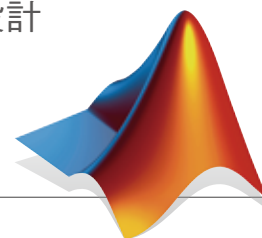


按步就班穩健邁向AUTOSAR 雙向工程(Round Trip)架構之路

利用Volcano VSA和MATLAB & Simulink工具解決軟體開發反覆設計之挑戰以符合AUTOSAR流程架構

作者：Guido Sandmann, MathWorks車輛產業行銷經理,
Michael Seibt, Mentor Graphics產品經理



汽車產業的電子開發向來是一個反覆設計 (iteration)的過程 — 甚至早在制定AUTOSAR之前即已如此。專案開發過程會不斷對需求進行修改或改善，設計的決策和架構也會跟著進行調整。AUTOSAR透過對不同設計文件的規範格式進行標準化，以明確定義這些過程步驟的執行方式。

同時，藉助許多先進工具的支援，AUTOSAR標準已經成功地應用於許多領域的產品開發。這些專案面臨的挑戰是，如何確保設計資訊不遺漏的前提下在各工具之間實現一致的資料交換。AUTOSAR尤其特別關注在開發個別軟體元件及功能軟體時，架構與模型化基礎設計 (Model-Based Design) 工具之間的互動，包含必須支援自上而下(top-down)和自下而上(bottom-up)的設計方法。

本文使用 MathWorks公司的MATLAB、Simulink和Embedded Coder (嵌入式程式碼轉碼器)以及 Mentor Graphics 公司的 AUTOSAR 開發工具 Volcano Vehicle Systems Architect (VSA) 作為例子示範，分成五個步驟示範架構與模型化基礎設計環境之間的互動性和互通性。

自上而下與自下而上工的作流程對比

在實務中，自上而下和自下而上工作流程都很常見。由於上述所提的反覆設計過程所需，這兩種方法一般不單獨使用，而是混合使用。

如果從頭開發一項新功能，最常用的方法是自上而下。這意味著一開始就使用設計工具來描述軟體架構，包括其軟體元件及介面規範，根據具體需求和資料可得性，內部行為（即可運行架構）也可會在此階段進行開發。利用該工具內建以及使用者自定義的一致性檢查和設計規則，可以協助檢查輸入資料的正確性和完整性，然後將最終匯出的規範格式導入到諸如 Simulink 這樣的設計工具中，在導入軟體元件規格的這一過程中，將可自動產生包含所有相關資訊（如介面或可運行架構）的模型框架。利用產生的模型框架，軟體工程師可以基於功能的需求，在 Simulink 環境中對功能行為進行建模。完成建模後，也可以借助 Embedded Coder(嵌入式程式碼轉碼器)產生與 AUTOSAR 標準相容的 C 程式碼，在此同時，導出的新軟體元件規範，繼而可由 VSA 導入以用於進一步整合。

另一方面，如果已經有既有的某個功能模型且已用於生產，它們還可被重複使用。首先必須為這些模型增加 AUTOSAR 特定資訊；之後，產生符合 AUTOSAR 標準的 C 程式碼和相應的 AUTOSAR 軟體元件規範，繼而由設計工具導入，完成進一步的處理。因此，自下而上的方法主要用於解決重複使用現有智慧財產權(Intellectual Property, IP)的問題。

如開頭所述，為了實現設計的改善或修改，需要對過程中不同步驟重複執行多次，因此，所用的工具必須支援反覆設計所需的開發過程。在此特定情況下，這些工具必須支援雙向工程(round-trip engineering)，這意味著架構和模型的設計工具間必須能夠進一步處理彼此的更新資料。

