

# 基于模型的嵌入式系统性能分析框架\*

朱雪阳



(计算机科学国家重点实验室(中国科学院 软件研究所),北京 100190)

通讯作者: 朱雪阳, E-mail: zxy@ios.ac.cn

**摘要:** 在现代嵌入式系统中,性能的重要性日益凸显.传统的基于测量的性能分析方法在运行时对性能进行测试,往往在代码实现后才考虑实施.若此时发现的问题是体系结构或设计因素造成的,修复的代价将非常昂贵.提出了一种基于形式化模型的性能分析(formal method-based performance analysis,简称 FMPA)方法框架,希望在系统开发的早期,对系统设计模型进行性能分析,以便尽早发现并解决潜在的性能问题.FMPA 具有统一的对外接口(UML-MARTE),基于多种形式化模型,并可对多种系统性能指标进行分析.该方法适用于基于模型的开发过程,可为实时嵌入式系统的设计开发提供多项性能指标的参考.通过介绍利用实时模型检测技术分析响应时间与吞吐量,利用概率模型检测技术分析系统可靠性,以及 FMPA 方法的支撑工具 FMPAer 的总体设计方案,说明了 FMPA 方法框架的可行性.

**关键词:** UML; MARTE; 模型检测; 响应时间; 吞吐量; 系统可靠性

中文引用格式: 朱雪阳.基于模型的嵌入式系统性能分析框架.软件学报,2016,27(Suppl.(2)):328-335. <http://www.jos.org.cn/1000-9825/16047.htm>

英文引用格式: Zhu XY. Model-Based performance analysis framework for embedded systems. Ruan Jian Xue Bao/Journal of Software, 2016, 27(Suppl. (2)): 328-335 (in Chinese). <http://www.jos.org.cn/1000-9825/16047.htm>

## Model-Based Performance Analysis Framework for Embedded Systems

ZHU Xue-Yang

(State Key Laboratory of Computer Science (Institute of Software, The Chinese Academy of Sciences), Beijing 100190, China)

**Abstract:** System performance becomes more and more important in modern embedded systems. Traditionally, system performance is measured after the system has been implemented. When it fails to meet the requirement due to the design of the application at this stage, the cost of fixing them would be high. This paper presents a framework of formal method-based performance analysis (FMPA), whose goal is to detect potential performance problems at the early stage of the model-based procedure of system development. FMPA is aimed to analyse multiple performance criteria. It provides unified input models (UML-MARTE) and various formal models as analysis models. The feasibility of FMPA is illustrated by throughput and response time analysis with real-time model checking, by system reliability prediction using probabilistic model checking, and is further confirmed by the implementation of its support tool FMPAer.

**Key words:** UML; MARTE; model checking; response time; throughput; system reliability

随着计算机系统的广泛使用,软件性能问题得到越来越多的关注;特别是安全攸关的嵌入式软件,其性能更是关键.在很多情况下,仅满足功能需求而无法满足不同性能需求的软件已不能算是正确的.本文将系统的非功能属性统称为性能,如时间、资源使用、系统可靠性等.

传统的基于测量的性能分析方法在运行时对性能进行测试,往往在代码实现后才考虑实施.实践经验表明,有些性能问题既不能通过代码的性能调优加以解决,也不能通过增加硬件得到改善.这是由于这些问题是体系

\* 基金项目: 国家自然科学基金(61572478, 61472406, 61472474)

Foundation item: National Natural Science Foundation of China (61572478, 61472406, 61472474)

收稿时间: 2016-09-20; 采用时间: 2016-11-17

结构或设计因素造成的,而不是低效的编码引起的.这样的性能问题在开发过程的早期已经被引入,到集成测试或更晚时才予以考虑,但为时已晚.修复这类问题必须回溯到系统的早期设计,势必增加开发成本及周期.因此,有必要研究在系统开发的早期利用模型对性能进行分析的方法,以便尽早发现并解决潜在的性能问题,提高开发效率并降低成本.

构建系统性能模型,将影响性能的各种因素纳入模型描述中,再通过对性能模型进行分析和预测,评估系统性能是否符合要求,对系统的设计、评价和改进都有十分重要的作用.自软件性能工程<sup>[1]</sup>提出以来,基于模型的系统性能分析方法得到了蓬勃发展<sup>[2,3]</sup>.传统的基于模型的性能分析方法用排队网络(queueing network)<sup>[4]</sup>为系统性能建模,以马尔可夫随机过程为数学工具.但排队模型描述能力还有不足,也难以集成到软件开发过程中<sup>[2]</sup>.

