基于 Real-Time Object-Z 语言的实时系统形式化描述*

魏艳铭[1,2], 张广泉[2]
(1. 重庆师范大学 数学与计算机学院, 重庆 400047; 2. 河南经贸职业学院 信息管理系, 郑州 450053)

摘 要:实时系统是一类需要考虑时间约束条件的反应系统,确保实时系统安全性和可靠性是至关重要的。形式化方法是建立在严密数学基础之上的开发方法,采用形式化方法对实时系统进行描述与验证,可以借助严密的数学证明提高实时系统的安全性和可靠性。本文讨论 Object-Z 的一种实时扩展语言 Real-Time Object-Z,它可以对实时系统进行形式化描述,文中以室温控制系统为例,详细说明了 Real-Time Object-Z 语言在实时系统形式化描述中的应用方法。

关键词:实时系统;Object-Z;Real-Time Object-Z;实时精化演算;形式化描述
中图分类号:TP311 文献标识码:A 文章编号:1672-6693(2007)04-0041-04

实时系统(Real-time System)[1]是一类需要考虑时间约束的反应系统(Reactive System),通常认为实时系统的正确性不仅有赖于处理结果的正确性,还取决于其运行时间的约束性。实时系统的应用场合非常多,如核反应堆、航运飞行控制、机器人控制以及军事领域等等,几乎所有安全有关的系统都是实时系统,因此确保其安全性和可靠性是至关重要的。形式化方法(Formal Methods)是建立在严格数学基础之上的软件开发方法[2],是一系列基于严格数学理论的、用来描述和分析系统性质和行为的符号、技术和手段。因此采用形式化方法对实时系统进行描述与验证,可以借助严密的数学证明提高实时系统的安全性和可靠性。

形式化描述(Formal Specification)是形式化方法中一个重要的研究内容,它是对系统做什么的数学描述,是用具有精确语义的形式化描述语言书写的系统功能和性质的描述。Object-Z 就是一种形式化描述语言,它是 Z 语言的面向对象扩展,可以描述复杂的数据结构,但却缺少描述时间约束和连续变量的机制,因此对 Object-Z 进行扩充,以适应实时应用的需求是非常重要的。Real-Time Object-Z 是 Object-Z 的一种实时扩展语言,它是 Object-Z 与实时精化演算的结合,它不仅可以描述系统复杂的数据结构,还可以描述系统所需的时间约束。本文首先对 Real-Time Object-Z 语言进行了概述,然后以室温控制系统为例,详细说明了 Real-Time Object-Z 语言在实时系统形式化描述中的应用方法。

1 Real-Time Object-Z 语言概述

1.1 Object-Z 语言简介

Object-Z[3-4]是形式化描述语言 Z 的面向对象扩展,它基于集合论与数理逻辑,具有严密的逻辑性,适合于精确地描述大型软件系统。类模式是 Object-Z中重要的面向对象结构,其结构如图 1 所示,类模式包括可见列表、父类列表、局部定义、状态模式、初始化操作模式和操作模式等。其中,状态变量定义在无名状态模式中,可以被该类中所有的操作使用,状态变量可以有其约束条件,即状态不变式,它定义了状态空间,对象的每个状态是其中的一个状态。初始化操作模式用来给状态变量赋初值,通常命名为 INIT。操作中可以有变量定义部分及谓词部分,变量定义部分可以声明在本操作中要改变的状态变量,并把它们放在 Δ 列表中,还可以定义局部的输入与输出变量,这些局部输入与输出变量只能在本操作中起作用,其它的操作不能更改这些局部变量,其中输入和输入变量的后缀标识符分别为?和!;谓词部分隐式定义了其前置条件及后置条件,每个操作只有在满足了它的前置条件时才会执行,而执行完毕一定又要满足它的后置条件。

* 收稿日期:2007-03-26 修回日期:2007-06-27
资助项目:重庆市教委科学技术研究项目(No. 040803)
作者简介:魏艳铭(1981-),女,河南濮阳人,硕士,研究方向为软件工程与形式化方法等。

