

基于 AUTOSAR 方法论的应用组件配置

徐鑫朋¹, 王 翔¹, 陆建华², 徐 军¹, 张晓先¹

(1. 华东计算技术研究所, 上海 200233; 2. 上海科技管理干部学院, 上海 201800)

摘 要: 针对目前主流商用的汽车电子软件设计工具链在通用性和重用性方面存在的不足, 根据汽车开放式系统架构标准分析应用组件配置过程, 提取应用组件配置的核心元素, 以核心元素建立对象, 实现一个基于 ReWorks 操作系统的国产汽车电子组件开发平台。该平台对汽车电子软件工具链进行了完善, 具有可插拔、易移植等特性。

关键词: AUTOSAR 方法论; 应用组件配置; 汽车电子; 运行时环境; 虚拟功能总线

Application Component Configuration Based on AUTOSAR Methodology

XU Xin-peng¹, WANG Xiang¹, LU Jian-hua², XU Jun¹, ZHANG Xiao-xian¹

(1. East-China Institute of Computer Technology, Shanghai 200233, China;

2. Shanghai Institute of Science & Technology Management, Shanghai 201800, China)

【Abstract】 Aiming at the problem that the main and commercial tool chains for designing automotive electronics software are lack of universality and reuse, based on the standard of AUTomotive Open System Architecture(AUTOSAR), this paper analyzes the process of application component configuration and extracts the kernel elements. Modeling of the elements, a platform for developing domestic automotive electronics software based on ReWorks operating system is realized. And the platform has the pluggable ability and portability.

【Key words】 AUTomotive Open System Architecture(AUTOSAR) methodology; application component configuration; automotive electronics; Run Time Environment(RTE); Virtual Function Bus(VFS)

1 概述

汽车开放式系统架构(AUTomotive Open System Architecture, AUTOSAR)定义了一整套支持分布式的、功能驱动的汽车电子软件开发方法和电子控制单元上的软件架构标准化方案。如图1所示, 在之前的开发模式下, 软件层和硬件层存在很多交叉和依赖关系, 软件和硬件模块必须串行开发; AUTOSAR提供的虚拟功能总线(Virtual Function Bus, VFB)的作用, 就是将硬件层与软件层的通信进行抽象的封装, 使两层隔离而可以并行开发设计, 缩短开发周期, 减少研发成本, 提高了软件的可重用性, 确保了软件的高质量和高效率。

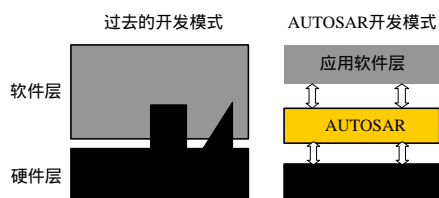


图1 AUTOSAR 开发模式

VFB是运行时环境(Run Time Environment, RTE)的集合, 由AUTOSAR提供的所有通信机制的抽象, 使得系统可以在抽象层进行设计, 而不必关心实际的硬件环境, 当一个具体系统的软件组件之间的连接被定义好之后, 通过VFB可以在开发的较早时期将它们虚拟集成^[1]。

RTE是AUTOSAR ECU结构中的核心部分, 每个具体的ECU单独生成一个RTE, 用来确保对应的RTE是最优的。

RTE主要为每个ECU实现VFB的标准接口, 提供组件间通信的基础服务, 使组件访问BSW资源更加容易。

目前, 主流商用的汽车电子软件设计工具链主要基于OSEK规范, 如Telelogic发布的Rhapsody OSEK。部分工具开始向AUTOSAR标准方向发展, 如美国明导科技发布的基于“AUTOSAR 3.0”的车载软件开发工具Volcano Vehicle Systems Architect(VSA)、在IBM Rational Systems Developer中也进行AUTOSAR系统建模, 它们都具有基于AUTOSAR的配置和代码生成功能, 但在通用性和重用性方面存在不足。本文在ReWorksOSEK操作系统的基础上, 采用基于插件的方法实现了一个基于AUTOSAR标准的汽车电子应用组件开发平台, 在拥有其他工具的优点的同时, 具有可插拔、易移植等特性。

2 AUTOSAR 配置过程

根据AUTOSAR方法论, 一个完整的基于AUTOSAR规范的配置生成过程主要分为3个部分: 应用组件配置, 系统配置和ECU配置, 如图2所示。

在配置过程中三者并没有先后之分, 系统配置的输入包含ECU配置和应用组件配置的相关模块, ECU配置又反作用于系统配置。3个配置层面相互支撑, 缺一不可, 共同构成统一的整体。