统一建模语言 UML 作为面向对象建模语言,因其建模的灵活性和实用性以及可对系统的多个侧面进行描述的特点,在软件开发中得到广泛应用,并已成为事实上的工业标准.为了对实时嵌入式系统进行建模与分析,国际对象组织(OMG)提出的关于实时嵌入式系统建模与分析(modeling and analysis of real-time embedded system,简称 MARTE)规范<sup>[5,6]</sup>,正是希望通过统一建模语言来集成不同的分析方法.但 UML-MARTE 缺乏精确的语义,难以直接对 UML-MARTE 模型进行性能分析.将 UML-MARTE 模型转换为不同的分析模型,再利用相应模型进行性能分析的研究也正成为热点<sup>[6-8]</sup>.

形式化方法是基于数学手段为软件和硬件系统提供精确规约、开发和验证的方法.严格的形式化模型可以弥补 UML-MARTE 在这方面的不足.在利用形式化方法对系统进行性能分析时,需要软件设计人员构造出系统的形式化模型.掌握形式化方法一般要求具有相关的数学知识背景,需要较长的学习时间的投入,这又在一定程度上妨碍了形式化方法在实际开发中的应用.

如何结合 UML-MARTE 模型与形式化方法的研究成果,形成优势互补,结合二者之长以对系统性能进行分析是一个重要且复杂的科学问题.本文提出的基于形式化模型的性能分析方法框架(formal model-based performance analysis,简称 FMPA),正是希望通过部分地解决这一问题,将形式化方法应用于软件性能分析,为实时嵌入式系统的开发提供多项性能指标评估作为参考.相关支撑工具(formal model based performance analyzer,简称 FMPAer)<sup>[9]</sup>具有统一的对外接口(UML-MARTE),基于多种形式化模型,并可对多种系统性能指标进行分析.FMPA 方法及其支撑工具 FMPAer 都提供了便于扩充的机制,以容纳更多的分析方法来分析更多不同的性能指标.

本文第 1 节概述 FMPA 的整体方法框架.第 2 节对 FMPA 方法目前使用的 UML-MARTE 模型子集进行简单介绍.第 3 节和第 4 节分别以利用实时模型检测技术分析响应时间与吞吐量和利用概率模型检测技术分析系统可靠性为例来说明 FMPA 方法框架的具体实施路径.第 5 节简述 FMPA 框架支撑工具 FMPAer 的总体设计方案.最后总结全文.

## 1 FMPA 方法框架

FMPA 方法框架具有统一的对外接口,基于多种形式化模型,并可对多种系统性能指标进行分析.对于不同性能指标和不同系统的特点,各种形式化方法在进行性能分析时各有优劣,利用多种形式化方法对系统进行分析可能得到更多性能指标的分析结果,为系统开发提供更有力的支持.FMPA 方法总体研究框架如图 1 所示.

为了兼顾工程上使用的方便性及分析所需的严格性,FMPA 建模语言分抽象与具体两个层次:抽象层提供一致的对外接口;具体层为形式化分析提供基础.为了方便工程师使用,抽象层采用工业界的事实标准 UML-MARTE 中的性能模型;具体层为针对不同性能指标、不同系统特点,采用不同的形式化语言,包括时间自动机、马尔柯夫决策过程及约束函数等.UML-MARTE 用状态机、活动图、顺序图等行为图描述系统行为,以相关性能分析参数作为注释,形成性能模型.针对不同的性能指标,我们研究如何将这些行为图再度抽象为具体的形式化语言,只保留性能分析所需元素以提高分析效率.

双层建模语言不仅保证了对外接口的一致性及分析方法的多样性,而且有足够的开放性以便于新的分析模型加入.当加入新模型时,只要建立起 UML-MARTE 性能模型到该模型的语义映射,即可将新的分析方法纳入我们的统一分析框架中.

