

Web 服务的模型检测技术探讨

刘如娟^{1,2}, 戴桂兰², 胡长军¹, 白晓颖³

¹(北京科技大学 信息工程学院, 北京 100083)

²(清华大学 信息技术研究院, 北京 100084)

³(清华大学 计算机科学与技术系, 北京 100084)

E-mail: susuliuxiao@163.com

摘要: 从模型检测的基本概念出发, 针对目前面向服务计算模式中, 组合Web 服务验证确认过程存在的问题, 着重讨论了模型检测应用于Web 服务验证中的关键技术, 例如如何验证动态绑定服务的可信性, 如何克服服务组合过程中的复杂性和不确定性; 比较全面地总结并比较了几种具有代表性的对Web 服务进行验证和确认的模型检测技术和相关工具; 并探讨了应用于Web 服务验证框架的模型检测技术研究中存在的一些问题及相应的解决方案

关键词: 面向服务的计算; Web 服务规范; 模型检测; 服务验证

中图分类号: TP311

文献标识码: A

文章编号: 1000-1220(2007)11-1921-07

Discussion on Model Checking Technology over Web Service

LIU Ru-juan^{1,2}, DAI Gui-lan², HU Chang-jun¹, BAI Xiao-ying³

¹(Institute of Information Engineering, Beijing University of Science and Technology, Beijing 100083, China)

²(Research Institutions of Information Technology, Tsinghua University, Beijing 100084, China)

³(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract Based on the fundamental concepts in model checking technology, this paper mainly focuses on the key model checking technology in the application of composite Web service verification. At present, there are some difficulties during the process of composite web service verification and validation with the service oriented computing pattern, for example, how to verify the trustworthiness when services are combined dynamically or how to solve the complexity and undecidability during the process of service composition. It also summaries and compares several kinds of typical model checking application in the verification and validation for web services. At last, it discusses some existing problems and provides corresponding solutions in model checking research when it is applied in the web service validation and verification framework.

Key words: service oriented computing; Web service specification; model checking; Web service verification

1 引言

面向服务计算(Service Oriented Computing)是一种基于Internet的全新的计算模式,它在传统的面向对象(Object Oriented)、基于构件的开发(Component Based Development)、分布式对象计算(Distributed Object Computing)及Web技术的基础上,提出了一种新的软件开发、部署和集成的模式,从而有效解决在分布、动态、异构环境下,数据、应用和系统集成的问题。在IBM、Intel、Microsoft、SAP、Oracle等各计算机公司的积极倡导和大力推动下,已成为未来计算机软件发展的主要方向。Gartner预测,到2008年,面向服务的体系架构(Service-Oriented Architecture)将成为80%的开发项目基础,它将结束传统的整体软件体系架构长达40年的统治地位,成为占有绝对优势的软件工程实践方法。

作为一种新的分布式计算模式,面向服务的计算模式为

软件开发提供了灵活的设计和开发模式,组合Web服务能够被重新组合和配置,有时甚至可以是不同开发商提供的组件间的组合,因此在分布式计算中的应用越来越普遍。面向服务的系统具有分布应用、存在各种运行时行为、涉及多种标准协议等特点,可能在硬件、软件、通信、对象管理等各个环节出现各种缺陷。其体系结构和应用的复杂性,以及技术和规范不断的发展变化,对验证技术的研究提出了新的方向。灵活的面向服务的软件架构为我们提供了软件开发美好的前景,但是如何提供一种统一的Web服务验证框架来验证不同服务提供商提供的服务的可信度并将这些服务很好的组合起来,是研究人员面临的重要挑战之一。目前在验证Web服务的研究中,常用的两种方法是自动测试和模型检测。模型检测是形式化方法的一种,它的优势在于能够为系统模型提供一种详尽的证据,验证该系统是否能满足某些属性,比如安全时态属性、可达性及公平性等。

收稿日期: 2006-07-17 基金项目: 国家自然科学基金项目(60603035)资助 作者简介: 刘如娟,女,1977年生,博士研究生,研究方向为Web服务检测,模型检测等;戴桂兰,女,1972年生,副研究员,研究方向为软件工程和软件测试等;胡长军,男,1963年生,教授,博士生导师,研究方向为并行计算和软件工程等;白晓颖,女,1973年生,副教授,研究方向为软件工程和软件测试等。

