实时嵌入式构件模型组装方法及时间性推理

字天文,刘晓燕,沈嘉权
ZI Tian-wen,LIU Xiao-yan,SHEN Jia-quan

昆明理工大学 信息工程与自动化学院,昆明 650051
School of Information Engineering and Automation,Kunming University of Science and Technology,Kunming 650051,China
E-mail:ztw_2005@163.com

ZI Tian-wen,LIU Xiao-yan,SHEN Jia-quan.Assembly and time reasoning methods of real-time embedded component model.Computer Engineering and Applications,2009,45(25):74-77.

Abstract: This paper presents a better adaptive software component model and component assembly mechanism,and describes component interface specifications and assembly specifications by using the formal methods.Besides,the methods of time reasoning about components are given.The target is to build a more useful,simpler,and more open component model and to build a more reasonable component assembly and reasoning mechanism for real-time embedded system.

Key words: real-time embedded system;software component model;component interface;assembly;time

摘 要:针对实时嵌入式系统特点,提出一种可行的具有较好普适性的软构件模型及构件组装机制,使用形式化方法描述构件接口规约及组装规约,并给出构件时间性的推理方法,旨在实时嵌入式系统中构建更加通用、简易、开放的构件模型和更加合理的构件组装推理机制。

关键词:实时嵌入式;软构件模型;构件接口;组装;时间性
DOI:10.3778/j.issn.1002-8331.2009.25.023 **文章编号**:1002-8331(2009)25-0074-04 **文献标识码**:A **中图分类号**:TP311

1 引言

基于构件的软件工程(Component Based Software Engineering,CBSE)已在通用系统开发中占据重要地位,而由于实时嵌入式系统的复杂特性使其在该领域的应用还尚未成熟。近几年,嵌入式领域构件技术的研究已有了一些成果,典型的如国外的 PBO[1]、Koala[2]以及 PECOS[3]等构件模型,国内的 DRSCDE[4]、CBMESP、Z—CCM 等构件模型,它们虽对 CBSE 开发都提出了有效的解决方案,但大多局限于具体领域,依赖于操作系统及特定平台,缺乏对构件实时性约束的描述,并在构件易用性、复用性、可移植性、可靠性等方面仍存不足,难以做到开放性、普适性。

国外的基于端口的对象 PBO[1]综合了面向对象设计及端口自动机理论,形成了针对基于传感器控制系统开发的框架,特别用于可重新配置的机器人应用中。其缺点是成员方法没有优先级处理机制,模型依赖于实时操作系统 Chimera 且不能移植。Koala[2]是菲利浦公司为消费电子领域软件开发而设计和使用的构件模型,可根据生产线结构进行裁剪,通过显示定义的接口与其他环境组件进行交互,构件对开发者可见且考虑了资源约束及时间的非功能属性。其缺点是未提供对构件测试及调试的显示支持,依赖于具体操作系统及编译器,其建模语言在内部定义和开发,可引入性较差。PECOS[3]是应用于现场总线技术的构件模型,对独立构件、子构件和连接件进行了规范,将构件组装到现场设备,并检查构件组合和现场设备的结构及其非功能属性。其优点是能够处理时间和内存耗费等非功能性属性,提供了体系结构建模和开发工具。其缺点是缺少跨领域协同,没有足够的 CASE 工具支持,且 PECOS 构件是黑盒构件,影响了开放性需求。国内的构件模型 DRSCDE[4]领域协同性较好,提供了构件抽象层、详细层的解决方案,支持实时特征及非功能性描述,但其局限于 C/S 构架,没提供有效的规约推理及可靠性验证机制。

一种统一的构件模型是不适用的,而由于各领域需求有许多共同特征,可以考虑设计一种抽象软构件模型来提高其普适性。鉴于以上现状,采用形式化方法提出了一种具有较好普适性并能够准确描述实时嵌入式系统功能及非功能特性的软构件模型——ESDCM。首先,提出构件模型及模型元素定义并给出接口规约;其次,提出构件组装机制及组装规约;最后,从实时性描述及验证角度给出构件时间性的推理方法。

2 ESDCM 构件模型定义

ESDCM(Embedded Software Developing Component Model)

基金项目:云南省教育厅科学研究基金项目(the Science Foundation of Yunnan Province Education Department under Grant No.07C10799);昆明理工大学人才培养基金(The Training Fund of Kunming University of Science and Technology.No.2008037)。

