

基于PREEvision和MATLAB开发AUTOSAR软件组件方法及实践

李长洲 陈兴 李超 高仕宁 王长乐 曾志豪

(中国第一汽车股份有限公司研发总院, 长春 130013)

【欢迎引用】李长洲, 陈兴, 李超, 等. 基于PREEvision和MATLAB开发AUTOSAR软件组件方法及实践[J]. 汽车文摘, 2025(3): 1-7.

【Cite this paper】LI C Z, CHEN X, LI C. Method and Practice of Developing AUTOSAR Software Components Based on PREEvision and MATLAB[J]. Automotive Digest (Chinese), 2025(3): 1-7.

【摘要】为了探究SOA架构开发方法及流程,通过方法研究、流程分析及开发实践,提出一种面向服务的架构框架下软件组件的开发方法,分析了符合AUTOSAR规范的软件组件开发流程。基于提出的方法及流程给出了从需求分析、服务设计、架构设计、应用开发、软件集成与测试的开发实践。通过应用证明了该开发方法有助于主机厂更高效地实现软件模块的集成和复用,具备很好的迭代和扩展性。

关键词:面向服务的架构;PREEvision;MATLAB;软件组件;软件架构

中图分类号:U463.61 文献标志码:A DOI: 10.19822/j.cnki.1671-6329.20240177

Method and Practice of Developing AUTOSAR Software Components Based on PREEvision and MATLAB

Li Changzhou, Chen Xing, Li Chao, Gao Shining, Wang Changle, Zeng Zhihao
(Global R&D Center, China FAW Corporation Limited, Changchun 130013)

【Abstract】In order to explore SOA architecture development methods and processes, through method research, process analysis and development practice, a software component development method under the framework of service-oriented architecture is proposed, and the software component development process conforming to AUTOSAR specification is analyzed. Based on the proposed method and process, the development practice from requirement analysis, service design, architecture design, application development, software integration and testing is given. The application proves that this development method is helpful for Oems to realize the integration and reuse of software modules more efficiently, with good iteration and expansibility.

Key words: Service-oriented architecture, PREEvision, MATLAB, Software component, Software architecture

0 引言

随着汽车“新四化”(智能化、网联化、电动化和共享化)的快速发展,智能汽车的功能愈加丰富,车辆电子控制单元(Electronic Control Unit, ECU)数量不断增长,系统间的交互越来越庞杂,对整车电子电气架构的要求也越来越高^[1]。传统汽车电子电气架构开发方法大多数是基于功能的、面向信号的传统架构,周期性发送固定信号至另一个控制器,基于静

态交互的软件无法满足当前快速迭代的功能需求。此外,还存在软件模块复用率低、与硬件存在强耦合关系、无法做到无缝移植、开发周期长以及迭代成本高等缺点^[2]。在软件定义汽车的大背景下,面向服务的架构(Service-Oriented Architecture, SOA)应运而生,其降低了模块与模块间、系统与系统间的耦合度^[3]。软件模块间使用明确定义的接口进行交互、模块组件化的方式,能够有效降低模块间的依赖性,延长软件模块生命周期,增强软件扩展能力,提升整体

的开发效率,提供了全新的汽车电子电气架构设计路线。

在SOA转型背景下,很多主机厂、供应商进行了SOA架构的开发探索。东风汽车技术中心提出基于模型的汽车开放系统架构(Automotive Open System Architecture, AUTOSAR)应用层软件开发环境^[4]。上汽乘用车提出一种基于PREvision的以太网SOA设计方法^[5]。东信创智科技有限公司提出一种基于PREvision的SOA设计^[6],力争在软件定义汽车时代取得技术领先。

针对面向服务的架构设计方法,各传统车企大多处于探索实践阶段,在这个过程中采用合适的方法、规范的流程、高效的工具链可以有效提升开发的质量与效率。本文在SOA架构框架下,探究整车AUTOSAR软件组件的开发方法、软件架构、开发工具及开发流程。基于完整的开发实践印证了方法流程的可行性,对新一代电子电气架构开发具有一定的借鉴意义。

1 AUTOSAR 软件组件开发方法

1.1 SOA 与 AUTOSAR

SOA是一种高内聚、松耦合的服务架构,服务间通过精确定义接口进行通讯,不涉及底层编程接口和通讯模型^[7]。接口从产品能力(Product Capacity, PC)角度出发,采用中立的方式进行定义,其独立于实现服务的硬件平台、操作系统和编程语言。该架构在不同系统中的服务可以以一种统一和通用的方式进行交互,便于实现软硬解耦^[6]。上层应用开发者无须关注底层硬件架构,而专注于应用开发。

