

构件式建模方法在系统建模中的应用研究^{*}

方林波, 余志勇, 武红霞, 付光远, 罗 眉

(第二炮兵工程大学, 西安 710025)

摘要: 为了使系统级建模过程工程化, 提高模型构件的重用度, 针对建模的实现过程, 提出了通用的建模模式和构件式建模方法。将数据抽象、信号原语、异步并发机制处理等过程归纳为建模模式, 并建立了典型的软硬件构件模型以便建模时重用, 方便了快速建模和模型构件的重用; 以无线接入系统为例, 运用构件式建模方法建立了其系统模型, 仿真分析了 WLAN 网络环境下站点数等参数对有效吞吐量的影响。实践表明, 构件式建模方法能加快系统的建模过程, 有助于实现建模过程的工程化。

关键词: 模型; 仿真; 构件式建模; 无线局域网

中图分类号: TP391.9

文献标志码: A

文章编号: 1001-3695(2012)10-3776-04

doi:10.3969/j.issn.1001-3695.2012.10.044

Study on component-based method of system-level modeling

FANG Lin-bo, YU Zhi-yong, WU Hong-xia, FU Guang-yuan, LUO Mei

(The Second Artillery Engineering University, Xi'an 710025, China)

Abstract: In order to engineer system-level modeling process, improve the reusability of the model components, this paper proposed a general modeling method and component-based modeling approach. Uniformly, data abstraction, signal primitives, asynchronous concurrent mechanism and other processes of modeling were summarized as modeling patterns. Furthermore, to facilitate the rapid prototyping and improve the reuse of model components, this paper created a typical hardware and software component model. Taking the wireless access system as an example, which was built according to component-based modeling method based on the simulation of the model, the influences of the number of stations on the valid throughput were analyzed in the WLAN (wireless local-area network). The practice shows that the component-based modeling method can speed up the modeling process and contribute to the achievement of the modeling process engineering.

Key words: models; simulation; componet-based modeling; wireless local-area network (WLAN)

嵌入式系统设计面临着系统复杂性的不断增加以及设计周期的逐渐缩短等挑战, 这些挑战来自于设计初期模糊的设计参数及非功能性需求 (低成本、低功耗等) 的不断增长等^[1]。传统的嵌入式系统设计方法很难保证最终产品设计的正确性及合理性。系统级设计方法在嵌入式系统的软/硬件实现之前, 通过建立系统的可执行模型, 定性和定量地分析系统的性能, 对系统的需求及规格说明进行有效评估, 从而确定最初设计方案的可行性, 并及时发现设计错误, 避免设计过程的反复^[2]。

系统级设计方法的难点在于如何快速有效地建立反映系统功能行为的性能模型。不同的嵌入式系统有许多构件 (如处理器、缓存器等) 是相同的或有相似的属性, 但是每次建立这些构件模型时均需要从头开始, 相同或类似的构件模型也得不到重用, 浪费了人力物力^[3]。因此, 如何避免嵌入式系统构件的重复建模, 达到构件式模型的重用, 可以缩短建模的时间。

为了快速构建复杂嵌入式系统的模型以便于系统性能分析, 并能够重用模型构件, 借鉴面向对象领域设计模式的理念, 本文提出了基于 POOSL 的构件式建模方法。

1 基于 POOSL 的系统级设计方法

典型的系统级建模及设计工具有 Berkely 大学的 Ptolemy、Cadence 公司的 VCC、ARM 和 Cadence 等公司提出的开放性设计标准 SystemC、荷兰 Eindhoven 大学的 POOSL 等^[4]。Ptolemy 用 MOC 来建模软件和硬件, 组件之间的交互也由 MOC 控制。VCC 支持重用行为库和算法库, 其体系结构部分也支持重用已有的 IP 组件。SystemC 基于面向对象语言 C++。

