基于UML2.0序列图的Web服务运行时验证方法

张亚红[1] 张琳琳[1] 赵 楷[1] 陈佳丽[1] 冯在文[2]
(新疆大学信息科学与工程学院 乌鲁木齐 830046)[1] (武汉大学软件工程国家重点实验室 武汉 430072)[2]

摘 要 为了确保包括非功能属性在内的服务规约与服务实际运行行为之间的一致性,提出一种Web服务运行时行为验证方法。首先对UML 2.0序列图进行扩展,将QoS属性和功能属性的描述统一起来,以精确表达Web服务的需求规约。然后,提出利用确定有限自动机构造出扩展序列图(Extended Sequence Diagrams,ESD)的语义模型的方法。最后,给出验证准则,根据Web服务的交互消息和规约建模的结果来验证Web服务运行时行为与需求规约之间的一致性。基于上述研究,设计开发了Web服务运行时验证工具(Runtime Verification Tool for Web Services,RVT4WS),以支持对Web服务运行时行为的验证。

关键词 UML 2.0序列图,确定有限自动机,Web服务,运行时验证
中图法分类号 TP311 **文献标识码** A

Runtime Verification Method for Web Service Based on UML2.0 Sequence Diagrams

ZHANG Ya-hong[1] ZHANG Lin-lin[1] ZHAO Kai[1] CHEN Jia-li[1] FENG Zai-wen[2]
(College of Information Science and Engineering,Xinjiang University,Urumqi 830046,China)[1]
(State Key Laboratory of Software Engineering,Wuhan University,Wuhan 430072,China)[2]

Abstract To verify the consistency between run-time behavior of Web service and its specification,a runtime verification method for Web Service was proposed. In this paper,UML2.0 Sequence Diagrams were extended to describe the specification of Web service from both functional and QoS aspects,and then Extended Sequence Diagrams (ESD) were transformed to Deterministic Finite Automata to indicate semantics,and verification criteria was given to verify the consistency between run-time behavior and the specification. In addition,a runtime verification tool for Web service (RVT4WS) was developed to support the runtime verification method we proposed.

Keywords UML2.0 sequence diagrams,Deterministic finite automata,Web service,Runtime verification

1 引言

Web服务作为一种新型分布式计算模型[1],近年来得到了学术界和工业界的极大关注。Web服务的运行时行为与用户需求规约的一致性验证问题是目前服务计算领域的研究热点之一。运行时验证是一种新兴的轻量级验证技术,它把形式化验证技术和系统的实际运行结果结合起来,监控系统的运行行为,以保证系统的运行符合用户的需求规约[2]。现有的Web服务验证技术中,缺乏将功能需求和QoS需求统一描述并进行验证的方法[16]。通常对功能需求和QoS需求的建模是分别进行的,并且两者建立的阶段可能不同,会导致验证结果与实际不相符。QoS作为系统非功能属性,是对系统功能需求、系统整体性质和系统开发过程的某种约束,在一定程度上影响着系统的功能性[3]。因此对Web服务行为的验证不仅应考虑Web服务的功能需求,还要考虑该服务所要满足的QoS需求。对Web服务的功能需求和QoS需求进行统一描述将有助于提高验证的有效性。文献[4]将QoS分为运行时相关的QoS、事务支持相关的QoS、配置管理相关的QoS和与安全性相关的QoS。本文由于关注Web服务的运行时行为,因此只对运行时相关的QoS需求规约进行描述。运行时相关的QoS包括ResponsTime,Latency,Throughput,Reliability,Availability等。

UML序列图作为一种规约描述语言[5],提供了一种直观的、可视化的方法来描述系统的需求规约,适合描述与消息相关的Web服务需求规约,但是UML序列图对QoS需求规约描述支持不足。QoS作为功能需求、系统整体性质和系统开发过程中的某种约束,与对象和消息是紧密相连的,也就是说一些QoS属性是可以通过具体的功能属性来展示和实现的。因此,本文针对目前Web服务功能和QoS统一描述和验证方面的不足,提出一种基于UML2.0时序图的Web服务运行时验证方法。通过对UML2.0时序图的扩展,从功能和QoS属性两方面统一描述与消息有关的Web服务的需求规约,基于此展开运行时验证工作。使用扩展UML2.0序列图描述需求规约,由于缺乏精确的语义,给验证工作带来了很大的挑战。确定有限自动机作为一种抽象的计算模型,具有明确的语义,能够准确表达序列图中各对象在发送、接受消息

