

特定应用的嵌入式操作系统构造方法研究^{*}

朱立新¹ 王飞跃²

(中国科学院自动化所智能控制系统工程中心 北京100080)¹

(美国亚利桑那大学系统与工业工程系 亚利桑那 AZ85721)²

摘要 随着通用型嵌入式实时操作系统的发展,一个以面向 Internet 网络的、特定应用的嵌入式操作系统成为实时嵌入式系统极为重要的发展方向。本文详细阐述了特定应用嵌入式操作系统 ASOS(Application Specific embedded Operating Systems)的概念和特点,提出了 ASOS 的三种构造方法:面向对象技术,组件技术,模式生成。

关键词 特定应用的嵌入式操作系统,构造方法,面向对象技术,组件技术,模式生成

Study on Construction Methods of Application Specific Embedded Operating Systems

ZHU Li-Xin¹ WANG Fei-Yue²

(Intelligent Control and System Engineering Center Institute of Automation CAS, Beijing 100080, China)¹

(Department of Systems & Industrial Engineering, University of Arizona, Tucson, AZ 85721, USA)²

Abstract The ASOS (Application Specific embedded Operating Systems) has now become a promising development field of the embedded system. The paper discusses the concept of the ASOS, and the main characteristics of the systems. The object-oriented technique, component-based technique and the pattern-generation technique are proposed in building the ASOS.

Keywords Application specific embedded operating systems, Construction methods, Object-oriented technique, Component-based technique, Pattern-generation technique

1 引言

嵌入式系统是“以应用为中心、软件硬件可裁剪的、适应应用系统对功能、可靠性、成本、体积、功耗等严格综合性要求的专用计算机系统”。嵌入式系统主要由嵌入式处理器、相关支撑硬件、嵌入式操作系统及应用软件系统等组成,它是集软硬件于一体的可独立工作的系统。

当今社会,人们的工作生活中随处都可以见到嵌入式系统的应用,从现代家用电器、便携式通讯设备、智能仪器仪表、信息终端、智能交通、航天航空、军事装备、制造工业过程控制等。各种嵌入式操作系统产生出来,如 VxWorks、QNX 系统、Windows CE 系统、PSOS 系统,到如今风起云涌的开放源码嵌入式 Linux 系统。

随着嵌入式系统的发展,特别是嵌入式网络技术的出现,在面向特定应用的领域时,嵌入式系统开发迫切要求面向应用的、可剪裁的操作系统,性能和指标在特定领域最优化的操作系统,我们称之为特定应用的嵌入式操作系统,简称 ASOS (Application Specific embedded Operating Systems)。

2 概念

在国外,ASOS 概念最初于20世纪90年代初提出,随后这一概念得到广泛的引用和延伸^[1,2]。王飞跃教授在文[3]中,重申了这一概念。国外,对 ASOS 构造的研究,产生出了很多方法^[4,5],这些方法对 ASOS 构造进行了有益的探讨,但并没有

形成统一的 ASOS 系统构造方法的标准。国内,在实时嵌入式系统方面侧重应用开发。对 ASOS 系统的研究,处于起步阶段^[3,6]。本文结合中科院自动化所中科智控有限公司自主开发的 CASIC 型交通信号机的实例,提出了构造 ASOS 系统中的三种方法。

特定应用的嵌入式操作系统的产生是因为每一个领域都有其不同的功能特点,对嵌入式操作系统的功能要求不一样。如果一个操作系统在一个实时系统领域是最优化的,则在另一应用领域并非是最优化的,或者根本不能运行。通用的嵌入式操作系统强调的是系统的通用性、资源可配置性和整体效率。特定领域的操作系统所要求的性能指标,通用的嵌入式操作系统不能完成。ASOS 侧重特定应用的功能的指标最优化。通用嵌入式操作系统要实现特定应用领域的功能,大多是提供 API 接口,一定程度上并不适用特定领域的需要。同时,特定领域的功能需要应用特制的高性能的操作系统。怎样针对特定领域的特点,构造最优化的操作系统,是 ASOS 所要研究的问题。随着嵌入式系统的飞速发展,特定应用操作系统的构造成为嵌入式系统研究的热点。

3 特点

ASOS 是特定专用的系统,具有“专”,“精”,“优”的特点。“专”,针对一类用途,实现特定领域的功能,但是又不失“全”,涵盖了该领域所有的功能。“精”,最简单最精简的配置。“优”,达到特定领域最优化的性能指标,如调度算法等各方面。具体

