

文章编号:1001-9081(2010)07-1959-03

基于 BPEL 的流程数据竞争问题研究

杨书新,李淑芝,张永进

(江西理工大学 信息工程学院,江西 赣州 341000)

(yimuyunlang@sina.com)

摘要:服务组合就是将多个基本的 Web 服务组合成可以满足用户需求的增值服务,为企业或企业间的业务过程集成提供技术支持。作为一种流程驱动的服务组合描述语言,BPEL 和其他采用不同方式的服务组合语言一样面临着流程描述的正确性问题,在执行服务组合之前要对其控制流和数据流进行分析和验证。数据竞争是 BPEL 数据流分析中亟需解决的问题之一。围绕 BPEL 流程数据竞争问题,在对 BPEL 规范中的活动类型、活动并发性进行分析基础上,给出了数据竞争的形式化描述,提出一种检测方法。该方法基于 XML 节点树的特性、活动之间的并发性以及活动关联的消息来判断是否存在潜在的数据竞争。最后,通过一个订单处理流程实例,说明了该方案的有效性。

关键词:业务流程执行语言;Web 服务;服务组合;数据竞争

中图分类号: TP311 **文献标志码:** A

Study of data race for process based on BPEL

YANG Shu-xin, LI Shu-zhi, ZHANG Yong-jin

(School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou Jiangxi 341000, China)

Abstract: Composing the existing Web services to satisfy a new value-added for user requirement is called services composition. It provides technical support for business process integration within an enterprise or across multiple enterprises. As a process-driven services composition description language, correctness is a key challenge to BPEL (Business Process Execution Language) the same as other description languages in the domain of services composition. Therefore, analysis and check of control flow and data flow is an essential step before execution. Data race is a well-known issue for BPEL. In this paper, our goal is automatic detection of the potential data race in a process. Firstly, a formal description for data race was given by analyzing the event types, concurrency specified in the BPEL. Furthermore, based on exploring the characteristics of XML nodes tree, concurrency of two activities, and messages correlative with activities, a detection algorithm was designed to find out data race in a process. Finally, an order processing example was demonstrated to illustrate the validity of the proposed solution.

Key words: business process execution language; Web service; Web services composition; data race

0 引言

Web 服务是通过 Internet 技术传递的、松散耦合的软件组件,通过接口由 Internet 将业务逻辑发布为服务,使用者通过 Internet 协议来查找、订阅和调用这些服务。以 Web 服务为基础的分布式计算模式——面向服务计算 (Service Oriented Computing, SOC) 强调以重用为中心,被认为是解决在分布、动态、异构环境下的企业以及企业间数据应用和系统集成问题的有效手段^[1-2]。

根据业务逻辑,将 Web 服务按照一定顺序集合在一起形成特定的控制流和数据流,从而完成某个任务称为服务组合。它是面向服务的计算范型中实现资源聚合与应用集成的主要模式,为企业或企业间的业务过程集成提供了技术支持。在服务组合方面,Rao 等人^[3]从使用技术的角度将 Web 服务组合归纳为基于工作流和人工智能规划两大类。就目前技术现状而言,基于工作流的服务组合实用化程度较高。

Web 服务的业务流程执行语言 BPEL4WS (Business Process Execution Language for Web Services) 目前已经成为最

重要的业界规范,它是一种 XML 语言,用来描述可执行业务流程的行为。

数据流和控制流是 BPEL 执行的所需的基本信息,BPEL 流程正确性可归结为控制流和数据流正确性问题,它在部署前如果没有验证可能会导致流程不能正确执行、数据不一致等错误^[4-6]。对于庞大的、复杂的模型来说,人工手动地去分析流程比较繁琐和耗时。对控制流分析和验证的技术主要有进程代数、Petri 网、自动机,对数据流的研究主要集中在消息匹配^[7-8]、数据依赖^[9-10]和数据竞争^[11]。文献[11]借多线程程序依赖图 (Threaded Program Dependency Graph, TPDFG) 思想,采用静态分析和图理论方法,在数据竞争检测之前将 BPEL 转换为 BAG (Bpel Activity Graph)、BSG (Bpel Segment Graph),通过活动之间的可达性和活动所在的分支号来判断并发的可能性。该方法实现环节比较复杂,对于 switch 或 pick 的分支活动简单的采用分支号来判断并发性,不能有效、合理地检测出嵌套在 switch 或 if、pick 的 flow 中并发的活动。