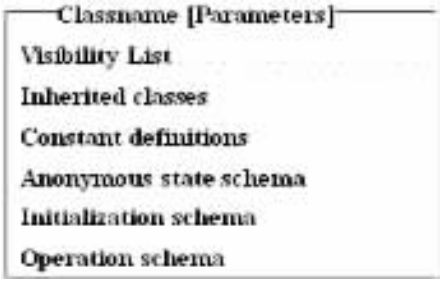

图1 Object-Z 类模式

采用 Object-Z 语言对系统进行形式化描述时,通常用一组个体类和一个系统类对整个系统进行描述,一旦个体对象的行为采用类来描述,那么从系统的观点看,整个系统就是由个体类合成的系统类来表示。另外基于面向对象的思想,Object-Z 提供了一种机制,使得类对象可以在另一个类中被引用,通过这种机制,系统类捕捉一组个体对象的行为、关系及交互,如,类 C 在另一个类中被引用,则可以声明为 $a:C$,此时状态变量 a 是此类对象的引用,要访问此对象的变量或常量 x,则表示为 $a.x$,同样,访问它的初始条件 INIT 和操作 Op 则分别表示为 $a.$INIT 和 $a.$Op。由于 Object-Z 支持面向对象,因此它方便地描述了面向对象的特点,如继承、多态和封装等。此外,Object-Z 语言还定义了许多操作符,可以满足操作表达式之间的变更和组合,如用于表示两个不同的操作同时发生的连接符"Λ"、表示两个并发操作之间的通信的并行组合符"‖"、可进行两个操作之间的非确定性选择的不确定选择符"□"和用于描述两个操作按顺序依次发生的相续组合符";"等等。

1.2 实时精化演算

实时精化演算(Timed Refinement Calculus)[5-6]是一种基于 Z 语言的、用来对实时系统进行描述和求精的表示法,1998 年,C. J. Fidge 等用建立在集合论基础之上的时间迹形式简化了实时精化演算的表示法,并少量地引入新符号,对实时精化演算作了扩展。下面是实时精化演算中用于描述时间约束的几种机制。

(1)时间集 T。是对自然界时间的模拟,用实数集表示,即 $T==\mathbf{R}$。保留字 τ 表示当前时刻,在本文中,假设时间单位为秒。

(2)时间迹(Timed Traces)表示法。在实时精化演算里,把随时间变化的变量用一个从时间集 T 到某个类型集合的映射函数表示。如用来表示交通系统中红灯是否点亮的布尔量 on,可以被记为 $on:T\rightarrow B$,其中 B 是布尔值集合;又如,描述外界温度的连续变量 $temp$,它是随时间连续变化的,被表示为 $temp:T \rightarrow Celsius$,其中 $Celsius==\mathbf{R}$,$\mathbf{R}$ 就是实数集。

(3)$<P>$ 用来表示谓词 P 为真的所有时间区间集合,这里的谓词 P 可以是常量、变量或是逻辑表达式,如,对于变量 $on:T\rightarrow B$,$<on>$ 表示布尔量 on 为真的所有时间区间;系统的操作名 Op 也可以出现在谓词 P 里,此时操作名 Op 被定义为布尔类型的变量,$<Op>$ 表示操作发生的所有时间区间集;另外,时间区间的时间段用保留字 δ 表示,如 $<\delta=1>$ 表示区间长度为 1 s 的所有时间区间集。不同的时间区间集之间可以通过集合论中的符号,如 $\subseteq$、$\subset$、$\cap$、$\cup$ 等,进行连接,来表示区间集里谓词之间的关系,如:

$$<u>10<\delta=0.1>;<v=u>$$

则描述了,对于 $u>10$ 时间区间里的任意时刻,都有 0.1 s 后 $v=u$ 成立,其中";"是两个时间区间集之间的连续符。

1.3 Object-Z 与实时精化演算结合

Real-Time Object-Z[7] 是 Object-Z 与实时精化演算的结合,这种结合并不是提出一套新的语法和语义,而是结合使用现有的符号和方法来描述系统的需求,并尽可能少地应用新符号,原来的推理方法、规则及工具仍旧可以使用。结合后产生的 Real-Time Object-Z 语言不仅可以描述系统复杂的数据结构,还能描述系统的时间约束,从而可以对实时系统进行形式化描述。

在 Real-Time Object-Z 语言里,类模式由水平线分为两部分,上面的部分是标准的 Object-Z 部分,系统中随时间变化的变量也定义在这部分里;下面的部分包含了对类环境进行约束的假设谓词(Assumption)和在类环境约束下成立的结果(Effect)谓词。如图 2 所示。

