

本体演化管理研究^{*})

刘柏嵩^{1,2} 高 济²

(宁波大学网络中心 宁波 315211)¹ (浙江大学计算机学院 杭州 310027)²

摘要 自 W3C 主席 Tim Berners-Lee 首先提出了语义 Web 的概念后,它正在成为计算机信息处理领域当前研究的热点之一。本体将在“语义 Web”中起到至关重要的作用,它通过定义精确的共享术语,以提供某一特定领域可重用的知识。但是这些知识并不是静态的,而是随着时间的推移不断演化。领域的改变、自适应不同的任务、或概念模型的变化都要求本体的变更。随着本体开发变成一个泛化的、协同的过程,本体版本控制和演化管理已成为本体研究中一个重要的领域。本文首先对本体演化的原因和所带来的问题进行分析,然后讨论了本体演化管理的关键技术,着重强调了 Web 上本体标识和本体变化机制的定义,并对今后的研究工作进行了展望。

关键词 本体,本体演化,本体管理,语义 Web

A Study on Ontology Evolution Management

LIU Bai-Song GAO Ji

(Network Center Ningbo University, Ningbo 315211)¹

(Institute of Computer Science, Zhejiang University, Hangzhou 310027)²

Abstract Since Tim Berners Lee, current W3C chairman, first proposed the Semantic Web, it is becoming the hot topic of computer information processing area. Ontologies are set to play a key role in the “Semantic Web”, as they provide a reusable piece of knowledge about a specific domain. However, those pieces of knowledge are not static, but evolve over time. Domain changes, adaptations to different tasks, or changes in the conceptualization require modifications of ontology. As ontology development becomes a more ubiquitous and collaborative process, ontology versioning and evolution becomes an important area of ontology research. The paper first analyzes the causes and the consequences of ontologies evolution. Then the key technologies of ontology evolution management are discussed. Ontologies identification and the definition of ontology change mechanisms are our focuses. Finally we propose the future work of ontology evolution management.

Keywords Ontology, Ontology evolution, Ontology management, Semantic Web

1 问题的提出

自 W3C 主席 Tim Berners-Lee 首先提出了“语义 Web”(Semantic Web)的概念后,它正在成为计算机信息处理领域当前研究的热点之一。语义 Web 的基本思想是对当前的 WWW 进行扩展,使得网络中所有信息都是具有语义的,是计算机能够理解和处理的,便于人和计算机之间的交互与合作^[1]。数据语义的明确表示和领域理论的应用将使得 Web 提供一种全新质量的服务,其最终目标是将人类知识编织成一个巨大的网络,并以机器处理的方式来实现它。各种自动化服务将帮助用户以计算机可理解的格式访问和提供信息,由此使得计算机自动化处理过程和 Web 信息集成更为方便。

本体(Ontology)定义了 Web 上用于描述和表示领域知识的术语^[2],是共享概念模型的明确的形式化规范说明。由于本体是语义 Web 实现的关键,这将使得本体的数量和规模大大增加。在万维网这样一个动态发展的环境,任何人可随时发送和更新信息。为此,用户必须同发布其它信息一样方便地发布 Web 本体,同时也必须允许对这些本体进行修改订正,并由此衍生了所谓的“本体演化”(Ontology Evolution)的问题。自从 20 世纪 90 年代以来,对有关基于本体的系统的研究非常活跃,围绕本体召开了为数众多的专题研讨会。但是,从作者检索文献的情况来看,讨论本体演化方面的研究比较少。目

前大多数本体库系统(Ontology Library Systems)并没有考虑本体演化的问题,通常这些系统是用于研究的系统原型,相互之间的关联性并不明显。对于集中式系统而言,本体的某一变化与任何相关信息的变化同步,因此无需变更管理。例如,在 Ontolingua 系统中,干脆忽略了对本体旧版本的管理。Helfin 指出 Web 上的本体需要不断演化,并提出一种基于 Web 的知识表示语言 SHOE 以支持本体的多个版本^[3]。其它研究领域也对该工作有一定的借鉴作用,如关系数据库的模式演化和模式版本控制。但是由于知识模型和应用方式的不同,两者有很多差别^[4]。另外 Natalya F. Noy 也提出一种比较本体版本的不动点算法 PromptDIFF^[5]。