2 模型检测关键技术

模型检测是一种自动验证有限状态系统的技术,它以穷尽搜索为基础,通过遍历系统模型的状态空间来检测系统模型是否满足给定的性质,该方法与系统或程序以及系统性质的表示有很大的关系^[2]。模型检测技术有两种实现方法,一种是基于对系统状态空间的有穷搜索,另一种是基于自动机理论。在后一种方法中,模型和待检测属性都被转化为自动机的输入语言,并输入到自动机中,通过相应的模型检测算法得出检测结果,从而判定系统是否满足给定属性。一些技术和方法已被提出以减少和压缩状态空间,解决由多个系统模型的并行组合而形成的状态空间爆炸问题(State Space Explosion Problem)^[43]。下面介绍应用模型检测验证系统的过程中涉及的一些关键技术

2.1 建模技术

对系统进行模型检测的过程中,建模技术是关键的技术之一。不论是硬件系统,还是软件系统或通讯协议,模型检测的第一步是对实际系统进行形式化建模,去除与验证属性无关的细节,并将服务规范用形式化方法描述^[5]。建立恰当的模型来表示原系统对验证属性过程的正确性和可行性至关重要。在建模过程中,系统中程序并发执行产生的状态空间爆炸问题,影响了模型检测算法的效率和速度,成为限制模型检测技术广泛应用于各种领域的瓶颈。因此如何建立有效的简化模型,保留与属性相关的细节,已经成为模型检测研究领域面临的重要问题之一^[3-4,6]。

与硬件系统不同,软件系统可能由并行的进程组成从而形成无限状态空间。因此当验证软件系统中一个具有无限状态空间和循环结构的程序是否满足给定属性时,就会出现不确定性,影响了程序验证的效率和可靠性。因此在建模过程中,采用抽象技术简化软件系统,抽取关键性质,例如用有限状态空间或其他方法,将无限状态空间的不确定性转化为具有可确定性的问题,来保证验证规范的完整性和正确性,满足验证过程的需要。因此,模型简化技术和抽象技术应用的程度成为提高模型检测算法的效率和速度的决定因素之一。

2.2 模型转换技术

模型转化技术是在模型检测过程中,在保留系统原有的属性的同时,对实际系统建立模型并进行简化,进而转化为检测器能够接受的形式化表示形式。模型转换执行算法的效率和可靠性直接影响验证过程本身的效率和可信度,因此将实际系统模型和验证规范自动转换为检测器能够接受的形式化表示,同时不影响验证过程的完整性和正确性也是模型检测的重要技术之一。

2.3 验证技术

模型检测的另一个关键技术是验证技术。建立合适的模型后,模型检测器将根据一定的算法对模型进行检测,判断其是否满足设定的属性,如可靠性、完整性、安全性等。验证技术直接影响模型检测的效率和效果,因此应用合理有效的验证方法十分重要。在基于状态空间的模型检测过程中,偏序规约

技术用来解决状态爆炸问题,在不影响检测属性精确度的前提下,简化状态空间的搜索范围,得出检测结果;在基于自动机的模型检测中,主要采用了反例驱动提取技术,调用定理证明器来简化验证过程,得出测试用例,检测系统模型。

2.4 性能优化技术

随着应用系统结构日益复杂,开源软件系统逐渐成为业界组建信息系统的重要部分,在组合之前对各个组件进行形式化验证就变得十分重要。模型检测作为一种重要的形式化方法,是否能得到广泛应用取决于提高算法的性能及效率。在验证服务的三个阶段(单个服务验证、组合服务验证及服务并发执行验证)都需要优化性能,在验证服务是否符合规范的过程中减少运行时间和存储容量。

基于抽象的技术能够提高性能,通过建立适合的模型和规范,模型检测能够较快的对中等规模的系统进行验证。而如何自动生成表示属性的规范也是影响性能的重要因素之一。目前规范大多由人工设置,一定程度上限制了执行效率,因此开发自动输入规范的技术十分必要。

3 模型检测技术在WEB服务验证中的应用

早期的模型检测技术主要侧重于硬件系统的验证,随着模型检测技术在硬件领域发展的不断成熟,这种技术的应用范围也在逐步扩大,包括对通讯协议、安全协议、控制系统、软件系统及Web服务的验证。而如何对Web服务进行形式化描述和验证是目前模型检测技术的一个重要的研究方向。目前已有的采用模型检测技术对组合Web服务进行验证的工具采用了不同的服务描述模型,不同的模型检测方法,检测了不同类型的属性。其中大部分方法采用BPEL语言^[28]或者类BPEL语言对Web服务建模,用SPN^[32]或者基于CCS的检测工具进行验证。

