软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2011,22(7):1413−1425 [doi: 10.3724/SP.J.1001.2011.03820]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于消息日志的 Web 服务接口业务协议挖掘*

李 翔+, 怀进鹏, 刘旭东, 孙海龙, 曲先洋

(北京航空航天大学 计算机学院,北京 100191)

Web Service Business Protocol Mining Based on Message Logs

LI Xiang+, HUAI Jin-Peng, LIU Xu-Dong, SUN Hai-Long, QU Xian-Yang

(School of Computer Science and Engineering, BeiHang University, Beijing 100191, China)

+ Corresponding author: E-mail: lixiang@act.buaa.edu.cn

Li X, Huai JP, Liu XD, Sun HL, Qu XY. Web service business protocol mining based on message logs. *Journal of Software*, 2011,22(7):1413−1425. http://www.jos.org.cn/1000-9825/3820.htm

Abstract: A Web service business protocol is used to describe the external behavior of a service and plays an important role in the service discovery, composition, verification, runtime service trustworthy guarantee, and so on. Presently, some research has been done on discovering the Web service business protocol from the invocation logs. Most of these works focused on the control-flow of Web service business protocols that give a temporal constraint among the operations of Web service. However, the data constraints and the consistency between the data-flow and the control-flow are also important and have not received enough attention. This paper studies the Web service business protocol from the service invocation logs and focuses on mining the relations, or the constraints between the message values and service operations. This paper proposes a Petri-net based model, called Business Protocol Net (simply, BPN), to describe the behavior of a service. Based on this model, a mining framework is proposed to automatically generate the BPN model from message traces. Experimental results illustrate that the method is effective in discovering the Web service business protocol from invocation logs.

Key words: Web service; business protocol; protocol discovery; process mining

摘 要: Web 服务接口的业务协议描述了 Web 服务的外部行为,对于 Web 服务的复用具有重要意义,可以作为服务发现、组合、验证和运行期可信保障等方面的重要基础.目前,已有一些工作研究了 Web 服务的协议发现问题,即从 Web 服务的调用消息日志中挖掘 Web 服务接口的业务协议.但已有方法主要关注服务的控制流约束,忽略了数据流约束以及数据流和控制流的相互约束.针对这一问题,研究了如何从 Web 服务的调用日志中自动挖掘 Web 服务接口,并侧重综合考虑 Web 服务的数据流和控制流.首先扩展了传统 Petri 网,提出了一种增加了数据流描述的 Web 服务接口模型——BPN(business protocol net)模型.在此基础上,进一步提出了一种自动化的挖掘框架,可以从 Web 服务调用消息记录中合成 Web 服务的 BPN 表示.最后,通过仿真实验验证了该方法的有效性.其结果表明,所提出的挖掘算法是正确而有效的.

关键词: Web 服务;业务协议;协议挖掘;流程挖掘

中图法分类号: TP301 **文献标识码**: A

* 基金项目: 国家高技术研究发展计划(863)(2007AA010301, 2006AA01A106, 2009AA01Z419)
收稿时间: 2009-06-16; 修改时间: 2009-09-11; 定稿时间: 2009-12-07

面向服务的体系结构(service oriented architecture,简称 SOA)架构和服务计算技术为企业信息系统的复用、重组和扩展提供了一个最佳实践[1],其核心思想是,将企业信息系统的模块封装并发布成为 Internet 上可访问的标准的 Web 服务,然后针对新的应用需求或环境变化重新组合这些服务使其形成新的应用软件,从而有效地降低了软件的成本,提高了开发效率.目前,自动化服务组合技术引起了学术界的广泛重视,已有大量自动服务组合技术被提了出来[2−5].然而这些技术难以应用在实际场景中,究其原因,这些技术都假定 Web 服务自身提供了充分的接口说明,如描述交互行为的自动机模型、描述功能的谓词逻辑模型或描述执行语义的本体等.这些接口说明刻画了 Web 服务的外部行为,称为 Web 服务接口的业务协议.对于一些有状态的服务来说,其内部用类似 BPEL 的工作流语言进行描述,具有复杂的业务流程,其接口的业务协议说明更为重要.除了支持服务的自动化组合外,Web 服务接口的业务协议对于服务的发现、验证、测试和服务可信保障,以及对于辅助开发人员理解服务的语义等方面,也都具有重要的意义.

然而目前,Web 服务接口的说明非常有限.一方面,目前的互联网环境中少有 Web 服务提供形式化的接口说明,即便提供也仅限于语法级别(如 WSDL)和自然语言的功能描述[6],无法支持自动化的服务组合;另一方面,目前还没有较为成熟的模型、描述语言和相应技术用以支持服务开发者手动提取或编写 Web 接口说明,即便采用形式化工具加以辅助,手工方式依然费时费力且容易出错,难以完成这一任务.因此,自动化的提取 Web 服务的接口及其业务协议模型是自动化服务组合技术的基础,对推动目前服务计算研究的进展具有重要意义.

Web 服务接口的业务协议主要是指服务的外部行为描述,它屏蔽了 Web 服务的实现细节,从较高抽象层面刻画了 Web 服务接口的名称、消息、行为以及功能方面的约束信息.文献[7]中将 Web 服务接口的规格说明分解为 3 个层次:

- 首先是语法约束(signature constraint),主要描述了服务的操作名称和消息类型方面的约束;
- 其次是一致性约束(consistency constraint),描述了服务交互过程中数据取值对调用操作的约束;
- 最后是协议约束(protocol constraint),描述了服务交互过程中操作之间的时序约束.

一般来说,软件组件的接口说明主要包括两方面内容:控制流,对于 Web 服务的接口来说,对应于上文的协议约束;数据流,对于 Web 服务的接口来说,对应于上文的消息约束.然而,这两种接口说明并不是完全分离的,数据的取值往往会决定服务的下一步操作.比如,资源遍历类型的服务通常会提供两个操作 hasNext 和 next,前一操作返回当前是否还有资源,后一操作则返回资源信息.一般地,如果 hasNext 在返回 false 的情况下调用 next 操作将会造成错误.在服务内部实现中,表示当前资源索引和资源总量的数据会决定访问下一资源的操作是否可以调用.这也就是说,数据流和控制流会有紧密联系,这种联系一般通过条件或断言的方式给出对应于上文的一致性约束.目前,自动获取 Web 服务消息约束(主要表现为 WSDL)的技术已经完全成熟,对于协议层也有部分工作给出了一定的结果[8−11].然而,综合处理这两方面约束从而获得一致性约束的研究工作目前还很少,这一考虑启发我们开展了本文工作.

本文关注服务接口的业务协议挖掘,并重点研究基于服务调用消息日志(message log)的服务接口业务协议自动挖掘问题.即已知 Web 服务的调用消息日志,如何自动化地提取 Web 服务的控制流和数据流的约束.通过扩展 Petri 网模型,提出了一种 BPN(business protocol net)模型,用以描述 Web 服务接口的业务协议.BPN 可以兼顾描述控制流和数据流,以及两者之间的约束关系.在此基础上,提出了一种根据 Web 服务调用日志自动生成 BPN 模型的方法.首先,我们利用成熟的程序参数分析工具 Daikon[12],并将其扩展用于分析业务流程.然后,我们利用经典的流程挖掘算法——α算法[13]获取 Web 服务接口 BPN 模型的控制流.最后,我们将前两步的分析结果合并到 BPN 模型中,用于生成数据流以及数据流与控制流之间的约束,形成完整的 BPN 模型,从而得到信息更为丰富且更为严格的 Web 服务接口的业务协议说明.

