动态演化环境中可信软件行为监控研究与进展 *

万灿军, 李长云
(湖南工业大学 计算机与通信学院, 湖南 株洲 412008)

摘 要: 提出了一个软件行为监控框架,在此基础上,从软件行为描述、软件行为监测、软件行为可信管理和软件行为控制四个方面,阐述了可信软件行为监控的研究进展,分析了目前研究中存在的问题,并探讨了其未来的发展趋势。

关键词: 动态演化; 可信软件; 行为监控; 行为可信
中图分类号: TP311 **文献标志码**: A **文章编号**: 1001-3695(2009)04-1201-04

Research and development on behavior monitoring and controlling of trusted software in dynamic evolution environment

WAN Can-jun, LI Chang-yun
(*School of Computer & Communication, Hunan University of Technology, Zhuzhou Hunan* 412008, *China*)

Abstract: Firstly, this paper presented a framework for behavior monitoring and controlling of software. Then, on this basis, discussed the research and development on behavior monitoring and controlling of trusted software, which including software behavior describing, software behavior monitoring, trust management of software behavior and software behavior controlling. Finally, analyzed the existing problems of research, and also discussed the development trend of future research.
Key words: dynamic evolution; trusted software; behavior monitoring and controlling; behavior trust

软件可信是指软件系统能按照其设定目标所期望的方式运行,能适应环境和需求的变化,有抵御异常情况的能力,在受到干扰时仍能持续提供服务。然而,软件常常发生故障和失误,甚至失效,给人们的工作和生活带来不利影响,甚至造成巨大损失。软件的可信性已成为一个相当普遍的问题,如仅2006 年中航信离港系统发生了三次软件故障,造成近百个机场登机系统瘫痪。因此,人们对软件的可用性、可靠性和安全性等可信性十分关注,如何在软件的开发和运行中保证软件具有高可信性已成为软件理论和技术的重要研究方向[1]。

持续可用性是软件可信性的一个重要方面,也是许多应用领域中的一个关键性要求,软件动态演化可提高软件的持续可用性。软件动态演化是指在软件系统投入运行后,应环境和需求的变化而进行的变更[2]。然而,动态演化过程中可能出现异常,导致系统出错,降低软件的可信性。软件本质上是代替人执行一定的行为。软件的可信性主要表现在其行为可信上,即运行行为可监测、行为结果可评估、异常行为可控制。软件行为的可信不是凭空而来的,需通过软件运行时的行为监测,收集可信性相关数据,验证软件行为是否满足行为规约,建立基于行为监测的可信评估和管理体系,并能根据行为可信评估结果和行为可信需求规约,对软件行为进行动态调控,保障动态演化的正确性和一致性,提高软件的可信性。软件行为监控主要涉及到软件行为的可信监测、可信管理和可信控制。本文旨在从行为描述、行为监测、行为可信管理和行为控制等方面探讨如何进行软件行为监控,为软件动态演化提供可信保障机制,最终为用户提供持续可用、可靠、安全的高可信服务。

1 软件行为监控框架

本文所研究的软件行为监控主要是面向基于中间件的构件软件,所监控的实体主要是软件构件,支持 EJB/CORBA/COM/DCOM/Web services 等不同形式的构件。如图 1 所示,以中间件平台为基础,构建一个可信软件行为监控框架,为基于中间件的构件软件建立一个可信的软件行为监控环境,提供软件行为可信监测、软件行为可信管理和软件行为可信控制三个基本服务,各部分相互协作,共同完成软件行为监控。

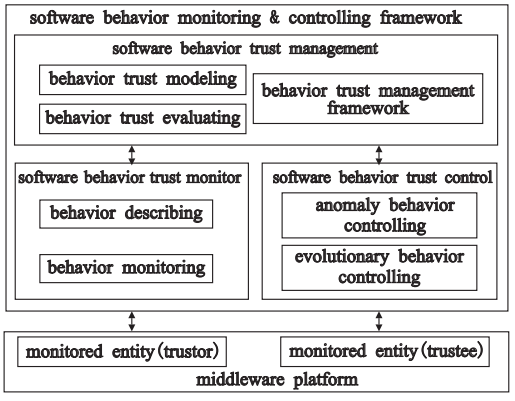

图1 可信软件行为监控框架

在图 1 所示的软件行为监控框架中,被监控实体称为信任实体,分为信任者(trustor)和被信任者(trustee)。如果 trustee 能够满足 trustor 所期望的行为,则 trustee 被认为是可信的。Trustor 若要信任 trustee,需要对 trustee 的运行时行为进行监