^{*} 中国科学院和国家计委“国外杰出人才引进计划”,“国家杰出青年基金”(基金号60125310)资助。朱立新 博士生,研究领域为实时系统,智能交通;王飞跃 博士生导师,美国亚利桑那大学系统与工业工程系终身教授,兼任中科院自动化所研究员,中科院自动化研究所智能控制与系统工程中心主任,研究领域为智能控制理论,智能交通系统,实时嵌入式系统。

体现在几个方面:

面向特定应用的简化型系统调用接口,专门支持一种或一类嵌入式应用;最小内核处理集,系统开销小,运行效率高,并可用于各种非计算机设备;可伸缩性、可裁减的系统体系结构,提供多层次的系统体系结构;具有各种即插即用的设备驱动接口;具有基本网络功能。

以中科院自动化所中科智控有限公司自主开发的一种新型交通信号机 CASIC 型交通信号机为例,它利用了计算机控制的最新概念和技术,实现了强大的交通管理与交通控制功能。其基本结构和功能是:

(1)处理器采用 Motorola PowerPC 8245,功耗小,抗干扰,尤其适合工业控制。(2)操作系统采用实时嵌入式 Linux 基础上,专用特制。(3)外部总线采用工业控制标准 COMPACT PCI 总线,多串口卡,实现多种方式的数据采集。

为了实现路口机的嵌入式操作系统达到最优化配置,我们在操作系统内核方面进行了处理,使内核只完成路口机的基本任务。内核主要包括以下四个功能:(1)内核的调度算法:针对路口机的任务不是硬实时的任务这一情况,我们更改了其中的调度算法的模块,采用了 EDF 的调度算法策略。(2)网络堆栈:网络方面,提供与监控系统上层通信的 TCP/IP 协议支持,其它的 UDP,PPP 网络协议未予支持。(3)内存管理:内存管理的基本文件占了大约100k 左右,完成内存寻址,任务调度,中断处理。(4)文件系统:只保留了内核运行必须的指令,其它的最低限度的配置文件。

这样的操作系统,完全是面向智能交通路口信号机的特定的功能任务,而设计完成的特定操作系统。它仅具备基本任务的内核支持;交通方面的数据采集的硬件驱动支持;网络通信支持;应用功能支持。

以这个操作系统为核心,外加不同嵌入式处理器的模块,智能交通系统数据采集卡的驱动模块,特定应用的功能模块,可以构成只适用于智能交通路口机的特定嵌入式操作系统 ITSOS。ITSOS 操作系统完全可以以商品化的方式,应用到智能交通这一特定领域中来。与通用嵌入式操作系统相比,ITSOS 更趋专业化,强调功能专用性,只是实现单一的性能;只注重针对智能交通的特定功能的高效性;伸缩性,裁剪性高;加载其它额外的功能,只要将功能模块动态连接进系统中;具备基本的网络通讯功能,实现与监控端的通信。ITSOS 的构造,如图1所示。

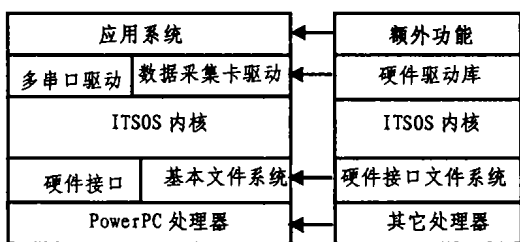


图1

这两部分构成了整个操作系统 ITSOS。额外功能包含智能交通信号机的所有功能,真正体现 ASOS 的“全”的特点;ITSOS 实现最精简的内核,实现“精”;ITSOS 针对智能交通路口机的数据采集,传输,处理等交通领域的一类功能,是“专”用于智能交通领域的 ASOS。

4 ASOS 的构造方法

ASOS 是包含硬实时、软实时、严格实时的嵌入式系统。

它是操作系统,是一个大“软件”,构造思想和方法同传统的操作系统和软件构造思想虽有相同之处,但相比较而言,ASOS 是面向特定应用,单一功能的嵌入式系统,嵌入式系统的条件更加“苛刻”,构造思想、方法区别更大。如为适应实时系统的发展,国外正在研究的精确统一建模语言 PUML;嵌入式系统软硬件协同设计的研究等,这些思想和方法是传统的操作系统和软件构造思想不具备的。

