

任务图模型中实时任务设计*

桂先洲¹, 戴金海¹, 周兴铭², 冯进国¹

¹(国防科学技术大学 航天学院,湖南 长沙 410073);

²(国防科学技术大学 计算机学院,湖南 长沙 410073)

E-mail: guixz@263.net

http://www.nudt.edu.cn

摘要: 仿真技术的进步推动了实时系统(如核反应堆、军事战场和 fly-by-wire 系统)的深入研究.用仿真方法解决实时问题,必须首先给出实时系统描述模型.TGM(task graph model)模型是实时系统描述模型之一,其模型最重要的成员是实时任务.使用量化仿真方法定义了实时任务的 4 种参数,并细分任务为子任务.此方法更易描述实时应用系统,更易构造 TGM 的执行模型,其层次化的结构更容易进行研究和软件实现.TGM 模型及其任务设计已成功地应用于分布式三星实时侦察仿真系统.

关键词: 实时任务;时态参数;功能参数;连接参数;资源需求参数

中图法分类号: TP391 文献标识码: A

经过几十年的研究和发展,实时系统(如核反应堆、军事战场和 fly-by-wire 系统)已经成为一门学科^[1].为了描述实时系统的时间约束、可靠性和环境交互问题,许多科技工作者提出了多种形式化方法,对实时系统进行规范和验证.从多年的研发情况来看,形式化方法只能解决“小玩具”的问题^[1,2].

仿真技术的进步推动了实时系统的研究.美国多所大学的教授,如: The University of Texas at Austin 的 A.K. Mok 教授和 University of Illinois at Urbana-Champaign 的 Jane W.S. Liu 教授对此进行的研究.使用仿真方法确定实时系统的行为主要有 Mok 教授的 Modechart^[3,4]和 Jane W.S. Liu 教授推出的 PERTS(prototyping environment for real-time systems)系统^[5].

用仿真方法解决实时问题,必须解决实时系统描述模型问题,即实时系统的建模问题.TGM(task graph model)模型^[6]就是实时系统描述模型之一.TGM 模型的发展离不开 Mok 教授、Jane 教授、Gupta 教授、Grimshaw 博士、Marlowe 教授,特别是 Silberman 博士所做出的贡献^[6-9].文献[10]进一步研究和开发了 TGM 模型,给出了 TGM 模型的形式化定义.

TGM 模型得到了应用,如:PERTS 系统的规范和分析、ObjectTime 系统的规范和分析^[11].文献[12]成功地将 TGM 模型引入到实时仿真系统参考模型之中.TGM 模型与 Modechart 模式图、时间 Petri 网^[13]相比,有以下特征:(1) TGM 模型更容易描述实时应用系统;(2) 更容易构造 TGM 的执行模型;(3) 层次化的结构更容易进行研究和软件实现.TGM 模型中最为重要的成员是实时任务.

TGM 的理论基础是描述实时应用系统的 3 个工作负载模型:周期任务模型(periodic-task model)^[14]、复杂作业模型(complex-job model)和不精确计算模型(imprecise computation model)^[15].

* 收稿日期: 2000-07-27; 修改日期: 2001-03-09

基金项目: 国家自然科学基金资助项目(60073003);国家“九五”国防预研基金资助项目(Z99-04);国家“十五”国防预研资助项目(41320070104)

作者简介: 桂先洲(1961 -),男,湖北麻城人,博士,副教授,主要研究领域为实时系统,计算机建模/仿真,仿真理论基础,仿真软件架构;戴金海(1943 -),男,河北秦皇岛人,教授,博士生导师,主要研究领域为系统建模与仿真,虚拟飞行试验,并行工程;周兴铭(1938 -),男,上海人,教授,博士生导师,中国科学院院士,主要研究领域为高性能计算机体系结构,并行与分布处理,分布移动协同工作;冯进国(1970 -),男,贵州铜仁人,讲师,主要研究领域为飞行力学,发射工程.

