嵌入式实时系统的软件需求检测*

毋国庆, 朱立松, 王 敏, 蔡持峰, 陈伟清

(武汉大学 软件工程国家重点实验室,湖北 武汉 430072);
(武汉大学 计算机科学系,湖北 武汉 430072)
E-mail: wgq@whu.edu.cn
http://www.whu.edu.cn

摘要: 以需求描述模型 HRFSM(hierarchical finite state machines based on rules)为基础,提出了一个嵌入式实时系统软件的动态执行模型(dynamic execution model,简称 DEM)和基于该模型的检测方法.由于 DEM 能将控制流、数据流和时间有效地集成为一体,故提出的检测方法能检测嵌入式实时系统的软件需求的一致性和完全性.该检测方法由 3 种侧重点不同的检测形式组成,并能在检测过程中提供一些重要的检测信息.分析员可以利用基于该检测方法的工具灵活地对嵌入式实时系统的软件需求进行检测,以提高分析和检测软件需求的效率.

关 键 词: 嵌入式实时系统;软件需求;需求规格说明;动态执行模型;构图
中图法分类号: TP311 文献标识码: A

软件开发的需求阶段首先是了解和澄清用户的需求,然后严格地定义被开发的软件系统的需求规格说明.对于软件开发来说,如果在需求规格说明中出现错误,将会增加软件的开发成本和开发时间或导致开发工作失败.为了减少需求规格说明中的错误,在需求阶段对其进行有效的检测是一项重要而又必不可少的工作.

与其他类型的软件系统(如管理信息系统)不同,嵌入式实时系统软件(以下简称实时系统)的主要特征是复杂的动态行为、时间限制和与外部环境的通信.因此,描述和检测实时系统的需求是十分困难的工作,特别是检测工作,因为它必须在需求阶段,且实时系统的大部分功能均未实现的情况下检测实时系统的需求规格说明.目前有关检测实时系统需求规格说明的方法已有不少,例如基于 SCR(software cost reduction)和基于 RSML(requirements state machine language)的检测方法[1~5]等.综合现有的研究,检测实时系统的需求规格说明的途径主要有:(1) 人工方式,(2) 形式化验证,(3) 模拟执行.这 3 条途径各具特长,而且有时需视具体情况而定.对于途径(1),其对于小而简单的系统来说是有效的,但对于大而复杂的系统来说,这种方法既耗时,且效果也不一定好.对于途径(2),具有代表性的是基于代数方法和基于特定模型的检测方法,这些形式化方法在严密性方面非常好,并且也无歧义性和模糊性,但其对使用者的要求很高.此外,这些方法有时需针对具体情况,因此它的通用性较差.途径(3)类似于原型化方法,其与原型化方法的区别在于,它用于描述需求的需求描述语言或模型是可执行的,以致在软件开发的初期就能动态地模拟实时系统的执行.途径(3)的特点是能帮助分析员找出规格说明中存在的错误,并协调参与开发工作的各种人员间的认识和选择最佳设计方案.因此,对于复杂的实时系统来说,这种方法相对来说是比较实用而又灵活的方法.在途径(3)中,用于定义需求的可执行的需求描述语言或模型是较严格的形式化语言,并且也是一种特殊的程序设计语言,作为这些语言的代表主要有 Statecharts[6], PAISLey[7],ASLAN[8],UML[9],RSML[10]和 ASTRAL[11]等,它们都不同程度地为检测需求规格说明发挥了较大的

* 收稿日期: 2000-05-23; 修改日期: 2001-03-01
基金项目: 国家自然科学基金资助项目(69873035);国家教育部博士点基金资助项目
作者简介: 毋国庆(1954－),男,湖北咸宁人,教授,博士生导师,主要研究领域为软件理论,需求工程;朱立松(1975－),男,湖北洪湖人,博士生,主要研究领域为软件工程;王敏(1978－),女,湖北天门人,硕士生,主要研究领域为软件工程;蔡持峰(1977－),男,湖北潜江人,博士生,主要研究领域为需求工程;陈伟清(1966－),男,广东揭阳人,讲师,主要研究领域为软件工程.

作用,并且有的也成功或正在用于实际的实时系统的开发中.此外,许多基于这些语言的检测方法和工具也正在不断地被提出或实现[12~16].然而,在基于途径(3)的现有的检测方法中也存在着一些问题:① 某些需求描述语言或模型的语法和语义过于与传统的程序设计语言接近,因而缺乏高度的抽象能力.② 除 Statecharts,SCR 和 RSML 等以外,某些需求描述语言或模型过于复杂,从而影响了需求规格说明的易理解性和可审查性.③ 在检测方法中只强调了对实时系统的动态行为进行检测,避开了对实时系统的主要特征之一的时间限制和数据流的检测.