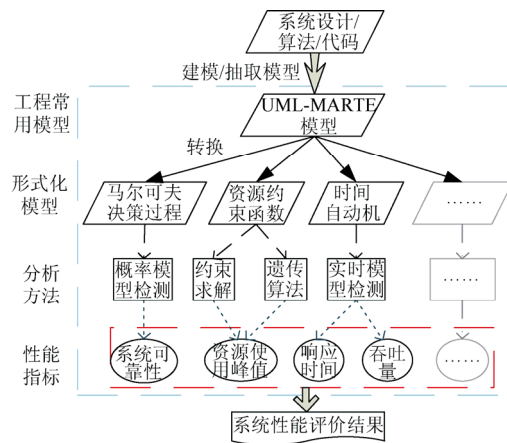


Fig.1 Framework of FMPA

图1 FMPA 总体研究框架

在为系统的性能建模时,将可能影响相关性能指标的平台特性抽象为参数,以便后继的分析可独立于系统所使用的平台.在具体的建模过程中重点考虑两个问题.一个是选择合适的模型粒度问题,一方面粒度不能太小以防状态空间过大,造成分析的复杂性过大;另一方面粒度又不能太大以防过度抽象,导致分析结果的精确度不够.另一个是参数的获取问题.这两个问题的合理解决,是本文方法得以正确应用的重要前提,但不属于本文讨论范围.我们将在对本文方法的进一步研究和应用过程中寻找对这两个问题的较好解决方案.

FMPA 方法的另一好处是,可以通过调整前端模型中的某些系统参数,如增加硬件资源或系统并发数等来得到不同的后端分析结果,以便我们观察不同配置下系统性能的变化,从而得出哪些系统参数的变化对系统性能的影响最大,进而更好地指导系统的设计.

我们将在第3节和第4节以利用实时模型检测技术分析响应时间与吞吐量<sup>[10]</sup>和利用概率模型检测技术分析系统可靠性<sup>[11]</sup>为例来说明 FMPA 方法框架的具体实施路径.

## 2 UML-MARTE 模型简介

MARTE 主要包括 3 个部分:MARTE 基础包(MARTE foundation)、MARTE 设计模型包(MARTE design model)和 MARTE 分析模型包(MARTE analysis model).其中,MARTE 基础包包括系统建模中的基本元素,如时间和资源等,是 MARTE 建模的最核心部分;MARTE 设计模型包用来对系统的细节进行建模和设计;MARTE 分析模型包用于对系统的调度和时间等一些非功能属性进行建模.

由于嵌入式系统的应用场景多种多样,不同场景涉及到的系统元素也不尽相同,很难用一种通用的建模方式来描述所有嵌入式系统,FMPA 目前先考虑一种较常见也较实用的应用场景:涉及多用户和多资源的并发实时系统.在建模过程中,我们选用 UML 的用例图(UseCase diagram,简称 UCD)来描述系统中的各类用户及系统功能,以及用户与系统功能模块(关联到具体的活动图)间的关联关系;选用部署图(deployment diagram,简称 DD)声明系统中的硬件资源;选用适合描述系统动态特性的活动图(activity diagram,简称 AD)来描述系统的动态行为.活动图是用例图中功能模块的具体执行过程描述.本文涉及的主要 MARTE 模型组件及其关系如图 2 所示.

MARTE 以构造型(stereotype)的形式绑定到 UML 模型上,是 UML 模型的一个扩充.我们可以通过在 UML 模型图的各个元素上添加构造型,然后设置构造型中标签值的方法实现 UML-MARTE 建模.如图 2 所示,《GaAnalysisContext》是加在整个模型之上的 MARTE 构造型,其标签值“concurrent”表示系统允许的最大并发数,即同一时刻系统最多能够同时处理多少个用户请求.图 3 所示模型是在各 UML 模型图及其元素上加上 MARTE 规范以实现系统的 UML-MARTE 建模的示例.

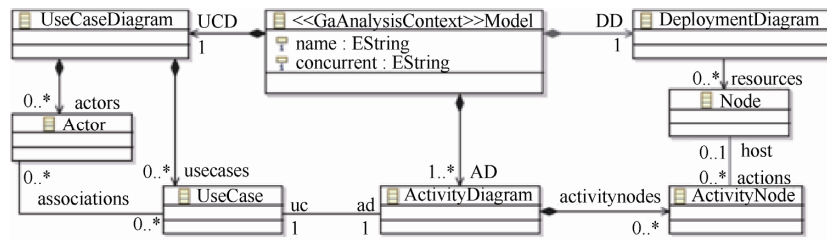


Fig.2 Components of a MARTE model in this paper

图 2 本文涉及的主要 MARTE 模型组件

下面结合图 3 所示的对某种脉冲血氧测量仪(pulse oximeter)的激发过程的 UML-MARTE 模型,简单介绍各模型图.

