基于扩展接口语义的 Web 服务匹配模型研究*)

于守健　夏小玲　乐嘉锦　黄晓虎

(东华大学计算机科学与技术学院　上海 200051)

摘　要　分析了 Web 服务组合的类型以及其中的接口匹配关系，通过对 WSDL 标准接口描述语言进行语义扩展，提出了一种轻量级的 Web 服务语义描述模型。该模型从服务接口的消息参数层、消息层和操作层分别扩展语义描述，将服务接口的描述从语法层提升到语义层，从而能够全方位地描述 Web 服务的功能、行为约束等属性。在此基础上，从这三个层次计算 Web 服务接口的语义匹配度，提出了多层次 Web 服务接口语义匹配模型。

关键词　Web 服务，匹配模型，服务描述语言，语义，WSDL

Web Service Matching Model Based on Extended Interface with Semantic

YU Shou-Jian　XIA Xiao-Ling　LE Jia-Jin　HUANG Xiao-Hu

(College of Computer Science and Technology, Donghua University, Shanghai 200051)

Abstract　This paper first analyzes the types of Web service composition and the interface matching relationship in it. By extending standard Web service interface description language—WSDL(Web Services Description Language) with semantic, a kind of light-weight semantic Web service description model is proposed. This model extends WSDL from three levels: message parameter, message and operation. Thus the description of Web service interface is enhanced from syntax level to semantic level, which can fulfill comprehensive service description including service capability and behavior constraint. Based on the semantic description model, by computing the semantic similarity between service interfaces from these three levels, the multi-level Web service matching model is proposed.

Keywords　Web service, Matching model, Service description language, Semantic, WSDL

1　前言

Web 服务使得 Web 向着一个基于 Internet 的计算平台发展。然而，现在的 Web 服务标准(UDDI, WSDL, SOAP 等)的描述能力有限[1]。这些协议由于缺乏语义描述能力，服务的提供者和服务的请求者之间没有共同的语义约定，因此不能满足客户提出的基于服务功能描述寻找服务的需求，即不能根据功能的匹配而寻找定位最佳服务[2,3]。WSDL 对接口的描述只限于语法层次[4]，不能表达语义信息。本文通过定义服务发布者和请求者共同遵守的本体(ontology)，从服务接口的消息参数层、消息层、操作层分别扩展语义描述，提出了 Web 服务的语义描述模型，从而将服务接口的描述，从语法层提升到语义层。在此基础上，通过从消息参数层、消息层、操作层计算服务接口的匹配分数，提出了一种综合多层次的服务接口匹配模型。该模型能够动态地根据当前系统中基本服务的情况，去发现服务，从而组合出满足用户需求的复杂服务。

2　Web 服务组合的类型

Web 服务提供的功能是通过调用其中的操作来完成的。根据 WSDL 的描述，Web 服务支持四种操作模式：notification, one-way, solicit-response 和 request-response。我们将四种操作模式总结为 *In/Out* 和 *Out/In* 两种类型：*In/Out* 操作模式首先接收一个输入消息，处理后返回一个输出消息；*Out/In* 操作模式首先发送一个输出消息并等待接收一个输入消息作为结果。一个操作的执行通常经过四个状态：就绪(*ready*)、开始(*start*)、活动(*active*)、结束(*end*)。我们用"→"表示状态之间关系，$S_1 \rightarrow S_2$ 表示状态 S_1 在 S_2 之前发生。因此，一个操作的全部执行状态可以表示为 *ready→start→active→end*。