下文將詳細介紹雙向工程所必須執行的所有步驟。

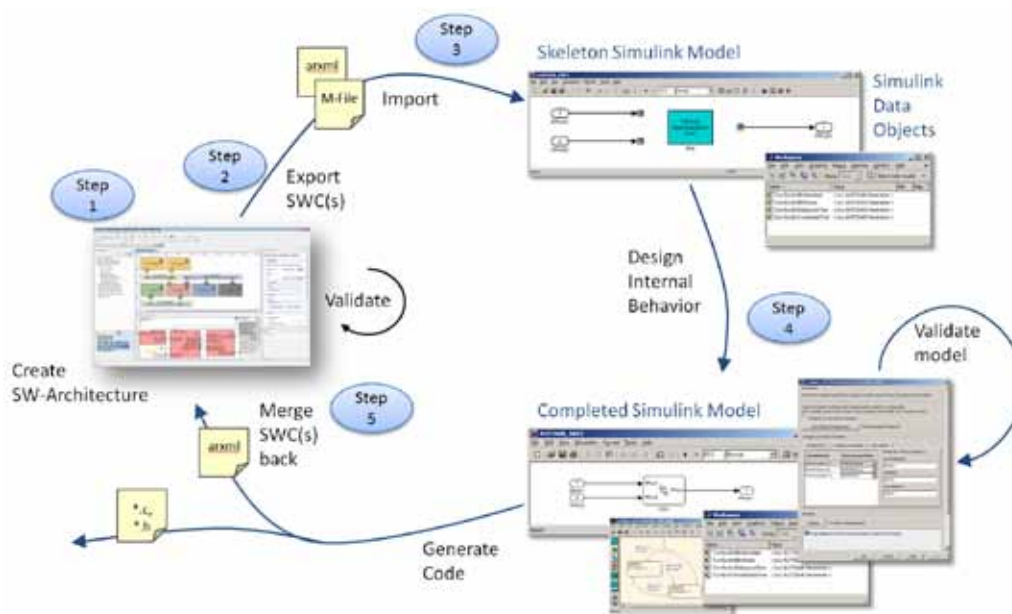


圖 1：利用Simulink 和VSA進行 AUTOSAR 雙向工程設計流程。

步驟 1：產生軟體架構和元件

一個依循 AUTOSAR 的設計工具（如 VSA），是可以產生軟體元件並將其連接到軟體架構的。根據所需的細微性，個別軟體元件的內部行為，可以被建成不同形式的模型，如可運行架構、接入點和執行環境(RTE, run-time environment) 等形式。

圖 3 顯示了座椅加熱系統例子的軟體架構（軟體構成），其中包括引用了軟體行為。設計工具提供針對特定領域的不同編輯器來建立各種複雜程度的設計，依據設計需求，使用者可以選擇樹狀編輯器、表格編輯器或圖形編輯器。

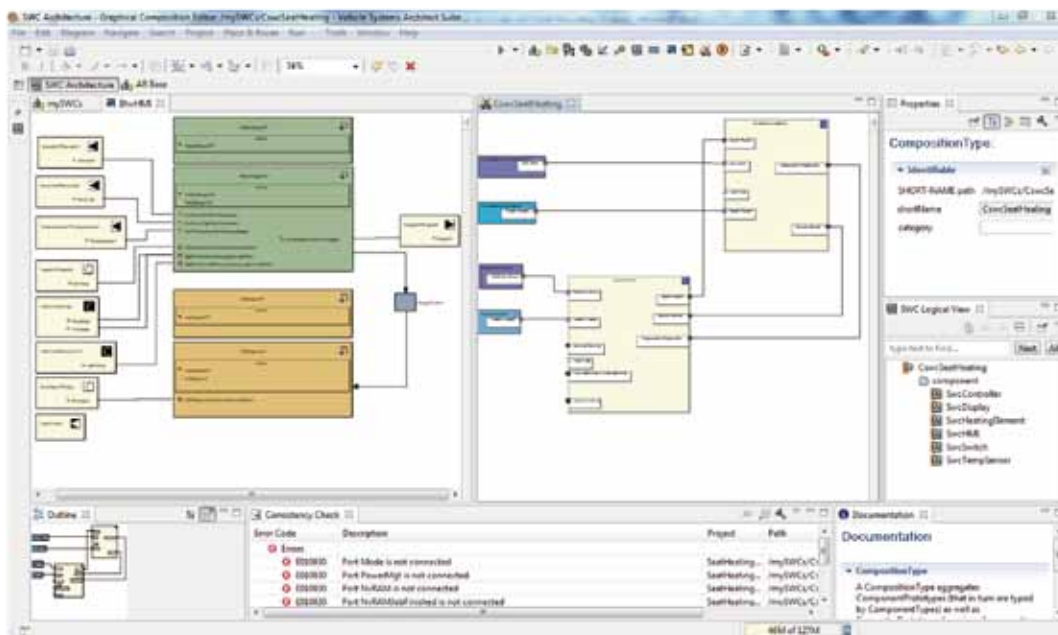


圖 2：VSA - AUTOSAR 設計工具

利用多項預先實現的一致性檢查，可以在架構的建模過程中連續檢查設計的完整性和合理性，這可由自訂的一致性檢查或設計規則檢查 (DRC)來進行補充檢查，現有軟體元件、介面規範和其他設計工件，也可以輸入成函式庫的形式，並在新設計中被重複使用。經由自動產生的全球唯一識別碼(Unique Universal Identifier, UUID)和指派的短名稱，每個 AUTOSAR 物件將獲得一個唯一標記，以便於在設計反覆運算過程中進行識別。