本文的主要贡献在于:

1) 提出一种新的基于 Petri 网的 Web 服务接口模型,在常规 Petri 网的基础上增加数据库所和接口描述,并通过条件变迁对数据库所取值进行判断,使数据流可以有效地约束控制流,从而可以表达服务参数取值对服务交互过程的控制;

2) 提出一种服务接口业务协议的自动挖掘框架,并在结合 Daikon 工具和α算法的基础上,进一步提出了一种从服务调用日志中自动生成 BPN 模型的算法;
3) 通过仿真实验验证了本文方法的有效性和实际运行效率,并将其应用于一个真实案例.实验结果表明,本文方法可以有效地获取 Web 服务接口的业务协议,算法的时间开销在实践中是可以接受的.

本文第 1 节通过一个简单案例说明服务接口的业务协议挖掘问题.第 2 节中给出相关研究工作的分析和比较.第 3 节形式化地介绍我们提出的基于 Petri 网的服务接口模型.作为本文的核心,第 4 节讨论 Web 服务接口业务协议挖掘问题,提出一种自动化挖掘机制的框架和算法,并分析这一算法的复杂性.第 5 节提出应用案例分析和实验分析,对我们方法的有效性进行验证评估.最后,第 6 节对本文工作进行总结和展望.

1 应用案例

本节通过一个银行贷款的应用场景来说明服务接口挖掘的问题.应用案例如图 1 所示,我们使用 BPMN 标准图元[14]进行描述,而不是采用被广泛接受的 BPEL 语言描述,后者不适合描述多方参与的业务流程.银行贷款业务是一项复杂业务,由 4 方构成,包括用户 Customer、贷款服务中心 Loan Center、贷款审批中心 Loan Approver、银行信用中心 Credit Center.本案例中,首先由用户向银行提交贷款申请,从而触发流程的启动.银行贷款申请流程接收在收到请求之后首先判断贷款的额度,如果小于 10 000 元,则调用本行及联盟银行信用中心提供的信用评估服务,获取贷款申请者的信用等级,信用等级为“高”时直接批准用户的贷款申请;如果额度大于 10 000 元,则将用户的身份信息发送给贷款审批流程,由银行信贷人员进行人工审查;服务结束前,最终的审批结果将反馈给用户.

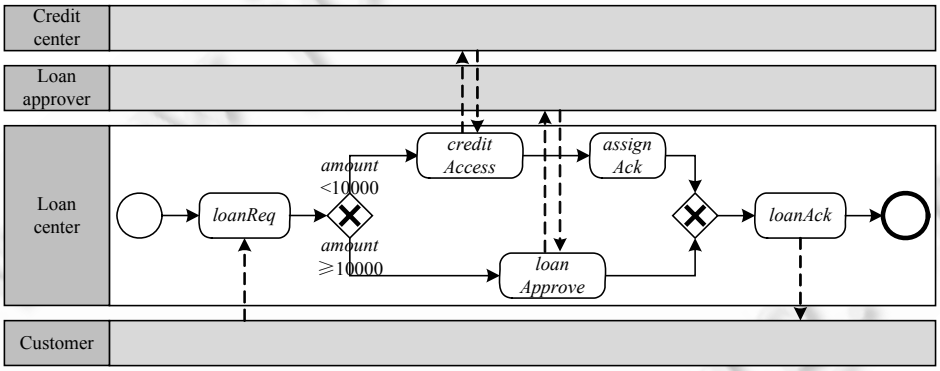


Fig.1 A BPMN for a bank loan service case

图 1 银行贷款应用案例的 BPMN 表示

目前,大多数 Web 服务容器或支持服务的工作流管理系统都提供了服务调用日志管理功能.通过这一功能,所有服务调用的 SOAP 报文会被记录下来.我们对调用日志进行筛选过滤,可以获得同一服务的所有调用记录.进一步分析这些记录,则可从中抽取出 Web 服务调用的具体操作和输入/输出参数.表 1 描述了 4 条银行贷款服务的调用记录,其中,Opr.列记录了操作名,而 Msg.列记录了操作对应的参数.为了便于书写,我们用 x 表示贷款额度,u 表示用户编号,c 表示用户信用等级,r 表示人工审查结果,a 表示贷款审批结果.每条记录经过分析可以获得其中的操作名和参数信息,如日志第 1 条调用记录描述了调用序列:*loanReq*(x=8000,u=001),*creditAcc*(u=001,c=low),*loanAck*(a=false);而第 2 条调用记录描述了调用序列:*loanReq*(x=20000,u=002),*loanAppr*(u=002,x=20000,r=true),*loanAck*(a=true).从中我们可以看到,操作的执行顺序与用户输入数据紧密相关,单独使用数据流约束或控制流约束,无法准确地描述银行贷款服务的调用约束.本文后续将重点讨论如何从这些记录中挖掘 Web 服务接口的调用约束,即 Web 服务接口业务协议.

Table 1 Four invocation message logs of the bank loan service

表 1 4 个贷款服务的调用消息记录示例

	Opr.	Msg.		Opr.	Msg.		Opr.	Msg.		Opr.	Msg.
1	*loanReq*	*x*=8000, *u*=001	1	*loanReq*	*x*=20000, *u*=002	1	*loanReq*	*x*=10000 *u*=002	1	*loanReq*	*x*=50000, *u*=001
2	*creditAcc*	*u*=001, *c*=low	2	*loanAppr*	*u*=002, *x*=20000, *r*=true	2	*creditAcc*	*u*=002 *c*=high	2	*loanAppr*	*u*=001 *x*=50000 *r*=false
3	*loanAck*	*a*=false	3	*loanAck*	*a*=true	3	*loanAck*	*a*=true	3	*loanAck*	*a*=false

2 相关工作

对于 Web 服务接口挖掘,目前已有一些工作取得了一定进展,本节从 3 个方面对相关工作进行分析和比较.

- Web 服务模型.对 Web 服务的接口和外部行为建模是 Web 服务应用得以进一步进行理论研究的基础,目前已有大量形式化模型被提了出来,其中最广泛采用的是自动机模型[2]、Petri 网模型[15]和基于本体的模型[4].除此之外,标签转移系统(label transition system,简称 LTS)和进程代数也受到较多关注.文献[16]介绍了这一领域的研究现状.由于我们更侧重考虑多方参与的复杂业务流程,这就要求模型应适合描述带有并发语义的业务流程.本文重点关注基于 Petri 网的 Web 服务模型;
- 软件规格挖掘.文献[8]中第一次提出了对软件模块接口规格说明自动挖掘的问题.在此基础上,文献[10,11,17]等利用不同技术对这一问题作了更深入的研究.文献[10]中提出,按照分析技术和分析数据的来源可以给出软件规格挖掘的两种分类:动态分析挖掘和静态分析挖掘,客户端分析和服务端分析.对于 Web 服务接口的挖掘则主要使用服务端的动态分析技术,这也是本文关注的重点;
- 业务流程挖掘.基于业务序列反向推导业务流程已有大量较为成熟的研究结果.文献[13]中,Van Der Aalst 等人提出了α算法,成为工作流挖掘领域中的经典算法.在α算法的基础上,Van Der Aalst 等人开发了流程挖掘系统 ProM[18].文献[19]对α算法作出了进一步的改进,获得了对短循环业务流程模式的支持.近期这些成果也逐步应用于 Web 服务接口的挖掘中,文献[9]中通过对服务调用日志的挖掘,提出了一种服务接口的概率模型,并利用决策论方法解决 Web 服务接口的协议层约束问题.