收稿日期: 2008-07-01; **修回日期**: 2008-09-23 **基金项目**: 国家自然科学基金资助项目(60773110); 湖南省教育厅优秀青年科研资助项目(06B023); 湖南工业大学科研资助项目

作者简介:万灿军(1982-),男,湖南岳阳人,硕士研究生,主要研究方向为软件体系结构(wancanjun2008@163.com);李长云(1972-),男,教授,博士,主要研究方向为软件体系结构、软件自动化.

测,在此基础上进行行为可信评估,从而决定是否信任 trustee,并决定对其行为是否进行动态调控。在软件行为监控框架中,软件行为可信监测主要包括行为描述和行为监测;软件行为可信管理主要包括行为可信建模、行为可信评估和行为可信管理框架;软件行为可信控制主要包括异常行为控制和演化行为控制。

2 软件行为可信监测

软件行为可信监测是软件行为可信评估的基础,为行为可信评估提供核心依据,主要涉及行为描述和行为监测。

2.1 行为描述

在行为监测研究中,所关心的主体行为是软件系统的体系行为,即软件构件之间的交互行为和软件系统的演化行为,对于软件构件的内部微观行为不必要也不可能被监测到。在构件软件中,构件以主体的身份在完成某项协作活动时,通过构件的接口在运行环境中与其他构件进行交互活动,称为构件的交互行为。为了能够对软件系统的体系行为进行监测、验证、评估和控制,首先需要进行软件行为描述。

行为描述是行为监测、验证、评估和控制的基础。一般而言,软件行为描述均基于某种形式化方法进行,Petri 网、自动机理论和进程代数是使用最多的三种方法。基于 Petri 网[3]和有限自动机[4]的方法在行为描述上采用图形方式较为直观,但当软件实体间的交互过程变得复杂时,这种图形方式容易出现状态空间爆炸,其计算和验证的复杂度也随即增加。因此,有研究者提出采用进程代数的方法来刻画软件行为。π 演算是 Milner 等人[5]对 CCS(通信系统演算)进行扩充而得到的进程代数理论。高阶多型 π 演算由 Sangiorgi[6]在一阶 π 演算的基础上发展而来,旨在描述体系结构和体系行为都不断变化的并发系统。D-ADL[7]是一种形式化的基于高阶多型 π 演算的语言,从运行时角度支持软件系统体系行为描述。相比其他形式化方法,π 演算具有更强的行为规约描述能力,其行为模拟和等价理论为可信软件的构造、验证和演化提供了良好的理论基础,比较适合于软件行为描述。

已有的软件行为描述研究没有考虑到软件可信性与软件行为之间的关系。为了提高软件行为的可信性,需要建立软件可信性与软件行为之间的内在联系和严格描述。另外,采用形式化方法描述软件行为后,一般需要进行形式化验证。模型检测是形式化验证的一种重要途径。常用的模型检测工具有 MWB、SMV、SPIN 等。其中 MWB 适合于验证用 π 演算描述的软件行为。

2.2 行为监测

行为监测是行为可信评估的基础,为行为可信评估提供原始数据。软件行为的可信来源于软件行为监测信息,需要对软件行为进行全面、准确、实时的监测,收集可信性相关数据。目前,在行为监测方面已取得了许多研究成果。Li Jun[8]提出了一个软件行为监测框架,基于全局因果跟踪技术捕获多维软件系统行为。Diakov 等人[9]提出了一种基于 CORBA 中间件平台的软件行为监测框架,能自动生成监测代码来监测构件之间的交互行为。Chen Feng 等人[10]提出了一种运行时行为监测框架 MOP。该框架能根据给定的行为规约自动生成监测器,动态监测系统运行行为,一旦发现违约行为,能立即触发用户定义的操作进行容错处理。Mariani 等人[11]构建了一个自动捕获构件行为的监测框架,使用构件包装器截取被监视构件与其他构件之间的交互行为信息。Gao 等人[12]引入了可追踪构件的概念,提出基于事件的行为跟踪模型,跟踪构件之间的交互活动。Avgustinov 等人[13]将 AOP 技术应用到行为监测中,使用 Aspect 来观察系统中发生的被关注的事件。Bodden[14]开发了一种运行时行为监测工具,使用线性时序逻辑描述系统行为,基于 AspectJ 的织入机制在 Java 源代码中植入行为监测工具。DynamicTAO[15]是一种反射式 CORBA 软件,基于 ORB 的反射机制,通过在对象调用路径中插入截取器,监测软件系统体系行为。