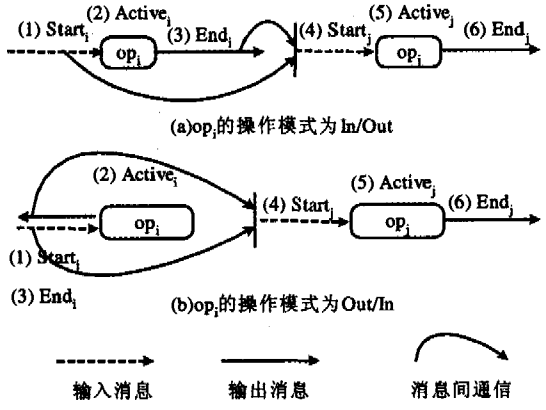

图 1　顺序组合

我们定义顺序组合和替换组合两种 Web 服务操作的组

*)基金项目：上海市科委产学研联盟项目(05DZ11C06)。于守健　讲师，博士，主要研究方向为 Web 服务、企业应用集成、数据库与数据仓库等；夏小玲　博士，副教授，主要研究方向为数据库与多媒体技术等；乐嘉锦　博士生导师，主要方向为软件工程、数据库与数据仓库等；黄晓虎　实验师，硕士，主要研究方向为构件与构件库。

合方式。顺序组合以与供应链结构相似的方式，进行操作的组合。假定 op_i 和 op_j 是两个顺序组合的操作，我们称 op_i 和 op_j 分别为源操作和目的操作，图 1(a)和(b)分别说明了当 op_i 的操作模式是 *In/Out* 和 *Out/In* 时的顺序组合情况。替换组合是指将操作 op_i 的功能委派给 op_j 执行。图 2(a)和(b)分别说明了当 op_i 的操作模式为 *In/Out* 和 *Out/In* 时的替换组合。以(a)为例，op_i 的操作模式为 *In/Out*，当 op_i 被调用时，它发送一个输入消息给 op_j，然后 op_j 代表 op_i 执行被请求的操作，并返回结果给 op_i，op_i 最后将结果返回给它的调用者，op_i 的输入消息与 op_j 的输入消息进行匹配，op_j 的输出消息与 op_i 的输出消息进行匹配。替换组合中操作之间状态的先后关系为：$Start_i \rightarrow Start_j \rightarrow Active_j \rightarrow End_j \rightarrow End_i$，$Start_i \rightarrow Active_i \rightarrow End_i$。

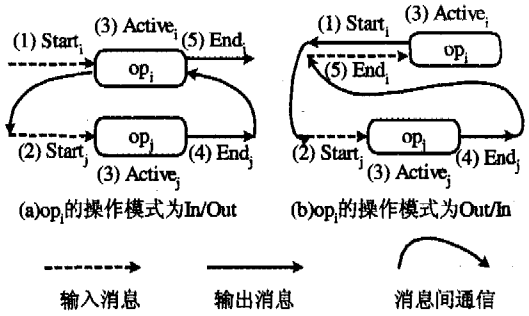

图 2　替换组合

3　Web 服务的语义描述模型

本文采用扩展 WSDL 的方法对 Web 服务进行语义描述。WSDL 分别从消息参数、消息、操作三个层面，对 Web 服务接口进行描述，因此我们也从这三个层面对服务描述进行语义扩展。

3.1　消息层与消息参数层语义描述

Web 服务的每个操作都关联了一个或两个消息，我们为每个消息关联一个消息类型(Message Type)，如定义 1 所述。

定义 1(消息)　一个消息 M 定义为一个三元组(*Name*, *P*, *Message Type*)：*Name* 表示消息名称；*P* 表示一系列参数名；*Message Type* 表示消息类型，也就是消息所表示的具体应用领域中的概念。

一个消息中有多个消息参数，为了给参数赋予语义，我们用领域本体中的一个概念(concept)描述参数的语义，如定义 2 所述。

定义 2(消息参数)　一个消息参数 P 定义为一个四元组(*Name*, *Data Type*, *Concept*, *Unit*)：*Name* 表示参数名称；*Data Type* 表示参数对应的 XML Schema 内置类型；*Concept* 表示参数对应的具体应用领域本体中的概念，*Unit* 表示参数的度量单位。

