

面向 CPS 系统仿真的建模方法研究与设计

刘 厦 王宇英 周兴社 张 凡
(西北工业大学计算机学院 西安 710129)

摘 要 CPS 仿真建模在构建 CPS 系统时具有重要意义,不仅能够对系统进行验证和测试,而且是模型驱动开发中的重要环节。CPS 系统通过精化和提炼可分为物理实体和计算实体,采用具有动态连续性、基于时间的运动状态行为模型构建物理实体;计算实体模型则采用基于有限状态机的离散系统行为模型进行构建。通过面向时间的状态精化对两种行为模型进行扩展,将两种扩展模型进行融合,完成 CPS 系统的建模仿真。分析了构建计算实体模型的统一建模语言(UML)以及构建物理实体模型的 simulink/RTW 建模工具的可行性,提出了基于 UML 框架的两种异质模型融合方法。

关键词 CPS 仿真建模,物理实体,计算实体,行为模型,面向时间的状态精化

中图分类号 TP391.9 **文献标识码** A

Research and Design for the Modeling of Simulation of CPS

LIU Sha WANG Yu-ying ZHOU Xing-she ZHANG Fan
(College of Computer, Northwestern Polytechnic University, Xi'an 710129, China)

Abstract The simulation and modeling of CPS are of great significance in the development of CPS-System, not only can help testing and validating in the process of building system, but also are the Model-driven development's important part. Cyber physical system can be divided into physical entity and computation entity by means of essence and refining. Physical entity can be constructed by dynamic continuous-and time-based motion-state of behavioral model and using finite state-machine of behavioral model to carry on construct computation entity. This paper extended these two behavioral models through time oriented state-refinement, and used the extended models to complete the simulation and modeling of CPS, besides, discussed the ability of using Unified Model Language to build computation entity and using Simulink/RTW tools to construct physical entity, and proposed the approach for integration of heterogeneous models at the framework of UML.

Keywords Simulation and modeling of CPS, Physical entity, Computation entity, Behavioral model, Time oriented state-refinement

1 引言

CPS(Cyber-Physical-System)最早由美国国家委员会提出,被认为有望成为继计算机、互联网之后世界信息技术的第三次浪潮,其核心是 3C(Computation, Communication, Control)技术融合^[1]。从广义上理解,CPS 就是一个在环境感知的基础上,深度融合了计算、通信和控制能力的可控、可信、可扩展的网络化物理设备系统。它通过计算进程和物理进程相互影响的反馈循环实现深度融合,并通过实时交互来增加或扩展新的功能,以安全、可靠、高效和实时的方式监测或者控制一个物理实体。CPS 的最终目标是实现信息世界和物理世界的完全融合,构建一个可控、可信、可扩展并且安全高效的 CPS 网络,并最终从根本上改变人类构建工程物理系统的方式。与传统的嵌入式系统相比,CPS 系统具有系统规模庞

大、系统异质性更加明显、控制更精确和智能化等特点。

CPS 系统根据功能和所属领域的不同,可以划分为 3 部分:物理实体、计算实体和交互实体(见图 1)。所谓物理实体就是信息物理融合系统运行所处的物理环境,也就是我们所处的真实世界,是 CPS 系统中的执行者和被控对象。所谓计算实体就是指在信息物理融合系统中对物理实体进行控制、调节的那部分,包括计算单元和控制单元。交互实体在物理实体和计算实体间传递信息,实现物理实体和计算实体的融合,是现实和虚拟之间的桥梁,包括信息采集单元、数据转换以及数据传输单元。在一个 CPS 系统中,这三者之间的划分是极其模糊的。比如说,一个复杂系统往往是由多个 CPS 节点组成,某个节点对于整个系统而言可能是物理实体或是信息实体,但是这个 CPS 节点却由物理实体部分和信息实体部分构成。

到稿日期:2011-08-30 返修日期:2011-11-24 本文受国家 863 计划主题项目(2011AA010101)和西安市科技计划项目(CXY1008)资助。

