

面向资源融合网格的数据语义框架^{*})

庞雄文 齐德昱 李拥军

(华南理工大学计算机科学与工程学院 广州 510630)

摘要 当前语义 Web 服务成了语义 Web 研究和应用的热点之一,提出了几种语义 Web 服务框架,但是这些架构主要是用于 Web 服务的自动发现、自动调用、自动服务组合,不适合描述网格环境下的数据语义。本文在目前的语义 Web 服务框架基础上,提出面向资源融合网格(FuseGrid)的数据语义框架(ODSF)。在 ODSF 中,数据对象的语义是自描述的;ODSF 包括 4 个顶层元素,分别为元属性、数据库本体、映射以及业务规则,其中元属性、数据库本体以及映射本体表示数据对象的静态语义,而业务规则描述数据对象的动态语义。同时分析了该框架的语义数据访问方法及其关键算法。

关键词 数据语义,资源融合网格(FuseGrid),本体

FuseGrid-based Data Semantic Framework

PANG Xiong-Wen QI De-Yu LI Yong-Jun

(School of Computer Science & Engineering, South China University of Technology, Guangzhou 510630)

Abstract Currently Semantic Web and Semantic Web Service have become hot spot of research and application in Semantic Web, several semantic Web service frameworks have been provided. But these service framework are designed for auto-discovery, auto-invoke and auto-composition for Web service, they are not suitable for describe the information in the grid. In this paper, we introduce a novel FuseGrid-based semantic data framework named ODSF whose top-level elements are meta-property, database ontology, mapping and business rule. Meta-property, database ontology, mapping of ODSF describe the static semantic of information, business rules of ODSF describe the dynamic semantic of information. This paper also discusses the semantic data accessing and its key algorithm.

Keywords Data semantic, FuseGrid, Ontology

1 引言

语义 Web 服务是 Web 服务和语义 Web 技术结合的产物,目前研究语义 Web 服务的主要机构是 W3C 成立的语义 Web 服务兴趣小组,提交的语义 Web 服务框架包括 OWL-S (OWL Web Ontology Language for Services)、WSMO (Web Service Modeling Ontology)、WSDL-S (Semantic Annotations for WSDL)、SWSF (Semantic Web Services Framework)。这几种框架之间的比较如表 1 所示^[1]。它们之间除设计目的(描述范围)均是用于 Web 服务自动发现、自动调用、自动服务组合和互操作外,模型的顶层元素、语义框架的静态描述语言、动态描述语言、对 Web 服务内部过程的描述等均不相同。从表 1 还可以看出,这些语义 Web 服务框架均不适合描述网格环境下的数据访问服务,数据访问服务除了要提供服务的自动发现、调用外,还需要对服务对应的分布式异构数据源提供语义支持,才能正确地访问数据。本文在目前的语义 Web 服务框架的基础上,提出了基于面向资源融合的数据语义描述框架(ODSF),该框架能正确地描述资源融合网格中数据的静态语义(数据结构、服务参数、安全控制以及服务能力)以及动态语义(逻辑规则和业务规则)。本文安排如下:第 2 部分介绍资源融合网格,第 3 部分提出了数据语义的概念,第 4 部分描述了 ODSF 框架,第 5 部分给出了基于 ODSF 的语义

数据访问过程及其关键算法,最后给出了本文的总结。

表 1 几种语义 Web 服务框架之间比较

	OWL-S	WSMO	SWSF	WSDL-S
范围	Web 服务的语义描述模型	Web 服务的语义描述模型与描述语言	OWL-S 模型的扩展	WSDL 文件的语义标注
顶层元素	服务参数、处理模型以及绑定	本体、目标、Web 服务以及映射	过程	操作或 WSDL 描述
服务级描述	服务的非功能性内容、服务级函数描述	能力、需求描述	未知	基于本体的关键词分类
操作级描述	过程的 IO-PE	服务自身的组合以及服务之间的交互	原子过程和复杂过程	WSDL 操作的前置条件和结果
静态语言	OWL	WSML	SWSLPOL & SWWLRule	未知
动态语言	过程模型和 OWL	抽象状态机	FLOWS	未知