3.2　操作层语义描述

我们从功能属性和非功能属性两个方面描述 Web 服务的操作。功能属性描述了操作的语法和语义特征，操作的语法特征包括操作的模式和绑定两个方面。我们从描述(description)、目的(purpose)和类别(category)三个方面描述操作的静态语义。操作的描述是总结操作功能特征的一段文本描述。操作的类别指定了操作的应用领域，如定义 3 所述。操作的目的指定了操作的功能，它由三个属性定义，如定义 4 所述。

定义 3　一个操作的类别(Category)定义为四元组(*Domain*, *Synonyms*, *Specialization*, *Overlapping*)。

定义 4　一个操作的目的(purpose)定义为一个三元组(*Function*, *Synonyms*, *Specialization*)。

功能属性(*Function*)描述了操作提供的业务功能，别名属性(*Synonyms*)和特例属性(*Specialization*)与类别中的定义相同。

动态属性描述了与操作执行时相关的属性，包括操作之间的执行关系和操作本身的行为特性。操作间属性定义了操作的执行顺序，我们用操作前驱和操作后继，来表示操作之间的执行关系。以下是操作之间前驱关系和后继关系的形式化定义：

定义 5　op_i 和 op_j 是两个操作，如果有 $End(op_i) \rightarrow Ready(op_j)$，那么 op_i 是 op_j 的前驱。

定义 6　op_i 和 op_j 是两个操作，如果有 $End(op_j) \rightarrow Ready(op_i)$，那么 op_i 是 op_j 的后继。

操作的行为表示了操作在给定条件下的输出结果，它是由多个形式为 $R_{op}^{m} = \dfrac{(\Pr\ eParameter_{op}^{m}, \Pr\ eCondition_{op}^{m})}{(PostParameters_{op}^{m}, PostCondition_{op}^{m})}$ 的业务规则定义的。其中，$\Pr\ eParameter_{op}^{m}$ 和 $PostParameters_{op}^{m}$ 表示操作的输入输出参数集，$\Pr\ eCondition_{op}^{m}$ 和 $PostCondition_{op}^{m}$ 分别是 $\Pr\ eParameters_{op}^{m}$ 和 $PostParameters_{op}^{m}$ 上的断言。规则 R_{op}^{m} 表示：当条件 $\Pr\ eConditioin_{op}^{m}$ 满足时，操作 op_i 进入开始状态，操作进入结束状态时，条件 $PostCondition_{op}^{m}$ 满足。

服务是通过操作被调用的，总结以上各个方面，我们从语法、静态语义、动态语义以及服务质量等多个方面，描述 Web 服务的一个操作，下一节在此基础上讨论扩展了语义的服务接口匹配模型。

4　多层次 Web 服务接口语义匹配模型

根据 Web 服务接口的定义，我们分别从消息参数、消息、操作三个层次，结合前一节对 Web 服务接口多层次的语义描述，进行 Web 服务接口匹配。

4.1　消息参数层匹配

对于简单类型，消息参数对应一个 XML Schema 内置类型，每个参数还与领域本体中一个概念相关联，用于表示参数的语义。对于源参数 $P_1(DT_1, Concept_1, Unit_1)$ 和目的参数 $P_2(DT_2, Concept_2, Unit_2)$，如果 DT_1 与 DT_2 的类型相同，那么我们称它们是直接兼容的。如果一个类型是由另一个类型派生出的，那么称这两个类型是间接兼容。直接兼容和间接兼容的数据类型 DT_1 和 DT_2，我们都称之为精确参数类型匹配。每个消息参数都引用了领域本体中的一个概念[5]，用以表示参数的语义，定义 7 给出了消息参数的语义概念匹配计算方法。

定义 7　对于源参数 $P_1(DT_1, Concept_1, Unit_1)$ 和目的参数 $P_2(DT_2, Concept_2, Unit_2)$：(1)如果 $Concept_1 \equiv Concept_2$，或者 $Concept_1 \subset Concept_2$，我们称 P_1 和 P_2 为精确概念匹配；(2)如果 $Concept_2 \subset Concept_1$，我们称 P_1 和 P_2 为近似概念匹配。

