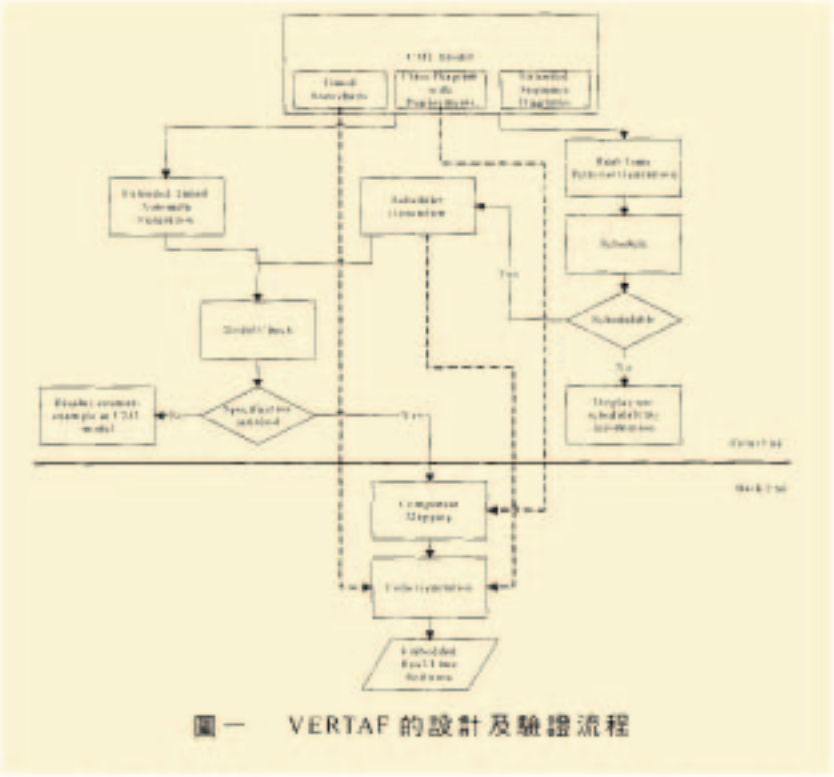
即時嵌入式軟體之合成工具設計

熊博安　國立中正大學資訊工程學系
計畫編號：NSC-93-2213-E-194-002

一、摘要

針對即時嵌入式系統之軟體開發需求我們設計了一個物件導向應用程式框架。我們採用目前廣泛使用的 UML（Unified Modeling Language）設計語言來建立系統規格模型。透過物件導向的設計來提高嵌入式系統的開發速度、可重複使用性及除錯能力。這個框架也整合了即時排程及正規驗證功能。驗證功能可以確保產生的嵌入式軟體符合系統規格。而排程功能可以確保產生的嵌入式軟體符合系統的即時特性，即滿足所有工作的時間限制。使用這個框架我們實作一個應用實例，即「門禁系統」。在實作過程中可以明確地看出使用這個框架來開發嵌入式軟體的優點。同時，也證明了框架的正確性及便利性。

二、研究動機與目的



圖一　VERTAF 的設計及驗證流程

在我們生活周遭，嵌入式系統佔了越來越重要的地位，小如行動電話，大如汽車，其中都包含了嵌入式系統，我們已經不知不覺中逐漸習慣依賴這些設備，因此，嵌入式系統的設計成為很重要的課題。但是嵌入式系統設計的特性在於它的行為很複雜卻又擁有非常緊迫的上市時間限制。傳統設計嵌入式系統必須仰賴工程師的經驗，因此如果是面臨一個新的平台，往往必須重新學習新的軟硬體。而程式完成後的除錯更是困難，傳統使用測試為基礎的除錯技術非常花費時間卻不能保證整個程式經過測試流程以後就一定不會出錯。

由於以上的限制造成嵌入式系統設計時的許多困難。為了解決這些問題，我們提出了一個物件導向嵌入式軟體開發框架 VERTAF。在其中整合了 UML 塑模技術，物件導向技術，軟體排程技術，軟體驗證技術及自動化程式碼生成技術。