到稿日期:2012-09-03 返修日期:2012-12-24 本文受国家自然科学基金资助项目(61100017,61262089),自治区高校科研计划项目(XJEDU2011S24),福建省自然科学基金项目(2012J01250,2011J05146),新疆大学博士毕业生科研启动基金项目(BS090142)资助。
张亚红(1987-),女,硕士生,CCF会员,主要研究方向为Web服务、形式化验证,E-mail:shinian.zyh@163.com;张琳琳(1974-),女,博士,副教授,主要研究方向为面向方面技术、软件体系结构;赵 楷(1976-),男,博士,讲师,主要研究方向为语义Web技术、SOA。

后的状态变迁和对象之间在消息交互过程中的状态迁移。UML2.0序列图中包含选择、循环等组合片段，包含在这些组合片段内的消息发生交互时需要判断消息交互的条件，只有满足条件，消息才能传递给对象，即对动作执行存在约束，传统自动机无法描述UML2.0序列图的这个新特征。为了能够精确描述扩展序列图的语义，本文采用基于消息扩展的确定有限自动机，即消息确定有限自动机描述扩展序列图的语义，以实现Web服务需求规约的运行时验证。

2 相关工作

与MSCs[6]和LSCs[7]等基于场景的形式化表示类似，UML序列图作为一种规约描述语言，使用越来越广泛。Harel等人在文献[8]中定义了MSD(Modal Sequence Diagrams)，MSD是UML2.0序列图的扩展，它将图、消息和约束定义为hot或者是cold。Autili等人在文献[9]中提出了PSC语言，其是对UML2.0序列图进行的扩展。PSC通过对消息设定fail和required属性来描述安全性和活性，使用Büchi自动机描述PSC语义。Ameedeen等人在文献[10]中提出将UML2.0序列图子集转化为Petri网，使用序列图子集描述系统行为。X. Li等人在文献[11]中使用UML 2.0 IDOs和序列图构建基于场景的需求规约，提出了Java程序异常一致性和托管一致性的运行时验证方法。

基于自动机的Web服务建模和验证也有不少研究成果。段振华等人在文献[12]中提出一种基于扩展有限自动机验证(EDFA)组合Web服务的方法，简化并自动化组合Web服务验证。文献[13]使用UML2.0序列图子集作为Web服务性质规约语言，并通过将这些图转化为自动机装置来验证Web服务的有限执行轨迹与性质描述之间的一致性，但是缺少对QoS需求规约的描述和验证。黄志球等人在文献[14]中针对BPEL4WS，定义了接口自动机和BPEL4WS之间的概念映射，提出了一种基于接口自动机的Web服务组合形式化模型。文献[15]将带有时间约束的Web服务智能体建模为时间自动机，通过并发组合构成时间自动机网络，最后使用时间自动机验证工具UPPAAL对组合Web服务的运行过程进行模拟和验证。