^{*}中国博士后科学基金资助项目(2005037582);广东省自然科学基金资助项目(05200300)。庞雄文 博士生,主要研究方向是网格计算、计算机系统机构、本体集成;齐德昱 教授,博士生导师,主要研究方向是分布式系统、计算机系统机构、网络安全、网格计算等。

2 资源融合网格(FuseGrid)简介

资源融合网格(FuseGrid)是我们提出的一种网格体系结构。“资源”是指一切可应用于构建应用系统的对象,如各种软件体(对象、构件、服务、Agent等)、各种数据信息(数据、信息、Ontology、知识等)、各种设备/虚拟设备(高性能计算设备、传感设备、存储设备等);“融合”是指对资源的各种加工、处理、合成,使资源型、量、质等方面扩展、升华;“网格资源融合”是指资源的融合按网格的模式与网格的要求进行。

在 FuseGrid 中,资源融合采用纵横双向融合的方法。该方法的基本思想是,对难以抽象为自相似的问题,采用面向方面(Aspects)的本体(Ontology)驱动的插件技术,使资源升华为具有自相似特性的可归纳解决的标准网格资源——横向融合。对经横向融合变为符合纵向融合规范的资源,我们提出了 DD(数据资源到数据资源融合)、DP(数据资源和处理资源融合)、DS(数据资源和服务资源融合)、SP(服务资源和处理资源融合)以及同型融合等 5 大类纵向融合模型,按树型归并模式,对资源进行从粗到精、从简到繁、从小到大、从低到高的逐层融合,使其升华为新的纵向资源。这两种(纵横)融合方式可交替进行,直到融合出可代表应用系统的资源为止。

本文的应用背景是 FuseGrid 中的 DD 融合(数据资源到数据资源的融合)。要对数据资源进行融合,必须理解数据资源的语义才能进行更好的融合,所以本文提出了数据语义的概念以及面向资源融合网格的数据语义框架。

3 数据语义

数据语义是和相应的数据模型联系在一起的,数据模型是对模型所代表的领域世界的概念化的表示,在该模型下的数据语义表示该模型能解决领域世界问题的能力,因此数据语义的首要问题是在数据模型和该模型所对应的领域世界之间建立联系并维护这种联系。由于本体可以以人和机器可理解的术语描述领域世界的实体、对象、关系以及过程,因此我们使用本体来描述数据对象的语义和数据模型所表示的领域世界。

定义 1(数据对象 D) 数据对象是编码在预定义结构(如文件、表或者数据库)中的可以直接用计算机进行处理的信息。数据对象可以是任意形式的结构化数据^[2,3]。

定义 2(数据对象的语义描述 SD) 是一个三元组: $SD = \{D, M, O\}$, 其中 D 表示数据对象, O 表示用于描述领域世界的本体或者本体片断, M 表示数据对象和本体之间的联系或者映射,是使用本体(本体片断)对数据对象进行标注的结果。

数据对象的语义用领域世界中的概念进行描述,所使用的描述语言为数据语义描述原语;数据语义描述原语是一组用于描述数据内容的符号集合,这些符号所代表的含义由语义数据模型解释。数据对象的语义描述可以是物理上耦合的,如 XML 文件本身具自描述能力,数据的语义描述可以和数据定义一起存在;也可以是在逻辑上耦合的,如可以由数据模型通过建立一个文件来描述数据对象及其语义描述之间的对应关系。

定义 3(本体) 本体是一个五元组: $O = \{C, R, A, I, N\}$, 其中 C 为概念集合, R 表示概念之间的关系, A 表示定义在概念以关系上的限定和规则, V 表示概念实例的集合, N 表示名义(无实例的概念或者是用在概念定义中的实例)。

4 面向 FuseGrid 的数据语义框架

在 WSMO^[1]、OWL-S^[1] 的基础上,本文提出了面向资源融合网格的数据语义框架 ODSF(FuseGrid-based data semantic framework), ODSF 的语义基于描述逻辑编程(DLP), ODSF 包括 4 个顶层元素,分别为:

