The Ontology Evolution Method in Open Environment

# 开放环境下本体演化方法

宋英杰　刘亚清　张　斌　辛　晓／著

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

The Ontology Evolution Method in Open Environment

# 开放环境下本体演化方法

宋英杰 刘亚清 张 斌 辛 晓/著



人民邮电出版社
北京

**图书在版编目（CIP）数据**

开放环境下本体演化方法 / 宋英杰等著. -- 北京 : 人民邮电出版社, 2017.9
ISBN 978-7-115-46907-6

Ⅰ. ①开… Ⅱ. ①宋… Ⅲ. ①电子计算机－研究 Ⅳ. ①TP3

中国版本图书馆CIP数据核字(2017)第239576号

## 内 容 提 要

本体是共享概念模型明确的形式化规范说明。其重要性已在知识工程、智能信息集成、信息检索与获取、软件工程、自然语言处理、普适计算等许多方面有所体现。

本书全面概括了本体演化的一些方法，重点研究开放环境下本体演化中的本体一致性推理算法、本体演化序列内部与序列之间的冲突检测、冲突诊断算法和本体演化多版本差异检测算法，同时就所涉及的算法进行了系统的实验和比较。

本书可作为本体推理、本体演化领域开发人员和技术人员的参考书，对本体演化冲突检测具有一定的参考价值。

◆ 著　宋英杰　刘亚清　张　斌　辛　晓
责任编辑　邢建春
责任印制　彭志环

◆ 人民邮电出版社出版发行　北京市丰台区成寿寺路 11 号
邮编　100164　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京鑫丰华彩印有限公司印刷

◆ 开本：880×1230　1/32
印张：5.25　2017 年 9 月第 1 版
字数：140 千字　2017 年 9 月北京第 1 次印刷

定价：49.00 元

读者服务热线：（010）81055488　印装质量热线：（010）81055316
反盗版热线：（010）81055315

# 前言

本体演化是为适应变更而对本体做出的及时变更，并要保持变更后本体及其相关应用的一致性。在 Web 这样一个开放、动态、协作环境中显现出一些难于解决的问题，如全局知识、语义冲突和不可计算性等。演化中的冲突会导致本体的不一致和推理错误等问题。

本书将本体演化的基本概念和技术进行了概述，分析了开放环境下本体演化的研究现状，提出了目前亟待解决的问题，重点对本体非标准一致性推理算法、本体演化序列冲突检测算法和本体演化多版本差异检测算法进行了深入探讨。

本书第 1 章介绍研究背景、本体演化及其面临的问题，阐述了国内外与本体演化相关的研究现状。第 2 章阐述本体演化的基本理论，包括描述逻辑与本体的关系、OWL 语言的逻辑基础、本体推理任务、本体推理常用方法等，是本书后续方法描述的基础。第 3 章的主要工作是给出本体语言 OWL 到 Alloy 转换规则和算法，为 OWL 本体提供了更加有效的推理支持。基于一阶逻辑的 Alloy 语言及其推理工具 Alloy Analyzer 弥补了现有同类本体推理工具的不足，并利用 Alloy 语言本身对关系的强调，对很多本体推理工具不能处理的具有复杂属性的本体提供推理支持。第 4 章将 OWL 本体文件根据定义的中间语言以及中间语言与 OWL 语言中间的转换规则转