设定参数精确匹配的匹配度 $C_P^{DT}=1$，否则 $C_P^{DT}=0$；精确概念匹配的匹配度 $C_P^{Concept}=1$，近似概念匹配的匹配度 $C_P^{Concept}=0.5$。式(1)表示了参数层总的匹配度的计算方法，其中 ω_P^{DT} 表示参数类型匹配的权重，$\omega_P^{Concept}$ 表示参数语义概念匹配的权重。其取值范围 $\omega_P^{DT} \in [0,1]$，$\omega_P^{Concept} \in [0,1]$，且 $\omega_P^{DT} +$

$\omega_P^{Concept}=1$。

$$C_P=\omega_P^{DT}\times C_P^{DT}+\omega_P^{Concept}\times C_P^{Concept} \quad (1)$$

4.2 消息层匹配

根据定义1对消息的定义，我们从消息参数和消息类型两方面进行消息层匹配。

一个消息中包含多个消息参数。对于两个操作 op_i 和 op_j，设 In_i 为操作 op_i 的输入消息，In_{ik} 为消息 In_i 中的一个消息参数；Out_i 为操作 op_i 的输出消息，Out_{ik} 为 Out_i 中的一个消息参数；In_j 为操作 op_j 的输入消息，In_{jl} 为消息 In_j 中的一个消息参数；Out_j 为操作 op_j 的输出消息，Out_{jl} 为消息 Out_j 中的一个消息参数。根据第2节中讨论的组合类型，我们得出以下的消息参数匹配方法：

(1)对于替换组合：当源操作的模式是"*In/Out*"时，In_i 应该与 In_j 连接，Out_j 与 Out_i 连接，如果 $\forall In_{jl}\in In_j$，$\exists In_{ik}\in In_i$，使得 In_{ik} 可以与 In_{jl} 相匹配，且 $\forall Out_{ik}\in Out_i$，$\exists Out_{jl}\in Out_j$，使得 Out_{jl} 可以与 Out_{ik} 相匹配，我们称 op_i 与 op_j 为精确消息参数匹配，匹配度设为1。如果操作不满足精确消息参数匹配，两个操作之间消息参数匹配度以式(2)计算：

$$C_M^P=\frac{\sum_{l=1}^{n}C_p(In_i,In_{jl})+\sum_{k=1}^{m}C_P(Out_j,Out_{ik})}{N+M} \quad (2)$$

其中：N 表示消息 In_j 中参数个数；$\sum_{l=1}^{n}C_p(In_i,In_{jl})$ 表示消息 In_j 中参数匹配度的总和；M 表示消息 Out_i 中参数个数；$\sum_{k=1}^{m}C_p(Out_j,Out_{ik})$ 表示消息 Out_i 中参数匹配度的总和。

(2)对于替换组合：当源操作的模式是"*Out/In*"时，精确消息参数匹配的计算与式(1)相似，匹配度设为1。如果操作不满足精确消息参数匹配，两个操作之间消息匹配度以式(3)计算：

$$C_M^P=\frac{\sum_{l=1}^{n}C_P(Out_i,In_{jl})+\sum_{k=1}^{m}C_P(Out_j,In_{ik})}{N+M} \quad (3)$$

(3)对于顺序组合：$In_i\cup Out_i$ 与 In_j 相连接。如果 $\forall ln_{jl}\in In_j$，$\exists P\in(ln_i\cup Out_i)$，使得 P 与 In_{jl} 相匹配，我们称 op_i 与 op_j 为精确消息参数匹配，匹配度设为1。如果操作不满足精确消息参数匹配，两个操作之间消息参数匹配度以式(4)计算：

$$C_M^P=\frac{\sum_{l=1}^{n}C_p(In_i\cup Out_i),In_{jl})}{N} \quad (4)$$

其中：N 表示消息 In_j 中参数个数；$\sum_{l=1}^{n}C_p((In_i\cup Out_i),In_{jl}$ 表示消息 In_j 中参数匹配度的总和。