SOA的开发需以AUTOSAR平台为依托,可扩展标记语言(AUTOSAR Extensible Markup Language, ARXML)文件是开发过程中重要的配置文件之一。将ARXML文件导入配置工具,可以实现对经典平台(Classic Platform, CP)及自适应平台(Adaptive Platform, AP)AUTOSAR组件软件的设计与配置。通过结合SOA、AUTOSAR和ARXML,汽车厂商可以和供应商更好地实现软件模块的复用和集成,SOA提供了灵活的架构,简化了模块之间的通信,而AUTOSAR和ARXML则提供了标准化的参数描述和接口规范,促进了模块间的互操作性。

1.2 AUTOSAR 软件架构

AUTOSAR软件整体框架为分层式设计,以中

间件(Runtime Environment, RTE)为界,隔离上层的应用层(Application Layer)与下层的基础软件(Basic Software)^[8]。AUTOSAR体系架构分层标准如图1所示。

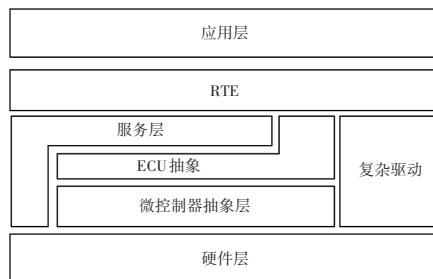


图1 AUTOSAR分层架构

应用层软件由1个或多个软件组件(Software Component, SWC)实现,SWC中封装了部分或者全部ECU功能,包括对其具体功能的实现以及接口的描述,但与ECU硬件系统没有连接。按照整车软件架构分层设计理论将软件模块划分为车队管理层、整车系统整体控制管理层、系统内部功能控制管理层、特定功能控制管理层、传感器执行器控制管理以及平台通用服务管理层^[9]。按照功能域可以划分为动力域、底盘域、车身舒适域、自动驾驶域以及信息娱乐域。不同层级、不同功能域的模块称为Module,可以根据模块的实际情况划分不同功能的软件组件,即SWC。本文中软件模块分层划分示意图如图2所示。

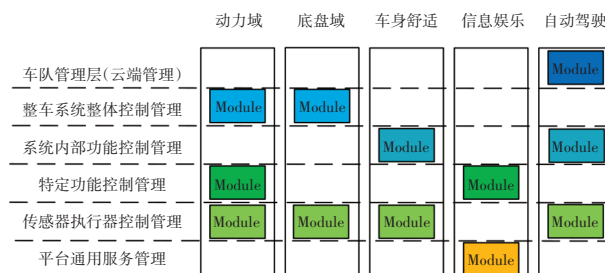


图2 软件模块分层划分示意图

RTE主要是为应用层提供通信手段,此处的通信是一种广义的通信,即为接口。应用层与其他软件的信息交互方式包含应用层中的不同SWC之间的信息交互和应用层SWC与基础软件之间的信息交互2种形式。RTE为上述交互使用接口的集散地,其汇总了所有交互接口。不同SWC间的通信通过调用RTE提供的接口函数(Application Programming Interface, API)实现,均受RTE管理和控制。每个API遵循统一的命名规则,且仅与软件组件自身描述有关。RTE层的软件均由工具供应商提供的配置工具自动生成,而SWC软件既可以通过工具生成也可以由用户自行编写,但要符合RTE层的API接口。

基础软件通常可以实现ECU的基础功能,如输入输出处理、数模转换、诊断、控制器局域网通讯以及操作系统。根据不同功能,可将基础软件继续划分为4个部分,分别为服务层(Service Layer),ECU抽象层(ECU Abstract Layer),复杂驱动(Complex Driver)和微控制器抽象层(Micro-Controller Abstraction Layer, MCAL)。上述4个部分间的互相依赖程度存在差异,为了满足不同ECU间的互操作性,基础软件通常借助工具自动生成。