此外，国内外学者采用多种形式化方法对Web服务建模和验证问题展开研究，如进程代数、Petri网、Pi-演算、状态机、线性时态逻辑等。文献[16]针对目前QoS属性在Web服务组合的系统建模中研究不足的现状，提出了一种时间概率代价进程代数TPPPA，并给出了TPPPA的语法和语义，TPPPA具有功能、时间、概率和代价的统一建模和分析能力。吴泉源等人在文献[17]中提出一种基于有色Petri网模型的运行时确保机制，使用服务交互行为CPN模型精确刻画服务交互行为及其重要属性，并结合服务实例，进一步对模型可达性和关键监控属性展开深入分析，提高服务运行的可靠性。文献[18]使用Event Calculus形式化方法描述Web服务行为并验证组合Web服务行为的一致性，为保证Web服务执行可靠性提供了逻辑基础。文献[19]对Pi-演算扩展了事务语义，将进程间的动作交互与跨组织膜活动相关联，来刻画多业务事务协调行为，并基于等价自动机转换思路集成现有模型检验技术，验证多业务事务是否满足人们需要的各种性质。Dimitris Dranidis等人在文献[20]中提出一个Web服务运行时验证方法，其使用Stream X-machines(SXM)来形式构建Web服务的行为说明并判定服务运行时行为是否违反了已定义的需求说明。文中还提出了一个Web服务运行时监视和验证框架，该框架通过一个中间件来解释服务端和服务提供者之间的交互信息。该框架捕捉SOAP请求消息，将消息传递给SXM模型实例，服务响应消息保存在日志文件中，通过行为分析器模块对行为进行分析。文献[21]中使用线性时态逻辑LTL描述软件性质，并提出三值语义来判断运行的系统是否满足LTL属性，实现软件的运行时验证。文献[22]采用基于时序逻辑的XYZ/ADL描述Web服务的交互行为和时间属性，提出一种复合模型检测工具UPPAAL规约的时间异步通信模型，利用UPPAAL验证服务组合系统异步通信行为的正确性。与这些工作不同的是，本文提出扩展序列图并用于Web服务的需求规约建模，以支持Web服务的功能需求和QoS需求的运行时验证。

3 UML序列图的扩展

序列图以图的形式描述了系统各对象之间的行为，强调对象之间的消息交互，它作为一种规约描述语言，适用于描述与消息相关的Web服务规约，应用也越来越广泛。UML2.0序列图添加了12种组合片段[5]，这些组合片段大大增加了其对系统需求规约及相关属性的描述能力。其中loop,opt,alt,break,par,neg会使执行路径产生不同分支，在描述需求规约中起着十分重要的作用，将作为本文的研究重点。基于QoS的特点，本节对UML2.0序列图进行扩展，下面给出扩展序列图的语法的形式化定义。

定义1(扩展序列图，Extended Sequence Diagrams，ESD) ESD用一个五元组 $ESD=(O,Msg,Frag,QoS,Num)$ 表示，其中：

- O 是有限对象的集合，每个 $o\in O$；
- Msg 是有限消息的集合，每个消息 $msg\in Msg$。消息 msg 标注为一个三元组(mID, $mName$, $fragID$)，mID 是消息 msg 的唯一标识符，$mName$ 是消息 msg 的名称，$fragID$ 是该消息所属的组合片段的标识符，若消息 msg 不包含在任何组合片段内，则标记为空。本文假设每个消息至多包含在一个组合片段内。
- $Frag$ 是组合片段的集合，每个组合片段 $frag\in Frag$。组合片段 $frag$ 标注为一个三元组($fragID$, $fragName$, $fragGuard$)，$fragID$ 是组合片段的唯一标识符，$fragName$ 是组合片段的名称，$fragGuard$ 是组合片段的执行条件。
- QoS 是消息集合 Msg 中的 msg 对应的QoS属性规约集合，每个属性规约 $qos\in QoS$，标注为一个四元组($qosID$, $qosName$, $qosValue$, mID)，$qosID$ 是该 qos 的唯一标识符，$qosName$ 是该 qos 属性的名称，$qosValue$ 是标注所描述的 qos 属性规约的值，mID 是该 qos 属性规约所对应的消息 ID。
- Num 是消息的个数 $|Msg|$。

在扩展的UML2.0序列图中，QoS属性是和消息配合在一起的，相当于为Web服务交互消息中所涉及的QoS属性建立了描述。因此，扩展后的UML2.0序列图描述的Web服务需求规约既包含了功能描述也包含了QoS属性描述，并且能在该描述的基础上对Web服务进行一系列验证工作。