针对 BPEL 数据竞争问题,本文在对 BPEL 2.0 规范^[12]中活动分析的基础上,提出一种基于 XML 节点树分析的数据

收稿日期:2010-01-19;修回日期:2010-03-04。

基金项目:江西省自然科学基金资助项目(2009GZS0043),江西省教育厅科技项目(GJJ09247;GJJ09522)。

作者简介:杨书新(1978-),男,江西九江人,副教授,博士,CCF 会员,主要研究方向:SOA、工作流管理;李淑芝(1964-),女,江西赣州人,教授,主要研究方向:SOA、数字水印;张永进(1987-),男,河北邢台人,主要研究方向:服务组合。

竞争检测方法。

1 BPEL 流程数据竞争分析

1.1 BPEL 流程活动类型

BPEL 模型在结构上是一种基于图形结构和块状结构混合的工作流模型,各个活动之间的关系通过流程结构化元素定义,结构化的活动规定了一组活动发生的顺序。

BPEL4WS 里面的活动有基本活动和结构化活动。基本活动定义了组成业务流程的任务,包括 Receive、Reply、Invoke、Assign、Wait、Empty。结构化的活动包括活动间的顺序控制(如 sequence、if、while、repeatUntil、forEach 的 serial variant)、活动间的并发和同步(flow、forEach 的 parallel variant)、基于外部事件的不确定的选择(pick)。流程(process)中的执行顺序可通过分析 BPEL 模型中的结构化元素和 flow 的 link 元素来决定。

定义 1 Web 服务自动机。Web 服务自动机定义为一个 Mealy 机:

$$M = \langle Q, \Sigma_{in}, \Sigma_{out}, \delta, \lambda, q_0, F \rangle$$

其中 Q 是有限状态集合, $Q = \{q_0, q_1, \dots, q_n\}$; Σ_{in} 是有限输入字符集合, $\Sigma_{in} = \{\sigma_0, \sigma_1, \dots, \sigma_n\}$; Σ_{out} 是有限输出字符集合, $\Sigma_{out} = \{\beta_0, \beta_1, \dots, \beta_n\}$; δ 为状态转移函数, $\delta: Q \times \Sigma_{in} \rightarrow 2^Q$; λ 为输出函数, $\lambda: Q \times \Sigma_{in} \rightarrow \Sigma_{out}$; q_0 为初始状态, $q_0 \in Q$, F 为终止状态集合, $F \subseteq Q$ 。

一个服务和其他服务交互过程是接受一个输入后产生输出,并从状态 q_i 转变为 q_j ,意味着对消息变量进行访问。

定义 2 消息访问。消息访问 $MA = \{ma \mid ma = (n, m, access\ model)\}$ 。其中 $n \in N, m \in M, M = \Sigma_{in} \cup \Sigma_{out}, access\ model \in \{read, write\}, N$ 为组成流程的活动节点集。 $(n, m, read)$ 表示当活动 n 被激活执行时,从输入参数 $m \in \Sigma_{in}$ 取值, $(n, m, write)$ 表示当 n 对应的活动实例被执行完后,向输出参数 $m \in \Sigma_{out}$ 赋值。

当 n 在一个谓词或分配中访问消息 m ,它读消息变量 m 的值,属于这种类型的有 assign 活动(m 在赋值表达式右边)、while(子元素 condition 的布尔表达式)、if(子元素 condition 的布尔表达式)、forEach、在 sourceLink 或 targetLink 使用 m 的活动、把 m 当做输入变量的 invoke 活动、reply 等;写消息变量 m 的值,属于这种类型的有 receive、把 m 当做输出变量的 invoke、pick、assign(m 在赋值表达式左边)等。

1.2 BPEL 流程的数据竞争问题分析

图 1 为一 BPEL 流程,图中矩形方框、虚线、虚线方框表示 BPEL 活动、数据流、待访问的消息变量。该流程执行顺序为 Receive、Flow_1、Reply。

由于 BPEL 是基于消息的流程描述语言,在数据处理上是基于消息的发送与接收模式,消息之间的引用可能是消息的一部分,如消息 m_1 和消息 m_2 之间的交互 $m_1.part1 = m_2.part2$ 。因此不能像传统的数据分析方法那样直接分析出消息的定义和引用,在分析之前需要将消息展开成一组变量的形式^[13],如图中的 $m.p1$ 。

