

一种基于随机 PETRI 网的 OWL-S 量化描述模型

胡新 王映康

(重庆市电力信息通信分公司 重庆 400044)

摘要 OWL-S 是一种面向领域本体的语义 Web 服务组合描述语言,也是一种可自动机器解释的语义标记语言,同时实现了 Web 服务功能和服务间互操作、互调用、互协作的语义化描述,是目前最重要的服务组合标准之一。提出了一种基于非马尔科夫随机 PETRI 网(NMSPN)的概率模型,对 OWL-S 描述的 Web 服务的控制流进行了建模和描述,并在 NMSPN 流程模型基础上建立了一系列的概率分析方法,实现了对语义 Web 服务的量化可信性的分析。

关键词 PETRI 网,OWL-S,Web 服务组合,服务量化模型

中图分类号 TP301 **文献标识码** A

Quantitative Description OWL-S Model Based on the Random PETRI Nets

HU Xin WANG Ying-kang

(Chongqing Electric Power Corporation, Information and Communication Branch, Chongqing 400044, China)

Abstract OWL-S is a semantic Web service composition-oriented ontology description and semantic markup language that can be explained by the automatic machine, while it also achieved the interoperability of Web service functions between the services, it is one of the most important service portfolio standards. This paper presented a probabilistic model based on non-Markov random PETRI NETS (NMSPN), which can model and describe the control flow of Web services described in OWL-S, then, the paper established a series of probabilistic analysis methods to achieve a quantitative credibility to analyze the semantics of Web services-based process model based on NMSPN.

Keywords PETRI nets, OWL-S, Web service composition, Service quantitative description models

1 前言

网络服务(Web Service)是一种依靠重用已有服务组件来组合新服务应用的技术,其依赖于一系列基于 XML 的协议和规范(如 SOAP, WSDL 等),以实现 Web 应用的自动整合和重构为目标。Web 服务已成为最成功、最流行的面向服务的体系架构(SOA)的形式。多数情况下,单一的 Web 服务很难满足实际的用户需求,一个复杂的商业流程通常需要多个 Web 服务的组合才能实现预期的目标。

在众多 Web 服务组合标准中,面向领域本体的 OWL-S 描述语言^[1](前身为 DAML-S)因支持 Web 服务的语义化描述和自动组合,近年来受到了学术界和工业界的重视。OWL-S 是一种可自动机器解释的语义标记语言,同时实现了 Web 服务功能和服务间互操作、互调用、互协作的语义化描述。目前,OWL-S 服务组合的理论研究多侧重于其功能描述和形式化验证方面^[2-7],而非功能特性的评估和量化特性(性能、可靠性等)的分析稍显不足,学术界和工业界仍然缺乏有效的模型和方法对 OWL-S 描述的组语义服务的控制流进行量化抽象和建模。

因此,本文的主旨即是设计一种用于定量描述 OWL-S 组合服务的方法,该方法以非马尔科夫随机 PETRI 网

(NMSPN)^[8]为载体,包含 OWL-S 规范中各种原子进程和复合进程到 NMSPN 的转换规则。在基于 NMSPN 的流程模型基础上,可以建立一系列的概率分析方法对语义 Web 服务的量化可信性进行分析。

2 量化描述模型描述

模型转换规则包括 OWL-S 中各种原子进程和复合进程的 NMSPN 转换规则。通过这些规则,任意一个基于 OWL-S 描述的组语义服务都可以转换为等效的随机 PETRI 网模型。由于 OWL-S 语言规范描述的细节纷繁复杂,其中很大一部分是用于对 Web 服务的功能进行语义化描述,因此提出的转换规则只对 OWL-S 中涉及流程控制、服务互调用、基本活动执行的要素进行 PETRI 网建模,力图从控制流层面对 OWL-S 进行行为而非语义上的建模,最终从中提取出时间/概率标注的量化过程模型,以为 OWL-S 组合服务的性能和可靠性分析提供支撑。

2.1 原子进程的转换规则

OWL-S 中的进程的基本功能是描述服务和服务接收方之间的交互行为。原子进程是最基本和不可再细分的进程,不再包含其他进程,它的功能是描述一个基本的服务接收/发送一个(或多个)消息。一个组合进程是包含一个或多个原子