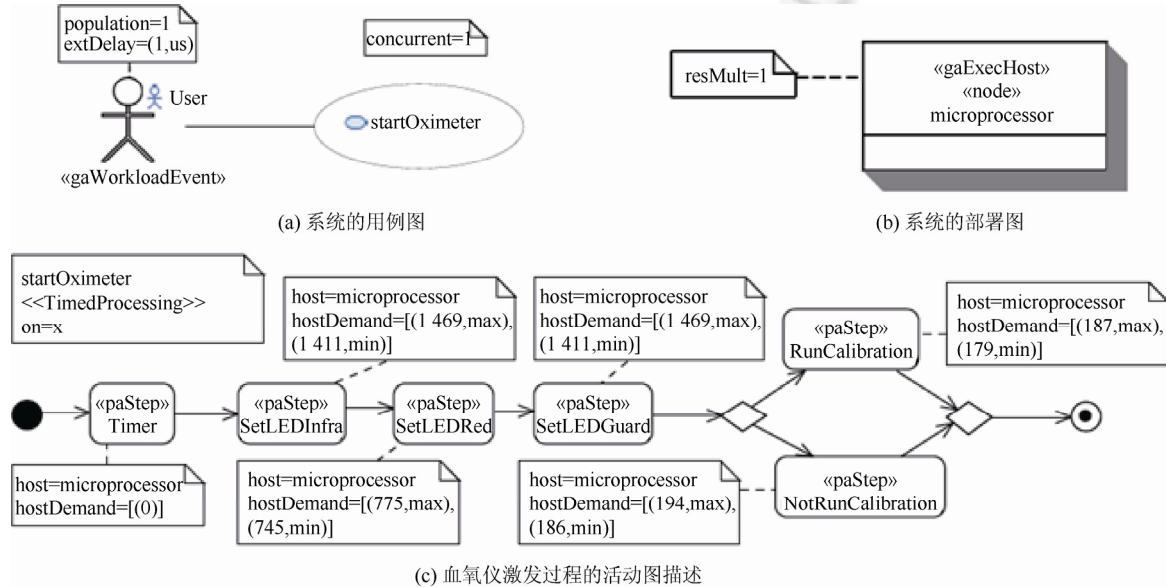


Fig.3 A UML-MARTE model for the starting procedure of a pulse oximeter<sup>[10]</sup>

图 3 脉冲血氧测量仪的激发过程的 UML-MARTE 模型<sup>[10]</sup>

用例图由用户和用例以及它们之间的关联关系组成,如图 3(a)所示.用户是系统的外部参与者,用人形图形表示;用例代表系统的各个功能模块(关联到具体的活动图),用椭圆表示;二者之间的关联关系规定了每类用户可以执行哪些用例,用一条连接用户和用例的线段表示.

部署图由一系列的资源结点组成,描述了系统所拥有的硬件资源以及它们的属性.资源节点用长方体表示.声明资源后,就可以将活动图中的动作结点关联到不同的资源结点.某个动作若要执行,其关联到的资源的数量就必须充足,否则需要等待其他动作释放资源.图 3(b)为脉冲血氧测量仪的部署图.在这个简单的例子中,系统有一个资源“microprocessor”;其标签值“result”为 1,说明该资源个数为 1.

活动图与用例图中的某个用例相关联,是该用例的具体行为描述.活动图是一种特殊形式的状态机,描述了一组顺序或并发的活动.各活动结点通过<<PaStep>>构造型的“host”标签值绑定到相应的资源上.<<PaStep>>是 MARTE 分析模型包中的一个构造型,用于对模型图中的元素(结点、边等)加上性能的约束.如图 3(c)所示,可在活动图的动作结点 SetLEDInfra 上加入 MARTE 的<<PaStep>>构造型,并设定其“host”和“hostDemand”的标签值.“host”标签值指定动作 SetLEDInfra 在资源“microprocessor”上执行;“hostDemand”标签值说明动作的执行时间在区间[1411,1469]内.

### 3 实时模型检测技术分析响应时间与吞吐量

对基于多核/多处理器平台的并发实时系统而言,吞吐量和系统响应时间是两个非常重要的性能指标.吞吐量是指单位时间内系统可以处理的用户请求的数目,它是衡量系统处理能力的重要性能参数.系统响应时间是对每个用户而言,从请求到达系统到被处理完毕的所用时间.系统响应时间直接关系到用户的使用体验,是并发实时系统的另一个重要性能指标.

针对这两个性能指标,实时系统开发人员关心的问题是:一个设计模型能否满足系统的吞吐量需求( $thr$ ),和/或响应时间需求( $rt$ ).我们除了能够回答这两个问题本身外,还可以更进一步地找出模型可能满足的吞吐量值和响应时间值的范围.