刘 厦(1986—),男,硕士生,主要研究方向为 CPS 系统仿真建模框架和嵌入式系统实时仿真,E-mail:lsxgdjs@gmail.com;王宇英(1978—),女,博士,讲师,主要研究方向为嵌入式实时系统以及 CPS 系统仿真;周兴社(1955—),男,教授,博士生导师,主要研究方向为分布式计算和嵌入式系统理论研究与应用;张 凡(1979—),男,博士,副教授,主要研究方向为嵌入式软件设计与验证。

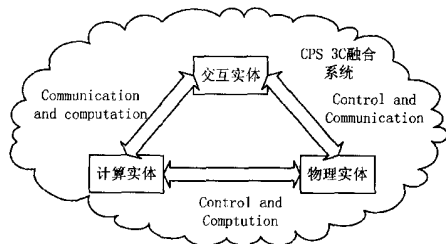


图1 CPS系统结构图

2 CPS建模仿真研究现状

CPS概念的产生具有深刻的历史背景,计算机的出现促使了将计算机集成到物理系统的研究历程。最初的研究是将计算机作为控制器,提高物理对象控制的灵活性,改善物理对象的性能。其中控制学科领域集中研究计算机控制系统,而计算机学科则致力于开发计算速度更快、存储容量更大的计算设备。不管是计算机科学研究计算设备的设计和开发,还是控制科学研究各种物理动态系统的计算机控制,其前提都是将系统抽象为某个模型,然后进行仿真分析和验证。由于这两个学科的研究对象和研究方法不一致,导致研究结果不兼容。计算学科研究计算的时候往往忽略时间因素,将系统抽象为离散事件模型;而控制领域对物理世界的研究,则往往是基于时间的,将系统抽象为连续时间模型,时间是模型中最重要的坐标之一,这将导致计算单元与物理实体模型在交互融合时产生冲突和无法预计的故障。如何创建不同实体模型与异质模型融合,是CPS研究的重要挑战之一^[2],也是本文研究的主要内容。

较之传统的嵌入式系统而言,CPS系统不仅系统规模更加庞大,系统异质性更加明显,而且具有网络化开放性、计算过程与物理过程融合、时空特性鲜明等特点。构建CPS系统更加困难,不同部门之间的构件验证不能及时进行,往往导致开发出来的CPS构件不能协同工作或是开发出来的CPS构件不能满足需求。而CPS系统建模仿真恰好能解决这个问题。在CPS系统构建开发初期,不同的构件首先进行系统级模型建模,不同构件间的交互验证能够在早期就可以很好地进行反复迭代。“模型即软件,建模即开发”是模型驱动软件开发的核心思想,也是研究CPS系统建模仿真的另一重要意义。构建完成的计算实体模型通过面向特定硬件的代码生成工具完成CPS计算实体的代码开发。

在CPS系统建模仿真研究上,美国密苏里科技大学的Lin Jing等人使用基于代理的建模方式(agent-based modeling)对CPS系统进行建模,用代理(agent)来消除CPS系统中信息层和物理层的差异,并用代理(agent)来表示信息层和物理层的属性^[3]。德克萨斯大学在面向服务的体系结构(SOA)的基础上提出了用于CPS建模的物理实体的面向服务体系结构(Physical-Entity service oriented model)^[4]。但是这些方法带有浓重的分布式计算和Web服务计算思想的烙印,不能很好地解决物理实体和计算实体的建模任务。CPS研究的领军人物伯克利大学的艾德华·李在深入研究CPS特征和挑战后认为,CPS系统也可以被称作深度嵌入式系统,通过总结现有仿真控制建模方法提出了“信息系统物理化和物理系统信息化”的概念,来解决物理实体和信息实体的建模以及二者的交互问题,对物理子系统进行软件化抽象封装,使物

理子系统具有可与信息层交互的特性;同时抽象软件组件和网络组件,使其具有物理世界实时的时间特性^[5,6]。目前国内对CPS的研究方兴未艾,对于CPS系统概念的讨论开展得如火如荼,但是对于CPS系统仿真建模的研究还处于起步阶段,相比国外的研究盛况,国内这方面的研究文献寥寥无几。

