基于 TLA 的 Web 服务组合研究

王一飞[1],李迎春[2],许 迅[1]
(1. 盐城工学院 信息工程学院,江苏 盐城 224053;2. 盐城市广电局,江苏 盐城 224001)

摘 要:随着以 Web 服务为基础的面向服务的体系结构的发展,如何有效地组合自治的、分布的、不同功能的 Web 服务以构建新的企业业务应用逐渐成为研究热点。简单介绍了 Conversation 模型的概念,利用该模型作为中介,将组合 Web 服务的 BPEL 描述转化为 Conversation 模型,再将 Conversation 模型转换为 TLA 表达式,并给出了具体的算法。

关键词:TLA;BPEL4WS;Conversation 模型;Web 服务组合验证
中图分类号:TP393 **文献标识码**:A

Research of Web services composition based on TLA

WANG Yi Fei[1],LI Ying Chun[2],XU Xun[2]
(1. Dept. of Information Engineering, Yancheng Institute of Technology, Yancheng 224053, China;
2. Yancheng Broadcasting Television Bureau, Yancheng 224001, China)

Abstract: With the evolution of service-oriented architecture, providing support for compositing distributed and autonomous Web services into business applications has become a key area in the software engineering research. The main functions including the following items. Firstly, the concepts of BPEL4WS and TLA are described. And then the algorithms for the automatic translation from BPEL4WS to TLA are discussed and suggested.

Key words: TLA; BPEL4WS; conversation model; Web service composition verification

TLA[1](Temporal Logic of Actions)是由 Lamport 提出的用来描述和推理并发系统的一种时序逻辑。Lamport 提出把并发系统的性质分为两类,即安全性(safety)和活性(liveness)。非正式地讲,安全性描述的是"系统不期望发生的事件永远不会发生",而活性描述的是"系统预期行为终究会发生"。时序逻辑将一阶逻辑加以扩充,增加了表示时序的模态算子,可以方便地描述并发和反应系统的这两类约束性质。本文中将主要利用 TLA 讨论系统的 safety 属性。

Web 服务是一种自包含、自描述、模块化的程序,它吸收了分布式计算、Grid 计算和 XML 等各种技术的优点,解决了异构分布式计算以及代码与数据重用等问题,具有高度的互操作性、跨平台性和松耦合性,引起了世界范围内学术界和工业界的极大兴趣[2]。

Web 服务业务流程执行语言 BPEL4WS(Business Process Execution Language for Web Services)[3] 能够实现 Web 服务调用、操纵数据、抛出故障或终止一个流程等不同功能,并且可以将它们连接起来,从而创建出复杂的流程。BPEL4WS 是 WSFL 和 XLANG 融合的产物,已成为学术界和工业界 Web 服务组合的主要描述语言之一,它集 WSFL 和 XLANG 两家之长(前者支持面向图形的流程,后者则支持流程的结构化构造)于一身,以非常自然的方式实现了各种类型的业务流程的结合。作为可执行流程的实现语言,BPEL4WS 的作用是将一组现有的服务整合起来,从而定义一个新的 Web 服务。与其他任何 Web 服务一样,整合服务的接口也被描述为 WSDL 的 portType 集合。整合结果(称为流程)指明了服务接口与整合总体执行的配合情况。

1 Web 服务组合的 Conversation 模型

1.1 Conversation 模型

Web 服务的 Conversation 模型[4]是 Tevifik Bultan 等人提出的一种 Web 服务组合模型,它从全局的角度描述了服务间的交互行为,从而为分析组合服务的正确性、可达性和等价性等问题提供了一个很好的框架。在该模型中,各个原子服务间通过发送和传递消息进行交互。由于是异步传送,每个服务配备一个 FIFO 队列,用于存放接收的消息,如图 1 所示。

