

基于 MDA 的语义 Web 服务的组合与验证

朱正东¹, 蓝荣贵¹, 陈彦萍², 李增智¹

(1 西安交通大学 计算机系, 陕西 西安 710049; 2. 西安邮电学院 计算系, 陕西 西安 710061)

摘 要: 针对语义组合 Web 服务的验证问题, 研究了模型验证相关技术、统一模型语言(UML), 提出了基于模型驱动架构(MDA)的组合方法. 该方法使用 UML 类图和用例图对 OWL-S 进行静态组合建模, 使用活动图对 OWL-S 进行动态组合建模, 实现了语义组合 Web 服务的 UML 描述, 然后将该描述转换为 Promela 语言代码, 在 Promela 代码之后增加 LTL 的声明, 使用 SPIN 工具进行正确性、安全性和活性验证. 该模型保证了组合过程的正确性.

关键词: UML; 语义 Web 服务; 组合验证模型; MDA

中图分类号: TP311

文献标识码: A

文章编号: 1000-7180(2010)11-0004-06

Semantic Composition Web Services Verification Based on MDA

ZHU Zheng-dong¹, LAN Rong-gui¹, CHEN Yan-ping², LI Zeng-zhi¹

(1 Department of Computer, Xi'an Jiaotong University, Xi'an 710049, China;

2 Department of Computer Science, Xi'an Institute of Post and Telecommunications, Xi'an 710061, China)

Abstract: In order to solve the problem of composition and verifying of semantic Web services, this paper proposes a composite method based on model-driven architecture (MDA) and a matching method based on semantic matching degree. In the static aspects, the OWL-S is modeled by the composite method using UML class diagrams and use-case diagram. The dynamic composition of OWL-S is modeled by the use of activity diagram. In the procedure of modeling, a matching method based on semantic matching degree has been used to choose a subset of Web services for determining the most appropriate composite Web service and the compositing UML model can be translated into the Promela language to be verified by the model checking tools, SPIN. The verified UML model, as a template, is stored in ontology knowledge base. The model has been developed to improve the efficiency of the semantic Web services, and ensure the process of dynamic composition accuracy. By the use of templates and semantic matching degree, the most suitable and available Web services can be found and selected real-time.

Key words: UML; concurrent Web service; composite and reuse model; MDA

1 引言

为了提高计算机系统的正确性和可靠性, 包括形式化技术在内的各种检测手段应运而生^[1]. 形式化技术是为说明和验证复杂计算机系统所采用的基于数学的语言和工具的技术, 它包括形式化规范(specification)和形式化验证. 形式化验证的方法有两种: 定理证明和模型检验. 模型检验是基于模型的完全自动的验证方法, 较定理证明更适用于语义组

合 Web 服务的验证. 该方法的一般过程可分为建模、规范和验证, 它是用来检验由模型描述语言描述的系统模型是否满足由性质说明语言描述的系统性质. 具体地说, 是采用一种特定的形式化模型^[2], 如用 Petri 网或有限状态机表示系统的状态转移结构等来描述系统模型, 用时序逻辑, 如计算树逻辑(CTL)、线性时序逻辑(LTL)等来描述系统的性质, 再生成该系统的形式化规范, 最后使用验证工具可以通过遍历有限状态机来检验系统是否具有时序逻

辑所表达的性质等。

目前,基于线性时序逻辑 LTL 的模型检验技术使用比较广泛^[3]。LTL 语言是一种重要的描述和检验并发系统特性的形式化逻辑语言,可用于表示线性时序关系,它能够方便准确地描述并发系统的重要性质,如安全性和活性。

软件界普遍使用模型驱动架构(MDA)来分析、设计和实现软件系统,然而使用 MDA 来建模语义 Web 服务及其组合还是一个新的研究热点。传统的构建 Web 服务描述的模型驱动方法大致可分为两类:一种是构建基于 WSDL 或 BPEL 的语法描述;另一种是构建基于 OWL 或 OWL-S 的语义描述。前者能够处理复杂的过程但不支持语义,后者支持语义但不能处理含有复杂控制结构的组合过程。