定义 1. TGM 模型用三元组有向图形式定义. $G=(V,E,f)$, 其中, V 是任务组成的节点集, E 是连接任务的边集. $V = \{\tau_1, \dots, \tau_n\} (n \geq 1, n \in N)$, τ_i 是任务. $E = \{e_1, \dots, e_m\} (m \geq 1, m \in N)$, $E \subset V \times V$, $e_k = \langle \tau_i, \tau_j \rangle$, $e_k \in E, \tau_i, \tau_j \in V, k=1, \dots, m, i, j=1, \dots, n$ 满足 $\exists f: V \times V \rightarrow E$ 成立.

任务图中边是有向的并且有 3 大属性: 依赖类型、数据容量、时态分隔. 它们表示数据、时态和控制依赖关系. 任务图中任务有 4 大属性: 时态参数、功能参数、连接参数和资源需求参数.

定义 2. 广义实时任务是元组 (Φ, P) , 其中: Φ 是任务执行时间数组 (Φ^1, Φ^2, \dots) , P 是最小时间间隔(其真正单位可为秒或毫秒), 也就是说, 两个相邻帧的就绪时间至少有 P 个时间单位. 任务的第 i 次连续执行时间之和不大干 Φ^i . 每一帧的死限是此帧就绪时间后的 P 个时间单位^[16].

定理 1. 实时任务 (Φ, P) 和任意整数 m 和 n , 设 $n = p^*m + q, 0 \leq q < m$, 第 n 次连续执行时间最长不大于 $p^* \Phi^m + \Phi^q$, 还假定 $q=0$, 则 $\Phi^q=0$. 最为典型的实时任务是周期任务和零星任务.

定义 3. 零星(sporadic)任务与周期任务相似. 零星任务的周期是一个变化的范围, 也就是说, 零星任务在相邻两次变成为可执行的时间间隔是在给定的区间 $(a, b) (b \geq a > 0)$ 之内, 它的周期服从某种概率分布, 例如, 均匀分布、指数分布等.

在实际系统中, 任务既可以是硬件操作, 也可以是外部传感器、作动器、用户等实体所做工作的抽象. 任务是计算颗粒、数据传送单元. 任务既可以是航空飞行操作, 也可以是原子反应堆中的原子反应. 实例是任务的一次动态执行.

Jane W.S. Liu 教授在文献[5]中对任务作了隐形限制: 任务的所有通信传递都在任务执行的开始和结束时进行. 其目的是简化任务的复杂性、减少研究的难度, 这为任务的划分提供了一种基本的参考准则. 然而, 任务使用输入数据和传递计算结果并不贯穿于任务执行的整个过程, 而是在某些子任务上消耗数据、在某些点上产生输出数据.

定义 2 和定理 1 只是概要地强调了实时任务的时间问题. 实际上, 客观存在的实时任务表现为多种行为, 我们必须从不同的视角对其进行描述. 如何定性、定量地描述这些行为是十分重要的. 一方面, 实时任务最为重要的行为是在给定的死限内完成自己的活动, 这种死限和活动表现了它的时态性和功能性; 另一方面, 实时任务是描述系统的基本单位, 任务之间必然存在优先关系, 这种关系表现了任务的连接性; 再一方面, 任务的执行需要一定的实施条件, 表现了资源的需求性.

我们将任务颗粒细分为子任务, 使用 4 种参数定义任务, 它们是时态参数、功能参数、连接参数和资源需求参数. 时态参数定义任务的时间相关行为; 功能参数描述任务的内部性质; 连接参数表明任务被连接的方式和类型; 资源需求参数规定任务使用资源的数目和类型.

1 任务的时态参数设计

定义 4. 反映实时任务的时间行为的指标为时态参数, 它是系统实时调度和资源分配的基本依据. 它由以下参数组成: 最大执行时间(max_exec_time)、最小执行时间(min_exec_time)、平均执行时间(aver_exec_time)、就绪时间(ready_time)、死限(deadline)、周期(period)、相位(phase).