4.1 面向对象技术

实时嵌入式系统构造过程中,模型是建立整个系统,分析时间限制的重要手段。实时系统建模的各种方法中,如 HRT-HOOD^[7]方法等,都渗透了传统的面向对象技术的软件工程设计方法,如 Booch 方法, Jacobson 方法^[8,9]。

统一建模语言 UML (Unified Modeling Language) 方法^[10]结合了 Booch, Rumbaugh 和 Jacobson 方法的优点,统一了符号体系,并从其它的方法和工程实践中吸收了许多经过实际检验的概念和技术。统一建模语言表达力丰富,成为实时嵌入式系统面向对象技术开发的主要方法。我们在构造 ASOS 的过程中,采用了 UML 建模思想。UML 结合实时性建模^[11]的研究已成为实时嵌入式系统建模分析的重要研究方向。

4.1.1 构造过程

1)特定应用领域的环境建模。包括硬件系统和软件系统的建模。硬件环境即硬件层的建模,涉及嵌入式处理器,硬件板卡等。通过静态图和配置图支持软件体系结构和硬件体系结构,显示软硬件系统之间关系的可视化表示。

2)ASOS 的体系结构建模。ASOS 的体系结构,是系统的骨架,是最重要的基础。一般在获取特定应用需要的同时,就应该开始分析系统的体系结构。体系结构现在一般是各个大的功能模块组合成,描述各个部分的关系。

3)ASOS 内部机制建模。内部机制指具体的对象或者类的组织,结合实现某一功能的策略。操作系统任务的执行情况,任务的异步同步通信,调度算法的选择,都是内部机制所要处理的各项功能。

4)具体到类,对象等构造。特定领域分析后,列出系统中的特定领域类、对象、属性、方法等。寻找系统处理的功能来进行特定领域分析,也可以通过用户和领域专家的讨论,识别出要处理的所有关键类及它们的相互关系。

4.1.2 案例简要说明

我们以智能交通路口机的系统 ITSOS 的两个部分为例,简单地说明。

(1)ITSOS 硬件环境建模体系结构。信号机的基本硬件结构如图2所示。

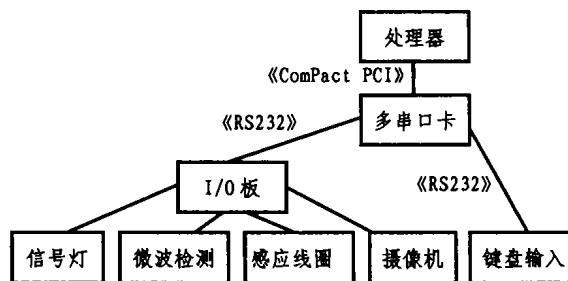


图2

硬件设计基于 PowerPC 微处理器的硬件平台。感应线圈,信号灯,微波检测,视频卡采集交通现场的各种类型信息,液晶显示屏和小键盘供操作人员输入和改变信号机的各项参数。信号机内部有多串口卡,以 PCI 总线相连。采集到的信息

通过多串口卡,送到微处理器中处理。整个信号机通过局域网、光纤或其它方式与上位机进行通讯。这是交通信号机环境下的硬件模型。

(2)ITSOS 应用系统通信交互功能。ITSOS 应用系统,与监控、数据端的通信交互是实现 ITSOS 特定功能的重要一部分。典型的是启动路口机,配置基本运行参数这一过程。图3简单地阐述了这一过程。操作人员启动路口信号机,ITSOS 建立与监控系统的联接,监控系统收到信号,设置路口机的运行参数。随后,智能交通路口机与监控系统,数据服务器实现交互通信。

