

Web服务语义描述方法进展评述

王 欣

(中国科学院国家科学图书馆 北京 100190)

(中国科学院研究生院 北京 100049)

(中国人民公安大学图书馆 北京 100038)

【摘要】回顾 Web服务语义描述的发展过程,详细介绍自 OWL - S和 WSDL - S之后 Web服务语义描述标准的发展状况,对各种语义描述机制包括 SAWSDL、W SMO - Lite以及 RESTful Web服务的描述方法 SA - REST、hREST、MicroW SMO的特点及应用状况做出研究和评述,并在此基础上分析语义技术在 Web上的应用发展方向。

【关键词】Web服务描述 语义 Web服务 OWL - S WSDL - S SAWSDL SA - REST hREST

【分类号】TP393

A Review of Web Service Semantics Description Evolution

Wang Xin

(National Science Library, Chinese Academy of Sciences, Beijing 100190, China)

(Graduate School of Chinese Academy of Sciences, Beijing 100049, China)

(Library of Chinese People's Public Security University, Beijing 100038, China)

【Abstract】This paper makes a retrospect for the evolution of semantic description of Web service, especially the evolution after OWL - S and WSDL - S. It also reviews the features and application status of all kinds of semantic description mechanism of Web service such as SAWSDL and W SMO - Lite, the description mechanisms of RESTful Web service as well as SA - REST and hREST, MicroW SMO. Finally, the author gives a future vision on development of semantic technology for semantic Web service.

【Keywords】Web service description Semantic Web service OWL - S WSDL - S SAWSDL SA - REST hREST

1 引言

在 Web服务描述中加入语义信息可以帮助更好地完成服务的自动发现、自动选择、自动组合以及服务间数据的自动映射和转换等功能,进一步帮助提高验证、配置、洽谈、合同签订等功能的自动化。语义网的研究者们最早提出了 Web服务的语义描述模型,包括 W SMO、OWL - S等。从 2001年 OWL - S^[1]的前身 DAML - S提出到现在,语义 Web服务描述走过了一个从复杂繁琐到简洁实用的演化过程,本文试图跟踪研究近几年 Web服务语义描述的发展变化,并介绍和评述几种主流的 Web服务语义描述机制、规范及其应用状况。

2 Web服务语义描述模型 OWL - S

OWL - S不是从扩展已经广泛应用的 WS - 协议簇角度提出,而是从语义技术的高度阐述如何描述 Web服务,而后使用 Grounding本体与已有标准兼容。OWL - S由三个子本体组成,包括 ServiceProfile、ProcessMode和

收稿日期: 2009 - 12 - 22

收修改稿日期: 2010 - 01 - 11

Grounding,其中 Service Profile用来表达服务是做什么的; Process Model说明服务如何工作;而 Grounding则把 Process Model映射到具体的底层协议或消息规范中去,例如 WSDL^[2]。

Service Profile主要用于服务的自动发现,通过对服务的描述以及服务需求方对需求的描述,计算机能够计算出该服务是否符合用户需求。Service Profile主要描述了服务的三个方面:

(1)功能,包括输入、输出、条件和结果;

(2)描述的服务的类目信息,即服务在一个业务领域目录的位置;

(3)描述了服务的非功能特性,例如安全、隐私和 QoS等。

Process主要定义了与 Web服务交互可能的模式,主要存在两种 Process:原子过程(Atom Process)和组合过程(Composition Process)。原子过程没有内部结构,只有一个简单的输入和与之相对应的输出,而组合过程包含一系列由控制流和数据组合起来的过程组件。每个过程同样具有用于描述过程的输入、输出、过程和结果属性。

Grounding主要用于定义在服务的消费者和服务的提供者之间抽象的信息交换如何转换为实体的信息交换。例如把 OWL - S映射到 WSDL中将会把 OWL - S中的每一个子过程的输入和输出都映射到 WSDL的输入和输出消息中。这种映射使得语义的输入输出信息与具体的无语义的服务输入输出描述信息(如 WSDL)能够相互转换,从而使得 OWL - S能够与现有规范兼容。

3 自下而上的 WSDL - S和 SAWSDL

3.1 WSDL - S

OWL - S为 Web服务的语义描述提供了一个完美的模型,从服务的发现、协调、洽谈到服务的组合过程和数据转换,涉及到 Web服务自动组合的整个过程,并提供了与原有标准结合使用的方法。但是 OWL - S在互联网上的应用状况正如其他语义技术一样因为其复杂繁琐而难以得到推广。