这些参数的形式化定义可见文献[10]. 时态参数和缺省值为

$$(\max_exec_time = \text{infinite}, \min_exec_time = 0, \text{aver_exec_time} = \max_exec_time, \\ \text{ready_time} = 0, \text{deadline} = \text{infinite}, \text{period} = \text{infinite}, \text{phase} = t).$$

平均执行时间(aver_exec_time)的值可用执行时间的概率分布给出. 就绪时间(ready_time)是任务可以开始执行的最早时刻. 就绪时间可以通过时态分隔或相对就绪时间(rel_ready_time)确定. 一个任务的相对就绪时间 r , 表明此任务的开始执行时间不得早于它所依赖的前驱任务的完成时间和 r 之和. 若任务间的依赖关系是数据依赖, 则此任务只有在数据到达后方可执行. Period 是任务的周期, 它是任务相邻两次就绪时间的差值. 相位(phase)为 m 时间单位, 表示第 1 个周期的启动时间为 m . 相对死限(rel_deadline)相对于任务的就绪时间. 绝对死限等于其就绪时间加上相对死限, 绝对死限可简称为死限. 当任务仅仅依赖于其前驱任务的子任务时, 此任务在这个子任务完成后就可以开始执行, 而不必等到前驱任务全部完成再开始.

例 1:周期任务.某半实物仿真中声纳控制回路有下列时态参数:

$(period=100ms, phase=10ms, rel_deadline=70ms, max_exec_time=30ms).$

任务从系统时钟的 10ms 开始执行,每 100ms 必须执行一次.任务在每个周期的开始变成就绪,它的执行最多花费 30ms,每个周期的前 70ms 内完成执行.所以,任务第 1 次执行的就绪时间为 10ms,而且绝对死限为 80ms;第 2 次执行的就绪时间为 110ms,而且绝对死限为 180ms,等等.

例 2:两个零星任务.第 1 个任务表示以零星方式连续采集和发送数据的传感器,其参数为

$(period=(1400ms,1600ms), rel_deadline=1000ms).$

此任务每次变成就绪的时间间隔在 1400ms~1600ms 之间.操作系统或运行系统的调度器可决定两相邻就绪时间真实区间.它非形式地对应着传感器上的数据必须每隔 $1500\pm 100ms$ 时间传送.

第 2 个任务的时态参数如下:

$(name=discrete_sensor, period=[(1400ms,1600ms)], rel_deadline=1000ms).$

当传感器采集了数据时,任务 `discrete_sensor` 就变成就绪且必须在 1000ms 内执行完.周期中的“[]”表示周期不确定、随机变化,并且数据采集的频率不受系统调度器的控制.所以,任务 `discrete_sensor` 每次变成就绪可以在 1400ms~1600ms 之间.

2 任务的功能参数设计

定义 5. 反映实时系统中任务自身能力的指标为功能参数.它也是系统调度和分配的基本依据,基本上由以下参数组成:名字(name)、权重(weight)、剥夺性(preemptibility)、松弛类型(lax_type)、松弛函数(lax_function)、子任务数目(num_subtask)、子任务(subtasks)、选择区间(op_interval)和周期部分(period_part).这些功能参数的形式化定义可见文献[10].

功能参数定义了每个任务的内部属性,其缺省值为

$(name=no\ default, weight=0, preemptibility=可剥夺, lax_type=BL, lax_function=无,$
 $num_subtasks=1, subtasks=all, op_interval=无, period_part=none)$

在任务图的图形表达中,同一个名字(name)对应多个节点表示同一个任务的不同实例.名字相同的两个节点都有相同的参数值,但边的参数一般不同.权重(weight)表明此任务的执行相对于其他任务的重要程度.权重可视为任务在死限前未完成执行所可能付出的代价,权重值高的攸关(critical)任务的失败,可能导致整个系统的毁坏.

例 3:在飞控系统中,主管控制航空器稳定的任务有非常大的权重,它反映了此任务的极为重要的属性.若此任务不能满足死限,航空器就可能毁坏,相对而言,控制内场飞行场景的记录器的权重则非常低.

剥夺性(preemptibility)反映了一个任务的执行可否被另一个任务取代的属性.当一个任务被剥夺时,它被迫暂时放弃掌握的资源.任务剥夺性的取值为:可剥夺、不可剥夺或原子.可剥夺的任务可以被另一个任务剥夺,它的执行可以挂起和恢复.若不可剥夺任务被强行剥夺,那么它的执行必须重新开始.原子任务的执行是不可中断的.任务的剥夺性不必与其资源的剥夺性相关.