本文首先结合实例分析了本体演化的原因、本体演化所带来的问题以及本体演化的模式,然后讨论了在语义 Web 环境下本体演化管理(OEM, Ontology Evolution Management)的关键技术。

2 本体演化分析

2.1 本体演化的原因

有多种原因会导致本体的演化。根据 Gruber 的定义,本体是指某一领域共享概念模型的明确表示和描述。因此,引起本体变化的原因包括如下几个方面:

1. 领域的变化 领域的改变非常普遍,它们的影响也与

^{*}) 本课题得到浙江省自然科学基金资助(M603010)和宁波市青年博士基金(2003A62002)资助。刘柏嵩 博士生,研究兴趣包括语义 Web、Agent 计算、本体工程。高 济 教授,博士生导师,主要从事人工智能、软件工程、CIMS 等领域的研究。

数据库模式的影响基本类似。Ventrone^[5]总结了现实世界的变化(领域演化)带来数据库模式的更新。例如,当两个具有不同管理机构的大学里的系进行合并时,描述该领域的本体需要变更以反映这种变化。

2. 共享概念模型的变化 概念模型的变化是由于领域视图或使用角度的改变引起的,即上下文语义(Context Semantic)的变化。当本体应用于新的任务或新的领域时,则概念化的表示也要相应地变化。例如,某一本体中“Class”用于表示“开设的课程”,如“Class(alg_Fall_2002)”,当变成表示“一堂课”时如“Class(Databases)”时,就会发生错误。

3. 表示的变化 表示(Representation)的变化是指一种转换,当本体由一种知识表示语言翻译成另一种语言表示时,就会产生显式定义的改变。这些语言不仅语法各异,而且更重要的是语义和表达也各不相同。因此,在转换过程中保持本体的语义一致并非易事。

2.2 本体演化带来的问题

2.2.1 本体变化的影响 本体演化所带来的重要影响之一是它可能导致不兼容。本体的不兼容是指原始本体被已改变的本体版本所代替时,会对遵从该本体的数据或应用系统产生以下影响:

(1)当本体用于定义数据的含义时,该数据可能得到不同的解释或采用了未知的术语,例如基于本体的 Web 页面标注。在这种情况下,兼容性意味着通过已改变的本体能够正确地解释所有数据。这与数据库模式的版本控制的兼容性非常类似,即所谓“实例(Instance)数据的维护”。

(2)如果该本体由其它本体构造,源本体的改变会影响目标本体的含义。这种本体逻辑(概念模型)形式的变化一般来说比较重要,其它导入该本体的正确性也取决于此,这种兼容性是“结果的维护”。

(3)基于本体的应用系统也会受到本体变化的影响。理想状况是,系统应用中的概念模型知识应只能由本体来定义。但是在实际应用中也会采用内部模型,即将本体(或领域知识)隐含在程序代码中,该内部模型可能与本体产生不兼容。这种兼容的解释是“本体查询应答的维护”。

2.2.2 典型变化及其定义 本体演化中另一个问题是对变化的定义方式。在本体中存在许多类型的变化,从简单的重命名到复杂的转换等。通过对本体变化的分析,典型本体的变化有:类名的变化,类的增加,类在分类层次中的重定位(在分类层次中的提升或下降,或水平移动),整个子树的重定位,类的融合,类的分割,以及删除操作等。但是,目前的版本控制技术比较简单,如在 UNSPSC 中,所有的变化都归结为增加、删除或编辑(名称变化)。

2.3 演化的两种模式

确定特定变化对本体版本之间的兼容性的影响是非常重要的。但是,由于 Web 高度分布的特性,我们不可能总是跟踪由一个版本到另一版本的变化。因此,在此我们可以区分本体演化的两种模式:可跟踪的和不可跟踪的演化^[4]。跟踪的演化可大体上类似于数据模式的演化,可将演化看作本体变化的一个序列。每一个改变本体的操作后(如增加或删除类,将槽附加到类等),考察实例数据和对本体的影响,影响结果由改变操作的组合所决定。

