文章编号：2095-4980(2020)06-1117-05

基于 AADL 和 SCADE 的模型驱动软件设计

刘芮滦，邓 杨，龚 彬

(中国工程物理研究院 电子工程研究所，四川 绵阳 621999)

摘 要：模型驱动开发逐渐应用于嵌入式系统的软件设计，在软件设计阶段重点关注的是软件的架构模型和详细功能模型。用于嵌入式系统软件建模的语言和工具很多，其中结构分析与设计语言(AADL)模型可以构建嵌入式软件的架构，高安全性应用开发环境(SCADE)模型可以描述嵌入式软件的逻辑功能，将两者统一使用可以满足嵌入式软件概要设计和详细设计的建模需求。针对某飞行器控制系统，本文分别使用 AADL 和 SCADE 对飞行器控制系统软件架构和功能进行建模，利用 KCG 工具从 SCADE 模型自动生成 C 代码，通过手工代码和自动生成代码的集成完成控制系统部分软件设计。实际应用表明，采用 AADL 和 SCADE 相结合的建模方法适用于模型驱动开发在嵌入式软件设计中应用。

关键词：AADL 模型；SCADE 模型；模型驱动开发；嵌入式系统；自动代码生成

中图分类号：TN967.6；TP391.9 **文献标志码：**A **doi：**10.11805/TKYDA2019233

Model driven software development based on AADL and SCADE

LIU Ruiluan，DENG Yang，GONG Bin

(Institute of Electronic Engineering，China Academy of Engineering Physics，Mianyang Sichuan 621999，China)

Abstract：Model-driven design has been used in embedded system software design. The software architecture model and detailed functional model are the focuses in the software design stage. There are many languages and tools for the embedded system software modeling. The architecture of embedded software can be built by the Architectural Analysis and Design Language(AADL) model, and the logic function of embedded software can be described by the Safety Critical Application Development Environment(SCADE) model. The integration of the two models can meet the modeling requirements of outline design and detailed design of embedded software. AADL and SCADE are adopted to model the architecture and function of the software of an aircraft control system, KCG tool is utilized to auto generate C code from SCADE model, and the aircraft control system software is partly designed through the integration of handmade and auto-generated codes. Actual application indicates that AADL associated with SCADE is suitable for model-driven design applied in the embedded software design.

Keywords：AADL；SCADE；model-driven design；embedded system；auto code generation

随着科技的进步，嵌入式系统的应用已无处不在，嵌入式软件开发方式也得到长足发展。目前，模型驱动开发[1-4]在嵌入式软件设计中得到应用，逐渐代替传统的基于文档的开发方式。通过建模和仿真可以提前进行软件的验证和确认，通过自动代码生成可以减少手工编码的工作量并有效提升代码质量。各种各样的建模语言和建模工具应运而生并日渐成熟[5-6]。结构分析与设计语言(AADL)[7-8]是由美国汽车工程师协会(Society of Automotive Engineers，SAE)在 2004 年以 SAE AS5506 标准发布的建模语言，用于设计和分析嵌入式系统的软、硬件体系结构及功能与非功能性质。AADL 采用系统组件层次化方法建立嵌入式系统的体系结构模型，该模型提供了运行时的、定义明确的、可分析的语义，并以接口规范及其实现详细描述了各软硬件组件，以及各个组件之间通信的交互规范。AADL 支持高安全嵌入式系统的结构建模与分析验证，适用于构建嵌入式系统的体系结构，但不太适合描述系统具体的逻辑功能[9-11]。高安全性应用开发环境[12](SCADE)面向高安全性系统和