3 CPS系统建模仿真方法

CPS系统按照功能划分为物理实体、计算实体和交互实体,但是交互实体往往是由计算构件和物理构件组成,表现出来的属性是物理实体和计算实体的组合特性。所以本文在对CPS系统进行建模仿真时,分别针对物理实体和计算实体进行讨论。

3.1 物理实体建模仿真方法

在CPS系统中,物理实体是系统的动作部分,是系统中被控制的物理世界。物理实体具有鲜明的动态连续性,状态的动态转换是基于时间的,时间和空间不是独立的,具有明显的时空转换特性。下面以无人机的飞行状态模型来阐述如何对物理实体进行建模。

首先建立无人机的三维空间坐标。依据右手原则, x 轴的正方向为页面右方, y 轴的正方向为竖直向上方向, z 轴正方向为页面的前方。根据牛顿定律对无人机的位置状态进行建模:

$$F(t) = M\ddot{x}(t) \quad (1)$$

式中, M 表示无人机质量, $\ddot{x}(t)$ 表示无人机的当前加速度, $F(t)$ 表示无人机 t 时刻的当前受力。并且 $M, \ddot{x}(t), F(t)$ 都是一个三维向量组。

假设无人机的初始速度为 $x'(0)$,那么对于任意时刻的速度 $x'(t)$ 都有:

$$\forall t > 0, x'(t) = x'(0) + \int_0^t \ddot{x}(\tau) d\tau \quad (2)$$

由式(1)和式(2)推出:

$$\forall t > 0, x'(t) = x'(0) + \frac{1}{M} \int_0^t F(\tau) d\tau \quad (3)$$

相应的位移公式为:

$$\begin{aligned} x(t) &= x(0) + \int_0^t x'(\tau) d\tau \\ &= x(0) + tx'(0) + \frac{1}{M} \int_0^t \int_0^\tau F(a) da d\tau \end{aligned} \quad (4)$$

通过式(4)不难看出,如果知道该无人机的初始位移、初始速度以及作用于该物体的力向量,那么任意时刻,该无人机的位移、速度、加速度等物理状态都能够计算出来。

在直线运动中,有 $F(t) = M\ddot{x}(t)$ 。在旋转运动中则有 $T(t) = \frac{d}{dt}(I(t)\dot{\theta}(t))$,即

$$T(t) = I(t)\ddot{\theta}(t) \quad (6)$$

式中, $T(t)$ 是力矩, $I(t)$ 是惯性张量, $\dot{\theta}(t)$ 是角速度, $\ddot{\theta}(t)$ 是角加速度。与速度 $\dot{x}(t)$ 、加速度 $\ddot{x}(t)$ 及位移 $x(t)$ 的推导过程类似,很容易得出任意时刻 t 的角速度 $\dot{\theta}(t)$ 方程:

$$\dot{\theta}(t) = \dot{\theta}(0) + \int_0^t \ddot{\theta}(\tau) d\tau \quad (7)$$

$$\dot{\theta}(t) = \dot{\theta}(0) + \frac{1}{I} \int_0^t T(\tau) d\tau \quad (8)$$

式中, $\dot{\theta}(0)$ 是初始时刻物体的角速度。任意时刻物体的偏转角度 $\theta(t)$ 的建模方程如下:

$$\theta(t) = \theta(0) + \int_0^t \dot{\theta}(\tau) d\tau = \theta(0) + \dot{\theta}(0)t + \frac{1}{I} \int_0^t \int_0^\tau T(\alpha) d\alpha d\tau \quad (9)$$

式中, $\theta(0)$ 是初始偏转角度。依据式(9), 如果知道物体的初始偏转角度、初始角速度, 则可以确定该物体的旋转运动状态^[7,8]。

通过对无人机的两种不同运动状态的建模, 可以总结出对物理实体的建模方法。对物理实体状态的刻画, 可以通过基于时间刻度的运动方程来描述, 用物理定律推导出物理实体模型。运用基于时间的建模方法对物理实体进行连续时间建模, 不仅具有很好的直观性和精确性, 而且对物理实体的空间和时间属性进行了很好的描述, 能够体现 CPS 系统中物理实体的时空转换性。