在Web服务执行过程中,服务被不同的实现平台部署在不同的地方,因此,建立一系列数据传输和服务描述的统一标准就十分重要。比如,不同的Web services间使用XML格式的消息进行通讯,XML Schema为XML消息提供了定义类型的格式;而SOAP这样的通讯协议用来传输XML消息,UDDI可以使互联网范围内的企业在互联网上发布自己所提供的服务。Web services间的接口用WSDL来描述,而服务组合的执行序列、服务交互行为用诸如BPEL、WSFL、OWL-S等服务规范来描述^[25-30]。基于这些协议的Web services有不同的实现平台支持,例如.NET、J2EE等。采用模型检测技术验证Web服务的相关协议和描述,主要是将Web服务的过程描述模型(如OWL-S、BPEL4WS)转化成形式化的模型(如有限状态机模型、Petri网模型等),借助模型检测器(Model Checker)等自动化工具验证协议和服务描述的完整性和一致性。其中,服务规约的建立及属性的描述、形式化模型的转换及选择模型检测器对特定属性的验证是模型检测技术实现的关键技术。

在采用模型检测技术对Web服务进行验证的过程中,首先要选择相应的Web服务描述模型并选择合适的模型检测器,将现有的Web服务的过程描述模型转化为model

checker 能够接受的描述形式,接着设定待检测属性的描述方法,并且保证这种表示形式也能够被模型检测器识别,最终由 model checker 通过相应算法对组合服务及其原子服务进行检测,根据其是否满足待检测属性,得出检测结果以及检测报告。

3.1 基于 BPEL 4WS 的模型检测

这类方法中比较典型的是由美国加州大学的开发人员提出的 WSA T (Web Service Analysis Tool)^[10-11]。该系统采用 SPN 作为检测工具,对基于 BPEL 定义的、通过异步 XML 消息交互的组合 Web 服务及服务间的交互活动进行形式化描述,可支持服务规约定义、服务同步和异步特性的分析和验证。BPEL 规范定义了单个服务如何与其他服务交互,但是如何从整体来分析各服务间的交互,验证这些异步通信的单个 Web 服务是一个亟待解决的问题。在 WSA T 中,首先将 Web 服务描述模型(例如 BPEL 4WS 等)转换为 WSA T 的模型语言 Guarded Automata,进而转换为 SPN 的输入语言 PROMELA 进行检验。作为一种中间转换语言,Guarded Automata 用来描述模型的交互,交互活动可以由一组状态集合和改变状态的行为集合表示。该系统提出的模型中,待检测属性用 LTL 来表示,给定一个组合 Web 服务 S 和一个 LTL 表示的属性,通过设定相应的属性,来分析组合 Web 服务是否满足这些性质,从而达到验证 Web 服务的目的^[7-9]。

另一种验证方法是由 Shin NAKAJIMA 等人提出的,该方法采用 SPN 作为验证工具^[12]。为了有效的利用 Web 服务,我们需要将分布在不同网络环境中的服务整合起来,因此需要一种语言来描述不同服务的整合过程,WSFL 正是这样一种服务流描述语言^[27],它描述了服务组合过程中各个服务间的控制流和数据流。验证过程是将 WSFL 描述转化成 SPN 的形式化描述语言 PROMELA^[33];应用程序待检测属性包括各服务节点的可达性、服务流无死锁及应用程序中的特定过程属性等,由 LTL (Linear Temporal Logic) 公式描述,也被输入到 SPN 中;SPN 接受 PROMELA 和 LTL 两种互补的描述信息,通过求二者的交集是否为空来判断系统是否满足特定属性。该方法较好检验了服务在运行前服务流描述规范中的错误,保证了 Web 服务组合过程的正确执行。

Howard Foster 等研究人员则提出了一种基于 FSP (Finite State Process) 的服务流描述语言和验证机制^[13]。FSP 是一种进程代数,用来精确的表示并发系统。该方法首先将 Web 服务整合过程用 MSC (Message Sequence Charts) 的形式描述,并采用 CASE 工具将 MSC 描述转换成 FSP 描述,接着由 LTSA (Labeled Transition System Analyzer) 编译成 LTS 以分析所有允许的状态序列。然后,采用 BPEL 4WS 规范描述相同的 Web 服务流的实现过程,也将其转换成机器易读的 FSP 表示形式。通过比较分别由 MSC 生成的设计过程描述和由 BPEL 4WS 生成的实现描述的 FSP 模型,检查二者的路径等价性,判断在 BPEL 4WS 中是否表示了设计阶段不被要求的过程,而在 MSC 中描述的所有属性是否在执行阶段都被描述。通过分析 FSP 模型生成的路径检查结果,循环执行检查过程,直到没有不一致的触发和服务流死锁出现。因此该方法通