收稿日期：2019-06-29；**修回日期：**2019-11-05

作者简介：刘芮滦(1987–)，男，硕士，助理研究员，主要研究方向为嵌入式软件建模与仿真。email:allencasai@126.com

嵌入式软件，适用于对嵌入式系统的功能和控制逻辑建模和仿真，其自动代码生成工具 KCG 代码生成器(KCG Code Generator，KCG)通过 DO-178C 最高标准认证，能够充分保证模型和代码的一致性。SCADE[13]支持文本和图形两种建模方式，文本方式使用 Lustre 语言，图形方式包括安全状态机和数据流图，这两种建模方式都具有严格的数学语义。SCADE 模型验证通过后，使用 KCG 以 SCADE 图形模型作为输入，首先将方程式、参数块等图形转变成 Lustre 语言描述，然后再将 Lustre 描述转换成面向工程的 C 语言。

1 飞行器控制系统 AADL 架构建模

本文以某飞行器控制系统为例建立 AADL 模型。该控制系统通过惯性测量单元(Inertial Measurement Unit，IMU)传感器接收 IMU 数据，GPS 设备接收 GPS 数据，依据控制策略计算控制命令，将控制命令传输给 Motors 设备控制飞行器飞行，同时通过 WiFi 无线传输装置将接收的 GPS 位置信息处理后发射给地面测控设备，或接收地面测控设备发送的遥控指令。飞行器控制系统的处理器为 ARM Cortex-M4(主频 100 MHz)，采用标准 CAN 总线用于内部设备通信。

AADL 模型实例使用进程、线程、数据描述控制系统软件体系架构，使用处理器、存储器、总线、设备组件描述控制系统的硬件体系架构。在模型中，IMU 传感器、GPS、Motors、WiFi 无线传输装置建模为设备组件，模型只关注设备的通信接口数据而忽略设备内部结构。飞行器控制功能表示为进程组件，软件运行时组件包括在该进程类型的实现中。控制系统中组件的接口是在组件类型中说明的端口特征，例如 IMU 组件经由数据端口 IMU_Data 输出加速度数据。通过端口可以实现组件之间的连接，例如在 Pilot_Control 进程的数据端口 Act_Cmd 与 Motors 设备的数据端口 Act_Cmd 之间建立连接，并标记为<<Data>>。通过定义处理器、存储器和总线，实现控制系统硬件建模。处理器 ARM_100 MHz 执行控制系统的代码(线程)，存储器 Stand_Memory 存储系统的可执行代码(进程)，总线 Stand_CAN_Bus 为系统提供物理路径。在设备组件、处理器组件、存储器组件中添加总线访问(requires bus access)说明，通过说明可以将它们与总线 Stand_CAN_Bus 连接，并标记为<<BusAccess>>。飞行器控制系统的 AADL 模型图形表示如图 1 所示。

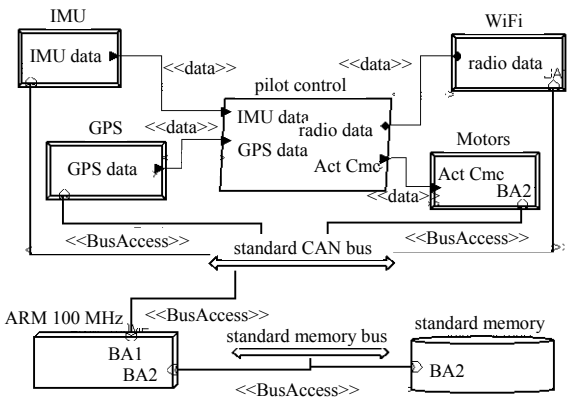

Fig.1 AADL graphical representation of aircraft control system
图 1 飞行器控制系统的 AADL 模型图形表示