作者简介:字天文(1984-),男(彝族),硕士生,主要研究方向:实时软件工程及软件开发环境;刘晓燕 (1964-),女,副教授,硕士生导师,主要研究方向:实时软件工程及软件开发环境;沈嘉权(1977-),男,硕士生,主要研究方向:实时系统验证技术。

收稿日期:2009-01-09 **修回日期**:2009-02-09

是一种新的面向实时嵌入式系统开发的抽象软构件模型。该构件模型提供了一套较完整的软构件描述推理机制,支持黑盒组装及白盒组装,它与具体嵌入式操作系统、构件组装平台实现了分离,除具有传统构件的优点外,该抽象模型具有较好普适性、高复用性、易用性及可扩展性。

基于 ESDCM 构件模型进行 CBSE 开发,可在构件制作或白盒组装时即给出整体的构件组装方案,确定构件功能及非功能需求,验证后期构件组装运行的正确性与可靠性,指导整个组装过程的顺利进行。在黑盒组装时,ESDCM 构件内部封装了控制逻辑,对外提供开放性接口,构件功能、非功能包括时间、行为等特性通过接口对外暴露。外界通过推理分析构件接口规约及组装规约,结合组装需求确定适宜的组装方法来完成组装。构件可根据需求进行嵌套以调整粒度,可根据系统架构组装成图形化的插头插座式体系结构风格。

3 ESDCM 构件模型元素

ESDCM 构件模型支持图形化来设计构件和连接件(简称 E 构件、E 连接件),并在服务协议的基础上通过接口完成构件之间的交互及构件组装。接口、E 构件、E 连接件和服务协议是 ESDCM 构件模型的主要元素,其中接口和 E 构件是最基本元素。如图 1 为 ESDCM 构件模型 E 构件的图形表示。

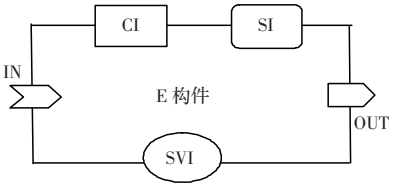

图 1 ESDCM 构件模型 E 构件图形表示

E 构件共有五个图形化接口,分别隶属于三种接口类型:功能接口、非功能接口及行为接口,前两种类型多用于系统架构,而后者多用于组装建模。功能接口描述构件所提供的功能服务;非功能接口描述构件在环境依赖、资源控制、时间性、调度性、聚合性及监控模拟等方面的非功能特性;行为接口刻画构件组装的行为活动。图 1 的五个图形化接口中服务接口 SVI(Service Interface)属于功能接口,状态接口 SI(State Interface)及配置接口 CI(Configuration Interface)属于非功能接口,IN、OUT 接口属于用于组装连接的行为接口。

4 ESDCM 构件模型接口规约

使用类 BNF 范式定义 E 构件功能性及非功能性接口规约,刻画 ESDCM 构件模型对实时嵌入式系统功能特性及非功能特性的描述能力。行为接口规约将结合组装规约于后续章节给出。

4.1 E 构件功能性接口规约

E 构件功能性接口由服务接口构成。SVI 规约描述如下:

SVI=<SA,SO,SP,DS>

SA::=<制作信息,构件粒度,功能信息,应用领域,构件类型,运行信息>

制作信息::=<构件号,构件名,实体类型,开发语言,版本号…

构件粒度::=<E 原子构件>|<E 复合构件>

功能信息::=<功能描述,性能描述>

构件类型::=<分析件>|<设计件>|<代码件>|<测试件>

运行信息::=<硬件环境,软件环境,优先级信息,运行方式>

实体类型::=<可执行文件>|<二进制代码件>|<源代码件>

性能描述::=<时间性能,资源性能…>

运行方式::=<交互式>|<自动式…>

SO::=<操作信息,服务对象(函数)信息,服务约束信息>

SP::=<交互方式,交互协议>

交互方式::=<硬件连接方式,软件连接方式>

硬件连接方式::=<串并口>|<网络接口>|<无线接口>|…

软件连接方式::= <时间驱动>|<信号驱动>|<事件驱动>|…

交互协议::= <组装协议>|<交互原语>|<异常处理协议>|….

