

$$THD_G = \frac{\sqrt{G_2^2 + G_3^2 + G_4^2 \dots + G_n^2}}{G_1} \quad (4-8)$$

其中 G_n 根據使用環境可替換為電壓 V 與電流 I ，是 n 次諧波的分量均方根值，而 $n=1$ 即為基本頻率。

表 11 電壓失真限制[16]

於 PCC 之匯流排電壓 V	單一諧波(%) $h \leq 50$	總諧波失真 THD(%)
$V \leq 1.0\text{kV}$	5.0	8.0
$1\text{ kV} \leq V \leq 69\text{kV}$	3.0	5.0
$69\text{ kV} \leq V \leq 161\text{kV}$	1.5	2.5
$161\text{ kV} \leq V$	1.0	1.5*

*在高壓系統中，總協波失真可以提升至 2.0%

對於電流之諧波限制則在 IEEE 1547 標準中進行了定義，針對諧波電流失真、單一諧波電流失真與總電流失真不該超出表 12 與表 13 之限制。需要注意

的是，關於該標準所定義之諧波限制僅針對連接到分散式電力資源(Distributed Energy Resource, DER)的電力系統。與 IEC 519 標準不同的是，IEEE 1547 對於電流諧波之衡量單位為總額定電流失真值(Total Rated Current Distortion, TRD)，其定義如公式 4-9 所示。

$$\%TRD = \frac{\sqrt{I_{rms}^2 - I_1^2}}{I_{rated}} \times 100\% \quad (4-9)[15]$$

其中

- I_1 於 PCC 點量測之基礎電流
- I_{rated} 額定電流
- I_{rms} 於 PCC 點量測之所有頻率分量之電流 rms 值

表 12 最大奇次諧波電流失真限制[15]

單一奇次 諧波 h	h < 11	11 ≤ h < 17	17 ≤ h < 23	23 ≤ h < 35	35 ≤ h < 50	TRD
百分比(%)	4.0	2.0	1.5	0.6	0.3	5.0

表 13 最大偶次諧波電壓失真限制[15]

單一奇次 諧波 h	h = 2	h = 4	h = 6	8 ≤ h < 50	TRD
百分比(%)	4.0	2.0	1.5	參考表 12	5.0

4.4.2 諧波對於絕緣壽命影響之計算

根據本計畫參考文獻[19]中的研究成果指出，對於諧波引起的電壓和電流波形失真，主要以電流諧波失真造成之影響較為顯著，具體結果如表 14 與表 15 所示。表 14 呈現了當總電壓諧波失真值上升時，無載損失的增加值。而表 15 則展示了當總電流諧波失真值上升時，對負載損失的影響。透過對兩項表格的比對，可以發現電流諧波失真的影響較為明顯。此外，根據 4-3 節所介紹的絕緣壽命老化計算方法進行計算，可以觀察到隨著總電流諧波失真值上升，絕緣壽命老化狀況越加劇烈，計算結果呈現於表 16。

表 14 每項總電壓諧波失真值的負載損耗[19]

THD _v (%)	0.0	3.0	5.0	8.0	10.0
無載損失上升(%)	0	0.19	0.15	0.33	0.51

表 15 每項總電流諧波失真值的負載損耗[19]

THD _i (%)	0.0	5.0	10.0	15.0	20.0	25.0
負載損失上升(%)	0	0.91	3.36	8.6	19.51	30.14

表 16 每項總電流諧波失真值造成的生命損失[19]

THD _i (%)	0.0	5.0	10.0	15.0	20.0	25.0
ΔL(%)	6.7	7.2	8.6	12.5	26.7	53.1

由於再生能源發電岸場通常為兆瓦(Megawatt, MW)規模，需要使用多台併連逆變器將發電之輸出並接再送入電網中。根據 IEC 61400 標準，其中闡述到當多台發電設備或是逆變器並聯運作時，可能會導致諧波電流的疊加進而產生超過限制之電壓或電流變動。該標準中提出之多台設備並聯之電流疊加公式為公式 4-10 所示。

$$I_{h\Sigma} = \sqrt{\sum_{i=1}^{Nwt} \left[\frac{I_{h,i}}{n_i} \right]^\beta} \quad (4-10)[17]$$

其中

N_{wt}	連結至 PCC 點之設備數量
$I_{h\Sigma}$	PCC 點之第 h 次諧波電流值
n_i	第 i 個設備的變壓器匝數比
$I_{h,i}$	第 i 個設備之第 h 次諧波電流值
β	指數， $h < 5 \beta = 1$ ； $5 \leq h \leq 10 \beta = 1.4$ ； $h > 10 \beta = 2$

綜上所述，本節將針對重點討論電流諧波失真的影響。以 4.3 節所提及之 IEEE Std C57.91 標準為基礎，多篇論文已針對諧波對於變壓器絕緣壽命之影響進行研究並提出其計算方式。計算方式與 IEEE Std C57.91 標準相同，都是根據電壓和電流造成的損失導致絕緣油溫度變化做為判斷變壓器絕緣壽命老化程度的主要衡量標準。較為不同的是，加入諧波之影響後，計算方式變得較為複雜，公式如下所示。

- 受諧波影響之頂部油溫 θ_{oil}

$$\frac{\frac{P_{LL-H} + 1}{P_{NL}}}{\frac{P_{LL-R} + 1}{P_{NL}}} \times [\Delta\theta_{O-R}]^{\frac{1}{n}} = \tau_O \frac{d\theta_{oil}}{dt} + [\theta_{oil} - \theta_A]^{\frac{1}{n}} \quad (4-11)[18]$$

其中

P_{LL-H}	因諧波增加的負載損耗，單位 W
P_{LL-R}	額定負載損耗，單位 W
P_{NL}	無載損耗，單位 W
P	額定電流下之 $I^2 \times R$ 損耗，單位 W
τ_O	相對於環境溫度之額定頂部油溫，單位 °C
θ_{oil}	頂部油溫時間常數
θ_A	環境溫度
n	非線性指數常數

- 因諧波增加的負載損耗 P_{LL-H}

$$P_{LL-H} = P \cdot \sum \left(\frac{I_h}{I_R}\right)^2 + P_{EC} \cdot \sum h^2 \left(\frac{I_h}{I_R}\right)^2 + P_{OSL} \cdot \sum h^{0.8} \left(\frac{I_h}{I_R}\right)^2 \quad (4-12)[18]$$

其中

P_{EC} 繞組之渦流損

P_{OSL} 雜散損

- 受諧波影響之繞組最熱點溫度 θ_H

$$\frac{\sum \left(\frac{I_h}{I_R}\right)^2 \cdot K_\theta + \sum h^2 \left(\frac{I_h}{I_R}\right)^2 \frac{P_{EC-R}}{K_\theta}}{1 + P_{EC-R}} \times [\Delta\theta_{H-R}]^{\frac{1}{m}} = \tau_H \frac{d\theta_H}{dt} + [\theta_H - \theta_{oil}]^{\frac{1}{m}} \quad (4-13)[18]$$

其中

P_{LL-H} 因諧波增加的負載損耗，單位 W

P_{LL-R} 額定負載損耗，單位 W

P_{NL} 無載損耗，單位 W

P 額定電流下之 $I^2 \times R$ 損耗，單位 W

τ_O 相對於環境溫度之額定頂部油溫，單位 °C

θ_{oil} 頂部油溫時間常數

θ_A 環境溫度

n 非線性指數常數

根據參考文獻中[18]中所給定之範例，圖 29 中，(a)為文獻給定之電流基波與諧波，(b)為環境溫度，將兩項數值代入公式 4-12 並計算出之頂部油溫 θ_{oil} 繪製於圖 30 之(a)中，可以看出在相同設備、溫度與損失的條件下，當總諧波失真增加時，頂部油溫也會隨之上升。將計算出之頂部油溫 θ_{oil} 帶入公式 4-13 計算後，便可以得出繞組最熱點 θ_H ，再配合前節所說明之公式 4-2~4-4，最終可求出老化因子 F_{AA} 與損失生命 $\%Life_{lose}$ 。從圖 30(b)可以看出，當總諧波失真上升時，繞組最熱點溫度也會隨之增加，而(c)則為當總諧波失真為 22%的老化因子 F_{AA} 曲線。由於台灣氣候較為炎熱，範例中之環境溫度並不適用於台灣之用電環境，因此本計畫基於台灣夏季氣溫重新進行計算。圖 31(a)為夏季氣溫一日曲線，圖 31(b)為重新計算出之頂部油溫曲線。圖 32 則為基於夏季溫度之繞組最熱點與老化因子曲線。可以發現由於氣候不同，各項數值皆有明顯上升。

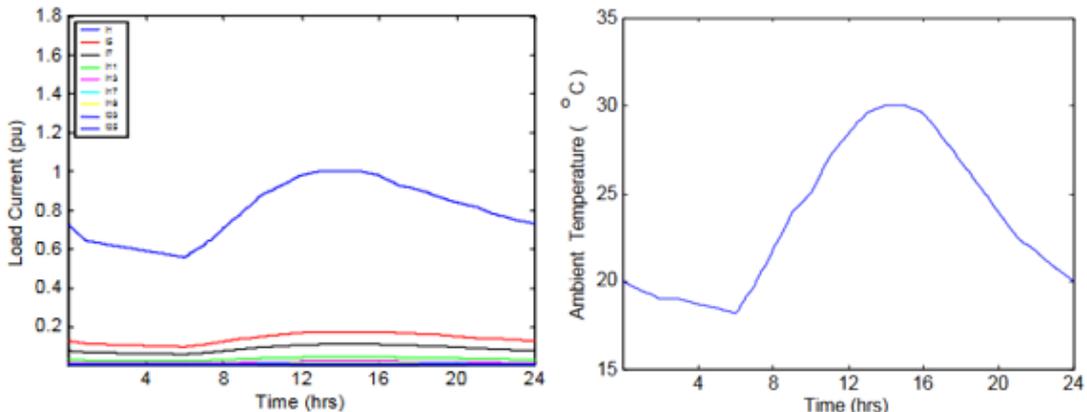


圖 29 (a)電流基波與協波 (b)環境溫度[18]

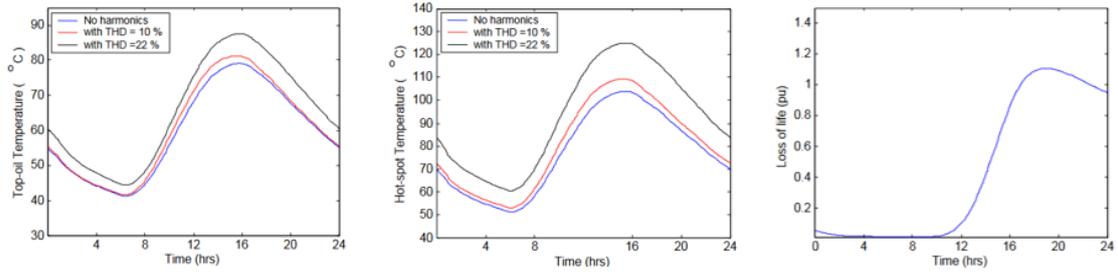


圖 30 (a)頂部油溫曲線 (b)繞組最熱點溫度曲線 (c)老化因子 F_{AA} 曲線[18]