在 Flow_1 执行时,Invoke_1 和 Invoke_2 可以并发的对 $m.p1$ 进行写入操作,可以形式化描述为 $ma1 = (Invoke_1, m.p1, write), ma2 = (Invoke_2, m.p1, write)$ 。多个用户程序并发存取同一数据的情况,若对并发操作不加控制就可能会

存取不正确的数据,破坏数据库的一致性。

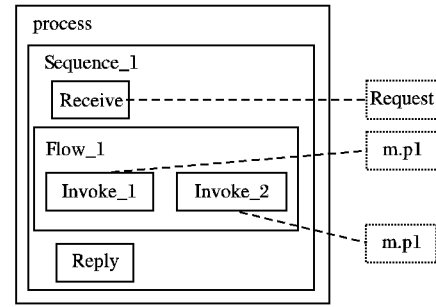


图 1 存在数据写竞争的流程

在 BPEL 流程 P 中,两个活动之间的数据竞争当且仅当满足以下两个条件。

- 1) 存在并发关系且访问相同的变量:
 $(i \parallel j) \wedge ((\Sigma(i) \cap \Sigma(j)) = V \neq \emptyset)$
- 2) 对共同的变量至少存在一个写访问:
 $W(i, v) \vee W(j, v); v \in V$

其中: i, j 表示标识号为 i 和 j 的活动; $\Sigma(i)$ 为活动 i 访问的消息变量; $\Sigma(i) = \Sigma_{in}(i) \cup \Sigma_{out}(i)$; $\Sigma(j)$ 为活动 j 访问的消息变量; $\Sigma(j) = \Sigma_{in}(j) \cup \Sigma_{out}(j)$; $W(i, v)$ 为活动 i 对共同的消息变量 v 进行写操作, $W(i, v) = (i, v, write)$; $W(j, v)$ 为活动 j 对共同的消息变量 v 进行写操作, $W(j, v) = (j, v, write)$ 。

2 BPEL 流程数据竞争检测方法

在 BPEL 流程中,所有直接嵌套在 flow 或 forEach 中的活动可以并发执行,其中 link 只能定义在 flow 之中,每个 link 连接一个源活动和目标活动。根据以上的分析,下面给出一个数据竞争检测算法。

输入: BPEL 文件 P 。

输出: 数据竞争信息 DRI。

begin

PairCheck = { }; DRI = { }

//初始化: PairCheck 为存放两个活动节点和检测状态的三维数

//组, DRI 为存放数据竞争信息的三维数组

FlowList = Find(P , Flow) //选出文件 P 中所有叫做 Flow 的元

素,在 XPath 表示为“// Flow”

For each node(cof) in FlowList

actList = getDescendant(cof); //获取 Flow 的子孙活动节点

for each descendant act(da_i) in actList

for each descendant act(da_j) in actList

if (!isChecked(da_i , da_j))

//在 PairCheck 中判断 da_i , da_j 是否检测过

{ if (conflict(da_i , da_j))

//判断 da_i , da_j 访问的消息是否存在不为空的交集

//且有一个为写方式

{ if (!isReachable(da_i , da_j) && !isParentSelect(da_i , da_j))

//isReachable(da_i , da_j) 判断 da_i 到 da_j 或

// da_j 到 da_i 的可达性

// isAncestorSelect(da_i , da_j) 判断两者共同的第

//一个祖先节点是否 If 或 Pick 选择型

DRI = DRI + (da_i , da_j , v) //将两个活动节点

//和产生竞争的消息变量添加到 DRI

}

PairCheck = PairCheck + (da_i , da_j , checked)

//将两个活动节点的检测状态设为 checked}

End for

End for

```

End for
forEachList = Find(P, forEach)//选出文件 P 中所有叫做 forEach
的元素,在 XPath 表示为“// forEach”
For each node(cofe) in forEach the serial variant
  If(cofe.parallel == yes)
    actList = getDescendant(cofe);//获取 forEach 的子孙活动节点
    for each descendant act(da_i) in actList
      for each descendant act(da_j) in actList
        if(!isChecked(da_i, da_j))
          //在 PairCheck 中判断 da_i, da_j 是否检测过
          {if (conflict(da_i, da_j)) //判断 da_i, da_j 访问的消息
            是否存在不为空的交集且有一个为写方式
            if(!isReachable(da_i, da_j) && !isParentSelect(da_i, da_j))
              DRI = DRI + (da_i, da_j, v) //将两个活动节点和
              //产生竞争的消息变量添加到 DRI
            }
          PairCheck = PairCheck + (da_i, da_j, checked)
          // 将两个活动节点的检测状态设为 checked}
        End for
      End for
    End for
  End