(1) 元属性(Meta Property)

元属性主要是为数据对象提供语义上下文(semantic context),一般的语义上下文包括数据的量纲、格式(长度和现实格式)以及对数据对象的通用描述信息。

(2) 数据库本体(Database Ontology)

数据库本体一般用以描述数据对象、数据库的能力参数、安全访问规则、数据隐私保护策略、QoS 参数等。

(3) 映射(Mapping)

在 ODSF 中的映射主要分为数据对象和领域本体之间的映射以及领域本体之间的映射两类。

(4) 业务规则(Business Rule)

规则包含数据对象之间的关系(主键、外键关系等)以及系统的应用层规则(如 VIP 客户的折扣是 8 折)两部分。数据对象之间的关系通过本体来表示,应用层规则使用 SWRL 表示。

ODSF 的顶层元素如图 1 所示,其中的元属性、数据库本体以及映射定义数据对象的静态语义,业务规则定义了数据对象的动态语义。

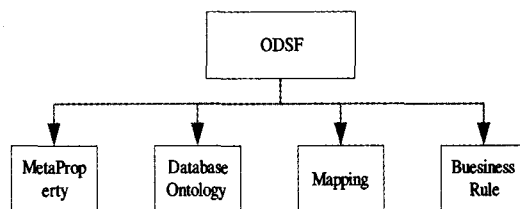


图 1 ODSF 顶层元素

4.1 元属性

元数据为数据对象提供语义上下文环境以帮助理解数据,元数据包括以下子属性:

(1) 量纲(unit)

量纲主要用来表示数值类型数据的计量单位,每一个量纲由以下属性和限制条件表示;量纲属性主要包括量纲名称、量纲类别、标称量纲(standradUnit)、转换系数(ratio)等。

(2) 数据对象属性限制(dataConstraints)

数据对象属性限制主要包括长度(length)、精度(digital)、显示格式(displayFormat)等属性。

(3) 数据对象属性描述(dataDescription)

主要的属性描述包括主题(subject)、主题描述(subject-Description)、创建日期(createDate)、创建者(creator)等属性;数据对象属性描述可以使 dublin 核心元数据集来描述数据对象的属性。

定义 4(简单数据语义上下文环境) SC 是一个三元组 $\{C, V, S\}$, 其中 C 表示数据对象或者概念, V 表示数据或者概念的实例, S 表示一个或者多个元属性的集合(这里的元属性只能是量纲或者是数据对象属性限制)。在语义上下文环境的三元组中, C 和 S 必须指定, V 是可选项;如果未指定 V , 表示定义的元属性集合适用于该数据对象(概念)的所有实例;

如果指定了V,则表述定义的元属性集合仅适用于该数据对象(概念)的指定实例。

如指定以下简单数据上下文环境{price, 200, {"USD", "currency", "RMB", 8.07}},表示 price 这个数据对象(概念)的实例是 200USD,并且货币的标称量纲是 RMB,从 USD 到 RMB 的转换系数是 8.07。

定义 5(复合数据语义上下文环境) CSC 是一个两元组 (C, S),其中 C 表示数据对象或者概念, S 表示一个或者多个元属性的集合(可以是任何元数据属性)。复合数据语义上下文环境主要用于标注数据对象(概念),不能用于标注实例。

4.2 数据库本体

数据库本体包含数据库模式(dbSchema)、能力(capacity)、服务质量(qos)以及访问策略(accessPolicy)4 个子概念。其中数据库模式概念表示数据库中有那些数据对象,能力概念表示数据库处理数据的能力(功能和性能)、服务质量概念表示数据库能提供的服务质量参数、访问策略概念定义数据库的访问策略。

(1) 数据库模式

数据库模式本体主要是描述数据库中的数据对象以及数据对象之间的关系,需要描述的主要数据对象包括数据表、数据列、主键,需要描述的主要关系包含数据库和数据表之间的关系、数据表和数据列之间的关系以及数据列之间的相互关系(引用关系、依赖关系)等^[4]。在使用 OWL 描述数据库模式本体时,使用到的主要类和属性如图 2 所示。