BM研究院、乔治亚大学 LSD IS实验室联合提出了 WSDL - S。他们提出了语义 Web服务描述的需求有以下 5条^[3]:基于现存的 Web服务标准;机制应独

立于语义标识语言;允许使用多种语言表示方法结合;支持 XML - Schema的语义注解;在 Web服务的模式类型和本体之间提供丰富的映射机制。

基于以上的需求,研究者们提出了 WSDL - S方法作为为 Web服务增加语义注解的机制。这个模型对语义标注语言未作规定,使用者可以使用任何一种语义标注语言来标注语义信息;WSDL - S兼容 WSDL,开发者可以使用已经很熟悉的 WSDL来描述非语义信息;使用增量工具开发者很容易从 WSDL升级到 WSDL - S。笔者认为 OWL - S在逻辑描述方面的基础以及 WSMO在 F - Logic方面的基础使得它们成为表达丰富的语义描述语言,但是对于扩展 WSDL这样的工业标准,增加语义信息是一种更为实际的方法。尽管 OWL - S也考虑到了与 WSDL的兼容,但是 WSDL的开发者认为 OWL - S是一种自上而下的标准,而 WSDL - S是一种自下而上的标准,更容易推广与普及,更符合现实的应用情况。

WSDL主要扩展了以下元素:

(1)一个扩展元素:ModeReference,允许 WSDL输入和输出类型模式与语义模型的概念一对一关联;

(2)一个扩展属性:SchemaMapping,允许 WSDL输入和输出类型模式元素与语义模型多对多关联;

(3)两个新元素:Precondition and Effect,作为 Operation元素的子元素,按照 OWL - S的方法描述语义,用于服务发现;

(4)一个关于接口元素的扩展属性:ServiceCategorization,包括服务分类信息,用于服务发布注册,与 OWL - S的 Categorization对应。

相比 OWL - S,WSDL - S的开发者们更强调语义描述框架的简洁、实用、开放性以及与现有规范的兼容性。

3.2 SAWSDL

2007年,W3C提出 SAWSDL为 Web服务语义注解的候选推荐标准^[4],SAWSDL来源于 WSDL - S,但是比 WSDL - S更为开放,应用性更强。SAWSDL的目标仅仅定位为增强 WSDL,这大大简化了为 Web服务增加语义信息的工作,有利于推动语义 Web服务的发展。SAWSDL仅仅定义了两个元素 ModeReference和 SchemaMapping(包括 LiftingSchemaMapping和 LoweringSchemaMapping),最大化地使用了 XML技术,尽可

能降低技术门槛。

SAWSDL 并不是一门语义注解的语言,而是为 WSDL 和 XML Schema 提供了一种增加语义注解的机制,使得 WSDL 和 XML Schema 中的组件可以与外部的语义概念模型联系起来。在 WSDL 中使用语义注解本身并不会给 WSDL 文档增加任何语义信息,而是给出了一个语义参考模型的链接地址。当服务之间需要进行语义交互时,可以通过中间的标准本体来进行数据的转换、服务的发现和选择。ModelReference 可以为任何一种 WSDL 的组件做注解,它直接指向一个概念模型的地址,而 LiftingSchema 用于把 XML 结构的数据映射到语义模型,LoweringSchema 把语义模型的数据映射到 XML 结构。下面的代码对名为“OrderResponse”的元素进行了语义注解,ModelReference 指向其语义模型,LiftingSchemaMapping 指向其与高级语义模型映射的方法。

```
<xs:element name="OrderResponse" type="confirmation" />
<xs:simpleType name="confirmation"
sawSDL:modelReference="http://www.w3.org/2002/ws/sawSDL/spec/ontology/purchaseorder#OrderConfirmation"
sawSDL:liftingSchemaMapping="http://www.w3.org/2002/ws/sawSDL/spec/mapping/Response2Ont.xslt">
...
</xs:simpleType>
```

3.3 WSMO - Lite

SAWSDL 把语义标注的工作交给了其他本体,而 SAWSDL 仅仅增加了一个通往相关本体的链接地址,人们并没有预先定义一个关于 WSDL 元素的本体结构,如何使用语义技术描述 Operation、Interface、Fault 这些 WSDL 中的元素成为需要解决的问题。WSMO 在 2008 年提出了 WSMO-Lite^[5],这是一种基于 WSMO 的轻量级的 Web 服务描述本体,借助 SAWSDL 机制,为 Web 服务语义描述提供了一套最小集。草案根据不同的技术发展水平提出了语义 Web 的协议栈,协议栈共有 5 层,而 SAWSDL 是无语义层与语义层衔接的关键部分,它通过简单的扩展允许 WSDL 组件可以被语义标识和注解,提供了语义层的基础。