吞吐量分析的具体方法是:找到值  $TP_{min}$ ,使得当  $thr \leq TP_{min}$  时,无论如何调度,模型的吞吐量都能达到  $thr$ (即使在最坏情况下,吞吐量需求都能够得到满足);找到值  $TP_{max}$ ,使得当  $thr > TP_{max}$  时,无论如何调度,模型的吞吐量都无法达到  $thr$ (即使在最好情况下,吞吐量需求都无法得到满足);当  $thr \in (TP_{min}, TP_{max})$  时,总能找到一个调度使得模型的吞吐量达到  $thr$ ,但也存在不满足的调度.模型吞吐量的可达范围如图 4(a)所示.

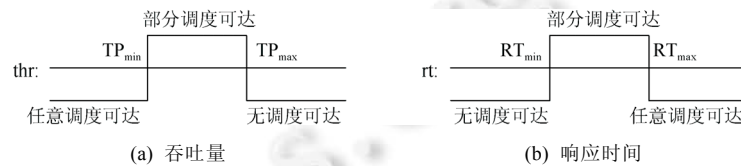


Fig.4 Ranges of achievable throughput and response time

图 4 吞吐量与响应时间的可达范围

响应时间的分析略有不同:找到值  $RT_{min}$ ,使得当  $rt < RT_{min}$  时,无论如何调度,模型的响应时间都不能达到  $rt$ (相对于当前系统设计,要求的响应时间太短,即使在最好情况下,都无法得到满足);找到值  $RT_{max}$ ,使得当  $rt \geq RT_{max}$  时,无论如何调度,模型的响应时间都在  $rt$  之内(相对于当前系统设计,要求的响应时间足够长,即使在最坏情况下,也能满足要求);当  $rt \in (RT_{min}, RT_{max})$  时,总能找到一个调度使得模型的响应时间不超过  $rt$ ,但也存在超时的调度.模型响应时间的可达范围如图 4(b)所示.

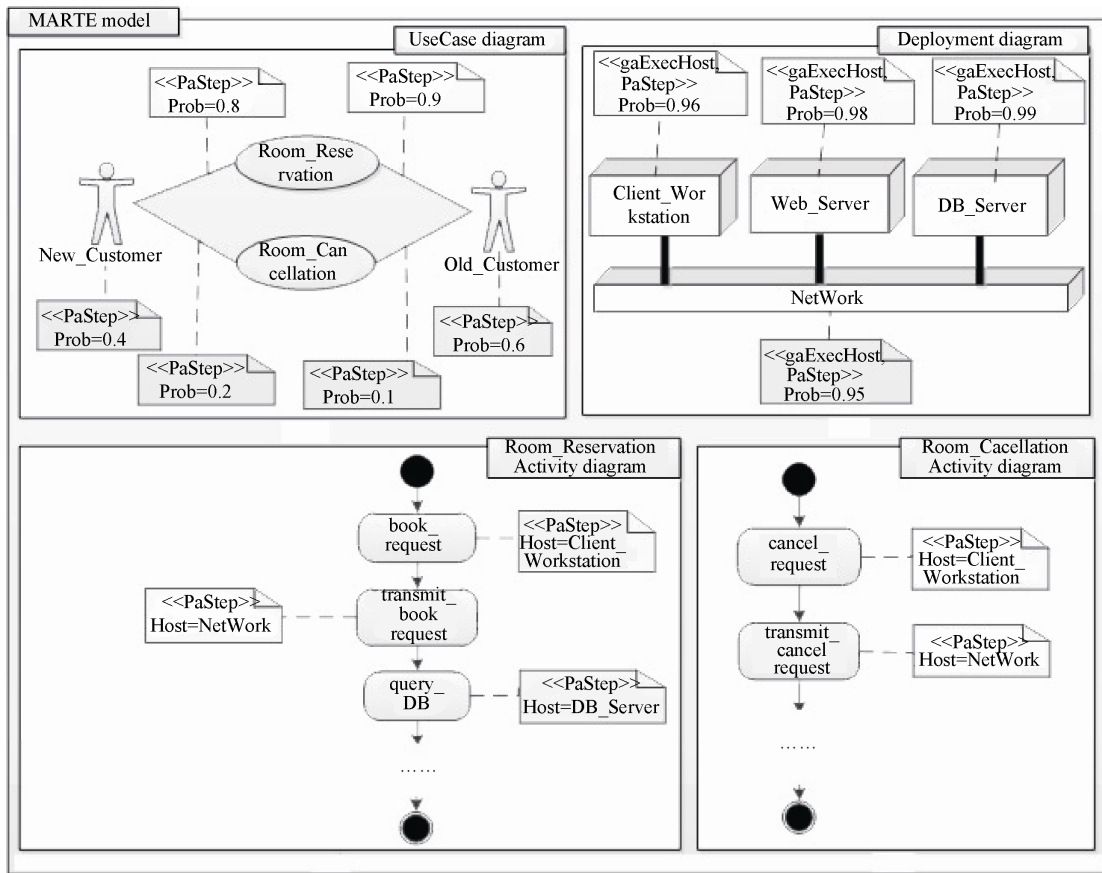
我们通过将 UML-MARTE 模型映射为时间自动机模型<sup>[12]</sup>,将分析目标形式化为分支时序逻辑(CTL)<sup>[13]</sup>公式,再利用实时模型检测工具 UPPAAL<sup>[14]</sup>对转换所得的时间自动机模型进行相关性质进行验证,就实现了对 UML-MARTE 模型的系统吞吐量及响应时间的评估.具体的转换方法及性质公式见文献[10].