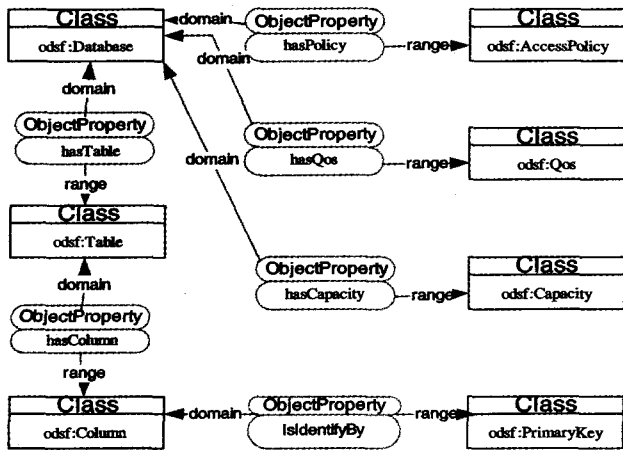


图 2 数据库模式

一般情况下,数据库模式本体不需要手工定义,可以根据数据库的 ER 模型自动生成。数据库模式本体为数据的本体化表示提供了一个缺省的表示方法,即使在没有定义映射规则的情况下都可以将数据根据缺省规则转换为本体实例在网络中过来传递。

(2) 服务能力

数据库的能力描述参数主要分性能和功能两类。其中性能用于表示数据库系统(数据库服务器)的和查询处理相关的性能参数(如 CPU 主频、平均可用内存等),这些参数主要用于查询优化;功能参数主要描述数据库系统相对标准查询语言所不具备或者特有的功能。对于标准查询语言中具备但是数据库系统不具备的功能,ODSF 提供功能补偿;对于数据库所具备的特有功能,ODSF 可以直接调用这些功能。

(3) 服务质量

在 ODSF 中描述服务质量本体时,使用的类包括 qos 参数类(odsf: qosQuality)、qos 参数属性(odsf: qosAttribute)、qos 参数值(odsf: qosMeasremen)、qos 参数之间关系(odsf: qosRealtion)以及复合 qos 参数类(odsf: qosComQuality),使用 qos 参数属性类和 qos 参数值类来描述一个 qos 参数,qos 参数之间可以具有独立、相关以及未知等三种关系;qos 参数可以按照类别划分为性能参数、可用性参数、数据表示类参数、成本类参数以及可靠性参数。qos 本体主要类组件的关系如图 3 所示。

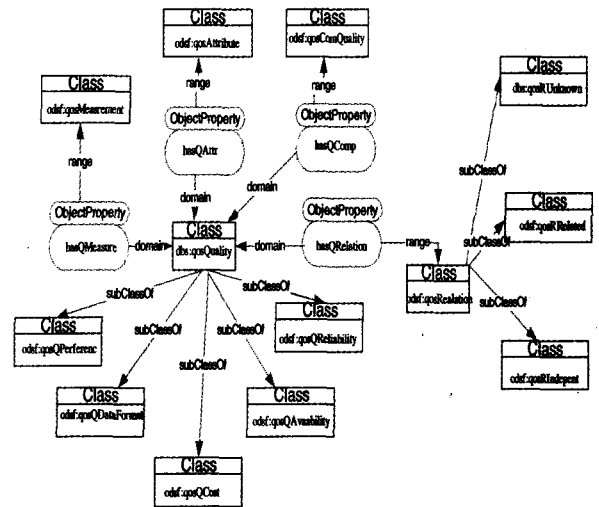


图 3 服务质量本体

(4) 访问控制

在 ODSF 中,采用基于规则的角色授权方法(RB-RBAC)来定义访问策略。在使用 OWL 定义访问策略时,主要的类包括用户类、角色类、权限类(分为对象权限和操作权限)及线程类,这些类之间的关系如图 4 所示。