```

在数据竞争检测算法中,从 P 中取出 Flow、ForEach 中的元素,并对其子孙活动节点进行判断,如果 da_i, da_j 满足在两者之间不存在一条可访问的路径(即不可达)、两者共同的第一个祖先活动节点是非选择型和访问的消息存在不为空的

交集且有一个为写方式,则将其写入 DRI 中。isReachable(da_i, da_j) 根据节点生成树的方法在 Flow 或 ForEach 构造块中求得所有从 da_i 可达的节点集,并判断 da_j 是否属于该节点集,反之观察 da_i 是否属于 da_j 可达的节点集。isAncestorSelect(da_i, da_j) 函数中两者共同的第一个祖先节点采用文献[9]提出的方法获取。考虑到 Flow、ForEach 元素内部相互嵌套的可能性,为避免重复的检测,使用 PairCheck 记录检测信息,如果 isChecked(da_i, da_j) 为 true,则忽略此次检查。相比于已有的一些方法,本算法不需对每个活动节点进行检测,只对 Flow、ForEach 的子孙活动进行检测,时间复杂度更低,从而提高了检测效率。

3 实例分析

以订单处理流程为例,如图2所示,图中虚线表示数据流。在接收用户订单(receive)请求后,两个任务可以并行处理,一个是根据用户选择的发货方式(如邮政、快递、EMS)来计算货运价格(ISHipping),另一个并行的任务是计算折扣。当购买金额大于1000时,并发执行 Sequence_2 和 VIP 级别折扣计算(IVip),否则直接打 0.95 折(Assign_2)。在 Sequence_2 里,根据金额计算折扣(IDiscount)、礼品赠送清单(ICoupons)串行执行。IVip 和 IDiscount 之间存在 Link,Link 的条件是 Vip 折扣大于 0.88,在这里有意将之忽略。最后发一个订单通知给客户(ISHippingNotice)。