面向对象的形式化建模语言 (parallel object-oriented specification language, POOSL) 是专门针对复杂实时软/硬件系统开发而设计的, 基于严密的数学语义, 能精确地描述系统的并发性和分布性等特性, 已经被证明是一种非常高效的分析和评估工业级复杂系统的建模语言, 已成功建模分析了以下系统: network processor^[5]、packet switch system^[6]、IEEE 802.11 协议^[7]等。本文以 POOSL 作为系统级建模语言, 通过实例分析, 研究构件式建模方法在嵌入式系统建模中的应用方法。POOSL 语言由数据类、进程类和聚类三部分组成。其中, 数据类基于传统的面向对象语言如 Smalltalk、C++ 等, 用于描述进程类之间产生和交换的数据信息; 进程类基于时间扩展的进程代数, 用于描述组成部分的并发行为; 聚类用于建立进程类或

收稿日期: 2012-03-29; 修回日期: 2012-05-02 基金项目: 国家自然科学基金资助项目 (60773186)

作者简介: 方林波 (1973-), 男, 四川营山人, 副教授, 博士, 主要研究方向为复杂系统建模及性能分析 (fanglb@bjut.edu.cn); 余志勇 (1972-), 男, 副教授, 博士, 主要研究方向为电磁兼容、信号处理; 武红霞 (1972-), 女, 讲师, 硕士, 主要研究方向为信号处理; 付光远 (1966-), 男, 教授, 博士, 主要研究方向为指挥信息系统; 罗眉 (1978-), 讲师, 硕士, 主要研究方向为嵌入式系统设计。

其他聚类的层次化结构,用于描述系统的拓扑结构^[8]。

2 建模模式和典型构件

面向对象领域中的一个重要突破就是提出了设计模式 (design pattern) 的概念。设计模式描述了一个在本文的环境中不断出现的问题,然后描述了该问题的解决方案的核心。通过这种方式可以无数次地使用那些已有的解决方案,无须再重复相同的工作。

建模模式是如何创建模型的规则,是对生成模型中必要组成部分过程的描述。针对嵌入式系统建模中反复出现的数据抽象、原语的抽象、异步并发的处理和功能实体抽象等实现过程,提出了通用建模模式和构件式建模方法,详细说明了几种建模模式的实现过程,并建立了典型嵌入式系统模型构件。给系统级建模人员提供实际建模的指导方法,包括数据抽象模式、原语抽象模式、异步并发事件抽象模式和功能实体抽象模式。下面对每一个模式的描述中,从解决的问题、解决方案、示例三个方面进行介绍。

2.1 数据抽象

a) 解决的问题。在嵌入式系统中均有数据的流动,因此系统级建模过程中需要解决的一个关键问题是如何抽象系统的被动资源,即数据对象,如何在模型的抽象建立过程中粗粒度地描述系统组件之间交互的数据。

b) 解决方案。用数据类进行抽象,抽象的原则是能够准确无误地描述系统组件的功能行为,但又不增加系统建模的复杂性,只抽取与模型行为相关的基本数据段,省略与模型行为无关的数据段中按位编码的意义。另外,在数据对象的方法中,增加了对相应字段进行修改和读取的函数。

c) 示例。在 802.11 协议栈的建模中,对 MAC 层协议数据单元 MPDU 的抽象采用了这种模式的方法。在 MAC 头字段中省略了实际位的编码意义,只抽取了源地址 (AddrS)、目标地址 (AddrD)、NAV、序列号 (sequenceNo) 等基本字段作为实例变量,为了描述时间特性,增加了进入队列时间 (enterTime) 和确认返回时间 (backTime)。

数据对象的方法提供了对数据对象自身组成属性的操作函数,通过这些方法,可以在对象实例化时设置数据对象的属性,在仿真执行过程中也可以修改数据对象的属性。例如,针对 AddrD 和 AddrS 的设置函数 set_AddrD() 和 set_AddrS() 可以对 MPDU 数据单元的源地址和目标地址进行设置和修改。另外,printString() 函数用于输出数据对象的信息。

2.2 信号原语抽象

a) 解决的问题。原语是协议中不同功能模块之间相互通知的信号,通常对某种操作进行请求 (request)、确认 (confirm)、通知 (indicate) 等,如何实现原语的交互过程是本模式解决的问题。

b) 解决方案。信号原语本身可以用字符串简单表达,也可以用数据类抽象复杂的原语;使用消息机制进行原语的收发,并引入信号量机制结合阻塞语句 (“[]”) 完成原语的交互过程。