WSMO - Lite 本体由 4 个子本体组成:Ontology、ClassificationRoot(Category)、NonFunctionalParameter、Condition and Effect(Capability)。Ontology 定义了服务信息模型的数据集;Classification 用于描述服务在类别目录

中的信息,从而描述服务的功能;NonFunctionalParameter 描述服务的非功能信息;而 Condition and Effect 用来描述服务可以做什么。从图 1^[5]中可以看出,WSDL 元素使用 ModelReference 分别指向 WSMO - Lite 不同的子本体,Schema 元素或者 Type 指向了 Ontology 本体,其中包含描述数据的本体和 XML 模式元素与高层本体的映射关系;Operation 指向了用于描述服务功能和能力的 ClassificationRoot 以及 Condition and Effect;而 Endpoint、Binding 元素的注释指向了 Non - Functional 本体。

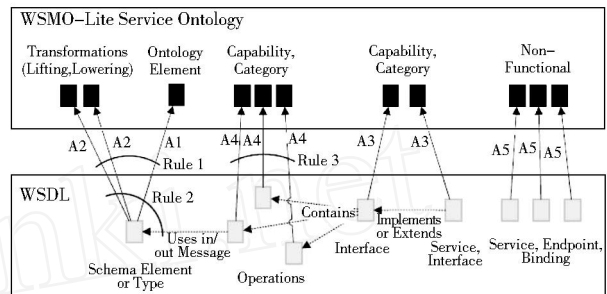


图 1 WSDL 注解和规则

WSMO - Lite 还提出了使用 WSMO - Lite 进行语义标注的三条规则:

(1)一致性规则,如果 XML 模式信息同时有 A1 和 A2 注释,那么 A1 指向的语义模型必须能够通过 A2 定义的转换关系与 XML 模式信息相互转换;

(2)完整性规则,WSDL 元素的每个输入信息都必须有一致的 A1 和 A2 (Lowering) 注释,每个输出信息都必须有一致的 A1 和 A2 (Lifting) 注释;

(3)完整性规则,一个接口的所有操作必须使用功能描述性注释,这条规则保证了接口内的所有操作都不被自动客户端忽视。

至此,WSDL 的语义描述有了一个较为完整的语义标注机制和描述模型,通过 SAWSDL 这样一种中间机制,使得有语义和无语义两种服务能够共同工作。

3.4 OWL - S 的自上而下和 SAWSDL 的自下而上

对比 OWL - S 和 SAWSDL,二者均出自为 Web 服务增加语义信息的目的,但是方法途径却是从两个方向而来。OWL - S 首先构造了 Web 服务描述的本体,然后通过 Grounding 与现存 Web 服务描述标准自上而下进行衔接;而 SAWSDL 并没有构建一个描述 Web 服务的标准本体,而是增加语义注解机制,通过语义注解机制使得 WSDL 元素及 XML Schema 与高层的语义模

型相关。

文献 [6] 对比了这两种方法的相同点和差异。笔者认为二者的不同主要在于出发点不同, SAWSDL 仅仅关注对 WSDL 的语义标注,并不试图定义一套对 Web 服务的语义描述标准,而是通过注解机制为 Web 服务的语义描述构造了一个开放的空间和结构,把选择何种语义表达方式和语义描述语言统统留给了用户和服务的开发者。用户完全可以根据自己对自动化的需求选择合适的语义模型,而不必被复杂的语义模型所牵绊。但是局限性也显而易见, SAWSDL 仅仅能够对 WSDL 中已有的元素增加语义注解,对 Web 服务的语义描述并不全面。而 OWL - S 则不同,其制定者来自语义研究机构,其出发点是构建 Web 服务标准完善的语义描述模型,完全从语义的角度来描述 Web 服务。由于目前语义网络的技术并没有得到广泛的应用,从现有标准向上选择恰当的语义模型,一步步为现有技术标准增加语义信息的做法肯定是优于使用繁琐复杂语义的描述模型向下兼容现有标准的做法。正如有学者指出:在后 SAWSDL 时代,与其在各种语义 Web 服务描述方法中争出一个胜负,不如去思考如何利用各种优秀的方法去增强 SAWSDL [7]。