本文主要围绕途径(3)来研究如何检测实时系统的需求规格说明.我们以需求描述模型 HRFSM[17]为基础,提出了一种分析和检测需求规格说明的方法,其中 HRFSM 是我们提出的一种新的需求描述语言,其有关内容详见第 1 节.我们的检测方法是建立于抽象的实时系统的动态执行模型之上的.由于该模型能将控制流和数据流等有机地结合到一起,故使得这个检测方法能用于检测实时系统软件需求规格说明的一致性和完全性等.此外,这个检测方法也为分析员提供了一些有用的检测信息,从而能有助于分析员提高分析和检测需求规格说明的效率.

1 需求描述模型 HRFSM

我们提出的需求描述模型 HRFSM(hierarchical finite state machines based on rules)是基于层次式状态机的.与其他研究不同,这个模型把层次式状态机表示成规则和模板的形式,其中一个模板对应于一个状态机,一条规则对应于状态机中的一个转换(由图形编辑工具自动实现).由于与状态机相关的规则集和信息可被写入到模板中,故用此模型写出的需求规格说明书可由一组模板组成,而且易于理解和阅读.下面我们将简单地介绍模型 HRFSM.

定义 **1.1**. HRFSM 是一个多元式$(S,E,Att,F,P,s_0,\leq,\|)$,其中

S:状态的集合,且 $s_0\in S$ 是启始状态,整个系统只能有唯一的启始状态.

E:事件的集合.$E=E_1\cup E_2$,E_1 是外部刺激事件的集合;E_2 是系统内部事件的集合.

Att:事件属性的集合.$\forall e\in E(att(e)\subseteq Att)$,$att(e)$是事件 e 的属性集合.

F:根据系统内部定义的变元值和事件的属性值进行计算的公式和函数的集合.

P:状态转换规则的集合,规则 $p\in P$ 的形式为

$$P{:}s_r,e(att(e))\Rightarrow s_{r+1}\ \text{When}\ Cond\ \text{Do}\ EF,$$

其中 $s_r,s_{r+1}\in S,e\in E,att(e)$可为空.$Cond$ 为布尔表达式,其值为 True 或 False.$EF\subseteq F$ 对应于一组计算公式和函数.规则 p 的含义为:当 s_r 是当前状态时,如果事件 e 发生且卫士条件 $Cond$ 被满足,则 EF 进行计算,然后 s_{r+1} 成为下一当前状态.

$\leq$:状态集合 S 上的偏序关系.该关系能把 S 中的状态定义为层次关系,从而可把 S 表示成具有根节点的树型结构.令 $x,y\in S$, $x\leq y$ 意指 y 是 x 的上层状态或 y 是 x 的同层状态.

$\|$:状态集合 $S-\{s_0\}$上的并行关系.其满足如下性质:令 z 的子节点集合为 Sub(z)

$$\forall x,y\in S-\{s_0\}\exists z(x\|y\rightarrow x,y\in \mathrm{Sub}(z)).$$

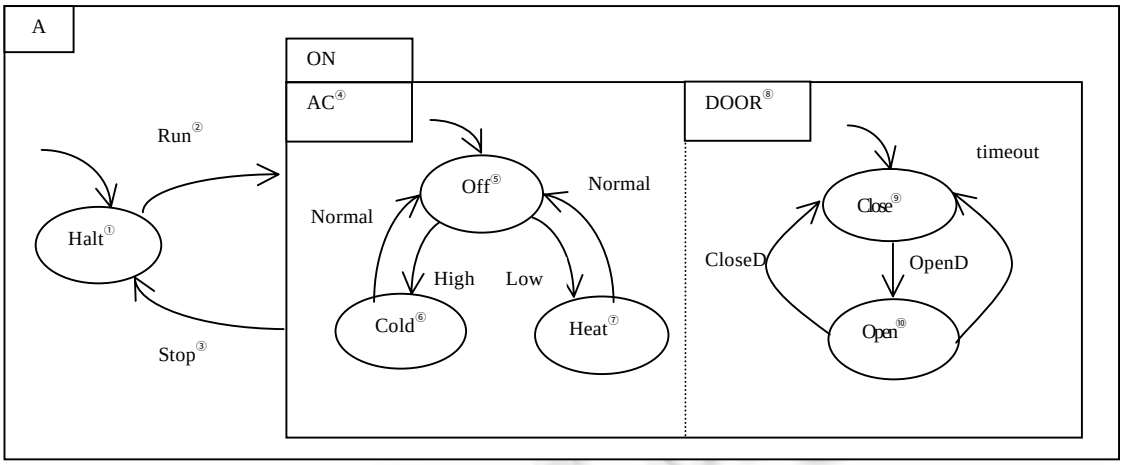
为表达关系“$\leq$”和“$\|$”,我们可用如下两个规则来表示上层状态的分解和结束分解:

分解规则:$s_i,e(att(e))\Rightarrow(s_{i1}\|\ldots\|s_{ir})$ When $Cond$ Do EF,其中 $s_{ik}\in\delta(s_i),1\leq k\leq r$.