通过 OSATE 工具建立飞行器控制系统 AADL 模型后自动生成相应的 AADL 文本表示。飞行器控制系统包括各个设备、数据连接和总线连接的各组件文本部分说明如下：

```
systemimplementation Complete.Aerocraft
subcomponents
    IMU: device IMU.Sensor;
    GPS: device GPS.Dev;
    Pilot_Control: process Aerocraft_Control.Pilot;
    Motors: device Actuator.Motors;
    WiFi: device WiFi.Transmit;
    ARM_100MHz: processor MCU.MHz;
    Stand_Memory: memory RAM.Standard;
    Standard_CAN_Bus: bus CAN.Standard;
connections
    DC1: dataport IMU.IMU_Data -> Pilot_Control.IMU_Data;
    DC2: dataport GPS.GPS_Data -> Pilot_Control.GPS_Data;
    DC3: dataport Pilot_Control.Radio_Data -> WiFi.Radio_Data;
    DC4: dataport Pilot_Control.Act_Cmd -> Motors.Act_Cmd;
    BC1: busaccess Standard_CAN_Bus<-> GPS.BA1;
    BC2: busaccess Standard_CAN_Bus <-> Motors.BA1;
    BC3: busaccess Standard_CAN_Bus <-> ARM_100MHz.BA1;
    BC4: busaccess Standard_CAN_Bus <-> Stand_Memory.BA1;
    BC5: busaccess Standard_CAN_Bus <-> WiFi.BA1;
    BC6: busaccess Standard_CAN_Bus <-> IMU.BA1;
end Complete.Aerocraft;
```

对进程 Pilot_Control 进行简化处理，建立 3 个线程组件，如图 2 所示。第一个线程组件 Sacle_Acc_Data 接收 IMU 传感器发送的加速度数据和对数据进行预处理，并向第二个线程组件发送处理后的数据；第二个线程组件 Acc_Control_Laws 是执行加速度控制规则计算，向 Motors 组件发送控制命令，并向第三个线程组件发送启动无线传输事件；第三个线程组件 WiFi_Transmit 接收到 Acc_Control_Laws 线程组件发送的事件 Start_Tran 后将接收的 GPS 数据发送给 WiFi 组件。

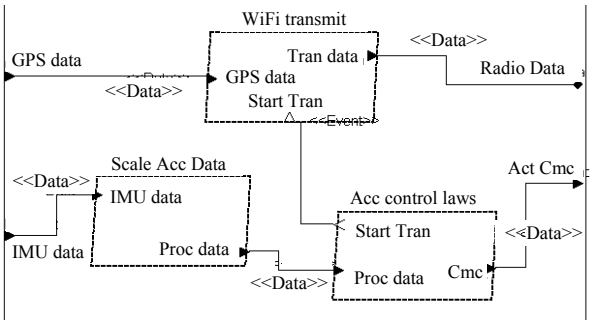

Fig.2 AADL graphical representation of process implementation
图 2 进程实现的 AADL 图形表示

通过文本表示可以定义线程组件的执行特性，如周期、执行时间等。例如线程组件实例 Acc_Control_Laws 的周期为 20 ms，执行时间为 3~5 ms。利用 OSATE 调度程序和调度分析插件可以分析系统的调度性[12]。

进程组件 Pilot_Control 的部分 AADL 文本表示如下：

```
threadAcc_Control_Laws:
features
Cmd: Out data port;
Pro_Data: in data port;
Start_Tran: out event port;
properties
Dispatch_Protocol => Periodic;
    Compute_Execution_Time => 3ms .. 5ms;
    Period => 20ms;
endAcc_Control_Laws:;

processimplementation Aerocraft_Control.Pilot
subcomponents
    Scale_Acc_Data: thread Read_Data.Acceleration;
    Acc_Control_Laws: thread Control_Laws.Pilot;
    WiFi_Transmit: thread WiFi_Transmit.DataTran;
connections
    DC1: dataport IMU_Data -> Scale_Acc_Data.IMU_Data;
    DC2: dataport Scale_Acc_Data.Proc_Data -> Acc_Control_Laws.Pro_Data;
    DC3: dataport Acc_Control_Laws.Cmd -> Act_Cmd;
    DC4: dataport GPS_Data -> WiFi_Transmit.GPS_Data;
    DC5: dataport WiFi_Transmit.Tran_Data -> Radio_Data;
    EC1: eventport Acc_Control_Laws.Start_Tran -> WiFi_Transmit.Start_Tran;
end Aerocraft_Control.Pilot;
```