根据定义2，我们为每个消息关联了一个消息类型(Message Type)属性来描述消息的语义。基于描述逻辑中的包含"⊏"和等价"≡"关系(统一用"⊆"表示)，我们给出消息类型匹配度计算方法。对于两个操作 op_i 和 op_j，MT_i^{In} 为操作 op_i 输入消息的类型，设 MT_i^{Out} 为操作 op_i 输出消息的类型；MT_i^{In} 为操作 op_j 输入消息的类型，MT_j^{Out} 为操作 op_j 输出消息的类型。

(1)对于替换组合：当源操作的模式是"*In/Out*"时，如果 $MT_i^{In}\subseteq MT_j^{In}$，且 $MT_j^{Out}\subseteq MT_i^{Out}$，我们称 op_i 与 op_j 是消息类型精确匹配；如果 $MT_i^{In}\cap MT_j^{In}\neq\Phi$ 且 $MT_j^{Out}\cap MT_i^{Out}\neq\Phi$，我们称 op_i 与 op_j 是消息类型近似匹配；

(2)对于替换组合：当源操作的模式是"*Out/In*"时，消息类型匹配方式与式(1)相似，在此不再赘述；

(3)对于顺序组合：如果 $(MT_i^{In}\cup MT_i^{Out})\subseteq MT_j^{In}$，我们称 op_i 与 op_j 是消息类型精确匹配；如果 $(MT_i^{In}\cup MT_i^{Out})\cap MT_j^{In}\neq\Phi$，我们称 op_i 与 op_j 是消息类型近似匹配。

式(2)、(3)、(4)定义了消息参数匹配度计算公式。对于消息类型匹配，我们设定消息类型精确匹配的匹配度 $C_M^{MT}=1$，消息类型近似匹配的匹配度 $C_M^{MT}=0$。式(5)表示了消息层总的匹配度的计算方法，其中 ω_M^P 表示消息参数匹配的权重，ω_M^{MT} 表示消息类型匹配的权重，其取值范围 $\omega_M^P\in[0,1]$，$\omega_M^{MT}\in[0,1]$，且 $\omega_M^P+\omega_M^{MT}=1$。

$$C_M=\omega_M^P\times C_M^P+\omega_M^{MT}\times C_M^{MT} \quad (5)$$

4.3 操作层匹配

第3.2节从语法层、静态语义和动态语义三个方面，对Web服务操作进行描述，因此本节从这三个方面进行Web服务操作的匹配。

对于语法层Web服务操作的匹配，我们从操作模式和绑定两个方面比较两个操作，如定义8和9所述。

定义8(操作模式匹配) 两个操作 op_i 和 op_j，M_i 表示 op_i 的操作模式，M_j 表示 op_j 的操作模式，当以下条件成立时，我们称 op_i 与 op_j 为操作模式精确匹配：

(1)$M_i=$"*notification*"且 $M_j=$"*one-way*"；或

(2)$M_i=$"*one-way*"且 $M_j=$"*notification*"；或

(3)$M_i=$"*solicit-response*" 且 $M_j=$"*request-response*"；或

(4)$M_i=$"*request-response*"且 $M_j=$"*solicit-response*"。

定义9 op_i 和 op_j 为两个操作，如果 op_i 支持的绑定协议 $Binding_i$，与 op_j 支持的绑定协议 $Binding_j$ 有公共部分，即 $Binding_i\cap Binding_j\neq\varnothing$，那么我们称 op_i 与 op_j 为绑定精确匹配。