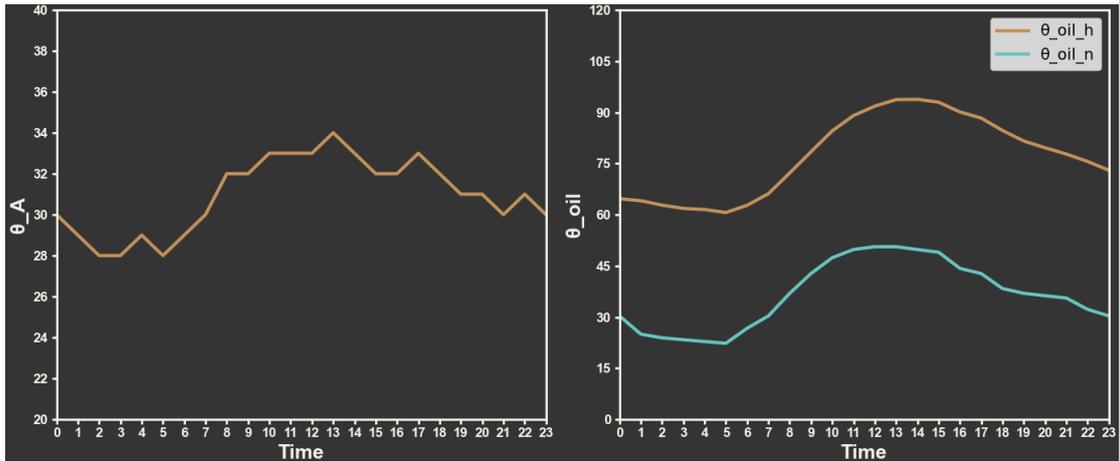


圖 31 (a)台灣夏季氣溫一日曲線 (b)頂部油溫曲線

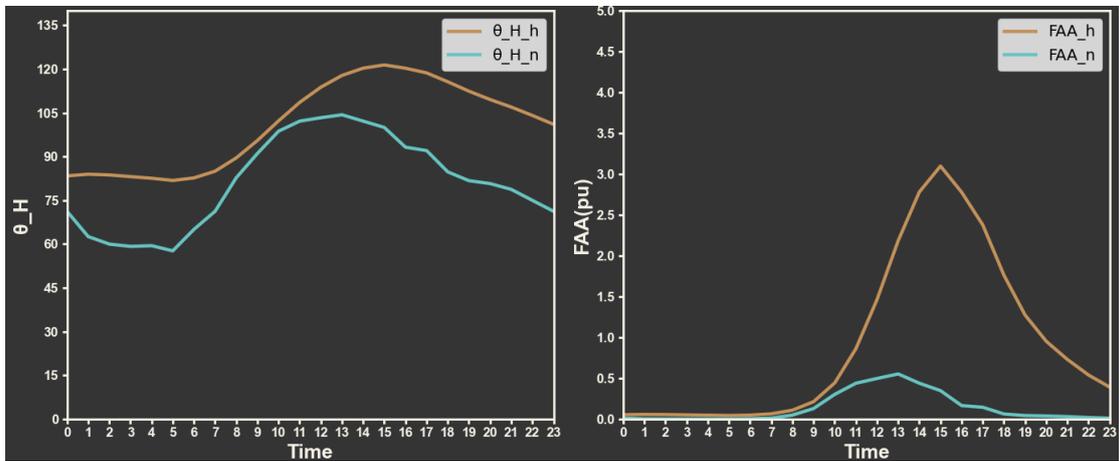


圖 32 (a)繞組最熱點曲線 (b)老化因子曲線

五、建構資訊模型傳輸系統架構並提出多系統間介接模式

本計畫建構之資訊模型傳輸系統架構如下圖 33 所示，首先會將 TTU 所量測之饋線電壓即時與歷史資料儲存於資料庫系統中，再藉由本計畫開發之存取模組進行所需要之資料存取與查詢，以達到有效的管理和存取。在存取模組中總共分為三個部分，資料庫介接 API、資料轉換程式與為 Web Service。資料庫介接 API 負責作為連結資料庫系統與存取模組的橋樑，是由資料庫系統維護商根據不同程式語言所提供之標準格式，該 API 提供一個標準的介面，以便系統組件可以進行資料的存取、查詢和更新，通常包含了資料庫的伺服器資訊、使用者帳號、密碼等資訊。資料轉換程式之功能為將資料庫提取出之原始資料(Raw Data)轉換為所需之 CIM XML 格式(Format)，此項工作需保證資料格式標準化與映射建立須符合所規定之 CIM XML 格式，並透過資料清理以去除不必要的訊息並發現處理缺失值，進一步確保資料品質。Web Service 為網路服務 API，其功能為實現跨系統間的通訊和資料交換。本論文以網路服務做為溝通管道，使得不同系統之間可以透過標準的通訊協定(例如前章所述之 REST、SOAP 等)進行 CIM XML 格式資料交換與傳輸。

透過實際測試與驗證，證實了此資訊模型傳輸系統的運作成功，為配電系統提供了穩固的資料處理和交換基礎，不僅體現在對 CIM XML 格式的準確轉換，也體現在 CIM XML 格式的有效傳輸和儲存。測試過程中，此架構可以成功透過資料庫介接 API 連結至資料庫系統，將儲存於資料庫系統中的實際量測饋線電壓資料透過資料轉換程式準確地轉換為所需要之 CIM XML 格式。最後，透過 Web Service 能夠有效地傳輸 CIM XML 格式資料。

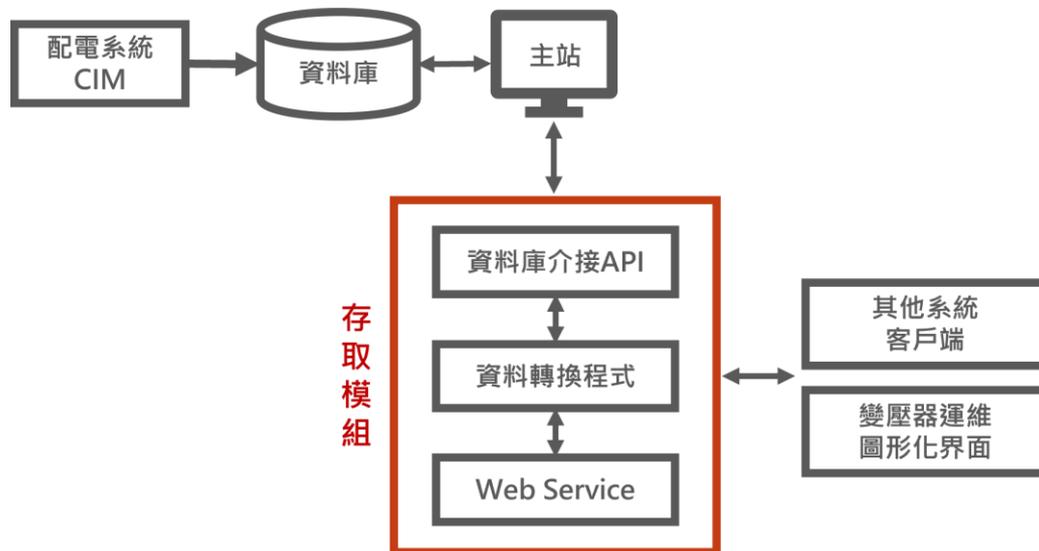


圖 33 配電系統資訊模型傳輸系統架構圖

5.1 傳輸系統之資料庫系統

在此架構中，資料庫系統為所有配電系統資料彙集與整合之目的地，因此在資料庫系統的選擇上顯得極為重要。在 2.3、2.4 節，已詳細介紹過資料庫系統的分類與優劣勢。本計畫將透過選擇 SQL 與 NoSQL 兩種資料庫系統進行饋線電壓即時與歷史資料儲存，並根據資料庫系統的性能、擴展性、安全性以及與系統的整合能力等因素，找出能高效處理資料、低延遲、有效儲存和檢索大量歷史數據的資料庫系統，以確保能夠即時地處理和更新饋線電壓的實時數據並支持分析和回顧配電系統的長期運行狀況。

5.1.1 Microsoft SQL Server

本計畫選擇之 SQL 測試資料庫為被廣泛使用的 Microsoft SQL Server。SQL Server 為微軟公司開發的關聯性資料庫系統，提供了全方位的資料管理、分析和業務智能解決方案，適用於各種企業和應用場景。下面為 Microsoft SQL Server 的一些主要特點和功能：

- 資料型別和結構:

SQL Server 支援多種資料型別，包括整數、浮點數、字串、日期等。更透過 Microsoft Transact-SQL 命令以達到允許開發人員使用 XML 和 JSON 等格式來存儲和查詢非結構化數據。這使得 SQL Server 適用於處理不同格式和結構的資料。

- 管理工具:

SQL Server 提供了豐富的工具和服務，如 Azure Data Studio、SQL Server Management Studio (SSMS)、SQL Server Data Tools (SSDT)、SQL Server Integration Services(SSIS)、SQL Server Analysis Services(SSAS)等，以幫助開發人員和管理員進行資料庫的設計、開發、部署、維護和監控，圖 34 為透過 SQL Server Management Studio 展現之 SQL Server 操作畫面。。

- 安全性和合規性：

透過驗證身分、作業授權、資料加密、資料遮罩與連線安全性等功能，保證了 SQL Server 具有高資料安全性並符合各種行業和法規的合規性要求。

- 雲端整合:

SQL Server 在 Microsoft Azure 雲端平台上的整合，提供了完全受管的 SQL Server 資料庫服務，使用戶能夠輕鬆部署、擴展和管理資料庫，此外，SQL Server 也支援混合雲部署，可以滿足不同規模和領域的資料庫應用需求。

- 社群支援和資源:

SQL Server 擁有龐大的社群和支援資源。Microsoft 官方網站提供了詳盡的支援文件、教學和培訓資源。此外，社群論壇和網站也是獲得協助和知識的寶貴資源。

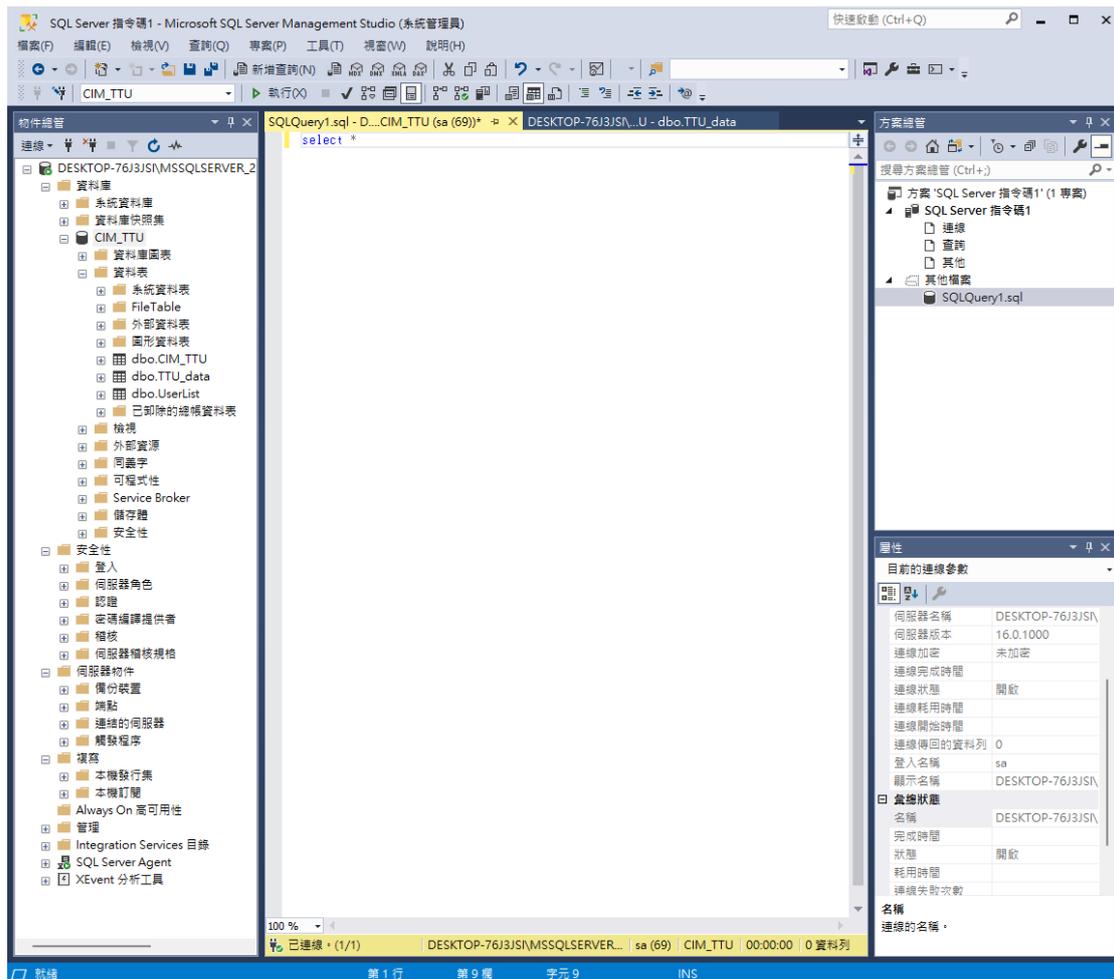


圖 34 SQL Server 操作畫面

5.1.2 BaseX

非關聯性資料庫則是選擇使用 BaseX 資料庫。BaseX 是一開源的原生 XML 資料庫管理系統和 XML 編輯器，專注於有效管理和查詢 XML 資料。其內置了 XQuery 查詢語言，支援複雜的 XML 查詢和轉換操作，使得對 XML 的查詢和分析變得快速而方便。且 BaseX 為一輕量化的資料庫系統，具有低資源需求、快速啟閉、模組化架構等特點，適合用於嵌入式應用場景，如嵌入式系統或 IoT 設備。BaseX 提供了一個簡單而直觀的介面，對開發者來說非常友好，使得學習和使用 BaseX 變得簡單，同時減少了開發和測試的時間，如圖 35 所示，圖 35 為 BaseX 資料庫圖形化介面。因此本計畫選擇其作為儲存 CIM XML 格式之非關聯性資料庫。