1.3 开发工具

德国Vector公司开发的PREEvision是使用最多的电子电气架构设计工具,可以进行不同层级的电子电气架构设计,以ARXML通用格式的文件向下传递设计意图^[10]。PREEvision向开发人员提供完整的协同开发平台,可用于基于AUTOSAR的软件架构设计,支持SWC的创建划分以及接口的定义设计,为软件架构的定义和端口连接提供图形支持,同时PREEvision还支持不同版本AUTOSAR软件技术规范的导入导出^[11-12]。使用PREEvision进行软件建模具有以下优势:PREEvision支持AUTOSAR标准,满足集中SOA架构项目的建设需求。PREEvision能够有效进行全链条追溯,若其中某一环节发生变动,PREEvision能够展示出所有关联变动,供开发人员参考^[7]。

嵌入式软件开发常采用MATLAB/Simulink作为CP侧开发软件,其具备应用广泛、易读性强等特点。通过整合PREEvision和Simulink进行联合开发,实现正向的、自上而下的软件组件设计,达到快速迭代的有益效果。

在过程辅助以集成配置工具,实现软件的底层配置与编译。常用的工具主要包括达芬奇^[13]、华为工具VOS Architect/VOS Configurator、HIGHTEC以及Tasking。

2 AUTOSAR软件组件开发流程

2.1 开发整体流程

AUTOSAR方法论描述了从系统需求到ECU可执行代码产生过程的设计步骤^[14]。可以分为建立抽象系统需求描述、建立虚拟功能总线(Virtual Function Bus, VFB)系统描述、开发软件组件、开发系统和子系统、开发底层软件(Basic SoftWare, BSW)以及软件集成6个步骤。AUTOSAR开发方法如图3所示。

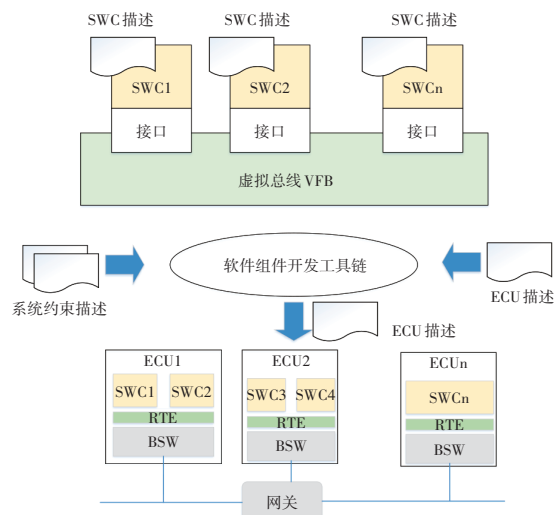


图3 AUTOSAR开发方法

通过建立抽象系统描述,可描述为1个或多个SWC组件。通过VFB系统,将1个或多个SWC组件组合形成完整系统架构。使用支持SWC软件开发组件的工具,根据ECU描述信息以及系统约束描述,实现软件需求到ECU的精准映射。具体操作流程为:首先借助配置工具生成系统配置描述文件,提取各ECU相关描述,并将SWC映射至相应ECU,之后将子系统进行独立划分。完成上述步骤后便可针对单个ECU开展SWC与BSW的开发工作,最终将生成的代码进行集成,生成可执行文件并下载至ECU中运行。

2.2 详细流程

本文基于项目开发实践,对软件组件的详细开发流程进行细化。开发角色主要包括设计功能设计师、物理架构设计师、软件架构设计师、应用软件设计师、网络通讯设计师和平台软件设计师。相关角色的协作流程及交互文件如图4所示。

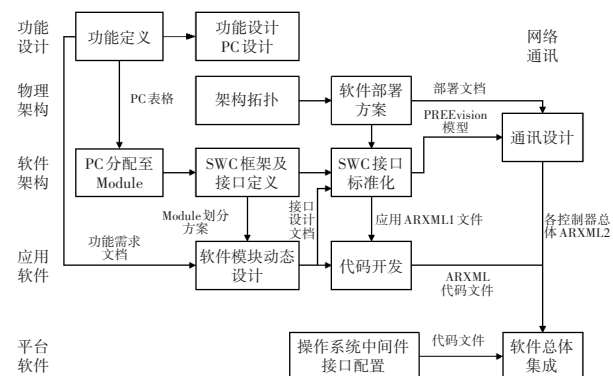


图4 软件组件开发流程

3 AUTOSAR软件组件开发实践

基于提出的开发方法,进行了基于PREEvision和MATLAB开发AUTOSAR软件组件的实践,完成了CP

侧SOA软件组件开发的全过程,对整车电子电气架构正向开发具有一定借鉴作用。本文将以车辆模式模块(Vehicle Mode Management, VMM)为例进行具体软件组件开发实践。