2.3 异步并发事件抽象

a) 解决问题。POOSL 语言能够精确地描述异步并发事件,在建模中描述异步并发事件的难点在于系统中没有类似同步并发的集中控制机制,如何处理异步并发的行为。

b) 解决方案。POOSL 语言提供了进程间消息发送 (或接收) 的同步通信机制,可以采用阻塞语句结合信号量 (semaphore) 的方式来处理并发事件。

c) 示例。如图 1 所示,假设 sender 进程类中有两个异步并发事件 eventA() 和 eventB(), eventA() 完成生成和发送数据帧, eventB() 完成已发送数据帧数量的统计工作,为了同步这两项处理任务,引入信号量变量 semaphore。当 eventA 发送完一帧之后,将 semaphore 变量置 “true”,而 eventB 一直阻塞 ([semaphore] 阻塞语句实现),直到 semaphore 变为 “true”,此时 numberOfSentFrame 变量值加 1。

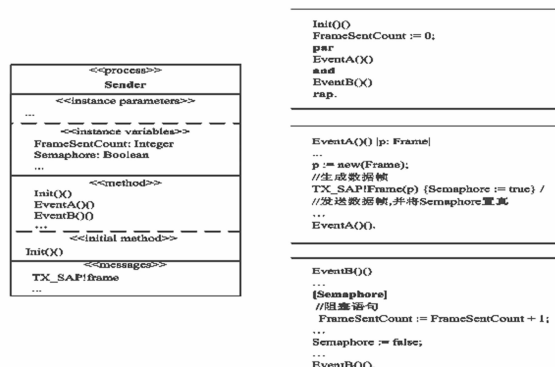


图1 异步并发事件建模模式

2.4 功能实体抽象

a) 解决的问题。在系统建模中,组件之间需要利用消息的发送和接收进行同步与通信,如何处理消息的接收与发送,并对接收到的数据进行相应的处理,典型的功能实体是上层服务访问点和下层服务访问点。

b) 解决方案。用一个实体 ID 表示一个具有上层 SAP 和下层 SAP 的功能实体,实体 ID 用于区分消息接收/发送的不同实体。功能实体用四种消息队列和相应的处理方法来描述。消息队列用于缓存从其他功能实体接收/发送的消息。根据来自不同 SAP 和方向的消息,将消息队列分为四种,即 queueFromUpperSAP、queueToUpperSAP、queueFromLowerSAP 和 queueToLowerSAP。例如,queueFromUpperSAP 用于缓存所有从上层 SAP 发送过来的消息;queueToUpperSAP 用于缓存所有发往上层 SAP 的消息。

c) 示例。如图 2 所示, $U_SAP_1, U_SAP_2, \dots, U_SAP_m$ 是上层的服务访问点, $L_SAP_1, L_SAP_2, \dots, L_SAP_m$ 是下层的服务访问点。

功能实体的行为用两个函数描述,一个是 init()(),另一个是 behavior()()。在函数 init()()中,初始化实体 ID 和消息队列。函数 behavior()()描述实体的所有函数,这些函数包括接收、发送和处理实体的消息。

2.5 典型软硬件构件

典型的嵌入式系统构件包括软件和硬件部分,其中硬件部分主要有 CPU、定时器、缓存器等;软件部分主要是性能分析构件库。典型的硬件构件是处理器,其构件模型采用聚类实现,它包括两个进程类任务调度进程 CPU_SCH 和任务执行进程 CPU_Instruction_S,如图 3 所示。CPU_SCH 进程负责接收处理程序模块的任务请求,然后对每个任务按照调度顺序到 CPU_Instruction_S 模块中进行任务指令的执行,执行完毕后返回给任务的调度模块。