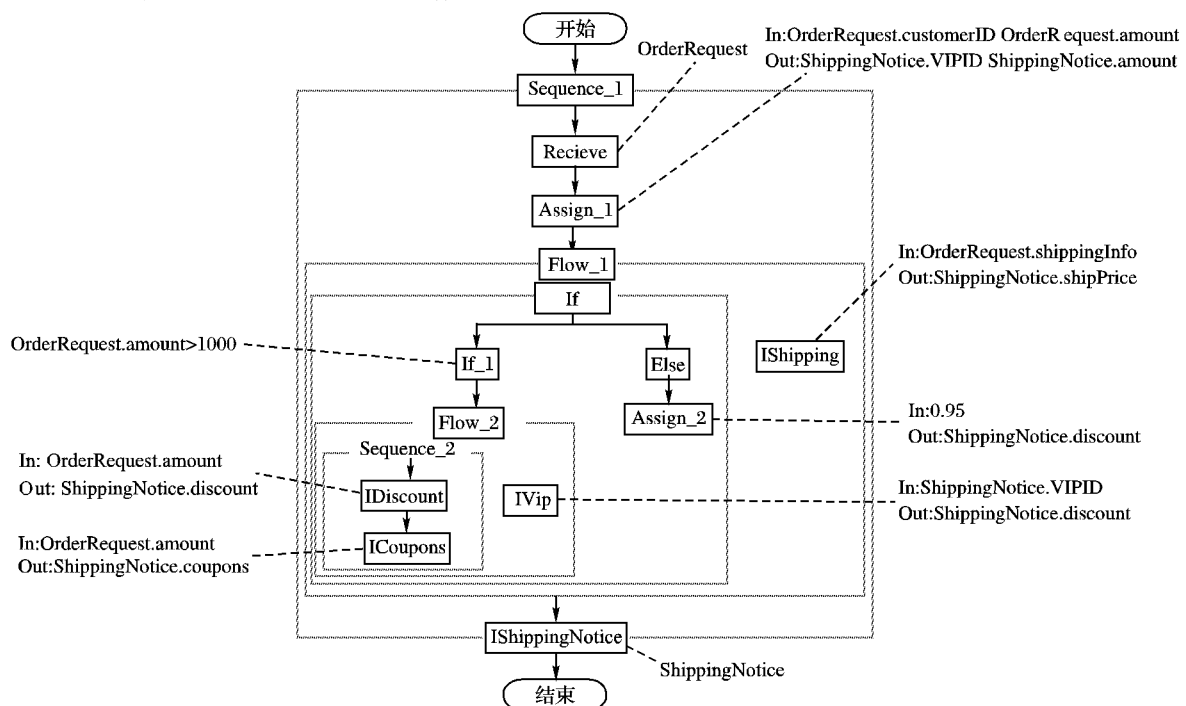


图2 订单处理流程

根据数据竞争检测算法,查找到 Flow_1 和 Flow_2 节点, Flow_1 的子孙活动节点包括 IShipping、If、Assign_2、Flow_2、Sequence_2、IVip、IDiscount、ICoupons,其中 IVip 和 IDiscount 之间不可达,两者共同的第一个祖先节点是 Flow_2(非选择),访问的消息交集为 {ShippingNotice.discount},且至少一个为写方式,因此 $DRI = \{IVip, IDiscount, ShippingNotice.discount\}$ 。Assign_2 与 IVip 都存在对 discount 写操作,但共同的第一个祖先节点是 If(选择型),因此不会产生并发。同样的理由,Assign_2 与 IDiscount 也不会产生并发。Flow_2 包含在 Flow_1 中,所以不需对 Flow_2 中检测过的子孙活动节

点进行重复检测。从结果可以看出该算法能识别出条件分支中的 Flow 里活动之间潜在的数据竞争。

4 结语

作为一种描述服务组合的业务流程编程语言,BPEL 现已成为被业界广泛认可和接受的进行 Web 服务编排的事实标准。利用设计工具(GUI)来描述一个业务流程,一旦流程定义完毕,就会产生与 BPEL 标准相符的业务流模板,该文件可以被 BPEL 引擎执行。但是在执行前必须确保流程的正确

(下转第 1966 页)

出类似于复杂网络理论中的“富者越富,穷者越穷”的现象,本文提出利用视频本身的语义相似性构建视频语义相似度网络,实验结果表明在不影响单个虚拟新闻表现效果的前提下,增加了每个视频被选择使用的机会,丰富了表现的视频画面,总体上提高了整个虚拟新闻系统的表现效果,从而提高了虚拟信息环境的逼真度。

表3 某回合实验后数据

视频 ID	使用次数	相似的 ID	所占比例/%
1761	3	—	3.19
1802	3	1761	3.19
1837	3	1761	3.19
1849	3	1761	3.19
1945	2	1761	2.13
2467	3	—	3.19
1627	3	2467	3.19
1388	3	2467	3.19
2766	3	—	3.19
1661	3	2766	3.19
1202	2	2766	2.13
2707	3	—	3.19
1401	3	2707	3.19

本文以不同对象之间内在的相似性为突破点,研究其相似度,增加相似性对象被选中的机会,对于其他信息检索领域类似的需求也具有一定的参考意义。

参考文献:

- [1] 董献洲, 胡晓峰, 吴琳. 虚拟新闻的表达与生成及其系统设计与实现[J]. 系统仿真学报, 2006, 18(12): 3634-3636.
- [2] 陈芳莉, 胡晓峰, 吴琳, 等. 虚拟新闻模拟系统的研究与设计[J]. 计算机仿真, 2007, 24(8): 5-7.