本文受重庆市电力公司科技项目资助。

胡新(1973—),男,硕士,高级工程师,主要研究方向为软件设计、服务质量管理;王映康(1970—),男,硕士,高级工程师,主要研究方向为电力行业信息系统设计与研发。

进程的复杂进程。所有原子进程都通过<PERFORM>结构调用。一个原子进程对应了基本服务在单次交互中能完成的一次基本操作,它可拥有一个或多个输入(由<HASINPUT>结构指定),也可拥有一个或多个输出(由<HASOUTPUT>结构指定)。一个原子活动可拥有无限个前提条件,只有当所有前提条件都被满足时,原子进程才能被成功地调用和执行。输出的状态由原子进程执行时内部的状态和状态变化所决定。原子活动的主要操作是通过 GROUNDING 机制(一个 GROUNDING 描述了抽象的原子进程和一个由 WSDL 文档指定的实际服务组件的映射关系)联系和调用一个实际的服务组件。

根据上述分析,OWL-S 中的一个原子进程可转换为图 1 所示的 NMSPN 模型。在图 1 中,ini 库所用于描述第 i 个输入条件(由<HASINPUT>结构指定)的状态,outi 库所用于描述第 i 个输出条件(由<HASOUTPUT>结构指定)的状态。值得注意的是,<HASINPUT>和<HASOUTPUT>结构的数量可以为 0。从 ini 库所到 ivk 变迁的弧表示,只有当所有的输入条件都被满足时原子活动的操作才能真正被激活。started 和 completed 库所用于表示原子进程的初始和完成状态。时间变迁 soap_d 表示原子进程通过 SOAP 消息与被调用的外部服务组件建立成功链接所需要的时间,变迁 ivk_d 表示原子进程所调用的外部服务完成既定任务所需的时间,时间变迁 timer 表示用于超时意外控制的时间阈值。实际网络情况下的 SOAP 链接为非稳定链接,可能存在丢失和延迟过长的意外,我们用一个立即变迁 soap_f 来表示这种意外的发生,一旦消息丢失,failed 库所(灰色)立即被标注,以表示失败的调用。如果 SOAP 消息不丢失但 SOAP 消息延迟超过了设定的时间阈值,则对外部服务的调用也被终止,且 failed 库所被标注。反之,若 timeout 变迁的触发被抑制且对外部服务的调用被允许,在此情况下,被调用的外部服务自身可能执行出错(通过触发 ivk_f 变迁)或成功(通过标注 finished 库所)。

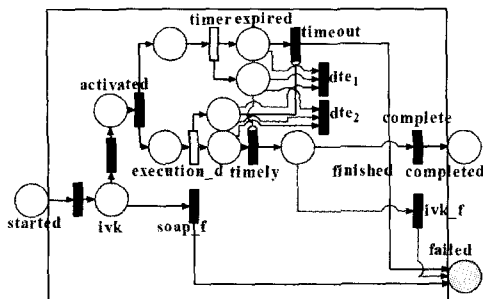


图 1 原子进程的 NMSPN 转换规则

2.2 复合进程的转换规则

所有的复合进程归根结底都由原子进程组成。OWL-S 语言中的<COMPOSEDOF>结构定义了复合进程中的控制流和数据流,并限定了其中各子进程之间的组合结构和各子进程的执行条件。

首先给出顺序复合进程,<SEQUENCE>的转换规则如图 2 所示。顺序进程包含一到多个子进程,其按照固定的顺序先后执行。图 2 中的顺序进程包含两个子进程,其分别用 P1 和 P2 表示。若 P1 和 P2 中任意一个执行出错,则其对应的错误库所 failed1 或 failed2 被标注,且整个顺序进程的 failed

库所立即被标注,以表示整个进程的执行错误。

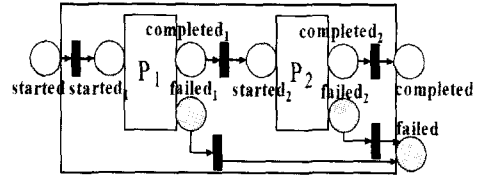


图 2 顺序进程的 NMSPN 转换规则

选择进程<CHOICE>要求在所包含的子进程(由<COMPONENTS>结构指定)中选择其中一个执行,其转换规则由图 3 给出。图 3 中的选择进程包含两个子进程,其分别用 P1 和 P2 表示,用异或方式相连。两个子进程中的一个在运行时随机地被选取执行。一旦被选中的子进程成功执行完成,则整个选择进程也成功执行完成(通过标注 completed 库所)。若选中执行的子进程执行出错,则整个选择进程的 failed 库所被标注。

