

模型驱动的开放式结构悬挂物管理系统设计方法

冯金富, 刘 安, 左 伟, 梁晓龙

(空军工程大学 工程学院, 陕西 西安 710038)

摘 要:开放式结构悬挂物管理系统是机载悬挂物管理系统的发展方向,可以将武器的设计与具体实现相分离,实现武器集成的松耦合。提出基于模型驱动的开放式结构悬挂物管理系统设计方法,采用 UML 设计平台无关模型,采用 AADL 设计平台相关模型,论述了基于 UML Profile 的 UML 模型到 AADL 模型的转换方法,最后,以一个设计实例描述了该方法的设计过程。结果表明:基于模型驱动的开发方法为开放式结构悬挂物管理系统的设计提供了有效的解决方案,实现了系统设计与平台实现的分离。

关键词:模型驱动架构;开放式结构;统一建模语言;UML 扩展机制;体系结构分析与设计语言

DOI:10.3969/j.issn.1009-3516.2010.04.001

中图分类号: TP311;TP391.9 **文献标识码:** A **文章编号:** 1009-3516(2010)04-0001-05

现有的机载悬挂物管理系统是针对特定的飞机平台和机载武器独立设计的,是与载机平台和机载武器紧密耦合的封闭式系统。当需要加装任何一种新型武器时,均需对悬挂物管理系统进行软、硬件的更改,甚至需要重新设计和重新构形,导致开发周期长和研制成本高等问题。飞机的作战需要使得开放式体系结构成为现代飞机悬挂物管理系统发展的必然趋势^[1-2]。目前,针对开放式结构悬挂物管理系统已经制定了部分标准,如通用开放体系结构框架(AS4893),飞机/悬挂物接口通用框架(AIR5532),飞机/悬挂物通用接口控制文件格式(AS5609)等。但是,关于如何设计开放式结构悬挂物管理系统却没有一个有效的解决方法,因为涉及到许多不同的开发语言、执行平台环境及通信过程,因此需要从方法学上对其提供指导。对象管理组织制定的模型驱动体系架构^[3](Model Driven Architecture, MDA)可以很好地解决以上问题,其强调通过对系统的建模来完成系统需求分析,体系结构设计、构建、测试、运行和维护工作^[4-5]。本文提出了基于模型驱动的开放式结构悬挂物管理系统设计方法,采用 UML 设计平台无关模型,采用 AADL 设计平台相关模型,研究了基于 UML Profile 的 UML 模型到 AADL 模型的转换方法。

1 总体设计

基于模型驱动的开放式结构悬挂物管理系统设计过程见图 1,该系统共分 5 大步骤。计算独立模型通过对开放式结构悬挂物管理系统领域要求进行需求分析,描述出系统最终所要达到的目标,这个阶段包括需求建模和需求验证;平台无关模型是在计算独立模型的基础上,对系统的功能进行深入地设计,以找到实现系统所有需求的方案,该阶段包括平台无关模型设计、模型的转换验证;平台相关模型在平台无关模型的基础上设计系统的具体实现,以找到实现系统所有功能的方案,该阶段包括模型设计、模型转换验证;代码生成器根据系统设计阶段产生的系统模型和配置文件与开放式结构悬挂物管理系统运行平台相结合,自动生成系统实现代码,然后通过软件集成将自动生成的代码分别编译成不同硬件平台上的软件,并通过系统软件部署策略集成为统一的软件应用系统,最终生成可执行的开放式结构悬挂物管理系统。系统的验证是在系统

* 收稿日期:2009-12-07

基金项目:国家“863 高技术研究发展计划资助项目(2007AAJ127;2007AAJ 10)

作者简介:冯金富(1964-),男,江苏泰兴人,教授,博士生导师,主要从事武器控制系统、数据融合等研究。

E-mail:blueandylaw@hotmail.com

模型的指导下进行的,所以在系统开发的不同阶段都可以对系统进行测试,每个阶段的测试都可以验证系统模型的正确性,如果测试结果需要修改系统模型,则开发进入了上一级系统模型阶段,进行新一轮的迭代开发。在本文中,计算无关模型和平台无关模型采用 UML 进行建模,平台相关模型采用 AADL 进行建模。