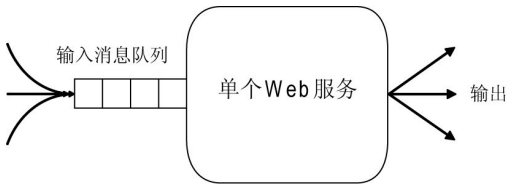

图 1 Conversation 模型中的单个服务

单个服务的内部流程使用 mealy 自动机描述。如图 2 所示。其中“?”表示接收一个消息,“!”表示发送消息。

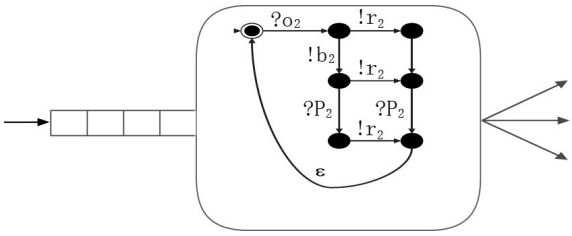

图 2 Web 服务的内部流程

1.2 Conversation 模型的形式化描述

下面给出 Conversation 模型的形式化描述[5]。在该模型中,一个组合 Web 服务可以表示为元组 $S=((P,M),A_1,A_2,\cdots,A_n)$。$(P,M)$中 P 是单个原子服务的集合,M 是所有消息类型的集合。A_1、A_2、…、A_n 对应着 n 个原子服务,对于第 i 个服务 A_i 是一个 mealy 自动机 $(M^{in}_i,M^{out}_i,T_i,s_i,F_i,\Delta_i)$。这里,$M^{in}_i(M^{out}_i)$是输入和输出的消息类型,$T_i$ 表示所有状态的集合,s_i 是初始状态的集合,F_i 表示最终状态的集合。Δ_i 为状态转换函数。Δ 函数有以下 3 种形式($\tau\in\Delta_i$, $q_1\in T_i$ 表示转换开始状态,$q_2\in T_i$ 表示转换完成后的状态):

(1) $\tau=(q_1,!m,q_2)$,发送消息 m 到接收者队列,并实现 q_1 到 q_2 的转换;

(2) $\tau=(q_1,?m,q_2)$,实现 q_1 到 q_2 的转换,并从消息队列里删除 m 类型的消息;

(3) $\tau=(q_1,\varepsilon,q_2)$, 实现 q_1 到 q_2 的转换, 不做任何操作。

对于上述的一个组合的 Web 服务 $S=((P,M),A_1,A_2,\cdots,A_n)$,有 $\gamma=(Q_1,t_1,\cdots,Q_n,t_n,w)$。对于第 i 个服务,Q_i 表示服务的消息队列,t_i 表示服务的状态,w 表示全局角度记录的 Web 服务交互的消息序列。如果存在转换函数 $\tau\in\Delta_i$, 转换执行前组合服务表示为 γ, 转换执行后表示为 γ',则称为组合服务的一次执行,表示成 $\gamma\rightarrow\gamma'$。其中,对应于 $\tau=(q_1,!m,q_2)$,其发送的消息将分别加入接收服务和全局记录 w 的消息序列中。

在上述框架下,组合服务 S 的一次运行可以表示为序列 $\gamma=\gamma_0,\gamma_1,\gamma_2,\cdots,\gamma_{|\gamma|-1}$,且满足以下条件:

(1) $\gamma_0=(\varepsilon,\overrightarrow{\perp},s_1,\cdots,\varepsilon,\overrightarrow{\perp},s_n,\varepsilon)$。这时,$s_i$ 是第 i 个原子服务的初始状态,且每个服务的输入消息队列和全局消息队列都为空;

(2) $\gamma_i\rightarrow\gamma_{i+1},i\in[0\ |\gamma|-1)$;

(3) $\gamma_{|\gamma|-1}=(\varepsilon,s'_b\cdots,\varepsilon,s'_n,w)$ 这时,s'_i 是第 i 个原子服务的最终状态,且所有原子服务的输入消息队列都为空。

组合服务 S 一次运行后 w 消息序列称为 S 的一个会话(Conversation),使用 $L(S)$表示 S 所有会话的集合。在本论文中, 主要研究如何在 Conversation 模型下使用 TLA 对系统进行描述并验证,而 $L(S)$相关的性质的讨论将放到以后进行。