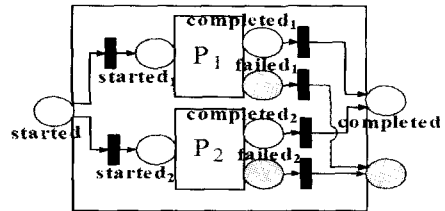


图 3 选择进程的 NMSPN 转换规则

分支进程<SPLIT>的转换规则如图 4 所示,它要求所包含的所有子进程同时被触发,并行地执行。只要子进程都被触发执行,则整个分支进程就成功结束,不等待各个子进程的执行结束。分支-合并进程<SPLIT-JOIN>的转换规则如图 5 所示,该进程也要求所包含的子进程同时被触发执行,但是只有当所有子进程都成功执行结束后,整个进程的执行才成功结束(即子进程同步)。

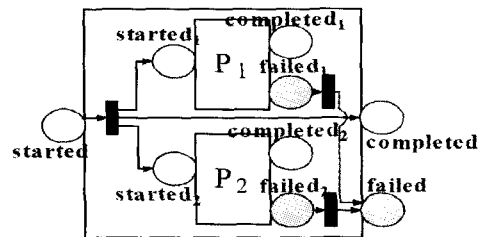


图 4 分支进程的 NMSPN 转换规则

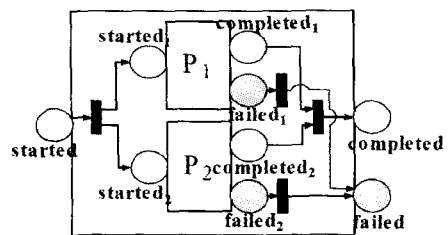


图 5 分支-合并进程的 NMSPN 转换规则

如果判断进程<IF-THEN-ELSE>的转换规则如图 6 所示,该进程为子进程关联了布尔判断条件。如果条件成真,则真分支对应的子进程被执行,反之假分支对应的子进程被执行。当选中执行的子进程成功完成执行后,整个进程成功结束。图 6 中的立即变迁 true 和 false 分别代表不二条件赋值

为真和假的情况。

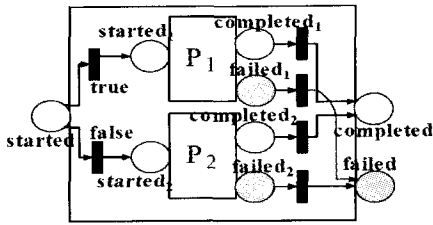


图6 如果判断进程的 NMSPN 转换规则

乱序进程<ANYORDER>的转换规则如图7所示,它要求所包含的子进程按照随机的先后顺序(非并行)执行。当所有的子进程的执行都完成后,整个进程的执行完成。库所 single 和从 single 发出的双向弧确保两个子进程 P1 和 P2 不会被并行地执行。

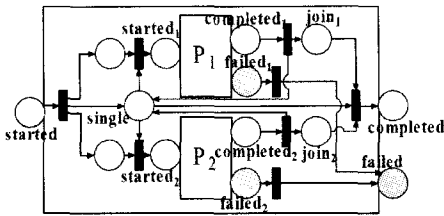


图7 乱序进程的 NMSPN 转换规则

仅当循环<REPEATWHILE>和直到循环<REPEATUNTIL>进程都是支持循环执行的组合进程,它们的转换规则如图8、图9所示。两种循环进程都带有一个布尔条件和一个子进程。在仅当循环中,先测试布尔条件,若成真则进入循环,反之则执行子进程。直到循环进程先执行子进程,然后测试布尔条件,若条件为真则退出,反之进入循环。因此,仅当循环有可能不执行所包含的子进程,而直到循环至少要执行一次。在图8、图9中,立即变迁 back 的作用是将控制流重新引导回初始以便开始新一轮循环,而立即变迁 skip 的作用是将控制流引出,结束循环执行。