如表 1 所示,上述行为可以分为基于框架的监测机制、自动代码植入机制、自动构件包装机制、反射机制四类。这四类行为监测机制各有优缺点,在实际应用中需要结合起来使用。为了保证软件行为可信,不仅需要进行静态的形式化验证,而且还需要在软件行为监测过程中验证其行为是否满足行为规约,分析其运行行为是否符合行为依据以及是否违背了行为约束。另外,基于中间件的构件软件中的软件实体大多来源于不同的第三方提供者,各软件实体以开放、自主的方式存在于 Internet 的各节点之上,这也给行为监测带来了挑战。

表 1 行为监测机制比较

Monitoring mechanism	Framework-based monitoring	Automatic code insertion	Automatic component wrapping	Reflective mechanism
source code	needed	needed	not needed	not needed
code separation	no	no	yes	yes
overhead	high	low	low	high
complexity	low	very high	low	low
flexibility	high	low	low	high
applicable components	in-house components	in-house components	in-house components and COTS	in-house components and COTS

3 软件行为可信管理

软件行为可信管理旨在为软件动态演化提供可信保障机制,主要涉及行为可信建模、行为可信评估和可信管理框架。

3.1 行为可信建模

软件行为可信建模是指构建一个描述、量化、传递软件实体间行为可信关系的可信模型。行为可信模型所关注的内容主要有行为可信描述和行为可信评估。研究者提出了多个可信模型。Abdul-Rahman 等人[16]从信任的概念出发,对信任内容和信任程度进行划分,并从信任的主观性入手提出了一个分布式信任模型。Beth 等人[17]提出的信任模型引入了经验的概念来描述和度量信任关系,将软件实体之间的信任关系分为直接信任关系和推荐信任关系,分别用于描述信任主体与信任客体、信任主体与信任客体的经验推荐者之间的信任关系。主体对客体的经验既可以直接获得,也可以通过推荐者获得。而推荐者提供的经验同样可以通过其他推荐者获得,直接信任关系和推荐信任关系形成了一条从主体到客体的信任链,而主体对客体行为的主观期望则取决于这些直接的和间接的经验。Herrmann[18]研究了一种构件化软件的信任模型,提出信任适应安全策略实施的概念,采用形式化安全策略描述构件间的行为可信关系。

上述信任模型提供了描述、量化、传递信任关系以及综合多方信任信息的功能,但是他们都试图用概率理论作为数学工具来解决信任模型中遇到的各种问题,无法确切地描述信任模型中的不确定性因素。Jøsang 等人[19]提出了基于主观逻辑的信任模型,使用 D-S(Dempster-Shafer)证据理论来试图解决不

确定性问题,但仍然将信任的主观性和不确定性等同于随机性,而没有考虑信任本身的模糊性。为了更准确地把握和反映信任的本质属性,唐文等人[20]考察了信任的模糊性,构造了一个基于模糊集合理论的主观信任模型,但概念比较抽象,缺少具体化的深入描述。

已有的行为可信建模研究仅着重于信任客体的可信描述,而对于信任主体本身的可信需求描述,除文献[21]提出的面向可信需求的描述框架外,很少引起研究者的足够重视。另外,动态演化环境下软件行为可信建模研究不应只关注软件实体层次的可信性,还要考虑软件系统体系结构层次的可信性。

3.2　行为可信评估

行为可信模型提供了建立和管理行为可信关系的框架。行为可信评估是行为可信模型的核心,它是指通过行为监测,评估主体收集,记录自身与评估客体的直接交互行为信息,并根据经验推荐者所提供的关于评估客体的间接经验推荐信息,得出评估客体的可信评估值,最终对评估客体的行为得出是否可信的判断,其结果将决定软件系统是否需要演化以及如何演化。一般来说,可信评估分为直接的和间接的。直接可信评估是指评估主体与评估客体之间不需要经过任何第三方信任实体的可信评估;间接可信评估是指主体对客体的可信评估需要经过某个或某些第三方信任实体。如图 2 所示,当评估主体 A 对评估客体 B 进行可信评估时,两类信任关系参与评估:a)直接信任关系,评估主体 A 与评估客体 B 存在直接交互行为,A 具有 B 能够完成某项协作活动的直接可信评估;b)推荐信任关系,评估主体 A 与评估客体 C 没有直接交互行为,需要推荐实体 B 提供关于 C 的协作经验信息,B 与 C 之间存在直接信任关系,A 通过 B 提供的经验推荐信息对 C 进行间接可信评估。

