

基于XML的UML模型向AADL模型的自动转换

万小平,李蜀瑜

(陕西师范大学 计算机科学学院,陕西 西安 710062)

摘要:UML已经成为复杂系统建模的工业标准,并可借助代码自动生成工具实现从分析到编码的开发过程自动化;AADL具有精确的语义和严格的语法规则,可用于描述嵌入式实时系统的软、硬件体系结构,并能分析系统的功能及非功能属性。UML和AADL模型的基本元对象有相互对应关系,并且两种模型的表示都能够采用标准的XML的交换格式。为充分发挥两种语言的优势,文中研究了UML模型向AADL模型的自动转换,以XML为媒介实现两种模型的元对象的对应转换,最终达到两种模型的转换,使用户在设计阶段能够结合运用AADL、UML工具的优点对系统进行分析。

关键词:统一建模语言;架构分析与设计语言;模型转换;实时系统;嵌入式

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2014)03-0071-03

doi:10.3969/j.issn.1673-629X.2014.03.018

Automatic Conversion of UML Model to AADL Model Based on XML

WAN Xiao-ping, LI Shu-yu

(Institute of Computer Science, Shaanxi Normal University, Xi'an 710062, China)

Abstract:UML has become an industry standard of complex system modeling, and can achieve the development processing automation from analysis to coding by means of automatic code generation tools. AADL with a precise semantics and strict syntax specification, can be used to describe the embedded real-time system software and hardware architecture, and analyze the system's functional and non-functional attributes. The basic meta-object of UML and AADL is a mutual correspondence, and the two models representations are able to use standard XML interchange format. In order to give full play to the advantages of the two languages, study the automatic conversion of UML model to AADL model, taking XML as a medium to achieve the two models corresponding meta-object conversion, and ultimately achieve the conversion of the two models, allowing users to combine design phase using AADL, UML tool to analyze the advantages of the system.

Key words:UML; AADL; model transformation; real-time systems; embedded

0 引言

统一建模语言(Unified Modeling Language, UML)是一种定义良好、易于表达、功能强大、用于编制软件蓝图的标准语言,提供用例图、逻辑图、组件图、合作图、时序图、部署图来共同描述软件系统的各个方面,UML扩展机制和对象约束语言增强了语言的非功能属性描述特定系统^[1],如用于嵌入式系统建模的MARTE^[2]。UML主要优点是详细分析软件的行为过程,并可借助代码自动生成工具实现从分析到编码的开发过程自动化^[3],但是对软件非功能属性诸如实时性、功耗性等分析不足^[4]。

AADL具有精确的语义和严格的语法规则,可用于描述嵌入式实时系统的软、硬件体系结构^[5],并能

分析系统的功能及非功能属性。AADL使用单一模型支持多种分析方式,可将系统设计、分析、验证、自动代码生成集成在一个框架中。AADL中系统是由构件组成。构件分为软件构件、执行平台构件以及系统构件3类。软件构件包括进程、线程、线程组、子程序和数据5种用于软件体系结构建模;执行平台构件包括设备、处理器、存储器和总线4种用于硬件体系结构建模;系统构件是软件、执行平台或者系统构件的组合。AADL能够同时获取与描述功能需求和非功能需求,并且在设计阶段能够对功能需求与非功能需求实现进行分析和验证,目前基于AADL模型的实时性、功耗性分析验证工具比较成熟^[6]。因此为了充分利用两种模型的优势,利用现成的分析工具,使模型开发工作

收稿日期:2013-05-27

修回日期:2013-08-30

网络出版时间:2014-01-07

基金项目:中央高校基本科研业务费项目(GK201002011)

作者简介:万小平(1987-),男,湖北武汉人,硕士研究生,研究方向为嵌入式系统;李蜀瑜,博士,副教授,研究方向为嵌入式系统。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140107.1648.028.html>

具有流程清晰,操作简介的特点,而且使用户在进行模型设计和分析验证时,既可以使用 UML 也可以使用 AADL,充分利用两种模型功能特点,在分析设计阶段更好的验证系统的特性,成为目前的研究热点。

文中将 UML 开发工具与 AADL 开发工具集成在一个开发平台,根据 UML 和 AADL 模型的基本元对象有相互对应关系^[7],在两种模型的表示都能够采用标准的 XML 的交换格式^[8]基础上研究 UML 模型向 AADL 模型的自动转换。