3.1 功能需求设计

车辆模式可分为基本车辆模式和特殊车辆模式2种类型。其中,基本车辆模式具有互斥性,即在特定时刻,车辆仅能处于一种基本车辆模式。基本车辆模式包括工厂模式、存储模式、运输模式、展厅模式以及正常模式。特殊车辆模式可以在特定基本车辆模式下进入,且不改变基本车辆模式。特殊车辆模式当前包括远程模式、空中下载(Over The Air, OTA)模式以及被盗追踪模式。特殊车辆模式可以随整车功能的丰富而拓展。车辆模式功能卡片如图5所示。

| | | | |
|--------------|--|---------|------------------|
| 功能ID | F803XX | 功能名称 | 车辆模式 |
| 二级系统 | 区域控制系统 | 一级系统 | 智控系统 |
| 功能描述 使用场景 | 支持车辆在制造完成后及运输过程中部分功能需求,也用于在交付前节省电量,保证安全等 | 用户视角 | 无感 |
| | | 软件模块 | Veh_VM_VehModMan |
| | | 硬件模块 | VDC_821MCU |
| 相关系统 | 电源系统、网联系统、云服务系统 | 相关部件 | VDC、CSC、ICC |
| 体验指标 验收项 | 1、车辆模式切换 2、不同模式下的整车表现 | 技术指标验收项 | 本地操作切换响应时间小于2s |
| 释放日期 | xxxx.xx.xx | 首发车型 | E00X |
| 核心交付 | PRD文档、软件详细设计文档、验收测试报告 | 质量要求 | 可靠性 |

图5 车辆模式功能卡片

3.2 以太网服务接口设计

依据功能规范文档的描述,按照服务列表设计模板进行以太网服务设计,标准服务列表中定义了符合要求的软件组件SWC及端口和各软件组件端口间通信所使用的接口数据类型,服务接口元素类型包括Event、Field和Method,一般Event类型用来传输数据,Notify和Method类型用来执行操作。

服务列表包含服务接口定义、数据类型定义、服务部署定义等信息。其中服务接口定义包含服务名称、服务ID、版本信息、服务接口元素名称、服务接口元素类型、服务接口元素标识符、事件组、传输层协议、方法参数信息、发送策略以及参考数据类型名称^[15]。数据类型定义包含数据类型名称、数据类型类别、基础数据类型、精度、偏移量、最小值、最大值、初始值、无效值以及枚举值等信息。服务部署包含参与者、虚拟局域网标识符、服务名称、服务ID、提供者或者消费方服务实例标识符、端口号以及组播IP地址等信息。基于车辆模式VMM功能规范,设计车辆模式VMM管理模块服务列表,其中关键信息定义如表1所示。

以太网服务列表设计完成后,每项参数均需经过严格的语法、语义及格式检查。对于服务接口数量庞大、数据结构复杂、数据类型复用的接口,人工核对将耗费大量的时间和精力,且不易发觉隐藏问题。为了应对大规模的整车工程开发,提高开发的整体效率和质量,需借助数字化的自动化校验工具。

基于Python编程语言开发的自动化工具用户交互界面如图6所示。基于单点校验规则和关联校验规则,可以迅速定位服务列表中设计不规范的位置,并提供修改建议,可以有效提高设计的效率与质量。条目化的校验规则,可以根据实际的校验需求,进行校验规则的灵活配置,具备较高扩展性。

3.3 软件组件建模

在PREEvision环境中开展软件组件SWC建模工作。在测试产品线内,基于PREEvision二次开发的插件将设计好的服务列表导入。服务列表自动化导入插件如图7所示。生成建模所需的基本元素,包括服务、服务接口、服务部署、通信接口、应用数据类型、实现数据类型以及Mapping文件。数据类型还包括计算方法(Computation Method, CM)、初始值等信息。其中Interface仅定义了通信数据的接口,对于Sender/Receiver Interface,其传输的数据实际在Data Element中定义。而对于Client/Server Interface,其传输的数据