2　SCADE 详细功能建模

使用 SCADE 对飞行器控制系统的逻辑功能进行建模。在该实例中飞行器控制系统的具体功能可以分解到 3 个线程组件 Sacle_Acc_Data、Acc_Control_Laws 和 WiFi_Transmit，对线程组件 Acc_Control_Laws 的功能建模进行具体说明。线程组件 Acc_Control_Laws 的主要功能是对加速度数据进行计算和判断，其 SCADE 模型如图 3 所示。该 SCADE 模型采用并行状态机设计，包含计算 IntegraCal 和有效数据计数 Count 两个并行状态，状态 IntegralCal 对加速度值进行计算获取当前速度，状态 Count 判断当前加速度值大小。当加速度和速度满足判定条件时，线程组件 Acc_Control_Laws 向线程组件 WiFi_Transmit 发送启动无线传输事件 Start_Tran。

状态 Count 有 TrajChkInit 和 TrajChkCount 等子状态，子状态 TrajChkInit 对变量 DataCnt 和 AvailCnt 清零，当变量 Data(加速度值)大于等于 2 时，从子状态 TrajChkInit 迁移到子状态 TrajChkCount，并执行计数动作，当变量 DataCnt 等于 100 而变量 AvailCnt 小于 80 则又从子状态 TrajChkCount 迁移回子状态 TrajChkInt，清零变量 DataCnt 和 AvailCnt。

SCADE 模型建好后首先利用建模软件进行模型检查。模型检查正确后对模型进行仿真，SCADE 仿真器基于生成的 C 代码进行，可以对模型进行连续、批处理的仿真。通过加载 LoadAccData.in 场景文件，加载模拟的加表数据，仿真周期设置为 10 ms。对重要输入和输出进行标记，在观察窗口中观察相关变量的值，运行模型

后可得到仿真结果。通过仿真可知当满足给定要求后变量 TrajChkFlag 置 1，通过判断该变量标志，线程组件 Acc_Control_Laws 可以向线程组件 WiFi_Transmit 给出事件 Start_Tran。