设定操作模式精确匹配的匹配度 $C_{op}^{mode}=1$，否则 $C_{op}^{mode}=0$；绑定精确匹配的匹配度 $C_{op}^{binding}=1$，否则 $C_{op}^{bingding}=0$。以操作模式匹配度和绑定匹配度的平均值，作为操作语法层匹配度，如式(6)所示：

$$C_{op}^{Syntax}=\frac{C_{op}^{mode}+C_{op}^{bingding}}{2} \quad (6)$$

根据3.2节，我们从类别(*category*)和目的(*purpose*)两个方面进行服务操作静态语义匹配。如果操作 op_i 类别的领域与操作 op_j 类别的领域是相同的或是别名关系；并且对于顺序组合，两个操作有相互交叠的领域，即 $C_j\in C_i$. *Overlapping*；对于替换组合，op_j 类别的 *specialization* 能够提供 op_i 的类别的 *specialization*，那么我们称 op_i 与 op_j 为精确类别匹配。目的匹配与类别匹配完全相同。设定类别精确匹配的匹配度 $C_{op}^{category}=1$，否则 $C_{op}^{categor}=0$；目的精确匹配的匹配度 $C_{op}^{purpose}=1$，否则 $C_{op}^{purpose}=0$。以类别匹配度和目的匹配度的平均值，作为操作静态语义层匹配度，如式(7)所示。

$$C_{op}^{Static}=\frac{C_{op}^{category}+C_{op}^{purpose}}{2} \quad (7)$$

操作的行为特性反映了操作的动态语义，操作的动态语义匹配是通过比较源操作和目的操作的行为规则而进行的。对于行为规则 $R_i^n=(\text{PreParameter}_i^n,\text{PreCondition}_i^n,PostParameter_i^n,PostCondition_i^n)$，其中操作层的消息匹配，已经计算了消息参数 PreParameter_i^n 和 $PostParameter_i^n$ 的匹配度，所以操作的动态语义匹配，只考虑操作前置条件 PreCondition_i^n 和操作后置条件 $PostCondition_i^n$。对于两个操作 op_i 和 op_j 的行为规则 $R_i=(\text{Pre}C_i,PostC_i)$ 和 $R_j=(\text{Pre}C_j,PostC_j)$，根据替换组合与顺序组合的分类，和前置条件之间和后置条件

(下转第136页)

2001
8 Bertino E, Ferrari E, Bonatti P A. TRBAC: A Temporal Role-Based Access Control Model. In: Proceedings of 5th ACM Workshop on Role-Based Access Control. Berlin, Germany, 2000
9 Convington M, Long W, Stinivasan S, et al. Securing Context-aware Applications Using Environment Roles. In: ACM Symposium on Access Control Model an Technology. Chantilly, VA, 2001
10 Zhang L, Ahn G, Chu B. A Rule-Based Framework for Role-Based Delgation and Revocation. ACM Transactions on Information and System Security(TISSEC), 2003, 6(3)
11 Atluri B, Huang WK. An Authorization Model for Workflows. In: Proceedings of the 5th European Symposium on Research in Computer Security, Lecture Notes in Computer Science Vol. 1146, Springer-Verlag, 1996. 44～64
12 Coulouris G, Dollimore J, Roberts M. Role and Task-Based Access Control in the PerDiS Groupware pPlatform. In: Proceedings of the ACM Workshop on Role-Based Access Control. George Mason University, 1998. 115～121
13 Thomas R, Sandhu R. Task-Based Authorization Controls (TBAC): A Family of Models for active and enterprise-oriented authorization management. In: Proc. of the IFIP WB11. 3 Conference on Database Security, August 1997
14 Thomas R, Sandhu R. Task-Based Authorization Controls (TBAC): Models for Active and Enterprise-oriented Authorization Management. In: Database Security XI: Status and Prospects, T. Y. Lin and X. Qian, eds. North Holland, 1997
15 Neumann G, Strembeck M. Design and Implementation of a Flexible RBAC-Service in an Object-Oriented Scripting Language. In: Proc. of the 8th ACM Conference on Computer and Communications Security (CCS), 2001
16 Neumann G Strembeck M. An Approach to Engineer and Enforce Context Constraints in an RBAC Environment. In: Proc. of 8th ACM Symposium on Access Control Models and Technologies (SACMAT), 2003