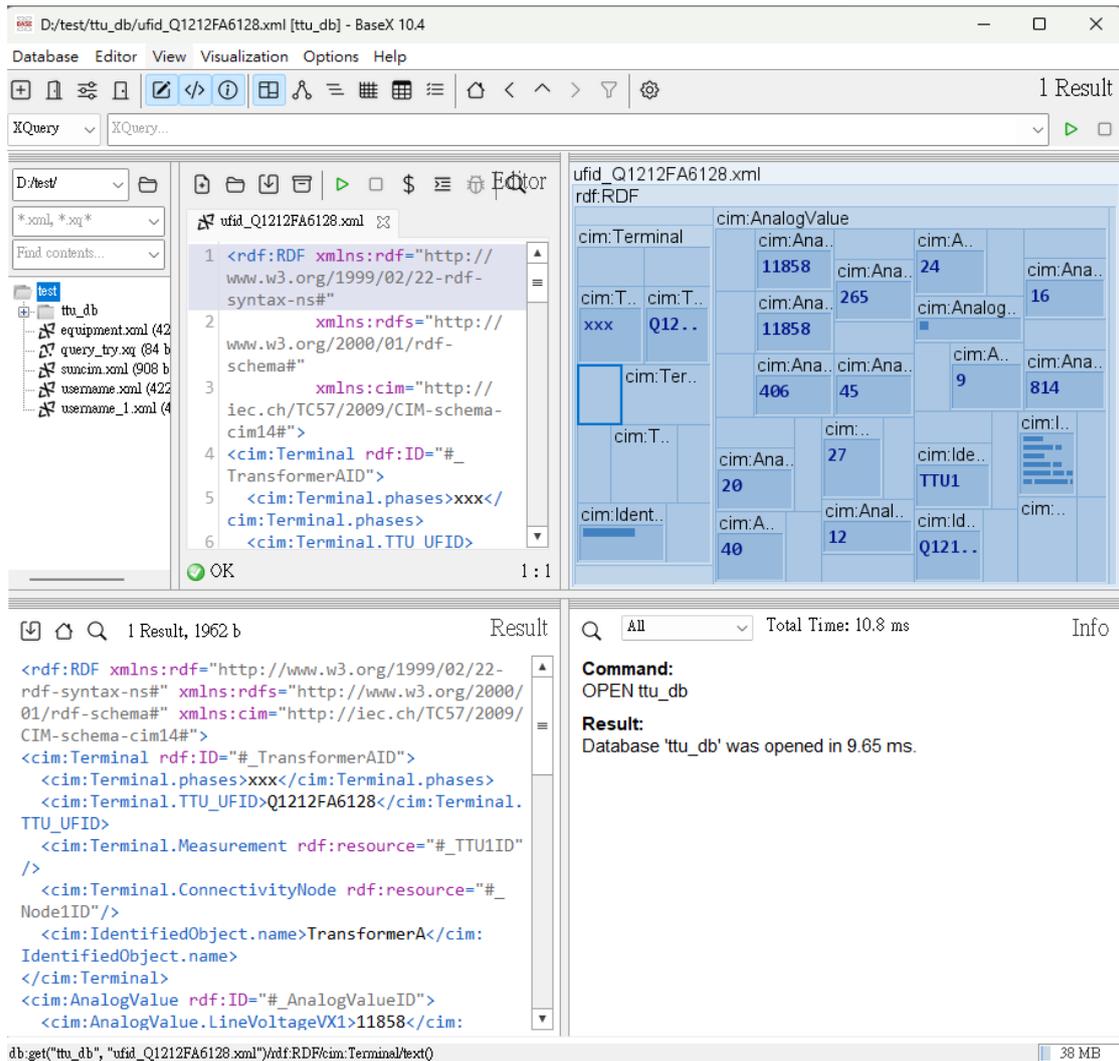


圖 35 BaseX 資料庫圖形化介面

5.2 傳輸系統之存取模組

在本架構中，存取模組扮演著資料轉換、資料過濾與清理和處理其他系統之請求的關鍵角色。這個模組在效能、維護難易度、擴展性等方面的表現上至關重要。在效能上，存取模組應該被設計為能高效運行，以確保快速的資料轉換和系統請求處理。這包括最佳化查詢、快速資料存取，以及有效率的資源利用。維護難易度方面，存取模組的程式碼應該具有清晰的結構，以便日後進行維護，且使用者能夠快速理解程式碼，在進行修改或添加新功能時能夠更加輕鬆快速。而存取模組應該具有良好的擴展性，以應對未來的系統需求變化。這包括能夠輕鬆整合新的資料庫或系統、支援新的資料轉換需求，並在不影響整體性能的情況下擴展。圖 36 為存取模組之示意圖。

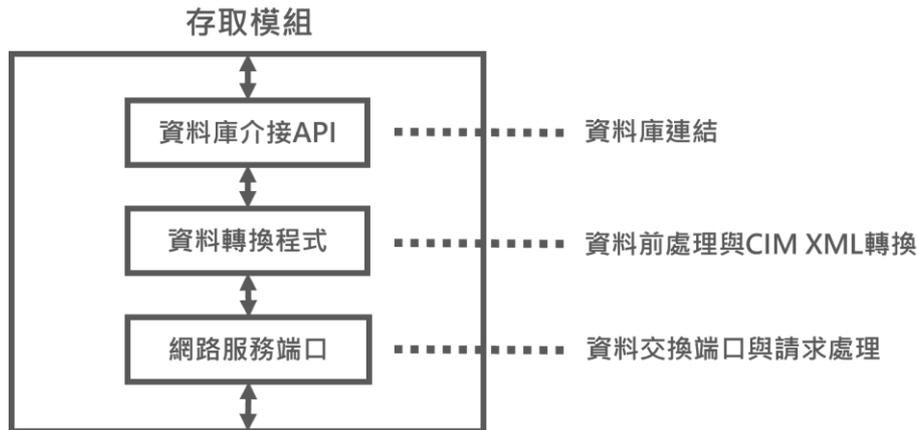


圖 36 存取模組示意圖

綜上所述，在存取模組的建立上本計畫將使用 Python 語言進行開發。Python 為一種被廣泛使用的高階程式語言，以其簡潔、易讀的語法聞名，這種語法設計使得開發者能夠以更直觀的方式表達思想，不僅使初學者容易上手，同時也提供了更好的可維護性，有助於減少錯誤和提高生產力。且 Python 擁有龐大而強大的函式庫，其包含了各式各樣的模組和套件，從資料處理、網路通訊，到資料庫連接等等，都能夠方便地進行操作，節省開發者的時間和精力。由於其易學性和廣泛應用性，Python 擁有龐大的開發者社群。這個社群的存在為開發者提供了豐富的資源和支援，同時也促進了開源項目的蓬勃發展。而 Python 的物件導向特性使其更具結構化，有助於開發者更好地組織和管理程式碼。種種的優點，造就 Python 適用在各種應用場景中，如網頁開發、資料科學、機器學習、人工智慧、自動化流程，以及資料分析等多個領域。

5.2.1 資料庫介接 API

隨著人工智慧與大數據技術的興起與迅速發展，如圖 37 所示，Python 在 TIOBE 已連續幾年名列前茅，TIOBE 指數是一項用於界定程式語言流行度的指標，其流行度是以該程式語言在搜尋引擎之搜尋結果與數量為基礎進行計算，該指標反映了該程式語言在開發者社群中的廣泛應用和受歡迎程度。而人工智慧與大數據等技術，都需要以大量的資料集為基礎，對於資料庫的整合與應用必定為不可或缺的一環。因此，各種類型的資料庫都會根據 Python 語言開發其介接 API，以提供使用者能快速的對資料庫進行連線並進行資料的操作。本計畫所使用之 SQL Server 與 BaseX 也不例外，其連結資料庫之介接 API 分別為 pyodbc 與 BaseXClient，透過輸入資料庫所訂定之伺服器位置、使用者帳號密碼、資料庫名稱等資訊便可連結至該資料庫進行資料的管理與操作。

Oct 2023	Oct 2022	Change	Programming Language	Ratings	Change
1	1		Python	14.82%	-2.25%
2	2		C	12.08%	-3.13%
3	4	▲	C++	10.67%	+0.74%
4	3	▼	Java	8.92%	-3.92%
5	5		C#	7.71%	+3.29%
6	7	▲	JavaScript	2.91%	+0.17%
7	6	▼	Visual Basic	2.13%	-1.82%
8	9	▲	PHP	1.90%	-0.14%
9	10	▲	SQL	1.78%	+0.00%
10	8	▼	Assembly language	1.64%	-0.75%

圖 37 Python 於 TIOBE 之排名

5.2.2 資料轉換程式

關聯性資料庫在資料存儲的過程是以資料表的形式呈現，意旨 SQL 資料庫無法直接儲存 CIM XML 格式。因此，在對 SQL 資料庫進行 CIM XML 格式資料查詢和管理時，需要先進行資料前處理。本計畫透過 CIM XML 格式的 UML 結構模型拆解 CIM XML 格式中的屬性參數，並將拆解後的屬性映射至相應資料表的結構中，確保每個類別、屬性和關聯都有正確的欄位和鍵值，如此便可將 CIM XML 格式之資料轉換成關聯式資料表的形式進行查詢和管理。

而資料轉換程式之主要功能為將儲存於資料表中之配電系統資料取出並透過 UML 結構模型將資料庫內之屬性參數重新映射為所需之 CIM XML 格式，保證資料的一致性和符合標準。從資料庫取出資料之過程中，不同資料庫之傳輸格式皆不固定，故資料轉換程式需要透過資料前處理，將多餘的資料剔除或進行轉換，以確保最終轉換出的 CIM XML 格式資料是整潔而合乎預期的。圖 38 為資料轉換程式之工作示意圖。

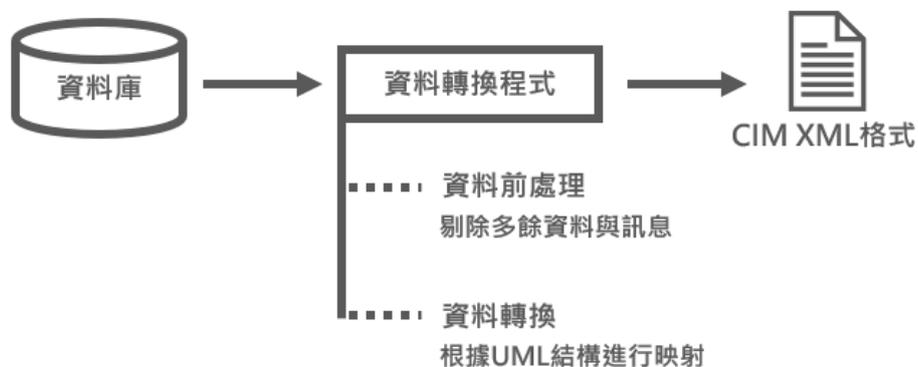


圖 38 資料轉換程式工作示意圖

5.2.3 Web Service

網路服務通過提供一個統一的通訊協定，讓不同系統間能進行互動，使得不同系統間不論其運行平台或程式語言為何，都能夠進行溝通和交換資料，為不同系統間的整合和協作變得更加簡單可行。這樣的統一機制有助於確保資料在系統之間的正確傳輸，同時也降低了系統間的相依性。而網路服務支援不同的通訊協定，其中便包括前面章節所討論的 REST 和 SOAP。因此，為了達到不同配電系統與設備間通訊與資料交換之目的，本計畫選擇使用網路服務做為系統間的溝通管道。透過網路服務，為不同系統與設備間的 CIM XML 格式資料提供了一個靈活、可擴展和標準化的資料交換方案，提高配電系統間的互通性，圖 39 為網路服務之概念圖。

在網路服務的設計上，RESTful 通常使用 HTTP 通訊協定和 URL 資源標示符，強調資源的表述和資源之間的狀態轉換，不需要像 SOAP 嚴格規定消息格式，使得開發者容易理解和使用，且簡潔的設計使得系統之間的通訊變得直觀和簡單。相對地，SOAP 為基於 XML 格式所發展的通訊協定，提供了一種更為結構化的通訊方式。結構化的設計使得 SOAP 能夠支援較複雜的消息結構和協定，同時提供了較豐富的錯誤處理機制。而如何針對 CIM XML 格式的傳輸與資料庫之存取去設計網路服務之功能為值得研究之目標。

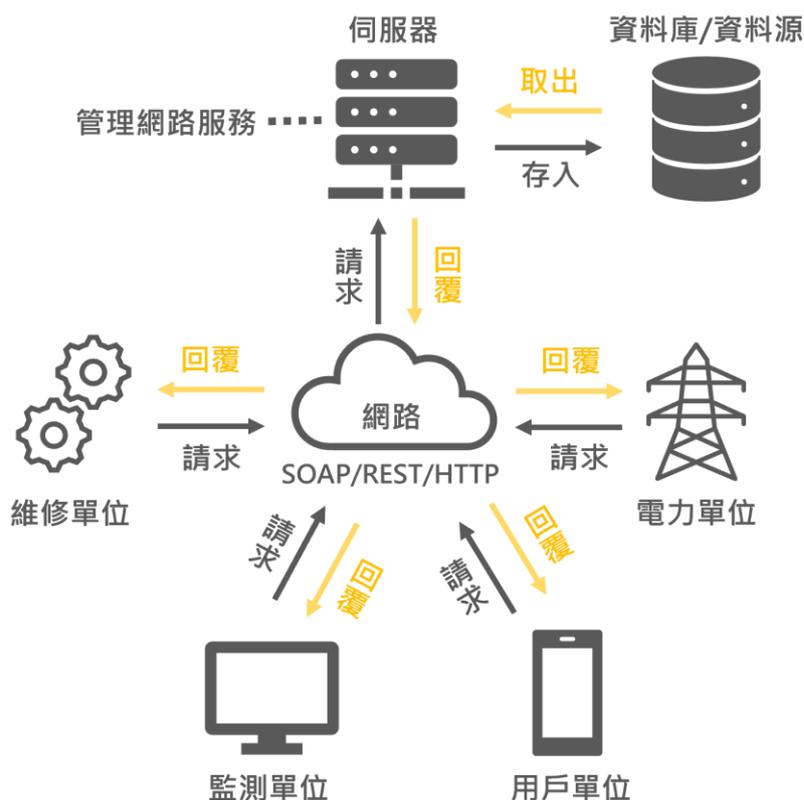


圖 39 網路服務概念圖

5.3 多系統介接模式

在第二章有提到，CIM 為基於 UML 定義的電力系統模型，再將其透過 XML，映射成一套標準訊息結構進行傳輸，實現不同的系統和應用程式間可以共享和交換電力系統資訊。且 IEC 也已透過相關標準對 CIM XML 標籤結構及映射格式進行定義。在 1.1.2 節中也已闡述，在 UML 語法中，如何透過套件將模型之複雜度進行簡化並根據需求進行調整與定義。而一個完整之 CIM 應用模型則由數個套件所建構而成，對於 CIM 之套件規範，IEC 已定義其細節並記錄於相關規範內。依據 IEC 所規範之套件進行延伸與套用，便可輕鬆定義配電系統不同設備之 CIM 結構，達到多系統間之統一資訊傳輸格式。