综上所述,Web 服务接口挖掘及其相关领域已受到广泛关注,并取得了一定成果.但目前的大多数工作仍然侧重于服务的控制流,即协议约束层面,而较少涉及数据流以及数据流与控制流的相互制约关系.这一现状启发了我们开展本文的工作.

3 Web 服务接口和日志模型

本节给出 Web 服务接口的业务协议模型和 Web 服务调用消息日志模型.首先给出本文所用形式化模型的基础和概念,在此,我们假定读者熟悉 Petri 网.一个 Petri 网可以用四元组(P,T,F,l)表示,其中:P 是库所的有限集,图中用圆圈表示;T 是变迁的有限集,图中用方块表示;$F\subseteq(P\times T)\cup(T\times P)$是关系集合,用带箭头的弧表示;$l:(P\cup T)\to L$ 是标签映射函数,其中,L 表示标签集合.对于一个普通 Petri 网 N,(N,M)称为标记 Petri 网,其中,$m:P\to N$ 是一个标记,对于任意属于 M 的 m;$m(p)$表示库所 p 中的资源数目.初始标记表示为 m_0,终止标记集表示为 M_F.令 x 为任意库所 $p\in P$ 或变迁 $t\in T$,$^\bullet x$ 表示 x 的前驱集合,$x^\bullet$表示 x 的后继集合.通常,如果$^\bullet p=\varnothing$,称 p 为开始库所;如果 $p^\bullet=\varnothing$,则称 p 为终止库所.对于一个标记 m,一个变迁 $t\in T$ 是使能的,当且仅当$\forall p\in{}^\bullet t:m(p)>0$ 成立,记为 $m[t>$.如果 t 是使能的,t 可以触发并导致标记变迁到 m',记为 $m[t>m'$,其中,对于所有库所,如果 $p\in{}^\bullet t\backslash t^\bullet$,则 $m'(p)=m(p)-1$ 成立;如果 $p\in t^\bullet\backslash{}^\bullet t$,则 $m'(p)=m(p)+1$;其他情况下,$m'(p)=m(p)$.当存在一个触发序列 $t_1,t_2,\ldots,t_n$,使得 $m[t_1>m_1[t_2>\ldots m_{n-1}[t_n>m'$,则称标记 m'是从 m 可达的.我们用 $R(N,M)$表示从 m 可达的标记的集合.

3.1 BPN模型

基于本文前面的讨论,Web 服务接口的业务协议应覆盖消息层、一致约束层和协议层等多个层次.综合以上考虑,在扩展标准 Petri 网的基础上,我们定义了 Web 服务接口业务协议的形式化模型——业务协议网

BPN(business protocol net).具体模型定义如下:

定义 1(BPN). 一个业务协议网 N 是一个五元组(P,T,F,I,O),其中:

- $P=P_C\cup P_D$,P 是库所的有限集合,其中,P_C 是控制库所集合,P_D 是数据库所集合.令 D 表示 P_D 中元素值域的集合,对于任意 $p\in P_D$,D_p 是 p 的值域,表示 p 的所有可能取值,$d_p\in D_p$ 表示 p 的一个具体取值,记为 $p.d$;
- $T=T_C\cup T_O\cup T_S\cup T_A$,$T$ 是变迁的有限集合,其中,T_C 表示条件变迁集合,T_O 表示 Web 服务操作的变迁集合,T_S 表示同步变迁,T_A 表示赋值变迁.$C:D\rightarrow boolean$ 是条件集合,对于任意条件变迁 $t\in T_C$,$p\in P_D$ 且 $p\in{}^\bullet t$,有 $c:D_p\rightarrow boolean$ 表示 t 上的条件,记为 $t.c$;
- $F\subseteq(P\times T)\cup(T\times P)$是弧的集合;
- $I\subseteq P$ 是输入库所集合,对于任意 $p\in I$,有${}^\bullet p=\varnothing$;
- $O\subseteq P$ 是输出库所集合,对于任意 $p\in O$,有 $p^\bullet=\varnothing$.

在 BPN 模型中,为了同时兼顾 Web 服务的控制流约束和数据流约束,我们在原有控制库所的基础上引入了数据库所 P_C.具体的图元表示采用了类似文献[3]中模型的图元表示,采用圆圈表示数据库所,菱形表示控制库所,变迁、弧以及标记的图元与传统 Petri 网一致.这里,变迁分为几类不同的表示几种不同的活动,包括条件判断、服务操作调用、同步和赋值操作.

除此之外,受文献[15]的启发,我们在传统 Petri 网的基础上引入了两个特殊集合:输入库所集 I 和输出库所集 O,分别对应于服务的输入接口和输出接口.用虚线框表示一个服务的接口,显然,BPN 的接口与服务 WSDL 是对应的.实际上,WSDL 文档规定了 Web 服务接口中操作和消息的语法约束,因此,可以从 BPN 模型中直接获得 WSDL.为了和 BPN 统一,我们重新定义 Web 服务的 WSDL 接口模型如下:

定义 2(WSDL). 令 $N(P,T,F,I,O)$表示一个 Web 服务的 BPN 模型,则该 Web 服务的 WSDL 文档可以形式化地根据 N 定义为一个五元组$(name,OP,MSG_I,MSG_O,F_{WSDL})$,其中:

- $name$ 表示 Web 服务的名称;
- $OP\subseteq T_O$,表示 Web 服务接口中的操作集合,且对于任意 $op\in OP$,有$({}^\bullet op\cap I)\cup(op^\bullet\cap O)\neq\varnothing$;
- $MSG_I\subseteq P_D\cap I$,表示 Web 服务接口中的输入消息集合;
- $MSG_O\subseteq P_D\cap O$,表示 Web 服务接口中的输出消息集合;
- $F_{WSDL}\subseteq(MSG\times OP)\cup(OP\times MSG)$,表示消息与操作之间的映射关系.