过对早期服务流的设计过程进行检测,避免了设计和执行阶段的不一致性。

Srinin Arayanan 和 Sheila A. McIlraith 借用语义网的概念,采用 DAML-S ontology 来描述 Web 服务的语义。通过将这种服务描述嵌入到一个 Petri 网形式化模型中,进行服务的模拟、测试、集成和验证工作,并且分析了对由 DAML-S 描述的组合服务进行不同限制的情况下的复杂性^[14,44]。他们还提供了实现这种技术的具体工具,将由 DALM-S 描述的 Web 服务作为输入,自动生成一个 Petri 网并进行相应的分析。这种工具既可以作为已有的 Web 服务组合工具的后端处理,也可以作为单独的 Web 服务开发工具使用。可以看出 Petri 网比 FSM 系统具有更强的表达能力,提供了描述过程交互规范的原语。在此基础上,X. Yi 和 K. J. Kochut 提出了一种扩展的 Petri 网模型 CP-nets,它提供了描述数据类型的定义和数据的操作的原语,允许对于并发和过程间同步的描述,因此比 Petri 网更精确的描述了系统,更好的适应了服务流的验证需要^[19]。BPEL 4WS 过程被转化为一个等价的 CP-nets 模型,用 Petri 网的分析工具 CPN 来分析该过程,通过形式化验证在设计过程中尽早发现错误,确保了组合过程的正确性。CP-nets 分析技术能够分析组合服务的很多重要属性,包括有界性、活跃性及可达性,这些都是验证服务组合正确性的一般标准。

3.2 基于 OWL-S 的模型检测

以上系统都是一类以 BPEL 作为 Web 服务描述模型的方法,这些方法都把原子服务作为黑盒处理。这类面向过程的模型较好地跟踪和验证了各原子服务的时态属性,但是由于在规范中没有表示出原子服务的内部服务,因此很难检测包含原子服务输出的更详细的属性,检测时态属性并不能取代内部属性的检测。针对该问题,Arizona State University 软件实验室研究人员提出了一种基于 OWL-S (Web Ontology Language for Web Services) 和模型检测器 BLAST 的 Web 服务检测方法^[15]。这种方法将模型检测与测试技术相结合。一方面,将 OWL-S 规范转化为类 C 代码的描述,采用扩展的 BLAST 模型检测器对其并发性进行检测;另一方面,将部分需要检测的特性嵌入到转化后的 C 代码描述中,并根据模型检测结果自动生成测试用例。该方法把 BLAST 应用到了检测 Web 原子服务的过程中,扩展了 BLAST 处理并发性的能力,更好的生成了测试用例。具体过程包括:将 OWL-S 模型转换为类 C 语言;将待检测属性嵌入到转换后的规范语言中;将规范语言输入到 BLAST 中;进行模型检测分别生成正、反测试用例。

布尔抽象和反例驱动提取技术是模型检测器检测软件系统和 Web 服务的主要技术,其中抽取由抽象的精确度来决定^[26]。BLAST 中采用的主要技术是 Lazy Abstraction^[35,41],由反例驱动提取和不均匀精确度抽象技术组成。这种检测工具使用反例驱动自动提取技术来建立一个抽象模型,检测模型的安全属性 Blade^[34]是另一种软件系统模型检测器,它在 lazy abstraction 技术的基础上引用了 Program Slicing 的思想,提出了程序切片 (Proof Slicing) 技术,精度信息被 proof

slices 表示^[16, 36, 40]。这些 proof slices 可以被将来的 proof 有选择的重用。通过在将来的搜索过程中有选择复用已存在的 proof slices, 减少了回溯过程中的复杂度和调用 theorem provers 的频率^[25]。同时 proof slices 集合也会随着将来反例的生成和去除不断的更新。实验数据证明, 和 BLAST 相比, Blade 在运行时间和存储空间需求方面都有较好的改进。Blade 采用 OWL-S 作为 Web 服务描述模型, 较好的实现了将模型检测技术应用到 Web 服务中并自动生成测试用例的过程中。