3.2 计算实体建模仿真方法

计算实体是在信息物理融合系统中对物理世界进行控制和计算的部分, 是 CPS 系统中的核心单元。一个 CPS 系统的功能属性以及安全性、可靠性、健壮性等非功能属性很大程度上取决于计算实体的设计和实现。

与物理实体相比, 计算实体最显著的特征是在系统中没有一个确定的时钟, 而且似乎也不需要这么一个时钟; 此外, 在物理世界中的运动过程一般都是基于时间的动态连续过程, 而计算世界中的系统行为一般是基于事件驱动的离散过程。下面以一个恒温器的控制过程来阐述如何应用有限状态机(Finite State Machine)图对计算实体进行建模仿真。

恒温控制系统是一个涉及加热和通风的空气调节系统, 是一个典型的反馈控制系统。一个对温度精确控制调节的模型, 无疑能够更好地减少能源的消耗, 并且使温度更加舒适。考虑这样一个恒温控制系统, 该系统要求温度控制在 20 摄氏度, 系统包括一个加热器。如果加热器不工作, 温度会随着时间的推移而逐渐下降。下面我们应用有限状态机来对系统进行建模。首先分析系统状态, 该系统包括加热(heating)和冷却(cooling)两个状态。系统的输入是温度(temperature), 并且 $temperature \in R$ (R 是全体实数); 其输出是数字信号(pure data) *heatOn* 和 *heatOff*。如果该系统设置为当温度低于 20 摄氏度时加热器就开始工作, 而一旦超过 20 摄氏度加热器就立刻停止工作, 这个系统就会出现剧烈的抖动, 加热器不停地进行开关动作, 这在实际情况中是不可想象的。为了消除这种抖动, 该系统设计为当温度低于 19 摄氏度时加热器开始工作; 当温度高于 21 摄氏度时加热器终止工作。该系统的状态转换图如图 2 所示。

input: temperature; R

outputs: heatOn, heatOff; pure

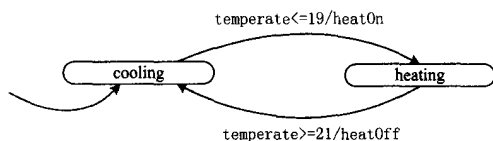


图 2 恒温器状态转换模型

有限状态转换图能够清晰地刻画计算实体的动态离散过程, 是当前用来描述计算实体模型的有效方法。但是这种方法的缺点是状态转换是基于离散事件的, 状态转换发生在前

置条件满足的那个时刻, 忽略了两个事件之间的系统运行状态。这种方法虽然简化了系统, 但是也造成了离散事件系统模型和连续时间系统模型无法交互的问题。

3.3 计算实体和物理实体混成建模方法

物理实体通过基于时间的运动方程构建连续时间模型, 计算实体采用有限状态机图来构建事件离散模型。尽管这两种建模方式能够胜任各自领域的建模仿真, 但是建立的两种异质模型存在无法交互融合的问题。下面介绍在上述仿真建模方法的基础上进行扩展, 使离散事件模型物理化、连续化; 使连续时间模型信息化、离散化, 然后将扩展的两种模型通过融合对 CPS 系统进行建模仿真。

物理实体建模采用运动方程、等价公式刻画。我们将方程的初始值 $X(0) \in R(X(0)$ 是初始状态, 比如说初始位移、初始角度、初始速度、初始偏转角构成的向量, R 是实数向量) 和时间 t 看作前置条件, 而将当前时刻系统受到的力或是力矩等系统受到的物理作用当作是在时刻 t 的系统输入, 而输出就是系统在时刻 t 的状态, 比如位移、速度、加速度、偏转角等系统状态。这样就可以采用类似于“黑盒”的方法, 将微分方程进行封装, 构建物理实体的行为模型。图 3 所示是无人机 6 自由度方程封装之后的行为模型。

