軟體工程 作業五 B0229053 黃宇薇

* Service-Oriented Architecture
* 定義：

SOA服務導向架構是一種新興的系統架構模型，主要概念是針對學校或企業需求組合而成的一組軟體元件。組合的元素通常包括：軟體元件、服務及流程三個部份。當學校或企業面對外部要求時，流程負責定義外部要求的處理步驟；服務包括特定步驟的所有程式元件，而軟體元件則負責執行工作的程式。SOA 已成為現今軟體發展的重要技術，透過 SOA 讓異質系統整合變得容易，程式再使用度也提高。不必自行開發或擁有所有程式元件，發展者可以視其需要組合網路上最好的服務。不受限於特定廠商的產品功能或是平台，達到真正的開放性(Openness)。從分散式元件架構到 SOA概念上，SOA 如同物件導向、軟體元件等軟體技術一般，運用小的零組件組合成應用系統。但 SOA 強調的是如何將彼此關係鬆散的應用系統功能元件在網路上發行、組合及使用。SOA 具有下列技術特性：

1. 分散式架構 (distributed)－SOA 的組成元件是由許多分散在網路上的系統組合而來，可能是區域網路，也可能是來自廣域網路。例如網站服務技術 (web services) 就是運作 HTTP來相互連結的 SOA。如此的作法，也使得網站服務技術很快的就成為所有支援網際網路的系統平台均能使用的技術。
2. 關係鬆散的界面 (loosely coupled)－傳統的系統主要是將應用系統功能需求切割成相互關聯的小零組件：模組、物件或元件，發展者要花費極大的心力了解零組件是如何設計及使用，以確保不會違反零組件連接關係限制。如此一來，若要以不同零組件替換原始設計，就成為一件困難的事。SOA 的作法是以界面標準來組合系統，只要符合界面要求，零組件可以任意替換，大幅提高系統變更的彈性度。
3. 依據開放的標準 (Open standard)－使用開放標準是 SOA 的核心特色，過去的軟體元件平台如 CORBA、DCOM、RMI、J2EE 採用專屬協定作為元件連結的規範，使得不同平台的元件無法相通。SOA 則著重於標準與互動性，將可避免不同平台 (.NET web services 與 Java web services) 開發程式間相互整合的困擾。
4. 以流程角度出發 (process centric)－在建構系統時，首先了解特定工作的流程要求，並將其切割成服務界面(包括輸入與輸出資料格式)，如此其他的發展者就可以依據服務界面開發 (或選擇) 合適的元件來完成工作。

* 例子：

舉一個學校常用的例子來說明SOA在實際應用上帶來的可能性。假設我們要建立一個線上投稿的網站，網站提供的服務包括了線上投稿作業、論文分派作業、論文審查作業、線上註冊及報名作業等。傳統的方式我們會儘快找一個類似的網站，再儘快將其他類似網站的原始碼(source code)拿來修改，但其他類似網站的原始碼所執行的平台有可能不是架站者所熟悉的作業系統。若要讓架站者作一個客製化，並符合自己投稿主題的線上投稿的網站，可能要熟悉這個平台並修改網頁及測試，再加上別人的網站可能有一些bug，如果要做到毫無問題，除錯時間可能要花上三個月的時間。但是，如果我們導入SOA的架構的話，可能將來只要花費心力將作業服務模組化、物件化或元件化，然後將它們整合到網站中即可，不需要再花費時間和資源自己去維護一個線上投稿的網站，也不需要再自行建立和資料庫連結、報名付款機制等。這個網站就像是建立在SOA上，整合了這些web services元件的一個應用程式系統。更重要的是，透過如http、XML、SOAP 等產業標準開放式協定，不必擔心這些服務使用甚麼平台、甚麼技術來建立，而將來如果有更好的服務或服務提供者時，也可以輕易的將服務更換或更新。對系統開發者來講，可以快速輕鬆的將系統建構完成，將心思專注在規劃更好、更完善的系統上；對服務提供者而言，只要能設計出一個好的服務，它的潛在使用者市場將不再受到使用者平台的限制而有無限的可能。單就這類應用所呈現的美好遠景，應該可以解釋為什麼會到處聽到有人在談論SOA了。

* 結論

1. SOA 的實作，就是將所有程式邏輯及服務內容全部包裹在服務內部，並實作一個標準的介面與外部作溝通，這種做法跟傳統的元件導向做法非常類似，唯一的差別是介面定義的方式、資料格式、與溝通管道必須是產業標準 (http、XML、SOAP 等)。 也就是說只要能實作出這樣的介面，不論介面後面是什麼，都可使成為 SOA。
2. SOA能帶來的幫助，有以下好處：
3. 增加企業盈收，或提升學校的服務品質。
4. 提供可變動的網路服務型態。
5. 降低學校或企業的成本。
6. 降低開發服務的時間。
7. 整合學校或企業的網路服務技術資源。
8. 降低整體風險及意外。