1 UML 模型与 AADL 模型转换

1.1 转换原理

在 Eclipse 平台上,AADL 模型与 UML 模型都是以 XML 结构形式存储。由于 XML 具有跨平台性、依赖于内容、结构化性和易被加载解析性,使得 XML 是当前处理结构化文档信息的有力工具。XML 的简单使其易于在任何应用程序中读写数据,虽然不同的应用软件也支持其他的数据交换格式,但这些工具软件都支持或者即将支持 XML。ATL 模型转换语言实际上也运用到 XML^[9],转换过程中需制定 UML 和 AADL 转换规则。转换流程见图 1。

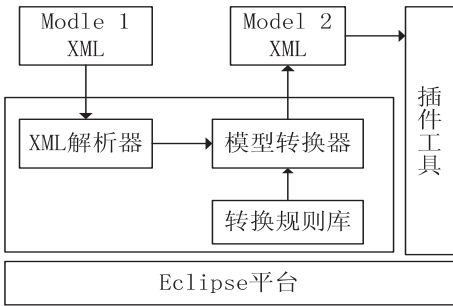


图 1 转换流程

模型转换系统是基于 Eclipse 平台之上,主要包含三个部分:模型 XML 描述,模型转换和插件分析工具^[10]。其中模型转换又包含三个部分:XML 解析器、模型转换器和转换规则库。XML 解析器主要解析模型 XML 文件,转换规则库存储不同模型之间的转换对应规则^[11],模型转换器根据 XML 解析器对 UML XML 的解析在转换规则库匹配搜索对应的规则将模型进行转换,生成转换后的模型 AADL XML 文件。插件工具是对模型的分析或评估工具。

1.2 转换规则

AADL 的构建被定义成两部分:类型和实现。一个 AADL 的构建拥有一个类型以及对应的多个或 0 个实现。构建的种类分为软件构建、硬件构建和组合构建,其中软件构建包括:数据、线程、线程组、进程、子程序。以及各构建之间的关系。

基本元素的对应:

● AADL 系统代表了软件和执行平台或者系统组件,由 UML 中 SysMLS 块代表。AADL 组件由 UML 中的部分代表。

● AADL BUS 和 DATA 组件连接:在 MARTE 中由 UML 的接口申明。

● AADL Ports 由 MARTE Ports 建模。

● AADL 子程序的调用,由 UML 序列图表示。

● AADL 模式变换由 UML 状态机表示。

软件构建的转换规则:

● 规则-1:每个非抽象 UML 类转换成 AADL 中的一个定义接口和定义了其组成实现的类型。

● 规则-2:每个抽象 UML 类转换成 AADL 中没有实现的类型。

● 规则-3:UML 中两个非抽象类 A 和 B,其中 B 继承 A,每个类转换成 AADL 的一个类型及其实现。在 AADL 中 B 类型变成类型 A 的扩展。

● 规则-4:UML 中一个抽象类 B 和非抽象类 A,其中 B 继承 A,转换成 AADL 中 B 类型是 A 的扩展,而且 B 类型没有实现。

● 规则-5:UML 一个非抽象类 B 继承抽象类 A。与规则 4 相反。

执行平台的转换规则:

● 规则-1:UML 硬件对象根据命名规则转换成 AADL 相应的 device,bus,memory。UML 扩展机制对象的属性转换成 AADL 的硬件限制属性。

2 实验与分析

实验在 Eclipse 平台上,运用 Papyrus 工具对汽车巡航系统进行建模,使用 OSATE 工具^[12]读取转换后的 XML 并分析,Papyrus 是基于 Eclipse 平台上的一个插件,能够使用 MARTE 对实时嵌入式系统的属性进行很好的建模,并将模型数据以 XML 的形式进行保存,汽车巡航系统的类图和序列图如图 2 和图 3 所示。