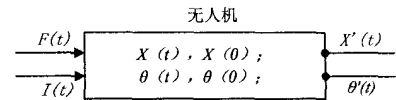


图 3 无人机行为模型

计算实体建模采用有限状态机来刻画系统的行为。状态机的输入是由输入集定义的, 输入集可以是数字信号(0 或者 1), 也可以是数值信号(每次状态转换时的数值)。输出是由状态机转换过程的行为决定的, 并且其也可以是数字信号或是数值信号。输入可以认为是行为模型的输入端口, 我们假定输入端口有 n 组输入, 分别用 i_1, \dots, i_n 表示。在每一步转换中, 这些输入或者是数字信号(0 或是 1), 或者是数值信号。输出端口也是类似的。状态转换条件(guards)决定输入端口的值, 行为动作决定输出端口的值。这样就可以很容易将状态机模型转换为行为模型。图 4 所示是将一个有限状态机模型封装后的行为模型。通过行为模型可以很直观地将时间连续信号作为输入, 归纳到有限状态机模型中去。

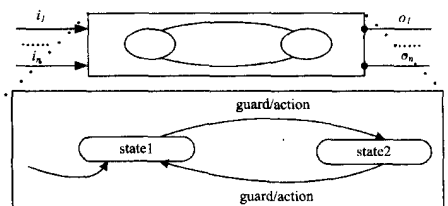


图 4 状态机行为模型

将时间连续模型与事件驱动的状态机行为模型融合起来, 还需要状态精化的概念。在一个 CPS 系统中, 需要对状态机的当前状态进行精化。精化就是将输出的动态行为作为输入的一个函数。大部分的 CPS 系统需要时间来衡量时间段和标记在某时刻的系统响应, 状态按照时间线性转换, 时钟作为测量时间段的工具具有简单而特殊的动态性。时间状态机是 Alur 和 Dill 在 1994 年提出的形式化建模方法, 能基于时钟构建简单复杂系统。依然以一个恒温系统来论述如何应

用状态精炼的概念以及时间状态机对 CPS 系统进行建模。

在前面的恒温器例子中,我们对恒温器的控制系统也就是计算系统进行了建模。下面我们对整个恒温系统进行建模。前面的恒温器例子中用独立的温度上限或是下限来消除抖动震荡。本例中我们采用另外一种方式,即加热器工作或是关闭至少有一个很小的时间段,而不管当前温度是多少。图 5 所示是该系统基于时间状态机和状态精化建立的模型,每个状态精化都有一个时钟 $s(t)$,这个时钟是一个连续动态时间信号,并且 $s'(t)=1, s(t)$ 的值随 t 线性增长。

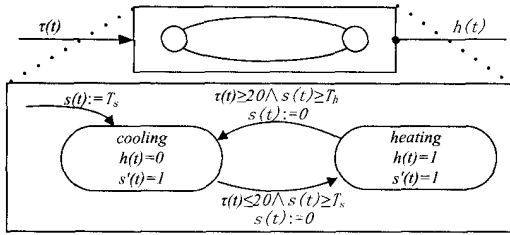


图 5 恒温控制系统的 CPS 系统模型

状态转换的起始条件是 $S(t) \geq T_s$,我们用符号: = 来强调这是一个赋值而不是断言。这个初始条件保证了当恒温器启动时,如果温度 τ 低于或者等于 20 摄氏度能够立刻转换到加热模式。状态转化条件 $s(t) := 0$ 表示的意思是重置时钟使其归零。状态转换条件部分 $s(t) \geq T_h$ 确保加热器在打开之后至少工作 T_h 时间。另一个条件 $s(t) \geq T_s$ 确保加热器关闭至少 T_s 时间。