OMG(Object Management Group)定义了统一建模语言 UML^[4]。在利用 UML 建模 Web 服务组合方面,Grønmo 等^[5]给出了 UML 与 OWL-S 的转换规则,使用活动图对语义 Web 服务组合进行了建

模;Skogan 等^[6]将 Web 服务的 WSDL 描述转换成 UML 模型,再对用 UML 表示的 Web 服务进行组合,生成基于 XML 语法的 BPEL4WS 和 WorkSCo。然而,这些方法不是不支持语义,就是没有对复杂过程的所有控制结构进行规范,而且都没有对 Web 服务组合过程进行有效的验证。

本文提出了一种有效的使用 UML 描述和组合语义 Web 服务的方法,引入验证机制保证语义 Web 服务组合的正确性。

2 语义 Web 服务的 UML 框架

在模型化语义 Web 服务之前,定义了 UML 框架,用三个扩充机制来扩大 UML 模型化语言的能力:构造型,标记值和约束。

图1是语义 Web 服务的 UML 框架元模型^[5]。UML 本体框架(UOP)^[7]用属性和关系捕获本体概念,该 UOP 具有两个元素:本体和本体类。一个本体类元素延伸了一个 UML 类,并且表示一个语义概念。

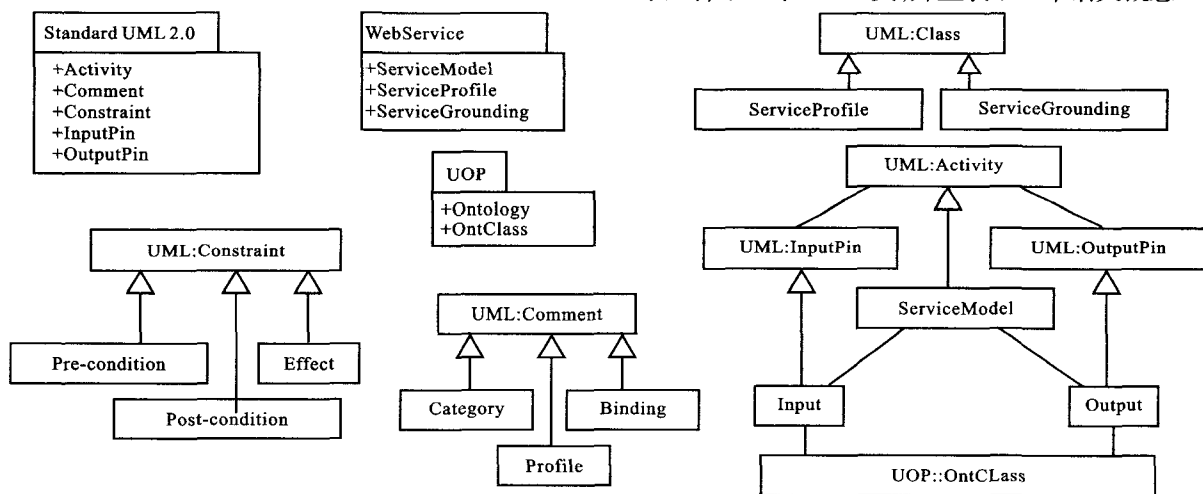


图1 语义 Web 服务的 UML profile 元模型

在框架中最重要的概念是 Web 服务包含的三个元素: ServiceModel, ServiceProfile 和 ServiceGrounding。Web 服务的每个元素与 OWL-S 中有相同名称的元素相对应。ServiceProfile 和 ServiceGrounding 延伸了 UML 类图,而 ServiceModel 延伸了 UML 活动图。

另外,ServiceModel 的输入输出元素是标准 UML2.0 InputPin 和 OutputPin 参数的延伸,前提条件、后置条件、效果是标准 UML2.0 限制的延伸,绑定、分类和框架是标准 UML2.0 注释的延伸。

3 语义 Web 服务的 UML 模型

Grønmo 等^[5]已充分证明 UML 在表达性、独立性和可读性三方面都能满足语义 Web 服务组合建模的需要。

UML 模型包括静态模型和动态模型。静态模型主要是对本体和原子服务的描述,而动态模型主要是对服务动态组合的描述。在 UML 模型中,Web 服务被描述成一个操作或行为,每个操作或行为都有标识名和输入输出。UML 动态模型对操作之间的控制流和数据流进行严格定义。

3.1 静态模型

静态模型使用类图对本体和原子服务进行描述. 首先, 用 UML 类图对本体的层次关系进行建模. 本体的概念可以泛化成许多的领域本体, 领域本体又可以泛化成许多实际的概念. 然后, 用 UML 类图对 Web 服务进行建模.