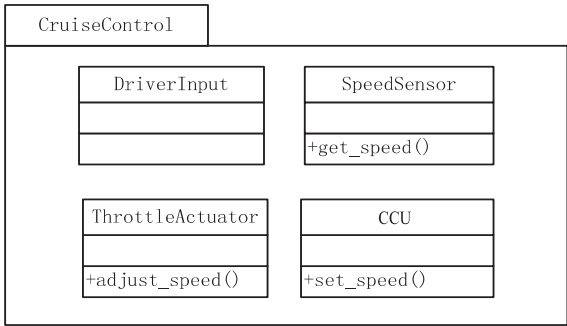


图 2 巡航控制系统类图

```
-<packageImportxmi:id="_3-Q68bSGEeKcWvyIV0Exyg">
<importedPackagexmi:type="uml:Model" href="pathmap://
UML_LIBRARIES/UMLPrimitiveTypes.library.uml#_0"/>
</packageImport>
```

```
-< packagedElementxmi; type = " uml; Package " xmi; id = " _
70JAULSGEeKcWvyIV0Exyg" name = " CrusieControl" >
  < packagedElementxmi; type = " uml; Class " xmi; id = " _
tATjILSHEeKcWvyIV0Exyg" name = " DriverInput" />
  -< packagedElementxmi; type = " uml; Class " xmi; id = " _
CD0ZMLSIEeKcWvyIV0Exyg" name = " SpeedSensor" >
    < ownedOperationxmi; id = " _ xpKvMLSIEeKcWvyIV0Exyg "
name = " get_speed" />
  </ packagedElement>
  -< packagedElementxmi; type = " uml; Class " xmi; id = " _
PXkHMLSIEeKcWvyIV0Exyg" name = " ThrottleActuator" >
    < ownedOperationxmi; id = " _ 3DnlsLSIEeKcWvyIV0Exyg" name
= " adjust_speed" />
  </ packagedElement>
  -< packagedElementxmi; type = " uml; Class " xmi; id = " _ eBv-
colSIEeKcWvyIV0Exyg" name = " CCU" >
    < ownedOperationxmi; id = " _ APBIULSJIEeKcWvyIV0Exyg "
name = " set_speed" />
  </ packagedElement>
  </ packagedElement>
  .....
  -< ownedBehaviorxmi; type = " uml; Interaction " xmi; id = " _ Ud-
MHwLSJIEeKcWvyIV0Exyg" name = " ControlSpeed" >
    .....
    < messagexmi; id = " _ 3F4aELSIEeKcWvyIV0Exyg" name = " set_
speed " messageSort = " asynchSignal " receiveEvent = " _
3FcVMLSJIEeKcWvyIV0Exyg " sendEvent = " _
3FSkMLSJIEeKcWvyIV0Exyg" />
    < messagexmi; id = " _ 5DNCYLSJIEeKcWvyIV0Exyg" name = " get
_ speed " messageSort = " asynchSignal " receiveEvent = " _
5CwWcbSJIEeKcWvyIV0Exyg " sendEvent = " _
5CwWcLSJIEeKcWvyIV0Exyg" />
    < messagexmi; id = " _ 6dWsMLSJIEeKcWvyIV0Exyg" name = "
adjust_ speed " messageSort = " asynchSignal " receiveEvent = " _
6dDxQLSJIEeKcWvyIV0Exyg " sendEvent = " _
6c6AQLSJIEeKcWvyIV0Exyg" />
  </ ownedBehavior>
```