需要占用处理器资源的模块用通信信道与 CPU_S 相关

联。任务用数据类 CPUTaskINS 表达。当占用处理器资源的进程执行任务时,首先构造一个任务对象,指定其任务类型,然后通过消息发送给 CPU_S 进程。CPU_S 解析任务后,根据任务类型到 CPU_Instruction_S 中进行任务执行。任务执行过程采用仿真的方式,在 CPU_Instruction_S 中引入了一个指令处理的数据类 CPUInstructionProcessing,它具有数据字典功能,在初始化时会从一个文本文件中读取不同任务的指令周期数列表,当 PPC_Instruction_S 接收到一个特定类型的任务时,会根据任务的类型到字典中查找相应的指令周期数,然后计算这个任务所需要的 CPU 执行时间,并进行延迟操作。

在建模中只需要指定任务级指令执行时需要的指令周期及 CPU 的主频就可以应用 CPU 构件模型,将任务映射到 CPU 上执行,当 CPU 执行完成时,将任务完成信息返回给功能实体。这种处理方法可以应用于其他硬件资源的抽象建模和性能分析中。

软件构件模型主要包括 POOSL 性能分析库,包括常用概率分布类、置信区间类及长期样本均值类,常用概率分布类用于指定行为发生的概率,置信区间类用于计算长期样本均值类指标的置信区间,长期样本均值类计算各类长期样本均值类指标的估计值和精度。在实际应用中,只需要指定相应的参数就可以方便地使用它们。

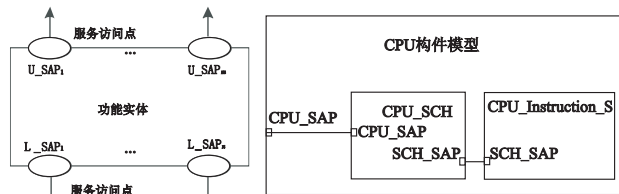


图2 功能实体建模模式

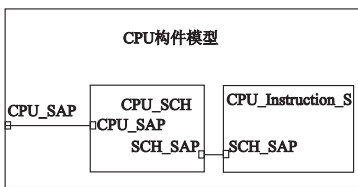


图3 CPU的构件模型

3 无线接入系统的建模及性能分析

随着无线局域网的广泛应用,公共热点服务不断延伸,接入设备不断普及使得无线局域网的应用迅速发展。与有线网络相比,无线网络具有信号传播距离有限和带宽资源非常有限等特点。因此,提高无线信道利用率的介质访问协议显得非常重要,对无线网络 MAC 层协议的性能进行分析和研究,是无线网络领域中的一个研究热点^[9,10]。无线接入点(access point, AP)是无线局域网中的关键设备,下面运用本文所提出的构件式建模方法为无线接入系统进行建模并进行性能分析。

本文的建模场景是 WLAN 网络环境。无线 AP 负责无线网络与有线网络的桥接,AP 与其通信范围内的所有无线站点(STA)构成一个基本服务集(basic service set, BSS)。STA 通过 AP 进行相互之间的通信,通过 AP 与有线网络进行通信。

3.1 建模模式的运用

对数据的抽象原则采用数据抽象模式,主要涉及到的数据单元有逻辑链路控制层协议的数据单元 LLC PDU、802.11 MAC 层服务数据单元 MSDU、802.11 MAC 层协议数据单元 MPDU、物理汇聚子层(PLCP)的数据单元 PPDU。DCF 机制中使用的 RTS、CTS 和 ACK 帧。

在这些数据单元的建模中,只抽取与模型行为相关的基本数据段,并增加了对相应字段进行修改和读取的函数。类 DOT11MSDU 继承于类 LLC PDU,因为逻辑链路层中的数据单元在 MAC 层中被叫做 MSDU,实际上是同一数据结构。为了

使类的名称有所分别,意义更明确,所以采用继承方式。实际上,类 DOT11MSDU 中没有增加新的属性和方法。类 DOT11DCFFrame 用来表示 RTS、CTS、ACK 三种数据帧,因为这三种帧在行为上比较相似,所以抽象为一个类,仅用 frameType 属性标志它们。

首先建立 IEEE 802.11 协议栈的模型,根据 MAC 层的模块划分^[10],将其抽象为一个聚类 IEEE802.11,它包含了 MAC_Data_Service、MPDU_Generation、protocol_Control、transmission、reception 和 MAC_Management_Service 等进程类,它们是 IEEE 802.11 MAC 协议的子系统(具体建模过程略)。在此基础上,建立了 AP 的抽象模型,如图 4 所示。