作者简介: 徐鑫朋(1985-), 男, 助理工程师、硕士, 主研方向: 嵌入式操作系统; 王 翔, 工程师; 陆建华, 讲师、硕士; 徐 军, 助理工程师; 张晓先, 高级工程师

收稿日期: 2010-04-20 **E-mail:** xxp1002@126.com

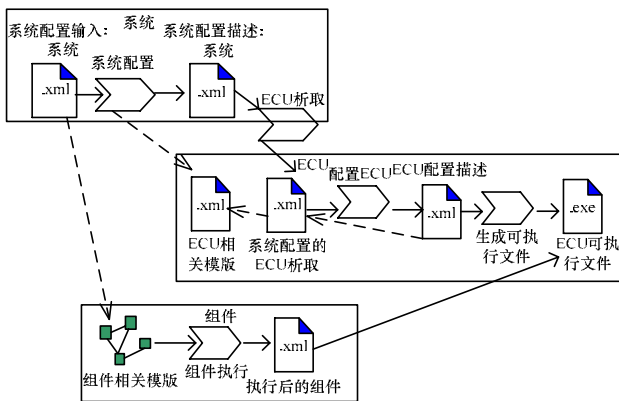


图2 AUTOSAR 方法论完整配置过程

3 应用组件配置

应用组件配置是针对应用软件层的配置过程。在AUTOSAR中,应用软件层在RTE之上,包含应用软件组件和传感器-触发器软件组件^[2]。其中,应用软件组件是独立于ECU的,而传感器-触发器是依赖于ECU硬件的,所以很容易根据性能或者效率进行重定位。这就意味着,需引入一系列的配置规则来对AUTOSAR软件组件配置进行有效的约束,使其可以适配给任意一个ECU配置,实现应用组件的配置开发与底层硬件平台相独立的特性^[3]。为此,RTE为AUTOSAR应用组件提供了标准的通信机制和服务,开发人员可以在对硬件一无所知的情况下进行应用软件的开发,并将这个软件应用在任何符合AUTOSAR标准的ECU中。由于传感器-触发器软件组件可直接获取本地ECU抽象层的地址,因此要访问远程的ECU抽象层,需要通过一个广播远程ECU信息的传感器-触发器软件组件作为中间媒介。一个应用组件可以自由任意地配置,AUTOSAR软件组件按照类型定义方式,定义了它的组件接口。组件类型在被特定ECU配置时实例化。组件类型可以在同一个ECU上被多次实例化,这种情况被称作“多重实例化”。RTE支持“每次实例化内存段”,这样就使得组件的每次实例化都能保持其私有状态。RTE同时支持软件组件源代码的实例化和目标代码的实例化。

AUTOSAR标准的汽车电子软件应用组件配置过程如图3所示,可见应用组件配置包括一个输入是BSW的服务描述,一个配置过程,即应用组件描述,以及一个代码生成过程。

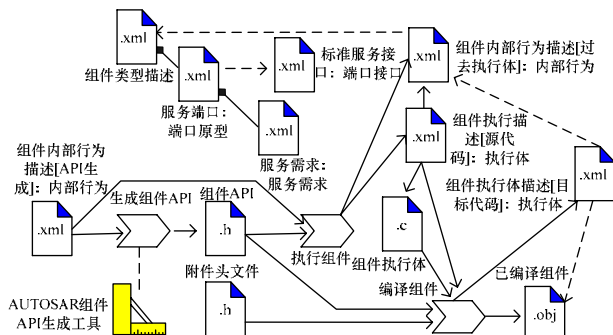


图3 AUTOSAR 方法论应用组件配置过程

BSW服务描述定义了所有RTE为封装BSW服务接口而预定义的相关接口,因为这些接口都是AUTOSAR标准接口,所以开发人员可以在对硬件一无所知的情况下进行应用组件的开发,并将这个组件应用在任何符合AUTOSAR标准的ECU中,从而实现了应用组件的配置开发与底层硬件平台相

独立的特性。

应用组件配置生成器支持组件级的工程决策。配置应用组件时一个重要的决策是需要将应用组件所需的端口信息以及内部运行信息描述完整,RTE将根据这些描述信息为应用组件生成运行时所支持的代码,这些生成的代码在RTE运行时,进行如下操作:

- (1)为应用组件分配所需的系统资源,如消息号、任务运行时环境等。
- (2)负责具体执行应用组件间通过端口进行的通信。
- (3)根据应用组件描述的信息,在适当时刻为应用组件提供事件或调度。

应用组件配置描述包括完整的应用组件端口、接口以及应用组件间连接的详细定义的内容。通过应用组件级的配置,可以完成对每一个应用组件的配置。这些应用组件可能包含了完整的汽车功能,也可能只包含部分功能。基于应用组件配置描述,完成基于AUTOSAR组件的应用软件的实现,最终生成应用组件的XML描述文件和相关的可执行代码。

4 基于AUTOSAR方法论的应用组件实现

4.1 应用组件核心元素定义

应用组件配置模型中的核心概念包括了软件组件、接口、端口和内部行为定义等,它们的关系如图4所示。

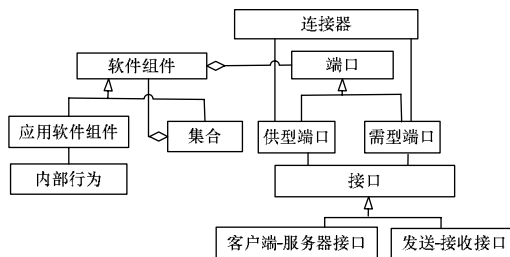


图4 应用组件核心元素

4.1.1 软件组件

软件组件是一组完成相关功能的代码集合,是分布在ECU上的最小代码单元。AUTOSAR中定义了各种软件组件以适应不同的需求,最普通的软件组件类型是应用软件组件,它是指实现全部或部分应用功能的原子软件组件,可以使用所有的AUTOSAR通信机制和服务。和具体传感器或触发器等硬件直接相关的原子组件被称为传感器-触发器软件组件,它通过I/O端口直接和ECU抽象层进行交互。组件和组件的集合构成一个组合组件,组合组件内部的原子组件间通过连接器进行端口连接,组合组件可以有自己对外通信的端口,也可以使用所有的AUTOSAR通信机制和服务。还有服务组件类型,它通过标准接口提供标准服务,这些组件直接和底层的某个BSW模块交互。除此之外,还有校准参数组件、ECU抽象组件、复杂设备驱动组件。

4.1.2 端口

端口分为8种,分别为支持服务型的由发送-接收接口实例化的发送端口、接收端口和非支持服务型的由发送-接收接口实例化的发送端口、接收端口,支持服务型的由客户端服务器接口实例化的客户端端口、服务器端口和非支持服务型的由客户端服务器接口实例化的客户端端口、服务器端口。

4.1.3 连接器

软件组件的端口间的连接关系通过连接器来实现。因为连接关系存在于软件组件的外部,需设计一个集合概念,所有的连接器存在于集合内部。一个连接器必须同时连接一

个供型端口和一个需型端口，并且这一对端口必须是同一个接口实例化出来的。

4.1.4 集合

集合是一个继承于软件组件的概念。从外观上看，集合也可以有端口。集合的内部可以包含其他集合或软件组件，以及表示这些集合或软件组件的端口之间连接关系的连接器。集合没有实现代码，也没有内部行为定义，不受软件组件必须分配在同一个上的限制。

4.1.5 组件接口

组件接口是独立于软件组件的概念。软件组件的通信通过端口完成，而端口由接口类型实例化产生。一对相互连接的端口，必须一个是提供数据或服务的供型端口，另一个是接收数据或请求服务的需型端口，且这对端口必须由同一个接口定义来实例化产生。接口分为发送-接收接口和客户端-服务器接口 2 种。

4.1.6 组件内部行为

在软件组件中，端口是软件组件的外观，即软件组件与其他软件组件之间通信的内容和方法定义。依据这些信息，可以知道如何组织软件组件间的通信。

组件内部行为模型用来描述组件使用上述通信机制的方法，根据内部行为模型，可以知道如何为软件组件执行通信机制。软件组织需要交给运行时环境的信息包括运行实体和事件，根据这些定义可以得到通信响应方式。

4.2 应用组件配置实现

根据 AUTOSAR 方法论，以及应用软件组件的核心元素定义，基于 Eclipse 架构对其进行软件实现，如图 5 所示。Basic Element Tree 以树结构的形式对应用组件进行详细配置，节点包括组件类型、接口、组合组件和数据类型，其中在组件类型内部，详细定义每个组件的输入端口(RPort)、输出端口(PPort)；在接口内部定义不同类型的接口(Interface)，以及接口内部的操作和数据；组合组件是原子组件的集合。