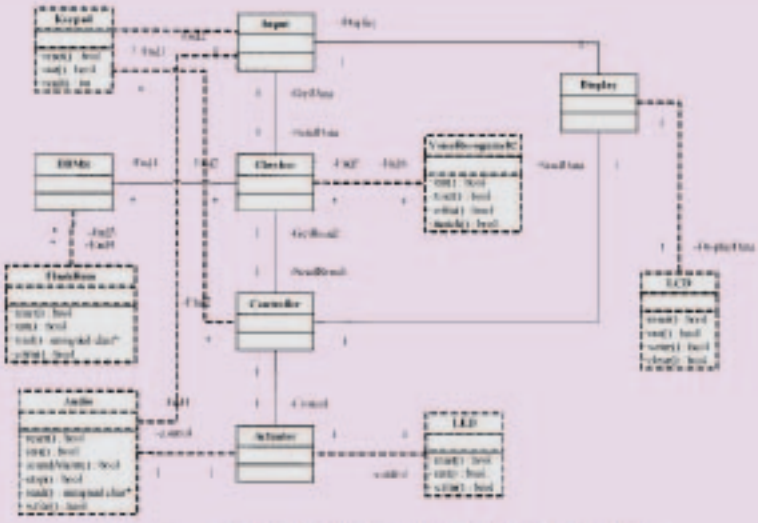
三、研究內容

這個系統的架構完整流程如圖一所示，實線代表的是工作流程，當一個工作完成接下來就進入箭頭指向的工作繼續執行。虛線代表的是資料流程，被虛線箭頭指向的工作需要虛線來源的這筆資料來完成工作。

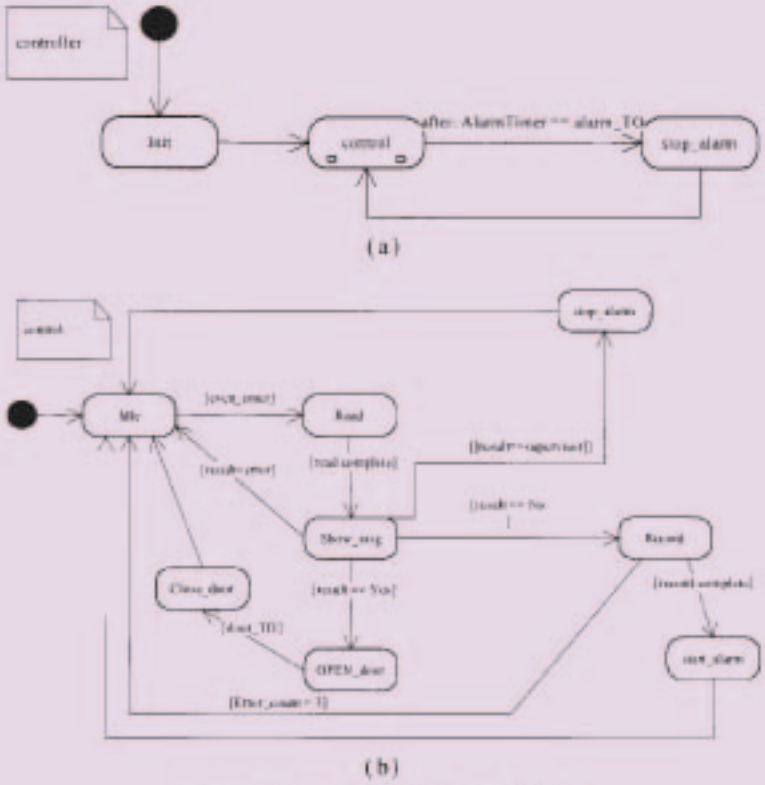
軟體合成可以分成兩個階段的程序：與平台無關的軟體建構階段和與平台相關的軟體實現階段。我們把這兩個階段當成前端和後端階段。前端又可再細分成三個階段，分別為 UML 模型輸入階段，嵌入式即時軟體排程階段，和正規驗證階段。而後端也分成兩個階段，分別是元件對應階段及程式碼生成階段。