通常,在死限时刻能否完成任务有一个代价的问题.Lax_type 松弛类型从数量上表示超过死限但完成其执行的相对代价.

Lax_type=BN 表示代价相当高.因此,若任务不能满足死限,最好不要调度此任务或流产此任务.高权重 BN 任务不能满足死限将导致系统致命的失败,有 BN 松弛因子的任务的死限被认为是强死限.

例 4:Lax_type=BN 的插销(bolt)任务是机器人控制系统的组装任务,它将小机械(widget)插到小装置(gadget)上.任务有下列参数值:

$(name=bolt, period=1000ms, rel_ready_time=500ms, rel_deadline=650ms,$
 $max_exec_time=100ms, lax_type=BN).$

若 bolt 任务迟到,它将 widget 插到错误的 gadget 上,两者都将被毁坏.若再晚一些,当下一个 gadget 到达时,它可能继续工作,引起组装线的拥挤.传统的处理方法是,当错过死限时,bolt 不必执行,无插销的 gadget 回送到生

产线上.

$Lax_type=BL$ 表示代价相对低,有 BL 松弛因子任务的死限被认为是弱死限. $Lax_function$ 松弛函数表示在任务执行超过死限以后,时间与未完成工作的函数关系,此函数为实时系统的预测性提供数据.

例 5:某型号的雷达屏有数据显示任务.其参数如下:

$$(name=display,period=2000ms,max_exec_time=500ms,aver_exec_time=300ms,rel_deadline=350ms,lax_type=BL,lax_function=1-t/550).$$

通常,display 执行快,偶尔此任务执行将超过死限.在这种情况下,也应让其完成执行.松弛函数表示在死限之后,函数值线性减少到 0.55s 时达到零值,即任务全部完成,从而可正确地进行时间预测.

任务的松弛类型和松弛函数与任务的权重无关.有较高权重的任务可能有 BL 松弛类型,有较低权重的任务可能有 BN 松弛类型.

根据 Jane 教授的不精确计算模型理论.有些任务有可跳过(或不被执行)的部分,我们称其为选择部分,其他部分称为必要部分(mandatory).我们使用选择区间来描述任务中的选择部分.选择区间由一个三元组构成:

$$(start,finish,frequency).$$

Start 和 finish 表示选择区间任务中的第 1 子任务和最后子任务,frequency 表示一个周期任务或零星任务中选择区间必须完成执行的频率.频率的缺省值为 0,它表示此选择区间不被执行.一个任务可含有多个选择区间,选择区间不能覆盖或嵌套,但可以有条件的.

例 6:含有选择区间的跟踪任务.此跟踪任务确定声纳的接触点,任务的必要部分提供接触点位置的近似值,而选择部分对此近似值进行精确分析.任务有以下参数:

$$(name=tracking,period=1000ms,rel_deadline=1000ms,max_exec_time=infinite,op_intervals=((2,3/6),(5,3/6),(6,last,0))).$$

此任务是周期的,周期为 1 秒.它有 3 个选择部分:第 1 部分由 tracking 的第 2 子任务组成;第 2 部分由 tracking 的第 5 子任务组成;第 3 部分由 tracking 的第 6 子任务到任务最后一个子任务组成.第 1 部分和第 2 部分都必须在连续 6 个周期内至少执行 3 次.第 3 部分永不执行.

3 任务的连接参数设计

定义 6. 反映实时系统中任务之间的优先关系的参数为连接参数.连接参数用于精确规定任务的直接前驱、直接后继的执行情况.连接参数将任务分成 in-type 类型任务和 out-type 类型任务.in-type 类型规定了一个任务执行所需已完成执行的直接前驱的数量.out-type 类型规定了在一个任务完成之后,其直接后继可以开始执行的数量.基本的连接参数是 in-type 类型任务、in-threshold 阈值、in-constraints 约束、out-type 类型、out-threshold 阈值和 out-constraints 约束.