结束分解规则:$(s_{i1}\|\ldots\|s_{ir}),e(att(e))\Rightarrow s_i$ When $Cond$ Do EF.

模板被表示成表格的形式,表格中的每栏分别记载与某状态机(或子系统)相关的信息,如状态机名(模板名)、父状态机名、输入事件的说明、变元与属性说明、性能与约束条件的说明、规则集和功能说明等.用户可根据自己的需要填写模板内容.由于用规则集来表示状态机,因而可通过规则的执行来描述系统的动态行为.限于篇幅,本文将只通过一个简例“室温控制系统”来说明如何使用 HRFSM,并为后面的讨论做些准备.

例 1:室温控制系统自动控制室温装置来调节室内温度,其也受房门“开”或“关”的影响,当房门打开并超过系统规定的时间后,系统将停止空调工作.有关系统的状态图如图 1 所示.



①停机,②运行,③停止,④空调,⑤关机,⑥制冷,⑦加热,⑧门,⑨关闭,⑩打开.

Fig.1 State chart of auto temperature-control system

图 1 室温自动控制系统的状态图

与室温自动控制系统的状态图相关的部分规则可表示如下:

Halt,Run⇒ON, /* Halt,ON 分别表示系统处于停止和运行状态 (1)

ON,∅⇒Off‖Close When 房门已关闭 /* Off:空调处于等待启动状态 (2)

Off,Low(T)⇒Heat When T<正常室温 Do Start(0) /* 0:空调制热 (3)

Off,High(T)⇒Cold When T>正常室温 Do Start(1) /* 1:空调制冷 (4)

Heat,Normal(T)⇒Off When T=正常室温 Do Stop() /* 停止空调工作 (5)

Cold,Normal(T)⇒Off When T=正常室温 Do Stop() (6)

ON∨Off∨Cold∨Heat,Stop⇒Halt (7)

… /* Close,Open 分别表示房门处于打开和关闭状态.

/* Cold,Heat 分别表示空调处于正在加热和制冷状态.

限于篇幅,我们在此省略模板的具体表示.

2 基于控制流和数据流的动态执行模型

如前所述,需求描述模型 HRFSM 通过关系“≤”和关系“‖”来描述状态层次和并行关系,从而可把对复杂系统的描述变为对若干较为简单的子系统的描述.就单状态机而言,由于其只允许在某一时刻有一个当前状态,故用单状态机描述的某一时刻的系统状态实际上可用两个因素给予表示,即状态和数据.前者是系统的抽象表示,后者是系统的实际表示,这样任何一个时刻的系统状态可表示为$(s,v(a_1),\ldots,v(a_n))$,其中 s 表示系统的当前状态,$a_i(1\le i\le n)$为变元和属性名,$v(a_i)$表示在当前状态 s 下 a_i 的值.在 HRFSM 中,由于每个并行成分可视为一个独立的顺序执行的状态机,故用 HRFSM 描述的某一时刻的系统状态应有别于单状态机,因为前者允许系统在某一时刻有多个当前状态出现,这就需要我们采用不同的方式来描述复杂的实时系统在某一时刻的系统状态.

定义 **2.1**. 用 HRFSM 描述的复杂系统在某一时刻可同时出现的一组当前状态称为系统的状态构图(以下简称构图),并用符号 C 表示,构图满足如下条件:令 $C=\{s_i|o\le i\le n$,且 s_i 为系统的当前状态之一$\}$

(1) $C\subseteq S,s_0\in C$;

(2) 如果 $s_1,s_2\in$Sub(s),且 $s_1\|s_2$,则 $s,s_1,s_2\in C$;

(3) 如果在某状态机中 s_2 为 s_1 的直接后继,则要么 $s_1\in C$,要么 $s_2\in C$;

(4) 如果 s_2 为 s_1 的缺省状态(如图 1 中的 Halt, Off, Close),则 $s_1,s_2\in C$.

在 HRFSM 中,一条规则与状态机中的一个转换 t 对应,而规则也可表示为 $s_i \xrightarrow{t} s_j$ 的形式,其中 t 可表示为 t =(事件/卫士条件,动作),s_i 为 t 的源状态,s_j 为 t 的终状态.此外,我们分别用 $Source(t)$,$Event(t)$,$Cond(t)$,$EF(t)$来表示与 t 相关的源状态、事件、卫士条件和执行动作的集合.根据 HRFSM 以及转换的概念,我们可定义实时系统的动态执行模型如下:

定义 **2.2**. 实时系统的动态执行模型为 $DEM=(GC,Co,GT,\theta,V,Gd,Gu,\delta)$,其中:

GC:构图的集合;

Co:系统的启始构图,$Co=\{s_0\}\subseteq GC$;