(一) UML 模型輸入

我們選用 UML 中的三種圖作為系統的輸入模型，這三種分別是類別圖（Class Diagram），狀態圖（State chart），及順序圖（Sequence Diagram）。針對即時系統的需求對這三種圖進行了小幅的修改，以描述更多即時嵌入式系統相關的行為。



圖二　增加了佈署資訊的門禁系統類別圖



圖三　控制器類別的真實狀態圖

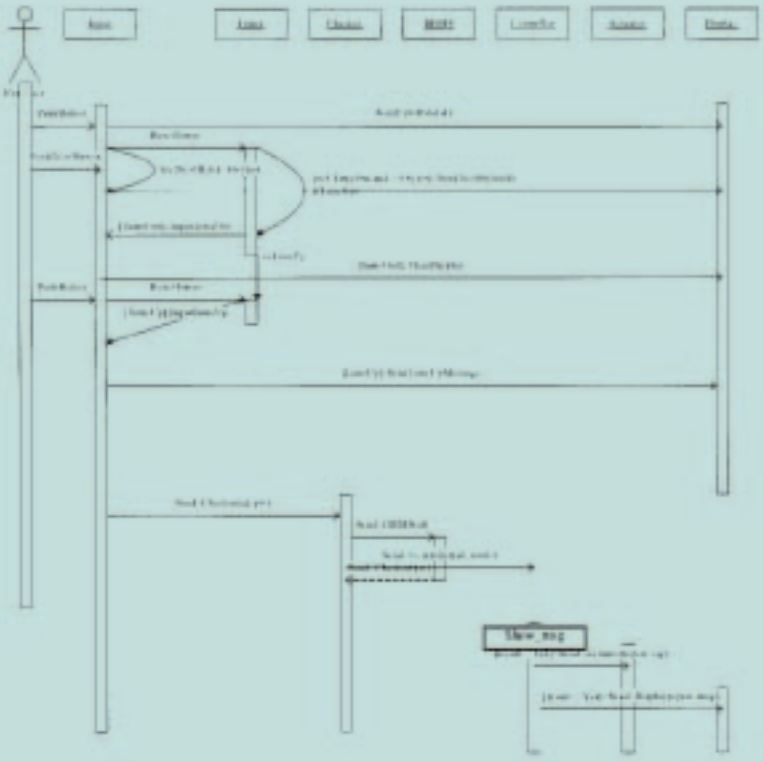
我們在類別圖中加入了軟硬體之間佈署的資訊。圖二是一個門禁系統應用程式的範例，圖中包含六個軟體類別分別佈署在六個硬體類別上，以虛線表示軟硬體物件之間的關聯，實線作為軟體物件之間的關聯。

在狀態圖中，我們針對時間驅動函式增加了三個關鍵字，分別是 start、stop 和 reset。它們分別可以啟動、結束及重設一個時間驅動函式。除了這三個關鍵字以外，我們另外還增加一個關鍵字 time-out，它可以用來指定函式的即時特性。圖三是門禁系統中，描述控制器類別的真時狀態圖（timed statechart），這是兩層式的階層架構，用來描述計時器中斷。

在順序圖中，我們增加了一個稱為狀態標籤（state marker）的符號，其中的狀態名稱是相對於狀態圖中的狀態名稱，它可以讓狀態之間的訊息傳送順序更加明確，也可提供更多資訊給後續的排程器。圖四是門禁系統的其中一個順序圖，描述一個正常使用下的情形，也就是一個合法的使用者通過語音辨識之後可以進入大樓的情形。在控制器上名稱為 *Show_msg* 的狀態標籤指出，當控制器在 *Show_msg* 的狀態時才可以發送 *Send_Actuator()* 訊息。