DS::=<输入数据>|<输出数据>|<存取数据>|<异常处理数据>|…

SVI 是一个四元组:SVI=<SA,SO,SP,DS>。SA(Service Attribution)表示构件的服务属性集,SO(Service Operation)表示构件的服务操作集,SP(Service Protocol)表示构件的服务交互协议集,DS(Data Specification)表示构件服务交互所使用的数据集。SA 对外体现构件的静态属性,外界以此了解构件的组装及复用信息。SO 对外体现构件的动态属性,外界可获取动态属性提供的操作方法从而实现对该构件的调用。DS 是对接口服务所使用的数据集合进行的说明,其对构件可存取数据进行设置,并指定流入和流出服务接口的数据流。

4.2 E 构件非功能性接口规约

E 构件非功能性接口包括配置接口和状态接口。

4.2.1 配置接口规约

配置接口 CI 支持静态及动态配置,CI 规约描述如下:

CI=<EC,RC,PC,AC>

EC::=<平台配置,环境配置,构件库配置,检索配置…>

RC::=<DN,DC>

DN::=<传感器,,激励器,逻辑芯片,电路板,计数器…>

DC::=<内存资源配置,读数据,写数据,设置设备状态…>

PC::=<SN,TC,SPC>

SN::=<命名配置,重命名配置>

TC::=<TTsn,ETsn,STsn,TPsn,TDsn>

TTsn::=<周期性>|<偶发性>|<混合性>

SPC::=<先来先服务,抢占式,时间片轮转,事件驱动,中断…>

AC::=<聚合关系,同步异步关系>

聚合关系::=<顺序>|<并行>|<选择>|<复制>|<中断>|…

同步异步关系::=<同步>|<异步>|<同步异步混合>

CI 是一个四元组:CI=<EC,RC,PC,AC>。EC(Environment Configuration)为环境配置接口,RC(Resource Configuration)为资源配置接口;PC(Performance Configuration)为性能配置接口;AC(Aggregation Configuration)为聚合关系配置接口。

RC 定义对被 E 构件监控或处理的外部实体或嵌入式硬件设备的配置。实时系统控制的外部设备的逻辑功能集成于构件服务集,RC 提供这些设备功能性的外部配置。受控硬件设备也是 CI 的外部图形化抽象。RC 是一个二元组集合:RC={<DN,DC>}。其中 DN(Device Name)表示外部设备名。DC(Device Configuration)表示对该设备的配置操作集合。

PC 定义 E 构件时间、调度等性能配置。PC 是一个三元组:PC=<SN,TC,SPC>。其中 SN(Service Name)表示构件服务名。TC(Time Configuration)表示对构件时间参数的配置。SPC(Service Priority Configuration)表示对构件服务优先级定义表的配置。构件时间描述子 TC 是黑盒组装的重要依据,其是一

个五元组:TC=<TTsn,ETsn,STsn,TPsn,TDsn>。sn代表指定的构件服务名。TTsn(Time Type)配置构件服务的时间类型(有三个取值:周期性、偶发性或混合性)。ETsn(Execution Time)配置构件服务的总执行时间。STsn(Starting Time)配置构件服务开始执行的时间。若是周期性的服务类型,TPsn(Time Period)表示对构件服务执行时间周期的配置。TDsn(Time Deadline)配置构件服务完成执行的时间限制。

4.2.2 状态接口规约

状态接口 SI 是 E 构件预留的可编程接口,结合构件验证测试工具可在该处定义 API 或监控器等,以监控构件状态并将监控信息传到构件验证测试工具进行分析处理。对于构件组装 SI 是可选的。SI 是一个二元组:SI=<MM,CM>,其中 MM(Monitor Method)表示构件状态监控方法,CM(Content of Monitoring)表示监控内容。SI 的类 BNF 规约描述如下:

SI=<MM,CM>

MM::=<API,监控器,验证工具,仿真工具…>

CM::= <组装状态,配置状态,服务线程运行状态…>

组装状态::=<组装行为,组装时间…>

配置状态::=<配置完整性验证,配置正确性验证…>

服务线程运行状态::=<初始,准备,运行,挂起,异常终止,终止>

组装行为::=<行为完整性,行为正确性,行为可靠性…>

组装时间::=<时间活性,时间安全性…>

5 ESDCM 构件模型构件组装机制

构件组装旨在构件间建立关联,根据关联协调它们的行为,把它们组织成为一个有机整体的过程,其关键体现在搭建合理的体系结构、E 复合构件粒度的选择和组装方法的选择等方面。E 构件的组装可根据交互协议通过对偶或匹配的接口与连接件的连接来实现,可通过接口绑定或接口组合来实现,亦可通过使用组装模式来实现。重点讨论 E 构件行为组装情况,IN、OUT 行为接口是主要建模元素,其他接口可选。依据 E 构件行为交互的普遍特点,给出 ESDCM 构件模型组装范型如图 2 所示。该组装范型的引入旨在介绍 E 构件行为接口交互方式及构件组装方法。