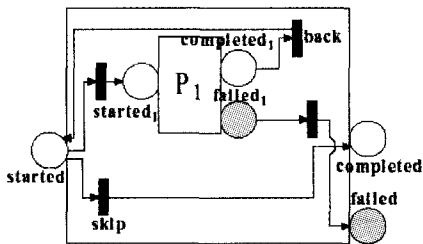


图8 仅当循环的 NMSPN 转换规则

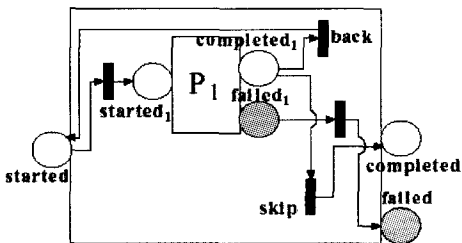


图9 直到循环的 NMSPN 转换规则

3 实例分析

实例分析中采用 OWL-S 组合服务样例 CongoProcess^[9]进行分析,并采用前面引入的转换规则对此样例进行了 PE-

TRI 网转换,转换出的 PETRI 网如图10所示。

在图10中,那些没有被转换规则描述的 OWL-S 要素,如输入关联(input-binding)、输出关联(output-binding)等,则没有显示。

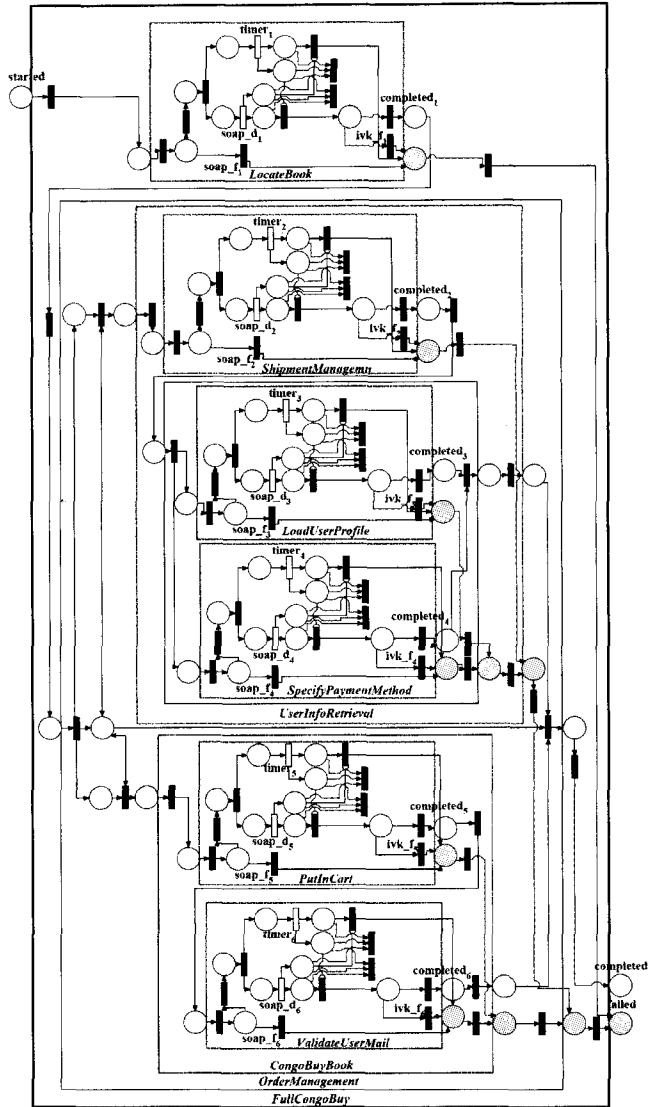


图10 样例转换出的 NMSPN

进程 FullCongoBuy 是处于最外层的顶端进程,它以乱序进程的形式封装了一个原子进程 LocateBook 和一个组合进程 OrderManagement。OrderManagement 进程包含两个组合进程,分别为 CongoBuyBook 和 UserInfoRetrieval。UserInfoRetrieval 进程以顺序进程的方式封装了两个原子进程,分别为 LoadUserProfile 和 ValidateUserMail。CongoBuyBook 进程也是以顺序进程的方式封装了一个组合进程 BuySequence。BuySequence 进程以顺序进程的方式封装了一个原子进程 PutInCart 和一个组合进程 SignInAndSpecify。SignInAndSpecify 进程以分支-合并进程的方式封装了两个原子进程,分别是 SpecifyPaymentMethod 和 ShipmentManagement。