表1 车辆模式VMM模块服务列表设计

| | | | | |
|------|-----------------------------|------------------------|-------|------------------------------------|
| 服务名称 | Veh_VM_VehicleModeManagment | | | |
| 服务ID | 0x900B | | | |
| 数据名称 | VMM_NotifyONChgVehicleMode | VMM_GetVehicleMode | | VMM_NotifyONChgVehicleModeReqVaild |
| 数据类别 | Field/Notify | Field/Getter | | Event |
| 数据ID | 0x8001 | 0x1001 | | 0x9005 |
| 数据类型 | Veh_VehicleMode_struct | Veh_VehicleMode_struct | | VMM_VehicleModeReqVaild |

实际在方法参数中定义。数据类型由应用数据类型、实现数据类型和基础数据类型组成。其中应用数据类型需与实现数据类型之间需建立映射关系以及转换方法,实现数据类型需与基础数据类型建立关联,以确保整个建模过程中数据的准确传递与有效整合。建模元素的生成如图8所示。

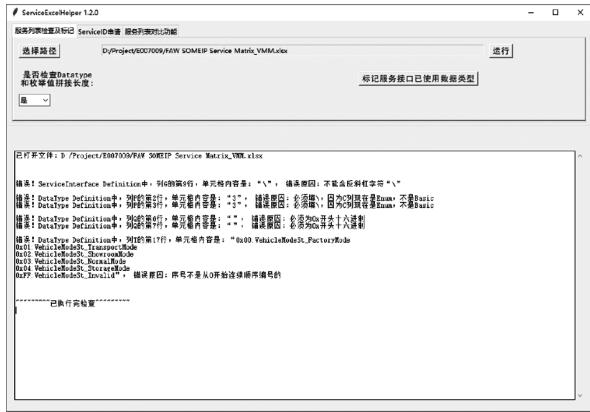


图6 VMM模块服务列表检查

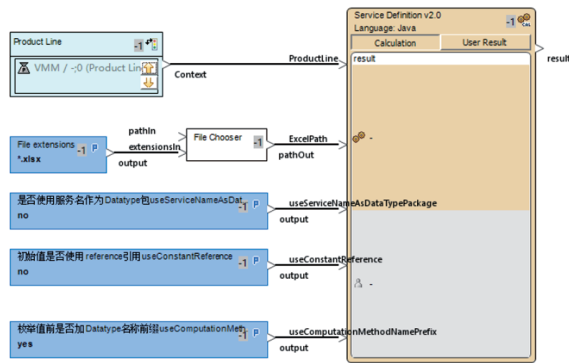


图7 VMM服务列表自动化导入

基于插件生成的基本建模元素,在正式产品线进行软件组件的创建、收发接口的配置、Internal Behavior内部行为定义,关联数据类型 Mapping包等配置。最后将所创建好的软件组件进行实例化,并完成发送与接收接口基于Interface的连线。

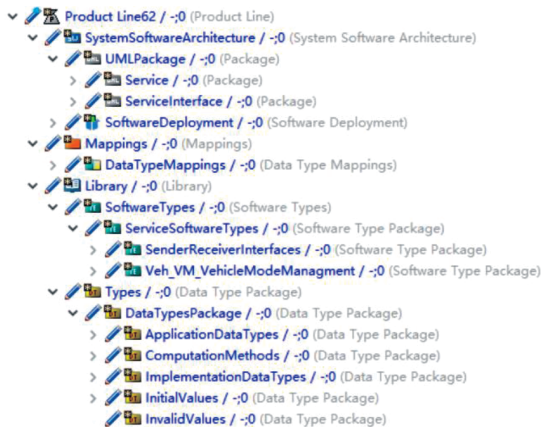


图8 VMM基础建模元素生成

基于车辆硬件拓扑,建立ECU芯片,配置MAC地址、IP地址、VLAN信息以及端口等信息,完成硬件层的配置。将设计完成的软件组件部署至硬件芯片,建立SWC至硬件芯片的映射Mapping。根据以太网通信配置需求进行路由和通信配置,生成通信层System Signal、Signal Transmission、PDU Transmission等信息。完成软件组件的软件层、硬件层及通信层的建模配置工作。

在SWC建模完成后,导出ARXML前可以基于PREEvision二次开发的插件,对相关参数信息进行一致性检查。将检查出的问题进行修正后,基于PREEvision自带的导出功能,将SWC按照标准AUTOSAR规范ARXML标准格式导出。分别导出应用软件层SWC的ARXML1及以硬件芯片为单位的带软件层及通讯层信息的ARXML2供下游开发专业配置。