GT:全局转换的集合,且每个全局转换至少应包含一个转换,即 $\forall gt\in GT(gt=\{t_1,\ldots,t_k|k\geq 1\})$;

θ:$GC\rightarrow GT$,函数 θ 使得一个构图能与一个全局转换对应,即 $\theta(C)=gt,C\in GC,gt\in GT$;

V:变元与属性的集合;

GD:$GT\rightarrow 2^V$,其使某全局转换 $gt\in GT$ 与某些变元和属性之间产生联系,而这些属性和变元是 $Cond(gt)$和 $EF(gt)$在执行过程中被重新赋值;

GU:$GT\rightarrow 2^V$,其定义与 GD 相同,不同的是其中的属性和变元是 $Cond(gt)$和 $EF(gt)$在执行过程中仅被使用;

δ:$GC\rightarrow\{time_1,\ldots,time_n\}$,其中 $time_i(1\leq i\leq n)$表示时间戳,可以取计算机系统的当前时间.由于全局状态的变迁具有时序特性,故在某构图成为系统的当前构图时附加一个时间戳,这将有助于与时间限制相关的处理.

定义 **2.3**. 令 C 为某一当前构图,e 为某一事件,$cond$ 为卫士条件,如果 $Source(t)\in C$,且 $Event(t)=e$ 和 $Cond(t)=cond$=True,则称转换 t 为可执行.同理,令 gt 为某一全局转换,如果 gt 中的每个转换 t 是可执行的,则称 gt 为可执行的.

根据定义 2.2 和定义 2.3,我们可定义基于 DEM(dynamic execution model)的一个执行序列.

定义 **2.4**. 基于 DEM 的执行序列是具有无限或有限个构图的序列,即 $Q=C_0,C_1,C_2,\ldots$,其中 C_0 为系统的初始构图.设 gt 为全局转换,当 $C \xrightarrow{gt} C'$为实时系统的某一执行步,并可表示为$(C,gt,gd,gu,time,C')$时,则该执行步的处理应包含:

(1) 令 $gt=(Event(gt)/Cond(t),EF(gt))=(\bigcup_{t\in gt} Event(t)/Cond(t),\bigcup_{t\in gt} EF(t))$,$gt$ 是可执行的;

(2) $C'=(C'\bigcup_{t\in gt} DS(t))\cup(\bigcup_{t\in gt} AS(t))$,其中 $DS(t)$和 $AS(t)$分别意指与转换 t 相关的当前状态和下一当前状态的集合;

(3) $gd=GD(gt)=\bigcup_{t\in gt} GD(t)$;

(4) $gu=GU(gt)=\bigcup_{t\in gt} GU(t)$.

类似于单状态机,由于变元和属性的值与转换的执行相关,故在执行序列中,变元值也在不断变化.动态执行模型 DEM 中的复合函数 $GD\cdot\theta$和 $GU\cdot\theta$基本反映了这一变化.因此,具有并发特性的实时系统在某一时刻的状态也可实际表示为$(C,v(gd\cup gu))$,其中 $v(gd\cup gu)$意指 gd 和 gu 中变元的值.

图 2 表示了例 1 中的部分构图.为便于理解,我们在图 2 中省略了全局转换 gti 中有关 $Cond(gti)$和 $EF(gti)$的表示.从图 2 可知,由一些构图例如 $C_0 \xrightarrow{gt_1} C_1 \xrightarrow{gt_3} C_3\rightarrow\ldots$就可形成实时系统的一条动态执行路径.由于构图实际上是由状态组合而成的,故实时系统的动态执行路径的条数随状态个数的增加也呈指数增长.因此,在状态个数较多的情况下,要检测完实时系统的所有执行路径是不可能的.因此,需要研究一些有效的检测方法.