3.3 其他方法

廖军等学者提出了基于 Pi-演算的 Web 服务组合的描述和验证, 在沿用 XLANG 中所定义的 Web 服务与 Pi-演算元素之间对应关系的基础上, 建立了服务组合的 Pi-演算模型^[17]。采用 Pi-演算的方法在对动态变化的系统建模方便具有较大的优势, 在 MWB (Mobility Workbench) 模型检测工具的支持下, 对组合服务进行推演和验证。

Mariya Koshkina 等人引入了基于 BPEL 4WS 的 BPE-

表 1 几种检测方法的比较

Table 1 Comparison of various method of model checking

WEB 服务描述模型	形式化模型表示	检测属性类型	采用的模型检测器或分析工具	提出者
基于 BPEL 4WS	LTL	支持服务规约定义、验证服务同步和异步特性	SPN	W S A T
基于 W SFL 服务流	LTL	各服务节点的可达性、服务流的无死锁及应用程序的特定过程属性	SPN	Shin N A K A J M A 等
基于 BPEL 4WS 和 MCS	FSP	检测不一致的触发和服务流死锁的出现	FSM	Howard Foster 等
基于 DAML-S	Petri-net 或 CP-nets	分析组合服务的复杂性和正确性	CPN	Srini Narayanan 等
基于 OWL-S	CTL	检测服务的并发并自动生成测试用例	BLAST 或 BLADE	A S U 软件实验室
基于 OWL-S	LTL	检测服务流和数据流的正确性和交互性	SPN	A n u p r i y a A n k o l e k a r 等
基于 Pi-calculus		推演和验证组合服务	MWB	廖军等
基于 BPEL 4WS	BPE-calculus	对各服务间的协作进行建模和验证	PAC 和 CWB-NC	M a r i y a K o s h k i n a 等

calculus, 并使用 PAC (Process Algebra Compiler) 和 CWB-NC (Concurrency Workbench of the New Century) 工具对 Web 服务协作进行建模和验证^[18]。BPEL 4WS 语言被转化为形式化定义的 BPE-calculus 的语法和语义, 并作为 PAC 的输入; PAC 产生输出模块并整合到 CWB 中, 为 Web 服务协作提供了一种基于 BPE-calculus 的验证工具。下表对各种检测方法从不同角度进行比较^[20-21]。

4 对现存问题的相关探讨

从目前模型检测在 Web 服务中的应用可以看出, 现有的服务模型检测技术主要涉及到了服务模型的形式化描述方法, 但由于受限于 Web 服务规约的不完善性, 模型本身的描述能力以及模型检测器的验证能力有限, 目前的研究还存在以下几方面的问题:

4.1 协同式在线模型检测技术

如何对服务及服务组合进行动态建模与验证, 实现服务动态性和自适应改变是服务验证时面临的重要问题, 例如当服务模型中的组合、检测属性和检测过程改变时, 形式化模型也能够相应及时做出改变。服务的动态发现、绑定及协作是面向服务体系架构的一个基本特点, 在开放的 Internet 环境下, 存在大量满足相同功能需求的服务, 根据服务质量、系统性能、价格等各方面的需求不同, 可以实时地选择不同的服务组

合, 服务之间动态地建立起相互合作的关系。在系统运行过程中, 不同的应用模块应能够从不同的地方被动态调用和激活, 甚至在系统运行状态改变系统的语义描述^[37-38]。

而如何构建一个在线服务组合检测框架, 实现动态的对服务进行建模与验证是该领域的难点之一。模型的建立和转



图 1 模型检测在服务验证框架的应用

Fig 1 Application of model checking in Web service verification & validation framework

化缺少自动化的机制及工具支持。现有的研究多是采用离线的、大量人工介入的方式完成服务形式化模型的建立以及向

模型检测器的输入 因此我们需要定义一种机制,使服务发布、发现、组装成为一个自动的过程,形成一种真正面向服务的整体架构 图 1 表示了模型检测应用于协同式W eb 服务V &V 框架中的情况