在參考文獻中，闡述了該國曾分布一項備選方案，IEC CIM 將被選定並描述為首選之標準資料交換模型。IEC CIM 具有不依賴特定技術，也不與特定商業模式相關聯之特性。且 IEC CIM 所描述的為實體電網中之內容，可做為功能組件間資料交換的標準訊息與資料模型。除此之外，IEC CIM 更可以透過建立複雜之模型來描述系統工程圖並做為其電力系統應用之可永久儲存資料模型。綜上所述，該論文提出一建議，將所有系統之介面逐漸更改為統一使用 IEC CIM，使其成為系統之新舊項目間強制性的標準，圖 40 為該文所建議之逐漸更換系統支持 IEC CIM 之示意圖。採用統一使用 IEC CIM 之優點：

- 採用國際通用標準，可以支援新的工具與功能。
- 結合國家之智慧電網政策，提供跨系統間的查詢與資料交換。
- 減少了在多系統間不同資料格式儲存之需要，提高了業務流程與效率。
- 允許關鍵資料的識別與唯一版本的訊息源與真相。

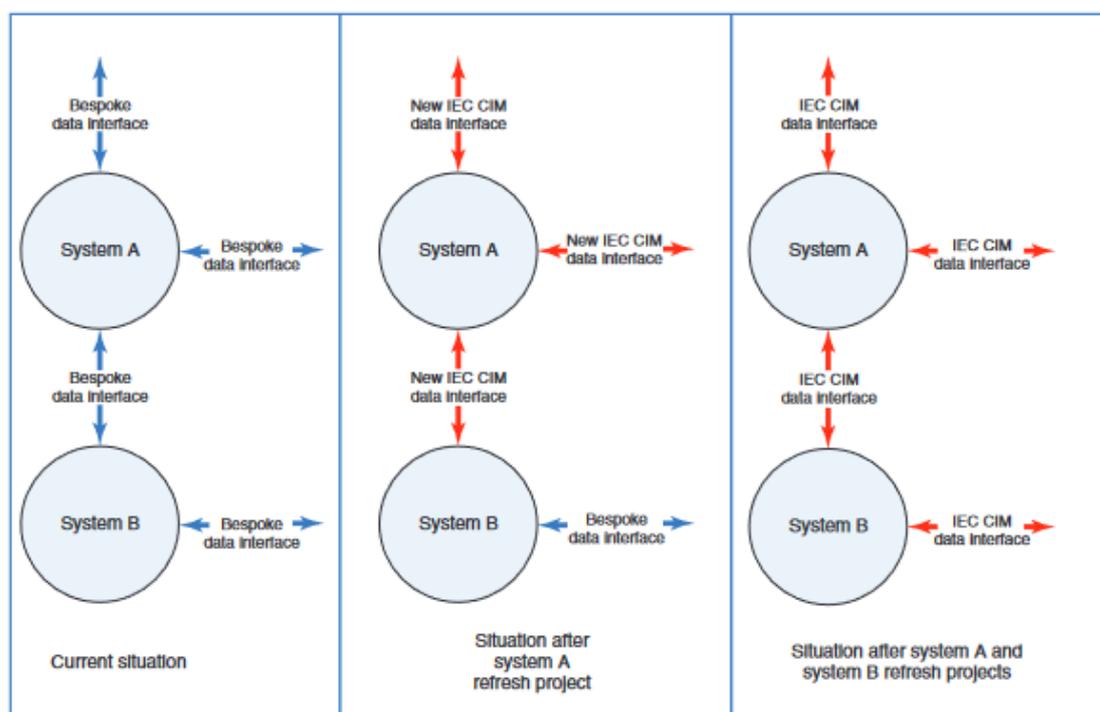


圖 40 系統介面之逐漸更換與支持 IEC CIM 示意圖[20]

六、分析資訊模型系統傳輸與存取效能及提出最佳化方法

6.1 測試架構

為達到提升配電系統之 CIM XML 格式交換與其在資料庫存取效能之目的，本論文針對此目標提出一系列測試，該測試主要分為兩部分，分別為資料存取測試與資料傳輸測試，其中資料傳輸測試再細分為資料獲取與資料查詢測試。首先，資料存取測試為評估在取得相同的 CIM XML 格式的條件下，將配電系統資料儲存於 SQL 與 NoSQL 資料庫，並透過介接 API 進行 CIM XML 格式的獲取，最後分析兩種資料庫其獲取格式所耗費時間做為效能比較之依據。而資料傳輸測試將使用 RESTful 和 SOAP 兩種網路服務，針對在資料存取測試中表現較佳的資料庫進行 CIM XML 格式的查詢與獲取。透過分析各種操作所耗費的時間，進行比較和評估。這將有助於了解在實際網路環境中，使用不同網路服務對資料存取效能的影響。上述兩部分的測試設計有助於評估不同資料庫和網路服務在 CIM XML 格式交換與存取方面的性能表現。最終的分析結果將有助於選擇最適合使用於 CIM XML 格式的資料庫和網路服務，以提升配電系統在資料交換與存取方面的效能和效率。圖 41 為探討 CIM XML 之資料交換與存取效能之測試概念圖。



圖 41 CIM XML 之資料交換與存取效能之測試概念圖

6.1.1 資料存取測試架構

此項測試之目的為評估不同資料庫在 CIM XML 格式存取方面的性能，依照比較兩種資料庫的平均獲取時間，選擇性能較佳之資料庫做為後續測試之測試資料庫。在進行資料存取測試的過程中，首先將配電系統之即時與歷史資料依據定義好的資料庫結構分別儲存於 SQL 與 NoSQL 資料庫中。接著透過資料庫介接 API 至兩種資料庫中獲取 CIM XML 格式，對於 SQL 資料庫，這包含了使用 SQL 查詢語句來擷取相對應的資料。而對於 NoSQL 資料庫，則需要使用特定的查詢語言來獲取相關數據。兩種資料庫在進行每一次 CIM XML 格式資料的獲取操作時，將詳細記錄每次操作的時間並計算其平均獲取時間，以此作為後續性能評估的基準。圖 42 為資料庫測試架構圖。

而此架構中，在兩種資料庫之存取模組中其設計有些微差異，SQL 與 NoSQL 之存取模組都包含了資料庫提供之介接 API，而因 SQL 資料庫之儲存特性，必須使用轉換程式將取出的資料轉為所需之 CIM XML 格式，故 SQL 資料庫之存取模組設置了資料轉換程式。而 NoSQL 資料庫可以直接存取 XML 格式，故不需要額外的資料轉換步驟。

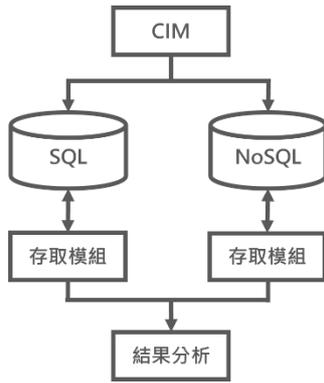


圖 42 資料庫測試架構圖

6.1.2 資料傳輸測試架構

在資料傳輸測試的架構中，基於先前資料存取測試所確定的資料庫選擇，使用 RESTful 和 SOAP 兩種不同的網路服務進行兩項不同的資料傳輸測試。整個測試流程包含了資料獲取和資料查詢兩個部分。首先是資料獲取測試，其流程為透過測試平台分別使用 RESTful 和 SOAP 兩種網路服務向資料庫發送獲取完整 CIM XML 格式之請求，存取模組將負責接收與處理請求並從測試資料庫中擷取完整的 CIM XML 格式的資料回傳給測試平台，最後記錄完整測試流程所耗費之時間。第二項資料查詢測試之流程與第一項測試流程大致相同，同樣為透過測試平台，使用 RESTful 和 SOAP 兩種網路服務向資料庫發送特定資料的查詢請求。存取模組接收並處理請求，執行特定資料的查詢操作，並將查詢結果發送回測試平台，同時也記錄整個流程的時間花費。

最後根據兩項測試之結果針對 RESTful 和 SOAP 兩種網路服務在資料庫資料獲取和查詢方面的表現進行分析與比較，最終的結果將能夠選擇出效能較優的網路服務傳輸方案，進而提升整體系統的資料傳輸效能。圖 43 為資料傳輸測試之架構圖。

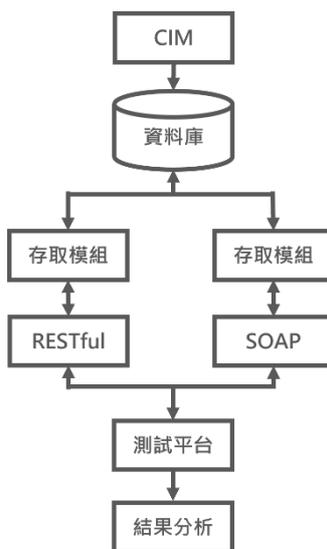


圖 43 資料傳輸測試之架構圖

6.2 實際測試與結果分析

6.2.1 測試平台

Postman 測試平台是一項被廣泛使用的 API 開發工具，可以幫助開發者設計、測試和管理網路服務 API。Postman 提供了直觀而功能豐富的介面，讓用戶可以輕鬆地設定和執行 HTTP 請求，包括 GET、POST、PUT、DELETE 等，並且可以檢查響應內容，以便檢查錯誤。除了測試 API 之外，Postman 還支援多種功能，例如模擬 API、自動化測試、共享 API 等。開發者可以使用 Postman 的模擬功能來測試 API 的性能和可靠性，並且可以通過自動化測試來減少測試時間和成本，而 Postman 還提供了一個 API 資料庫，讓開發人員可以輕鬆地共享 API 和測試範例。Postman 還具有可擴展性和整合性，支援多種網路服務技術和第三方應用程式，例如 JSON、XML、OAuth、AWS 等，開發人員可以根據自己的需求來選擇相應的技術和應用程式。Postman 還可以整合到其他開發工具中，例如 Visual Studio、Eclipse 等，使得開發人員可以更加高效地開發和測試 Web 服務。使用 Postman，開發人員可以更輕鬆地設計和開發網路服務，同時減少錯誤和測試時間。因此，本計畫選擇 Postman 做為負責向存取模組發送請求與進行測試之工具，圖 44 為 Postman 測試頁面。

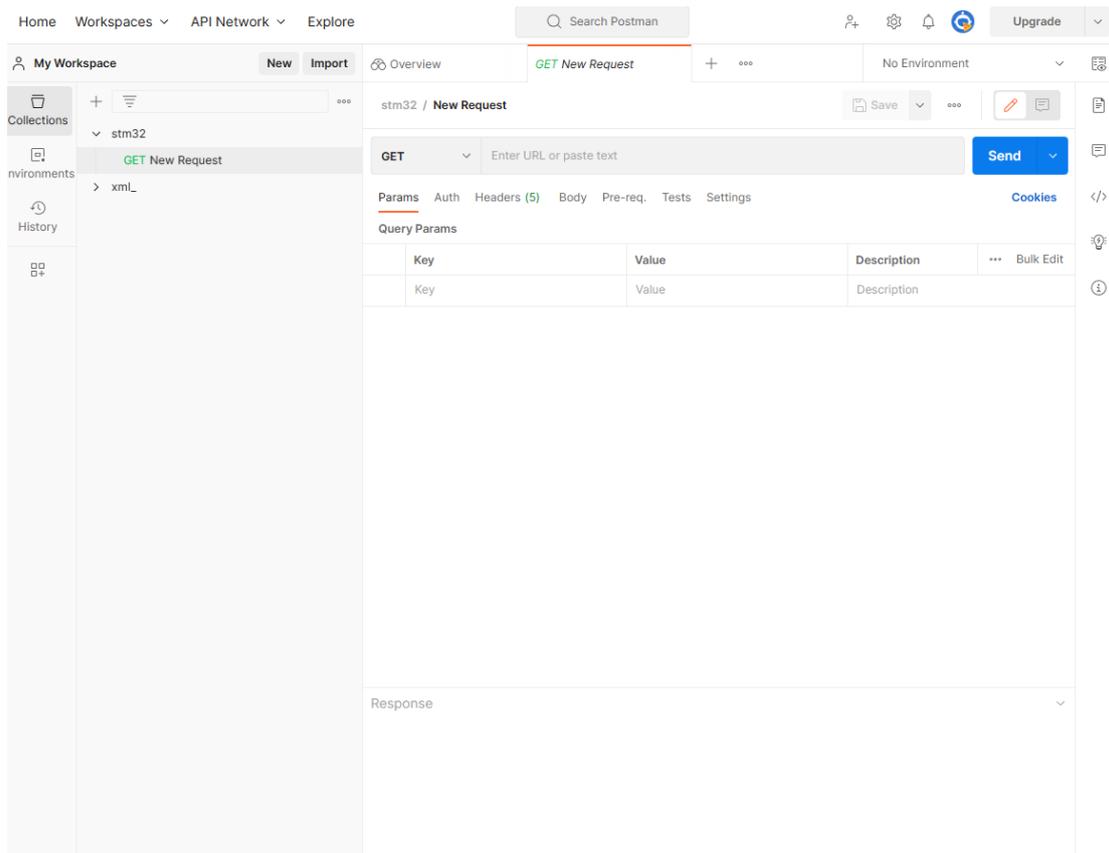


圖 44 Postman 測試頁面

6.2.2 測試資料

本計畫採用 TTU 之 CIM XML 實現作為測試用 CIM XML 格式。TTU 之 CIM XML 實現需要先根據其 UML 結構模型進行定義，如圖 45 所示，圖 45 為根據