4 消息确定有限自动机 MSDFA

ESD作为规约描述语言，能够从功能和QoS属性两方面统一描述Web服务需求规约，但是其缺乏语义。确定有限自动机(Deterministic Finite Automata)是能够实现状态转移的自动机，对于一个给定的属于该自动机的状态和一个属于该自动机字母表 Σ 的字符，其都能根据事先定义的状态转移函数转移到下一个状态。Web服务之间是通过SOAP消息驱动的，对于Web服务来说，建立SOAP消息的目的是为了在运行时刻传输数据。结合这一特点，为了准确表达ESD中各对象在发送或接受消息后的状态迁移，本文采用消息确定有限自动机(Message Deterministic Finite Automata，MSDFA)来表示ESD需求规约的语义。基于自动机理论，下面给出消息确定有限自动机的形式化定义。

定义2(MSDFA) 消息确定有限自动机AM是一个八元组 $AM=(S,M,F,Q,\Sigma,\tau,s_0,T)$，其中：

S 表示状态的非空有限集合，对于 $\forall s\in S$；

M 是有穷消息的集合；

F 是组合片段执行条件的集合，若无组合片段或是无参数组合片段，执行条件为空；

Q 是消息的QoS属性需求规约集合；

Σ 表示输入字母表 $\Sigma=\{(f,m,qos)\mid f\in F,m\in M,qos\in Q\}$；

τ 表示状态转移函数；

s_0 表示AM的初始状态，$s_0\in S$；

T 表示AM的终止状态集合，$T\subseteq S$。

MSDFA是确定有限自动机的扩展，MSDFA的一个状态的迁移描述了一次消息交互，迁移的标注包含3项内容：第一项是消息交互条件，即组合片段执行条件，若无组合片段，执行条件为空；第二项是传递的消息的名称；第三项是该消息上相应的QoS属性值。状态迁移表示在消息传递过程中，对象接受或者发送消息之后，从一个状态转移到另一个状态。显然MSDFA能够描述出Web服务中各对象之间的消息交互和QoS属性规约。

5 基于ESD的Web服务运行时验证

准确地描述Web服务的需求规约是Web服务运行时验证的重要基础。如何使用ESD描述Web服务的需求规约，以及如何将其转化为带精确语义的消息确定有限自动机从而实现自动化验证，是需要克服的两个难点。基于ESD的Web服务运行时验证方法的基本思想是首先使用ESD准确地描述出Web服务的需求规约，其次将其转化为带有精确语义的消息确定有限自动机，最后基于消息确定有限自动机来实现Web服务的运行时验证。根据第3节提出的ESD语法的五元组定义，每个需求规约的定义都会产生一个相应的五元组，而消息确定有限自动机定义是一个八元组，因此如何产生正确的五元组定义，并如何有效地将一组需求规约的五元组定义转化为八元组的消息确定有限自动机表示是下文的主要内容。本节通过一个简单的实例来详细介绍基于ESD的Web服务运行时验证方法的过程。

5.1 ESD描述需求规约

假设Web服务Loan Web Services(LWS)实现用户网上贷款。用户通过Web网页填写贷款申请信息(姓名、纳税人Id、贷款金额等)，Web服务最终返回贷款申请状态。LWS首先是判断用户的信用值是否符合系统要求，若用户信用值不符合要求，则拒绝该用户贷款请求；如果用户的信誉值满足要求，判断贷款金额，若贷款金额少于￥5000(包括5000)，则系统自动批准，否则进入手动批准流程。如前所述，QoS作为Web服务非功能属性，通常是通过具体的功能属性来展示和实现自身的。具体到ESD来说，QoS是与消息紧密相关的。例如，在LWS中，“消息CheckCreditRequest的可用性是0.97”、“消息CreditScore的可靠性是0.98”，都说明QoS属性(可用性和可靠性)是与ESD中的消息紧密相连的。因此本文定义，在ESD中QoS的表示法为：若一个QoS属性需求对应于ESD中的某个消息，则在该消息的定义结束后，标注出该QoS属性，QoS属性与消息的定义之间用分号“;”相隔，多个QoS属性之间用逗号“,”相连。