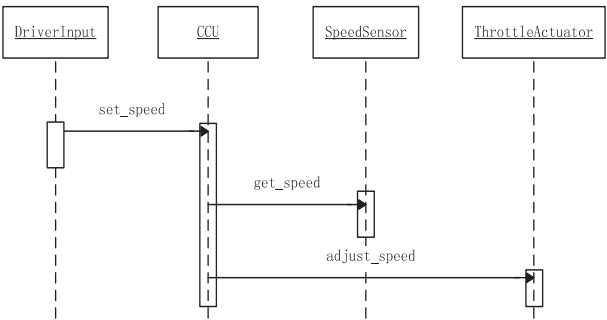


图 3 巡航控制系统序列图

转换后的 AADL 模型 XML 描述为:

```
<? xmlversion = " 1. 0" encoding = " UTF-8" ? >
<core; AadlSpec xmi; version = " 2. 0" xmlns: xmi = " http://
www. omg. org/ XMI" xmlns: core = " http:// AADL/ core" name = "
```

```
CrusieControl" >
  <systemType name = " CruiseControl" />
  <systemImpl name = " CruiseControl. imp" compType = "/"
aadlSpec[ @ name = CrusieControl ]/systemType[ @ name = CruiseC-
ontrol" />
  <threadType name = " CCU" />
  <threadImpl name = " CCU. imp" compType = "/" aadlSpec[ @
name = CrusieControl ]/threadType[ @ name = CCU" />
  <callSequences>
    <callSequence>
      <call name = " get_speed" classifier = "/" aadlSpec[ @
name = CrusieControl ]/subprogramImpl[ @ name = get _ speed.
imp" />
      <call name = " adjust_speed" classifier = "/" aadlSpec
[ @ name = CrusieControl ]/subprogramImpl[ @ name = adjust _ speed.
imp" />
    </ callSequence>
  </ callSequences>
</ threadImpl>
<deviceType name = " SpeedSensor" />
<deviceType name = " ThrottleActuator" />
<deviceType name = " DriverInput" />
<subprogramType name = " get_speed" />
<subprogramType name = " adjust_speed" />
<deviceImpl name = " DriverInput. DeviceImpl1" compType
= "/" aadlSpec[ @ name = CrusieControl ]/deviceType[ @ name = Driv-
erInput" />
<deviceImpl name = " SpeedSensor. imp" compType = "/" aadl-
Spec[ @ name = CrusieControl ]/deviceType[ @ name = SpeedSen-
sor" />
<deviceImpl name = " ThrottleActuator. imp" compType = "/"
aadlSpec[ @ name = CrusieControl ]/deviceType[ @ name = Throttle-
Actuator" />
<subprogramImpl name = " get_speed. imp" compType = "/"
aadlSpec[ @ name = CrusieControl ]/subprogramType[ @ name = get_
speed" />
<subprogramImpl name = " adjust_speed. imp" compType
= "/" aadlSpec[ @ name = CrusieControl ]/subprogramType[ @ name =
adjust_speed" />
</ core; AadlSpec>
```

3 结束语

基于 XML 的 UML 模型到 AADL 模型的转换过程中利用现成的工具,根据转换规则完成两种模型的自动转换生成,传统的 UML 模型对软件的非功能属性分析验证不足,文中采用 UML 模型转换成 AADL 模型改变了这种状况,使应用系统开发过程中其他阶段对模型的分析验证变为可能并且为运用其他分析验证工具打下了基础。

- [2] Hadad E, Lamsade J. TQoS: Transactional and QoS-aware selection algorithm for automatic Web service composition[J]. IEEE transactions on services computing, 2010, 3(1): 73-85.
- [3] Zeng Liangzhao, Benatallah B, Dumas M, et al. Quality-driven Web services composition [C]//Proc of 12th international conference on the World Wide Web (WWW). [s. l.]: [s. n.], 2003: 411-421.
- [4] 孔维梁, 刘清堂, 杨宗凯, 等. 基于动态 QoS 的 Web 服务组合[J]. 计算机科学, 2012, 39(2): 268-272.
- [5] 万里平, 高春鸣, 肖伟, 等. 基于 QoS 的 Web 服务组合的全局优化方法[J]. 计算机工程与应用, 2007, 43(24): 65-68.
- [6] 蔡美玲, 李茂桂, 周杰. 基于多目标遗传算法的多选择 Web 服务组合方法[J]. 计算机工程与应用, 2010, 46(13): 202-205.
- [7] 王飞明, 董荣胜, 胡元闯, 等. 全局优化的分层动态 Web 服务组合[J]. 计算机工程与应用, 2011, 47(12): 72-74.
- [8] Shani G, Pineau J, Kaplow R. A survey of point-based POMDP solvers[J]. Auton Agent Multi-Agent Syst, 2013, 27(1): 1-51.
- [9] Eker B, Akin H L. Solving decentralized POMDP problems using genetic algorithms[J]. Auton Agent Multi-Agent Syst, 2013, 27(1): 161-196.
- [10] Bernstein D S, Givan R, Lmmerman N, et al. The complexity of decentralized control of Markov decision process[J]. Mathematics of operations research, 2002, 27(4): 819-840.
- [11] Bosse T, Hoogendoorn M, Michel C, et al. Modeling collective decision making in groups and crowds: Integrating social contagion and interacting emotions, beliefs and intentions[J]. Auton Agent Multi-Agent Syst, 2013, 27(1): 52-84.
- [12] Chesani F, Mello P, Montali M, et al. Representing and monitoring social commitments using the event calculus[J]. Auton Agent Multi-Agent Syst, 2013, 27(1): 85-130.
- [13] Spiliopoulos L. Strategic adaptation of humans playing computer algorithms in a repeated constant-sum game[J]. Auton Agent Multi-Agent Syst, 2013, 27(1): 131-160.
- [14] Wang Shuying, Shen Weiming, Hao Qi. An agent-based Web service workflow model for inter-enterprise collaboration[J]. Expert systems with applications, 2006, 31(4): 787-799.
- [15] 周宁. Web 服务组合中若干问题的研究[D]. 南京: 南京大学, 2011.
- [16] 刘莉平. 动态 Web 服务组合关键技术研究[D]. 长沙: 中南大学, 2011.
- [17] Azar M, Munos R, Ghavamzadeh M, et al. Speedy Q-learning [C]//Proc of twenty-fifth annual conference on neural information processing systems. [s. l.]: [s. n.], 2011: 1-9.
- [18] Bertsekas D P, Yu Huizhen. Q-learning and enhanced policy iteration in discounted dynamic programming[J]. Mathematics of operations research, 2012, 37(1): 66-94.
- [19] Saad H, Mohamed A, ElBatt T. Distributed cooperative Q-learning for power allocation in cognitive femtocell networks [C]//Proc of Vehicular Technology Conference (VTC Fall). Quebec City: IEEE, 2012: 1-5.
- [20] Chen Zhe, Qiu R C. Cooperative spectrum sensing using Q-learning with experimental validation [C]//Proc of IEEE. [s. l.]: [s. n.], 2011: 405-408.