國際規範 IEC 61970-301 中量測套件內已定義完成之量測與控制模型為基礎，配合台電針對 TTU 所定義之資料傳輸格式所建立之 TTU 的 UML 結構模型。在 TTU 的 UML 模型中，針對 TTU 的設備識別、即時監測與能量測量等資料皆進行了類別與屬性的定義。在「Terminal」類別中，為了表示設備的身分識別與相關連結性之端口，設置了「TTU_UFID」屬性，賦予每台設備獨立的 UFID 作為設備辨識序號，並使用「PhaseCode」屬性用於描述所量測之配電變壓器的相別。而在「BassCapacity」類別中則表示了所量測之變壓器的容量值(KVA)。由於 TTU 主要用於即時記錄與監控饋線之狀態，因此在其 UML 結構模型中設置了實時量測值「AnalogValue」類別，並在其屬性中定義了「lineVoltageVX」、「lineCurrentX」、「ApparentPowerS」、「ActivePowerP」、「ReactivePowerQ」等即時狀態測量值。而該饋線所產生之實功與虛功，則以「KWHplus」、「KWHminus」、「KVARminus」與「KVARplus」等屬性包含在「AccumulatorValue」類別中以進行監測與累計。圖 46 為根據 TTU 之 UML 模型所定義之 TTU CIM XML 格式，其中包含了依台電所定義之 TTU 資料傳輸內容，而本計畫選擇查詢其中之「LineVoltageVX1」作為資料查詢測試範例。

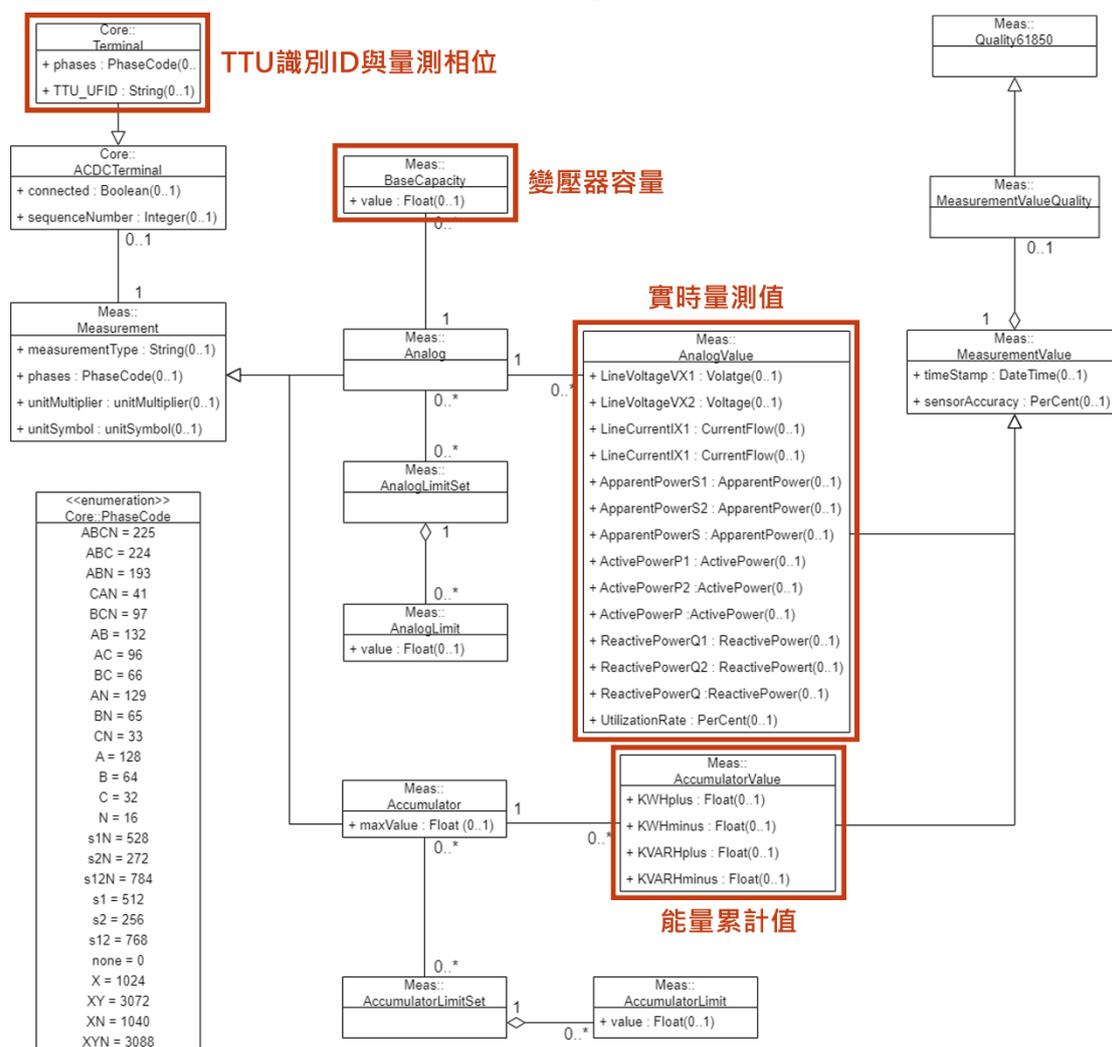


圖 45 TTU 之 UML 結構模型

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:cim="http://iec.ch/TC57/2009/CIM-schema-cim14#">
  <cim:Terminal rdf:ID="#_TransformerAID">
    <cim:Terminal.phases>xxx</cim:Terminal.phases>
    <cim:Terminal.TTU_UFID>Q1212FA6128</cim:Terminal.TTU_UFID>
    <cim:Terminal.Measurement rdf:resource="#_TTU1ID"/>
    <cim:Terminal.ConnectivityNode rdf:resource="#_Node1ID"/>
    <cim:IdentifiedObject.name>TransformerA</cim:IdentifiedObject.name>
  </cim:Terminal>
  <cim:AnalogValue rdf:ID="#_AnalogValueID">
    <cim:AnalogValue.LineVoltageVX1>118</cim:AnalogValue.LineVoltageVX1>
    <cim:AnalogValue.LineVoltageVX2>118</cim:AnalogValue.LineVoltageVX2>
    <cim:AnalogValue.LineCurentIX1>63</cim:AnalogValue.LineCurentIX1>
    <cim:AnalogValue.LineCurentIX2>62</cim:AnalogValue.LineCurentIX2>
    <cim:AnalogValue.ActivePowerP>16</cim:AnalogValue.ActivePowerP>
    <cim:AnalogValue.ReactivePowerQ>16</cim:AnalogValue.ReactivePowerQ>
    <cim:AnalogValue.ApparentPowerS>16</cim:AnalogValue.ApparentPowerS>
    <cim:AnalogValue.ActivePowerP1>16</cim:AnalogValue.ActivePowerP1>
    <cim:AnalogValue.ReactivePowerQ1>16</cim:AnalogValue.ReactivePowerQ1>
    <cim:AnalogValue.ApparentPowerS1>16</cim:AnalogValue.ApparentPowerS1>
    <cim:AnalogValue.ActivePowerP2>16</cim:AnalogValue.ActivePowerP2>
    <cim:AnalogValue.ReactivePowerQ2>16</cim:AnalogValue.ReactivePowerQ2>
    <cim:AnalogValue.ApparentPowerS2>16</cim:AnalogValue.ApparentPowerS2>
    <cim:AnalogValue.UtilizationRate>xxx</cim:AnalogValue.UtilizationRate>
    <cim:IdentifiedObject.name>TTU1</cim:IdentifiedObject.name>
    <cim:IdentifiedObject.aliasname>Q1212FA6128</cim:IdentifiedObject.aliasname>
    <cim:IdentifiedObject.description>voltage monitoring of feeders connected the
    secondary side of the Transformer A</cim:IdentifiedObject.description>
    <cim:Measurement.Terminal rdf:resource="#_TransformerAID"/>
  </cim:AnalogValue>
</rdf:RDF>

```

圖 46 TTU 之 CIM XML 格式

根據 5.2.2 節所討論，由於 SQL 資料庫無法直接儲存 CIM XML 格式，故需要經過資料轉換的步驟，才能將 CIM XML 格式轉換為適合 SQL 資料庫的形式。資料轉換過程中，必須以其 UML 結構模型為基礎，將 CIM XML 格式中的屬性參數與其對應之類別與屬性進行資料庫點位轉換，以實現將 CIM XML 格式轉換成關聯性資料表的形式進行儲存。圖 47 展示了針對 SQL Server 資料庫所設計的資料表及其相關的屬性參數，其中包含了根據 UML 結構模型所拆解及儲存的 TTU CIM XML 格式之屬性參數，同時為了滿足後續應用程式對電力數據的排序和查詢需求，更加入了時間標記（Time Stamp）和 UFID 標號欄位。相對於 SQL 資料庫，另一測試資料庫 NoSQL 資料庫 BaseX，因為其專門設計於存取與

6.2.3 測試結果

- 資料存取測試

在前一節中，已完成了 CIM XML 格式在兩種資料庫中的資料儲存部分。此小節將根據 6.1.1 節的資料存取測試架構圖，透過兩種資料庫的存取模組以獲取 CIM XML 格式，即本計畫開發之 SQL Server 資料庫存取模組和 BaseX 資料庫存取模組。

在 SQL Server 資料庫的存取模組中，首先為 SQL Server 資料庫提供的資料庫介接 API，如圖 49 所示。而因為 SQL Server 資料庫無法直接存取 XML 格式，故需要藉由資料轉換程式將取得之 CIM XML 屬性資料根據 UML 結構模型之定義重新組成 CIM XML 格式，圖 50 為資料轉換程式部分節錄。經過資料轉換程式的處理，從 SQL Server 資料庫取出之 CIM 屬性參數可以成功組合成標準之 CIM XML 格式，圖 51 為經過 SQL Server 資料庫存取模組轉換後得出之 CIM XML 格式。

相對的，BaseX 資料庫作為一個專門設計於管理與存儲 XML 檔案的資料庫，則無需進行額外的資料轉換，便可以直接 CIM XML 格式的資料。圖 52 為 BaseX 資料庫所提供之資料庫介接 API 的程式節錄，圖 53 為透過 BaseX 存取模組所得之 CIM XML 格式。

```
import pyodbc
#sql server 資訊
server = 'DB Server'
database = 'DB Name'
username = 'User Name'
password = 'User Password'

conn = pyodbc.connect('DRIVER={ODBC Driver 18 for SQL Server};SERVER='+server+';DATABASE='+database+';UID='+username+';PWD='+password+';TrustServerCertificate=yes')
```

圖 49 SQL Server 資料庫提供之資料庫介接 API

```
#data前處理
def rows_deal(rows):
    rows=str(rows)
    rows=rows.replace("'",'').replace('(',').replace(')',').replace('[',').replace(']',').replace(' ','').split(',')
    rows.pop(0)
    return rows

def create_xml_string(data):
    # 創建文檔
    dom = xml.dom.minidom.Document()

    # 創建根元素
    root = dom.createElement("rdf:RDF")
    root.setAttribute("xmlns:rdf", "http://www.w3.org/1999/02/22-rdf-syntax-ns#")
    root.setAttribute("xmlns:rdfs", "http://www.w3.org/2000/01/rdf-schema#")
    root.setAttribute("xmlns:cim", "http://iec.ch/TC57/2009/CIM-schema-cim14#")

    # 添加子元素
    terminal = dom.createElement("cim:Terminal")
    terminal.setAttribute("rdf:ID", "_TransformerAID")
    root.appendChild(terminal)

    phases = dom.createElement("cim:Terminal.phases")
    phases.appendChild(dom.createTextNode("xxx"))
    terminal.appendChild(phases)
```

圖 50 SQL Server 資料庫資料轉換程式部分節錄

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:cim="http://iec.ch/TC57/2009/CIM-schema-cim14#">
  <cim:Terminal rdf:ID="#_TransformerAID">
    <cim:Terminal.phases>xxx</cim:Terminal.phases>
    <cim:Terminal.TTU_UFID>Q1212FA6128</cim:Terminal.TTU_UFID>
    <cim:Terminal.Measurement rdf:resource="#_TTU1ID"/>
    <cim:Terminal.ConnectivityNode rdf:resource="#_Node1ID"/>
    <cim:IdentifiedObject.name>TransformerA</cim:IdentifiedObject.name>
  </cim:Terminal>
  <cim:AnalogValue rdf:ID="#_AnalogValueID">
    <cim:AnalogValue.LineVoltageVX1>11858</cim:AnalogValue.LineVoltageVX1>
    <cim:AnalogValue.LineVoltageVX2>11858</cim:AnalogValue.LineVoltageVX2>
    <cim:AnalogValue.LineCurentIX1>406</cim:AnalogValue.LineCurentIX1>
    <cim:AnalogValue.LineCurentIX2>265</cim:AnalogValue.LineCurentIX2>
    <cim:AnalogValue.ActivePowerP>45</cim:AnalogValue.ActivePowerP>
    <cim:AnalogValue.ReactivePowerQ>20</cim:AnalogValue.ReactivePowerQ>
    <cim:AnalogValue.ApparentPowerS>40</cim:AnalogValue.ApparentPowerS>
    <cim:AnalogValue.ActivePowerP1>27</cim:AnalogValue.ActivePowerP1>
    <cim:AnalogValue.ReactivePowerQ1>12</cim:AnalogValue.ReactivePowerQ1>
    <cim:AnalogValue.ApparentPowerS1>24</cim:AnalogValue.ApparentPowerS1>
    <cim:AnalogValue.ActivePowerP2>18</cim:AnalogValue.ActivePowerP2>
    <cim:AnalogValue.ReactivePowerQ2>9</cim:AnalogValue.ReactivePowerQ2>
    <cim:AnalogValue.ApparentPowerS2>16</cim:AnalogValue.ApparentPowerS2>
    <cim:AnalogValue.UtilizationRate>814</cim:AnalogValue.UtilizationRate>
    <cim:IdentifiedObject.name>TTU1</cim:IdentifiedObject.name>
    <cim:IdentifiedObject.aliasname>Q1212FA6128</cim:IdentifiedObject.aliasname>
    <cim:IdentifiedObject.description>voltage monitoring of feeders connected the secondary side of the Transformer A</cim:IdentifiedObject.description>
  </cim:AnalogValue>
</rdf:RDF>