步驟 2：從設計工具匯出資料

VSA使用的元模型(metamodel)技術，是一個完整的AUTOSAR模型規範並支援完整的匯出功能，還能針對不同的 AUTOSAR 版本提供相應的模型到模型的轉換。在 VSA 的軟體架構內部完成特定軟體元件的定義後，可以採用 AUTOSAR 定義的標準規範來匯出這些元件。此外，它也支援MATLAB執行程式(script)，可輸出導入到後續的 MATLAB 和 Simulink 中。

步驟 3：將軟體元件規範導入到 Simulink

Simulink包含了介面函數，可以導入在步驟 2 中產生的工件，以及相應的介面和埠規範、內部運行的變數或有關運行架構（內部行為）的資訊，然後產生包括軟體元件結構在內的模型框架，作為功能呼叫子系統；這些變數包括資料類型，會在工作區(workspace)中自動產生。圖 4 表示，將座椅加熱系統的可運行和內部運行變數之軟體元件導入至Simulink中。

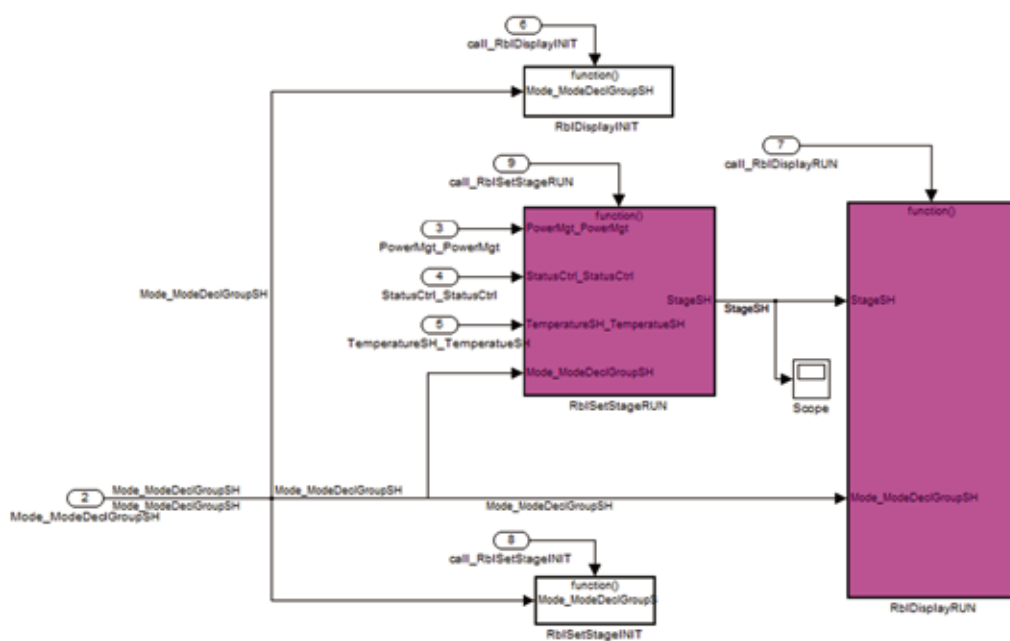


圖 3：將軟體元件(SWC)規範導入到 Simulink。

步驟 4：功能行為建模和程式碼產生

在接下來的步驟中，我們將透過模型化基礎設計(Model-Based Design)來實現功能行為。利用 Simulink和Stateflow(事件導向系統模擬軟體)，可以循序漸進地建立和改善模型，依專案而定該模型可用於多種用途。在最早期階段，可透過模擬來驗證功能行為，隨後可以借助快速硬體原型測試，將模型調整成熟到可以進行生產程式碼的產生，此時可以利用 Embedded Coder(嵌入式程式碼轉碼器)產生符合 AUTOSAR 相容的規範格式。在此過程中，慣用的Simulink 元素仍然可以使用，且可直覺對應到 AUTOSAR 工作中；最關鍵的是，模型化基礎設計及其相關的設計和驗證工具還是可以照常應用。接下來步驟是產生符合 AUTOSAR 相容的 C 程式碼和AUTOSAR 軟體元件規範的 arxml 格式，在下一步驟中可以導入並與VSA 進行整合。