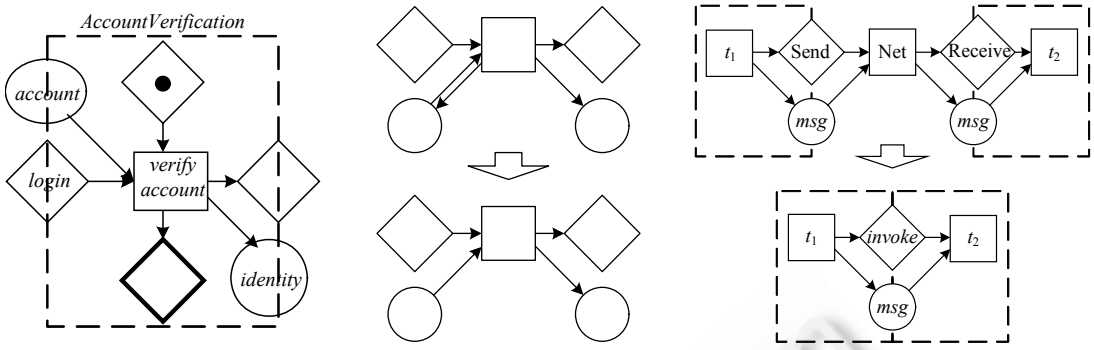
我们以用户登录验证服务作为示例,该服务的 BPN 表示如图 2(a)所示.服务的操作 *verifyAccount* 对应于图中的变迁,该操作的输入消息 *account* 和输出消息 *identity* 对应于相应的数据库所.

通过扩展普通 Petri 网的发射规则得到如下 BPN 中变迁的发射规则:

定义 3(业务协议网的发射规则,firing rule of BPN). 对于一个 $BPN(P,T,F,I,O)$的标记 m,一个变迁 $t\in T$ 是使能的当且仅当$\forall p\in P$ 且 $p\in{}^\bullet t$,有 $m(p)>0$,特别地,对于 $t\in T_C$ 还要求$\forall p\in P_D$ 且 $p\in{}^\bullet t$,有 $t.c(d_p)$=true,即条件变迁中的条件判断为真,记为 $m[t\rangle$.如果 t 是使能的,则 t 可以触发并导致标记变迁到 m',记为 $m[t\rangle m'$.

除此,为了方便使用 BPN,这里我们给出一些 BPN 模型具体的发射规则和化简方法:

- 首先,一般情况下,所有数据库所的输出弧都会有一个相应的输入弧,表示数据一旦赋值就始终有值.为了简便,在不影响语义的情况下可以只画输出弧,如图 2(b)所示;
- 其次,所有的服务调用可以表示为不同参与方 BPN 片断之间通过变迁进行连接,如图 2(c)上图所示.
- 然而为了简便,通常情况下,表示网络传输的变迁可以省略,如图 2(c)下图所示.本文后续部分均不加说明地使用化简后的 BPN 模型.



(a) An example of BPN (a) 一个 BPN 示例 (b) Simplification of data place (b) 数据库所化简 (c) Simplification of service invocation (c) 服务调用化简

Fig.2 An example of BPN and the simplification of BPN

图 2 BPN 的示例和化简方式

3.2 Web服务调用消息日志

本节定义 Web 服务的调用消息日志.

定义 4(WML). 一个 Web 服务调用消息日志(WS message log,简称 WML)是一组 Web 服务调用消息的序列 $l=m_0,m_1,\ldots,m_n$,其中,$m_i(0\leqslant i\leqslant n)$是一条消息,表示 Web 服务操作的一次调用.令 $WSDL(name,OP,MSG_I,MSG_O,F_{WSDL})$为该 Web 服务的 WSDL 文档,消息 m 可以表示一个四元组(id,op,msg,io),其中:

- id 是 Web 服务的执行标识,Web 服务的每次执行都产生一个唯一的 id.因此,具有相同 id 的所有消息属于 Web 服务的同一次执行;
- $op\in OP$ 表示一次 Web 调用中的调用操作,这里不失一般性,我们假定每个操作的名称都不相同;
- $msg\subseteq MSG_I\cup MSG_O$ 表示 Web 服务调用的消息集合,从中我们可以解析出调用操作的输入/输出参数;
- $io\in\{i,o\}$表示 Web 服务调用消息的方向,可以是 i 表示输入(接收)或者是 o 表示输出(发送).

Web 服务的调用消息日志可以从 Web 服务容器或 Web 服务所在的业务流程管理系统中获得,其中顺序记录了一段时间内 Web 服务调用产生的所有输入/输出消息.这些消息是我们获取 Web 服务接口的信息来源,下节详细介绍如何利用这些消息日志抽取 Web 服务的接口.

4 Web 服务接口业务协议挖掘

本节重点介绍我们提出的 Web 服务接口自动挖掘方法.首先,提出了一种结合数据流分析和控制流挖掘技术的 Web 服务接口自动挖掘的框架,如图 3 所示.

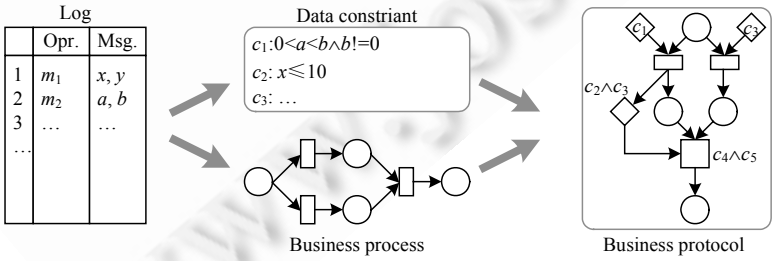


Fig.3 Process of Web service business protocol mining

图 3 Web 服务接口业务协议挖掘流程

该框架包含了 4 个主要步骤:

(1) 预处理执行记录;

(2) 获取数据约束;

(3) Web 服务接口的业务流程挖掘;

(4) 合成 Web 服务接口的业务协议.

下面分别就这 4 步进行详细的说明.

4.1 预处理执行记录

在这一步中,主要是对 Web 服务的调用消息日志进行预处理.首先,对日志中的所有调用消息按照 Web 服务进行分组,即所有同一服务的调用消息合并到同一集合中.然后,按照 Web 服务的执行 *id* 对同一服务的消息再进行分组,从而可以获得同一服务每次执行的消息序列.最后,在得到的分组消息中过滤掉出错或未完成的执行消息序列,因此得到的分组消息都是完整的消息序列.除此以外,我们进一步合并具有相同操作的调用序列,如第 1 节案例中的第 1 条和第 3 条记录,合并结果见表 2.后续算法只针对合并后的调用消息记录进行挖掘.

Table 2 Merge result of the second and fourth records in loan service message logs

表 2 前两条贷款服务的调用消息记录合并结果

	1	3	4
Opr.	*loanReq*	*creditAcc*	*loanAck*
Msg.	x=8000, u=001 x=10000, u=002	u=001, c=low u=002, c=high	a=false a=true

4.2 获取数据约束

在上一步的基础上,我们从消息记录中分析获取程序的数据约束,这些约束主要体现为流程分支变迁中的条件.在这一过程中,我们主要借鉴了 Daikon 系统[12]中的程序不变量分析算法,该方法通过分析程序中的变量和相应变量取值之间的关系,自动生成数据之间的约束关系.Daikon 处理数据的名值对$\langle p,d\rangle$,并自动生成所有数据之间的约束关系.生成的约束关系将通过概率阈值进行过滤,以避免偶然出现的约束关系.令 D 表示数据值域的集合,C:$D\rightarrow$*boolean* 表示数据之间的约束关系.举例来说,若 x 为 Web 服务接口中的一个参数,则 c:0<x<5 表示一条数据约束.Daikon 一旦发现了一条数据之间的约束 $c\in C$,则表示对于 c 中的相应数据,在日志中的所有取值 $d\in D$ 都有 $c(d)$=true.图 4 给出了一个 Daikon 处理过程的示例.