对于不可跟踪的演化,用户只能看到两个版本的本体,而对由一个版本到另一个版本的变化步骤不得而知,本文侧重研究该模式的本体演化。由此需要以自动或半自动化方式找出两个版本之间的差异。查找本体(版本)之间的差异的问题实际上与本体融合(Merge)非常接近,二者都有两个交叉的

本体且需要确定它们的元素之间的映射(Mapping)。当进行本体融合时,我们集中于相似性,而在本体演化中需要强调差异,这是一个互补的过程。

2.4 本体演化实例

定义 1 本体是一个三元组 (V, A, E) ,其中词表 V 是谓词符号的子集,公理 A 是合适公式的子集,而 E 是本体扩展集。

定义 2 令 K 为 $R \rightarrow 2^R$,将每一资源映射为一组合适公式的函数,称 K 为“知识函数”,因其提取包含在资源中的知识并为其提供公理。 R 为因特网资源集,包括任何通过因特网提供信息的事物,如 Web 页面、新闻组或 E-mail 信息等。

$O_u = \langle \{Faculty\}, \phi, \phi \rangle$
 $K(r_1) = \{Faculty(Dr Li)\}$
 $K(r_2) = \{Faculty(Dr Zhang)\}$

变化后的本体和资源:

$O'_u = \langle \{Faculty, AssistProf, AssocProf, Professor\},$
 $\{AssistProf(x) \rightarrow Faculty(x);$
 $AssocProf(x) \rightarrow Faculty(x);$
 $Professor(x) \rightarrow Faculty(x)\},$
 $\phi \rangle$
 $K(r'_1) = \{Faculty(Dr Li)\}$
 $K(r'_2) = \{Faculty(Dr Zhang)\}$
 $K(r'_3) = \{AssocProf(Dr Wang)\}$

图 1 本体变更:增加术语项

在图 1 中说明了当一个新的术语项添加到本体中时所带来的影响。该例中 O_u, r_1 和 r_2 表示一个简单的“University”本体及两个遵从该本体的资源。本体定义为一个三元组 (V, A, E) 。

因此, O_u 包含一个术语 Faculty, 而 r_1 和 r_2 应用“Faculty”谓词的资源。

经过一段时间后, O'_u, r'_1, r'_2 和 r'_3 代表相关 Web 对象的状态。本体 O'_u 表示一个新版本的 O_u , 它包括表示“Faculty”子类的术语。当本体设计者以这种方式增加术语项时,就有可能增加公理(原子公式)。如 $Professor(x) \rightarrow Faculty(x)$ 以帮助定义术语。其中 r'_1 和 r'_2 是变化后的 r_1 和 r_2 , 因为 $K(r'_1) = K(r_1)$ 且 $K(r'_2) = K(r_2)$, 这些资源并未发生变化。由于词表 O'_u 的词汇表 V' 是 V 的一个超集,对于 O'_u 而言 r_1 和 r_2 仍是定义明确的。一旦 O_u 改变为 O'_u , 我们就可应用 O'_u 中的新术语项创建资源, r'_3 就是包含有关 Dr Wang 断言的一种资源。

但是,如果从本体中删除一条术语项,则现存的资源可能是非明确定义的。例如,某一资源所遵循的本体不再包括其断言中应用的谓词。另外,如果术语的含义发生变化,也会产生严重的问题,甚至会由此得出错误的结论。

3 关键技术

从前面的分析可以看到,在 Web 环境下本体的演化是不可避免的。因此为了实现不同版本的本体表示和知识之间的互操作,有必要在本体版本之间建立连接,实施本体演化管理^[6]。但是,在 Web 这个分布式环境下就本体达成一致并对其维护是一项具有挑战性的任务,目前没有此类支持工具。本节在对本体演化管理需求分析的基础上,对其关键技术作进一步的探讨。

3.1 本体演化管理需求分析

根据本体工程和本体应用的实践,本体演化的需求如下:

- (1)它必须能解决本体变化的问题,并确保底层本体和所有相关本体的一致性;
- (2)它应能让用户更方便地监控和管理变化;
- (3)由于相同的本体可能由不同的用户自行修改,需要经

常集成本体的不同版本,它应能将同一本体的不同版本再次集成;

