维普资讯 http://www.cqvip.com

85-87 软件开发方法 并发软件规约 Petri网

计算机科学2006Vol.27№.7

一种并发软件规约方法的构架*)

A Framework for the Specification of Concurrent Systems

刘剑 李彤

(云南大学计算机科学系 昆明650091)

TP311.52

Abstract Framework has become popular as a vehicle for reuse of solutions in specific areas of application. In this paper, we extend the application of framework to express a class of dual specification methods for concurrent systems. A dual appplication uses graphical notation for the specification of behavior and a temporal logic to express formally, and reason about this behavior.

Keywords Framework, Concurrent system, Formal specification, Software development

1 引言

构架(framework)作为一种解决特定应用领域软件开发的方法而越来越广泛地为人们所接受。它为特定领域中的应用系统提供了可重用框架,即通过抽象相关应用系统的共同特征,为这类系统提出了一种通用的体系结构,这一结构通常由模块、连接以及它们之间的约束组成,这样每个新系统即可通过实例化、具体化这一通用体系结构而建立。

在本文中我们将应用构架来描述一类并发系统的规约方法,用构架来描述这类规约方法将更有利于我们对这类方法的理解和使用。

2 并发软件开发方法

并发软件系统有别于传统的顺序串行程序。一个并发系统通常包括多个并行存在的任务,这些任务既独立又统一,并发地完成系统功能。这种并发性导致系统执行结果常常具有不确定性,使系统开发人员很难设计测试用例。又因为并发软件常常用于要求较高的环境如:实时控制环境、安全关键环境等,因此提供保证其功能正确、性能可靠的手段就显得尤为重要。

并发系统的开发一般包括两个阶段,开发者首先要将整个问题分解为适当规模的可并行计算的子任务,为了产生结构化、可重用和可移植的软件,开发者应为规约设计建立模型。下一阶段就是将并发软件设计模型映射到具体的目标环境上,这种映射可由相应的支持系统自动完成,也可由程序员手工完成。

为了保证所得的最终软件产品功能正确、性能可靠,不仅要求在映射阶段中将所得到的系统模型正确地用目标语言实现(即语法的正确转换),更要求在规约设计阶段得到一个真正满足用户需求的系统模型(即语义的正确描述)。因此在许多并发软件的开发过程中都选用了适当的形式化方法或工具,以便对系统设计建立形式化抽象模型以及对模型进行验证。对并发系统而言,常用的形式化规约工具可分为三类:

一类是抽象的通信描述语言如:CSP、CCS、SL等;这类语言的计算模型是基于描述并发系统中各进程之间的同步或异步关系。因为这类语言主要描述进程间的关系,所以其抽象程度较低。此外,这类语言讨论的是事件、进程等这类系统中行为的关系性质,对系统的状态没有提供有效的表达、验证方式,从而不便于对系统的状态性质进行形式化验证。

一类是Petri网模型,包括各种低级网和高级网如:P/T系统、Pr/T系统、CPN(Colored Petri Net)、APN(Algebraic Petri Net)、OOPN等,Petri网是一种集形式化和图形化为一体的建模规约工具。其核心是用网模型中的库所(Place)和变迁(Transition)来分别描述一个并发系统中的状态和行为,用流关系(Flow)来描述它们之间的因果关系。现行的Petri网验证、分析手段主要是通过求解线性方程组来得到其S-不变量和T-不变量。S-不变量和T-不变量并不是一种广泛的、有意义的系统性质的语义表示,用S-不变量和T-不变量来表达系统性质不便于进行分析推理。此外对高级网(如:OOPN,APN等)而言,其抽象层次和复杂性都比较高,对其求解S-不变量不易于,甚至不可能实现。

一类是时序逻辑如:UNITY、TLA(Temporal Logic of Action)、RTL(Real Time Logic)、TRO等。这类逻辑语言是经典时序逻辑的应用或扩展,其主要思想是基于一阶谓词逻辑和高阶时序逻辑来表达、验证

*)本项目的研究得到云南省自然科学基金(98F005G)和云南省中青年科技带头人培养基金(1998-37)资助。

维普资讯 http://www.cqvip.com

并发(实时)系统应具备的状态性质.这类规约逻辑一般包括两个部分.一个部分用来表达并发系统具有(应具有)的性质;另一部分是一推理规则集,用来从已有性质推理证明目标性质.这类规约方法着眼于表达和验证系统中的较低层次的、较细的状态性质.所以对一个并发系统它能给出其严格的、有意义的逻辑状态语义,有利于对系统的性质进行表达和推理,但这类方法有以下难点.

·难于学习,这类规约方法都以一阶逻辑和高阶逻辑为基础。

·不形象直观,其表示一般都遵循定义、性质、定理、推理规则的数学表示方法。

·难以从较高的层次来对整个系统进行建模.因为这类方法强调对并发系统中状态性质的验证,而对系统中行为的描述能力不强,因此难以从系统行为的层次上来描述整个系统的行为、结构.