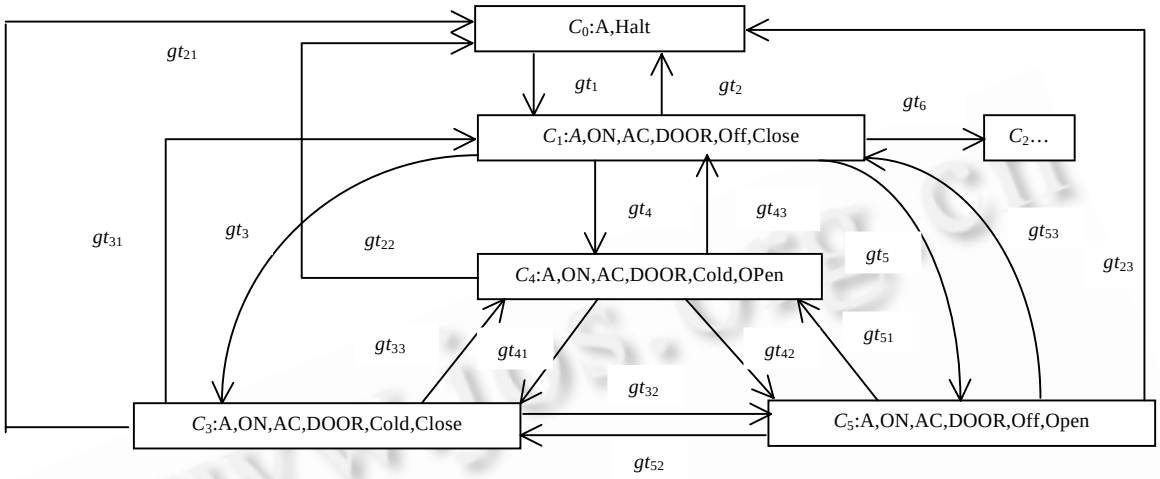
3 检测方法的探讨

本节将首先讨论应该检测哪些内容(即检测什么),然后再提出具体的检测方法(即如何检测).

3.1 检测的内容

在检测需求规格说明之前,一个重要的问题是首先必须清楚需求规格说明使用的是何需求描述模型,然后

才能结合该模型的特点来有效地检测需求规格说明.由于本文使用的需求描述模型是HRFSM,且是基于分层状态机的,因此,该模型的基本要素有 5 个:状态、转换、卫士条件、事件(包括事件所带的属性及属性值)和时间.如果对这 5 个基本要素进行分析和检查,需求规格说明中存在的一些问题就会被检测出来.例如下面一些具体问题就可说明检测这些基本要素的作用.



gt_1=({Run}) gt_2=({Stop})=gt_{21}=gt_{22}=gt_{23} gt_3=({High})

gt_{31}=({Normal∨timeout}) gt_{32}=({OpenD,Normal}) gt_{33}=({OpenD})

gt_4=({OpenD,High}) gt_{41}=({CloseD}) gt_{42}=({Normal})

gt_{43}=({Normal,CloseD}∨{timeout}) gt_5=({OpenD}) gt_{51}=({High})

gt_{52}=({High,CloseD}) gt_6=(…)

…

Fig.2 Part configurations of example 1

图 2 例 1 中的部分构图

(1) 如果某一事件不能引起任何转换或规则执行,或即使执行但产生错误的状态转换,则说明事件或是多余的,或是对该事件的处理有错.

(2) 对于外部刺激事件和系统响应事件所带有的属性和属性值,特别是由外部刺激事件带入到系统的属性值,它们必须在实时系统中被使用或处理,如果没有的话,则说明有些重要的处理被忽略.同理,应随响应事件输出到环境的属性值,如果没有被产生或产生不正确的话,则也说明某些重要的动作被忽略或有错.

(3) 需求不仅涉及到事件以及处理,而且也与状态和转换相关.每一个状态必须至少与一个转换相关,如果状态间存在着路径,则其能由事件和卫士条件唯一确定.所有的状态是可抵达的,如果某一状态是不可抵达的,则有两种可能,一是该状态是多余的,二是可能某些转换被忽略掉.此外,在无并行状态的情况下,如果一个事件同时引起多个规则或转换执行,则状态机会成为非确定的状态机,亦即规格说明中含有歧义和含糊的地方.

(4) 时间限制是实时系统的一个重要特征,由于需求阶段无法实际执行系统的功能,因而检测与时间限制相关的需求是一件困难的工作,故在现行的许多研究中避开了这个问题,并只强调对系统的动态行为进行检测.有关时间限制的检测应包括从外部刺激事件抵达系统到系统发出响应事件的处理时间限制、事件间的间隔时间限制、外部刺激事件抵达系统后等待处理和系统响应事件发出后等待外部环境响应的时间限制等,以及违反时间限制的非正常事件的处理等.

实际上,将以上列举的具体问题综合后可以看出,对这些问题的检测可归结于检测需求规格说明是否满足一致性、完全性和安全性要求[12~14].对于 HRFSM,一致性包含两个方面,一方面是在无并行状态的情况下,任何一个事件的发生不能引起多个转换或规则同时执行,以致在某一时刻状态机中同时出现多个当前状态.另一个方面是在有并行状态的情况下,任何两个转换或规则在执行过程中不允许发生冲突,例如由同一事件引起两个

转换或规则同时要对某一全局变元分别进行修改和使用等.完全性意指对于每一个可能发生的事件以及对事件的响应都必须被明确说明.安全性意指实时系统不但能在规定的时间内处理正常的事件,而且也必须处理非正常的事件(如干扰等).下面,我们将结合 HRFSM 的特点和上述的内容来探讨如何检测基于 HRFSM 的需求规格说明.