一个 Web 服务可能由一个或多个原子服务组成, 每个原子服务都能且只能完成一个操作功能. 将一个原子 Web 服务与类图中的一个操作相对应, 同时 Web 服务的输入消息与类图的输入相对应, Web 服务的输出消息也与类图的输出相对应. 图 2 是一个 Web 服务 FindCheaperBook 的类图.

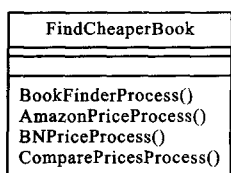


图 2 FindCheaperBook 的类图

另外, 静态模型使用用例图对人与 Web 服务的

交互进行建模. 将一个原子服务与一个用例相对应. 一个 Web 服务中可能有一个或多个原子 Web 服务要与包括用户在内的参与者进行交互. 使用用例图对此进行建模, 能够更加清晰地表示出各个参与者与 Web 服务中的哪些原子服务进行交互.

3.2 动态模型

使用 UML 活动图对 Web 服务动态组合进行建模. 活动图是一种表述过程基理、业务过程以及工作流的技术. 在很多方面, UML 活动图所起的作用与流程图相类似, 但是, 与流程图的主要区别是活动图支持并行行为. 由于 OWL-S 的动态组合结构是由 ServiceModel 中的 8 种基本结构构成的, 所以用活动图来对语义 Web 服务建模, 主要是确立如何用 UML 元素描述 ServiceModel 中的概念、属性及结构. OWL-S ServiceModel 包括的 8 种基本结构是: Sequence, Split-Join, Split, Choice, If-Then-Else, Repeat-While, Repeat-Until, Any-Order. 下面用活动图分别对这 8 种 OWL-S 的控制结构进行描述, 如图 3 所示.

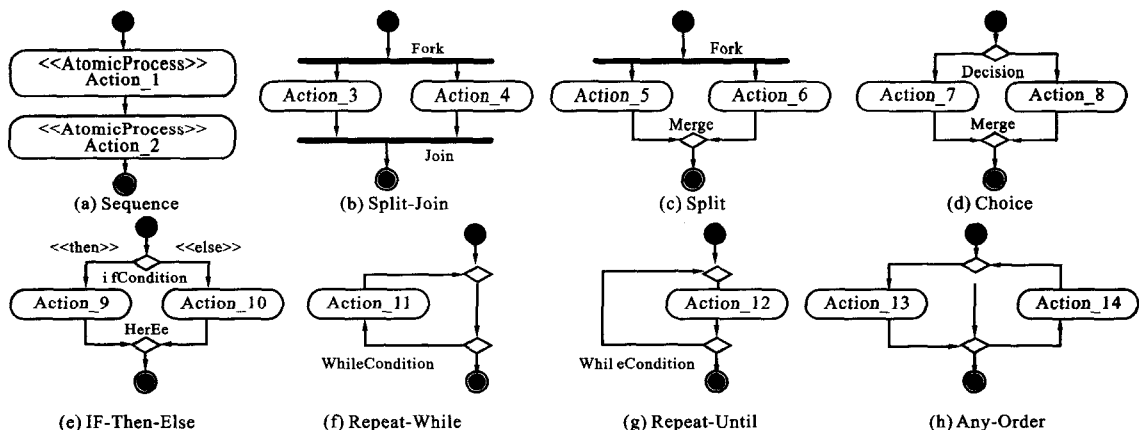


图 3 OWL-S 各种结构的 UML 模型

活动图中的一个动作与一个原子服务相对应. Sequence 结构表示几个动作前后顺序执行. Split-Join 则表示两个动作并行执行, 并且只有当两个动作都执行完后整个活动才结束. Split 表示两个动作选择执行, 其中有一个动作结束则活动结束. If-Then-Else 中有两个分支, 根据比较条件判断选择执行哪一个分支. Choice 结构与 If-Then-Else 类似, 只不过 Choice 结构中有 2 个或 2 个以上个分支, 根据条件选择其中的一个执行. 循环结构包括 Repeat-While 和 Repeat-Until 两种情况, Repeat-While 是如果满足比较条件则执行某动作, 而 Repeat-Until 是执行某动作直到不满足比较条件为止. Any-Order 有两个动作, 两个都要执行, 但两者不能同时执行,