结束语 本文提出了一种采用非马尔科夫随机 PETRI 网(NMSPN)建模 OWL-S 组合服务的方法。设计了一系列的转换规则,可将 OWL-S 的各种原子进程和组合进程转换为等效的 PETRI 网描述。在基于 NMSPN 的流程模型基础上,可以建立一系列的概率分析方法,对语义 Web 服务的量

参考文献

- [1] OWL-S Coalition; OWL-S: Semantic Markup for Web Service [EB/OL]. <http://www.ai.sri.com/daml/services/owl-s/1.2/overview/>, 2006
- [2] 朱俊, 郭长国, 吴泉源. 一种基于广义随机 Petri 网的 Web 服务组合性能预测模型[J]. 计算机科学, 2011(8)
- [3] Brogi A, Corfini S, Iardella S. From OWL-S Descriptions to Petri Nets[C]//ICSOC Workshops, 2007: 427-438
- [4] Norton B, Foster S, Hughes A. A Compositional Operational Semantics for OWL-S[J]. EPEW/WS-FM, 2005(3670): 303-317

- [5] 周敏, 张为群, 林已杰, 等. 一种基于扩展 Owl-S 本体的 Web 服务质量度量及评价方法的研究[J]. 计算机科学, 2010(5)
- [6] Mitra P, Wiederhold G. An Ontology-Composition Algebra[M]. Handbook on Ontologies, 2004: 93-116
- [7] Moldt D, Ortmann J, DaGen. A Tool for Automatic Translation from DAML-S to High-Level Petri Nets [J]. FASE, 2004 (2984): 209-213
- [8] German R. Performance analysis of communication systems: Modeling with non-Markovian stochastic Petri nets[M]//John Wiley and Sons, Chichester, 2000
- [9] <http://www.daml.org/services/owl-s/1.1/CongoProcess.owl>

(上接第 440 页)

(2) 本文算法与两种经典算法的空间复杂度相当。

(3) 本文算法的最坏、平均和最好时间复杂度区别最明显, Kruskal 算法次之, Prim 算法几乎不存在三者之分。根本原因是 Prim 算法几乎不受大循环内分支结构影响, 而本文算法和 Kruskal 算法受大循环内分支结构影响明显。

(4) Prim 算法的时间复杂度与边无关, 适合于稠密图。本文算法和 Kruskal 算法的时间复杂度都与边息息相关, 适合于稀疏图。

(5) 当 $n-1 < e < 2n-2$ ($e-n+1 < n-1$) 时, 邻接表存储结构下的性能优于 Kruskal 算法和 Prim 算法, 这正如一袋含有较少砂的米, 要将米沙分离, 拈沙总比拈米快。

(6) 本文算法性能受悬挂边的影响非常大, 但能显著改善算法性能, 但对 Kruskal 算法和 Prim 算法的性能改善几乎没影响。

(7) 输出结果明显优于此两种经典算法, 本文算法的输出结果是 MST, 本质上是图, 在其他函数或类中使用时可以调用图的一切算法; 此两种经典算法的输出结果是 MST 的边集合, 本质上是数组, MST 的所有边和顶点依然没被联系在一个数据结构中, 在其他函数或类中使用不便。

综上所述, 若图是稠密图 ($e \rightarrow n(n-1)/2$), 则建议采用 Prim 算法并用邻接矩阵存储结构; 若图是稀疏图 ($2n-2 \leq e \ll n(n-1)/2$), 则建议采用 Kruskal 算法并用邻接表存储结构; 若图是稀疏图 ($n-1 < e < 2n-2$), 则建议采用本文算法并用邻接表存储结构。这样的选择既可节省存储空间, 又能提高算法性能。另外, 在悬挂边天然较多和希望输出结果比较形象生动的情况下, 优先选用本文算法。