3.2 检测方法

虽然 HRFSM 是可执行的,并且我们也试图用模拟执行的方法来检测需求规格说明.但在实际检测中还存在许多问题,例如在实时系统中,除系统的内部事件外,外部刺激事件的发生大部分是随机的,而且有些属性值是由这些事件带入的.此外,规则中的动作部分并不都是可执行的等等.因此在需求规格说明并不能真正地动态执行和不可能对所有动态执行路径进行检测的情况下,我们可通过对一些指定的路径以及构图覆盖的方法对规格说明书进行检测,然后以表格的形式给出某些重要的检测信息,以达到模拟执行的目的.

(1) 单路径单步检测

所谓单路径单步检测意指由分析员指定状态转换的路径,并在每一个转换中提供检测所需的条件,例如根据当前状态提供相关的事件名、属性和变元值,以及可执行的动作(也可暂不考虑提供具体动作).检测系统根据这些信息计算卫士条件,并执行一条规则.在执行到分解规则时,检测系统向分析员发出提示信息,然后由分析员决定下一当前状态.这种检测为分析员重点和仔细了解需求规格说明中的有关部分提供了便利条件,而且它也适合对单状态机进行检测.为便于理解,我们使用了例 1 来说明这种检测,下面来看例 2.

例 2:需检测的问题:当室内温度超过正常室温后,空调装置开始制冷.当室内温度达到正常室温后,空调装置停止工作,但控制系统仍处于运行状态.

假设分析员指定的初始状态为 Off,由图 1 可知,转换路径是 Off→Cold→Off,并且分析员需提供的信息为:

在初始状态 Off:事件名 High,当前的室温 T,函数 Start(1).

在状态 Cold:事件名 Normal,当前的室温 T,函数 Stop().

检测系统执行规则(4)和规则(5).如果分析员不提供可执行的函数或程序,检测系统执行空动作.

(2) 状态可达性检测

状态可达性检测意指由分析员指示启始状态、事件名和终止状态.检测系统把其中的启始状态作为系统的当前状态,并根据事件名找出可执行的规则,然后执行这条规则,直至到达指定的终止状态.在检测过程中,规则中的卫士条件和事件均被假设已确定,且如果规则的动作部分是可执行的话,则执行该动作部分,否则视其为空动作.此外,对于分解和结束分解规则,检测系统在执行完分解规则后,将顺序地执行各并行成分中的规则,最后执行结束分解规则.在检测完毕后,检测系统将提供如表 1 所示的某些重要的检测信息,分析员可根据此表进行分析.例如:

(a) 在无并行状态的情况下,与当前状态和发生的事件对应的规则是唯一可执行的,而且这也可由卫士条件确定.否则,就会出现多个下一当前状态或无下一当前状态.这种情况说明已定义的需求规格说明中存在着不一致,或者规则不对或不充分,导致需求规格说明不完全.

(b) 令 s_i 和 s_i'分别为当前状态和相应的下一当前状态,且 c_i 和 e_i 分别为与 s_i 相关的卫士条件和事件,当 $s_i \wedge c_i \wedge e_i \Rightarrow s_j$ 或 $s_i \wedge c_j \wedge e_i \Rightarrow s_j$ 或 $s_i \wedge c_j \wedge e_i \Rightarrow s_i'$,表明状态机的设计不合理或者变元的定义有误或书写有误等.

(c) 在执行分解规则后,一个当前状态可能对应多个下一当前状态.在有并行状态的情况下,应该说这是正常情况.因此,分析员必须区别有或无并行状态的情况.此外,分析员也可根据基于同一事件的不同当前状态在其对应规则执行的过程中使用和定义变元的情况来检测是否会发生状态冲突等情况.

Table 1 Checking information

表 **1** 检测信息

Current state①	Name of event②	Guard condition③	Next current state④	Used variable⑤	Defined variable⑥
…					…

①当前状态,②事件名,③卫士条件,④下一当前状态,⑤使用的变元,⑥定义的变元.

(3) 构图覆盖检测

当实时系统的动态执行路径过于庞大时,要根据执行路径来检测实时系统的动态行为和时间限制是相当困难的.但我们可用构图覆盖的方法来缓解这一问题,也可达到有效检测的目的.所谓构图覆盖的方法主要是根据构图和执行步的定义,把构图视为动态执行系统的状态,对于每个执行步$(C,gt,gd,gu,time,C')$,可使用如下方法构成新的构图 C'(下一当前构图),令 C 为当前构图.

(a) 如果 C 中不包含并行状态,按定义 2.1 中的条件(3),C 中唯一的当前状态的直接后继状态成为下一当前状态,从而形成新的构图 C'.

(b) 如果 C 中含有并行状态,选择其中某一并行状态,并将其对应的转换作为本执行步的全局转换,使得 C 到 C'的状态变化只与该并行成分的状态变化相关,而其他并行状态在此执行步中暂不变化.当 C'成为当前构图时,在新的执行步中将选择在上一执行步中未发生变化的某一并行状态,通过其状态变化,从而又产生新的下一当前构图.在所有的并行状态依次执行一次后,再从第 1 个并行状态开始重复执行上述过程,直至当前构图中不再含有并行状态.