4 RESTful Web 服务描述方法

4.1 RESTful Web Service

2000 年, Fielding 在他的博士论文中提出适用于分布式超文本系统的 REST (REpresentational State Transfer) 架构 [8]。这种简洁而轻量级的软件架构风格非常适合互联网上的信息存取服务, Web 上大量的服务属于信息存取类服务,复杂的交互非常少,这时使用 HTTP 协议中原有的 Get, Post, Delete, Put 4 个命令就完全可以胜任全部工作,而不必使用重量级的 SOAP 协议。这一架构风格很快在互联网上行风起来,原有大多数提供基于 SOAP 协议的 Web 服务系统都已经开始提供 REST 服务,如 Amazon, Google。由于 RESTful Web 服务并非标准的 Web 服务,也无法使用 WSDL 来描述,通常人们只是使用人类可读的 HTML 文本来描述服务信息,手工来构建服务的获取方法。

4.2 RESTful Web Service 描述方法

随着 Web2.0 观念的深入人心,人们不再满足于仅仅从 Web 上被动获取信息,而且要创造信息;不仅

仅被动地使用固定模式的信息服务,而且要使用各种信息服务构建新的信息模式。Mashups 就是这样一种使用网络上开放的接口获取信息,构建新的信息应用的方法。目前关于 Mashups 还没有统一的定义,文献 [9] 认为 Mashups 是一种基于 RESTful Web 服务的组合,与通常意义上的服务组合相比, Mashups 更为轻量级。Mashups 是一种简单模式的服务组合,完全具备向大众推广的潜力,使得不具备编程技巧的普通 Web 用户也能够方便地使用,这也正是 Mashups 拥护者们的初衷。但是目前的情况却不是这样,一些研究者认为主要原因在于各种 RESTful Web 服务之间的数据格式、结构不一致,导致用户在制作 Mashups 时必须使用编程语言进行大量的数据中转 (Data Mediation) 工作 [9]。如果数据的中转能够实现完全的自动化,那么则可以大大简化用户制作 Mashups 的工作流程,笔者称这种 Mashups 为 SMashups (Semantic Mashups)。文献 [9] 提出了 SA - REST 方法,为 RESTful Web 服务增加语义信息。SA - REST 方法借鉴了 SAWSDL 的标注方法,为服务输入输出数据提供映射模型,通过 Lifting Schema 把输出的数据映射到与数据相关的本体模型中,通过 Lowering Schema 把输入的数据从本体模型转换为 XML 数据。这样两个服务的数据中转通过独立的本体进行转换和计算,而服务仅仅需要提供服务输入输出数据与本体的映射关系。图 2 显示了服务之间数据中转的流程。

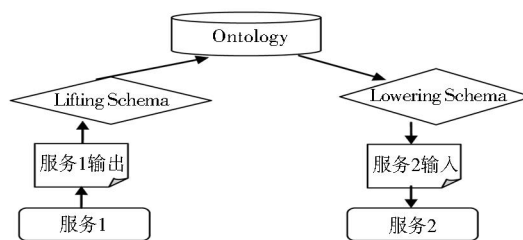


图 2 服务之间数据中转流程

使用 SOAP 协议的 Web 服务语义标注可以直接在 WSDL 文件中进行,而 RESTful Web 服务通常只使用 HTML 文档描述服务的获取方式。微格式能够方便地加入对服务的语义注解,而且不增加专门的描述文档。它是一种轻量级的子格式,通常用于嵌入到一些宿主语言中,如 HTML、XHTML 或 Atom。利用微格式,可以在描述 Web 服务的 XHTML 文档中嵌入语义注解,从而使得为人类编写的文档也能够被计算机识别 [10]。

SA - REST提出使用 RFDa嵌入语义注解,使用 GRDDL提取嵌入的语义信息。2008年 SA - REST的部分研究人员又提出 hRESTS——一种专用于描述 RESTful Web服务的微格式标准,基于 hRESTS之上的 SA - REST要比使用 RFDa更为简单^[11]。

2008年,WSMO组织针对 RESTful Web服务发布了 MicroWSMO^[12],与 WSMO - Lite类似, MicroWSMO也由 Ontology, Schema Mapping, Capability and Category, 以及 No - Functional Property 4个子本体组成。一个 RESTful Web服务通常包含若干信息资源,每个资源的输入输出数据使用具体的与数据集相关的本体以及本体与服务数据之间的映射关系注解;对于整个 Web服务使用 Capability和 Category来描述功能语义,用于服务的发现;同时对每个信息资源的操作语义也要使用 Capability和 Category描述;最后服务的非功能特性由 No - Functional Property描述。与 WSMO - Lite相同, MicroWSMO并不提供对每个操作的非功能语义注解。