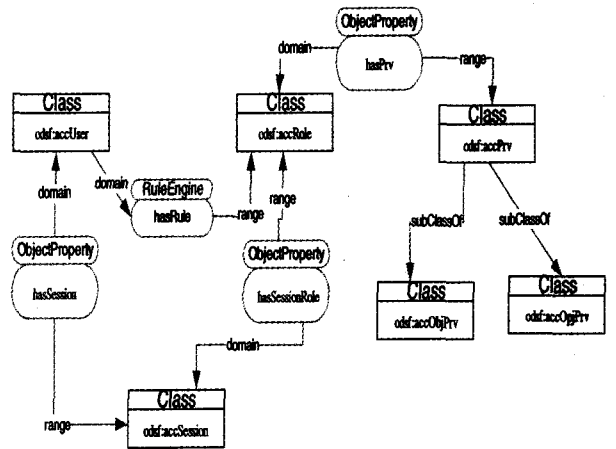


图 4 访问策略本体

4.3 映射

由于数据库和领域知识之间是独立设计的,因此数据库中和本体所表达的知识不一定是完全相同的;同理,数据对象和本体中概念所表达的知识也不一定完全相同,它们之间的关系如图 5 所示^[5]。

从图 5 可以看出,数据对象和本体之间的映射包括直接映射(数据库中的一个数据对象(表或列)对应本体中的一个概念)、多对一映射(一对多映射)以及多对多映射。本文仅以

直接映射为例说明直接映射的定义和语义。

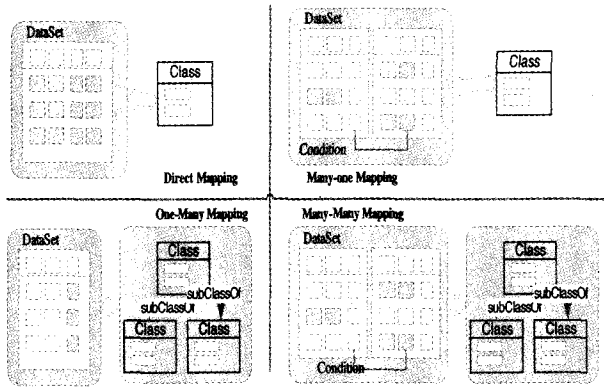


图5 数据本体和领域本体中概念的关系

定义6(直接映射) DP是一个四元组{D,C,TP,R},其中D表示数据对象(一般是一个数据列或者是一个数据表),C表示本体中的概念或者属性,T表示转换过程(定义数据库中的数据如何转换为本体中概念(属性)的实例),R表示映射过程的可信度。对于手工定义的映射过程可行度一般为1,对于自动产生的映射过程,可信度表示该映射过程的可靠程度。转换过程是一个将数据库中的数据转换为本体中实例的函数或过程,如在数据库中,使用{1,2,3,4,5,6,7}表示7种基本颜色,而在本体中使用{red,blue,...}表示7种基本颜色,数据转换过程需要定义它们之间的转换关系,如:0→red,1→blue等。

根据直接映射的定义,直接映射过程可以描述如下:给定两个本体S(源本体)和T(目标本体),对S中的一个元素 s_i ,使用本体的相关信息(本体定义、实例、业务规则或者限制、用户反馈),在T中找一个概念 t_i ,使 s_i 和 t_i 具备以下语义关系且这种语义连续的可信度为R(R为0~1之间的一个常数)^[6]:

- $s_i \subset t_i$,该直接映射称为合理的
- $s_i \supset t_i$,该直接映射称为完备的
- $s_i \equiv t_i$,该直接映射称为精确的

在直接映射中,我们只考虑 $s_i \equiv t_i$ 的情况;但是由于本体构建的分布性,在两个本体之间找到精确匹配的元素比较困难,因此我们通过相似度函数fsd计算两个本体元素之间的相似度,当相似度大于预先指定的阈值时,我们认为这两个本体元素之间是语义相等的,记为 $s_i \cong t_i$ 。