如图 3(h) 所示, 图中三个分支能且仅能执行一次, 第二个选择节点第一次执行 Action_14, 第二次则选择结束.

为满足 OWL-S 中参数、参数绑定、条件、分类等信息, 需要在模型中引入标记值 (Tagged Value)、约束 (Constraint) 和饰针 (Pin). OWL-S 中的输入输出与模型中的饰针相对应, 参数绑定就是饰针与饰针的关系, 分类、描述信息等与模型中的标记值相对应, OWL-S 中的条件与模型中约束概念相对应.

详细的 OWL-S 与 UML 活动图的映射关系见表 1. 同时, 为了更好地说明如何用活动图对 Web 服务动态组合进行建模, 也给出了组合 Web 服务 FindCheaperBook 的动态组合模型, 如图 4 所示.

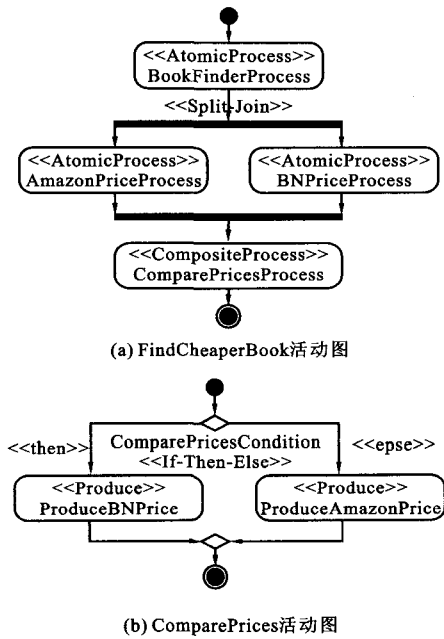


图 4 FindCheaperBook 的 UML 活动图

表 1 OWL-S 与 UML 活动图之间的映射关系

类型	OWL-S	UML2.0
控制结构	Sequence	Sequence
	Split	Fork + Merge
	Split-Join	Fork + Join
	Choice	Decision + Merge
	If-Then-Else	
	Repeat-While	Decision
	Repeat-Until	
参数	Any-Order	
	Input	Pin
	Output	
类别	parameterType	Property of Pin
	CategoryName	Tagged Value
	Taxonomy	
外形	ServiceName	
	TextDescription	
条件	Precondition	
	Result	Constraint
	Effect	
参数绑定	InputBinding	Pin to Pin
	OutputBinding	

4 组合 Web 服务的验证

为保证 Web 服务组合的正确性和可靠性,对 Web 服务组合进行验证是十分必要的.这是因为 Web 服务组合的建模是一个复杂且易出错的过程,如果组合服务在投入运行之后才被发现错误,则可能造成一些意想不到的后果,并且修复错误的代价会更高.

SPIN(Simple Promela Interpreter)^[8]是适合于并行系统辅助分析的检测工具,主要用于分布式系统的形式化验证. SPIN 以 Promela 为输入语言,通过分析其 Promela 程序来检测问题,跟踪其逻辑设计错误,检查逻辑一致性.

对语义 Web 服务组合使用 SPIN 进行验证,主要是使用 Promela 语言来描述 UML 动态组合模型,即使用 OWL-S 中 8 种基本结构相对应的活动图.所以本节我们主要阐述 UML 活动图所对应的 Promela 程序.