为了克服以上难点,在并发系统的开发过程中,还需要另一类系统建模规约工具,这类工具应该有利于设计人员从系统行为的角度对整个系统进行分析建模,有利于设计人员对设计模型进行交流、存档、求精,同时应结合一定的技术来处理好大规模应用系统的复杂性膨胀问题.图形化表示方法为以上问题提供有效的解决方案。

图形化表示工具提供了一种大粒度的系统描述方法,它通过一套大家共同认可的框图、线条和注释等基本符号可在较高的层次上描述一个具体应用系统的整体结构和行为,因此在并发系统的开发过程中开发人员通常都选用或有针对性地扩充一种图形化建模工具来描述规约系统,常用的有.DF、LGDF(Large-grain Data Flow)、StateChart、TSC(Timed StateChart),如上所述的Petri网也是图形化的系统建模工具中的一种,只是其形式化程度较高,具有严格的运行规则。

3 一种并发系统规约方法的构架

基于以上论述,人们想到了用一种二元表示方法来规约并发系统,二元的表示方法包括一个图形化规约工具和一个逻辑规约工具,这样的结合具有以下优点:

·可以使用不同的表示方法来表达一个并发系统的不同侧面;

·在开发过程中可针对不同抽象层次来建立系统的模型;

·该方法具有一个良好的理论基础,这样将有利于继承使用已有的自动化工具,同时有助于实用CASE环境的开发。

下面我们用构架来描述这类二元的并发系统规约方法.该构架包括七个模块,可分为三类:两个主模块用于描述所用的两种规约工具;三个应用模块用来分别表达具体并发系统的模型、行为逻辑和性质,两个操作(关系)模块定义了应用模块之间的关系。

3.1 主模块

·图形工具模块:是对所使用的图形化表示工具的描述.该工具用来描述、规约系统的整体行为和结构。

·逻辑工具模块:是对所用的逻辑规约工具的描述.该工具用来给出系统模型的严格逻辑语义以及对系统性质进行验证。

3.2 应用模块

·系统模型图示:用图形工具模块中提供的图形表示工具来对某一并发应用系统进行建模,表达系统的整体行为和结构。

·模型的逻辑表示:是用上述逻辑工具模块中说明的逻辑规约工具来严格地表达系统模型的语义,即对系统模型进行公理化表示。

·系统性质:是用逻辑规约工具提供的性质描述手段来表达系统应具备的良好性质。

3.3 操作(关系)模块

·映射关系:描述某一并发系统模型的图示和逻辑表示之间的对应关系。

·性质证明器:用逻辑规约方法所提供的定义、定理、推理规则来证明某一系统的逻辑模型满足所要求的性质,若所用的逻辑系统有现成的机器证明系统,否则指手工证明.

两个主模块是在方法一级上对所用的规约工具进行描述;三个应用模块是针对某一具体的并发应用系统用主模块提供的工具分别表示其模型图、模型的逻辑表示和性质;两个操作(关系)模块则说明模型图、逻辑表示和性质之间的关系。

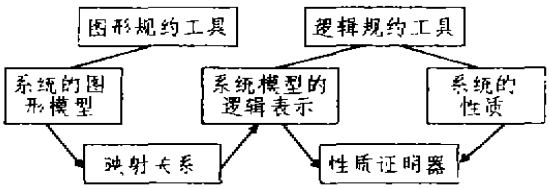

图1 一类并发系统规约方法的构架

上述并发系统规约方法的构架如图1所示。图中最上面的两个框表示主模块;中间的三个框表示应用模块;最下面的两个框表示操作(关系)模块.主模块和应用模块间的连线表示的是所选择的规约工具如何来支持描述系统模型的不同侧面;应用模块和操作(关系)模块间的连线表示并发系统模型中不同侧面之间的关系。

4 两个实例

上述的构架是对一类并发系统规约方法的抽象,选取不同的规约工具嵌入该构架内将得到不同的规约

维普资讯 http://www.cqvip.com

方法，下面给出两个具体的并发系统规约方法，这两种方法都是上述的规约构架的实例化。

例1 ArchSM

ArchSM（Architectural Specification Method）[3]项目主要是研究实时反应系统的形式化规约方法和开发实用的CASE环境。ArchSM支持对反应系统的三个重要方面：实时性质、行为逻辑和系统结构进行建模和验证。ArchSM使用TSC（Timed State Charts）作为图形规约工具来描述一个并发系统的行为结构模型。TSC是状态图的扩展，它将实时的时间表示引入到状态图（StateCharts）中。同时ArchSM使用RTL（Real Time Logic）来建立系统模型的逻辑表示和对时序性质进行验证。ArchSM可看作是以上框架的一个具体实例。即：