离散事件是嵌入到一个个时间段中的,系统在不同的状态和操作中进行离散状态转换,而在每个状态之内的行为用基于时间的精化状态来描述,从而实现物理实体和计算实体的混合建模。在混合建模方法中,系统被划分为两部分,也可以认为是两个层次。状态机用来刻画计算实体,设计满足状态转换的逻辑;精化状态系统用来描述物理实体以及每个模式的时间连续行为。

4 建模仿真工具研究

针对物理实体和计算实体的特征和建模方法,分别采用 Simulink/RTW 和 UML/Rhapsody 对物理实体进行建模和对计算实体进行建模,然后按照 CPS 系统异质模型融合思想将两个不同工具的模型进行融合。

物理实体建模方法涉及到大量的微积分方程计算和时间连续行为,故采用 Simulink/RTW 来进行建模和代码生成。Simulink 被广泛应用于控制领域建模仿真,是一种高度图形化的建模仿真工具,用来对动态系统进行建模、仿真和分析,是一种基于时间连续系统建模的可视化建模工具。Simulink 具有强大模块库,能够完成各种建模需求,并且应用 S 函数机制,用户可以方便地编写各种功能的 Simulink 模块^[9]。而且 Simulink 和 Matlab 的无缝连接,使 Simulink 能够很容易调用 Matlab 的数学计算工具,具备强大的计算能力。RTW (real-time workspace) 是 Simulink 的代码生成工具,能够实现从平台无关的 Simulink 模型生成各种语言的平台相关可执行代码,比如 C, C++, ADA(一种面向嵌入式系统和实时系统的编程语言)等。

统一建模语言 (UML) 是模型驱动的软件开发方法中广泛使用的建模和规约语言。最常用的 UML 图包括用例图、

类图、序列图、状态图、活动图、组件图和部署图。UML 建模语言以丰富的图形种类从不同角度支持系统设计与开发的各个阶段,是面向对象设计的标准语言,支持模块复用,在系统设计方面具有很大的优势,支持对模型设计的系统仿真,验证设计的各个模块之间的交互、通信和依赖、继承等关系^[10]。Rhapsody 是 IBM 推出的 UML 统一建模工具,遵循 UML2.1 协议,通过增加和定制特征文件(profile)来支持面向特定领域的仿真建模,支持模型代码的自动生成,适用于事件驱动的计算实体建模仿真。

UML/Simulink 是一个功能强大的组合。在设计阶段,用户可以为每一个不同的活动选择最合适的建模工具。使用 UML,用户可以按需求捕获和分析函数,定义系统和软件架构,组织算法逻辑,模拟测试每一个模块的正确性,构建计算实体模型。使用 Simulink,用户可以创建设备模型定义的物理元素。它与系统交互并生成逻辑算法,控制这些设备模型的动态行为。代码可以从原型中使用的控制算法里产生,同时可以从硬件回路测试的设备模型中生成,模拟物理实体的动态特性^[11]。

图 6 所示是 UML 模型和 Simulink 模型的融合结构。从外部看,这是一个 CPS 节点模型,但是在系统级上划分为物理实体模型和信息实体模型。底层的模块则采用 Simulink 和 UML 模块来实现。最底层的 C 代码是将两种工具生成的代码手工融合或是通过相关工具糅合起来。UML 模型和 Simulink 模型的融合在 Rhapsody 中可以通过下述方式来实现。图 7 所示是模型融合流程图。