活动图中的 Sequence 结构在 Promela 中表示几个进程前后顺序执行. Split-Join 用 Promela 表示如下所示,其中 5 到 10 行实现 Join 功能. Split 的 Promela 表述和 Split-Join 差不多,只需跳过图中的 5 到 10 行.

```
(1) proctype Split-Join(chan synChan, dataChan) {
(2)   chan childSync = [2] of { int, mtype };
(3)   pid childA = run A(childSync);
(4)   pid childB = run B(childSync);
(5)   if
(6)     :: childSync?? eval(childA), done ->
(7)     if
(8)       :: childSync?? eval(childB), done;
(9)     fi
(10)  fi
(11)  synChan! -pid, done;
(12) }
```

活动图中的 Choice 结构的 Promela 表述如下所示,由 if 语句和守卫条件实现的,根据判断守卫条件选择一个分支进行执行,其他的分支自动跳过. If-Then-Else 结构是 Choice 结构的一个特例,其只有两个分支.

```
(1) if
(2) :: (a>b) -> option1
(3) :: (a=b) -> option2
(4) :: else -> skip
(5) fi
```

Repeat-While 结构的 Promela 表述如下所示, 当条件 c 满足时执行进程 p , 否则退出. Repeat-Until 结构的 Promela 表述与 Repeat-While 相近, 它的区别在于进程 p 一直执行直到条件 c 为真为止.

```
(1) do
(2) ::  $c \rightarrow p$ 
(3) :: else  $\rightarrow$  break
(4) od
```

活动图中的 Any-Order 结构是由 If-Then-Else 结构和 Repeat-Until 结构组成的, 所以 Any-Order 的 Promela 表述也可以通过 If-Then-Else 的 Promela 表述和 Repeat-Until 的 Promela 表述来实现.

首先在进程 A 和空进程间选择一个分支执行, 然后执行进程 B, 执行完进程 B 后执行进程 A 和空进程中没有被执行过的那一个进程, 最后退出.

FindCheaperBook 活动图的 Promela 表述如下:

```
(1) proctype FindCheaperBook ( chan syncChan, dataChan ) {
(2) chan childSync = [2] of { int, mtype };
(3) chan childData = [1] of { int };
(4) int book; int price;
(5) pid childA = run BookFinderProcess (
childSync );
(6) if
(7) :: childData? book  $\rightarrow$  childChan! book;
(8) :: childSync?? eval( childA ), done;
(9) fi
(10) pid childB = run AmazonPriceProcess (
childSync );
(11) pid childC = run BNpriceProcess( childSync );
(12) if
(13) :: childData? price  $\rightarrow$  childChan! price;
(14) :: childSync?? eval( childB ), done  $\rightarrow$ 
(15) if
(16) :: childData? price  $\rightarrow$  childChan!
price;
(17) :: childSync?? eval( childC ), done;
(18) fi
(19) fi
(20) pid child = run ComparePricesProcess (
childSync );
(21) syncChan! -pid, done;
(22) }
```

最后在 Promela 表述之后增加 LTL 声明, 用 SPIN 工具对其进行检验.

5 验证实例说明

使用上例的 Promela 模型进行验证, 指定 LTL

公式及其命题符号定义生成其 nerver 声明. 下面说明可达性和响应性验证.

(1) 可达性: $\langle \rangle$ BookFinderProcess

状态 BookFinderProcess 的可达性的 nerver 声明表示如下:

```
# define p BookfinderProces
nver { /* ! ( $\langle \rangle$  p) */
accept _init:
to _init:
if
:: (! ((p)))  $\rightarrow$  goto to _init
fi;
}
```

在该模型的可达性验证中, 状态向量 V 为 56byte, 搜索深度为 0, 消耗的时间很少, 也就是很快就能完成, 并且没有发现错误.

(2) 响应性: $[]$ (BookFinderProcess $\langle \rangle$ BNPriceProcess $\wedge$ AmazonPriceProcess)

该 LTL 公式表示找书处理完成之后, 就要进行 BNPriceProcess 和 AmazonPriceProcess 的价格处理. 命题符号可定义如下:

```
# define p BookFinderProcess
# define q BNPriceProcess
# define r AmazonPriceProcess
```

该 LTL 公式可表示为 $[] (p \rightarrow \langle \rangle q \wedge r)$, 其 nerver 声明形式与上例类似.

实验表明, FindCheaperBook 的 UML 模型中不存在死锁, 符合可达性、安全性等要求. 这一方面说明了提出的语义 Web 服务组合的建模方法是正确的, 另一方面也表明我们给出的基于 DMA 的语义 Web 服务组合检验机制也是准确可靠的.