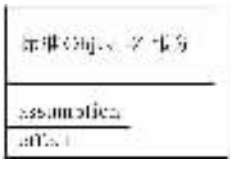

图2 Real-Time Object-Z 的类模式框图

2 实例应用

2.1 室温控制系统需求说明

室温控制系统由门、控制系统(门控系统和温

控系统)、开关按钮和空调组成。最初,系统处于停止状态,在按下“运行”按钮后,整个系统开始工作;正常室内温度范围是 22 ~ 26℃,处于正常室温下,空调停止工作,而当温控系统检测到室内温度低于该范围,就向空调发出命令,使其开始制热,相反,当室内温度高于该范围时,空调受控,开始制冷;在空调运行中,由门监控器监测门的状态,并把状态传递给门控系统,当门控制器检测到门被打开的时间超过了 60 min,那么它就向空调发出停止命令;整个系统在按下“停止”按钮后,被强行停止。如图 3 所示,其中箭头上的英文为相应的输入和输出变量。

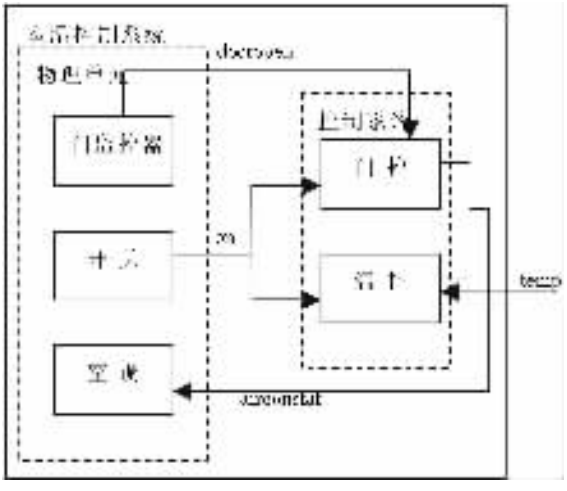

图 3　室温控制系统结构图

2.2 系统的形式化描述

对室温控制系统进行逐层分解:1)室温控制系统由控制系统和物理单元两部分组成;2)控制系统由门控系统和温控系统两部分组成;3)物理单元由系统开关、门监控器及空调组成。

本文只列举控制系统和整个系统类的形式化描述,在开始对各部分进行描述时,先定义以下几个自由类型:

$$Celsius == \mathbf{R}$$

$$Airconstate ::= \{0,1,2\}$$

Celsius 是所有室内温度的集合,用实数集 **R** 来表示;*Airconstate* 表示空调工作的状态,其中 0、1、2 分别表示空调的停止、制冷和制热。

在描述外界温度时,由于它是随时间连续变化的,并不受系统里某个操作的执行而改变,因此把它作为系统的环境变量,定义为:$temp?: T \to Celsius$,并且这里假设时间单位为秒,用常量 $min = 60$ 表示分钟。

下面分别给出门控、温控和系统类形式化描述。

(1)门控类模式 DoorControl(见图 4)。对于门控的形式化描述中,最重要的是要描述出,在门开这一状态保持 60 min 之后,对空调发出停止命令。笔者在此设置了一个 Add 计数操作,并且此操作每分钟执行一次,使变量 *num* 自加一,通过变量 *num* 的变化来表示时间的推进,在 $num > 60$ 时,自动清零。在 assumption 和 effect 谓词部分里,就是对 Add 操作的时间约束,即在门开和空调正在运行的时间环境约束下,Add 操作每一分钟就要执行一次,保证了 Add 操作的循环执行,从而正确的描述出了此操作的时间约束。

图 4　门控类模式

(2)温控类模式 TempControl(见图 5)。在温控类模式里,assumption 和 effect 谓词部分里很好地描述出了在整个系统工作的时间段里,当处于不同温度范围内时,向空调发出不同信号的 3 个操作执行的时间约束。

(3)室温控制系统类模式 TempControlSystem(见图 6)在系统类的形式化描述中,必须保证所有对象中的变量一致;从室温控制系统类模式中可以看出,它的 4 个操作表达式是通过操作符把对象的操作表达式组合而得到的。到此,整个室温控制系统的形式化描述就完成了。

图5 温控类模式

图6 室温控制系统类模式

3 总 结

本文讨论了 Real-Time Object-Z 语言的形式化描述方法,并用该语言对一个实时系统——室温控制系统进行了形式化描述。Real-Time Object-Z 语言是 Object-Z 语言与实时精化演算的结合,它不仅可以描述系统的数据结构,还可以描述系统中所要求的时间约束,因此它可以用来对实时系统进行形式化描述与验证,从而提高实时系统的安全性和可靠性。

参考文献:

[1] 李广元. LTLC:面向实时与混成系统的连续时序逻辑[D]. 博士学位论文,北京:中国科学院,2001.