定义7 语义相似函数:

$$fs(s_i, t_i) = F(O, I, PM, R, U) \text{ 且 } fs(s_i, t_i) \in [0, 1]$$

其中O表示源本体S和目标T的定义(概念、角色以及关系),I表示实例,R表示规则或者约束条件,PM表示在S和T中的以前使用或映射,U表示用户反馈的信息。相似函数的输出为0到1之间的一个数;0表示 s_i 和 t_i 不存在任何已知的语义联系,1表示 s_i 和 t_i 是相等元素,返回的数值越大,表示 s_i 和 t_i 是相等概念的可能性越大。

在本文中,将本体的定义、本体的实例以及本体的规则或约束称为本体结构,根据语义相似函数的定义可以假设,本体之间的结构相似性越大,则本体之间的语义相似度也就越大。

定义8 直接映射的形式化定义:

$$\forall s_i \subset S, \forall t_i \subset T, DP = \max(fs(s_i, t_i))$$

其中S表示源本体,T表示本体本体。从定义可以看出,直接映射表示对于源本体S中的一个元素 s_i ,在目标本体中找到

一个元素 t_i ,使 s_i 和 t_i 之间的语义相似性最大。

直接映射的语义可以通过映射函数M来描述。假设用户理解领域知识U(可以是真正的领域知识,也可以是用户理解的个人知识),直接映射就是寻找一个的映射函数M,该映射函数可以将源本体S和目标本体T中的元素 s_i 和 t_i 映射为U中的元素 $M(s_i)$ 和 $M(t_i)$,使 $fs(s_i, M(s_i))$ 和 $fs(t_i, M(t_i))$ 最大;因为用户理解领域知识U,所以用户可以理解 $M(s_i)$ 和 $M(t_i)$ 的语义关系。 $t_i \cong M(t_i)$ 可以解释为:对真实世界W,存在针对源本体S的解释 I_S 和针对领域知识U的解释 I_U ,可以将 t_i 和 $M(t_i)$ 分别映射到其中的同一个概念,记为 $\{I_S, I_U\} = \{s_i \cong M(s_i)\}$ 。

定义9(映射函数M) S和T表示源本体和目标本体,U为用户理解的领域知识,将S和T的元素映射到U的映射函数M表示:分别在S、T和U中存在解释 I_S, I_T 和 I_U ,满足以下条件:

$$\begin{aligned} \forall s \subset S, \{I_S, I_U\} &= \{s \cong M(s_i)\} \\ \forall t \subset T, \{I_T, I_U\} &= \{t \cong M(t_i)\} \end{aligned}$$

从直接映射的定义可以看出,直接映射具有对称性,即 $DP(s_i, t_i) = DP(t_i, s_i)$;另外直接映射还具有传递性,假设 $DP(s_i, t_i)$ 和 $DP(t_i, h_i)$,则 $DP(s_i, h_i)$ 。

4.4 业务规则

使用描述逻辑(SHOIN和SHIF语言)进行和个体相关的推理(特别是实例提取)是比较困难的。如选择一个类所有的实例,需要对该类的每一个实例进行检查;当类的实例比较多时,数据提取过程就很满。提高描述逻辑中个体推理效率的主要方式是将描述逻辑和规则结合起来,使用描述逻辑进行属于定义和类之间的推理,使用规则描述同语言进行个体相关的推理。在ODSF中,提出了基于描述逻辑程序(DLP)^[7]的规则描述语言SWRL,来描述业务规则(将在以后的文章中介绍)。

5 ODSF中的数据语义分析

从以上ODSF的分析可以看出,ODSF中数据的语义是通过数据上下文环境以及数据对象的语义描述来体现,因此我们可以认为: $SD = \sum_{i=1}^N (D_i, SD_i)$,其中 D_i 数据对象, SD_i 表示 D_i 这个数据对象的语义描述,数据对象之间的语义联系由对应的语义描述之间的语义联系来确定。

性质1 设 SD_i 和 SD_j 是语义数据模型中的两个数据描述,称 SD_i 和 SD_j 具有语义联系,如果:

- (1) 包含语义
 $SD_i \subset SD_j$,当且仅当 $\forall x \subset SD_i$,则 $x \subset SD_j$;
- (2) 相同语义
 $SD_i \subset SD_j$,当且仅当 $\forall x \subset SD_i, \exists x \subset SD_j$;
 $\forall y \subset SD_j, \exists y \subset SD_i$;
- (3) 不相同语义
 $SD_i \neq SD_j$,当且仅当 $\forall x \subset SD_i, \rightarrow \exists x \subset SD_j$;
 $\forall y \subset SD_j, \rightarrow \exists y \subset SD_i$;

(4) 语义并。在语义数据模型中,如果 SD_i 和 SD_j 描述的是相同的数据对象(使用同一本体中不同类描述、使用不同本体中概念来描述),则称呼 $SD_i \cup SD_j$ 为该数据对象的语义并。

(5) 等价语义。假设某一数据多项的语义可以通过某一领域本体来完整描述并且使用语义描述来描述数据对象的语

义的描述过程是完整的,如果 SD_i 和 SD_j 描述的是同一个语义对象,并且 SD_i 和 SD_j 中的概念(类)分别来自不同的领域本体,则称 SD_i 和 SD_j 为等价语义,记为 $SD_i \cong SD_j$ 。

(6) 依赖语义。如果 SD_i 和 SD_j 存在依赖关系(属性关系、部分-整体关系、业务规则上的依赖关系),则称 SD_i 和 SD_j 之间具有语义依赖关系,记为 $SD_i \xrightarrow{ds} SD_j$ 。

性质2 设 D_i 和 D_j 是语义数据模型中的两个数据对象, SD_i 和 SD_j 分别是 D_i 和 D_j 对应的语义描述,称 D_i 和 D_j 具有语义联系,如果:

- (1) 如果 $SD_i \subset SD_j$, 则 D_i 是 D_j 的子对象
- (2) 如果 $SD_i = SD_j$, 则 D_i 是 D_j 的相同对象
- (3) 如果 $SD_i \neq SD_j$, 则 D_i 是 D_j 的数据对象

如果 $SD_i \xrightarrow{ds} SD_j$, 则称 D_i 是 D_j 是相互依赖的数据对象。

6 基于 ODSF 的数据访问

在 ODSF 中数据访问过程 AP 如图 6 所示。

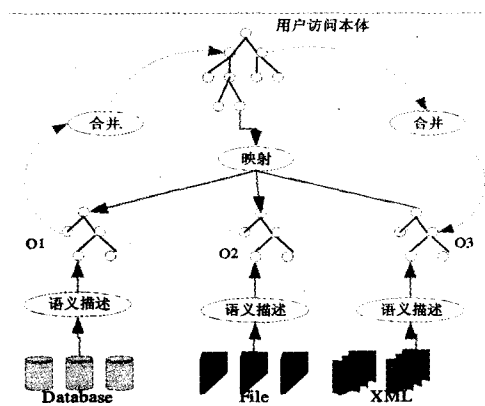


图 6 ODSF 下语义数据访问过程

AP 是一个的 5 元组: $AP = \{D, SD, O, OM, SM\}$ 。其中 D 表示数据对象的集合; SD 表示数据对象语义描述的集合; O 表示描述数据对象所描述的领域世界的本体的集合; OM 表示本体集合 O 中本体之间映射关系; SM 数据对象对象机器描述本统之间的映射关系。

主要过程描述为:

(1) 用户通过使用用户访问本体进行抽象层次上的数据访问,也可以直接使用数据本体中概念进行数据访问。这里假设在用户本体和数据本体之间没有其他层次的本体。

(2) 将对用户访问本体的访问映射为描述数据对象的本体(简称为数据本体)的访问,由于数据本体和用户访问本体之间的创建过程可能是独立进行的(也可能是通过合并数据本体进行的),因此可以采用 LAV(local-as-view)模式,将用户访问本体视为全局模式,数据本体视为全部模式的局部视图。