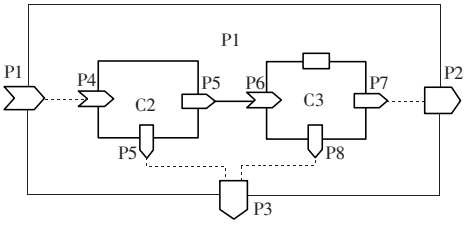

图 2 E 构件组装范型

在组装范型中,C1 是一个 E 复合构件,其中包含子构件 C2 及 C3,C3 包含配置接口表示其含有硬件控制,各构件的 IN、OUT 接口按照各自的命名空间(遵循 CSP[5]端口命名定义)依次命名,构件接口间通过连接件连接交互。对于不同类型的行为接口,其接口间的交互定义为匹配组装,应用于同级构件间的行为交互,如 P5 到 P6 的交互;对于同类的行为接口,其接口间的交互定义为接口绑定,应用于复合构件到自身子构件的交互情况,如 P1 到 P4、P7 到 P2 的交互;接口组合是接口绑定的特例,其特点是多个同类行为接口组合成一个功能更强大的同类行为接口,如 P5、P8 组合成 P3。在 ESDCM 构件模型中,大多构件的交互都可由这三种交互方式构造而成,鉴于它们的不同用途,匹配组装的连接件用实直线表示,接口绑定和接口组合的连接件用虚直线表示。

除了使用组装范型的三种交互方式来组装构件,还可借鉴程序设计思想抽象出一些常用的构件组装方法来实现组装,如顺序组装、并行组装、选择组装、复制组装及中断组装等,这类方法被许多构件模型所采用,在 ESDCM 构件模型中被称作组装模式,并把它作为一种组装机制。组装模式相当于广义的连接件,各自拥有不同的组装协议及适用条件,且可根据需求自定义和扩展以提高组装效率。对于组装模式,不再赘述,详细内容可参看文献[6]。

6 ESDCM 构件模型构件组装规约

为了准确描述 E 构件交互及组装,采用形式化语言 CSP[5]及 TCSP[7]搭建可扩展的 Wright[8]体系结构描述语言框架进行组装描述。一个构件将对应一份组装规约,该文件随同构件一起保存于构件库中。Wright 根据构件、连接件和配置等基本体系结构元素抽象构造,其行为描述遵循 CSP 语言结构。扩展后的 Wright 增加了 TCSP,扩展了实时行为描述。E 构件组装规约结构遵循扩展的 Wright 语言描述框架如下:

```
System SysName
    Description Language = [language specification]
    Style StyleName
        Style specification…
    End Style
    …
    Component CompName
        Port port1= [port1 specification…]
        Port port1= [port1 specification…]
        …
        Computation = [Computation specification…]
    …
    Connector ConName
        Role RoleName (Source) [source specification…]
        Role RoleName (Sink) [sink specification…]
        Glue GlueName [glue specification…]
    Instances = [Instance specification…]
    Attachments = [Attachments specification…]
End System
```

在该描述框架中,Style 语义体现了构件构造的可复用性并提高规约描述的规范化,如可定义构件的接口风格及组装模式等,可由此定义针对特定问题域的构件。其他语义体现了构件的行为描述,嵌入 CSP、TCSP 语言来进行该描述,其中 CSP 描述非实时行为,TCSP 描述实时行为。Computation 重点体现了构件行为,它是黑盒组装的重要依据。Description Language 定义了构件的行为是否是实时行为。通过该描述框架来构造构件组装规约,E 构件可以灵活的扩展组装范型完成各种复杂的组装活动,提高了 ESDCM 构件模型的普适性。对于三种不同的交互方式,匹配组装主要体现了这种描述框架,接口绑定的组装规约体现为构件接口的规约复制,接口组合的组装规约则体现为构件接口的行为推导,后两种情况的连接件属于虚连接。对于组装模式组装,可在描述框架的 Style 中定义,其组装规约体现为子构件组装的行为推导。行为推导可使用 CSP 进程变换来实现,方法可参看文献[6]。