3.4 软件组件ARXML多工具复合校验

将导出的ARXML文件基于多工具进行多维度的校验,如图9所示,可以通过脚本导入MATLAB生成Simulink模型,基于软件插件AUTOSAR Component Designer工具进行检查,检测ARXML文件与MATLAB规则的一致性,用于适配后续的应用层软件开发。

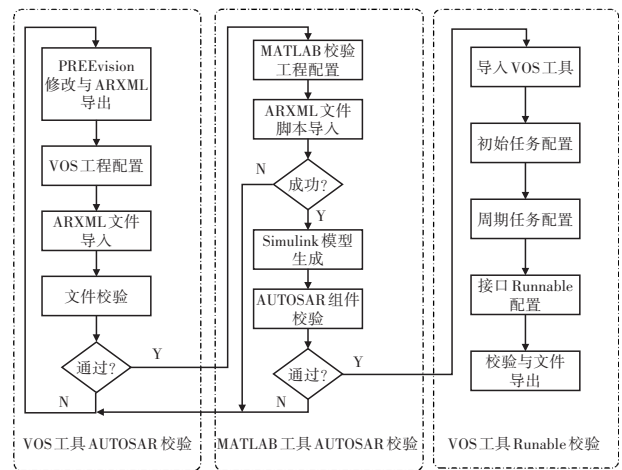


图9 多工具复合校验及Runnable配置流程

VOS Architect是一款软件组件校验和配置工具。通过图形视图或文本网格视图,设计师可以检查软件组件,包括端口(Port)、数据类型(Data Type)、连接器(Connector)和内部行为(Internal behavior)。通过一致性检查,可以验证SWC设计是否符合AUTOSAR标准。导入VOS Architect工具,基于工具自带的AUTOSAR规范检查插件,检查软件集成与代码生成的一致性。同时将以硬件为单位的ARXML文件导入华为VOS Configurator工具,检查通讯层配置的报

错情况。检查无误后可以正式释放,按照开发流程向下传递 ARXML 文件,用于软件底座开发。校验及配置流程如图 10 所示。

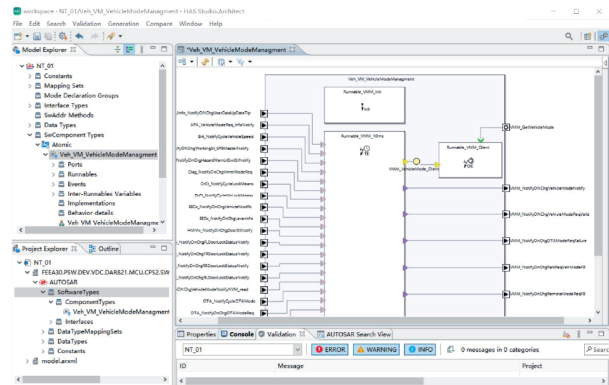


图 10 ARXML VOS 校验与运行实体配置

完成软件组件的框架部分后,还需完善软件组件的内部行为。一般包括 RTE 事件(Event)、运行实体(Runnable Entity)以及函数间变量(Inter Runnable Variable)。运行实体是软件组件内最小的逻辑功能单元,可以被 RTE 事件触发(如被定时器触发、被接收数据触发、被操作调用触发)。图 10 为基于 VOS Architect 工具进行一致性检查与运行实体配置的过程。

3.5 应用软件开发

应用软件获取到软件架构设计释放的标准 ARXMLI 文件后,将配置运行实体的文件导入 MATLAB,生成 Simulink 模型。模型的收发信息已配置完毕,应用层需进行接口的匹配与应用层逻辑的搭建。基于软件开发需求文档,搭建子系统内部软件架构,划分单功能控制模块,进行详细的控制逻辑开发。

以车辆模式中的 OTA 模式及远程模式为例分析模式的跳转逻辑。OTA 模式进入需同时满足以下 4 项条件:(1)接收到 OTA 业务进入 OTA 模式请求;(2)电源模式非 Driving 状态,车辆处于静止状态,挡位处于 P 挡且以及电子驻车制动已施加;(3)当前未处于远程模式;(4)当前不存在绝缘监测故障。而 OTA 模式在满足以下任一条件时将退出:(1)接收到 OTA 业务退出 OTA 模式请求;(2)超时;(3)车辆下电且电源模式进入 Pre-off 状态。