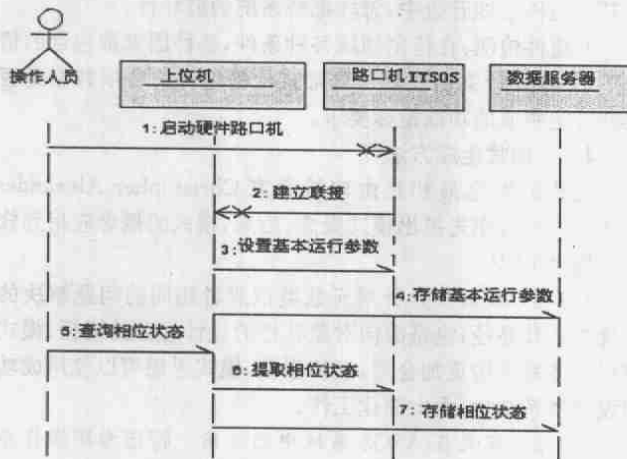


图3

4.2 组件技术

组件技术是面向对象思想的深化和延伸。对象设计是以类的形式表达对象。类虽然提供了封装性、多态性和继承性,但需要依赖于具体的编程语言,耦合度高,且需要用户对类库的结构,语言有较深入的掌握,因此,不能完全达到软件重用的可移植性,可复用性,跨平台,互操作性要求。较之面向对象软件工程思想,组件技术能够使软件复用,易于维护,软件重构,缩短周期。

4.2.1 实时组件 设计可复用实时组件,比设计一般系统的组件,更为复杂。这是由嵌入式实时系统的特点决定的:

1)嵌入式目标系统硬件资源有限。程序固化在有限的闪存和 ROM 中,同时内存小,但是必须完成实时任务,否则会带来灾难性的后果。2)系统的安全性,可靠性要求更高。嵌入式系统处理并发性的复杂任务时,任务发生的时间,发生的顺序,持续时间,系统必须在期限内,作出反应。实时系统的组件必须具有可预测突发事件的能力,同时能够确保任务正确无误地完成。3)稳定性比普通系统更强。一般嵌入式系统需要长时期连续运行,系统的稳定性要好;同时在恶劣的环境下,正常工作。

4.2.2 ASOS 组件库构成 基于组件的 ASOS 开发流程如图4所示。ASOS 组件库的建立,是嵌入式实时系统基于组件技术的开发,维护,升级的核心,实现 ASOS 基于组件技术的关键。

ASOS 组件库中,组件定义主要包括:1)组件的标识:主要包括组件名称,组件的版本号,属性,属性的说明。2)组件接口的定义:输入输出参数名称,参数类型,如是否是整型,结构型,自定义数据类型;相关的接口描述。3)组件功能定义:说明组件的功能,实现什么样的操作,完成何种任务。4)组件依赖的硬件环境描述:特别是与嵌入式系统硬件密切相关的组件,如在处理器是 Arm 系列类型下成功运行的组件,在

PowerPC 系列的处理器环境下,不能运行。5)软件环境依赖描述:某个特定应用的组件,要依赖其它类型的组件提供参数,才能运行正确;或者,需要与其它组件通信交互,异步还是同步,都是需要描述的指标。

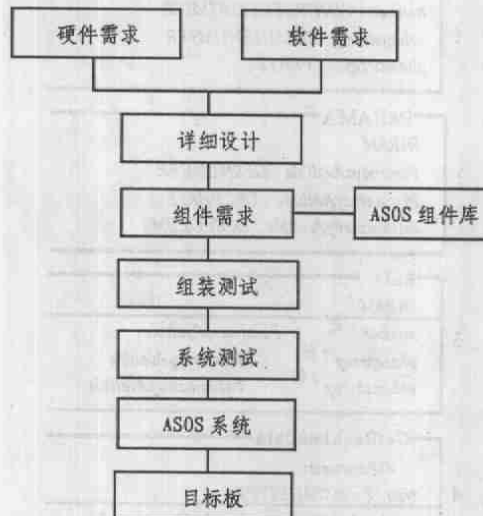


图4

我们引入了形式化的方法来描述组件,对组件精确定位,同时便于组件库的操作。形式化语言消除了组件的模糊性,这样可以使组件标准化,为特定领域系统组件库提供一种选择标准。形式化语言,可以设计,检测,维护系统,可以同系统兼容,很多工具也可以提供支持。形式化语言,语法语义采用数学的一些概念,规划系统虽然繁复,但是形式化语言定义,规划系统是其它的非形式化方法所不能取代的。

组件形式化的方法有多种,如 VDM, B, Z 方法^[12]。我们采用了 Z 方法的定义方式描述组件。Z 方法同 B, VDM 方法通常被称为基于状态的语言。Z 方法产生于牛津大学,是一种形式化语言,建立在数学的集合论和一阶逻辑的数学基础上,有着成熟的形式化表示体系。Z 是一种广泛使用的形式化规约语言。它可以表示集合、元组和序列,也可以表示类,对象,操作。对整个系统,也可以采用 Z 语言各种手段描述^[13]。