这种处理虽然是顺序的,但其可覆盖所有的构图以及各种动态执行的情况.此外,通过时间戳,我们可估算出每一执行步或规则中处理动作的执行时间$(time(C')-time)$,然后由累加计算估算出响应一个事件的总处理时间或事件间的间隔时间限制,以及确定事件的等待时间限制是否合理等.例如当用这种检测形式来检测例 1 时,我们可将图 2 中的构图组成一条检测路径 $C_0 \xrightarrow{gt_1} C_1 \xrightarrow{gt_3} C_3 \xrightarrow{gt_{33}} C_4 \xrightarrow{gt_{42}} C_5 \xrightarrow{gt_{53}} C_1 \xrightarrow{gt_6} \ldots$,且这条路径覆盖了图 2 中的所有构图.此外,在每一步检测中,我们还可将某些重要的检测信息填入到表 2 中,以供分析员进行分析.表 2 中的内容表示了上述检测路径中的部分检测信息.

Table 2 Checking information

表 **2** 检测信息

Time stamp①	C	$Event$②(gt)	$Cond$③(gt)	EF④(gt)	gd	gu	C'
$time_1$	C_0	gt_1	The door is closed⑤				C_1
$time_2$	C_1	gt_3	T>Normal room temperature⑥	Start(1)		T	C_3
$time_3$	C_3	gt_{33}	...	...	...	...	C_4
$time_4$	C_4	gt_{42}	T= Normal room temperature	Stop( )		T	C_5
…	…	…	…	…	…	…	…

①时间戳,②事件,③卫士条件,④动作⑤房门已关闭,⑥正常室温.

在检测过程中,构图覆盖也面临两个问题,一是检测路径中构图的顺序问题.对于有并行状态的情况,我们并没有规定并行成分的执行顺序,而是任意地选择并行成分中的某一成分执行.这种情况可能会导致在检测中会出现不合理或不能执行的情况,为了解决这一问题,分析员可以按合理的方式来规定并行成分中的先后执行顺序.二是时间问题,因为当某些转换或规则中没有具体的执行动作时,检测系统视其为空动作,故时间无法估算.对于此问题,分析员可以在不给出执行动作的情况下,自己预分配时间到相应的执行步中,以模拟检测某些时间限制等.

以上,我们提出了一个较为完整的测试方法,而且该方法是以前面提出的模型为基础的.这个方法虽然把 3 种检测作为一个整体,但分析员既可按(1)～(3)的步骤进行,也可按自己的需要来选择某种检测.这个方法对于基于 HRFSM 的需求规格说明来说,应该是一种有效的检测方法.

4 结 语

本文以需求描述模型 HRFSM 为基础,提出了将控制流、数据流和时间集成一体的动态执行模型 DEM.作为探讨和研究,我们根据动态执行模型又提出了通过模拟执行来检测基于 HRFSM 的需求规格说明的方法.这个检测方法具有如下几个特点:

(1) 由于该方法是基于模型 HRFSM 和模型 DEM 的,因此具有较好的理论基础.

(2) 在该方法中,由于各种检测的任务和侧重点不同,故使得该方法具有较好的灵活性,从而使该方法具有

较好的实用性.

(3) 该方法可以较好地支持有关时间限制的检测,因而弥补了目前研究中存在的不足.

(4) 使用该方法进行检测的过程,可以通过执行规则来实现,这使检测系统具有部分推理的功能.

(5) 提供的检测信息比较全面和丰富,且易于理解和分析,能为分析员提供较大的方便.

(6) 该方法可与软件部件化研究相结合,例如可用软件部件来实现规则中的执行动作部分.

目前,我们已经开发出基于 HRFSM 模型的需求规格说明书的自动生成工具的原型,而且也准备在该工具系统的基础上研制检测工具系统,以使需求规格说明书的生成与需求规格说明的检测能成为一个整体.由于有关实时系统的需求分析与检测这一研究难度很大,因此在研制过程中我们可能还会面临许多问题,但我们相信本文的研究内容是有意义的,并且我们也打算在此检测方法的基础上进一步探讨如何完善和精化模型 DEM 和检测方法,以及可视化地模拟执行实时系统的方法.

References:

[1] Heitmeyer, C.L., Jeffords, R.D., Labaw, B.G. Automated consistency checking of requirements specifications. ACM Transactions on Software Engineering and Methodology, 1996,5(3):231~261.

[2] Atlee, J.M., Gannon, J. State-Based model checking of event-driven system requirements. IEEE Transactions on Software Engineering, 1993,19(1):25~39.

[3] Wu Guo-qing, Liu Xiang, Ying Shi. Automated analysis of the SCR-style requirements specifications. Journal of Computer Science and Technology, 1999,14(4):401~407.