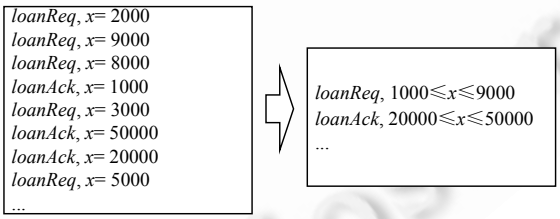


Fig.4 An example of conditions generated by Daikon

图 4 Daikon 工具计算出的流程分支条件示例

经过 Daikon 的处理,我们可以在 Web 服务调用消息日志的基础上得到带有数据约束的消息序列 W,对于任意 $w\in W$,有 $w=(m_0,c_0),(m_1,c_1),\ldots,(m_n,c_n)$.

4.3 Web服务接口的业务流程挖掘

在这一步中,我们从 Web 服务的调用消息序列中合成 Web 服务接口的控制流约束,具体计算方法是直接借用经典的业务流程挖掘算法:α算法.限于篇幅,对于α算法,这里我们不作详细介绍,只简述其主要思想,具体算法细节可参考文献[13].首先,α算法通过识别调用序列中的活动,建立业务流程 Petri 网模型中的变迁集合.然后,α

算法中定义任意两个活动 a,b 之间依据流程执行日志 W 存在如下 4 种关系:

- 后继:$a>_W b$,当且仅当存在一个序列,其中 b 紧跟在 a 后面出现;
- 因果:$a\rightarrow_W b$,当且仅当 $a>_W b$ 并且 $b\not>_W a$;
- 无关:$a\#_W b$,当且仅当 $a\not>_W b$ 并且 $b\not>_W a$;
- 并发:$a\|_W b$,当且仅当 $a>_W b$ 并且 $b>_W a$.

通过分析α算法,建立执行序列中任意两个活动之间的关系,并根据这些关系确定业务流程 Petri 网模型中的库所集合和相应的弧.

我们应用α算法挖掘 Web 服务的调用消息记录,可以获得 Web 服务接口协议中的业务流程部分,并将其输出映射为 BPN 表示,即只含有控制库所和 Web 服务调用变迁的 BPN 模型.以此方法对银行贷款应用案例中的 Web 服务调用日志进行挖掘,获得的 BPN 模型如图 5 所示.其中,受限于α算法,所获得的 BPN 模型存在部分语义不清晰的地方,如变迁 *loanReq* 既是服务调用变迁又是同步变迁,中间部分的选择分支缺乏触发条件,使用者无从判断何时选择哪条分支路径.

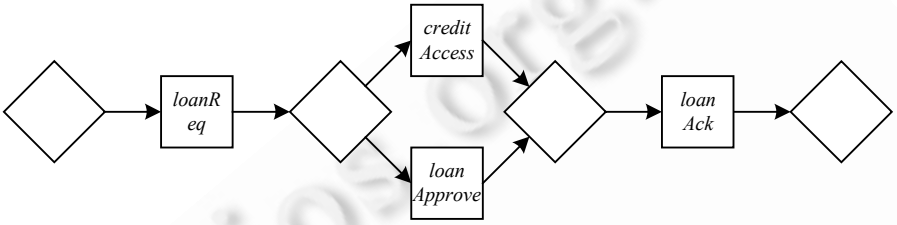


Fig.5 Process of loan service derived by α algorithm

图 5 银行贷款服务调用消息日志利用α算法的计算结果

4.4 合成Web服务接口的业务协议

在完成以上步骤的基础上,可以进一步合成 Web 服务接口的业务协议.其主要思想是,在第 3 步获得的 BPN 模型的基础上,利用 Web 服务自身提供的数据信息和获得的数据约束对其进行扩展,从而获得完整的 BPN 模型.

定义 5(数据-变迁关系). 令 $W=\{w_1,\ldots,w_m\}$表示一组 Web 服务的调用消息记录,则$\forall i=1,\ldots,m$,有 $w=(m_0,c_0),(m_1,c_1),\ldots,(m_n,c_n)$,对于数据 d、变迁 a,b 和条件 c:$D\rightarrow boolean$,可以依据消息记录定义如下关系:

- 条件因果: $a\xrightarrow{c}_W b$ 当且仅当 $a\rightarrow_W b$,且存在一个序列 w_i 中存在相继的条件消息$(a,c_a)(b,c_b)$使得条件 c 成立;
- 输入:$d>_W a$ 当且仅当存在一个接收消息 $m_i\in w_j,0\leqslant i,j\leqslant n$,其中,$d\in m_i.msg,t=m_i.op_i$ 且 $m_i.io=$‘i’;
- 输出:$a>_W d$ 当且仅当存在一个发送消息 $m_i\in w_j,0\leqslant i,j\leqslant n$,其中,$d\in m_i.msg,t=m_i.op_i$ 且 $m_i.io=$‘o’;
- 条件参数: $d\xrightarrow{c}_W a$ 当且仅当$\forall m_i\in w_j,0\leqslant i,j\leqslant n$,且 $d\in D(c)$,有条件 c 成立.

条件因果是对α算法中因果关系的扩展,即在条件 c 成立的情况下,变迁 a 和变迁 b 存在因果关系.条件 c 可以通过 W 计算获得,计算方法是$\forall w_i\in W$,且 w_i 中有相继的条件消息$(a,c_a)(b,c_b)$,有 $c=V_i(c_a\wedge c_b)$.在以上关系定义的基础上,我们给出合成 Web 服务接口的业务协议的算法,具体如算法 1 所示.其中,*addCPlaceBefore* 和 *addCPlaceAfter* 表示在变迁前、后插入控制库所,*addDPlaceBefore* 和 *addDPlaceAfter* 表示在变迁前、后插入数据库所,*insertCondition* 表示在控制库所与其后继变迁之间插入一个控制库所和一个条件变迁,参数 c 表示该条件变迁的条件,*insertSynchronizerBefore* 和 *insertSynchronizerAfter* 表示在变迁与其所有前趋或所有后继之间插入一个控制库所和一个同步变迁.经过扩展α算法的处理,我们可以从 Web 服务的调用消息日志中自动挖掘出 Web 服务接口的 BPN 模型.图 6 展示了经过我们的方法所获得的银行贷款服务接口的业务协议模型.

算法 1. Web 服务接口业务协议合成算法.

输入:带数据约束的消息序列 $W,\alpha(W)=(P_W,T_W,F_W)$,

Web 服务对应的 WSDL 文档$(name,OP,MSG,F_{WSDL})$;

输出:Web 服务接口的 BPN 模型 N.