7 ESDCM 构件实时行为时间性推理

ESDCM 构件模型实时行为的时间性采用 TCSP 语义模型理论[9]来推理,以完善配置接口的外界组装时间依据,完善状态接口验证机制中构件组装的时间有效性及安全性描述。

7.1 TCSP 语义模型

TCSP 语义模型[7]使用观察(observation)来操作一系列带时间的事件。每个观察用二元组(s,X)表示,s 表示带时间的迹(可观察的带时间的事件记录),X 表示带时间的拒绝集(带时间的被拒绝的事件的记录集合)。带时间的事件用二元组(t,a)表示,其中 a 为事件,t 为事件发生的时间。时间值域为:$TIME==[0,\infty)$。记 $\sum$ 为事件全集,则带时间的事件集定义为:$T\sum==TIME\times\sum$。一个带时间的迹是一个按时间顺序展开的事件序列,带时间的迹集为:$TT==\{s\in seq T\sum \mid <(t_1,a_1),(t_2,a_2)>\leqslant s\Rightarrow t_1\leqslant t_2\}$,其中 $s_1\leqslant s_2$,当且仅当 s_1 是 s_2 的子序列。拒绝托肯集定义为 $RT=\{[t_1,t_2)\times A \mid 0\leqslant t_1\leqslant t_2<\infty\wedge A\subseteq\sum\}$,带时间的拒绝集 X 是在特定执行中的拒绝事件记录的集合,带时间的事件(t,a)是 X 中的元素,当且仅当在执行中时间 t 时拒绝事件 a。带时间的拒绝集定义为 $TR==\{\cup R \mid RT\wedge R$ 是有限的$\}$。所以任何对拒绝行为的观察可由有限个拒绝托肯的联合来刻画。带时间的观察定义为:$TF==TT\times TR$。以下是迹上的操作语法定义:

取时间操作,返回操作对象中的时间定义如下:

$times(s)==\{t \mid \exists a\cdot <t,a>\}$

$times(X)==\{t \mid \exists a\cdot <t,a>\in X\}$

$times(s,X)==times(s)\cup times(X)$

取事件操作,返回操作对象中的事件定义如下:

$\delta(s)==\{a \mid \exists t\cdot <t,a>\}$

$\delta(X)==\{a \mid \exists t\cdot <t,a>\in X\}$

$\delta(s,X)==\delta(s)\cup\delta(X)$

迹或拒绝集上的 during 操作符 $\uparrow I$ 操作返回一段时间内的迹或拒绝集,其中 $I\in TIME$:

$<>\uparrow I=<>$; $X\uparrow I=X\cap(I\times\sum)$;

$$(<(t,a)>\frown s)\uparrow I=\begin{cases}<(t,a)>\frown(s\uparrow I) & \text{当 } t\in I\\ (s\uparrow I) & \text{其他}\end{cases}$$

如果 inf 表示时间的下界,sup 表示时间的上界,时间空集的下界为 0,上界为 ∞,$[0,end(s,X)]$表示观察的持续时间。取最初事件、最末事件的时间操作定义如下:

$begin(s)=inf(times(s))$ $\quad end(s)=sup(times(s))$

$begin(X)=inf(times(X))$ $\quad end(X)=sup(times(X))$

$begin(s,X)=inf\{begin(s),begin(X)\}$

$end(s,X)=\max\{end(s),end(X)\}$

7.2 ESDCM 构件时间规约推理

4.2.1 节 ESDCM 配置接口规约中定义了 TC 接口,TC 五元组 TTsn,ETsn,STsn,TPsn,TDsn 是每个 E 构件时间属性量化的对外体现,它是外界进行黑盒组装的时间依据,由构件制作者完成配置。以下给出其基于 TCSP 语义模型的推理方法。

推理 1 TTsn 时间类型配置有三个参数:周期性、偶发性和混合性。TTsn 参数选择取决于对该构件带时间的事件集 $T\sum$ 的观察,若在该构件的 $T\sum$ 中仅存在周期性事件 a,则配置为周期性。偶发性同理类推。而若在该构件的 $T\sum$ 中既存在周期性事件 a,又存在偶发性事件 b,则配置为混合性。