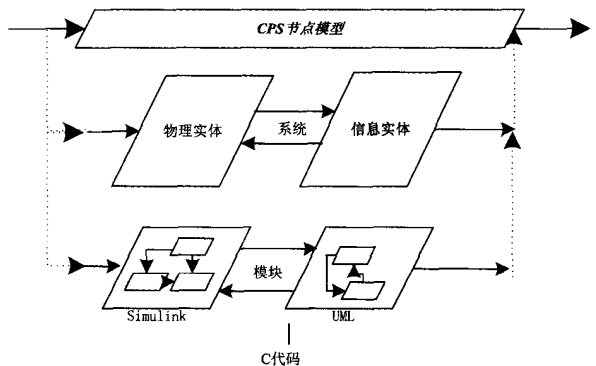


图 6 Simulink 模型和 UML 模型融合结构图

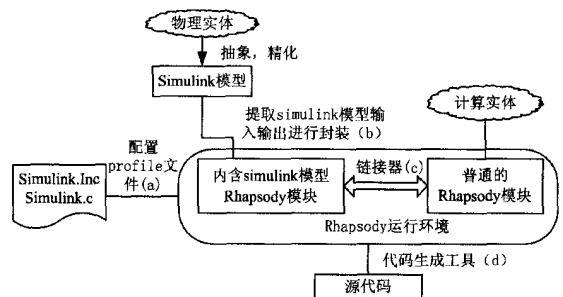


图 7 Simulink 模型和 Rhapsody 模型融合流程图

a) 在 UML 中定义了模块扩展机制, Rhapsody 通过设置 Profile 文件来实现。如果要在 UML 模型中包括 Simulink 模型需要首先在 Profile 中添加 Simulink 和 SimulinkInc 两个配置文件。前者支持在 UML 中通过调用 MATLAB/Simulink 的 RTW 工具生成 C++ 代码, 后者支持生成 C 代码。

(下转第 68 页)

参考文献

- [1] Papapetrou E, Karapantazis S, Pavlidou F N. Distributed on-demand routing for LEO satellite systems[J]. Computer Networks, 2007, 51(15): 4356-4376
- [2] 唐剑, 余春东, 徐志明. LEO/MEO 卫星网络动态多径路由协议[J]. 计算机科学, 2009, 36(10): 42-45
- [3] 尹志忠, 张龙, 周贤伟, 等. 3 层体系结构卫星网络的性能分析[J]. 计算机工程, 2010, 36(12): 10-12
- [4] Zhou Y, Sun F, Zhang B. A novel QoS routing protocol for LEO and MEO satellite networks [J]. International Journal of Satellite Communications and Networking, 2007, 25(6): 603-617
- [5] 潘艳辉, 王韬, 李华. 非静止轨道卫星网络面向应用仿真分析[J]. 计算机仿真, 2011(8): 112-116
- [6] Nguyen H N, Jukan A. An Approach to QoS-Based Routing for LEO Satellite Networks[C]//Proceedings of IEEE Global Telecommunications Conference (GLOBECOM' 00). Vol. 2, San Francisco, CA, IEEE, 2000: 1114-1118
- [7] Xu Hui, Huang Fei, Wu Shi-qi. A distributed QoS routing based on ant algorithm for LEO satellite network [J]. Chinese Journal of Electronics, 2007, 24(6): 765-771
- [8] Karapantazis S, Papapetrou E, Pavlidou F N. Multiservice on-demand routing in LEO satellite networks[J]. IEEE Transactions on Wireless Communications, 2009, 8(1): 107-112
- [9] 肖甫, 孙力娟, 叶晓国, 等. 面向卫星网络的流量工程路由算法[J]. 通信学报, 2011, 32(5): 104-111
- [10] Huang Q, Yeo B S, Kong P Y. An enhanced QoS routing algorithm for provision of end-to-end delay guarantee in low earth orbit satellite networks[C]//Proceedings of 2005 IEEE Wireless Communications and Networking Conference (WCNC2005). Vol 3, New Orleans, USA: IEEE, March 2005: 1485-1490
- [11] Svigelj A, Mohoric M, Kandus G, et al. Routing in ISL networks considering empirical IP traffic[J]. IEEE Journal on Selected Areas in Communications, 2004, 22(2): 261-272
- [12] 朱军, 饶元, 傅雷扬, 等. 基于移动代理的卫星网路由性能研究[J]. 计算机工程与应用, 2012, 48(3): 69-72
- [13] Taleb T, Mashimo D, Jamalipour A. Explicit Load Balancing Technique for NGEOSatellite IP Networks with on-Board Processing Capabilities [J]. IEEE/ACM Transactions on Networking, 2009, 17(1): 281-293