图1展示了使用ESD描述LWS要满足的一个需求规约 p 实例：在进行所有操作之前要检验贷款者的信誉值，且各消息需要满足指定的QoS属性需求规约，如Availability，Latency、Reliability等。

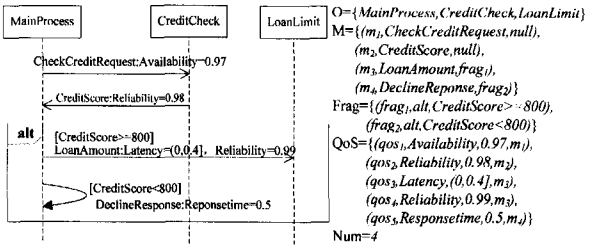

图1 ESD描述Web服务需求规约 p

5.2 MSDFA构造算法

本节讨论MSDFA构造算法。如前所述，ESD描述需求规约的语义是由消息确定自动机MSDFA表示的，ESD中各对象之间消息的一次交互，则对应一次MSDFA的状态迁移，在消息交互过程中会形成不同的状态，这些状态在MSDFA中对应于不同的状态节点。在ESD和MSDFA定义的基础上，下面给出MSDFA构造算法。

算法1(MSDFA构造算法) 对于每一个使用ESD描述的需求规约 p，都对应于一个消息确定有限自动机 AM_p，本文假设每条消息至多在一个执行片段中。

输入：需求规约p的ESD定义(O,Msg,Frag,QoS,Num)

输出：需求规约p的消息确定有限自动机 $MSDFA_p$

定义：messageList=“”;//消息序列

fragList=“”;//组合片段序列

qoSList=“”;//qos属性序列

n==Num，则 $S=\{s_0,s_1,\cdots,s_n\}$，其中 $T=\{s_n\}$

CheckFragList (M m){ return f in fragList where (f. fragID=m. fragID) }

//查找并返回fragList中fragID等于特定值的组合片段f

CheckQoSList(M m) { return q in qosList where (q. mID=m. mID) }

//查找并返回qosList中mID等于特定值的qos值的m

step1 创建初始状态 s_0；

step2 创建状态 $s_1,s_2\cdots s_n$；

step3 确定状态之间的迁移；

(1)如果 m_i 和 m_{i+1} 都不包含在任意组合片段中，

q=CheckQoSList (m_{i+1}) //消息 m_{i+1} 的QoS属性规约

$\tau(s_i, s_{i+1})$=[q. qosName=qosVal]/m_{i+1}//s_i 和 s_{i+1}状态迁移函数

(2)如果 m_i 和 m_{i+1}包含在相同的组合片段的相同的操作域中

q=CheckQoSList (m_{i+1})

$\tau(s_i, s_{i+1})$=[q. qosName=qosValue]/m_{i+1}

(3)如果 m_i 不包含在任意组合片段中，而 m_{i+1}包含在组合片段中

f=CheckFragList (m_{i+1})

q=CheckQoSList (m_{i+1})

$\tau(s_i, s_{i+1})$=[f. fragGuard;q. qosName=qosValue]/m_{i+1}

(4)如果 m_i 包含在组合片段中，而 m_{i+1}不包含在组合片段中

q=CheckQoSList (m_{i+1})

$\tau(s_i, s_{i+1})$=[q. qosName=qosValu]/m_{i+1}

(5)$\tau(s_i, s_j)$由消息 m_i 和 m_j 所处的组合片段确定

step4　确定 MSDFA 的终止状态

(1)如果 m_i 是组合片段 opt,loop,alt,par,break 内传递的最后一个消息，且 m_{i+1}不存在

T={s_n}∪{s_i} //s_i 和 s_n 为终止状态

(2)如果 m_i 是 alt 和 par 任意操作域内的传递的最后一个消息，且片段之后无消息

T={s_n}∪{s_i}//s_i 和 s_n 为终止状态

(3)若 m_i 在组合片段 opt|loop|break 外，m_{i+1}在组合片段 opt|loop|break 内，且片段之后无消息发生

T={s_n}∪{s_i} //s_i 和 s_n 为终止状态

通过以上算法，可以获得需求规约 p 语义的消息确定有限自动机表示 $MSDFA_p$。值得注意的是，如果 ESD 描述需求规约中包括 neg 组合片段，包含在该片段内的消息序列是不允许发生的，由该组合片段中的消息产生的序列都是不可执行的。图 1 中的需求规约 p 的消息自动机 $MSDFA_p$ 如图 2 所示。