推理 2 ETsn 配置构件服务的总执行时间。TF 为某构件带时间的观察,操作$[0,end(s,X)]$为观察的持续时间,则 $ETsn=end(s,X)-begin(s,X)=\max\{end(s),end(X)\}-inf\{begin(s),begin(X)\}$。即 ETsn 值由取事件时间上下界的操作 $sup(times(s))$ 或 $sup(times(X))$与 $inf(times(s))$或 $inf(times(X))$求差值确定。

推理 3 $STsn$ 配置构件服务开始执行的时间。TF 为某构件带时间的观察,操作 $begin(s,X)$为该构件最早事件的发生时间,则 $STsn=begin(s,X)=inf\{begin(s),begin(X)\}$,即 $STsn$ 值由取事件时间上界的操作 $inf(times(s))$或 $inf(times(X))$确定。

推理 4 若构件服务类型为周期性,TPsn 表示对执行时间周期的配置,即某个进程转移到另一进程的固定周期。TPsn 的取值将在 TCSP 行为描述中反映,由于 TCSP 行为描述可通过多种 TCSP 语法操作[7]来实现这类情况,如带时间 t 的事件前缀、超时及递归操作等,则 TPsn 值需视具体情况配置。

推理 5 TDsn 配置构件服务完成执行的时间限制。TF 为某构件带时间的观察,操作$[0,end(s,X)]$为观察的持续时间,则 $TDsn=end(s,X)=\max\{end(s),end(X)\}$,$TDsn$ 值由取事件时间下界的操作 $sup(times(s))$或 $sup(times(X))$确定。

7.3 ESDCM 构件时间活性及安全性

4.2.2 节 ESDCM 构件模型状态接口规约中包含了时间属性描述,提出了时间活性及安全性,它们是时间有效性及正确性验证的基础,也是白盒组装时的构件时间约束体现。活性(LIVE)关注使构件能够正常工作所需具备的时间性质,通过研究 TCSP 语义模型行为观察(s,X)中的事件所组成的迹上的时间性质来描述。安全性(SAFE)关注使构件能够安全运行所需具备的时间性质,通过研究 TCSP 语义模型行为观察(s,X)中的事件所组成的迹上与时间相关的安全性质来描述。任何 E 构件的时间属性在正常组装状态下需满足时间活性及安全性并集 $LIVE\cup SAFE$。LIVE 和 SAFE 的构造规则如下。

规则 1 E 构件时间活性及安全性需对每一个观察(s,X)直接构造,记为 $LIVE(s,X)$和 $SAFE(s,X)$。活性及安全性由 TCSP 语义模型函数 F_T:syntax→PA 定义,其中 syntax 为语法,PA 为由 TF 的幂集定义的满足活性或安全性要求的规格。

规则 2 时间活性或安全性与规格的关系用满足性刻画,满足活性的规格记为 PA_L,即 $PA_L\ Sat\ LIVE(s,X)==\forall(s,X)\in F_T[PA_L]\cdot LIVE(s,X)$。满足安全性的规格记为 PA_S,即 $PA_S\ Sat\ SAFE(s,X)==\forall(s,X)\in F_T[PA_S]\cdot SAFE(s,X)$。

规则 3 时间活性重点描述保障构件正常工作的时间性质,表现为某一事件条件下在某一特定时间内不能拒绝某一事件的发生,否则构件的功能需求是无效的。例如:构件某观察为 s,若在 t 时刻发生 a,则在$(t+3,\infty)$时间范围内不能拒绝 c 的发生。记为:$<(t,a)>\leqslant s\Rightarrow c\notin\delta(X\uparrow(t+3,\infty))$。

规则 4 时间安全性重点描述在构件正常工作的基础上构件安全运行的时间性质,表现为在某一事件条件下或在某一特定时间内事件发生的一系列发生约束。发生约束包括:事件不发生、不再发生、至少发生 1 次或 n 次等。例如:构件某观察为 s,每个长度为 T 的时间间隔内,a 至少应发生 1 次。记为 $S=s'\frown<(t,a)>\frown s''\Rightarrow(\exists t'\cdot<(t',a)>\leqslant s'\wedge t-t'<T)$。

运用以上规则可对 E 构件时间活性及安全性进行准确推理,从而完善 ESDCM 构件模型状态接口时间规约的描述,保证实时嵌入式构件的实时准确性和可靠性。关于规则 3 及规则 4 的子规则较多,由于篇幅所限,在此不再详细展开。

(下转 85 页)