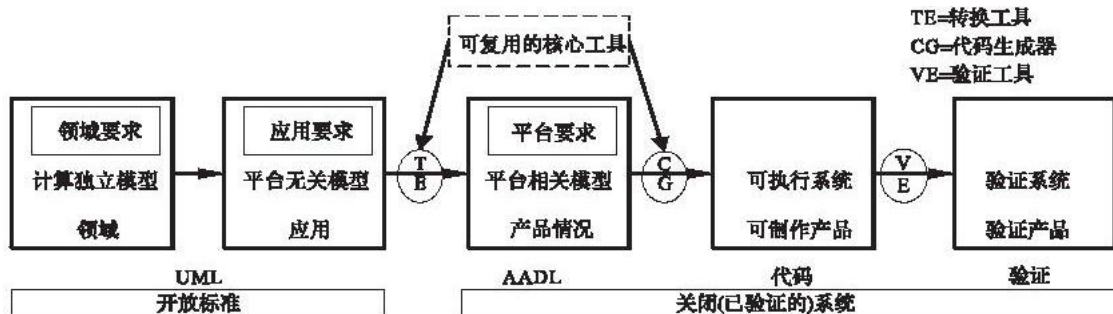


图1 开放式结构悬挂物管理系统的开发流程

Fig. 1 Development flow of open architecture store management system

2 关键技术

表1 UML 模型与 AADL 模型的映射规则

Tab. 1 Mapping rule of UML model and AADL model

类型	AADL 模型	UML 模型
软件部件	进程	类
	线程(组)	状态图
	数据	属性
	子程序	操作
执行平台部件	处理器	类
	内存	类
	总线	类
	装置	参与者
属性	属性集	属性
操作方式	方式	状态机
特性和共享访问	端口(组)	端口
	子程序特性	操作
	子程序参数	参数
	子程序访问	接口
连接和流程	端口连接	构件图内端口之间的连接
	参数连接	活动图内对象数据流的连接
	访问连接	需要特定数据访问的数据端口
	流程规范	活动图内对象接口组成数据流
组件联系	扩展	泛化
	实现	实现

2.1 模型到模型的转换

2.1.1 UML 模型与 AADL 模型的映射规则

在源模型与目标模型之间建立映射规则,基于映射规则,遵循元模型的任何模型都可进行相应的转换。UML 模型与 AADL 模型的映射规则主要有 3 类^[6]:①UML 类到 AADL 构件(软件构件和执行平台构件)的映射;②UML 类的主要属性等映射到 AADL 构件的主要属性;③AADL 构件的特有属性如端口由 UML 的依赖聚合关系来映射。通过对 UML 模型与 AADL 模型的分析,可以得出两者之间的映射规则见表 1。

2.1.2 基于 UML profile 的模型转换

本文采用 UML profile 实现 UML 平台无关模型到 AADL 平台相关模型的转换。UML profile 是利用构造型和标记值来对 UML 元类进行扩展的机制^[7]。本文中的 UML profile for AADL 包括基础包(AADL-Foundations)和设计模型包(AADL-DesignModel),总体结构见图 2。

2.1.2.1 基础包(AADL-Foundations 包)

基础包提供 UML 模型到 AADL 模型转换的基础扩展机制,主要包括以下子包:

1) 非功能属性子包(NFPs Package):主要用来描述与功能无关的扩展,比如:数据单位、数据类型及约束类型等。

2) 时间建模子包(Time Package):用来描述与时间相关的模型与语法扩展,对于开放式结构悬挂物管理系统,时间建模主要考虑 3 种情况:异步时间建模、同步时间建模和真实物理时间建模,该子包对 3 种建模方式提供了基本框架(时间的类型、属性、约束方式及事件的种类等),为 AADL-DesignModel 包中硬件资源建模和软件资源建模提供基础。

3) 通用资源建模子包(GRM Package):用来描述与开放式结构悬挂物管理系统平台实现相关的通用资

源建模,主要包括处理器调度资源、并发/同步/互斥资源、通信资源及通用的任务调度算法等。

4) 分配建模子包(Alloc Package):从空间和时间上对各种资源分配到具体执行平台上进行建模,使最终的开放式结构悬挂物管理系统能够保证资源的可分配性和算法的可调度性。

2.1.2.2 设计模型包(AADL-DesignModel包)

设计模型包继承 AADL Foundations 包,通过硬件资源建模子包(HwResourceModel Package)和软件资源建模子包(SWResourceModel Package)实现 UML 模型到 AADL 模型的转换。

1) 硬件资源建模子包:主要对 UML 建立的平台无关模型映射到由 AADL 描述的具体硬件平台进行建模,包括逻辑和物理子包,逻辑子包主要考虑具体硬件平台的处理器资源、内存资源、存储器资源、时间资源、通信资源及设备资源方面的建模;物理子包主要考虑具体硬件平台的布局和电源管理方面的建模。