- [3] 周生, 胡晓峰, 罗批. 战略对抗演习中的态势表现方法研究[J]. 装备指挥技术学院学报, 2009, 20(3): 96-99.
- [4] 司光亚, 胡晓峰, 吴琳. “沉浸式”战略决策训练模拟系统研究与实现[J]. 系统仿真学报, 2006, 18(12): 3581-3583.
- [5] 周生, 胡晓峰, 罗批. 虚拟新闻系统的自动化设计研究[J]. 计算机工程与应用, 2008, 44(12): 20-23.
- [6] 汪小帆, 李翔, 陈关荣. 复杂网络理论及应用[M]. 北京: 清华大学出版社, 2006.
- [7] BARABASI A L, BONABEAU E. Scale-free networks[J]. Scientific American, 2003(5): 50-59.
- [8] 杨士强, 孙立峰, 崔鹏, 等. 视频的语义挖掘[J]. 中国计算机学会通讯, 2009, 5(7): 30-35.
- [9] HACID M S, DECLEIR C. A database approach for modeling and querying video data[C]// Proceedings of the 15th International Conference on Data Engineering. Washington, DC: IEEE Computer Society, 2000: 729-750.
- [10] AYGUN R S, YAZICI A. Modeling and management of fuzzy information in multimedia database applications[J]. Multimedia Tools and Applications, 2004, 24(1): 29-56.
- [11] WANG Y, XING C X, ZHOU L Z. THVDM: A data model for video management in digital library[C]// Proceedings of the 6th International Conference on Asian Digital Libraries. New York: ACM, 2003: 178-192.
- [12] 余卫宇, 余英林. 视频语义信息的研究[J]. 计算机工程与应用, 2004, 40(6): 27-29.
- [13] 王煜, 周立柱, 邢春晓. 视频语义模型及评价准则[J]. 计算机学报, 2007, 30(3): 331-351.
- [14] 杨易, 郭同强, 庄越挺, 等. 基于综合推理的多媒体语义挖掘和跨媒体检索[J]. 计算机辅助设计与图形学学报, 2009, 21(9): 1307-1314.

(上接第1961页)

性,本文围绕 BPEL 数据竞争问题,在对 BPEL 2.0 中消息和活动之间关系进行分析的基础上,结合 XML 节点树特性,提出一种根据活动之间并序关系来检测潜在数据竞争的方法,为流程顺利部署奠定基础。

参考文献:

- [1] 何积丰, 金芝, 李宣东. 面向服务的计算专利前言[J]. 软件学报, 2007, 18(12): 2965-2966.
- [2] 喻坚, 韩燕波. 面向服务的计算[M]. 北京: 清华大学出版社, 2006.
- [3] RAO J, SU X. A survey of automated Web service composition methods[C]// Proceedings of Semantic Web Services and Web Process Composition. Berlin: Springer-Verlag, 2004: 43-54.
- [4] SADIQ S, ORLOWSKA M, SADIQ W, *et al.* Data flow and validation in workflow modelling[C]// Proceedings of the 15th Australasian Database Conference. Darlinghurst, Australia: Australian Computer Society, 2004: 207-214.
- [5] CHUN OUYANG, VERBEEK E, van der AALST M P, *et al.* Formal semantics and analysis of control flow in WS-BPEL[J]. Science of Computer Programming, 2007, 67(2/3): 162-198.
- [6] MARTENS A, MOSER S, GERHARDT A, *et al.* Analyzing compatibility of BPEL processes[C]// International Conference on Internet and Web Applications and Services/Advanced International Conference on Institute of Electrical and Electronics Engineers Computer

- Society. Washington, DC: IEEE Computer Society, 2006: 147-155.
- [7] 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.
- [8] WOMBACHER A, FANKHAUSER P, MAHLEKO B, *et al.* Matchmaking for business processes based on choreographies[C]// Proceedings of the 2004 IEEE International Conference on E-Technology, E-Commerce and E-Service. Washington, DC: IEEE Computer Society, 2004: 359-368.
- [9] ZHENG YONGYAN, ZHOU JIONG, KRAUSE P. Analysis of BPEL data dependencies[C]// Proceedings of the 33rd EUROMICRO Conference on Software Engineering and Advanced Applications. Washington, DC: IEEE Computer Society, 2007: 351-358.
- [10] KHALAF R, KOPP O, LEYMAN F. Maintaining data dependencies across BPEL process fragments[J]. International Journal of Cooperative Information Systems, 2008, 17(3): 259-282.
- [11] 陈胜, 鲍亮, 陈平, 等. BPEL 流程数据竞争和死锁检测算法研究[J]. 西安电子科技大学学报: 自然科学版, 2008, 35(6): 1058-1062.
- [12] OASIS. Web Service Business Process Execution Language(WS-BPEL)[EB/OL]. [2009-12-1]. <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>
- [13] 翟岩龙, 宿红毅, 张晗, 等. 基于数据流优化的 BPEL 流程分割方法[J]. 华南理工大学学报, 2009, 37(4): 24-30.