除上述两个经典算法外, Sollin 算法也是构造 MST 的常用算法, 其也是利用贪心算法设计策略, 最终同样是选出 MST 的边集合, 时间复杂度为 $O(n^2 \log n)^{[3, 20, 23]}$; 文献[15]指出管梅谷教授提出了破圈法来构造 MST 的思想, 实现情况未知。本文提出的算法利用了贪心算法设计策略, 同时也吸取了动态规划法和回溯法两种算法设计策略的精髓。有关贪心法、动态规划法和回溯法等设计策略, 详见文献[7, 10-13]。

结束语 本文阐明了一种全新 MST 构造算法的完整科学体系——思想、设计、实现、实验、分析、证明, 填补了 MST 基础理论和应用领域的一项空白。

参考文献

- [1] 耿素云, 屈婉玲. 离散数学(修订版)[M]. 北京: 高等教育出版社, 2004

- [2] 尹宝林, 何自强, 许光汉, 等. 离散数学(第三版)[M]. 北京: 高等教育出版社, 2011
- [3] Horowitz E, Sahni S, Anderson-Freed S. 数据结构基础(第二版)[M]. 朱仲涛, 译. 北京: 清华大学出版社, 2009
- [4] 严蔚敏, 吴伟民. 数据结构(C语言版)[M]. 北京: 清华大学出版社, 1997
- [5] 朱成立. 数据结构——使用 C 语言(第四版)[M]. 北京: 电子工业出版社, 2009
- [6] 李春葆, 尹为民, 李蓉蓉, 等. 数据结构教程(第二版)[M]. 北京: 清华大学出版社, 2007
- [7] 廖明宏, 郭福顺, 张岩, 等. 数据结构与算法(第四版)[M]. 北京: 高等教育出版社, 2007
- [8] 陈媛, 何波, 卢玲, 等. 算法与数据结构(第二版)[M]. 北京: 清华大学出版社, 2011
- [9] 张铭, 王蛟龙, 赵海燕. 数据结构与算法[M]. 北京: 高等教育出版社, 2008
- [10] 陈平, 褚华. 软件设计师教程(第二版)[M]. 北京: 清华大学出版社, 2006
- [11] 张铭, 赵海燕, 王腾蛟, 等. 数据结构与算法实验教程[M]. 北京: 高等教育出版社, 2011
- [12] 王晓冬. 计算机算法设计与分析(第三版)[M]. 北京: 电子工业出版社, 2007
- [13] Alsuwaiyel M H. 算法设计技巧与分析[M]. 吴伟昶, 方世昌, 等, 译. 北京: 电子工业出版社, 2010
- [14] 张宏, 温永宁, 刘爱利, 等. 地理信息系统算法基础[M]. 北京: 科学出版社, 2006
- [15] 马良, 朱刚, 宁爱兵. 蚁群优化算法[M]. 北京: 科学出版社, 2008
- [16] 李明哲, 金俊, 石瑞银. 图论及其算法[M]. 北京: 机械工业出版社, 2010
- [17] 滕国文. 数据结构课程设计[M]. 北京: 清华大学出版社, 2010
- [18] 陈国龙, 郭文忠, 涂雪珠, 等. 求解多目标最小生成树问题的改进算法[J]. 软件学报, 2006, 3(17): 364-370
- [19] 宁爱兵, 熊小华, 马良. 最小生成树灵敏度分析算法研究[J]. 小型微型计算机系统, 2011, 4(32): 744-746
- [20] 孙凌宇, 冷明, 谭云兰, 等. 赋权有向图的最小生成树算法[J]. 计算机工程, 2010, 2(36): 61-66
- [21] 王立冬. 约束最小生成树算法研究[D]. 西安: 西安电子科技大学, 2009
- [22] 王海涛. 内点带权的最小生成树[D]. 上海: 复旦大学, 2006
- [23] 陈晔君. 基于最小生成树的上海板块股票网络分析[D]. 北京: 北京工业大学, 2010
- [24] Grimaldi R P. 离散数学与组合数学(第五版)[M]. 林永钢, 译. 北京: 清华大学出版社, 2007
- [25] West D B. 图论导引(第二版)[M]. 李建中, 骆吉洲, 译. 北京: 机械工业出版社, 2005