(4)它应能向用户提供反复本体精化的建议。

其核心问题是:在开放的、分布式环境下,必须采取何种机制和方法以支持来自不同信息源本体的合并和改变? 本体演化管理的基本目标就是提供管理本体变更的机制和技术,并同时实现与已有数据和应用的互操作^[7]。该基本目标进一步分解为以下要素:

(1)标识:对每一个概念、关系及本体自身,系统应为其潜在的定义提供一种无歧义性参考。详见本文 3.3 节。

(2)变更机制的定义:包括定义本体版本之间的关系、本体定义变更和概念化变更的差异以及本体更新的方式(粒度)。在以下 3.2 节中作进一步说明。

(3)透明演化:系统尽可能自动对本体版本和数据源进行翻译和关联,以实现透明的信息访问。

3.2 本体变更机制的定义

一般来说,本体包括一组类(Class)或概念(Concepts)定义、属性(Attributes)定义和相关公理(Axioms)。类、属性和公理相互关联并形成一部分世界的模型。某一变化构成了一个新版本的主体,并由此形成了本体的原始版本和新版本中概念和属性定义之间的版本关系。

本体内部概念的关系,如类 A 和类 B,一个概念的两个版本之间的版本关系有着本质的区别,例如类 $A_{1.0}$ 和类 $A_{2.0}$ 。在第一种情形下,关系是一种单纯的领域概念关系;而后者,该关系描述了概念变化的元信息。

但是一个概念的两种版本仍有某种概念关联。也就是说,更新(Update)关系本身虽不是一种概念关系,但概念的各种版本(如 $A_{1.0}$ 和 $A_{2.0}$)之间仍具有某种概念(逻辑)关联。另一个本体变更的要素是显式定义的改变和概念模型的变更之间的差异。本体是指一种概念模型的定义,概念和属性的定义因此是一种概念模型的特定表示;即相同的概念可能有不同方式的定义。因此,定义的变化并不一定意味概念模型的变更。

因此,按本体的变更是否影响概念模型定义如下:(1)概念级变化:指领域解释(概念模型)的变化,它导致本体概念的不同或这些概念之间关系的差异;(2)解释变化:概念模型定义方式的变化,并不改变概念模型本身。

第三个更新的要素是本体变更的方式。这也是开发本体管理系统中一个重要的实际问题。对于变化定义的实现,我们可以区别出两种形式,一是定义的粒度;既可是单一“定义”的层次,也可是整个“文档”的层次;二是定义的方法,包括转换定义、替换法和映射方法等。

3.3 语义 Web 上本体的标识

由前面的讨论可以看出,在网络上对本体版本的标识非常重要。本体描述了一种部分世界的统一视图并作为该特定概念模型的参考,因此,它们应具有唯一而稳定的标识。遵循某一本体的 Agent 或应用系统,应能无歧义性地指向它。

在语义 Web 环境下,本体可看作是一个在网络上的文件。每一个导致本体不同特征表示的变更构成了一个本体的修订版本(Revised Version)。如果本体的逻辑定义没有改变,则需要确定修订是否为概念级变化并形成新的概念模型,或仅仅是相同的概念模型的不同表示。

为了将本体关联到 Web 资源和它们的 ID,可以借鉴当前万维网上的标识机制(包括 URIs, URNs 和 URLs)。W3C 将 Web 上的事物称作“资源”。根据统一资源标识符 URIs 的定义,“资源可以是任何具有标识的事物”,或者说“资源是一种概念实体”。这两个定义都可将本体包含进去,因此本体又

可以看作是一种资源。这样看来,URI 作为“一种标识抽象或具体资源的字符串”可用来标识本体资源。与 URL 只限于资源定位相反,URIs 提供了一种通用的标识机制。

另外,还考虑到本体的标识应与定义该本体的 Web 文件完全区别开来,也就是说,本体资源类不同于文件资源类。本体的修订一般定义为一个新文件,每一个修订版是一个新的文件资源并获得一个新的标识,但并没有自动获得一个新的本体标识。综合考虑以上因素,文[2]提出一个基于如下原则的标识方法:

(1)区分三类资源:文件、本体和向后兼容本体;