5 结 语

语义网代表着互联网发展的方向,但是从目前的互联网向语义网络转变是一个漫长的过程。对语义技术的应用,即知识表达语言的深度广度是由用户对自动化的需求决定的。SAWSDL和 hREST, SA - REST这样开放式的语义注解方式把语义注解的选择权留给了用户,允许使用者根据实际需求一步步增加语义信息,是更为符合实际应用的方法。正如语义 Web社区中一句著名的格言所说“一点点语义足够走很长的路(A Little Semantics Goes a Long Way)^[13]!”。这句话最早是印在1997年 SHOE(Semantic HTML Ontology Extensions)^[14]项目的宣传 T恤衫上面,用以表达在无语义的环境下,加入一点语义可以极大地促进 Web系统的发展。而后在语义 Web业界又被多次提起^[15],用以提醒人们:在当前的应用状况下,并不是语义越多、表达越丰富就越好,而是要与使用需求和技术发展状况相匹配。

参考文献:

[1] Martin D, Paolucci M, McIlraith S, et al Bringing Semantics to Web Service: The OWL - S Approach [C]. In: *Proceedings of 1st*

International Workshop on Semantic Web Services and Web Process Composition, San Diego, CA. 2004: 26 - 42.

- [2] Martin D, Burstein M, McDermott D, et al Bringing Semantics to Web Services with OWL - S [J]. *World Wide Web*, 2007, 10 (3) : 243 - 277.
- [3] Akkiraju R, Farrell J, Miller J A, et al **BM Research Report Web Service Semantics—WSDL - S** [R]. RC2854 (W0601 - 132), 2006.
- [4] W3C. Semantic Annotations for WSDL and XML Schema [EB / OL]. [2009 - 10 - 11]. <http://www.w3.org/TR/sawsdl/>.
- [5] Vitvar T, Kopecky J, Fensel D. D11V0. 2 WSMO - Lite: Lightweight Semantic Description for Service on the Web [EB / OL]. [2009 - 12 - 22]. http://wsmo.org/TR/d38/v0.1/20080219/d38v01_20080219.pdf
- [6] Paolucci M, Wagner M, Martin D. Grounding OWL - S in SAWSDL [C]. In: *Proceedings of the 5th International Conference on Service - Oriented Computing*, Vienna, Austria Berlin, Heidelberg: Springer - Verlag, 2007: 416 - 421.
- [7] Martin D, Domingue J. Semantic Web Service, Part 2 [J]. *IEEE Intelligent Systems*, 2007, 22 (6) : 8 - 15.
- [8] Fielding R T. Architectural Styles and the Design of Network - based Software Architectures [D]. University of California Irvine, 2000.
- [9] Lathem J, Gomadam K, Sheth A P. SA - REST and (S) Mashups: Adding Semantics to RESTful Services [C]. In: *Proceedings of the International Conference on Semantic Computing*. Washington, DC, USA: IEEE Computer Society, 2007: 469 - 476.
- [10] 季书宁. 基于微格式的信息组织与处理框架 [J]. *图书情报工作*, 2007, 51 (8) : 35 - 37.
- [11] Kopecky J, Gomadam K, Vitvar T. hRESTS: An HTML Microformat for Describing RESTful Web Services [C]. In: *Proceedings of WIAI*. 2008 (1) : 619 - 625.
- [12] Kopecky J, Vitvar T, Fensel D. D38V0. 1 MicroWSMO: Semantic Description of RESTful Services [EB / OL]. [2009 - 12 - 22]. http://wsmo.org/TR/d38/v0.1/20080219/d38v01_20080219.pdf
- [13] Hendler J. A Little Semantics Goes a Long Way [EB / OL]. [2009 - 12 - 11]. <http://www.cs.pitt.edu/~hendler/LittleSemanticsWeb.html>
- [14] SHOE [EB / OL]. [2009 - 12 - 20]. <http://www.cs.umd.edu/projects/plus/SHOE/index.html>
- [15] Hendler J. The Dark Side of the Semantic Web [J]. *IEEE Intelligent Systems*, 2007, 22 (1) : 2 - 4.

(作者 E - mail: wangxin@mail.las.ac.cn)