### 4 概率模型检测技术分析系统可靠性

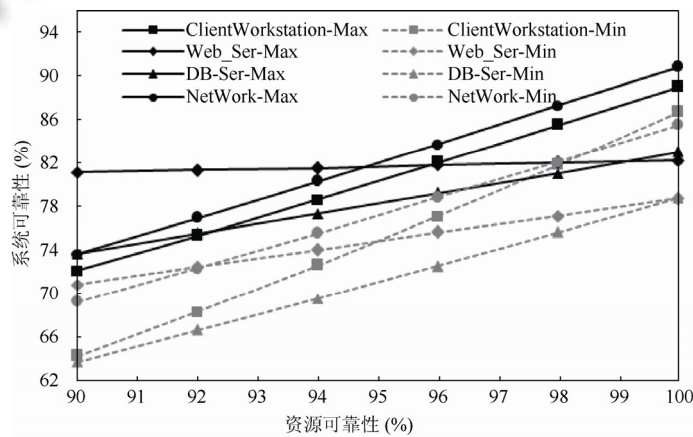
系统可靠性是在规定的条件和时间内,系统无故障运行的概率<sup>[15]</sup>.它是系统的重要非功能属性之一.对于具有  $n$  个角色、 $m$  个用例的 UML-MARTE 模型,其系统可靠性定义<sup>[16]</sup>为:每个用例被执行的概率乘以该用例所对应的活动图无故障运行的概率,再求和.当活动图包含条件选择结点时,选择不同的分支执行得到的系统可靠性值不同.我们通过计算系统可靠性的最大值和最小值来评估系统的可靠性范围.

为了评估系统可靠性,需要在 UML-MARTE 模型中添加相应的概率参数.在用例图中,通过在每个角色的  $\langle\langle PaStep \rangle\rangle$  构造型的“prob”标签值来赋予这个角色使用系统的概率;通过角色和用例之间的关联的  $\langle\langle PaStep \rangle\rangle$  构造型的“prob”标签值来赋予角色执行该用例的概率.用例的具体功能由相应的活动图实现.当用例图中有  $n$  个角色, $m$  个用例时,满足:  $\sum_{x=1}^n q_x = 1$ , 其中  $q_x$  是以第  $x$  个角色使用系统的概率.对于第  $x$  个角色,需要满足  $\sum_{y=1}^m p_{xy} = 1$ , 其中  $p_{xy}$  表示第  $x$  个角色执行第  $y$  个用例的概率.在部署图中,通过资源结点的  $\langle\langle PaStep \rangle\rangle$  构造型的“prob”标签值来描述资源的可靠性,即资源无故障执行的概率.活动图中的每个活动结点通过  $\langle\langle PaStep \rangle\rangle$  构造型的“host”标签值将其部署到部署图的资源上,表示活动由其相应的资源来执行;该活动的可靠性由所绑定的资源可靠性定义.一个带概率参数的酒店预订系统模型如图 5(a)所示.





(a) 模型



(b) 分析结果

Fig.5 An example for the results of modeling and reliability analysis of a system<sup>[13]</sup>

图5 系统可靠性建模及分析结果示例<sup>[13]</sup>

在图 5(a)所示的资源 *ClientWorkstation*, *Web\_Server*, *DB\_Server* 和 *NetWork* 的可靠性分别为 96%, 98%, 99% 和 95% 的情况下, 我们可预测系统可靠性的最大值和最小值分别为 82% 和 77%。我们的方法除了可以在给定资源可靠性情况下对系统进行分析外, 还可以通过调整资源可靠性值来观察某个资源可靠程度对系统的影响。例如, 将图

5(a)所示的模型中各资源的可靠性值从 90%逐一变化到 100%(当一个资源值变化时,其他资源的值不变),系统可靠性变化如图 5(b)所示.其中具有不同形状标记的实线为最大值,虚线为最小值.从图 5(b)中我们也可以看到,资源 *Web\_Server* 的可靠性变化对系统可靠性最大值和最小值的影响最小,资源 *NetWork* 和 *ClientWorkstation* 的可靠性变化对系统可靠性最大值的影响较大,资源 *ClientWorkstation* 的可靠性变化对系统可靠性最小值的影响较大.