4 2 基于多种服务规范模型的形式化模型转换

W eb 服务协议栈中的各个协议之间是相互补充的关系,对各个协议的分析相对孤立,如服务流程以每个服务的定义为基础,在服务的执行过程中,流程中抽象的每个节点描述必须映射并绑定到具体的服务实现上,如果仅仅分析一个流程定义本身而忽略服务节点的特性,势必会影响验证结果 单个原子服务内部有其自己的控制流,而新的组合服务间又存在新的控制流,模型检测的目的之一就是检测二者间可能存在的冲突 因此在W eb 服务验证过程中,我们不仅要验证组合W eb 服务间的流程是否满足检测时态属性,还应检测原子服务的执行过程及检测其数据属性,同时还应考虑如何检测数据流和控制流在进行交互时,可能出现的错误 这是研究验证组合W eb 服务过程中有待解决的问题之一^[31]

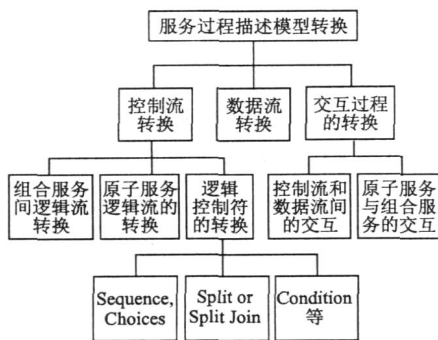


图2 服务过程描述模型转换过程

Fig 2 Transformation from service process model to formal model

如何有效地将W eb 服务的过程描述模型(比如BEP L4W S, OWL-S 等)转换为模型检测器能够接受的表示形式,包括组合服务间的控制流、原子服务内部的控制流和数据流 通过将转换后的表示形式及属性描述输入到检测器中来验证系统是否满足相应的时态属性和数据属性等 这也是模型检测技术应用到组合W eb 服务的验证过程中的一个研究方向 在下一步的工作中,将对相应的服务描述规范和协议进行转换,如 OWL-S, BPEL4W S 及 WSDL 等,采用 BLSAT 或 BLADE 作为模型检测器 在这类协议中,可能存在多个并发的线程间的异常交互 服务描述模型向形式化描述模型的转换包括对数据流、控制流及服务间的交互过程的转换 其中对控制流的转换包括组合服务间的控制流、原子服务内部的控制流及各种逻辑控制符(例如Sequence, Choice, Condition 等)的转换;在对服务的交互过程转换过程中,主要处理原子服务与组合服务控制流间的冲突,检测控制流和数据流间的可能的交互冲突 图2 给出了服务描述模型的转换步骤

4 3 基于启发式的不确定性问题探讨

在模型检测应用到组合W eb 服务的过程中,服务本身及验证过程都存在不确定性 当验证具有循环结构、递归函数级动态数据对象的程序时,不确定性(U ndecidability)问题影响了验证速度和效率 在目前应用的方法中,启发式模型检测方法较好的适应了W EB 服务的特点 这种方法结合了建立模型、程序语言设计、测试及仿真领域的研究,主要是考虑如何将人工智能技术结合到模型检测过程中来解决不确定性 W eb 服务在组合过程中具有不确定性和动态性等特点,验证过程应尽可能符合服务本身的性质,对服务的变化做出及时调整 启发式模型检测在传统技术基础上加入了人工交互界面并建立自学习理论对知识库进行及时更新,满足了W eb 服务的要求

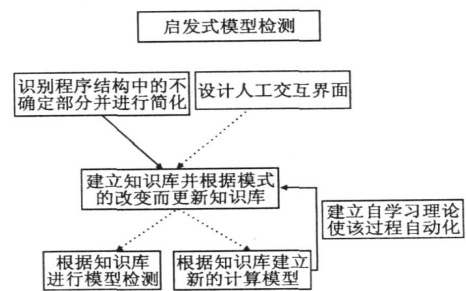


图3 启发式模型检测的执行过程

Fig 3 Process of heuristic model checking

主要思想包括:识别程序结构中的不确定部分,并进行相应的简化和抽象使其具有确定性;设计人工交互界面,以便在检测过程中进行人工辅助;在前两步的基础上建立知识库,主要包括能识别具有确定性程序结构的模式,建立在相应程序结构之上的规范及与程序结构和规范相关的抽象技术,知识库根据程序结构模式的变化进行相应更新;根据知识库结合仿真和测试技术进行模型检测;在知识库的基础上建立新的计算模型;建立自学习理论使得整个执行过程实现自动化 图3 表示了该思想