ASOS 组件库中,主要分三类组件:系统内核组件依赖于硬件的组件和特定专用组件。

定义1 系统内核组件, $C_{ker} = (Attr, (Func_i)_{i \in I})$ 。

系统内核组件实现任务调度,任务管理的组件。Attr 是系统内核组件属性的集合,包括组件,参数,方法,函数的名称。 $(Func_i)_{i \in I}$ 与 Attr 参数对应的需要的组件功能, I 表示所有对应的组件功能集合。如 Attr 参数定义了调度算法这项功能, $(Func_i)_{i \in I}$ 可以从调度算法集合中选择单调速率算法,或其他满足特定要求的算法。

定义2 依赖于硬件的组件是指针对特定硬件的组件模块, $C_{hard} = (Attr, Init)$ 。

Init 是启动处理器,系统运行的初始化参数。不同处理器,初始化的参数不同。

定义3 实现功能的特定专用组件,以 ITSOS 的应用系统组件为例,监控系统同 ITSOS 通信,获取路口机参数,Z 语言可以表示为: $C_{spec} = (P, Rult, (Op_i)_{i \in I})$ 。

C_{spec} 是 Z 语言的抽象数据类型 (ADT),具有 $P, Rult, (Op_i)_{i \in I}$ 3元变量。P 状态图包含所要获取的参数,参数因操作功能的不同而不同。Rult 状态图是操作执行后,获取后的参数值。 $(Op_i)_{i \in I}$ 功能操作, I 表示监控系统获取路口机参数这一类属的所有操作集合,包括获取统计周期,获取缺省运行参

数等各项操作。

假设监控系统获取路口机的实时统计数据,只包括交通流量和路口相位,这一过程可以用 Z 语言表达为图4。

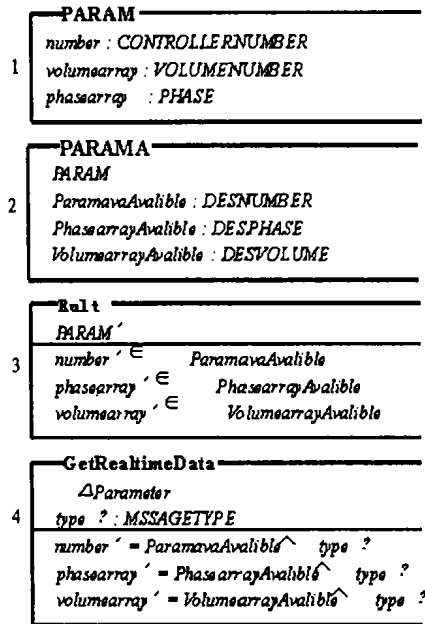


图4

图4中,1表示参与的参数,2数据服务其中的参数值,3变化后的值,4逻辑语言表示获取参数过程。具体的类型定义,声明,逻辑语言表示和实现,限于篇幅,省略。

4.2.3 设计 ASOS 组件的困难 ASOS 组件,因为嵌入式系统的内存和时间需求的限制,必须要满足特定应用条件下的功能需求。合格的 ASOS 组件应具备以下几个条件:1)实时性:在时间限期内,组件任务必须很好地完成。2)可预测性:组件对承担的任务,具有动态地可预测的功能,如按照优先级执行的任务,组件对突发的事件,有改变任务的优先级的能

力。3)可靠性:组件必须具有容错能力,保证整个系统的健壮性。

同时,ASOS 组件必须经过更严格的检验,才能应用到实际的特定领域的环境中。ASOS 的组件发展的难点主要存在以下方面:

1)没有统一的标准:组件交流的种类和信息是通过组件的标准约束和规定实现的。因为时间限定,事件预测性,普通组件的标准 COM, CORBA, JavaBeans 不适合实时组件的标准。实时组件标准的建立是实时系统组件发展的一个趋势。

2)组件的升级:特定领域功能的增加,需要组件的升级,尤其是组件在线升级中,怎样维持系统的健壮性。

3)组件检测:在特定领域各种条件,各种因素都包含的情况下,怎样设计案例或者仿真形式检验组件的各项数据是否达到特定领域的功能指标要求。

4.3 模式生成方法

模式的概念最初是由建筑学家 Christopher Alexander 在建筑学领域率先提出模式概念。后来,模式的概念应用到软件工程之中^[14]。

4.3.1 模式 是针对无数类似或者相同的问题解决的最优方法和路径,也是面向对象软件的设计经验的模板。模式可以使体系结构更加合理,架构强壮。模式思想可以复用成功的设计体系结构,大大简化工作。