这些参数的形式化定义可见文献[10].边通过连接参数定义任务间的关系.其缺省值为

$$(in_type=and,in_threshold=(and-in 任务=n,or-in 任务=1),in_constraints=none),$$

$$(out_type=and,out_threshold=(and-out 任务=n,or-out 和 conditional 任务=1),out_constraints=none).$$

任务 τ 的 in-type 类型取值为 and 或 or.

第 1,若 $in_type=and$,当 τ 的所有直接前驱完成后, τ 方可执行.我们称 $in_type=and$ 的任务为 and-in 任务.

第 2,若 $in_type=or$,我们称 $in_type=or$ 的任务为 or-in 任务.or-in 任务有一个相应参数 in-threshold,此参数表示任务执行前直接前驱必须完成执行的数量.注意到,每个 and-in 任务的 in-threshold 取值为它直接前驱的数目.若 or-in 任务 τ 有 $in_threshold=n$,当 τ 的 n 个直接前驱完成后, τ 方可执行.

In-constraint 是对 or-in 任务的约束,in-constraint 是布尔表达式,其元素为任务的直接前驱,它表示布尔表达式中的前驱组合必须在此任务执行之前完成.

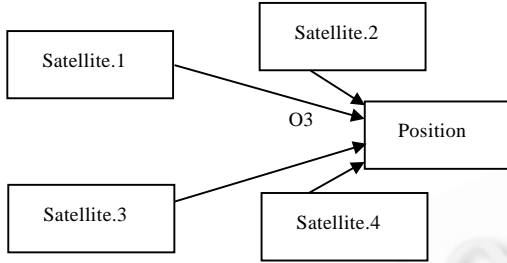
例 7:GPS 定位系统中的 or-in 任务(如图 1 所示).有 5 个任务分别为 Satellite.1,...,Satellite.4 和 Position 任务.Satellite.x 是导航任务.Position 任务使用 Satellite.x 的输出数据来确定车辆的位置.所有的边表示时态和数据依赖.

它有下列参数:(*name=Position,in-type=or,in-threshold=3*).它表示 Position 的执行至少需要 3 个导航任务完成.记号 O3 表示 Position 任务的互连参数:*in-type=or,in-threshold=3*.

任务 τ 的 out-type 有 3 种取值:and,or 或 conditional.

第 1,若 *out-type=and*, τ 的所有后继必须在 τ 完成后才开始执行,我们常称 *out-type=and* 的任务为 and-out 任务.And-out 任务的 out-threshold 等于其出边的个数,out-type 缺省值为 and.

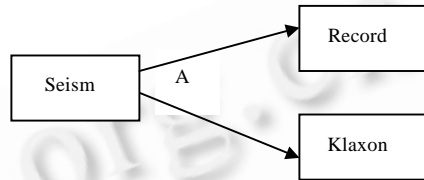
例 8:地震测量中的 And-out 任务(如图 2 所示).任务 Seism 表示地震测量.当地震发生时,任务 Record 和任务 Klaxon 必须都执行,



卫星 1, 卫星 2, 卫星 3, 卫星 4, 定位.

Fig.1 Or-In tasks in GPS navigation systems

图 1 GPS 导航系统中的 Or-in 任务



地震, 记录, 警报.

Fig.2 And-Out tasks in earthquake measurement

图 2 地震测量中 And-out 任务

Record 记录地震的测量数据,Klaxon 发出地震报警.记号 A 表示任务 Seism 的后继任务相继执行.

第 2,若 *out-type=or*,此任务完成后,其某些直接后继需要执行,另外一些直接后继无需执行,我们称此任务为 or-out 任务.Or-out 任务有一个相应参数 out-threshold,它用于表示有多少直接后继必须随此任务的完成后执行.

Out-threshold 的取值可以是整型数或整型数后加符号 e,前者表示直接后继执行的最少数目,后者表示直接后继执行的准确数目,out-threshold 的缺省值为 1.

例 9:作战系统中雷达跟踪任务就是 or-out 任务(如图 3 所示).有以下参数:

(*name=radar,period=1000ms,deadline=500ms,out-type=or,out-threshold=1e*);

(*name=plot,period=1000ms,phase=30ms,deadline=700ms*);

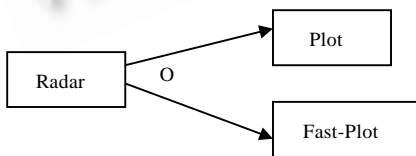
(*name=fast-plot,period=1000ms,phase=500ms,deadline=500ms*).