当选中某个组件时，右侧显示选中组件的详细信息，可对选中组件进行配置。在树型结构中可以增加、删除各种组

件、接口、操作等，所做修改按照规范保存在*.arxml 文件中。也可以将*.arxml 文件导入到工程中，工程会对其进行解析，按照结构生成符合规范的组件树。

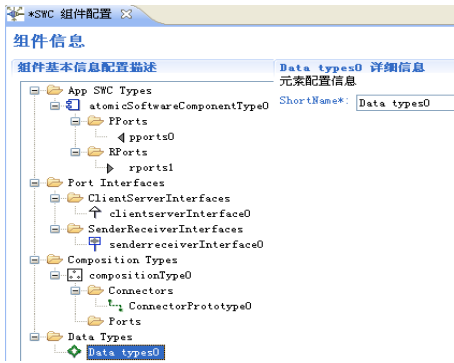


图 5 应用组件配置模块的功能操作

5 结束语

AUTOSAR 是一套统一标准的规范，当底层的硬件配置升级而不需改动整个系统。AUTOSAR 架构有利于车辆电子系统软件的更新，并为高效管理愈来愈复杂的车辆电子、软件系统提供了一个基础。此外，AUTOSAR 在确保产品及服务质量的同时，提高了成本效率。本文介绍了使用 AUTOSAR 方法论进行应用组件配置的过程，并根据 AUTOSAR 规范设计实现了具有可插拔特性的汽车电子应用组件开发平台。下一步将会对系统配置、ECU 配置模块进行研发设计，与应用组件配置模块一起形成一个完整的基于 AUTOSAR 标准的汽车电子集成开发环境。

参考文献

[1] 龚进峰. 几种新兴汽车电子技术[J]. 世界汽车, 2004, (2): 30-31.
[2] 李晓军. 基于量子框架的开放式汽车电控系统体系架构及应用研究[D]. 杭州: 浙江大学, 2008.
[3] 杨国青. 基于模型驱动的汽车电子软件设计开发方法研究[D]. 杭州: 浙江大学, 2006.

编辑 顾逸斐

(上接第 239 页)

表 1 2 个 C6455 间的数据传输性能

特性	Br=1.25 Gb/s	Br=2.5 Gb/s	Br=3.125 Gb/s
数据传输量/Gb	12.5	12.5	12.5
传输时间/s	14.50	7.25	5.80
传输速率/(MB·s ⁻¹)	110	220	275
传输效率/(%)	86.0	86.0	86.4

当 3 片 DSP 同时向一片发送数据时，仍然能像 2 片一样稳定传输。

3.2 DSP 间的 SRIO 加载

以 TMS320C6455 为例研究 SRIO 加载，C6455 有 6 种加载模式：no boot，HPI/PCI，EMIF，主 I2C，从 I2C，SRIO。选择哪一种引导模式是通过复位和上电时采样管脚 BOOTMODE[3:0]决定的。当 BOOTM3=1、BOOTM3=X 时，选择的是 SRIO。

加载的文件格式不能为.out，必须将其解析，提取有用的部分。可以使用 TI 提供的十六进制转换工具，转换生成二进制形式的 boot table。

上电后从 DSP 的初始化配置是通过固化在其内部的 ROM 完成的。主 DSP 完成对 SRIO 端口的初始化配置后，主从 DSP 间相互发送同步信号，直至链接建立。主 DSP 通过 Flash 加载，而从 DSP 是主 DSP 通过 SRIO 加载完成的。

4 结束语

本文对多处理器的 SRIO 进行了深入研究，实现了多个处理器间 SRIO 的通信，效率达到 86%。

RapidIO 代表了总线技术的发展方向，随着半导体与通信技术的发展，RapidIO 互连必将成为嵌入式系统互连的最佳选择。

参考文献

[1] 林玲, 蒋俊, 倪明. RapidI/O 在多处理器系统互连中的应用[J]. 计算机工程, 2006, 32(4): 244-246.
[2] 尹亚明, 李琼. 新型高性能 RapidI/O 互连技术研究[J]. 计算机工程与科学, 2004, 26(12): 85-87.
[3] Texas Instruments. TMS320C645x Serial RapidI/O[Z]. 2006-03.

编辑 张正兴