我们通过将 MARTE 模型转换成马尔可夫决策过程(Markov decision process,简称 MDP)<sup>[17]</sup>模型,再借助概率模型检测工具 PRISM<sup>[18]</sup>来分析系统可靠性的最大值和最小值.具体的转换方法及性质公式见文献[11].

## 5 FMPAer 工具总体设计方案

我们基于 Eclipse/Java 实现了上述 FMPA 方法,并部分集成到 FMPAer 工具<sup>[9]</sup>中.FMPAer 包括图形建模工具、语言转换器及形式化分析工具;结合自主开发与开源工具,实现了从 UML-MARTE 建模、模型转换、底层形式化方法性能分析、到返回分析结果的全过程.具体包括:将 UML-MARTE 模型转换成各种形式化模型的语义转换,语义模型到各具体底层形式化分析工具语言的语法转换,相关性能指标自动转换为底层形式化分析工具的性质描述,最后将底层工具分析结果返回前端.FMPAer 总体设计框架如图 6 所示.

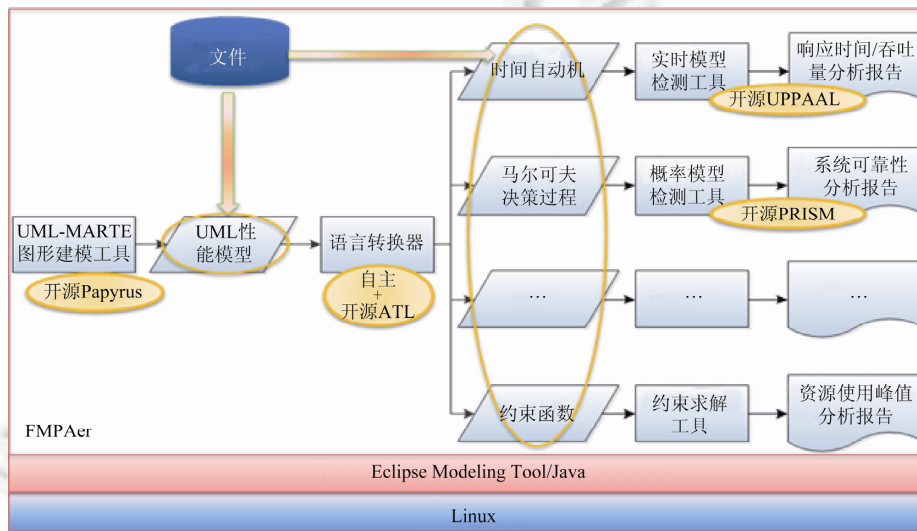


Fig.6 Architecture of FMPAer

图 6 FMPAer 总体设计框架

FMPAer 的图形建模工具,主要采用开源软件 Papyrus<sup>[19]</sup>,为用户提供 UML-MARTE 模型的图形化编辑环境. FMPAer 的语言转换器包括从 UML-MARTE 到各形式化语言转换的多个工具,采用语义和语法两层分离的转换.在语义转换层,采用元模型转换技术,根据基于 ATL 模型转换语言<sup>[20]</sup>、必要时适当扩充的方法,将 UML-MARTE 元模型转换为各种形式化模型的元模型;在语法转换层,将以 FMPAer 格式存储的各种形式化模型文件转换为底层相应形式化分析工具的入口文件.底层形式化工具目前包括开源实时模型检测工具 UPPAAL<sup>[14]</sup>和开源概率模型检测工具 PRISM<sup>[18]</sup>.随着研究的深入,我们还将调整并吸收进更多底层形式化工具以增强 FMPAer 的功能及效率.例如,我们已经实现但限于篇幅未在本文介绍的系统资源峰值分析工具则是利用约束求解工具 Z3Opt<sup>[21]</sup>进行分析.

## 6 总结

本文提出了基于形式化模型的性能分析方法框架 FMPA,希望在系统开发的早期,利用模型对性能进行分析,以便尽早发现并解决潜在的性能问题.FMPA 方法具有统一对外接口(UML-MARTE),基于多种形式化模型,

可对多种系统性能指标进行分析.它适用于基于模型的开发过程,旨在为实时嵌入式系统的设计开发提供多项性能指标的参考.我们已经实现了两种方法:利用实时模型检测技术分析响应时间与吞吐量,利用概率模型检测技术分析系统可靠性,并且实现了 FMPA 方法的支撑工具 FMPAer,这充分说明了 FMPA 方法框架的可行性.