Radar 任务的跟踪结果需要显示,其死限为 500ms.Plot 较慢,它将雷达屏的数据以精确曲线显示出来,Fast-plot 则可以快速完成不精确曲线的描绘.每次 Radar 任务执行后,Plot 和 Fast-plot 两者之一执行.任务图并不规定选择谁来执行,调度决策依赖于运行系统或操作系统.记号 O 表示 Radar 任务的后继执行两者之一.

第 3,若 *out-type=conditional*,此任务称之为 conditional 任务.Conditional 任务有对应的 out-threshold 参数,out-threshold 参数的值表示直接后继执行的最大数目,若整型数后跟着 e,则表示后继执行的准确数目,out-threshold 的缺省值是 1e.Conditional 任务在任务图中表示条件测试,此任务的后继对应于条件测试分枝.

Out-constraint 是对 or-out 或 conditional 任务的约束,Out-constraint 是布尔表达式,其元素是此任务的直接后继.In-constraint 和 out-constraint 用于规定 in-type 和 out-type 的复杂的依赖关系.

例 10:战斗机的 conditional 警报任务(如图 4 所示).



雷达, 记录, 快速记录.

Fig.3 Or-Out tasks in radar tracing

图 3 雷达跟踪中 or-out 任务



探测, 警告.

Fig.4 Warning tasks of fighter planes

图 4 战斗机的警报任务

Detection 有参数:(*out-type=conditional,out-threshold=1*).它确定本战斗机附近是否有另一个战斗机,若有,

则任务 Warning 发电警笛;否则,detection 的直接后继无需执行.out-type=conditional,out-threshold=1 表明最多只有一个后继执行.

4 任务的资源需求参数设计

定义 7. 反映实时任务在运行环境中所需资源的数量和类型称为资源需求参数.任务的资源需求参数规定了任务使用资源的要求.任务可以请求一个或多个资源,资源需求参数是:类型(type)、单元数(units)和需求子任务(reqed_subtasks).

资源需求参数可以使用三元组来定义:

$(type, units, reqed_subtasks)$.

Reqed_subtasks 可以取值为 all,表示资源在整个执行期间被使用,典型的例子就是计算任务使用处理机资源;units 可以取值为 all,表示需要系统内所有此种类型的资源.资源需求参数和缺省值为

$(type=no\ default, units=1; reqed_subtasks=all)$.

例 11:在战场对抗仿真演练中,一个后勤运输车辆所代表的任务:在某个时刻加满了燃料,它就被调度,并开始执行,当任务执行时,它又将动态获得和释放铁路或公路交叉路口许可证信号量资源.燃料和许可证信号量二者都是资源.

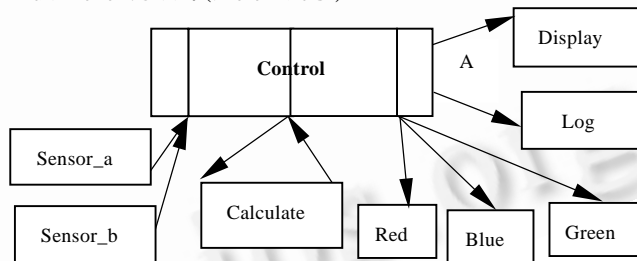
$(type=vehicle, units=1; reqed_subtask=all)$;

$(type=fuel, units=1; reqed_subtask=2)$;

$(type=across_light, units=all; reqed_subtask=7)$.

5 TGM 模型举例

分布式三星实时侦察仿真系统正是基于 TGM 模型的实现系统.它的目的就是模拟分布式三星不间断监视、侦察热点地区.由于此系统建模复杂、工作量大,详细的 TGM 模型和任务可见文献[10].为了说明问题,现给出机械手喷漆过程的 TGM 模型和任务设计(如图 5 所示).



控制, 传感器 a, 传感器 b, 计算, 红色, 蓝色, 绿色, 日志, 显示.