目前,在可信评估方面已有大量研究。Abdul-Rahman 等人[22]提出的可信评估模型,强调可信度传递的条件性,给出了可信度的传递协议和计算公式,但没有给出可信度综合计算公式。Beth 等人[17]提出了一种基于经验和概率统计的可信评估模型,以对实体完成任务的期望为基础,根据肯定经验和否定经验计算出实体能够完成任务的概率,以此概率作为实可信度的度量,并给出了由经验推荐所引出的可信度推导和综合计算公式;但是其可信度计算采用简单的加权求和与算术平均,无法反映信任关系的真实情况。Jøsang[23]引入了事实空间和观念空间的概念来表述和度量信任关系,以描述二项事件后验概率的 Beta 分布函数为基础,给出了一个由观察到的肯定事件数和否定事件数来确定的概率确定性密度函数,并以此为基础计算实体产生每个事件的概率可信度。Jøsang 还定义了一套用于可信度计算的主观逻辑算子,主要包括推荐算子和合意算子,分别用于可信度推导计算和可信度综合计算。Wang Yao 等人[24]使用贝叶斯网理论来解决可信评估问题,但该方法过于依赖专家经验,对经验可靠性要求较高,主观性较强。

对行为可信关系进行评价的主要依据来源于相关的直接交互行为信息和间接经验推荐信息,判断一个实体是否成功地完成某项协作活动与判断一个实体是否诚实地提供推荐信息相比,要涉及较少的主观判断。上述可信评估方法大多依赖于主观经验数据,较少利用软件行为本身的监测数据,而在动态演化环境下更应注重依赖实时监测到的可信相关数据,采用基于行为监测的可信评估方法。

3.3　可信管理框架

可信管理框架旨在为软件行为可信建模和行为可信评估等提供管理活动。在可信管理框架方面,国内外也有比较深入的研究。Zheng Yan 等人[25]研究了构件化软件中的可信管理框架,重点关注基于运行观察的自主可信管理,主要包括四个方面:信任建立、信任监控、信任评估、信任控制和重建。北京大学[26]以 Internet 环境下的软件服务为研究实体,构建了一个基于中间件的五层可信管理框架,旨在提供通用的可信管理基础设施。南京大学[27]针对当前软件服务协同环境下安全问题呈现出的新特点,提出了一个基于经验的可信度计算模型,并在此基础上设计了一个基于可信度的可信管理框架。

上述可信管理框架对于软件行为可信关注不足,尤其是对如何通过动态演化机制使得软件行为可信关注不足。为了保障动态演化环境下软件行为可信,需要多关注支持行为可信的软件动态演化机制。鉴于上述研究工作中存在的问题,本文初步提出了一个基于行为监测的可信管理框架,如图 3 所示。

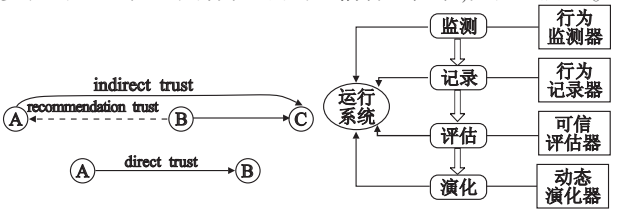

图2　行为可信评估

图3　基于行为监测的可信管理框架

基于行为监测的可信管理框架主要由行为监测器、行为记录器、可信评估器和动态演化器构成,各部分相互协作,共同完成行为可信管理。在运行系统中每个软件实体发生交互的地方植入行为监测器,监测实体的交互行为,收集其交互行为信息,并使用行为记录器实时地存储这些信息,为行为可信评估提供核心依据。可信评估器根据评估主体和评估客体的直接交互行为信息,以及推荐实体给出的经验推荐信息,对评估客体的行为动态地进行可信评估,其结果为运行系统的动态演化提供依据。动态演化器根据行为可信需求规约和行为可信评估结果对运行系统进行动态演化,主要包括运行系统体系结构的动态配置和体系行为的动态调控。