5 结束语

总之,作为一种新的计算模式,面向服务的计算已经开始在不同的应用领域发挥越来越大的作用 基于Internet 的应用的迅速发展,改变了整个人类社会生活方式,创造了巨大的社会和经济回报,在不久的将来将成为电子商务、电子政务等新的商业模式的主要支撑平台 而如何保证W eb 服务的可信性就成为一个关键的问题 今后W eb 服务测试的研究重点将集中在如何建立完善的理论模型并将其应用到实际的W eb 服务的验证和确认中,如何开发具体的测试工具以解决现有的各种W eb 服务测试问题 例如怎样动态测试组合W eb 服务、验证服务的正确性和完备性等 确保W eb 服务的可靠性已经成为W eb 服务是否能真正满足各个应用领域需要的一个重要的决定因素,因此在今后的工作中,应将理论研究和实际应用相结合,改善模型检测器的开发效率和质量,提高组合W eb

服务验证和测试的可信度

References

- [1] Lin Huimin, Zhang Wen-hui Model checking: Theories, techniques and applications[J]. Acta Electronica Sinica, 2002, 30(12): 1907-1912
- [2] Clarke E M, Grumberg O, Peled D A. Model checking[M]. The MIT Press, Cambridge, MA, USA, 2000
- [3] Emerson E A, Sistla A P. Symmetry and model checking[A]. Lecture Notes in Computer Science 697-5th International Conference on Computer Aided Verification (CAV '93)[C]. Berlin: Springer-Verlag, 1993. 463-47A.
- [4] Clarke E M, Grumberg O, Long D E. Model checking and abstraction[J]. ACM Transactions on Programming Languages and Systems, 1994, 16(5): 1512-1542
- [5] Gerard J Holzmann, Doron Peled. An improvement in formal verification[C]. 7th IFIP WG6.1 International Conference on Formal Description Techniques V II (FORTE 1994). London: Chapman and Hall, 1995, 197-211.
- [6] Clarke E M, Long D E, McMillan K L. Compositional model checking[A]. 4th IEEE Symposium on Logic in Computer Science (LICS '89)[C]. Los Alamitos: IEEE Computer Society, 1989, 353-362
- [7] Fu X, Bultan T, Su J. A analysis of Interacting BPEL Web services[C]. In: Proceeding of the 13th International World Wide Web Conference, 2004, 621-630
- [8] Fu X, Bultan T, Su J. Model checking XML manipulating software[C]. In Proceedings of the 2004 ACM SIGSOFT International Symposium on Software Testing and Analysis, 2004, 252-262
- [9] Fu X, Bultan T, Su J. Realizability of conversation protocols with message contents[C]. In: Proceedings of the IEEE International Conference on Web Services, 2004, 96
- [10] Fu X, Bultan T, Su J. WSA T: a tool for formal analysis of web services[C]. In 16th International Conference on Computer Aided Verification, 2004, 510-514
- [11] Fu X, Bultan T, Su J. Model checking interactions of composite Web services[R]. Tech Report 2004-05, Computer Science Department, University of California at Santa Barbara, 2004
- [12] Nakajima S. Model-checking verification for reliable web service[C]. In: Proc OOPSLA '02 Workshop on OO Web Services, 2002
- [13] Foster H, Uchitel S, Magee J, et al. Model-based verification of web service compositions[C]. In: Proc ASE'03, 2003
- [14] Narayanan S, McIlraith S. Simulation verification and automated composition of web services[C]. In: Proc WWW '02 ACM, 2002
- [15] Huang H, Tsai W T, Paul R, et al. Automated model checking and testing for composite Web services[C]. 8th IEEE International Symposium on Object-oriented Real-time Distributed Computing (ISORC), Seattle, May 2005, 300-307.
- [16] Huang H, Tsai W T, Paul R. Proof slicing with application to model checking web services[C]. In: Proceedings of the 8th IEEE International Symposium on Object-oriented Real-time Distributed Computing, 2005, 292-299
- [17] Liao Jun, Tan hao, Liu jin-de. Describing and verifying Web service using Pi-Calculus[J]. Chinese Journal of Computers, 2005, 28(4): 635-643
- [18] Koshkina M, van Breugel F. Modeling and verifying Web service orchestration by means of the concurrency workbench[J]. ACM SIGSOFT SEN, 2004, 29(5): 1-10
- [19] Yi X, Kochut K J. A CP-nets-based design and verification framework for Web services composition[C]. Proceedings of the IEEE International Conference on Web Services, March, 2004
- [20] Ankolekar A, Paolucci M, Sycara K. Spinning the OWL-S process model towards the verification of the OWL-S process models[C]. In: Semantic Web Services: Preparing to Meet the World of Business Applications Workshop at ISWC 2004
- [21] Ankolekar A, Paolucci M, Sycara K. Towards a formal verification of owl process models[C]. 4th International Semantic Web Conference, 2005. To appear
- [22] Ball T, Rajamani S K. Automatically validating temporal safety properties of interfaces[C]. In: Proceedings of the 8th International SPIN Workshop on Model Checking of Software, 2001, 103-122
- [23] Gruhn V, Laue R. Using timed model checking for verifying workflows[C]. In: Computer Supported Activity Coordination, 2005, 75-88
- [24] Grumberg O, Long D E. Model checking and modular verification[J]. ACM Transactions on Programming Languages and Systems, 16: 843-872
- [25] Simple Object Access Protocol (SOAP) 1.1 [EB/OL]. W3C Note 08 May 2000. Available at: <http://www.w3.org/TR/SOAP/>.
- [26] Web Services Description Language (WSDL 1.1) [EB/OL]. W3C Note 15 March, 2001. Available at: <http://www.w3.org/TR/WSDL/>.
- [27] Web Services Flow Language (WSFL 1.0) [EB/OL]. IBM Software Group, May 2001.
- [28] Specification: Business Process Execution Language for Web Services [EB/OL]. <http://www-128.ibm.com/developerworks/library/ws-bpel/>, May, 2003
- [29] OWL-S: Semantic markup for Web services [EB/OL]. <http://www.daml.org/services/owl-s/1.1B/owl-s.pdf>, 2004
- [30] W3C Extensible Markup Language (XML) page [EB/OL]. <http://www.w3.org/XML>.
- [31] Huang Hai. Model Checking: novel techniques and applications [D]. PhD Thesis, 2005
- [32] Holzmann G J. The SPIN model checker: primer and reference manual[M]. Addison-Wesley Professional, 2003
- [33] Holzmann G J, Peled D. The state of SPIN [C]. In: Proceedings of the 8th International Conference on Computer Aided Verification, Springer, 1996, 385-389
- [34] Blade: A su proof slicing model checker [EB/OL]. <http://asusrl.eas.asu.edu/blade/>, 2005