Fig.5 TGM model about the process of robot painting

图 5 机械手喷漆过程的任务图模型

机械手喷漆过程的 TGM 模型描述如下:

Control 是周期任务,它代表喷漆过程控制的主控循环,共分为 4 个子任务.第 1 个子任务完成后,Control 接收传感器 Sensor_x 的输入.它只需接收两个传感器的输入.到第 2 个子任务以后,Control 请求任务 Calculate 执行,并且等待计算结果,按不同的结果采取不同的动作.在第 3 个子任务尾部,Blue,Green 和 Red 这 3 个任务中最多一个被调用.这些任务的工作是调整各自的颜色数量而形成混合色(若颜色是所需的,则 3 个任务中一个也不调用).第 4 个子任务执行后,弱死限任务 Display 和 Log 被调用.分别在操作员的控制台上显示和在日志中记录此循环中系统的行为.

Control 的每一个子任务有下面的连接参数:

$(subtask=2, in_type=and, in_threshold=2, out_type=and, out_threshold=1)$;

$(subtask=3, in_type=and, in_threshold=1e, out_type=conditional, out_threshold=1)$;

(*subtask=4,out-type=and*).

第 1 个子任务没有输入和输出参数,第 4 个子任务没有输入参数.为了节省篇幅,其他的任务参数可参见文献[17].

6 结束语

本文讨论了 TGM 模型的重要成员实时任务的设计问题.我们使用时态参数、功能参数、连接参数和资源需求参数描述实时任务的属性,详细定义了这 4 大属性中各种各样的参数和缺省值.TGM 模型和其任务设计已经成功地应用于分布式三星实时侦察仿真系统.实时任务的设计已经成功地应用于实时仿真系统的参考模型之中^[12].

References:

- [1] Shin, K.G., Ramanathan, Parameswaran. Real-Time computing: a new discipline of computer science and engineering. Proceedings of the IEEE, 1994,82(1):6~24.
- [2] Stankovic, J.A. Misconception about real-time computing: a serious problem for next-generation systems. IEEE Computer, 1988, 21(10):10~19.
- [3] Jahanian, F., Mok, A.K. Modechart: a specification language for real-time systems. IEEE Transactions on Software Engineering, 1994,20(12):933~947.
- [4] Rose, A., Perez, M., Clements, P. Modechart toolset user's guide. Technical Report, NML/MRL/5540-94-7427, Austin: University of Texas at Austin, 1994.
- [5] Liu, J.W.S., Liu, C.L., Deng, Z., *et al.* PERTS: a prototyping environment for real-time systems. International Journal of Software Engineering and Knowledge Engineering, 1996,6(2):111~126.
- [6] Silberman, A., Marlowe, T. A task graph model for design and implementation of real-time system. In: Proceedings of the 2nd IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'96). New York: IEEE Press, 1996. 432~441.
- [7] Mok, A.k., Sutanthavibul, S. Modeling and scheduling of dataflow real-time systems. In: Proceedings of the 1985 Real-Time Systems Symposium. New York: IEEE Press, 1985. 178~187.
- [8] Gupta, R., Spezialetti, M. A compact task graph representation for real-time scheduling. Real-Time Systems, 1996,13(7):13~21.
- [9] Grimshaw, A. Mentat: an object-oriented macro data flow system [Ph.D. Thesis]. Urbana: University of Illinois, 1988.
- [10] Gui, Xian-zhou. Design of the model and programming language for complex real-time systems [Ph.D. Thesis]. Changsha: National University of Defence Technology, 1999 (in Chinese).
- [11] Smith, K.W. ObjectTime CASE tool simplifies real-time software development. Dr. Dobb's Journal, 1997,14(12):34~45.
- [12] Gui, Xian-zhou, Dai, Jin-hai, Zhou, Xing-ming. The reference model for real-time simulation systems. Journal of System Simulation, 2001,13(4):469~472 (in Chinese).
- [13] Berthomieu, B., Diaz, M. Modeling and verification of time dependent systems using time Petri nets. IEEE Transactions on Software Engineering, 1991,17(3):259~273.
- [14] Liu, C.L., Layland, J.W. Scheduling algorithms for multiprogramming in a hard real-time environment. Journal of the ACM, 1973,20(1):46~61.
- [15] Liu, J.W.S., Shin, W.K., Lin, K.J., *et al.* Imprecise computations. Proceedings of the IEEE, 1994,82(1):83~94.
- [16] Mok, A.K., Chen, Deji. A Multifcme model for real-time tasks. In: Proceedings of the IEEE 17th Real-Time System Symposium. New York: IEEE Press, 1996. 78~87.
- [17] Gui, Xian-zhou, Qin Yu-jie. Design of TGM modeling and tasks attribute for robot painting. Technical Report, Changsha: National University of Defence Technology, 1999 (in Chinese).