处理器资源(HwComputing)建模子包:描述针对一个具体开放式结构悬挂物管理系统所采用的处理器类型,如通用 CPU(x86 系列、POWER PC、ARM 等),ASIC, FPGA, CPLD 等。

内存资源(HwMemory)建模子包:一方面为处理器资源建模提供必须的内存如 RAM, ROM 和高速 CACHE 等,另外则提供系统断电后的永久性存储如 FLASH, EEPROM 等。

存储器资源(HwStorage)建模子包:描述存储器管理的策略如随机访问、DMA 或 MMU 方式等。

通信资源建模(HwCommunication)子包:描述一个具体开放式结构悬挂物管理系统各个硬件资源之间的通信模型如端-端通信、基于总线的通信、基于桥结构的通信等。

时间资源(HwTiming)建模子包:描述系统的真实物理时间,对精确的时钟周期进行建模。

设备资源(HwDevice)建模子包:描述系统的其它硬件外设的类型,例如传感器、输入激励设备、输出控制设备等。

2) 软件资源建模子包主要用于对映射到具体硬件平台上的软件进行建模,包括并发管理、通信交互和资源管理 3 方面。

并发管理(SWConcurrency)建模子包:描述实时操作系统中进程(或线程)的就绪、运行、挂起等状态的建模,实现多进程(或线程)的调度及系统的中断管理。

通信交互(SWInteraction)建模子包:对进程(或线程)之间的通信资源和通信方式进行建模,通信资源包括通知类型、消息类型及互斥量类型等,通信方式包括设置临界区、信号量和高级通信原语等。

资源管理(SWResource)建模子包:对软件部件(数据、子程序、线程、线程组和进程)进行分类说明,实现软件部件的连接、通信的流程以及工作状态方式的表示。

2.2 模型到代码的转换

本文采用基于模板机制的方法实现 AADL 模型到代码的转换,其基本原理是:首先设计一个通用的模板框架,在该模板中预先指定特殊的占位符来保存从模型中提取的数据,通过查询的方式从元模型库中选择和提取占位符处的相关数据,然后这些数据通过基本块中定义的相关表达方法,按照模板框架的规则通过模型到代码转换工具最终转换成源代码。具体方法及步骤参见文献[8]。

3 应用实例

基于本文提出的设计方法完成了一个简单的开放式结构悬挂物管理系统的设计,该系统以武器管理计算机为中心,通过 1553B 总线模块(RT 模式)接入机载航电总线,实现与导航系统、火控系统及座舱显示系统等的通信;通过 1553B 总线模块(BC 模式)接入 GJB1188A 总线,实现与机载武器挂点之间的通信;另外,通过离散量输入/输出模块完成关键性指令的处理。系统结构见图 3。

悬挂物管理计算机是实现开放式结构悬挂物管理系统核心,负责对各个挂点上的悬挂物进行管理,包括悬挂物初始化、悬挂物准备和悬挂物发控逻辑:①悬挂物初始化:负责悬挂物信息查询、确定及初始化管理;

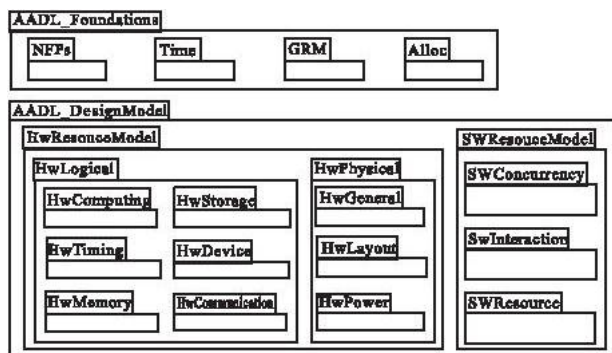


图 2 UML profile for AADL 总体结构

Fig. 2 Overall structure of UML profile for AADL

②悬挂物准备:包括武器使用方案管理和各型机载武器的准备控制流程,悬挂物管理系统根据武器的准备状态、作战条件和目标状态,形成允许放射信号以解除对机载武器发射的联锁;③悬挂物发控逻辑:负责在飞行员进行发射/投放或投弃操作的前提下,依据已选定武器使用方案和武器的准备状态,完成武器的发射/射击/投放控制^[9]。下面以悬挂物管理计算机为例介绍基于模型驱动的系统设计方法。