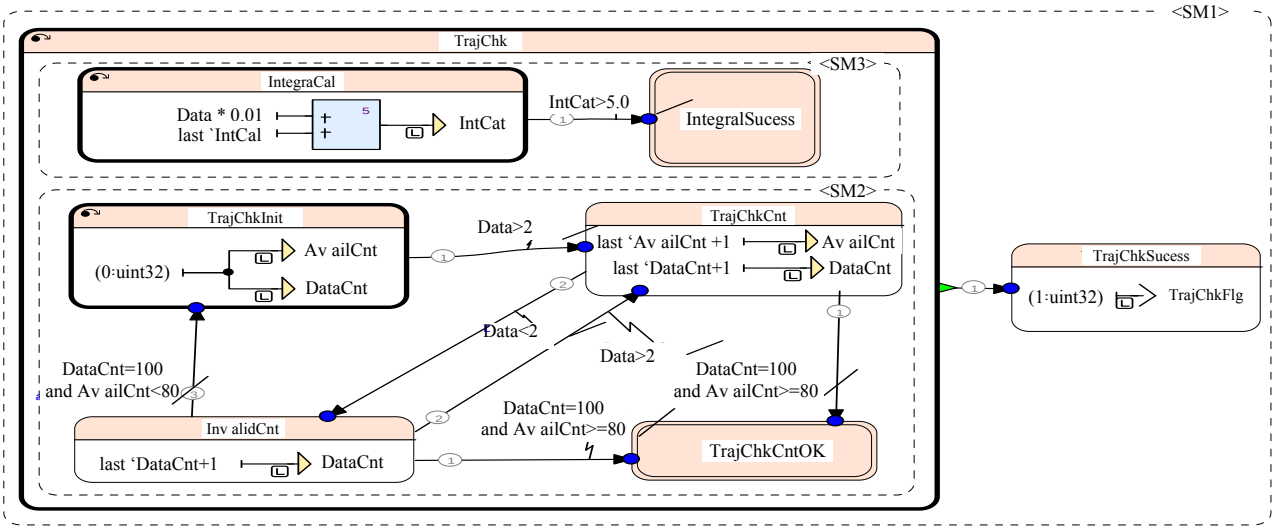

Fig.3 SCADE model of acceleration data calculation function

图 3 加速度数据计算功能的 SCADE 模型

3 自动代码生成与代码集成

除了模型检查、模型仿真，SCADE 模型还可以进行覆盖率分析和形式化验证。经过验证后的模型保证了其正确性和安全性，可以自动生成标准的 C 代码。SCADE 模型中的每个变量在其作用域内只赋值一次，没有递归、死循环、动态指针、动态内存分配，因此满足软件的高安全性要求[13−15]。

线程组件 Sacle_Acc_Data、Acc_Control_Laws 和 WiFi_Transmit 对应 C 代码中与平台相关任务入口函数，例如 Acc_Control_Laws 对应函数 void Acc_Control_Laws_Task(void*arg)。使用 KCG 工具可以从线程组件 Acc_Control_Laws 的 SCADE 模型自动生成 C 代码，代码集成时需要使用其中 7 个头文件和源文件，其中文件 Acc_Control_Laws.c 包含的模型函数 TrajChk()是 SCADE 模型的具体实现。

函数的参数包含两个结构体指针，结构体中的元素分别包含输入和输出变量。其中，kcg_float32 Data 对应 AADL 模型中线程组件 Acc_Control_Laws 的输入数据端口 Pro_Data；kcg_uint32 TrajChkFlg 对应输出事件端口 Start_Tran。在设计飞行器控制系统软件时，函数 Acc_Control_Laws_Task()接收函数 Scale_Acc_Data_Task()发送的处理后的加速度值进行计算判断，计算判断功能可以直接调用 KCG 自动生成 C 代码 Acc_Control_Laws.c 文件中的函数 TrajChk()，实现自动生成代码和手工代码的集成。当输出变量 TrajChkFlg 为 1 表示当前加速度满足给定要求，可以向函数 WiFi_Transmit_Task()发送启动无线传输事件 Start_On。函数 Acc_Control_Laws_Task()的伪代码如下：

```
void Acc_Control_Laws_Task(void* arg)
{
    while(1)
        {
            /*阻塞等待线程 Scale_Acc_Data_Task 发送加速度数据 fAccData */
            ......
            inC->Data = fAccData;
            /*调用自动生成代码对加速度数据进行计算处理*/
            TrajChk(inC_TrajChk_CK *inC, outC_TrajChk_CK *outC);
            if(outC-> TrajChkFlg == 1)
                {
                    /* 加速度满足条件，向线程 WiFi_Transmit_Thread 发送事件 Start_On*/
                    ......
                }
        }
}
```

4　结论

本文基于 AADL 和 SCADE 建模语言及工具完成某飞行器控制系统建模，通过代码自动生成技术完成嵌入式系统部分逻辑控制功能的软件实现，提高了软件开发效率，后期还需要通过 AADL 模型仿真分析实现软件架构模型的优化设计，以及根据 AADL 模型自动生成框架代码来进一步提高软件开发的自动化程度。实际应用表明，AADL 和 SCADE 相结合进行软件建模是模型驱动开发的一种具有可操作性的实践示范，具有在嵌入式系统软件设计中应用和推广的价值。

参考文献：

[1] 刘兴华,曹云峰. 一种模型驱动的嵌入式控制软件设计技术研究[J]. 系统仿真学报, 2013,25(7):1530-1567. (LIU Xinghua,CAO Yunfeng. Research on model driven development of embedded control software[J]. Journal of System Simulation, 2013,25(7):1530-1567.)

[2] 赵勇,陈香兰. 基于模型驱动的实时嵌入式系统[J]. 计算机系统应用, 2017,26(8):83-87. (ZHAO Yong,CHEN Xianglan. Real-time embedded system design method based on model-driven[J]. Computer Systems & Applications, 2017, 26(8):83-87.)