远程模式的进入需同时满足以下条件:(1)接收到远程主控业务进入远程模式请求;(2)电源模式非 Driving 状态,车辆处于静止状态,挡位处于 P 挡且以及电子驻车制动系统(Electronic Parking Brake, EPB)已施加;(3)当前未处于 OTA 模式;(4)当前未处于工厂模式;(5)当前不存在绝缘监测故障。远程模式满

足以下任一条件时将退出:(1)接收到远程主控业务退出远程模式请求;(2)遥控或数字钥匙进行本地操作上电;(3)电源模式请求退出远程模式。

基于设定功能逻辑进行应用层的软件的详细设计,图 11 为最终完成的应用层集成控制模块模型。

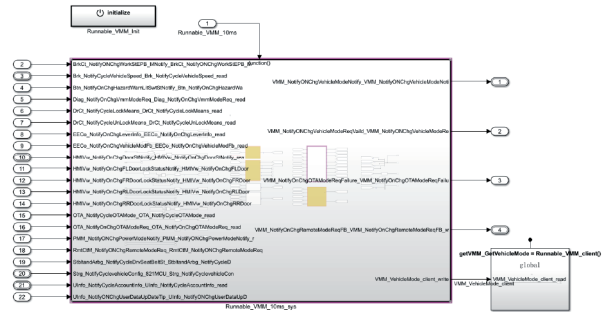


图 11 应用层 VMM 集成控制模块

模块开发完成后要进行系统内部测试,系统内部测试可以基于子系统模型的模型在环(Model In Loop, MIL)测试、基于代码的代码在环(Software In Loop, SIL)测试及基于子系统硬件的硬件在环(Hardware In Loop, HIL)测试。MIL 测试过程中可以基于 MATLAB 自带 Simulink Test 工具,创建 Test Harness 测试框架,基于表格设计测试用例,进行覆盖度的测试,并生成测试报告。在测试通过后进行嵌入式代码生成工作。将生成的代码文件及 ARXML 文件释放进行系统集成和后续测试工作。

3.6 系统集成与测试

配置完成运行实体的 ARXML 文件和带通讯层的以硬件为单位的 ARXML 文件可以配置硬件整体的 ARXML 文件,导出的 ARXML 文件可以用于域控底座的开发工作。

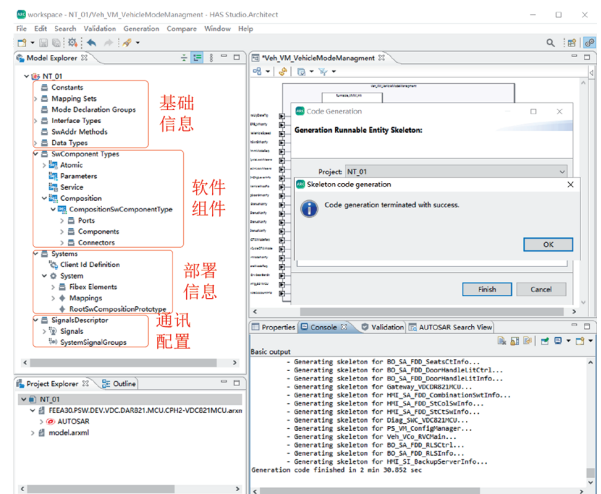


图 12 应用需求框架代码生成

域控底座基础软件开发包括标准模块配置开发、通讯服务中间件开发、复杂驱动开发、系统服务中间

件开发以及平台软件集成。其中通信服务中间件开发包括数据分发服务(Data Distribution Service, DDS)、面向服务的中间件(Scalable service-Oriented MiddlewarE over IP, SOME/IP)、基于以太网的诊断服(Diagnostic communication Over Internet Protocol, DOIP)、CAN等。系统服务中间件包括标定、诊断、OTA、时间同步、日志管理、上网服务、系统状态管理、内存管理、电源管理以及下线检测等。涉及安全需求的还需进行功能安全、预期功能安全、信息安全以及操作系统安全内核的开发工作。

底座开发的每项内容均需匹配相应的测试任务,一般底座的测试项主要关注通信测试、诊断测试、刷写测试、资源测试、平台功能测试;其中通信测试包括CAN通信、以太网通信、标定测试、信号路由、端到端校验;资源测试包括微控制器负载及其资源、微处理器资源测试;平台功能测试包括Log功能测试、定时管理、时间同步、网络管理等。经过详细测试的底座方可正式发布。