系统模型图示	TSC（Timed State Charts）
模型逻辑表示	RTL（Real Time Logic）
映射关系	一系列转换规则 TSC→RTL
性质证明器	HOL

例2 TRIO

TRIO是一种一阶时序逻辑，该逻辑的时序算符可用来表达和验证系统的实时状态变化。Felder等在文[3]中提出了一种将TPN（Timed Petri Nets）和TRIO融合在一起来规约、验证并发系统的方法。该方法用TPN来对系统整体的行为和结构进行图形建模，用TRIO来表达TPN系统模型的逻辑表示，其转换过程的核心是将TPN的网拓扑转换为TRIO的逻辑公式，这样一个并发实时系统的性质即可通过TRIO得以表示和验证。TRIO也是上述框架的一个实例化。即：

系统模型图示	TPN（Timed Petri Nets）
模型逻辑表示	TRIO
映射关系	TPN的TRIO公理化
性质证明器	手工证明

参考文献

1 Shaw M, Garlan D. Software Architecture: Perspectives of an Emerging Discipline. Prentice-Hall, 1996

2 Barroca L, et al. Language and environments for the pragmatic applications of formal methods, second report [JNICT/BC-project report]. Departamento de Informatica da Universidade do Minho, Feb. 1996

3 Felder M, et al. Proving properties of real-time systems through logical specifications and petri net models. IEEE Transactions on Software Engineering, 1994, 20(2): 127～141

4 Jelly J, Gorton I. Software engineering for parallel systems. Information & Software Technology, 1994, 36(7)

5 姚淑珍，金茂忠. 并发软件开发方法学的研究. 计算机科学，1997，24(5)

（上接第70页）

量"信息I_0和伺服模信息I_s的公式，理论和实践一致表明：在非平衡相变点附近，由于"序参量"的组态出现涨落，I_0在相变点存在极大值，而I_s在非平衡相变后略有增加，因此I_0+I_s具有极大值。

4. 若"序参量"形成，转第5步，否则转第2步。

5. 将"序参量"应用于系统，反馈信息后转第1步。

可以看出，"序参量"在"机器智能系统"中，是一个永远工作的程序，类似于操作系统的作用，如果说操作系统架起了计算机硬件和软件之间的桥梁，那么，"序参量"则架起了微观和宏观之间的桥梁。

关于"序参量"工作原理我们再作几点说明：第一，该过程的5步不能作为算法，因为它违反了算法的有穷性；第二，伺服原理的中心内容是绝热消去原理。设系统的n个状态为$q(x,t)=(q_1,\cdots,q_j,\cdots,q_n)^T$，演化方程形式如下：

$$\dot{q}=L(q,a)+N(q)+F(q)$$

其中，$\dot{q}=dq/dt$，a为一组控制参量，L，N，F分别为线性项、非线性项、涨落力。当系统进入相变点附近时，状态变量明显地分为两类，慢施豫变量q_s和快施豫变量q_f，q_s支配q_f或者说q_f伺服于q_s，演化归结为金兹堡-朗道方程：

$$\dot{q}_s=a_sq_s-tq_s^3$$

第三，MIP旨在各种可能微观组态信息中求其极大值，从而确定系统转变后的"序参量"及分布函数。所用方法类似于统计物理学中求熵函数条件极值的拉格朗日乘子法。约束条件是"序参量"的二和四阶矩阵，可通过试验测定，而不需用微分方程。

结论 从上面讨论可知"序参量"具有Agent的本质属性，所以可以认为Agent是系统从无序到有序的序参量。一个智能系统的"序参量"一开始不一定表现得特别明显，也不需要具有惊人的效率，而应当在环境中逐渐适应，最后达到一种秩序，随着环境的改变所达到的这种秩序还会被另一种秩序代替。

另外"序参量"不能在所讨论的可能世界找到，就象严格的数学中也有一些题目是本系统内不能判定一样，从这个意义上讲，我们面临的任务就是找到"序参量"的等级序列。

参考文献

1 Poggi A, Fox G A. Multi Language Environments to Develop Multi Agent Applications. Lecture Note in Artificial Intelligence 1193 ECAI'96 Workshop, 1996 325～340

2 Oragani A S, Giorgini P. Multi-Agent Environment, Lecture Note in Artificial Intelligence 1193 ECAI'96 Workshop, 1996. 103～116

3 Luck M, et al. From Agent Theory to Agent Construction. A Case Study. Lecture Note in Artificial Intelligence 1193 ECAI'96 Workshop, 1996. 50～60

4 Etzioni O, Weld D W. A Softbot-Based Interface to the Internet. Communications of the ACM, 1994, (37)7: 72～79

5 程显毅. 序参量在智能系统中的作用. 江苏理工大学，1997，17(5)

6 H. 哈肯. 协同学. 上海：上海科学普及出版社，1988