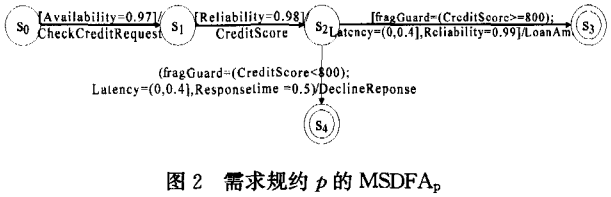

图 2　需求规约 p 的 $MSDFA_p$

5.3　Web 服务运行时验证

验证 Web 服务运行时行为是否满足已定义的需求规约，是验证 Web 服务在初始状态下根据交互消息，能否发生正确的迁移并且最终达到终止状态。基于自动机原理，本节给出了消息确定自动机识别语言的形式化定义，将 Web 服务运行时的交互消息构成消息序列，判断该消息序列是否为消息确定自动机的识别语言，以实现 Web 服务的运行时验证。下面给出消息确定有限自动机识别语言的定义。

定义 3(MSDFA 识别语言，L(MSDFA))　设 $AM=(S, M, F, Q, \Sigma, \tau, s_0, T)$ 是一个 MSDFA，对于 $\forall x \in \Sigma^*$，如果 $\tau(s_0, x) \in F$，则称 x 被 MSDFA 接受；如果 $\tau(s_0, x) \notin F$，则称 M 不接受 x，则

$$L(MSDFA)=\{x \mid x \in \Sigma^* \cap \tau(s_0, x) \in F\}$$

在 Web 服务运行时获取交互 SOAP 消息及计算待检测的 QoS 属性值，构建带有 QoS 的消息序列 *seq*，依据定义 3，判断 *seq* 是否是 MSDFA 识别语言：如果 *seq* 是 MSDFA 识别语言，说明 Web 服务运行时行为满足需求规约描述，否则 Web 服务运行时行为不满足预定义的需求规约，需要根据特定的错误和异常采取相应补偿措施。

以图 2 需求规约 p 的 $MSDFA_p$ 为例，由定义 3 得到 $MSDFA_p$ 的识别语言是

$L(MSDFA_p)$ = {([Availability = 0. 97]/CheckCreditRequest, [Reliability = 0. 98]/CreditScore, [fragGuard = (CreditScore >= 800), Latency = (0, 0. 4], Reliability = 0. 99]/LoanAmount); ([Availability = 0. 97]/CheckCreditRequest, [Reliability = 0. 98]/CreditScore, [fragGuard = (CreditScore < 800); Latency = (0, 0. 4], Responsetime = 0. 5]/DeclineReponse)}

而 Web 服务运行时获取的 SOAP 消息所构成的序列 *seq* 为：

seq=([Availability=0. 9]/CheckCreditRequest, [Reliability = 0. 95]/CreditScore, [fragGuard = (CreditScore>=800), Latency=(0, 0. 4], Reliability=0. 99]/DeclineReponse)

此处，*seq* 第一个消息中的 *Availability*=0. 9，与规约中的描述不一致，并且当 *fragGuard*=(*CreditScore*>=800)时，Web 服务并没有返回正确的消息。因此，由定义 3 可得 *seq* 不是 $MSDFA_p$ 的可识别语言，说明 Web 服务运行时行为没有发生正确的迁移，在初始状态下根据接收到的消息无法达到终止状态，即 Web 服务运行时行为与需求规约描述不一致。