将验证后的 UML 组合模型生成规范的 Web 组合服务描述, 如 OWL-S, BPEL4WS 等. 最后将产生的 Web 组合服务的描述与单个 Web 服务一样发布到 Web 服务注册中心.

6 结束语

在 Internet 的应用中, Web 服务实现了软件重用, Web 服务的组合是重用的升华. 本文用 UML 对用 OWL-S 描述的语义 Web 服务进行描述, 实现了基于 MDA 的语义 Web 服务的组合, 提高了开发语义 Web 服务的效率, 同时提高了系统的可用性及实用性; 通过将 UML 模型转化为可验证的 Promela 语言, 使用 SPIN 工具进行验证, 保证了动态组合过程的正确性, 在更高层次上抽象地描述语义 Web 服务

对 Web 服务的动态绑定也更为有利,为组合 Web 服务的使用奠定了基础;本模型用于语义 Web 服务组合、验证和使用,能最大程度地满足用户需求.作者将进一步完善目标本体,将参考已有成果^[9],进一步研究 Web 服务的验证方法、调度与执行算法,并进一步降低计算复杂度.

参考文献:

- [1] Clarke E M, Grumberg O, Peled D A. Model checking [M]. Cambridge, Massachusetts: MIT Press, 2001.
- [2] 张琳霞,李茵,李小丽. 基于 Petri 网和 Web 服务的工作流建模方案[J]. 微电子学与计算机, 2006, 23(3): 150 - 153.
- [3] 滕星权,李增智. 基于 OWL-S 的 Web 服务组合的验证研究[J]. 微电子学与计算机, 2007, 24(12): 62 - 66.
- [4] The Object Management Group. UML 2.0 Superstructure Specification, OMG Adopted Specification ptc/03 - 08 - 02[R]. 2003.

- [5] Grønmo R, Jaeger C, Hoff H. Transformations between UML and OWL - S [C]// Proceedings of Conference on Foundations and Applications. European, 2005: 269 - 283.
- [6] Skogan D, Grønmo R, Solheim I. Web service composition in UML[C]// Proceedings of 8th IEEE International Enterprise Distributed Object Computing Conference (E-DOC 2004). USA, California, 2004.
- [7] Dragan Djuric. MDA - based ontology infrastructure[J]. Computer Science Information Systems (ComSIS), 2004, 1(1): 91 - 116.
- [8] Spin - Formal Verification[EB/OL]. [2007 - 02 - 08]. <http://spinroot.com/spin/whatispin.html>.
- [9] 杨文军,李涓子,王克宏. 领域自适应的 Web 服务评价模型[J]. 计算机学报, 2005, 28(4): 514 - 523.

作者简介:

朱正东 男, (1963 -), 博士. 研究方向为计算机网络及其应用.

(上接第 3 页)

比度,解决了映射变换中灰度范围难以确定的问题,是一种有效的图像增强方法.

参考文献:

- [1] 张便利,常胜江,李江卫,等. 基于彩色直方图分析的智能视频监控系統[J]. 物理学报, 2006, 55(12): 6399 - 6404.
- [2] Ching - Chung Yang. Image enhancement by modified contrast - stretching manipulation [J]. Optics & Laser Technology, 2006, 38(3): 196 - 201.
- [3] 王震洲,刘教民,李穆,等. 低压电器开关电弧图像增强的直方图算法[J]. 电工技术学报, 2008, 23(6): 50 -

53.

- [4] 张懿,刘旭,李海峰. 自适应图像直方图均衡算法[J]. 浙江大学学报:工学版, 2007, 41(4): 630 - 633.
- [5] Subr K, Majumder A, Irani S. Greedy algorithm for local contrast enhancement of images[C]// Proceedings of International Conference on Image Analysis and Processing. Cagliari, 2005: 171 - 179.
- [6] 张长江,汪晓东,张浩然,等. 图像对比度增强的非线性变换法[J]. 光电工程, 2006, 33(6): 65 - 67.

作者简介:

李仁和 男, (1963 -), 研究员. 研究方向为电子产品装备.