2 BPEL 转化为 Conversation 模型的研究

2.1 Conversation 模型的形式化描述

BPEL 语言可以将现有的 Web 服务整合起来, 从而定义出一个新的服务。在一个 BPEL 文档里, 可以获得以下信息:该流程由哪几个 Web 服务组合而成,这几个 Web 服务按怎样的工作流结构进行组合,组合服务执行的每一步中原子服务之间是如何交互的。通过这些信息, 可以将 BPEL 描述文档转化为一个 Conversation 模型。

假设设计 BPEL 文档时知道原子服务是如何交互的, 一种解决方法是将所有原子服务都描述成 BPEL 流程, 每个 BPEL 流程都给出与组合服务以及其他原子服务是如何交互的。这样,整个组合服务被看成一个并发系统, 分别解析每一个原子服务以及组合服务的 BPEL 文档,根据所得的信息,就可以将其完整地转化为一个 Conversation 模型。

2.2 解决方案及算法

组合 Web 服务的 Conversation 模型中, 每一个原子服务都是含有 FIFO 队列的 Mealy 自动机,因此 BPEL 转化为 Conversation 模型的问题也就变为如何将 BPEL 流程转化为一个这样的自动机。问题的解决分为以下两步:

(1) 按一定的规则生成 Conversation 模型中传递的消息;

(2) 根据每个原子服务的 BPEL 描述文档生成相应的 Mealy 自动机。

2.3 消息的生成

BPEL 中有<invoke>、<receive>和<reply>等活动,针对每个活动中包含的信息可以生成相应消息。生成消息的过程中,需要保证同一消息的发送和接收方对应活动所产生的名称是相同的,即要求组合服务以及所有原子服务的 BPEL 文档中的<partner>以及<serviceLinkType>元素保持一致。消息生成的算法如下:

(1)<invoke>活动:生成消息 sOwner_Operation_IN。这里,sOwner 是被调用的 Web 服务的进程名,Operation 是该活动的<operation>属性值;

(2)<receive>活动:生成消息 Owner_Operation_IN。这里,Owner 是<receive>活动所在 Web 服务的进程名,Operation 是该活动的<operation>属性值;

(3)<reply>活动:生成消息 Owner_Operation_OUT。这

里,Owner 是<reply>活动所在 Web 服务的进程名,Operation 是该活动的<operation>属性值。

2.4 将 BPEL 文档转化为 Mealy 自动机

首先,将基本活动转化为一个自动机[6],以下对不同的基本活动分别进行讨论:

(1)<assign>活动:构造自动机,该自动机包含 2 个状态,设 q_1 为初态,q_2 为终态,则状态转移函数为 $\tau=(q_1,\varepsilon,q_2)$;

(2)<receive>活动:构造自动机,设 q_1 为初态,q_2 为终态,状态转移函数为 $\tau=(q_1,?\text{Owner\_Operation\_IN},q_2)$;

(3)<invoke>活动:构造自动机,设 q_1 为初态,q_2 为终态,状态转移函数为 $\tau=(q_1,?\text{sOwner\_Operation\_IN},q_2)$;

(4)<reply>活动:构造自动机,设 q_1 为初态,q_2 为终态,状态转移函数为 $\tau=(q_1,!\ \text{sOwner\_Operation\_IN},q_2)$。

其次,对于结构化活动,算法的基本思想是按一定的规则,将其所包含活动对应的自动机组合成一个更大的自动机[6]。这里仅讨论<sequence>和<flow>这两种结构化活动,其他的结构化活动可以使用类似的方式处理:

(1)<sequence>活动:只要将包含的所有活动对应的自动机首尾相连即可;

(2)<flow>活动:在<flow>中,首先找出满足 link 条件的一系列活动,再将这些活动对应的自动机做笛卡儿乘积(cartesian product)。

3 Conversation 模型转化为 TLA

得到组合 Web 服务的 Conversation 模型后,可以按照下面的算法将 Conversation 模型转化为 TLA 表达式:

设基于 Conversation 模型的组合服务 $S=((P,M),A_1,A_2,\cdots,A_n)$

(1)针对每个服务 A_i 引入变量 $Curstate_i$,Q_i。这里,$Curstate_i\in T_i$,Q_i 表示该服务的消息队列;

(2)初始化。$\forall A_i\in S$,有 $Curstate_i=s_i$,$Q_i=<>$,将它们合取,得到 *Init* 表达式;

(3)对于每个服务 A_i,$\forall\tau_j\in\Delta_i$

①如果存在 $\tau_j=(q,!\ msg,q_2)$,且消息是发送给服务 A_k 的,则添加表达式:

$A_i\_step\_\tau_j\equiv$

$\wedge Curstate_i=q_1$

$\wedge Appand(Q_k,msg)$

$\wedge Curstate'_i=q_2$

$\wedge UNCHANGED<other\ variables>$

该式将 $Curstare_i$ 状态由 q_1 变为 q_2,同时将消息 *msg* 存放到目的服务 A_k 的队列中,并保持其他变量的值不发生改变。

②如果存在 $\tau_j=(q_1,?msg,q_2)$,则添加表达式:

$A_i\_step\_\tau_j\equiv$

$\wedge Curstate_i=q_1$

$\wedge Appand(Q_i)=msg$

$\wedge Tail(Q_i)$

$\wedge Curstate'_i=q_2$

$\wedge UNCHANGED<other\ variables>$

该式将 $Curstate_i$ 状态由 q_1 变为 q_2,检测服务 A_i 的队列头是否为 *msg* 消息,如果是则从队列中删除该消息,并保持其他变量的值不发生改变。

(4)针对组合服务一次执行终止时的状态,添加表达式:

$S\_step\_rest\equiv$

$\wedge\forall A_i\in S:Curstare_i\in F_i,Q_i=<>$

$\wedge\forall A_i\in S:Curstate'_i=s_i,Q_i=<>$

这个式子表示当所有服务都进入自己的终态且各自队列为空时,所有变量将恢复到初始状态,并开始新一轮的运行周期。

(5)将步骤(3)和(4)中得到表达式析取,得到 Next 表达式;

(6)给出整个系统的 TLA 表达式。

Web 服务能够较好地解决异构应用之间、松散耦合环境下的互操作、集成和协作问题,成为国内外软件技术研究的重要方向。Web 服务的组合是正在兴起的新技术,它将彻底改变提供电子商务和客户软件应用的方式,是国内外在信息集成、软件工程等领域的关注的焦点,也是 Web 服务技术的主要发展方向之一。

参考文献

[1] LAMPORT L.Specify system:the TLA+ language and tools for hardware and software engineers[M].Addision Wesley,2002.

[2] 李曼,王大治,杜小勇,等. 基于领域本体的 Web 服务动态组合[J].计算机学报,2005,28(4):644-650.

[3] Business process executive language for Web service (BPEL4WS) version 1.0 [S/OL].http://www-106.ibm.com/developerworks/library/ws-bpel.

[4] BULTAN T,FU X,HULL R,et al.Conversation specification:a new approach to design and analysis of E-service composition [C].In:Proc. of the 12th International World Wide Web Conference,2003.

[5] FU X,BULTAN T,SU J.Conversation protocols:a formalism for specification and verification of reactive electronic services[C].In:Proc. of the 8th International Conference on Implementation and Application of Automata, 2003.

[6] FU X,BULTAN T, SU J.Analysis of interacting BPEL Web services[C].In:Proceedings of the 13th International World Wide Web Conference,2004.

(收稿日期:2009-10-20)

作者简介

王一飞,男,1982 年生,硕士,主要研究方向:Web 服务组合、工作流、TLA;

李迎春,女,1969 年生,硕士,主要研究方向:Web 服务匹配;

许迅,男,1979 年生,硕士,主要研究方向:Web 服务选择、TLA。