[2] 张广泉. 关于软件形式化方法[J]. 重庆师范大学(自然科学版),2002,19(2):1-4.

[3] SMITH G. The Object-Z Specification Language[M]. Advances in Formal Methods. Kluwer Academic Publishers, 2000.

[4] SMITH G. A Fully Abstract Semantics of Classes for Obj-ect Z[J]. Formal Aspects of Computing,1995,7(3):289-313.

[5] MAHONY B P,HAYES I J. A Case-study in Timed Refinement:A Mine Pump[J]. IEEE Transactions on Software Engineering,1992,18(9):817-826.

[6] FIDGE C J, HAYES I J, MARTIN A P, et al. A Set-theoretic Model for Real-time Specification and Reaso-ning [J]. Mathematics of Program Constructing(MPC'98),Lecture Notes in Computer Science, Springer-Verlag, 1998, 1422:188-206.

[7] SMITH G,HAYES I. An Introduction to Real-Time Object-Z[J]. Formal Aspects of Computing,2002,13(2):128-141.

The Formal Specification of Real-Time System Based on Real-Time Object-Z

WEI Yan-ming[1,2], *ZHANG Guang-quan*[1]

(1. College of Mathematics and Computer Science, Chongqing Normal University, Chongqing 400047;
2. Dept. of Information Management, Henan Economy and Trade Vocational College, Zhengzhou 450053, China)

Abstract:Real-time systems are reactive systems that take into account time constraints. There are many computer control systems belonged to real-time systems, for example, nuclear reactor、control in aeronautics、dispatch of railroads and so on. These systems have stronger properties than other systems, such as real-time、concurrent and interactive. Because of their extensive usage in the safety-critical domains, real-time systems lay the most important emphasis on the safety and reliability properties. Based on rigour mathematics,

formal methods can use the symbols of mathematics to specification and verify the systems. When formal methods are adopted to specify and verify real-time systems, it is certain to improve safety and reliability of real-time systems under rigour proofs. Formal specification is an important part of research for formal methods, Object-Z is a formal specification language, and it is an extension of Z which is also a formal specification language to facilitate specification in an object-oriented style. It is suitable to specify the complex data structures of systems. Usually a system is comprised of many parts. We use the Object-Z notation to specify the system by specifying a group of individual classes and a systemic class respectively, and we can use the Object-Z notation to specify the object-oriented characteristics conveniently, for example, inheritance and so on. But its utility is limited by its inability to specify continuous variables and real-time constrains. So it is necessary to extend Object-Z for accommodating the requirements of real-time. Real-Time Object-Z discussed in this paper is a real-time extension of Object-Z. It is the integration of Object-Z with timed refinement calculus. The way of the integration does not put forward a suit of new syntax and semantics, but use existing symbols and methods to specify the requirement of systems and import less new symbols to the greatest extend. Primary reason、rules and tools are still usable. Real-time Object-Z can specify not only the complex data structures of systems, but also the continuous variables and real-time constrains, so it is suitable for specifying real-time systems. Classes in Real-Time Object-Z comprise of two parts separated by a horizontal line. The part above the line is essentially the standard Object-Z local definitions and schemas. The part below the line contains further constraints on the class specified in the timed refinement calculus notation. In this paper we present Real-time Object-Z and use it to specify indoor temperature control system which is a real-time system. In section 2, we introduce the Object-Z notation, present the specification notation of the timed refinement calculus, and at last show how Object-Z and timed refinement calculus are integrated to produce a new specification language: Real-time Object-Z. For an example, in section 3, we illustrate the application of Real-Time Object-Z by specifying an indoor temperature control system. the indoor temperature control system is comprised of door controller、temperature controller、switch、door detector and air-condition. In the paper, we use a variable to denote the working state of air-condition ,and only present the classes schemas of door controller、temperature controller and the whole system. Finally, we conclude the paper in section 4.

Key words: real-time system; Object-Z; Real-Time Object-Z; timed refinement calculus; formal specification

(责任编辑 游中胜)