第1阶段:采用UML的状态图建立系统的平台无关模型,然后通过UML profile for AADL 对其进行扩展,见图4。

第2阶段:扩展后的平台无关模型通过模型转换得到由AADL建立的系统平台相关模型,图5是系统平台相关模型的部件图:硬件部件的CPU采用三星公司ARM系列处理器S3C2440A实现,SRAM由2片1M*16bit的IS61WV102416实现,SDRAM由1片1M*32bit的MT48LC4M32B2实现,FLASH由1片64M*8bit的K9F1208U0M实现,FPGA由EP1C12Q240I7实现,S3C2440A通过AMBA总线与Memory,Flash和FPGA进行通信。软件部件由初始化子程序(PnPWinInitiate)、准备子程序(PnPWinPreapare)和发控子程序(PnPWinFireControl)组成。图6表示UML活动图(图4)中的数据处理(Data Process)过程映射到图5所示硬件平台上的线程模型,该模型采用的监听/分派的方式:监听线程ListenThread以固定周期读1553B总线上的数据,并写入接收数据缓存BufferIn;数据处理线程DataThread周期性的读取接收数据缓存内的数据,对数据进行处理后,立即(非周期性的)写入发送数据缓存;分派线程DispatchThread通过周期性的读取发送数据缓存,将数据写入1553B总线。通过对图5中的UML活动图各个过程建立平台相关的线程模型,最终实现系统软件的平台相关建模。

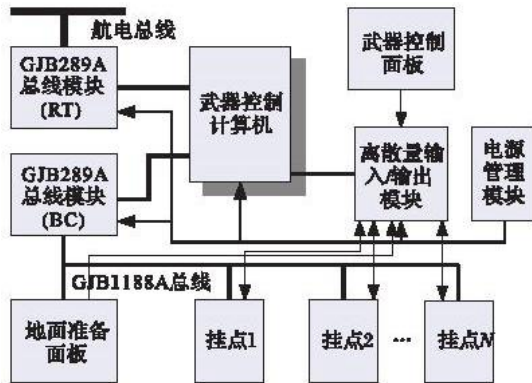


图3 一个简单的开放式结构悬挂物管理系统
Fig.3 A simple open architecture SMS

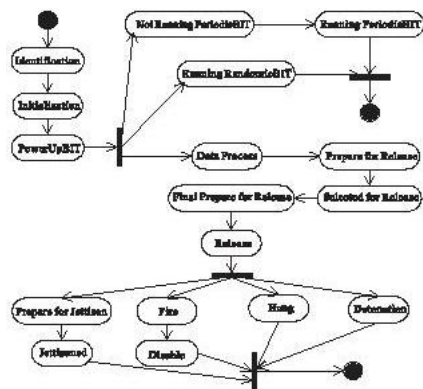


图4 悬挂物管理计算机UML状态图
Fig.4 UML state-machine graph for store management computer

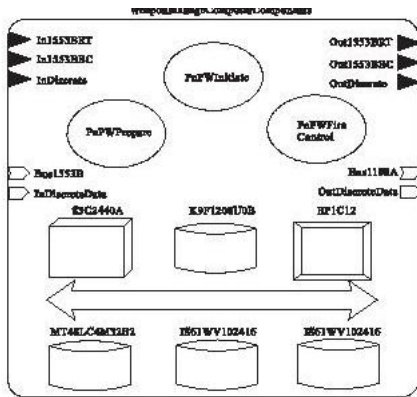


图5 悬挂物管理计算机部件图
Fig.5 AADL component graph for store management computer

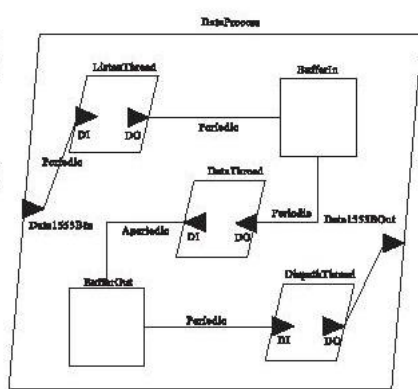


图6 数据处理线程模型
Fig.6 Model of data process threads

第3阶段:通过AADL建模工具OSATE对经模型转换后得到的AADL模型进行分析,比如系统结构分析,可调度性分析和端-端数据流分析,根据系统分析结果判断生成后的系统是否满足设计需求。

1)系统结构分析:包括模型静态分析、端口连接一致性检查、总线负载分析、资源分配及资源消耗预算分析等通过系统结构分析可以检查和验证AADL模型在系统结构上是否满足设计者的需求,保证整个系统能够正常运行;另外为设计者选择哪种类型的系统结构(比如选择哪种类型CPU,CPU的工作频率是多少、CPU缓存是多少、采用哪种总线与其它设备通信等)提供了判断标准。