(2)采用 URL 作为文件标识;

(3)文件的改变即产生一个新的文件标识符;

(4)概念级或逻辑定义的变更产生一个新的本体标识符,而非逻辑的解释性变化则不用;

(5)采用两级数字方案的 URI 作为本体标识(如 3.21):小数作为向后兼容的修改(以一个小数结尾的本体 URI 标记某一特定的本体);主数(整数)作为不兼容的变化;

(6)单个概念或关系,如果标识符只存在小数的差别,则互等价;

(7)由某一本体 URI 参考的本体带有相应的主版本号 and 最小附加约束,即最少必须的小数版本修订版本号。

区分本体标识和文件标记有很多好处,由此文件的改变和位置的变更(如本体的拷贝)能够与本体的变化区别开来。通过应用独立的 URI,只需对所需信息编码,同时也避免了与定义位置的 URL 混淆。

结论及进一步的工作 在语义 Web 中本体将起到重要的作用以定义和关联描述 Web 上数据的概念。但是,Web 的分布性和动态性的特点导致了多版本本体的出现。在这种情况下,本体可能由多个人开发,并随着时间的推移而不断演化。而且,领域的变化、对不同任务的自适应,或概念模型的改变,都可能导致本体的修改。可以说,本体演化问题在语义 Web 上将无处不在。本文在分析本体演化机理的基础上,讨论了本体演化管理的关键技术,尤其是本体标识和本体变更机制的定义问题。

为了进一步深入对本体演化的研究,我们将着重考虑本体版本比较问题,提出本体相似性比较的机制和算法,并开发相应的系统帮助用户管理在线本体的变化。今后,我们将在 PromptDIFF 算法的基础上^[9],结合机器学习方法和概念图匹配算法^[10],采用分层原理的思想,对本体的表示(语法)和本体的语义(包括概念、关系和公理)的变化进行综合比较。

参 考 文 献

- 1 Berners-Lee T, Hendler J, Lassila O. The Semantic Web. *Scientific American*, 2001, 284(5):34~43
- 2 Klein M, Fensel D. Ontoview: Web-based ontology versioning. In: 1st Intl. Semantic Web Conf. Sardinia, Italia, June 2002
- 3 Heflin J, Hendler J. Dynamic Ontologies on the Web. In: Proc. of the Seventeenth National Conf. on Artificial Intelligence (AAAI-2000). AAAI/MIT Press, Menlo Park, CA, 2000. 443~449
- 4 Klein M. Supporting evolving ontologies on the internet. In: Proc. of the EDBT 2002 PhD Workshop, Prague, Czech Republic, March 2002
- 5 Ventrone V, Heiler S. Semantic Heterogeneity as a Result of Domain Evaluation. *SIGMOD Record Special Issue. Semantic Issues in Multidatabase Systems*, 1991, 20(4):46~54
- 6 Maedche A, Motik B, Stojanovic L, Stojanovic N. User-driven Ontology Evolution Management. In: Proc. of the 13th European Conf. on Knowledge Engineering and Knowledge Management EKAW, Madrid, Spain, Oct. 2002

网格数据库管理模型与策略

史 隆 都志辉

(清华大学计算机科学系 北京 100084)

摘 要 越来越多的网格应用需要管理大容量和广域分布的数据。开放网格服务体系结构中的网格服务提供了动态创建、管理和在网格服务中交换的一致接口。本文探讨了以 OGSA 网格服务管理网格数据库的模型,网格数据库服务提供支持数据访问的控制和发现、执行数据管理的操作,实现数据资源的虚拟化,通过网格实现现有数据库的访问与集成。同时讨论了相关的优化策略。

关键词 网格数据库服务,开放网格体系结构,数据网格

Grid Database Management Model and Strategy

SHI Long DU Zhi-Hui

(Department of Computer Science, Tsinghua University, Beijing 100084)

Abstract An increasing number of grid applications manage data at very large scale, of both size and distribution. OGSA defines Grid Services with consistent interfaces for creating, managing and exchanging information among Grid Services dynamically. This paper explores the model of managing grid databases with OGSA Grid Service. Grid database service provides in supporting discovery of and controlled access to data, in performing data manipulation operations, and in virtualising data resources. It implements access and integration of data from existing databases via the Grid. It also discusses the relevant performance tuning strategy.