5.4　RVT4WS 工具

为了支持本文提出的使用 ESD 描述需求规约的一致性验证方法，本节设计一个 Web 服务运行时验证工具 RVT4WS(Runtime Verification Tool for Web Services)。RVT4WS 支持使用 ESD 对 Web 服务需求规约进行建模，并通过构造算法获得 ESD 的语义模型，最后通过对 Web 服务运行时 SOAP 消息的监听和 QoS 属性的计算，获得 Web 服务运行时行为与需求规约的一致性验证结果。RVT4WS 是在 Windows 环境下使用 Eclipse STP+GEF 开发的原型工具，数据库采用 MySQL 5. 1. 6。目前该工具的开发处于集成测试阶段。

RVT4WS 的设计框架如图 3 所示。RVA4WS 主要由 6 个模块组成：需求规约管理模块、自动机管理模块、QoS 属性管理模块、SOAP 消息监听模块、验证模块和检测结果报告模块，下面分别对上述模块的功能进行说明。

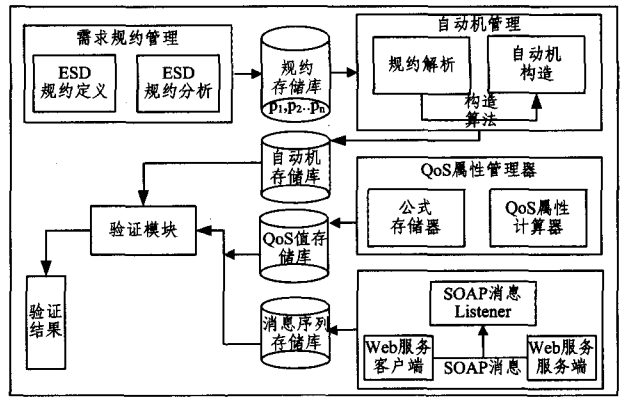

图 3　Web 服务运行时验证工具架构

(1)需求规约管理模块。在该模块中,通过图形用户界面使用 ESD 对 Web 服务需求规约进行建模,检验是否满足 ESD 语法,并将需求规约存储在规约存储库中。

(2)自动机管理模块。该模块是实现需求规约模型的语义表示。首先从规约存储库中提取需求规约记录,根据 MSDFA 构造算法,构建相应的消息确定有限自动机。

(3)QoS 属性管理模块。QoS 属性管理模块用于计算需求规约中的 QoS 属性值。该模块由公式存储器和 QoS 属性计算器两部分组成。公式存储器存储 QoS 属性的计算公式,QoS 属性计算器记录相应的时间和时刻,根据公式计算 QoS 属性值,并存储在 QoS 值存储库中。

(4)SOAP 消息监听器。SOAP 消息监听器实现各对象之间交互消息的监听,并将其存储到当前消息存储序列中。值得注意的是,SOAP 消息监视器可自动过滤与当前待检验需求规约不相关的消息。

(5)验证模块。验证模块用于 Web 服务运行时行为的验证。以 QoS 属性、消息序列和消息确定自动机模型为输入,判断带有 QoS 属性的消息序列是否是消息确定自动机的识别语言。

(6)验证结果报告模块。该模块将验证结果通过图形用户接口回显给用户,用户根据验证结果做出相应的决策。

结束语 对 Web 服务进行运行时验证,保证 Web 服务运行时行为与需求规约一致,是 Web 服务领域的研究热点之一。针对现有研究成果中较少对 Web 服务的功能性和 QoS 进行统一验证的问题,本文提出基于 UML2.0 序列图的运行时验证方法。首先对 UML2.0 序列图进行扩展,使其能够支持 Web 服务功能和 QoS 需求规约的统一描述。由于序列图缺乏精确语义,提出构造算法,将统一描述功能需求和 QoS 需求的 ESD 转换为 MSDFA。最后基于 ESD 的自动机语义模型,依据验证准则完成 Web 服务运行时行为与需求规约的一致性验证。设计并实现了 Web 服务运行时验证工具 RVT4WS,为本文方法提供支持。实例表明,ESD 有效支持 Web 服务的功能和 QoS 需求规约的统一描述和验证。基于 RVT4WS,对 Web 服务验证过程中发现的错误和异常提供补偿和异常处理机制是下一步考虑的工作。