附中文参考文献:

- [10] 桂先洲. 复杂实时系统模型和语言设计[博士学位论文]. 长沙:国防科学技术大学,1999.
- [12] 桂先洲,戴金海,周兴铭. 实时仿真系统参考模型. 系统仿真学报,2001,13(4):469~472.

[17] 桂先洲,秦玉杰.机械手喷漆的 TGM 建模和任务属性参数设计.科技报告,长沙:国防科学技术大学,1999.

Real-Time Tasks Design in Task Graph Model*

GUI Xian-zhou¹, DAI Jin-hai¹, ZHOU Xing-ming², FENG Jin-guo¹

¹(School of Aerospace, National University of Defence Technology, Changsha 410073, China);

²(School of Computer, National University of Defence Technology, Changsha 410073, China)

E-mail: guixz@263.net

http://www.nudt.edu.cn

Abstract: The advancement of simulation technology promotes the deep research of real-time systems in urgent need such as nuclear reactor, military field and fly-by-wire systems. According to the simulation approaches, the description model of real-time systems must be presented. Task graph model is one of the description models. The real-time task in TGM model is very important. With the use of quantitative simulation method, the task is defined with four kinds of parameters and subtasks combining a task. By using this method, the real-time systems application can be described easily, the execution model of TGM can be constructed quickly, and its software can be researched and implemented quickly. TGM model and tasks design are successfully applied to the real-time simulation system of three surveillance satellites.

Key words: real-time tasks; temporal parameters; functional parameters; connection parameters; resource requirement parameters

* Received July 27, 2000; accepted March 9, 2001

Supported by the National Natural Science Foundation of China under Grant No.60073003; the Defence Pre-Research Project of the 'Ninth Five-Year-Plan' of China under Grant No.Z99-04; the Defence Pre-Research Project of the 'Tenth Five-Year-Plan' of China under Grant No.41320070104

全国第 4 次程序设计语言发展与教学学术会议

征文通知

全国第 4 次程序设计语言发展与教学学术会议定于 2003 年春季在江苏扬州召开.本次会议由东南大学承办,扬州大学、南京大学、武汉大学等院校协办.欢迎大家积极参加并踊跃投稿.

一、征文范围

程序设计语言历史、现状与发展;面向对象语言及相关技术;各类建模语言及其设计、实现与应用;面向网络应用的程序设计语言(XML、HTML、PERL 等);其他各种新型程序设计语言(包括逻辑型语言、函数型语言等);程序设计语言分析、评价与比较;程序设计语言语法、语义与语用以及形式化描述技术与方法;并发、并行与实时程序设计语言;软件开发过程中各类描述语言(包括软件体系结构描述语言等);第四代语言与数据库语言;程序设计语言教学、教材与课件;各类写作语言与工具;其他.

二、征文要求

(1) 来稿一般不得超过 6000 字,并且未被其他会议、期刊录用或发表.为了便于正式出版论文集,来稿必须附有中、英文摘要,关键词和主要参考文献,并注明作者姓名、工作单位、详细通讯地址(包括电子邮件地址和电话)和作者简介.

(2) 欢迎电子投稿,来稿不退,请留底稿.

三、重要日期

征文截止日期:2002 年 10 月 15 日

录用通知发出日期:2002 年 11 月 15 日

修改稿截止日期:2002 年 12 月 15 日

四、联系方式

联系人:东南大学 计算机科学与工程系 徐宝文

电话:025-3793977

邮政编码:210096

E-mail: bwxu@seu.edu.cn