//根据α算法的结果$\alpha(W)$初始化 $N(W)$的元素

1. $P_{CW}=P_W,P_{DW}=I=O=\varnothing$;

//添加接口上的控制库所以及与变迁之间的弧

2. for each $m_i\in w_i\wedge m_i.io=$'i'$\wedge w_i\in W\wedge t(m_i.op)\in T_W$ do

3. $I=I\cup\{addCPlaceBefore(t(m_i.op))\}$;

4. for each $m_i\in w_i\wedge m_i.io=$'o'$\wedge w_i\in W\wedge t(m_i.op)\in T_W$ do

5. $O=O\cup\{addCPlaceAfter(t(m_i.op))\}$;

//添加接口上的数据库所以及与变迁之间的弧

6. for each $d\in m_i.msg\wedge m_i.io=$'$i$'$\wedge m_i\in w_i\wedge w_i\in W\wedge t(m_i.op)\in T_W$ do

7. $I=I\cup\{addDPlaceBefore(t(op),d)\}$;

8. for each $d\in m_i.msg\wedge m_i.io=$'$o$'$\wedge m_i\in w_i\wedge w_i\in W\wedge t(m_i.op)\in T_W$ do

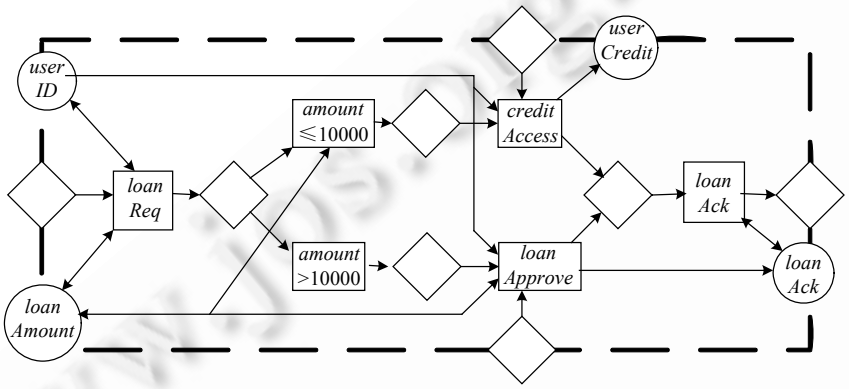


Fig.6 BPN model of loan service derived by method of this paper

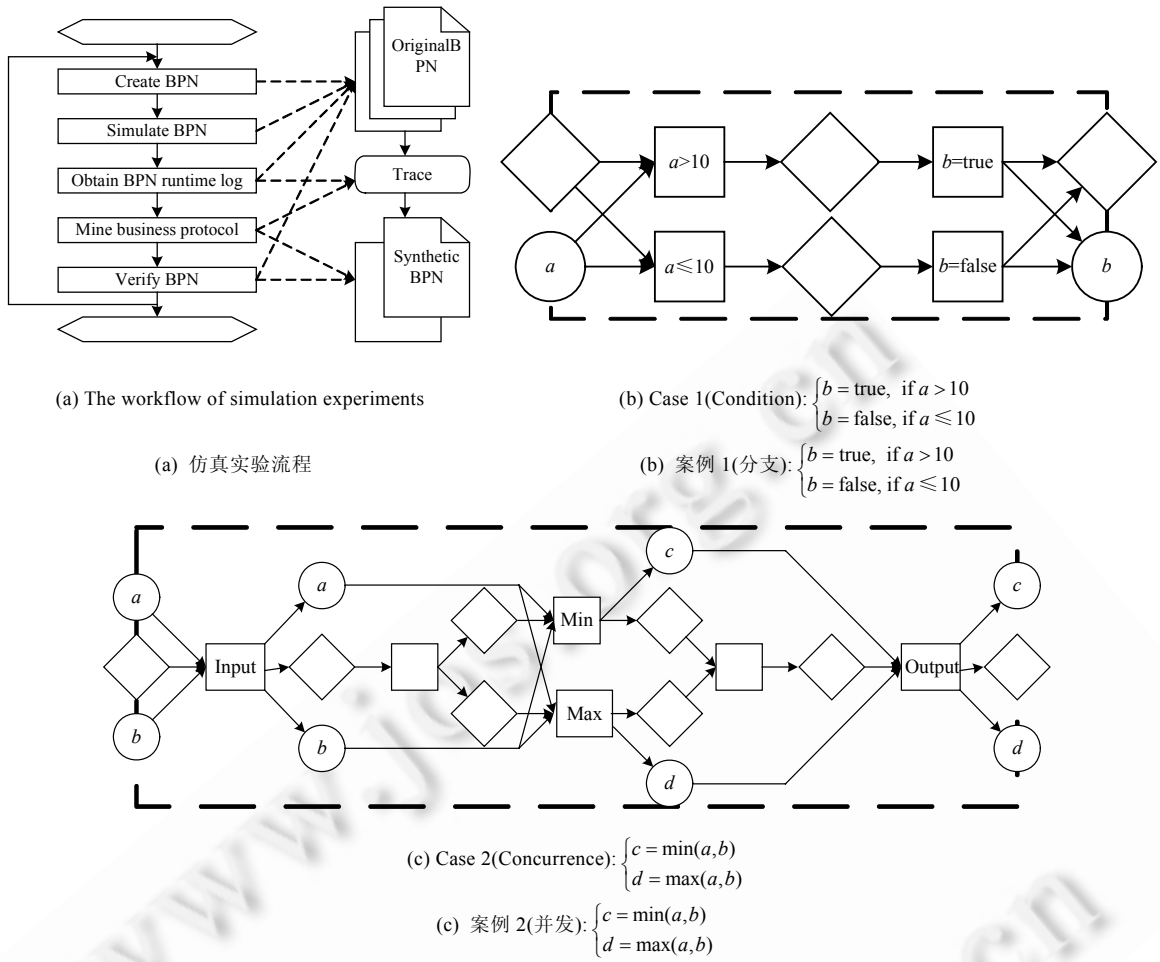
图 6 采用本文方法最终合成的银行贷款 Web 服务接口的业务协议

算法 1 以前两步的挖掘结果为输入,合成 Web 服务接口的业务协议.算法最后的业务协议合成部分类似于α算法,由数据-变迁关系主导,而这一关系的计算是与服务调用消息记录的规模呈线性关系,因此,整个合成 Web 服务接口的业务协议过程的复杂度也是多项式时间的.考虑到 Daikon 工具本身算法[12]和α算法的复杂度都是多项式时间的[13],整个挖掘方法的复杂度也是多项式时间的,这对于大多数服务协议挖掘的应用场景来说是可以接受的.

5 实验评估

5.1 案例分析

本节通过仿真实验对本文方法的有效性和效率进行评估,大体的仿真实验方案如图 7(a)所示.首先,选取两组包含了典型流程模式的 Web 服务,如图 7(b)、图 7(c)所示,通过随机生成测试数据,对这些案例的 BPN 模型进行仿真运行,并记录 BPN 中位于接口处的服务变迁的调用消息.然后,利用本文方法对这些记录进行挖掘,从而获得了以上案例中服务接口的业务协议.最后,将挖掘得到的业务协议 BPN 模型与原 BPN 模型进行比对,以确定前者的正确性.比对过程主要是通过对两个 BPN 模型进行模拟运行,然后比对两个模型产生的行为序列.如果两者产生的序列完全相等,则认为挖掘结果正确.显然,依照这种方法对两个 BPN 模型进行比对,只能确定两模型之间是否存在执行序列相等(trace equal),而两模型在形式上仍有可能存在差异.