(上接第 73 页)

参考文献:

- [1] Quadri I R, Gamatié A, Boulet P. Expressing embedded systems configurations at high abstraction levels with UML MARTE profile: Advantages, limitations and alternatives[J]. Journal of systems architecture, 2012, 58(5): 178-194.
- [2] Colbert E. Overview of the UML profile for the architecture analysis & design language [EB/OL]. 2004. <http://aadl.sei.cmu.edu/aadl/documents/041103%20AADL%20Profile%20SEApdf>.
- [3] Marchal B. Working XML: UML, XMI, and code generation, part1 [EB/OL]. 2004-04-15. http://www.ibm.com/developerworks/library/xwxm23/index.html?S_TACT=105AGX52&S_CMP=cn-a-x.
- [4] 李双庆, 曹银龙. 基于 UML 扩展的实时系统建模方法[J]. 计算机工程与设计, 2006, 27(24): 4778-4780.
- [5] Feiler P H, Gluch D P, Hudak J J. The Architecture Analysis and Design Language (AADL): An introduction [R]. [s. l.]: [s. n.], 2006.
- [6] Li Qiuyan, Tian Jie, Pei Qihong, et al. Automatic transformation technology from AADL model to UML model [C]//Proc of 2011 IEEE 3rd international conference on communication software and networks. [s. l.]: [s. n.], 2011: 255-258.
- [7] 许峰, 王志坚. 基于 MDA 的构件开发方法的研究[J]. 计算机应用与软件, 2009, 26(11): 30-33.
- [8] 熊永刚, 唐慧佳. 基于 XMI 的 UML 模型到 XML 文档转换的实现[J]. 计算机应用与软件, 2010, 27(4): 68-70.
- [9] 应一舟, 叶丽君, 郭义喜. 基于 XML 的 UML 时序图向 Petri 网的转换[J]. 计算机工程, 2009, 35(22): 84-87.
- [10] 赵俊峰, 周建涛. UML 时序图向 PNML 转换的研究与实现[J]. 武汉大学学报(理学版), 2011(6): 511-516.
- [11] 严丽, 马宗民, 刘健, 等. 模糊 XML DTD 到 UML 数据模型的转换[J]. 小型微型计算机系统, 2009(4): 586-593.
- [12] 王瀚博, 周兴社, 董云卫, 等. 结构分析和设计语言 AADL 研究[J]. 计算机工程与应用, 2009, 45(16): 1-4.

基于XML的UML模型向AADL模型的自动转换

作者：[万小平](#)，[李蜀瑜](#)，[WAN Xiao-ping](#)，[LI Shu-yu](#)
作者单位：[陕西师范大学 计算机科学学院](#),[陕西 西安](#),[710062](#)
刊名：[计算机技术与发展](#)

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2014(3)

本文链接：http://d.wanfangdata.com.cn/Periodical_wjfz201403018.aspx