4　软件行为可信控制

软件行为可信控制旨在提高软件的持续可用性,提高软件行为的可信性,主要包括异常行为控制和演化行为控制。动态演化环境下软件可信要求软件系统能随环境和需求的变化而进行动态演化。Garlan 等人[28]研究了基于运行时体系结构的自适应系统 Rainbow。它采用外置运行时体系结构,通过 Probe-Gauge-Gauge Consumer 三层监控机制,获取和度量系统变化来触发自适应规则实现自适应演化。南京大学[29]也提出了一种面向体系结构的自适应系统 Artermis-ARC。它采用内置运行时体系结构,通过 Agent-Gauge-Monitor 三层监控机制,驱动软件系统进行自适应演化。

反射式中间件是一种能够通过与系统运行状态和行为具有因果关联的系统自述来监测并调整系统状态和行为的中间件系统。OpenCorba[30]是一种基于 CORBA 的反射式中间件,通过元类将 ORB 内部特征分离并单独实现,从而允许软件系统在运行过程中监测并调整这些内部功能单元。北京大学[31]开发的 PKUAS 是一个基于软件体系结构的反射式中间件,它

所提供的实时监控工具能监测到运行时体系结构的运行状态和行为并加以调整。文献[32]中提出了基于体系结构空间、支持动态演化的软件模型 SASM,使用反射技术通过对体系结构空间的观察,可获知系统的结构和行为信息,通过对体系结构空间的在线调整可实现系统的非预设动态演化。

从软件行为可信的角度分析上述研究,在软件动态演化过程中,它们都没有将行为可信性考虑进去。由于缺乏行为监测,不能进行行为可信评估,不考察主体行为的可信性,很难保证动态演化的正确性,也难以满足动态演化环境下软件可信性需求。在软件运行过程中,软件系统依据行为可信需求规约和行为可信评估结果,可能需要采取相应的控制策略(如故障报警、容错处理和诊断恢复等)使得异常行为可控制,采用合适的演化策略对软件行为进行动态调控,从而提高软件的自适应性和可信性。

5 结束语

本文已从软件行为描述、软件行为监测、软件行为可信管理和软件行为控制四个方面,总结归纳了动态演化环境下可信软件行为监控的主要研究进展,并分析了其存在的主要问题。在此基础上,笔者认为在如下几方面值得进一步展开研究:a)在行为描述方面,需要建立软件可信性与软件行为之间的内在联系和严格描述;b)在行为监测方面,需要设计有效的收集可信性相关数据的行为监测机制;c)在行为可信管理方面,需要研究基于行为监测的可信评估方法,构建基于行为监测的可信管理框架;d)在行为控制方面,需要研究支持行为可信的软件动态演化机制。

参考文献:

[1] 陈火旺, 王戟, 董威. 高可信软件工程技术[J]. 电子学报, 2003, 31(12): 1933-1938.

[2] KRAMER J, MAGEE J. The evolving philosophers problem: dynamic change management[J]. IEEE Trans on Software Engineering, 1990, 16(11): 1-33.

[3] 罗军舟, 沈俊, 顾冠群. 从 Petri 网到形式描述技术和协议工程[J]. 软件学报, 2000, 11(5): 606-615.

[4] HELKE S, KAMMIILLER F. Representing hierarchical automata in interactive theorem provers[C]//Proc of the 14th International Conference on Theorem Proving in Higher Order Logics. London: Springer-Verlag, 2001: 233-248.

[5] MILNER R, PARROW J, WALKER D. A calculus of mobile processes[J]. Information and Computation, 1992, 100(1): 1-40.

[6] SANGIORGI D. Expressing mobility in process algebras: first-order and higher-order paradigms[D]. Edinburgh: University of Edinburgh, 1992.

[7] 李长云, 李赣生, 何频捷. 一种形式化的动态体系结构描述语言[J]. 软件学报, 2006, 17(6): 1349-1359.

[8] LI Jun. Monitoring and characterization of component-based systems with global causality capture[C]//Proc of the 23rd International Conference on Distributed Computing Systems. 2003: 422-431.

[9] DIAKOV K, BATTERAM J, ZANDBELT H. Monitoring of distributed component interactions[C]//Proc of IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing. New York: [s. n.], 2000: 229-243.