2)可调度性分析:包括查看处理器线程分配、进程分配结果以及处理器可调度能力分析等功能。该分析结果可用于描述开放式结构悬挂物管理系统多任务的调度情况,为开放式结构悬挂物管理系统是否满足实时性要求提供定量分析依据。

3)端-端数据流分析:该分析结果可用于描述开放式结构悬挂物管理系统中,一组由若干任务组成的

事件的数据传输延时情况,为开放式结构悬挂物管理系统中所有的事件是否满足其响应时间提供定量分析依据。其结果分为3类:错误、警告和正确。其中错误表示某个端-端数据流超过系统的设计要求;警告表示某个端-端数据流可以满足系统设计要求,但是会随着系统资源利用的紧张程度及系统负载的不同等因素出现未知情况;正确表示某个端-端数据流完全满足系统的设计要求。

第4阶段:根据第3阶段经过验证的AADL平台相关模型,通过基于模板机制的模型到代码转换,最终得到在具体硬件平台上的可执行源代码。

4 结论

本文通过一个简单的开放式结构悬挂物管理系统的设计,验证了基于模型驱动的开发方法可以为开放式结构悬挂物管理系统的设计提供了一个有效的解决途径,采用UML profile能很好地实现UML模型到AADL模型的自动转换,采用模板机制能很好地实现AADL模型到代码的自动转换。该方法分离了开放式结构悬挂物管理系统的设计和功能的实现,优化了开放式结构悬挂物管理系统的设计层次,降低了设计的复杂性,实现了武器系统集成的松耦合。

参考文献:

- [1] Gibbs Michael R, Wit Jeffrey S, Benedick Fred L. The Impact of the Development of Universal Interfaces and Model Driven Architecture Development on Attaining Plug and Play Systems[R]. AIAA 2007 - 2921.
- [2] Douglas Gregory. Air Weapons Integration - Plug and Play Weapon[R]. London: SAE Aerospace ASD, 2006.
- [3] Object Management Group. Unified Modeling Language: Superstructure Version 2.0[S]. 2005.
- [4] Object Management Group. MDA guide version 1.0.1[S]. 2003.
- [5] OMG Adopted Specification. A UML Profile for MARTE: Modeling and Analysis of Real - Time Embedded Systems, Beta 2 [DB/OL]. [2008 - 06 - 08]. <http://www.omg.org/spec/ptc/2008 - 06 - 08>.
- [6] SAE. SAE Standards: Architecture Analysis & Design Language (AADL, AS5506) [S]. 2004.
- [7] Hugues J, Zalila B, Pautet L. Rapid Prototyping of Distributed Real - Time Embedded Systems Using the AADL and Ocarina [C]//Proceedings of the 18th IEEE/IFIP International Workshop on Rapid System Prototyping (RSP07). Brazil: IEEE Press, 2007: 106 - 112.
- [8] OMG Available Specification. MOF Model to Text Transformation Language [DB/OL]. [2008 - 01 - 16]. <http://www.omg.org/spec/MOFM2T/1.0/PDF>.
- [9] 刘安,冯金富,胡杰,等. 数字化机载武器控制系统半实物仿真平台研究[J]. 系统仿真学报, 2009, 21(1): 96 - 99.
LIU An, FENG Jinfu, HU Jie, et al. Research on Hardware - In - The - Loop Simulation Platform of Digital Airborne Weapon Control System [J]. Journal of System Simulation, 2009, 21(1): 96 - 99. (in Chinese)

(编辑:徐敏)

Model Driven Design Method for Open Architecture Store Management System

FENG Jin - fu , LIU An , ZUO Wei , LIANG Xiao - long

(Engineering Institute, Air Force Engineering University, Xi'an 710038, China)

Abstract: Open architecture store management system is a developing direction of the future airborne weapon system. The motivation of the open architecture SMS is to separate weapon realization from weapon design and to realize loose coupling in weapon integration. A model driven design method for open architecture SMS is presented. In this system, a platform independent model is designed by using UML and platform specific model is designed by using AADL. Some key techniques, such as UML profile for UML model to AADL model transform method etc., are explained in detail. By taking the weapon management computer design for example, the designing flow of the system is provided. Experimental results show that the use of the developing method of the model driven architecture can provide a general solution for the design of the open architecture SMS and achieve the separation of weapon design from the weapon realization.

Key words: model driven architecture; open architecture; unify model language; UML profile; architecture analysis & design language