(上接第 35 页)

b) 将一个 Simulink 模块添加到 UML 模型采用类似于“黑盒测试”的方法, 提取 Simulink 模块的输入、输出向量, 为 Simulink 模块增加数据流端口 (flowport)。数据流端口是 UML 中定义的模块交互接口。图 8 所示是封装输入输出接口的 Simulink 模块。

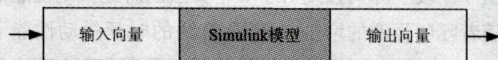


图 8 封装为 UML 模块的 Simulink 模型

c) 在 Rhapsody 中的 Simulink 模块之间的交互或是 Simulink 模块和其他 UML/Rhapsody 模块之间的交互与普通的 UML/Rhapsody 模块的流端口之间的交互方式一样, 都是采用链接器 (link connectors) 来实现, 链接器是 UML 定义的模块间交互方法。

d) 包含有 Simulink 模型的 Rhapsody 模块生成代码时, Simulink 模型通过调用 RTW 来实现代码生成, UML/Rhapsody 模块则通过 Rhapsody 自带的代码生成工具来实现, RTW 生成的代码则被包裹在 UML 生成的代码之中。

结束语 本文首先在分析 CPS 系统的特征和属性基础上将 CPS 系统划分为物理实体、计算实体和交互实体, 再通过精化、提取将 CPS 模型简化, 将系统划分为物理实体和计算实体。采用不同实体不同建模方式的思想, 分别提出了针对物理实体的基于连续时间的运动方程行为模型和面向计算实体的基于事件驱动的有限状态机行为模型。然后通过扩展物理实体行为模型和计算实体行为模型, 实现了物理模型信息化和计算模型物理化, 使两种模型融合在一起协同工作, 刻画 CPS 系统。最后对两种模型所能采用的建模工具进行了可行分析, 提出了物理行为模型采用 Simulink/RTW 来构建和生成代码, 计算实体模型采用统一建模语言 (UML) 进行建模, 并在此基础上提出了将 Simulink 模型转化为 UML 模型

方法, 完成了两种异质模型的协同建模仿真。

参考文献

- [1] 何积丰. Cyber-Physical-System [J]. 中国计算机学会通讯, 2010, 6(1): 25-29
- [2] Derler P, Lee E A, Sangiovanni-Vincentelli A L. Addressing modeling challenges in cyber-physical systems[R]. UCB/EECS-2011-17. EECS Department, Mar. 2011
- [3] Lin Jing, Sedigh S. A General Framework for Quantitative Modeling of Dependability in Cyber-Physical Systems; A Proposal for Doctoral Research[C]//33rd Annual IEEE International Computer Software and Applications Conference. 2009
- [4] Huang J, Bastani F. Extending Service Model to Build an Effective Service Composition Framework for Cyber-Physical Systems[J]. Service-Oriented Computing and Applications (SOCA), 2009, 1(1-8): 14-15
- [5] Lee E A. Cyber Physical Systems; Design Challenges[C/OL]. International Symposium on Object /Component /Service-Oriented Real-Time Distributed Computing (ISORC). 2010-03-22
- [6] Lee E A. Computing Foundations and Practice for Cyber-Physical Systems; A Preliminary Report[R]. No. UCB/EECS-2007-72. 2007
- [7] 杨新, 王小虎, 等. 飞机六自由度模型及仿真研究[J]. 系统仿真学报, 2000, 12(5): 210-213
- [8] 李文强, 彭学锋, 郑志强. 基于 Simulink 的无人机六自由度仿真[J]. 系统仿真学报, 2007, 19(19): 4604-4606
- [9] Hooman J, Mulyar N, Posta L. Coupling Simulink and UML models[C]//Proc. Symposium (FORMSIFORMATS). 2004: 304-311
- [10] 沙静, 蒋昌俊. 基于 UML 实时状态图的定量分析[J]. 系统仿真学报, 2008, 20(8): 1957-1960
- [11] Farkas T, Meiseki E, Neumann C. Intergration of UML with Simulink into Embedded Software Engineering[C]//ICROS-SICE International Joint Conference 2009. 2009: 474-479