[10] CHEN Feng, GRIGORE R. MOP: an effcient and generic runtime verification framework[C]//Proc of OOPSLA 2007. 2007: 569-588.

[11] MARIANI L, PEZZE M. A technique for verifying component-based software[J]. Electronic Notes in Theoretical Computer Science, 2005, 116(1): 17-30.

[12] GAO J, ZHU Y, SHIM S, *et al*. Monitoring software components and component-based software[C]//Proc of the 24th IEEE Annual International Computer Software and Applications Conference. Taiwan: [s. n.], 2000: 403-412.

[13] AVGUSTINOV P, TIBBLE J, BODDEN E, *et al*. Aspect for trace monitoring[C]//Proc of Formal Approaches to Testing Systems and Runtime Verification. Seattle, WA: [s. n.], 2006: 20-39.

[14] BODDEN E. A lightweight LTL runtime verification tool for Java[C]//Proc of OOPSLA 2004. 2004: 306-307.

[15] FABIO K, MANUEL R. Monitoring, security, and dynamic configuration with the dynamicTAO reflective ORB[C]//Proc of Middleware 2000. New York: [s. n.], 2000: 121-143.

[16] ABDUL-RAHMAN A, HALIES S. A distributed trust model[C]//Proc of Workshop on New Security Paradigms. Langdale: [s. n.], 1997: 48-60.

[17] BETH T, BORCHERDING M, KLEIN B. Valuation of trust in open network[C]//Proc of the European Symposium on Research in Secuirty. Brighon: [s. n.], 1994: 3-18.

[18] HERRMANN P. Trust-based procurement support for software components[C]//Proc of ICECR'01. Dallas: [s. n.]. 2001: 505-514.

[19] JøSANG A, KNAPSKOG S J. A metric for trusted systems[C]//Proc of the 21st National Security Conference. 1998: 541-549.

[20] 唐文, 陈钟. 基于模糊集合理论的主观信任管理模型研究[J]. 软件学报, 2003, 14(8): 1401-1408.

[21] ZHU Man-ling, LIU Lin, JIN Zhi. A social trust model for services[C]//Proc of the 11th Australian Workshop on Requirements Engineering. Adelaide: [s. n.], 2006: 35-42.

[22] ABDUL-RAHMAN A, HALIES S. Using recommendations for managing trust in distributed systems[C]//Proc of IEEE Malaysia International Conference on Communication. Kuala Lumper: [s. n.], 1997

[23] JøSANG A. An algebra for assessing trust in certification chains[C]//Proc of Network and Distributed Systems Security Symposium. San Diego: [s. n.], 1999.

[24] WANG Yao, VASSILEVA J. Bayesian network trust model in peer-to-peer networks[C]//Proc of the 2nd International Workshop on Agents and Peer-to-Peer Computing. Berlin: Springer-Verlag, 2004: 23-34.

[25] ZHENG Yan, MacLAVERTY R. Autonomic trust management in a component based software system[C]//Proc of the 3rd International Conference on Autonomic and Trusted Computing. Wuhan: [s. n.], 2006: 279-292.

[26] ZHOU Ming-hui, WEN Jiao-pin, MEI Hong. Customizable framework for managing trusted components deployed on middleware[C]//Proc of the 10th ICECCS. Shanghai: [s. n.], 2005: 283-291.

[27] 徐锋, 吕建, 郑玮, 等. 一个软件服务协同中信任评估模型的设计[J]. 软件学报, 2003, 14(6): 1043-1051.

[28] GARLAN D, CHENG Shang-wen, HUANG An-cheng, *et al*. Rainbow: architecture-based self-adaptation with reusable infrastructure[J]. IEEE Computer, 2004, 37(10): 46-54.

[29] 马晓星, 余萍, 陶先平, 等. 一种面向服务的动态协同架构及其支撑平台[J]. 计算机学报, 2005, 28(4): 467-477.

[30] LEDOUX T. OpenCorba: a reflective open broker[C]//Proc of the 2nd International Conference on Reflection. Heidelberg: Springer-Verlag, 1999: 197-214.

[31] 黄罡, 王千祥, 梅宏, 等. 基于软件体系结构的反射式中间件研究[J]. 软件学报, 2003, 14(11): 1819-1826.

[32] 李长云, 李莹, 吴健, 等. 一个面向服务的支持动态演化的软件模型[J]. 计算机学报, 2006, 29(7): 1020-1028.