Keywords Grid database service, OGSA, Data grid

1 前言

誉为互联网第三次浪潮的网格技术^[1,2]正从科学计算领域推向广阔的工业领域,在企业资源管理、供应链管理、客户关系管理、企业知识管理、电子商务、电子政务等应用领域的这些信息管理系统,其核心是数据库。网格按应用可分为计算网格、数据网格、科学网格、知识网格等,数据库是数据网格的重要部分,网格数据库通过现有数据库的网格化,以供客户持续的、可靠的、高性能低价格的网格数据库服务。对于高负载的大数据应用,应用网格数据库服务,综合利用网格中的数据库资源,通过资源共享和分布计算,实现负载均衡,将作业分配到多台数据库服务器上执行,提高性能。对于小数据库应用,不再需要配置数据库服务器,加入网格虚拟组织,并从中申请网格数据库服务即可,得到更高的性价比,使用更加便捷。本文阐述了网格数据库与网格的关系,如何管理网格数据库,创建网格数据库服务,网格数据库复制以及服务质量策略。

网格是一个集成的计算与资源环境,试图汇聚分布于网络上的各种高性能计算、服务器、PC、仪器设备、软件、知识等资源,在动态的、包含多个机构的虚拟组织中,协同资源共享和问题解决。网格所协调的资源 and 用户一般不是中央控制的,往往存在于多个控制域,为了在这样一种松散耦合的结构中

控制和管理各种资源,需要使用标准的、开放的和通用的协议和接口来解决诸如认证、授权、资源发现和资源访问这些基本问题,网格作为一种新的计算基础设施,向虚拟组织中的大量用户提供服务,它必须能提供高质量的服务,这种服务基础设施必须是可靠的、易于访问的、可伸缩的、安全的并且价格低廉。

网格体系结构可从两个角度来描述。一个是以协议为中心的五层沙漏结构^[1],另一个是以服务为中心的开放网格体系结构(Open Grid Service Architecture, OGSA)^[1,2]。

五层沙漏结构从底层开始分别为构造层、连接层、资源层、汇聚层和应用层。对网格数据库而言,构造层包括数据库系统(计算机、存储系统、数据库管理软件,作为一个整体来考虑)资源、软件资源、网络资源、目录资源、工具包。连接层有通信协议、安全认证 GSI、统一的基于 PKI 的认证、授权、消息保护机制、工具包。连接层有通信协议、安全认证 GSI、统一的基于 PKI 的认证、授权、消息保护机制、单点登录、委托代理、身份映射等。资源层包含信息协议(资源的结构和状态信息;数据库系统的配置、负载、使用策略,配置包括数据库的类型、性能,如关系型,XML, LOTP, LOAP,性能测试值 TPC-C、TPC-H, TPC-R 等,数据库最大容量)、管理协议(磋商对资源的访问,分配、预留和监视、控制)。汇聚层提供协同分配、调度及代理服务、数据库复制服务、监视和诊断、故障恢复服务、元

*)本项目受清华大学基础研究基金、华为基金以及 IBM 基金的资助。史 隆 硕士研究生,主要研究方向为数据网格、网格数据库、PKI 和 J2EE 等。都志辉 副教授,主要研究方向为网格计算与集群式计算。

7 Maedche A, Motik B, Stojanovic L, Studer R, Volz R. Managing Multiple Ontologies and Ontology Evolution in Ontologging. In: Proc. of the Conf. on Intelligent Information Processing, World Computer Congress 2002, Montreal, Canada, Kluwer Academic Publishers, Sep. 2002

8 Noy N F, Klein M. Ontology Evolution: Not the same as schema evolution. Knowledge and Information Systems, 5. in press, Jan. 2003

9 Noy N F, Musen M A. PromptDiff: A Fixed-Point Algorithm for Comparing Ontology Versions. In: The Eighteenth National Conf. Artificial Intelligence (AAAI-02), Edmonton, Alberta, Aug. 2002

10 Doan A, et al. Learning to map between ontologies on the Semantic web[A]. WWW2002, May 2002