域控底座软件释放后集成经过单元测试的应用层代码,可以进行整车级的电子电气测试。整车电子电气测试基于建立的统一测试场景库,进行仿真测试、实车测试全流程测试。仿真测试包括整车代码在环、硬件在环、整车在环测试,实车测试包括场地测试和开放路测试,支撑整车功能测试验证。场景库的搭建主要基于标准法规、事故场景、自然驾驶以及危险极限等,进行模型、算法、评价3个维度的测评。

代码在环测试用于模型生成代码和手写算法分析,进行产品级代码验证。硬件在环测试是真实控制硬件在环,用于控制器算法、硬件、网络等验证。整车在环测试为实车在环仿真平台,实现车辆在环测试、极限场景评价等。场地测试是实车在封闭场地、示范区及公开道路测试验证及评价。开放路段测试是真实道路环境下的应对能力。测试项主要对功能逻辑、功能降级、报警提示、交互测试、安全测试、法规项、复杂场景测试等进行测试和评价。

4 结束语

本文提出一种面向服务的架构框架下软件组件的开发方法,设计了符合AUTOSAR规范的软件组件开发流程。基于提出的方法及流程给出了从需求分析、服务设计、架构设计、应用开发、软件集成与测试的开发实践。研究表明,在集中式SOA开发框架

下,主机厂架构师可以在PREEvision工具中,从整车系统层面定义符合AUTOSAR规范的软件架构,以ARXML文件的形式与下游设计软件MATALAB、华为VOS、DaVinci等无缝对接,从而保证设计信息的准确传递。为了应对汽车电子功能的快速迭代,汽车制造企业可以通过这种设计方式,基于内部合理的开发流程,主导汽车从需求、设计、开发到测试的全过程,实现功能的快速实现与迭代升级,提升车型开发效率及质量。

参 考 文 献

- [1] 李丹, 吕颖, 李骏, 等. 面向服务的体系架构[J]. 汽车文摘, 2021(10): 52-57.
- [2] 李海明, 程福明, 吴文杰, 等. 一种面向服务的智能驾驶系统设计方法: CN202111456035.2[P]. 2022-3-1.
- [3] 王东, 李井波, 张晨晖, 等. 一种基于SOA架构的总线系统. CN202011633659.2.[P]. 2021-5-18.
- [4] 周耀, 杨柳春, 张凡武. 基于模型的AUTOSAR应用层软件开发环境[J]. 2015中国汽车工程学会年会论文集, 2015(62): 752.
- [5] 宋杰静, 马云杰, 田鹏飞. 基于PREEvision的以太网SOA设计方法[J]. 2019中国汽车工程学会年会论文集, 2019(13): 1275.
- [6] 詹德凯, 高越. 基于PREEvision的SOA设计[J]. 汽车实用技术, 2022, 47(23): 62-70.
- [7] 梁瑜, 杨雪珠, 李海霞, 等. 电动助力转向系统, 车辆及电动助力转向系统的控制方法: CN202111519164.1[P]. 2022-3-25.
- [8] 孙怡琳. 基于AUTOSAR标准架构的智能远程防盗系统设计与实现[D]. 杭州: 浙江大学, 2021.
- [9] 金劭男, 高仕宁, 李超, 等. 面向SOA的E/E架构设计数智化转型实践[J]. 汽车文摘, 2024(3): 31-38.
- [10] 刘敏, 李吉宗, 张文学. 基于电控框架模型的硬件资源管理研究[J]. 汽车实用技术, 2021, 46(9): 68-71.
- [11] 王永辉. 基于PREEvision的汽车电子电气架构设计介绍[J]. 汽车实用技术, 2019(15): 111-112.
- [12] 匡小军, 唐香蕉, 周涛, 等. 基于PREEvision的汽车电子电气架构工具链研究[J]. 汽车电器, 2019(8): 62-64.
- [13] 宋杰静, 马云杰, 田鹏飞. 基于PREEvision的以太网SOA设计方法[C]// 中国汽车工程学会年, 2019.
- [14] 蔡伟杰. 基于模型符合AUTOSAR的电动汽车控制系统开发[D]. 西安: 长安大学, 2019.
- [15] 杨伯欣. 基于SOA的企业公文管理系统设计与实现[D]. 天津: 南开大学, 2015.

(责任编辑 梵玲)