#### References:

- [1] Smith CU, Williams LG. Performance engineering of software systems. In: Lavagno L, Martin G, Selic B, eds. UML for Real: Design of Embedded Real-Time Systems. Dordrecht: Kluwer Academic Publishers, 2003. 343–365.
- [2] Balsamo S, Marco AD, Inverardi P, Simeoni M. Model-Based performance prediction in software development: A survey. IEEE Trans. on Software Engineering, 2004,30(5):295–310.
- [3] Becker S, Koziolok H, Reussner R. Model-Based performance prediction with the palladio component model. In: Proc. of the 6th Int'l Workshop on Software and Performance. ACM, 2007. 54–65.
- [4] Balsamo S, Personè VDN, Inverardi P. A review on queueing network models with finite capacity queues for software architectures performance prediction. Performance Evaluation, 2003,51(2–4):269–288.
- [5] UML Profile for MARTE: Modeling and analysis of real-time embedded systems. OMG Adopted Specification ptc/2010-08-33, ptc/2010-08-34.
- [6] Selic B, Gerard S. Modeling and Analysis of Real-Time and Embedded Systems with UML and MARTE: Developing Cyber-Physical Systems. New York: Elsevier, 2013.
- [7] Lopez-Grao JP, Merseguer J, Campos J. From UML activity diagrams to stochastic Petri nets: Application to software performance engineering. In: Proc. of the 4th Int'l Workshop on Software and Performance (WOSP 2004). 2004. 25–36.
- [8] Petriu DC, Woodside CM. Performance analysis with UML. In: Lavagno L, Martin G, Selic B, eds. UML for Real: Design of Embedded Real-Time Systems. Dordrecht: Kluwer Academic Publishers, 2003. 241–270.
- [9] FMPAer. <http://lcs.ios.ac.cn/~zxy/tools/fmpaer.htm>
- [10] Yan G, Zhu XY, Yan R, Li G. Formal throughput and response time analysis of MARTE models. In: Proc. of the 16th Int'l Conf. on Formal Engineering Methods (ICFEM 2014). LNCS 8829, 2014. 430–445.
- [11] Chai YS, Zhu XY, Yan RJ, Zhang GQ. MARTE models based system reliability prediction. Computer Science, 2015,42(12):82–86,91 (in Chinese with English abstract).
- [12] Alur R, Dill DL. A theory of timed automata. Theoretical Computer Science, 1994,126(2):183–235.
- [13] Clarke EM, Emerson EA, Sistla AP. Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Trans. on Programming Languages and Systems, 1986,8(2):244–263.
- [14] Larsen KG, Pettersson P, Wang Y. UPPAAL in a nutshell. Int'l Journal on Software Tools for Technology Transfer, 1997,1(1): 134–152.
- [15] Gokhale SS, Trivedi KS. Analytical models for architecture-based software reliability prediction: A unification framework. IEEE Trans. on Reliability, 2006,55(4):578–590
- [16] Cortellessa V, Singh H, Cukic B. Early reliability assessment of UML based software models. In: Proc. of the 3rd Int'l Workshop on Software and Performance. 2002. 302–309.
- [17] Forejt V, Kwiatkowska M, Norman G, Parker D. Automated verification techniques for probabilistic systems. In: Proc. of the 11th Int'l School on Formal Methods for the Design of Computer, Communication and Software Systems. Springer-Verlag, 2011. 53–113.
- [18] Kwiatkowska M, Norman G, Parker D. PRISM 4.0: Verification of probabilistic realtime systems. In: Proc. of the 23rd Int'l Conf. on Computer Aided Verification. Berlin: Springer-Verlag, 2011. 585–591.
- [19] <http://www.eclipse.org/papyrus/>
- [20] <http://www.eclipse.org/at/>
- [21] Z3Opt. <http://rise4fun.com/z3opt/tutorialcontent/guide>

#### 附中文参考文献:

- [11] 柴叶生,朱雪阳,晏荣杰,张广泉.基于 MARTE 模型的系统可靠性预测.计算机科学,2015,42(12):82–86,91.



朱雪阳(1971—),女,福建莆田人,博士,副研究员,CCF 专业会员,主要研究领域为嵌入式系统设计,性能分析与优化及形式化方法.