(a) The workflow of simulation experiments

(b) Case 1(Condition): $\begin{cases} b = \text{true}, & \text{if } a > 10 \\ b = \text{false}, & \text{if } a \leqslant 10 \end{cases}$

(a) 仿真实验流程

(b) 案例 1(分支): $\begin{cases} b = \text{true}, & \text{if } a > 10 \\ b = \text{false}, & \text{if } a \leqslant 10 \end{cases}$

(c) Case 2(Concurrence): $\begin{cases} c = \min(a,b) \\ d = \max(a,b) \end{cases}$

(c) 案例 2(并发): $\begin{cases} c = \min(a,b) \\ d = \max(a,b) \end{cases}$

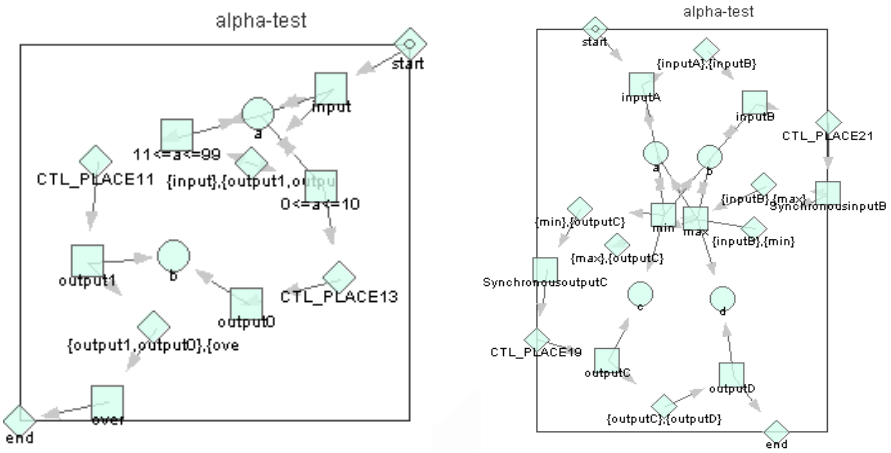
Fig.7 Process of simulation and the cases used in experiments

图 7 仿真实验的设计方案和案例

仿真程序使用 JDK1.5 开发,并在一台 CPU 为 1.86G、内存 1G 的 PC 上完成了实验.我们利用随机生成的测试输入对这两个案例进行仿真运行,每个案例各重复 100 次运行.挖掘程序在获得消息记录的基础上可以自动地对这两个典型案例中的 Web 服务接口业务协议进行挖掘,并准确地获得图 7 所示的 BPN 模型.挖掘结果的界面截图如图 8 所示,其中,BPN 模型是实验中采用的 Web 服务接口的业务协议.

我们进一步对本文方法的时间开销进行了评估,其时间开销与调用消息日志的记录数直接相关.我们将案例 1 和案例 2 作为待测服务,分别模拟运行产生 1 000~10 000 条日志,以 1 000 为单位进行 10 组挖掘实验,获得的时间开销分别如图 9 所示.从图中可以看出,完成 10 000 条日志的挖掘最大耗时为 8 535ms,这种量级的时间开销对于大多数服务协议挖掘场景来说是可以接受的.除此以外,我们也注意到本文方法对于不同案例时间开销有一定差别,但趋势大致相同.然而,同一案例的不同步骤产生的时间开销有较大差别,图中自下而上依次为预处理、获取数据约束、获取控制约束和协议合成的时间消耗曲线.较为耗时的步骤是预处理和获取数据流约束这两步.经过分析发现,这主要是因为预处理部分涉及大量消息解析和文件操作,而获取数据流约束则需要调用 Daikon 工具,并以文件方式与之交换数据,从而造成了较大开销.预处理部分的时间随调用日志记录数目的增多呈线性增长,但后续步骤基本变化不大.消息的解析处理过程和获取数据流约束的过程容易通过程序设计技巧来优化,可以在后续工作中加以改进.性能分析结果表明,本文方法的时间开销主要取决于消息的解析处理过

程,与日志记录数目基本呈线性关系,可以应用于实际的服务协议挖掘中.

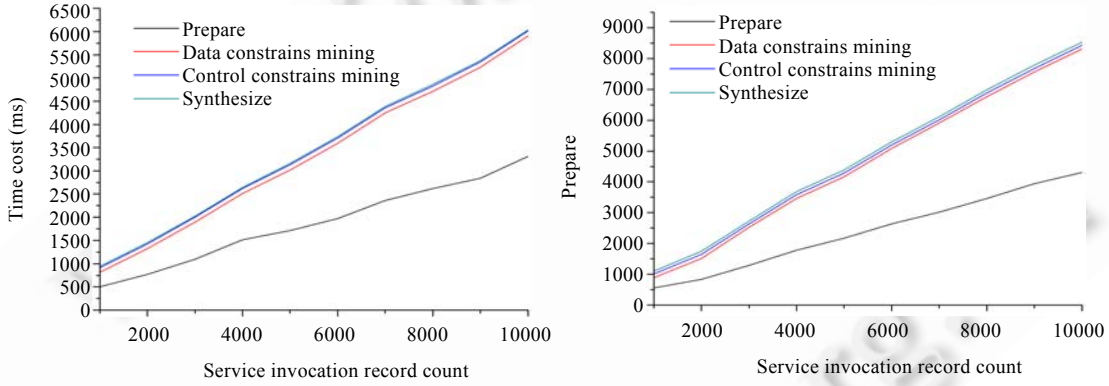


(a) BPN mined from case 1
(a) 案例 1 挖掘结果的 BPN 模型截图

(b) BPN mined from case 2
(b) 案例 2 的挖掘结果的 BPN 模型截图

Fig.8 Screenshot of BPN mining tool

图 8 BPN 挖掘工具界面截图



(a) Case 1 performance
(a) 案例 1 性能分析

(b) Case 2 performance
(b) 案例 2 性能分析

Fig.9 Time cost of mining with changes in the number of logs

图 9 挖掘过程的时间开销随日志记录数目变化情况

实验结果表明,本文方法可以有效地从 Web 服务调用消息日志中挖掘出 Web 服务的协议模型,且其时间开销是可接受的.

5.2 讨 论

在实验过程中我们发现,本文方法在应用于实际的 Web 服务接口业务协议发现过程中存在着一定的使用限制,这主要是来自 Daikon 工具和α算法自身的限制.

首先,由于 Daikon 工具是基于对日志的统计分析获得数据之间的约束关系的,因此可能存在一定的误差,这种误差可以通过分析更多的日志来减小.然而,误差是一定存在的,这致使获得的 BPN 模型对于 Web 接口来的业务协议来说只是充分条件,而非必要条件.即符合 BPN 模型的行为一定符合 Web 服务接口,而不符合 BPN 模

型的行为却并不一定不符合 Web 服务接口.因此,通过本文方法获得的 BPN 模型对于 Web 服务接口来说过于严格,在使用时必须加以注意.