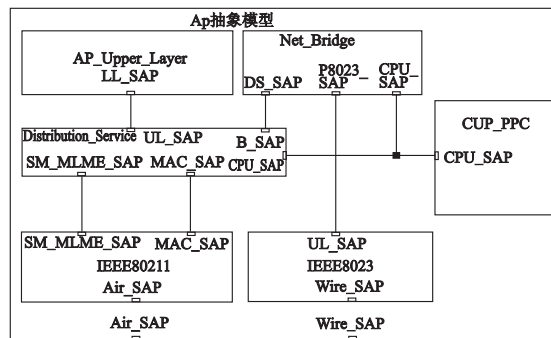


图4 AP的抽象模型

IEEE 802.11 模块是聚类 IEEE 802.11 的实例,IEEE 802.3 模块是 802.3 有线网络协议的进程类,根据建模需要对其进行了抽象,只对数据进行数据头的封装和收发。Distribution_Service 进程负责无线数据帧到有线数据帧的转换和数据的分发功能。AP_Upper_Layer 进程是 upper_Layer 的实例,它负责完成数据的发送和接收功能,是数据链路层以上网络层次的抽象。Net_Bridge 进程是网桥的抽象,负责有线网络和无线网络的桥接处理。CPU_PPC 模块应用了图 3 的处理器构件,需要占用 CPU 资源的各个功能实体通过映射到处理器模块上运行,以分析处理器性能。

在建模过程中,大量运用了信号原语抽象模式和异步并发事件抽象模式,如 protocol_Control 模块进行数据发送时,会给 transmission 模块发送 TxRequest 原语,发送完成后,会对 protocol_Control 发送 TxConfirm 原语进行确认;还有一些具有特殊意义的原语,如提供信道状态的原语 busy、idle、slot。当 reception 模块接收到物理层的数据帧需要对接收帧进行应答时,reception 会给 transmission 发送 needAck 原语。

对于功能实体的抽象比较典型的是 SAP 的抽象,SAP 在不同的网络层次中均有使用,采用了功能实体建模模式。以 MAC_Data_Service 进程为例进行说明:MAC_SAP 服务访问点负责上层逻辑链路层数据单元的接收与发送,MPDU_SAP 服务访问点负责向下层 MPDU_Generation 进程发送数据,P_SAP 服务访问点负责从 protocol_Control 进程接收数据。每个 SAP 均设置两个缓冲队列(接收队列和发送队列)。例如,MAC_SAP 服务访问点设置了 MSDUQueueFromMAC 和 MSDUQueueToMAC 两个队列,MSDUQueueFromMAC 用于接收数据,MSDUQueueToMAC 用于发送数据。相应地,每个 SAP 均有收发进程和处理进程,如 MAC_SAP 的收发进程 MAC_SAP_RS()负责数据的收发,只要端口上有数据就进行接收,然后放入接收队列 MSDUQueueFromMAC 中,发送队列 MSDUQueueToMAC 中只要有数据就进行发送。处理进程 MAC_

SAP_Processing() 对接收队列中的数据进行处理,并将处理后的数据发送给相应端口。

3.2 模型验证

根据模型仿真执行的过程中产生的动态交互图可以验证本模型是正确的,在已经建立好的行为模型的基础上加入数据统计类,将行为模型扩展为性能分析模型,以便将功能属性和非功能属性的评估区分开,进行性能指标的计算和精度判定,从而根据预先要求的精度及时自动中止模型的执行,并将分析的结果输出到文件中进行分析和处理。这些类包括 LRSampleAverage、LRRateAverage 和 LRTimeAverage^[8]等,可以在模型仿真执行的过程中计算吞吐量、平均延时和缓冲区占用率等长期样本均值类指标的的点估计值、区间估计值和置信度等统计结果,当仿真执行的精度达到要求时自动中止执行。