(上接第 131 页)

之间的三种关系：等价关系(⇔)、蕴涵关系(⇒)、没有关系(null)，操作动态行为匹配度以如下方法计算：

(1)对于替换组合：当源操作的模式是 *In/Out* 时，如果 $\forall PreC_j^n \in R_j$，$\exists PreC_i^n \in R \mid (PreC_i^n \Leftrightarrow PreC_j^m)$ 且 $\forall PostC_i^n \in R_i$，$\exists PostC_j^m \in R_j \mid (PostC_j^m \Leftrightarrow PostC_i^n)$；或者 $\forall PreC_j^m \in R_j$，$\exists PreC_i^n \in R_i \mid (PreC_i^n \Rightarrow PreC_j^m)$，且 $\forall PostC_i^n \in R_i$，$\exists PostC_j^m \in R_j \mid (PostC_j^m \Rightarrow PostC_i^n)$，我们称 op_i 和 op_j 是行为精确匹配的，匹配度设为 1；否则动态行为匹配度的计算公式如下：

$$C_{op}^{Behaviour} = \frac{\sum_{i=1}^{n} C_{op}^{Behaviour}(PreC_{ik}, PreC_{jl}) + \sum_{k=1}^{m} C_{op}^{Behaviour}(PostC_{jl}, PostC_{ik})}{N+M} \tag{8}$$

其中：N 表示操作 op_j 中前置条件个数；$\sum_{j=1}^{n} C_{op}^{Behavior}(PreC_{ik}, PreC_{jl})$ 表示操作 op_j 中前置条件匹配度的总和；M 表示操作 op_i 中后置条件个数；$\sum_{k=1}^{m} C_{op}^{Behavior}(PostC_{jl}, PostC_{ik})$ 表示操作 op_i 中后置条件匹配度的总和。

(2)对于替换组合：当源操作的模式为 *Out/In* 时，精确匹配的计算与(1)相似，匹配度设为 1；否则动态行为匹配度的计算公式如下：

$$C_{op}^{Behavior} = \frac{\sum_{i=1}^{n} C_{op}^{Behavior}(PostC_{ik}, PreC_{jl}) + \sum_{k=1}^{m} C_{op}^{Behavior}(PostC_{il}, \Pr eC_{ik})}{N+M} \tag{9}$$

(3)对于顺序组合：操作模式是“*In/Out*”或者“*Out/In*”，如果 $\forall \Pr eC_j^n \in R_j$，$\exists r \in (\Pr eC_i \cup PostC_i) \mid (r \Leftrightarrow \Pr eC_j^n)$；或者 $\forall \Pr eC_j^n \in R_j$，$\exists r \in (\Pr eC_i \cup PostC_i) \mid (r \Rightarrow \Pr eC_j^n)$，我们称 op_i 和 op_j 是行为精确匹配的，匹配度设为 1；否则动态行为匹配度的计算公式如下：

$$C_{op}^{Behavior} = \frac{\sum_{i=1}^{n} C_{op}^{Behavior}((\Pr e_{ik} \cup Post_{ik}), \Pr eC_{jl})}{N} \tag{10}$$