其次,由于α算法自身的限制,要求被分析的 Web 服务接口的业务流程必须是合理的[13],特别是存在一类特殊的业务流程是α算法无法识别的.由于本文方法直接基于α算法的计算结果,因此也继承了所有这些限制.一方面,当 Web 服务接口中包含α算法无法识别的流程时,其接口业务协议也无法通过本文方法获得,这在使用过程中也是需要注意的.不过,对于大多数常见流程模式及其嵌套形式,本文方法还是可以对其进行准确挖掘的.另一方面,α算法所识别的流程与原流程模型之间是执行序列相等(trace equal),并非模型上完全相等,因此,模型的形式与原流程可能存在偏差.不过,这一点并不会影响本文方法在服务协议挖掘场景中的应用.这是因为,大多数服务的内部流程实现较其外部可观察的流程要复杂得多.挖掘出的协议一般用于指导服务使用者正确使用服务,因此只需保证内部流程与外部协议行为一致即可,而不必过分关心两者之间在形式上的差异.综上所述,本文方法对于大多数服务是适用的.

6 结论和下一步工作

本文的主要贡献是,针对 Web 服务协议挖掘问题提出了一种基于日志的自动挖掘方法,并侧重于考虑 Web 服务协议中的数据流和控制流共同构成的一致性约束.为了描述同时带有数据约束和控制约束的 Web 服务协议,基于 Petri 网,我们提出了一种增加了数据流描述的 Web 服务接口模型——BPN 模型.在此基础上,提出了基于日志的 Web 服务协议自动化挖掘框架.借鉴了已有的研究成果,框架中利用 Daikon 工具和α算法分别完成数据流约束和控制流约束的自动挖掘,然后再利用本文提出的合成算法获取一致性约束.最后,通过仿真实验验证了本文方法的有效性,并详细讨论了本文方法的适用范围.

目前,由于缺乏丰富的实际 Web 服务调用日志,本文提出的方法还未能在实际面向服务的企业信息系统中加以应用.在下一步工作中,我们将重点考虑如何消除之前所述的本文方法的限制,并将本文方法相应的原型系统应用于实际的企业信息系统,以便对其进行验证和改进.另一方面,针对目前 Web 服务调用日志缺乏的现状,未来考虑在本文方法的基础上结合主动测试技术,尝试通过反复试探的方法获取 Web 服务的接口信息.

致谢 在此,我们向全体参与“可信的国家软件资源共享与协同生产环境”课题研发工作的科研人员,尤其是北京航空航天大学课题组参与面向服务软件生产线子课题研发的老师和同学表示感谢.尤其感谢北京航空航天大学金若凡同学帮助我们完成了本文的仿真实验.

References:

[1] Zhang LJ, Zhang J, Cai H. Services Computing. Beijing: Springer-Verlag, Tsinghua University Press, 2007.

[2] Berardi D, Calvanese D, de Giacomo G, Lenzerini M, Mecella M. Automatic service composition based on behavioral descriptions. Int'l Journal of Computational Intelligence Systems, 2005,14(4):333−376.

[3] Bonchi F, Brogi A, Corfini S, Gadducci F. A behavioral congruence for Web services. In: Arbab F, Sarjani M, eds. Proc. of the Fundamentals of Software Engineering. LNCS 4767, Berlin: Springer-Verlag, 2007. 240−256. [doi: 10.1007/978-3-540-75698-9_16]

[4] Desai N, Singh MP. Protocol-Based business process modeling and enactment. In: Proc. of the IEEE Int'l Conf. on Web Services (ICWS 2004). IEEE, 2004. 35−42. [doi: 10.1109/ICWS.2004.1314721]

[5] Fan WF, Geerts F, Gelade W, Neven F, Poggi A. Complexity and composition of synthesized Web services. In: Proc. of the PODS 2008. Vancouver: ACM, 2008. 231−240. [doi: 10.1145/1376916.1376949]

[6] Al-Masri E, Mahmoud QH. Investigating Web services on the World Wide Web. In: Proc. of the WWW 2008. Beijing, 2008. [doi: 10.1145/1367497.1367605]

[7] Beyer D, Chakrabarti A, Henzinger TA. Web service interfaces. In: Proc. of the WWW 2005. Chiba, 2005.

[8] Ammons G, Bodík R, Larus JR. Mining specification. In: Proc. of the POPL 2002. 2002. [doi: 10.1145/503272.503275]

[9] Serrour B, Gasparotto DP, Kheddouci H, Benatallah B. Message correlation and business protocol discovery in service interaction logs. In: Proc. of the CAISE 2008. Berlin: Springer-Verlag, 2008. 405−419. [doi: 10.1007/978-3-540-69534-9_31]

[10] Shoham S, Yahav E, Fink S, Pistoia M. Static specification mining using automata-based abstractions. In: Proc. of the ISSTA 2007. London: ACM, 2007. 174−184. [doi: 10.1145/1273463.1273487]

[11] Acharya M, Xie T, Pei J, Xu J. Mining API patterns as partial orders from source code: From usage scenarios to specifications. In: Proc. of the ESEC/FSE 2007. Cavtat Near Dubrovnik: ACM, 2007. 25−34. [doi: 10.1145/1287624.1287630]

[12] Ernst MD, Cockrell J, Griswold WG, Notkin D. Dynamically discovering likely program invariants to support program evolution. IEEE Trans. on Software Engineering, 2001,27(2):99−123. [doi: 10.1109/32.908957]

[13] van der Aalst WMP, Weijters T, Maruster L. Workflow mining: Discovering process models from event logs. IEEE Trans. on Knowledge and Data Engineering, 2004,16(9):1128−1142.

[14] Object Management Group. Business Process Modeling Notation. Vol.1.1, OMG Available Specification, 2008.

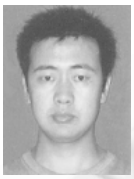
[15] van der Aalst WMP, Massuthe P, Stahl C, Wolf K. Multiparty contracts: Agreeing and implementing inter-organizational processes. The Computer Journal, 2010,53(1):90−106.

[16] Motahari H, Benatallah B, Saint-Paul R. Protocol discovery from imperfect service interaction data. In: Proc. of the VLDB 2006. Seoul: Springer-Verlag, 2006. http://ceur-ws.org/Vol-170

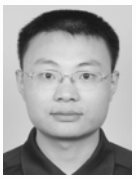
[17] Lorenzoli D, Mariani L, Pezzè M. Automatic generation of software behavioral models. In: Proc. of the ICSE 2008. Leipzig: ACM, 2008. 501−510. [doi: 10.1145/1368088.1368157]

[18] Prom. 2009. http://prom.win.tue.nl/research/wiki/prom/start

[19] de Medeiros AKA, van Dongen BF, van der Aalst WMP, Weijters AJMM. Process mining: Extending the α-algorithm to mine short loops. In: BETA Working Paper Series, WP 113. Eindhoven: Eindhoven University of Technology, 2004.

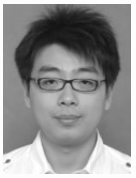


李翔(1977－),男,山东济南人,博士生,CCF 学生会员,主要研究领域为服务计算,软件设计与生产.



孙海龙(1979－),男,博士,讲师,CCF 会员,主要研究领域为服务计算,网络计算.

怀进鹏(1962－),男,博士,教授,博士生导师,主要研究领域为计算机软件与理论,网络计算技术,信息安全.



曲先洋(1984－),男,硕士生,主要研究领域为服务计算,软件设计与生产.

刘旭东(1965－),男,博士,教授,博士生导师,CCF 会员,主要研究领域为可信网络计算技术,中间件技术.