步驟 5: AUTOSAR 的比較與合併

為了實現雙向工程，需要一個選項來將修改後的設計資料回饋到 AUTOSAR 設計工具，並將其與原始設計資料進行同步。為達此目的，VSA 提供了 AUTOSAR 比較與合併的功能，利用該功能，可以在 AUTOSAR 物件平面上比較和具體地合併設計物件。可以根據全球唯一的識別碼 (Unique Universal Identifier, UUID) 或短名稱執行此操作。

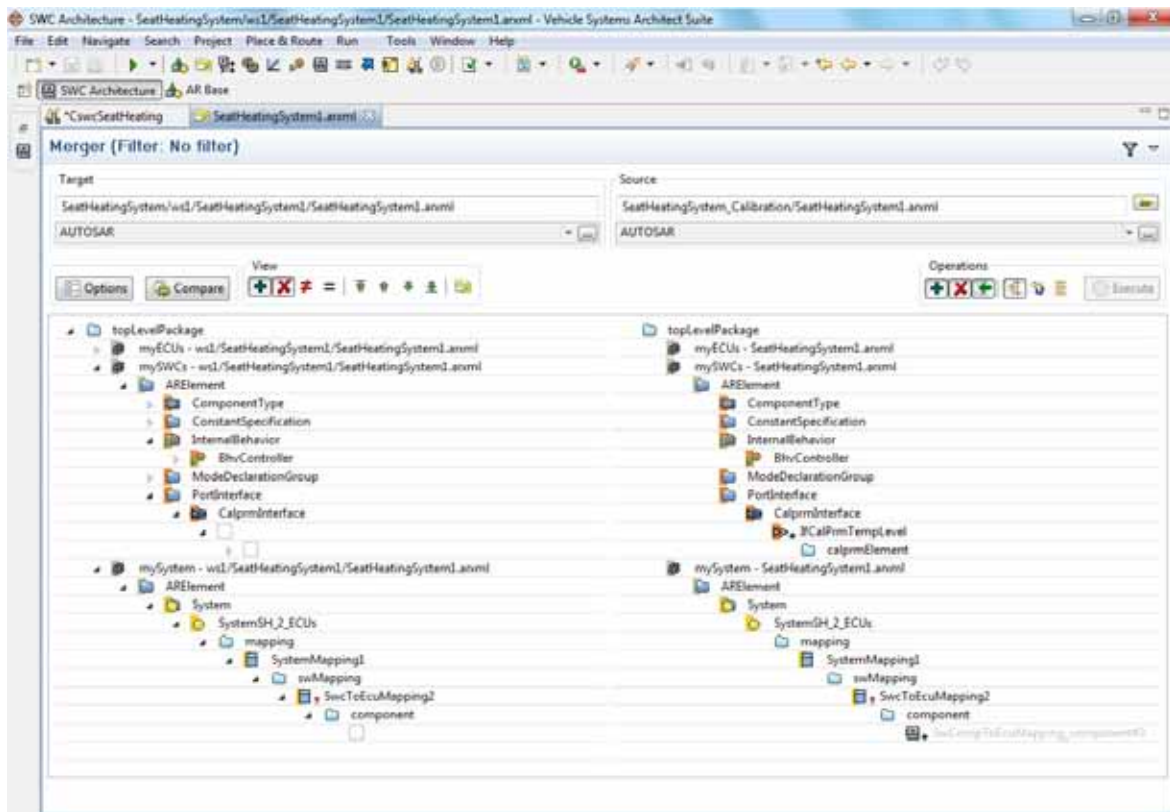


圖 4：VSA 的比較與合併工具。

其他額外的"規則"

目前，各類公司都在使用 AUTOSAR 和協調一致的工具實現可靠的雙向工程。然而，這必須在遵守其他額外 "規則" 的前提下才能實現。

使用者角色和權利必須綜合考量現有流程和新定義流程間的協調。必須利用內部 "風格指南" (如建模指導原則和命名習慣) 來具體限制 AUTOSAR 框架內的自由度。最後，必須與第一階供應商協調這些風格指南，以便在內部以及供應鏈上的企業之間實施雙向工程。再次強調，軟體工具在 OEM 公司及其供應商之間的 AUTOSAR 工件交換中，確實扮演著相當關鍵的作用。