```

圖 51 SQL Server 資料庫存取模組資料轉換後之 CIM XML 格式

```

from BaseXClient import BaseXClient
# create session
session = BaseXClient.Session('DB Server', 'Port', 'User Name', 'User Password')

```

圖 52 BaseX 資料庫提供之資料庫介接 API

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:cim="http://iec.ch/TC57/2009/CIM-schema-cim14#">
  <cim:Terminal rdf:ID="#_TransformerAID">
    <cim:Terminal.phases>xxx</cim:Terminal.phases>
    <cim:Terminal.TTU_UFID>Q1212FA6128</cim:Terminal.TTU_UFID>
    <cim:Terminal.Measurement rdf:resource="#_TTU1ID"/>
    <cim:Terminal.ConnectivityNode rdf:resource="#_Node1ID"/>
    <cim:IdentifiedObject.name>TransformerA</cim:IdentifiedObject.name>
  </cim:Terminal>
  <cim:AnalogValue rdf:ID="#_AnalogValueID">
    <cim:AnalogValue.LineVoltageVX1>11858</cim:AnalogValue.LineVoltageVX1>
    <cim:AnalogValue.LineVoltageVX2>11858</cim:AnalogValue.LineVoltageVX2>
    <cim:AnalogValue.LineCurentIX1>406</cim:AnalogValue.LineCurentIX1>
    <cim:AnalogValue.LineCurentIX2>265</cim:AnalogValue.LineCurentIX2>
    <cim:AnalogValue.ActivePowerP>45</cim:AnalogValue.ActivePowerP>
    <cim:AnalogValue.ReactivePowerQ>20</cim:AnalogValue.ReactivePowerQ>
    <cim:AnalogValue.ApparentPowerS>40</cim:AnalogValue.ApparentPowerS>
    <cim:AnalogValue.ActivePowerP1>27</cim:AnalogValue.ActivePowerP1>
    <cim:AnalogValue.ReactivePowerQ1>12</cim:AnalogValue.ReactivePowerQ1>
    <cim:AnalogValue.ApparentPowerS1>24</cim:AnalogValue.ApparentPowerS1>
    <cim:AnalogValue.ActivePowerP2>18</cim:AnalogValue.ActivePowerP2>
    <cim:AnalogValue.ReactivePowerQ2>9</cim:AnalogValue.ReactivePowerQ2>
    <cim:AnalogValue.ApparentPowerS2>16</cim:AnalogValue.ApparentPowerS2>
    <cim:AnalogValue.UtilizationRate>814</cim:AnalogValue.UtilizationRate>
    <cim:IdentifiedObject.name>TTU1</cim:IdentifiedObject.name>
    <cim:IdentifiedObject.aliasname>Q1212FA6128</cim:IdentifiedObject.aliasname>
    <cim:IdentifiedObject.description>voltage monitoring of feeders connected the secondary side of the Transformer A<
  </cim:IdentifiedObject.description>
    <cim:Measurement.Terminal rdf:resource="#_TransformerAID"/>
  </cim:AnalogValue>
</rdf:RDF>

```

圖 53 BaseX 資料庫存取模組獲取之 CIM XML 格式

此項測試的主要目的是評估兩種資料庫在處理 CIM XML 格式資料時的性能，因此此項測試以獲取 CIM XML 格式時所耗費的時間長短作為為評估兩種資料庫在處理 CIM XML 格式的性能之依據。圖 54 展示出了兩種資料庫所耗費之時間結果，此項測試結果表明，SQL Server 和 BaseX 在處理 CIM XML 資料的性能上存在顯著的差異。從圖中可以看出，BaseX 在耗費時間上

比 SQL Server 少了一倍以上，這取決於資料庫的特性和儲存結構。由於 SQL Server 資料庫存取 CIM XML 格式必須經過資料轉換的步驟，這導致在獲取時花費的時間相對較長。而相比之下，BaseX 資料庫作為原生 XML 資料庫，無需額外的資料轉換步驟，可以直接將 CIM XML 格式儲存於 BaseX 中，使得從 BaseX 資料庫中提取 CIM XML 格式的資料更加直觀和高效。

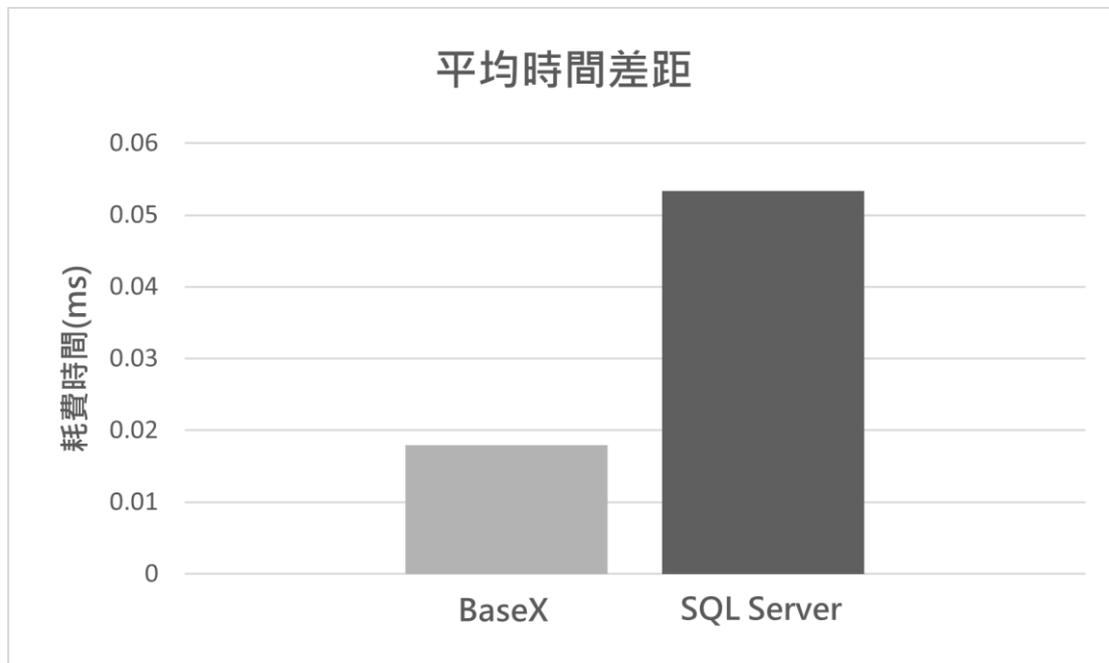


圖 54 資料庫存取測試時間差距圖

總結來說，根據此項測試結果，本計畫選擇使用 BaseX 資料庫作為處理 CIM XML 格式的資料庫，將有助於更有效的提高系統的性能和效率。並考慮 BaseX 資料庫的輕量化設計和易於擴展的能力，使其在需要處理大量資料的配電系統中提供更好的效能表現。基於以上的比較和分析，BaseX 成為此項計畫接續之測試資料庫的理想選擇。

- 資料傳輸測試

在上一階段的資料存取測試中，性能較優之 BaseX 資料庫被選定作為接下來之測試資料庫。根據 6.1.2 節的資料傳輸測試架構圖，本項測試的流程為透過測試平台向存取模組發送 CIM XML 格式獲取與查詢請求，存取程式處理請求後將對應之資料從資料庫取出並回覆給測試平台，最後紀錄完整流程所耗費之時間。本階段之測試目的為透過 RESTful 與 SOAP 兩種傳輸方式進行資料庫的資料獲取與查詢測試，以其花費之時間作為判斷其性能優劣之依據，並選出性能較優之傳輸方式。

欲使用測試平台向存取模組發送請求，需要先在存取模組中針對 RESTful 與 SOAP 兩種傳輸方式進行網路服務端口配置。首先，RESTful 網路服務其核心是基於對 URI 進行 HTTP 之 POST、GET 等操作。因此，必須事先對定義 URI 的結構，具體結構如圖 55 所示。本計畫所定義之 URI 由三個組件組合而成，首先為基底 URI，其中包含了 IP 位置與所使用的網路服務。其次為資料庫資訊與欲操作之檔案，此組件會與實際資料庫之路徑相匹配。最後為延伸 URI，可用於針對檔案內部之詳細資料進行查詢與管理。完成 URI 定義後，為符合 RESTful 網路服務之標準，本計畫針對存取模組中的網路服務端口進行設計，以定義完成之 URI 為基礎，針對 POST、GET 等操作進行相對應之資料庫功能設定，圖 56 為存取模組中遵循 RESTful 準則設計，以 HTTP 操作為基礎之 URI 示意圖，其於本計畫所設置之網路服務端口中之具體實現如圖 57 所示。

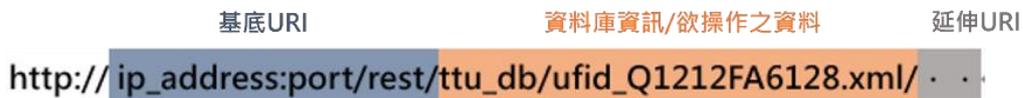


圖 55 RESTful 服務之 URI 示意圖

HTTP操作	URI	結果
GET	http://ip_address:port/rest/ttu_db/ufid_Q1212FA6128.xml/	取得完整之CIM XML格式
POST	http://ip_address:port/rest/ttu_db/ufid_Q1212FA6128.xml/	存入完整之CIM XML格式
GET	http://ip_address:port/rest/ttu_db/ufid_Q1212FA6128.xml/A	查詢A元素之內容
POST	http://ip_address:port/rest/ttu_db/ufid_Q1212FA6128.xml/A	存入A元素之內容

圖 56 使用 HTTP 對 RESTful 服務之 URI 操作示意圖

```
@app.route('/rest/ttu_db/<ufid>', methods=['GET'])
def get_ufid(ufid):
    # 連線到basex並獲取xml
    session = BaseXClient.Session('10.3.10.123', 1984, 'admin', '0000')
    xml_path = f"ttu_db/{ufid}"
```

圖 57 RESTful 服務之 URI 具體實現部分節錄

根據第 3.2 節所述，相對於 RESTful 網路服務，SOAP 主要使用 HTTP 的 POST 操作，而主要的資訊內容正文會被置於 SOAP 訊息封套中的主體裡，並置於 POST 封包的 Body 中進行傳輸。因此，在配置存取模組的 SOAP 網路服務端口時，需要針對資料庫的相關功能進行操作設定，以提供使用者根據不同操作修改 SOAP 訊息封套內的正文。而 SOAP 不需要對 URI 進行定義，故以 RESTful 網路服務之 URI 為基礎，簡化為僅含有 IP 位置與資料庫訊息。圖 58 展示了 SOAP 網服務端口的 URI 與 SOAP 之操作概念圖。根據需求於網路服務端口定義不同功能之具體實現範例如圖 59 所展示。

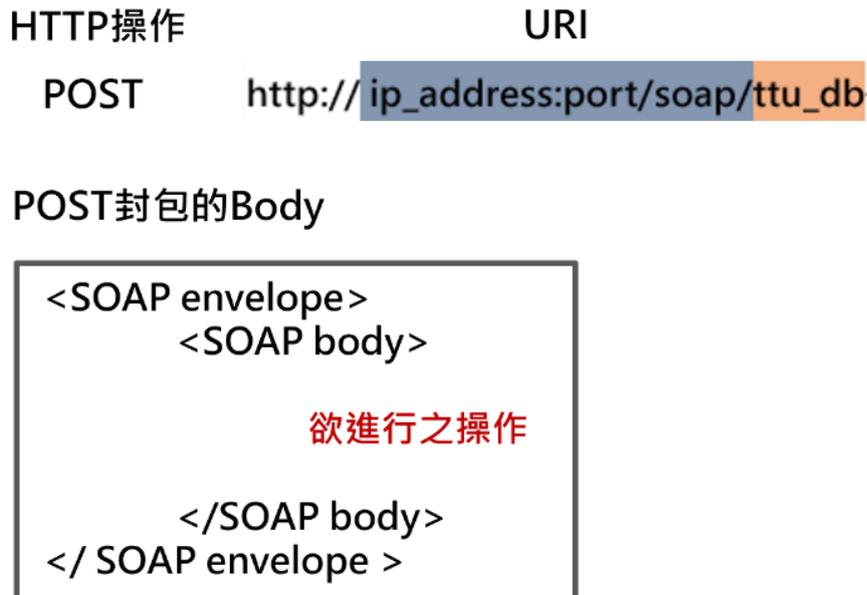


圖 58 SOAP 服務之 URI 與操作概念圖