换成可满足性问题，利用 SAT 求解器对子句集进行求解确立本体的可满足性。对于找不到可满足解的情况，通过提取极小不可满足子句集来定位原因。与第 2 章中基于 Alloy 的本体推理方法相比，本章方法的优势在于支持更大规模的本体一致性检测。第 5 章分析开放环境下本体演化序列内部和演化序列之间可能存在的三种冲突。在此基础上，针对三种不同的冲突分别给出检测算法。尤其是针对变更序列之间不一致冲突，为了最大限度地满足用户的变更需求提出最大一致变更子序列获取算法。最后，针对已有的基因本体进行实验设计，验证算法效率。第 6 章提出一种基于冲突变更序列矩阵的动态冲突诊断算法，为此将具有语义冲突的变更序列对模型化为冲突变更序列矩阵，在此基础上，提出最小碰集矩阵的计算方法，更进一步，提出基于贝叶斯概率推理的碰集集合概率排序，最终确定最有可能解释冲突的原因。第 7 章提出一种基于概念格的本体版本空间表示模型，在此基础上提出并论证了有参差异检测和无参差异检测的相关算法。相比较现有的方法，基于概念格模型的本体版本差异检测算法的执行效率更高。第 8 章根据本书的理论部分设计并实现了一个支持开放环境下本体演化的原型系统。系统用仿真的方式模拟了本体演化的过程，通过对同一本体的多个变更序列的冲突检测来保证本体变更的顺利进行，并且利用本体一致性非标准推理方法反复对变更本体进行一致性检测，保证变更后本体的正确性。

本书由笔者与大连海事大学刘亚清博士等共同完成，并得到山东工商学院计算机科学与技术学院冯烟利院长、窦全胜教授的大力支持，在此表示衷心的感谢!

同时，本书的研究工作得到了山东省高校智能信息处理重点实验室（山东工商学院）的资助，特此表示感谢。

本书描述的非标准算法、变更序列冲突检测算法及多版本本体差异检测算法是新的尝试，不足之处尚有很多，在实际应用中还存在大量的问题需要解决，在不同领域本体中的应用还有待检验。由于基础薄弱，希望能得到更多学者的帮助。

宋英杰

2017 年 8 月 15 日于山东工商学院

# 目 录

# 第1章 绪 论

## 1.1 背景知识

哲学上本体论（Ontology）研究存在的本质，而 Quine[1]的 Ontological Commitment 把本体研究推动到概念世界的形式理论。1993 年，计算机科学家 Gruber[2]扩展了 Quine 的工作，把本体视为概念模型的明确规范说明，其中“概念模型”是指通过抽象客观世界中一些现象的相关概念而得到的结构化模型；“明确”强调使用的概念及其约束和语境是有明确定义的。随后 Studer[3]扩展了 Gruber 的本体概念，进一步认为本体是共享概念模型明确的形式化的规范说明。这里“形式化”是指本体是计算机能处理的，而“共享”强调了概念（包括实体、属性和过程）、定义和关系是领域、社会范畴内共同认可的知识。与隐式地使用知识不同，显式地使用本体使领域知识的重用变为可能。目前，在计算机科学和信息科学领域，本体是描述领域概念及其关系的知识表示方法，支持知识推理和验证，是知识库的基础性扩展，其重要性已在知识工程、智能信息集成、信息检索与获取、软件工程、自然语言处理、普适计算等许多方面表现出来[4]。

随着语义 Web 的深入研究，本体作为语义 Web 的知识载体，也越来越受到人们的重视。本体使语义 Web 上的知识呈现结构化（如图 1-1 所示）表示形式。可以看出，唯一资源标识（URI，Uniform Resource Identifier）以及 Unicode 技术构成了语义网的最底层基础。使用 XML 来定义定制的标签格式以及用 RDF 来灵活地表达数据，XML 结构化文档的表层语法对文档没有任何语义约束；RDF 描述对象（或者资源）以及它们之间关系的数据模型，为数据模型提供了简单的语义，这些数据模型能够用 XML 语法进行表达。下一步需要的是用一种 Ontology 网络语言（如 OWL）来描述网络文档中术语的明确含义和它们之间的关系。作为语义网（Semantic Web）体系结构的重要组成部分，本体给数据赋予了明确的含义，在人类多种语言的环境下能够通过计算机形成一种通用的信息交换形式，使机器能够自动地处理和集成网上可用的信息。