- [35] BLAST model checker[EB/OL]. <http://embedded.eecs.berkeley.edu/blast/>, 2005.
- [36] Mark Weiser. Program slicing[J]. IEEE Transactions on Software Engineering, 1984, 10(4): 352-357.
- [37] Tsai W, Cao Z, Chen Y, et al. Web services-based collaborative and cooperative computing [Z]. In Workshop on Cooperative Computing, Internetworking, and Assurance.
- [38] Tsai W, Chen Y, Paul R, et al. Cooperative and group testing in verification of dynamic composite Web Services[Z]. In Workshop on Quality Assurance and Testing of Web-Based Applications, in Conjunction with COMPSAC, 2004. to appear.
- [39] Beyer D, Chlipala A J, Henzinger T A, et al. Generating tests from counterexamples[C]. In: Proceedings for the 26th International Conference on Software Engineering, 2004, 326-335.
- [40] Henzinger T A, Jhala R, Majumdar R, et al. Abstractions from proofs [C]. In: Proceedings of the 31st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, 2004, 232-244.
- [41] Henzinger T A, Jhala R, Majumdar R, et al. Lazy abstraction [C]. In: Proceedings of the 29th Annual Symposium on Principles of Programming Languages, 2002, 58-70.
- [42] Larsen K G. Modal specifications [C]. In: Proceedings of the 1989 International Workshop on Automatic Verification Methods for Finite State Systems, Springer, 1989, 232-246.
- [43] McMillan K L. Symbolic model checking: An approach to the state explosion problem [Z]. Kluwer Academic, 1993.
- [44] Paolucci M, Srinivasan N, Sycara K, et al. Towards a semantic choreography of Web Services: from WSDL to DAML-S[EB/OL]. <http://pericles.cimds.ricmu.edu:8080/wsd12owls/wsd12danls.pdf>, 2004.

附中文参考文献

- [1] 林惠民, 张文辉. 模型检测: 理论、方法和应用[J]. 电子学报, 2002, 30(12): 1907-1912.
- [17] 廖军, 谭浩, 刘锦德. 基于Pi-演算的Web服务组合的描述和验证[J]. 计算机学报, 2005, 28(4): 635-643.