```

class MyService(ServiceBase):
    @rpc(Unicode, _returns=Unicode)
    def GET_TTU_CIM(self, ufid):
        # 連線到basex並獲取xml
        session = BaseXClient.Session('10.3.10.123', 1984, 'admin', '0000')
```

圖 59 SOAP 服務功能定義部分節錄

完成兩種傳輸方式的網路服務端口的配置後，便可以於測試平台中進行資料獲取與查詢測試。透過事先定義的 URI，使用 HTTP 操作便可實現向 BaseX 進行基於 RESTful 網路服務的資料獲取與查詢測試。圖 60 展示出使用 Postman 測試平台透過對 RESTful 網路服務端口定義之 URI 進行 GET 操作以獲取 CIM XML 格式的實際測試畫面，其獲得之 CIM XML 格式如圖 61 所示。而在資料查詢測試方面，只需要在 URI 後加上欲查詢之資料，同樣的使用 GET 方法，便可獲得所需之資料。圖 62 為透過 URI 進行資料查詢的測試，欲查詢之資料為 CIM XML 格式內的 LineVoltageVX1 元素，因此在 URI 後加入 LineVoltageVX1 便可以得到想查詢之資料內容，圖 63 為獲得之 LineVoltageVX1 查詢資料。SOAP 之資料獲取與查詢測試操作則是透過測試

平台向網路服務端口 POST 一段符合 SOAP 協議的 XML 訊息。如圖 64 所示，圖 64 展示了透過 SOAP 服務進行資料獲取測試的運作過程，將 SOAP 封套訊息加入在欲傳送之資料封包的 Body 中，而在 SOAP 封套內的主體中包含了對存取模組發送資料獲取的請求，圖 65 為其獲得之 CIM XML 格式。在 SOAP 資料查詢測試中，大致與其資料獲取測試相同，只需要改變 SOAP 封套內的主體正文，將資料查詢的請求放入其中，如圖 66 所示。獲得之查詢資料內容如圖 67 所示。



圖 60 對 RESTful 服務之 URI 進行 GET 操作之資料獲取測試畫面

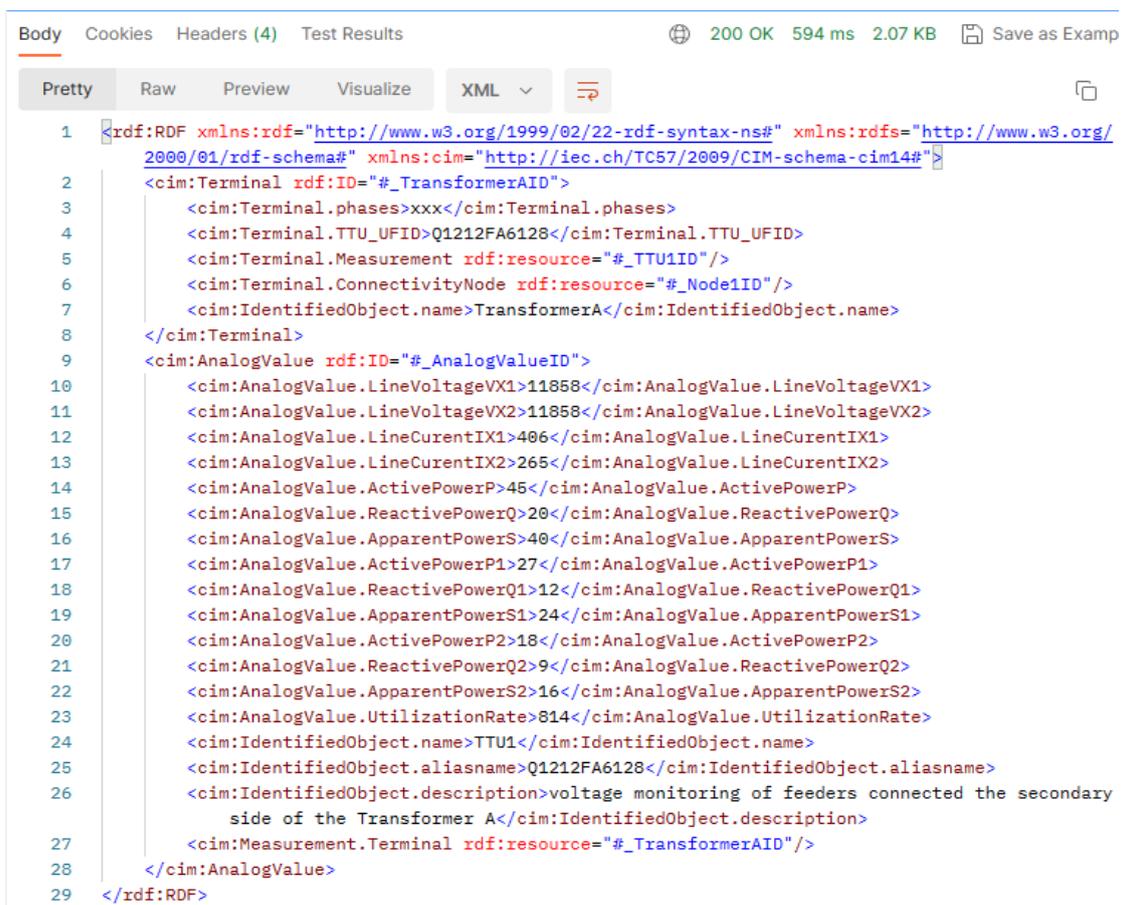


圖 61 RESTful 服務回覆之 CIM XML 格式



圖 62 對 RESTful 服務之 URI 延伸進行資料查詢之測試畫面



圖 63 RESTful 服務回覆之查詢資料內容

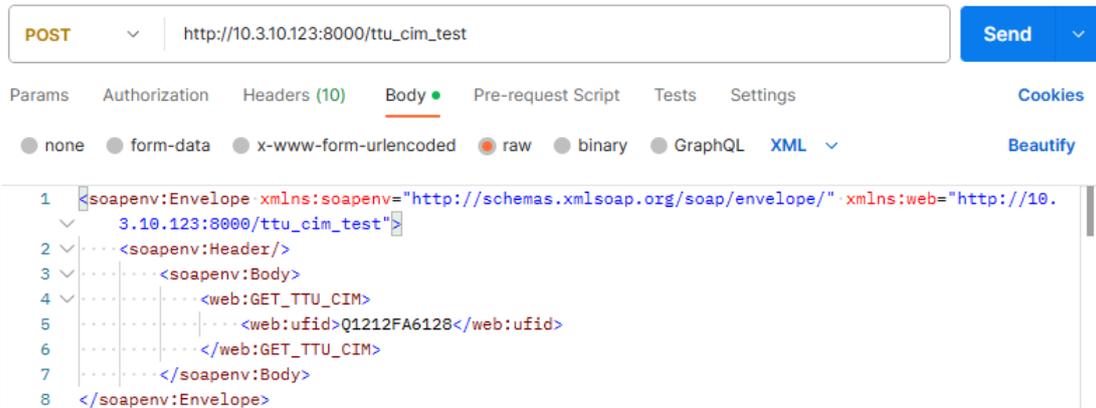


圖 64 對 SOAP 服務之 URI POST 資料獲取測試封包之實際畫面

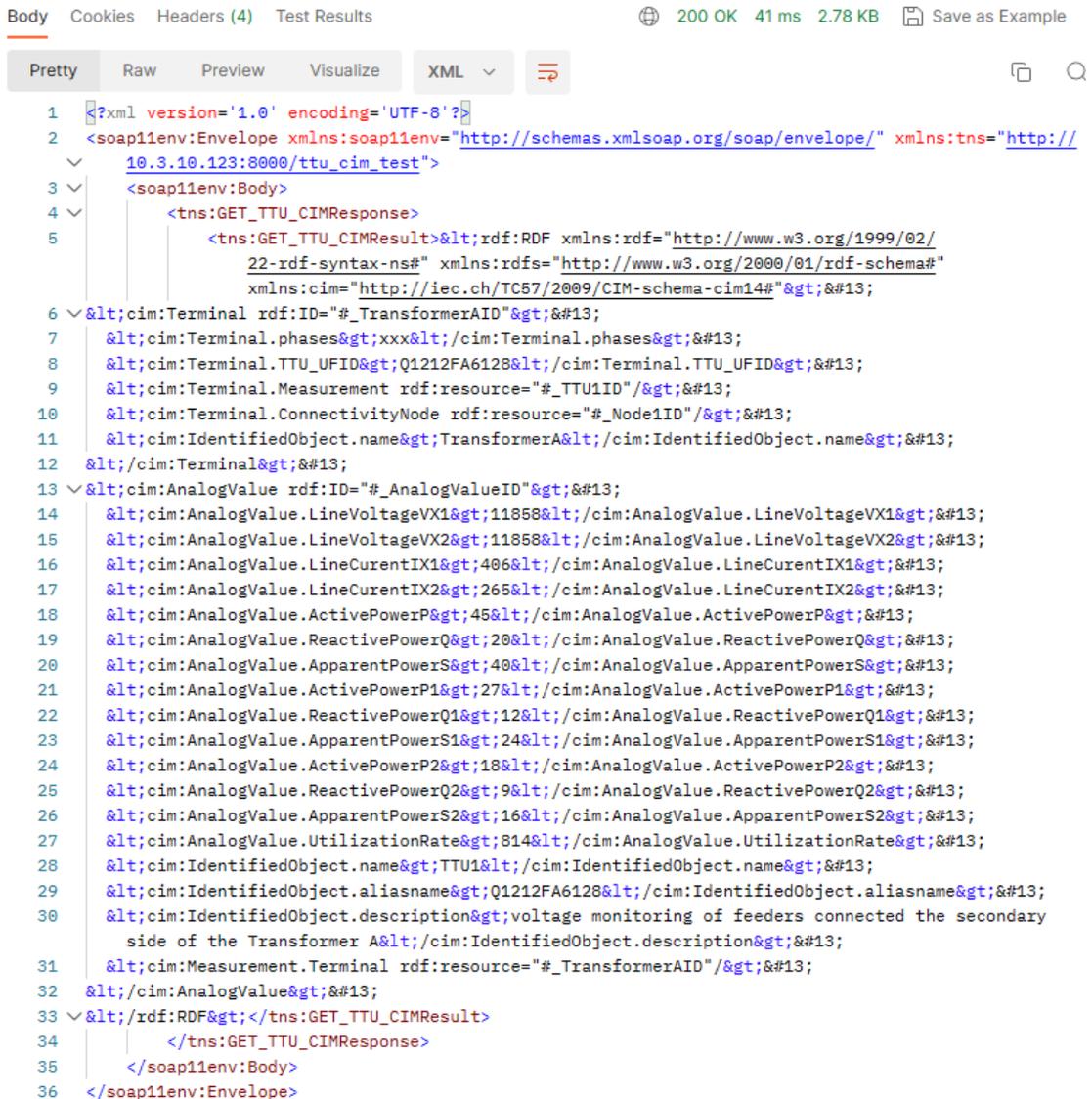


圖 65 SOAP 服務回覆之 CIM XML 格式

```
POST http://10.3.10.123:8000/ttu_cim_test Send
Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies
● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL XML Beautify
1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://10.
  3.10.123:8000/ttu_cim_test">
2   <soapenv:Header/>
3   <soapenv:Body>
4     <web:QUERY_DATA>
5       <web:ufid>Q1212FA6128</web:ufid>
6       <web:element>LineVoltageVX1</web:element>
7     </web:QUERY_DATA>
8   </soapenv:Body>
9 </soapenv:Envelope>
```

圖 66 對 SOAP 服務之 URI POST 資料查詢測試封包之實際畫面