(3) 将对数据本体的访问映射为对数据对象的访问,一样采用 LAV(local-as-view)模式,将数据本体视为全局模式,数据对象视为全部模式的局部视图。

(4) 将来自各数据对象的查询结果进行合并后返回给用户。

算法描述如下(由于版面关系,本文不详细讨论该算法):

Algorithm: 基于 ODSF 的语义数据访问
 Inputs: query - 用户本体表示的查询
 Context - 上下文环境
 Global: UO - 用户本体

```

DO - 数据本体
Outputs: 查询结果
Method:
//检查查询语句的有效性
If context = "UO" then
    Check_Valid(query, UO)
Elseif context = "DO" then
    Check_Valid(query, DO)
End if
//在用户本体和数据本体之间映射
If context = "UO" then
    DGM(query);
Elseif context = "LO" then
    //转到查询重写
End if
//根据数据本体中的规则重写查询
Inferred_rule = GetInferredRule(query);
RewritingQuery1(query, Inferred_rule)
//将以本体概念表示的查询转换为以数据源表示的查询
for i = 1 to query.COUNT //查询对应的数据源个数 //载入数据对象的语义描述
    SD[i] = LoadSD(i);
    //重写查询为标准格式
    SubQuery[i] = RewritingQuery2(query, SD[i])
    //查询重写后查询的正确性
    Check_Valid(SubQuery[i], SD[i])
Next
//在数据源上执行查询
For j = 1 to SubQuery.count
    //重写查询为对应数据源的格式
    ExeQuery[j] = RewritingQuery3(SubQuery[j], SD[j])
    //执行查询
    Result[j] = TDOExecite(ExeQuery[j])
Next
//使用数据融合服务消除重复数据
FuseData(Result)
//显示结果
DisplayData(Result)
    
```

总结 本文首先分析了数据语义的含义,给出了数据语义的相关概念;并在目前语义 Web 服务框架的基础上提出了面向资源融合网格的数据语义表示框架(ODSF)。在 ODSF 中,数据对象的语义是自描述的;ODSF 包括 4 个顶层元素,分别为元属性、数据库本体、映射以及业务规则,其中元属性、数据库本体以及映射本体来表示数据对象的静态语义,业务规则来描述数据对象的动态语义;给出这 4 个顶层元素的详细定义。文章还提出了基于 ODSF 的语义数据访问过程及其关键算法。

当前,我们正在研究 ODSF 中的关键过程,如自动本体映射、具有服务质量保证的查询重写、一致的用户访问界面等,同时也正在实现基于 ODSF 的语义数据访问原型系统。当然,还有一些实现问题需要进一步的研究和解决。

参考文献

- 1 Stollberg M. State of Affairs in Semantic Web Services. In: 1st Canadian Semantic Web Working Symposium (CSWWS 2006), 2006
- 2 Kiryakov A, Ognyanov D, Kirov V. An Ontology Representation and Data Integration(ORDI) Framework: The DIP project Glossary. <http://dip.semanticweb.org/deliverables/D22ORDIv1.0.pdf>
- 3 Borgida A, Mylopoulos J. Data Semantics Revisited [J]. In: LNCS 3372, 2005. 9~26
- 4 de Laborda C P, Conrad S. Relational OWL - A Data and Schema Representation Format Based on OWL. In: Second Asia-Pacific Conference on Conceptual Modelling 2005(APCCM), 2005. 89~96
- 5 Barrasa J, Corcho O, Gómez-Pérez. R2O: An Extensible and Semantically Based Database-to-Ontology Mapping Language. In: Second Workshop on Semantic Web and Databases (SWDB2004), 2004. 92~119
- 6 Haase P, Motik B. A mapping system for the integration of OWL-DL ontologies. In: the First International Workshop on Interoperability of Heterogeneous Information Systems (IHIS05), 2005. 181~190
- 7 Grosf B N, Horrocks I, Volz R, et al. Description Logic Programs: Combining Logic Programs with Description Logic. In: Proc. of the Twelfth International World WideWeb Conference, 2003. 45~87