式(6)说明了语法匹配度计算方法，式(7)说明了静态语义匹配度计算方法，式(8)、(9)、(10)说明了操作动态语义匹配度的计算方法。综合这三个方面匹配度的计算方法，操作层总的匹配度的计算方法如式(11)所示。其中，ω_{op}^{Syntax} 表示语法匹配的权重，ω_{op}^{Static} 表示静态语义匹配的权重，$\omega_{op}^{Behaviour}$ 表示操作动态语义匹配的权重，其取值范围 $\omega_{op}^{Syntax} \in [0,1]$，$\omega_{op}^{Static} \in [0,1]$，$\omega_{op}^{Behaviour} \in [0,1]$，且 $\omega_{op}^{Syntax} + \omega_{op}^{Static} + \omega_{op}^{Behaviour} = 1$。

$$C_{op} = \omega_{op}^{Syntax} \times C_{op}^{Syntax} + \omega_{op}^{Behaviour} \times C_{op}^{Behaviour} + W_{op}^{Behaviar} \times C_{op}^{Behaviar} \tag{11}$$

4.4 多层次服务匹配度计算

第 4.1～4.3 节分别从消息参数、消息、操作三个层次，进行 Web 服务接口的匹配。表 1 综合了各匹配的层次以及相关的权重，用户可根据具体应用自行定义这些权重。消息层匹配度的计算，依赖于消息参数层匹配度计算的结果，因此我们从消息层和操作层两方面，计算一个服务操作的匹配度，如式(12)所示。其中，C_M 为 4.2 节消息层匹配度计算结果，ω_M 为消息层匹配的权重；C_{op} 为 4.3 节操作层匹配度计算结果，ω_{op} 为操作层匹配的权重，$\omega_M + \omega_{op} = 1$。

$$C = \omega_M \times C_M + \omega_{op} \times C_{op} \tag{12}$$

多层次接口语义匹配模型能够量化地计算服务接口之间的相似程度，对候选 Web 服务实例按匹配相似程度排序，从而能够动态地从当前系统中，选择最优的服务参与组合。作者将在另一篇文章中详细介绍基于该匹配模型的 Web 服务自动组合过程。

总结 WSDL 是当前 Web 服务接口描述语言的标准，但是，它对接口的描述只限于语法层次，不能表达语义信息。本文通过对 WSDL 接口描述语言进行语义扩展，提出了一种基于扩展接口语义的 Web 服务描述模型。该模型从服务接口的消息参数层、消息层和操作层分别扩展语义描述，从而能够全方位地描述 Web 服务的功能以及服务质量等属性，并且这种语义描述模型，使得语义信息可以直接应用到当前的工业标准中。在此基础上，提出了多层次 Web 服务接口语义匹配模型。该模型能够对候选 Web 服务实例按匹配相似程度排序，从而能够动态地从当前系统中，选择最优的服务参与组合，这也是我们下一步的研究工作。

表 1 Web 服务接口语义匹配的层次和相应权重

消息参数层		消息层			操作层	
参数类型	语义概念	消息参数	消息类型	语法	静态语义	动态语义
ω_P^{DT}	ω_P^{Smcep}	ω_M^{P}	ω_M^{MT}	ω_{op}^{Syntax}	ω_{op}^{Static}	$\omega_{op}^{Behavion}$
ω_M					ω_{op}	

参 考 文 献

1 Casati F, Shan M C. Definition, Execution, Analysis, and Optimization of Composite E-Services. IEEE Data Engineering Bulletin, 2001, 24(1): 29～34
2 Akkiraju R, Goodwin R, Doshi P, et al. A Method for Semantically Enhancing the Service Discovery Capabilities of UDDI. In: Proceedings of IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03), Acapulco, Mexico, August 2003. 87～92
3 Lara R, Roman D, Polleres A, et al. A Conceptual Comparison of WSMO and OWL-S. In: Proceedings of European Conference on Web Services (ECOWS 2004), Erfurt, Germany, September 2004. 254～269
4 Christensen E, Curbera F, Meredith G, et al. Web Services Description Language (WSDL)1. 1. http://www.w3.org/TR/wsdl, March 2001
5 Bechhofer S, Harmelen F, Hendler J, et al. OWL Web Ontology Language Reference. World Wide Web Consortium. http://www.w3.org/tr/owl-ref, February 2004