```
Body Cookies Headers (4) Test Results 200 OK 17 ms 476 B Save as example
Pretty Raw Preview Visualize XML
1 <?xml version='1.0' encoding='UTF-8'?>
2 <soap11env:Envelope xmlns:soap11env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tns="http://
  10.3.10.123:8000/ttu_cim_test">
3   <soap11env:Body>
4     <tns:QUERY_DATAResponse>
5       <tns:QUERY_DATAResult>11858</tns:QUERY_DATAResult>
6     </tns:QUERY_DATAResponse>
7   </soap11env:Body>
8 </soap11env:Envelope>
```

圖 67 SOAP 服務回覆之 CIM XML 格式

經過以上測試流程後，皆證明了 RESTful 和 SOAP 兩種傳輸方式可以成功獲取 CIM XML 格式與查詢到所需要之資料。以測試之結果為基礎，進行兩種傳輸方式所耗費的時間之分析與比較，確定哪一種傳輸方式在性能上表現較優。從圖 68 可以看出，在資料獲取方面，RESTful 服務相比於 SOAP 在較短的時間內完成了 CIM XML 格式的獲取。意味著 RESTful 在資料傳輸過程中能夠更快速地將 CIM XML 格式傳送至客戶端，提供更高效率的資料交換服務。同樣的，RESTful 服務在資料查詢方面同樣展現出優越的效能。圖 69 顯示出 RESTful 服務在查詢所需資料的時間明顯快於 SOAP 服務，這表示 RESTful 服務在處理查詢請求時更為快速。值得注意的是，RESTful 服務的設計相對簡潔，URI 的定義清晰明確，能降低操作的複雜度。而 SOAP 服務在配置上則需要針對封套內的資料正文進行定義與解讀，較為複雜。

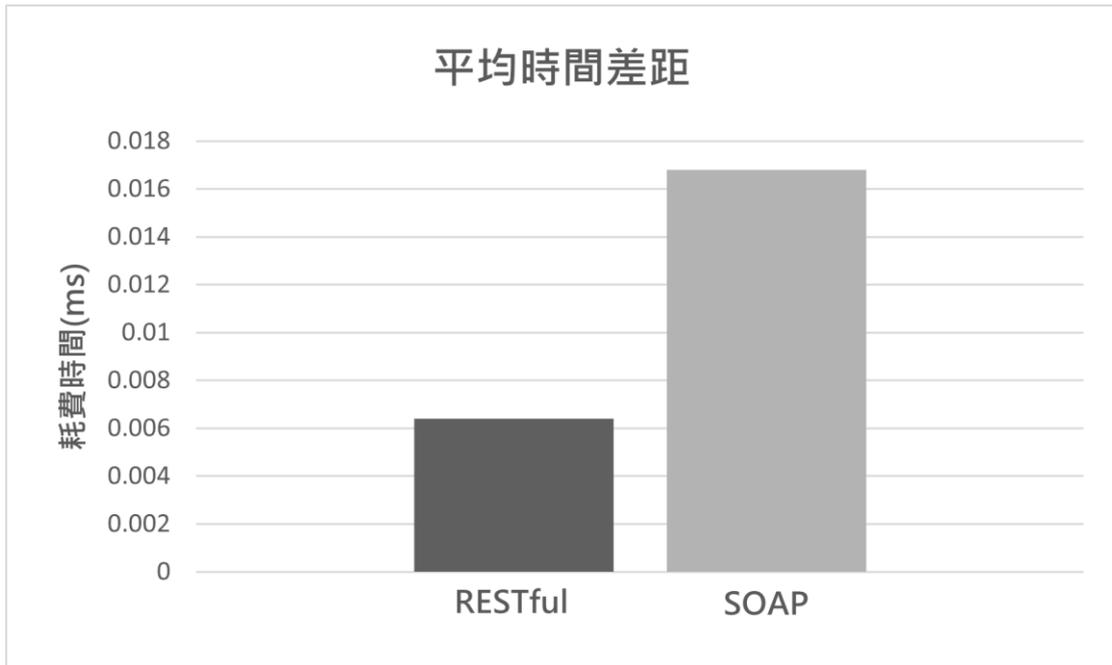


圖 68 資料傳輸測試時間差距圖

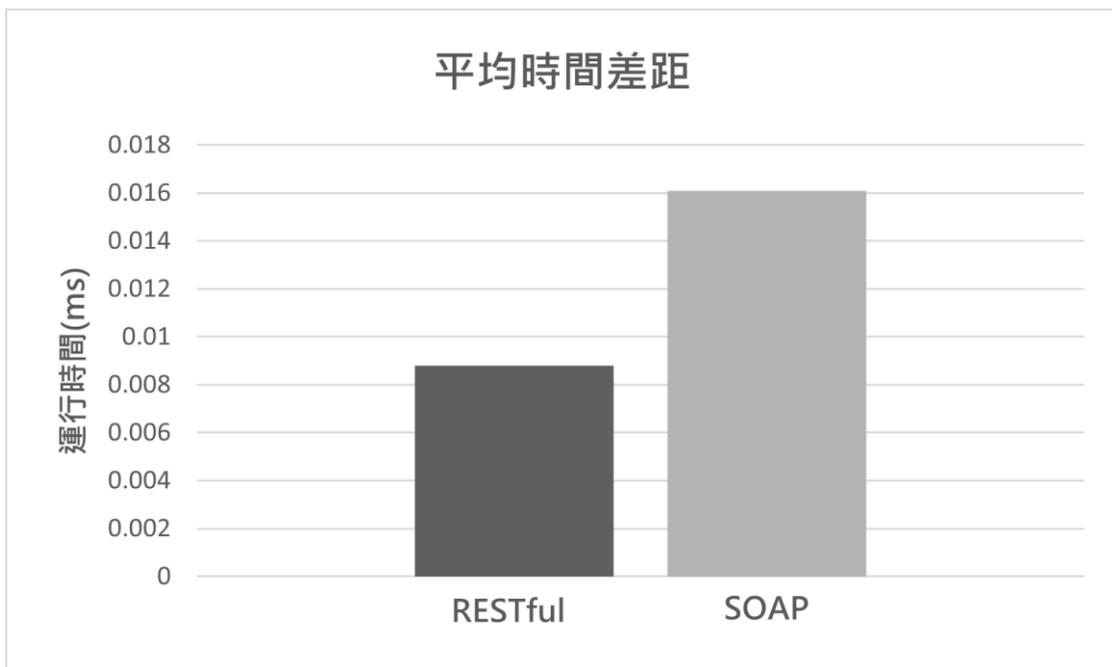


圖 69 資料查詢測試時間差距圖

經過資料存取與傳輸兩項測試後，分別得出性能較優之選擇。資料存取方面，BaseX 相比於 SQL Server 在 CIM XML 格式的處理與相容性上較為優秀。而資料傳輸方面，RESTful 網路服務在資料獲取與查詢上擁有比 SOAP 網路服務更快之傳輸速度，而操作複雜度更為簡潔，易於維護。因此，本計畫選擇以 BaseX 資料庫為基礎，結合 RESTful 網路服務，以實現對 CIM XML 格式的高效存取和傳輸。這一選擇不僅考量到了資料存取的效能，也在網路服務的選擇上追求了簡單性和高效性，為整體系統提供最佳的性能優勢。

七、變壓器運維狀態與圖形化介面

本計畫以 CIM 為基礎，針對變壓器之運維狀態架設一圖形化介面。此圖形化介面之概念為以本計畫所探討之 CIM 資料傳輸系統為延伸，針對配電系統設備之運轉與維護狀態進行即時監控，達到便於維護人員監控與管理配電系統備之功能。本圖形化介面是以 Python 語言中的 GUI 解決方案之一，PyQt6 進行建置與開發。本圖形化介面主要分為前後端兩個部分，前端部分之主要功能為即時運維狀態呈現與用戶操作響應，而後端部分則負責前端操作功能定義與連結架設之 RESTful 網路服務端口進行需求處理，向資料庫提取配電系統設備相關資訊。以下為本計畫提出之變壓器運維圖形化介面之簡單範例圖片，該圖形化介面含有安全性登入、資料表格顯示、資料曲線轉換、重要資訊顯示與變壓器健康度示警等功能，如圖 70 與圖 71 所展示。



圖 70 安全性登入

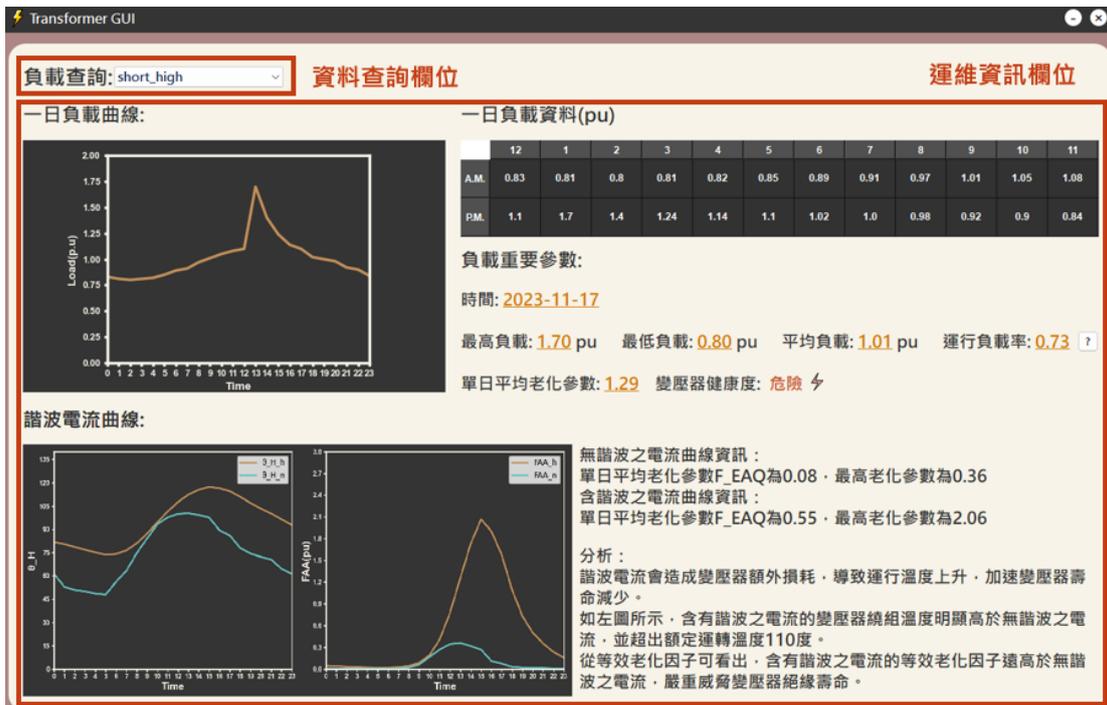


圖 71 運維狀態圖形化介面

參、結論

本計畫針對 CIM 之資料交換及存取方式提出一系列測試，該測試分為資料存取測試與資料傳輸測試兩部分。在資料存取測試中，結果顯示，由於 NoSQL 資料庫提供專門設計於處理 XML 之資料庫，故在處理 CIM XML 格式上顯得有更高的相容與速率，而 SQL 資料庫雖有結構化資料處理以便於管理及靈活查詢等優點，但因無法直接存取 XML 格式在處理 CIM XML 格式的存取上較 NoSQL 資料庫緩慢與複雜。而在資料傳輸測試方面得出，因 SOAP 是一種基於 XML 的協議，並支持較多的安全性和事務處理，使用較為複雜的消息格式，使得其效能較低。相對地，RESTful 為一種基於 HTTP 的傳輸方式，使用較為簡單的消息格式，因此支援較高的效能和可擴展性。根據上述之結果，得出 RESTful 網路服務搭配 NoSQL 資料庫進行資料傳輸與存取為最佳選擇方案。本計畫之 CIM 資料傳輸與存取選擇方案，相信將有助於提升配電系統資料交換的效能與降低其存取複雜度，以更好地應用於實際配電系統控制與管理中。而在本計畫中，亦針對配電變壓器運維狀態進行研究與探討，根據 IEEE 與 IEC 提出之標準，以負載與諧波兩項因素為基礎，針對配電變壓器運轉狀態與壽命進行圖形化界面的建立與呈現，達到操作人員便於進行配電設備即時監控與維護之目的。

肆、參考文獻

- [1] Electric Power Research Institute(EPRI),"Common Information Model Primer", December 2011
- [2] I. Lendak, E. Varga, A. Erdeljan, and M. Gavric, "RESTful web services and the Common Information Model (CIM)," in *2010 IEEE International Energy Conference*, Manama, Bahrain: IEEE, Dec. 2010, pp. 716–721.
- [3] J. V. De Barros and J. B. Leite, "Development of a Relational Database Oriented on the Common Information Model for Power Distribution Networks," in *2021 IEEE URUCON*, Jan. 2021, pp. 63–66.
- [4] W3C Recommendation, "RDF Schema 1.1", https://www.w3.org/TR/rdfschema/#ch_summary
- [5] W3School, "XML RDF", https://www.w3schools.com/xml/xml_rdf.asp
- [6] W3C Recommendation, "W3C XML Schema Definition Language (XSD) 1.1 Part 1", <https://www.w3.org/TR/xmlschema11-1/>
- [7] W. Khan, T. Kumar, Z. Cheng, K. Raj, A. M. Roy, and B. Luo, "SQL and NoSQL Databases Software architectures performance analysis and assessments -- A Systematic Literature review," *arXiv.org*, Sep. 14, 2022. <https://arxiv.org/abs/2209.06977v1> (accessed May 08, 2023).
- [8] A. Meier and M. Kaufmann, *SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management*. Wiesbaden: Springer Fachmedien Wiesbaden, 2019.
- [9] W3C Recommendation, "SOAP 1.2 Part 1: Messaging Framework", <https://www.w3.org/2000/xp/Group/1/08/29/soap12-part1.html>
- [10] W3C Recommendation, "Web Services Description Language (WSDL) 1.1", <https://www.w3.org/TR/wsdl.html>
- [11] R. T. Fielding, "in Information and Computer Science," 2000.
- [12] "IEEE Guide for Loading Mineral-Oil-Immersed Transformers and Step-Voltage Regulators," *IEEE Std C57.91-2011 (Revision of IEEE Std C57.91-1995)*, pp. 1–123, Mar. 2012
- [13] 蘇偉府,黃世杰,"配電變壓器維護管理之研究",2016
- [14] 蘇偉府,黃世杰,劉志文,"配電變壓器設置地點之規劃設計研究",2013
- [15] "IEEE Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces," IEEE.
- [16] "IEEE Standard for Harmonic Control in Electric Power Systems," IEEE.
- [17] IEC 61400-21 Edition 1.0 (2001-12), Part 21: Measurement and assessment of power quality characteristics of grid connected wind turbines.

- [18] A. Elmoudi, M. Lehtonen, and H. Nordman, "Effect of harmonics on transformer's loss of life," in *Conference Record of the 2006 IEEE International Symposium on Electrical Insulation*, Jun. 2006, pp. 408–411.
- [19] H. F. M. Mantilla, A. Pavas, and I. C. Durán, "Aging of distribution transformers due to voltage harmonics," in *2017 IEEE Workshop on Power Electronics and Power Quality Applications (PEPQA)*, May 2017, pp. 1–5.
- [20] R. Santodomingo et al., IEC 61970 for Energy Management System Integration. 2016. doi: 10.1002/9781118755471.sgd094.

伍、附錄

一、國際期刊論文

二、國內電力研討會論文