[3] ROUDIER Y,IDREES M S,APVRILLE L. Towards the model-driven engineering of safety requirements for embedded systems[C]// 2013 International Workshop on Model-Driven Requirements Engineering(MoDRE). Rio de Janeiro,Brazil: IEEE, 2013:55-64.

[4] HUTCHINSON J,ROUNCEFIELD M,WHITTLE J. Model-driven engineering practices in industry[C]// 33rd International Conference on Software Engineering(ICSE). Honolulu,HI,USA:IEEE, 2011:633-642.

[5] 王文全,宋科璞,王勇,等. 基于模型驱动的机载嵌入式软件应用[J]. 计算机技术与发展, 2013,23(8):145-148. (WANG Wenquan,SONG Kepu,WANG Yong,et al. Airborne embedded software application based on MDA[J]. Computer Technology and Development, 2013,23(8):145-148.)

[6] 张涛,秦凯,王楠,等. 面向航天领域的模型驱动软件设计开发方法[J]. 航天控制, 2017,35(5):74-79. (ZHANG Tao, QIN Kai,WANG Nan,et al. Research on model-driven aerospace software development method[J]. Aerospace Control, 2017,35(5):74-79.)

[7] Aerospace SAE. Architecture analysis & design language(standard SAE AS5506)[S]. 2004.

[8] Aerospace SAE. Architecture analysis & design language(standard SAE AS5506A)[S]. 2009.

[9] 杨志斌,皮磊,胡凯,等. 复杂嵌入式实时系统体系结构设计与分析语言:AADL[J]. 软件学报, 2010,21(5):899-915. (YANG Zhibin,PI Lei,HU Kai,et al. AADL: an architecture design and analysis language for complex embedded real-time systems[J]. Journal of Software, 2010,21(5):899-915.)

[10] 刘承威,杨志斌,周勇,等. 面向限定自然语言需求的 AADL 自动生成工具[J]. 小型微型计算机系统, 2019,40(5):984-995. (LIU Chengwei,YANG Zhibin,ZHOU Yong,et al. Automated tool to derive AADL models from requirements in restricted natural language[J]. Journal of Chinese Computer Systems, 2019,40(5):984-995.)

[11] 王飞,杨志斌,黄志球,等. 基于限定自然语言需求模板的 AADL 模型生成方法[J]. 软件学报, 2018,29(8):2350-2370. (WANG Fei,YANG Zhibin,HUANG Zhiqiu,et al. Approach for generating AADL model based on restricted natural language requirement template[J]. Ruan Jian Xue Bao/Journal of Software, 2018,29(8):2350-2370.)

[12] 陈淑珍,陈荣武,李耀. 基于 SCADE 的安全软件开发方法研究[J]. 铁路计算机应用, 2015,24(3):14-18. (CHEN Shuzhen,CHEN Rongwu,LI Yao. Method of SCADE-based safety software development[J]. Railway Computer Application, 2015,24(3):14-18.)

[13] 傅亮,潘明罕,严俊. 基于 SCADE 模型驱动的软件集成设计[J]. 航空电子技术, 2013,44(3):26-30. (FU Liang,PAN Minghan,YAN Jun. A model driven software integration design based on SCADE[J]. Avionics Technology, 2013,44(3):26-30.)

[14] 陈明铝,张立臣. AADL 对月球车导航系统的设计与建模[J]. 计算机应用与软件, 2013,30(11):235-237. (CHEN Minglyu,ZHANG Lichen. Designing and modelling lunar rover navigation system with AADL[J]. Computer Application and Software, 2013,30(11):235-237.)

[15] 李虎,马晋,郑凤,等. SCADE 模型驱动开发过程研究及高安全性分析[J]. 航空电子技术, 2013,44(1):15-19. (LI Hu, MA Jin,ZHENG Feng,et al. SCADE model-driven development process research and high security analysis[J]. Avionics Technology, 2013,44(1):15-19.)