（二） 嵌入式即時軟體排程

當設計一個嵌入式系統時，需要考慮到這個系統是否有任何的時間限制，這些時間限制可能是區域性或全域性的。



圖四　門禁系統的順序圖--正常使用的情形

我們採用的排程方法為「延伸半靜態排程」（Extended Quasi-Static Scheduling, EQSS）及「類似動態排程」（Quasi-Dynamic Scheduling, QDS）排程方法，這二個方法針對系統中指定的全域時間限制及區域時間限制將程序進行排程，在排程後調整過先後順序的程序，讓系統可以依序進行而不會牴觸系統的各種時間限制，用以達到即時系統的需求。為了使用這個排程方法，我們必須先要將一開始的 UML 模型轉換成它的輸入模型。它使用的輸入模型是 Petri-net。

（三） 正規驗證

在系統驗證部分我們採用的是正規驗證的方法，使用模型驗證（model checking）作為我們的驗證方式，模型驗證是一個以狀態為基礎的分析程序，可以讓我們知道系統是否滿足時間限制。模型驗證需要有一個正規的系統模型，及一個正規的時間性質規格。UML 模型會被轉成延伸真時狀態機，而物件限制語言（Object Constraint Language，簡稱 OCL）的性質則轉成真時計算樹形邏輯公式（TCTL formulas）來表示。

根據使用者給的 UML 模型，我們藉由平面化（flattening）的方法，為每個狀態圖產生一個或多個正規延伸真

時狀態機模型。當我們從使用者指定的 UML 狀態圖產生得到一組延伸真時自動機的模型，再依據排定的階段，將它們合併在一起，最後產生一個狀態圖形（state-graph）。所用來做模型驗證的主要核心工具，叫做「狀態處理器」（State Graph Manipulators，SGM）。在系統中，有二個部份的系統性質需要被驗證：

1. 系統定義的性質，包括死狀態、死結（deadlocks）以及活結（livelocks）。
2. 使用者在物件限制語言中所定義的性質。這二部份的性質都要被自動化地轉成以真時計算樹形邏輯（TCTL）的格式來表示，以便輸入到 SGM 這個驗證工具來做模型驗證。

（四）元件對應

此階段開始有較多的硬體相關部份，在類別圖中的所有硬體類別，而且這些硬體類別屬於某些類別資料庫(class libraries)。這個階段會產生硬體系統和作業系統的設定並且自動產生設定檔、makefiles、標頭檔及相依性質檔案。而這些對應的硬體類別 API 將會在編譯時自動連結進來。

（五）程式碼生成

這部分要進行的工作是將以上的程式生成可執行的程式碼。我們採用的方法是由 Miro Samek 提出的量子程式法。這種做法是將每個物件中的狀態圖都當作獨立的單元，每個單元都自行運作，單元間只藉由訊息的傳遞來建立關聯。使用這種做法，我們可以直接將使用者輸入的狀態圖轉成相對應的程式。

四、結論與應用範圍

本計畫所發展之工具，非常適用於即時嵌入式系統開發與設計。設計師以 UML 為輸入的模型，可以減少設計師建立模型或學習新語言的時間，並且利用 UML 的特性完整的描述系統的行為。物件導向技術使得軟體元件能夠擁有重複使用性，設計師不需要花很多的時間去適應新的平台，他可以直接使用現存的軟體元件加以整合，大大地減低了開發的時間。軟體排程技術確保產生的程式能夠符合使用者指定的時間限制，對於即時嵌入式系統的設計有很大的幫助。軟體驗證技術可以讓設計者經由正規驗證的方式，完整地驗證系統行為的正確性，避免系統執行可能發生的錯誤。自動程式碼生成技術讓使用者可以減少自行撰寫程式碼的需要，可以減少人為的程式錯誤及撰寫程式碼的時間。

作者簡介

熊博安
國立中正大學資訊工程學系副教授
國立台灣大學電機工程博士
專長：軟硬體共同設計；系統設計與正規驗證；即時嵌入式系統；晶片系統；平行與分散式系統
電話：(05)2720411 轉 33119
傳真：(05)2720859