[4] Heimdahl, M.P.E., Levenson, N.G. Completeness and consistency in hierarchical state-based requirements. IEEE Transactions on Software Engineering, 1996,22(6):363~377.

[5] Jaffe, M.S., Levenson, N.G., Heimdahl, M.P.E., *et al*. Software requirements analysis for real-time process-control systems. IEEE Transactions on Software Engineering, 1991,17(3):241~258.

[6] Statecharts, D.H. A visual formalism for complex systems. Science of Computer Programming, 1987,8(3):231~274.

[7] Zave, P. An insider's evaluation of PAISLey. IEEE Transactions on Software Engineering, 1991,17(3):212~225.

[8] Auernheimer, B., Kemmerer, R.A. RT-ASLAN: a specification language for real-time systems. IEEE Transactions on Software Engineering, 1986,12(9):879~889.

[9] Booch, G, Rumbaugh, J., Jacobson, I. The Unified Modeling Language User Guide. MA: Addison-Wesley, 1999.

[10] Leveson, N.G., Heimdahl, M.P.E., Hildreth, H., *et al*. Requirements specification for process control systems. IEEE Transactions on Software Engineering, 1994,20(9):689~707.

[11] Porisini, A.C., Ghezzi, C., Kemmerer, R.A. Specification of real-time system using ASTRAL. IEEE Transactions on Software Engineering, 1997,23(9):572~598.

[12] Gargantini, A., Heitmeyer, C. Using model checking to generate tests from requirements specifications. Software Engineering Notes, 1999,24(6):147~162.

[13] Thompson, J.M., Heimdahl, M.P.E., Miller, S.P. Specification-Based prototyping for embedded systems. Software Engineering Notes, 1999,24(6):163~179.

[14] Chan, W., Andson, R.J., Beame, P., *et al*. Improveing efficiency of symbolic model checking for state-based system requirements. Software Engineering Notes, 1998,23(2):102~111.

[15] Dutertre, B., Stavridou, V. Formal requirements analysis of an avoinics control system. IEEE Transactions on Software Engineering, 1997,23(5):267~277.

[16] Kim, Y.G., Hong, H.S., Bae, D.H., *et al*. Test cases generation from UML state diagrams. IEE Proceedings of Software, 1999,146(4):187~192.

[17] Wu Guo-qing, Xiao Hai-feng, Zheng Pen, *et al*. Specifying requirements of real-time system with rules and templates. Wuhan University Journal of Natural Sciences, 2000,5(3):278~284.

Software Requirements Checking of Embedded Real-Time Systems*

WU Guo-qing, ZHU Li-song, WANG Min, CAI Chi-feng, CHEN Wei-qing

(*State Key Laboratory of Computer Software Engineering, Wuhan University, Wuhan* 430072, *China*);

(*Department of Computer Science, Wuhan University, Wuhan* 430072, *China*)

E-mail: wgq@whu.edu.cn

http://www.whu.edu.cn

Abstract: On the basis of requirements description model HRFSM (hierarchical finite state machines based on rules), a DEM(dynamic execution model) of embedded real-time systems software and a checking method based on DEM are presented in this paper. Because DEM can integrate with control flow, data flow and time, the checking method can check consistency and completeness of software requirements for embedded real-time systems software. The checking method consists of three forms that checking purposes are different and can provide some important checking information in the checking procedure. Analysts can check software requirements of embedded real-time systems software effectively by using checking tools based on the proposed method so that efficiency of analyzing and checking software requirements can be improved.

Key words: embedded real-time system; software requirement; requirement specification; dynamic execution model; configuration

* Received May 23, 2000; accepted March 1, 2001

Supported by the National Natural Science Foundation of China under Grant No.69873035; the National Research Foundation for the Doctoral Program of Higher Education of China

++

首届计算机图形学与空间信息系统应用 2002 年国际学术会议通知

由中国国家自然科学基金委员会、中国科学院地理科学与资源研究所、北京大学、浙江大学、国家遥感中心农业应用部、中国农学会计算机农业应用分会、中国自动化学会计算机图形学及辅助设计专业委员会联合组织“首届计算机图形学与空间信息系统应用”2002 年国际学术会议将于 2002 年 8 月 6 日~9 日在北京召开.本次会议是首次技术学科与应用学科同台交流,是一个多学科综合交流的国际学术会议.其目的是让应用学科走技术学科研究成果的捷径,跨越前沿,高水平的发展.会议将评选优秀论文在相关国家一级刊物发表.并同时设有会议成果展览.

详细情况请查看会议网页:http://www.cgconference2002.com.

联系方式:

陈宝雯: 电话: 010–64889810;010-64877339 传真: 010–64854230

E-mail: bwchen@cgconference2002.com; bwchen@cern.ac.cn

丛升日: 北京大学计算机系(100871)

电话: 010–2754939 传真: 010–2754939

E-mail: srcong@263.net