4.3.2 模式在 ASOS 系统中的应用 特定专用操作系统的构造思想中,已经包含了模式的设计思想。如操作系统的内核采用了层(Layered)的模式。在基于特定应用的系统中,可以结合特定领域的特点和对操作系统的功能要求,按照模式的生成规则,构造特定功能的专用操作系统。

生成模式可采用规则表达或框架系统。依据特定应用的需求,并基于模式生成规则,生成针对特定应用的 ASOS。一阶和高阶逻辑,谓词逻辑,专家系统构成模式生成规则的基础,是基于模式生成方法的关键。基于模式的生成方法最终可以发展成计算机辅助设计的 ASOS 的开发系统,如图5所示。

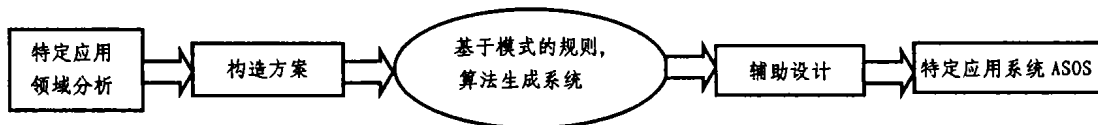


图5

ASOS 系统的模式的生成能力是模式创建并生成整个 ASOS 系统的能力。模式系统可以是一种语言,一种规则体系。模式语言是当今基于模式发展的一个重要方向。模式可以既是模式语言的要素,同时又是模式语言的规则。模式语言的发展可以如数学公式一样,可以通过模式的推导、演绎生成系统架构。当前,模式语言还不完善,还不具有如此强大的生成能力。对于 ASOS 系统基于模式的发展我们正在研究中。

结论 ASOS 从概念和内容上,体现了嵌入式系统的发展趋势和前景。国内,在 ASOS 的研究理念上,仅仅局限于通用嵌入式操作系统的裁剪,而在构造细想和方法方面,研究并不深入,可以说是一个全新的内容。国外,从实时组件,到实时嵌入式系统的体系结构,可重构操作系统的研究正日益成为实时嵌入式操作系统领域的研究方向之一。特定应用的嵌入式操作系统 ASOS 的构造思想和方法,对可重构实时嵌入式操作系统理论和技术,有重要的指导意义。我们将在本文的基础上,进一步阐述 ASOS 的构造思想。

ASOS 已引起许多相关行业的高度重视,美国出现了一些专门对此进行研制开发的公司。随着对高性能与低成本的

不断追求,ASOS 将成为信息家电领域,医疗仪器领域,智能交通领域等许多产品尤其是智能型产品的核心技术,其应用前景十分广阔。

参考文献

- Anderson T. The Case for Application Specific Operating Systems. In: the Proc. of Third Workshop on Workstation Operating Systems[C]. USA, Florida :April 1992
- Stankovic J, et al. Application Specific Operating Systems [EB/OL]. <http://www.cs.virginia.edu/~stankovic/asos.html>.
- 王飞跃,吴朝晖. ASOS:嵌入式系统的发展趋势[N]. 计算机世界, 2000-11-20(B6-B14)
- Gauthier L, Yoo S, Jerraya A A. Automatic generation and targeting of application -specific operating systems and embedded systems software[J]. IEEE transactions on computer-aided design of integrated circuits and systems, 2001, 20(11): 1293~1301
- Bershad B N, et al. SPIN An Extensible Microkernel for Application-specific Operating System Services, ACM SIGOPS European Workshop[C]. Dagstuhl Castle, Germany, 1994
- 赵宇,冯强. 面向应用的嵌入式操作系统[J]. 航空计算技术, 2002, 32(1): 43~47

多安全策略集成性问题的分析与解决^{*})

吴新勇 熊光泽 桑楠

(电子科技大学计算机科学与工程学院 成都610054)

摘要 解决多安全策略的集成性问题是安全操作系统支持多策略和动态策略的基础。本论文采用形式化的方法为安全系统建立了全局安全状态的迁移模型,以 TE 和 RBAC 策略为例分析了不同策略作用下安全关联行为对安全状态的影响,并根据 T&R 集成模型提出了解决多策略集成性和一致性问题的思想,为安全操作系统的实现奠定了基础。

关键词 安全策略,安全操作系统,状态迁移模型,集成性

Analysis and Resolution of Integration of Multi Security Policies

WU Xin-Yong XIONG Guan-Ze SANG Nan

(Computer Science and Engineering College, UEST of China, Chengdu 610054)

Abstract Resolution of integration of multi security policies is the base problem in secure OS which supports multi security policies and dynamic policies. This paper uses a state transition approach to formally analyze a security system and presents the different effects resulted by security-depended actions of different policies on security states. We will analyze TE, RBAC model and T&R model (their integration model), present a thinking to tackle the hybrid integration and consistency of multi-policies, which lay the foundation for implementing secure OS.