3.3 性能分析

采用 Rotalumis 工具对模型进行仿真执行,以分析网络中活跃站点数对有效吞吐量的影响,通过仿真测试分析,如果要求每个 STA 的有效吞吐量不小于 128 kbps,那么一个无线 AP 最多可以同时支持多少个站点呢?这对特定应用场景下协议的设计具有重要意义。

有效吞吐量随站点数变化的测试结果如图 5 所示,在不同的站点数和竞争窗口大小的情况下,对基本接入方式和 RTS/CTS 方式分别进行了测试。仿真执行中各参数的取值采用 IEEE802.11 标准的缺省值^[10],置信度为 90%。

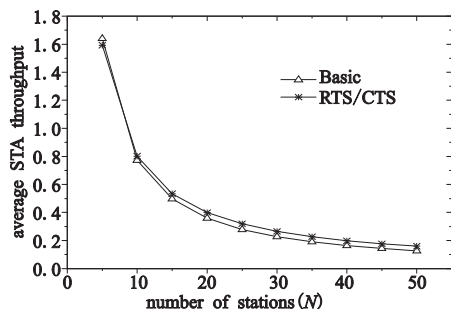


图5 有效吞吐量随站点数的变化曲线

由图可知,在保证每个 STA 的有效吞吐量不小于 128 kbps 的情况下,对于基本接入方式,AP 最多可以支持大约 40 个站点,对于 RTS/CTS 方式,最多可以支持大约 50 个站点。

另外,还仿真测试了 AP 使用不同频率的处理器器的情况下,处理器利用率和 AP 处理延时情况。

4 结束语

本文提出了构件式建模的方法,即将一些通用的系统模块制作成构件,在建模过程中只需要设置一下构件的参数即可重用,并建立了一些典型的嵌入式系统构件的模型。针对数据抽象、信号原语抽象、异步并发事件的处理、通信机制的实现以及行为模型扩展为性能分析模型等建模过程中反复出现的问题,归纳出了通用的建模模式,以指导实际的建模过程。以无线接入系统为例,建立了系统级抽象模型,分析了有效吞吐量随站点数变化情况。结果表明,本文所提构件式建模方法能够有效应用于复杂嵌入式系统的系统级建模。

参考文献:

- [1] KOPETZ H. Real-time systems: design principles for distributed embedded applications[M]. 2nd ed. New York: Springer, 2011.
- [2] ADLER R, SCHAEFER I, TRAPP M. Component-based modeling and verification of dynamic adaptation in safety-critical embedded systems[J]. ACM Trans on Embedded Computing Systems, 2010, 10(2): 1-39.
- [3] BOUASSIDA N, BEN-ABDALLAH H. Extending UML to guide design pattern reuse[C]//Proc of IEEE International Conference on Computer Systems and Applications. New York: IEEE Press, 2006: 1131-1138.
- [4] GRIES M. Methods for evaluation and covering the design space during early design development[J]. Integration, 2004, 38(2): 131-183.
- [5] THEELEN B D, VOETEN J P M, KRAMER R D J. Performance modelling of a network processor using POOSL[J]. Journal of Computer Networks, 2003, 41(5): 667-684.
- [6] HUANG Zhang-qin, VOETEN J P M, THEELEN B D. Modelling and simulation of a packet switch system using POOSL[C]//Proc of PROGRESS. Utrecht: STW Press, 2002: 18-27.
- [7] FANG Lin-bo, WU Hong-xia, HUANG Zhang-qin. Performance modeling and analysis of IEEE 802.11 protocol using POOSL[C]//Proc of the 6th IEEE Conference on Industrial Electronics and Applications. New York: IEEE Press, 2009: 1330-1335.
- [8] 方林波,黄樟钦,侯义斌,等. POOSL 的扩展及其在性能分析中的应用研究[J]. 系统仿真学报, 2007, 19(13): 2908-2912.
- [9] 毛建兵,毛玉明,冷鹏,等. 一种利用信道侦听的 IEEE802.11 自适应优化算法[J]. 软件学报, 2010, 21(8): 1968-1981.
- [10] 高峰. IEEE 802.11 无线网络 DCF 协议性能分析及提升研究[D]. 北京: 北京邮电大学, 2010.