参 考 文 献

[1] Christopher F, Joel F. What are Web services? [J]. Communications of the ACM,2003,46(6):31

[2] Leucker M,Schallhart C. A brief account of runtime verification [J]. The Journal of Logic and Algebraic Programming,2009,78(5):293-303

[3] 张岩,梅宏. UML 类图中面向非功能属性的描述和检验[J]. 软件学报,2009,20(6):1457-1469

[4] Ran S. A Model for Web Services Discovery With QoS[J]. ACM Special Interest Group on Electronic Commerce,2003,4(1):1-10

[5] Hamilton K,Miles R. Learning UML2.0 [M]. O'Reilly Media,2006:179-222

[6] ITU-TS. Message Sequence Chart 1996,ITU-TS Recommendation Z.120[R]. Geneva,TU-TS,1996

[7] Damm W,Harel D. LSCs:Breathing Life into Message Sequence Charts[J]. Formal Methods in System Design,2001,19(1):45-80

[8] David H,Shahar M. Assert and nagate Revisited:Modal semantics for UML sequence diagrams[J]. Software and Systems Modeling,2008,7(2):237-252

[9] Autili M,Inverardi P,Pelliccione P. A Scenario Based Notation for Specifying Temporal Properties[A]//Proceeding of the 2006 International workshop on Scenarios and state machines:models,algorithms,and tools,2006[C]. NewYork:ACM,2006:21-28

[10] Ameedeen M A,Bordbar B. A Model Driven Approach to Represent Sequence Diagrams as Free Choice Petri Nets[A]// The 12th International IEEE Conference on Enterprise Distributed Object Computing,2008[C]. NewYork:IEEE Computer Society,2008:213-221

[11] Li X, Qiu X, Wang L, et al. UML Interaction Model-driven Runtime Verification of Java Programs [J]. The Institution of Engineering and Technology,2011,5(2):142-156

[12] 雷丽晖,段振华. 一种基于扩展有限自动机验证组合 Web 服务的方法[J]. 软件学报,2007,18(12):2980-2990

[13] Jocelyn S,Yuan G,Marsha C,et al. Runtime Monitoring of Web Service Conversation[J]. IEEE Transaction on Services Computing,2009,2(3):223-244

[14] 苏焕成,黄志球,刘林源. 基于接口自动机的 BPEL4WS Web 服务组合形式化模型[J]. 计算机应用研究,2009,26(5):1774-1777

[15] 骆翔宇,轩爱成,沙宗鲁. 基于时间自动机的 Web 服务模型检测[J]. 计算机科学,2010,37(8):139-142

[16] 肖芳雄,李燕,黄志球,等. 基于时间概率代价进程代数的 Web 服务组合建模和分析[J]. 计算机学报,2012,35(5):918-935

[17] 朱俊,郭长国,吴泉源. 基于 CPN 的服务交互行为关键属性的运行时确保机制[J]. 电子学报,2011,39(5):1064-1071

[18] Walid G l,Sami B,Mohsen R. Event-based Design and Runtime Verification of Composite Service Transactional Behavior [J]. IEEE Transactions on Services Computing,2010,3(1)32-45

[19] 袁敏,黄志球,胡军. 跨组织多业务事务建模与验证方法[J]. 软件学报,2012,23(3):517-538

[20] Dimitris D, Ervin R, Dimitrios K. Runtime Verification of Behavioral Conformance for Conversational Web Services[A]// The 7th IEEE European Conference on Web Services, 2009 [C]. NewYork:IEEE Computer Society,2009:139-147

[21] Andreas B,Martin L,Christian S. Runtime Verification for LTL and TLTL [J]. ACM Transactions on Software Engineering, 2011,20(14):14-77

[22] 石慧娟,戎玫,张广泉,等. 基于 XYZ/ADL 的异步 Web 服务组合描述与验证[J]. 计算机科学,2011,38(12):139-143