Keywords Security policy, Secure OS, State transition model, Integration

1 引言

安全操作系统是保障信息系统良好运作的基石。经过30多年的发展,安全操作系统目前的研究重点集中在支持多策略和动态策略^[1]。多策略要求系统支持当今流行的商用或军用安全策略模型;动态策略要求在多策略的基础上,系统可以在线升级策略,并可保持系统的一致性和连续性。由于策略之间的差异性,安全操作系统的研究必须解决多策略的可集成性问题。

虽然各种安全策略之间上有关联性,但安全内核在策略验证时必须分别满足多策略逻辑。多策略的集成涉及到系统设计的完备性和合理性,既兼顾所有策略的需求,又要使得系统开销最小。多策略的集成要求所有策略对特定系统是可实现,即要求满足 Schneider^[2]提出了安全策略可实施的判定谓词:

$$P(\Pi): (\forall \sigma \in \Pi; \hat{P}(\sigma))$$

其中 Π 表示安全相关的行为集, σ 表示行为序列,比如用户对对象实体的创建、删除、读、写和修改等,而 \hat{P} 是访问控制规则集,即策略的实施部分,可实施的策略要求所有的 σ 能够满

足访问控制规则。行为集 Π 触发了系统状态的变迁,主要是访问权限的变化。采用形式化的方法分析安全关联操作引起的系统安全状态迁移模型有助于理解各种策略之间的逻辑关系,从而获得多策略集成的方案,它也是设计和实现安全核的必要前提,便于对安全关键系统的安全性进行分析和论证,特别是对安全策略和安全核的一致性的验证。本文利用形式化方法,定义了 TE 和 RBAC 策略作用下系统的安全状态集及状态迁移模型,分析了两种策略的可集成性,最后得出可实现的 T&R 集成模型,并且引申到其他的安全策略。对安全操作系统的实现及多策略的集成具有实践性指导意义。

本文第2节介绍了安全策略,定义了安全状态模型;第3、4节分别分析了 TE 和 RBAC 策略下的安全状态迁移模型;第5节就 TE 和 RBAC 集成模型 T&R 模型的可实现性展开讨论;第6节分析了多策略集成的支撑技术;最后给出了结论。

2 安全策略及安全状态模型

2.1 安全策略

安全策略是一套规则或约束集,用来管理一个组织如何管理、保护和发布敏感信息。在计算机系统里,我们关心的是

^{*}) 本文获十五国防预研项目资助。吴新勇 博士研究生,主要研究方向:系统安全及可靠性。熊光泽 教授,博士生导师,主要研究方向:实时计算机系统及软件开发支持。桑楠 副教授,主要研究方向:高可靠实时软件。

- 7 Burns A, Wellings A. HRT-HOOD: A Structured Design Method for Hard Real-Time Ada Systems [M]. Amsterdam, Netherlands: Elsevier Science, 1995
- 8 Booch G. Object-Oriented Design With Applications [M]. Redwood City, CA, USA: Benjamin Cummings, 1991
- 9 Jacobson I, Christerson M, Jonsson P, Overgaard G. Object-oriented Software Engineering: A Use Case Driven Approach [M]. Boston, MA: Addison-Wesley, 1992
- 10 Booch G, Rumbaugh J, Jacobson I. The Unified Modeling Language User Guide [M]. Boston, MA: Addison-Wesley, 2001
- 11 Douglass B P. Real-Time UML: Developing Efficient Objects for Embedded Systems [M]. Boston, MA: Addison-Wesley, 2000
- 12 Frappier M, Habrias H. Software Specification Methods [M]. London: Springer, 2001
- 13 Derrick J, Boiten E. Refinement in Z and Object-Z [M]. London: Springer, 2001
- 14 Gamma E, Helm R, Johnson R, Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software [M]. Boston, MA: Addison-Wesley, 1995