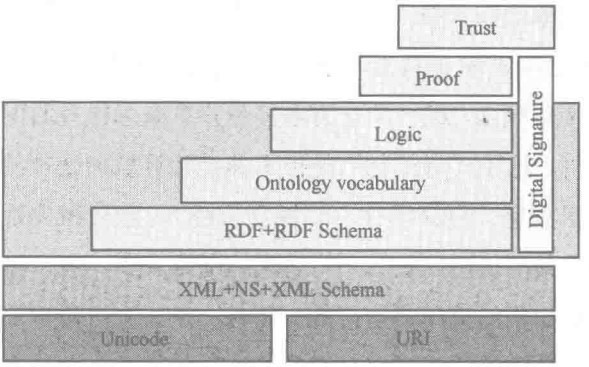


图 1-1　语义 Web 系统结构

随着本体数量的不断增加和语义 Web 的发展，如何修改现有本体以适应新知识和变化的需求变得十分重要。本体的存在不是一成

不变的。开放环境中领域的变化[5]、共享概念模型的变化、需求的变化、新知识的引入、开放动态的外部环境、表示的变化和本体集成等都可能导致本体演化[6,7]。本体数量的增加和适应变更需求的昂贵代价使本体演化变得愈发重要，本体演化研究具有深层次的研究意义和重要的应用前景。

近年来，本体演化的研究主要集中于研究集中式环境下的本体演化：变化发现[8~11]、变化传播[12~14]、版本控制[15]、演化管理[10,16]、一致性推理[17~19]和本体演化工具[20~22]等。在集中式环境的本体演化中，不同的用户可以通过协商进行本体修改以满足本体演化的需求，但集中式的本体构建和演化模式无法满足网络大规模、海量数据的需求，存在一系列的不足，如集中式演化成本高，且不能灵活地随着领域和用户需求的变化而及时做出反应。更进一步，这种开发方式，缺少开发者和用户之间的良好沟通，会导致用户在使用本体的过程中缺乏对本体的理解而使认知能力受到约束。随着本体发展和规模不断扩大，本体的构建、管理和演化已经不是一个人或几个人的团队能够胜任的工作，越来越多不同地域的本体管理者和领域专家参与到本体的开发和演化工作中。开放式协同模式成为网络本体发展的必然趋势[23]。

开放环境下（如语义 Web）本体演化表现出以下三方面特点。（1）分布性，现在的本体开发以非集中式开发为主，是分工协作的过程。所以相互关联的、存在于不同地域本体之间的依赖性需要重点考虑。（2）复杂性，本体是用于表示现实世界的模型，本体中的关系错综复杂，每个概念或者属性的变化都会影响相关的整个区域，即便是影响范围小的变更操作也会因为积累而变得难以处理。

（3）依赖性，本体经常存在重用的情况，相互依赖的本体之间的变化是相关的。显然，开放环境下随着本体规模的增大和本体间依赖关系的增多，也使本体演化愈加复杂。

开放式环境下本体演化使大规模本体的变更需求得到满足，与此同时，开放式环境下的本体演化更要关注不同的参与者之间由于知识结构和角色等不同而产生的冲突。并且本体修改后可能会导致“不可满足概念”“不一致”等这样的不相容问题。严重的话，副作用还会传播到依存的数据、应用、智能体（Agent）以及其他本体。语义 Web 分布、多源、异构、开放、协同和非集权等特点[24]使现有方法显露出一些难于解决的问题，如全局知识、副作用、不可计算性、公平性、合理性等。

本体演化研究有待分析开放动态、不确定、协同环境下不同用户之间的冲突发现、冲突解决以及本体一致性推理等，有待扩展支持本体修改、本体诊断和本体错误定位的演化模式，以此建立理性公平的演化方法。为此，本书围绕本体一致性推理，开放式环境下的本体演化序列的冲突发现、冲突解决以及多版本本体差异检测等方面的内容展开，建立既能满足演化的需求、解决分布式开发者之间存在的冲突，又要保证演化后本体一致性的本体演化方法。

## 1.2 国内外相关研究

### 1.2.1 本体演化及其过程

#### 1.2.1.1 本体演化的原因

文献[11]认同的本体发生变化的原因有以下几个。（1）一个本

体就像任意一个包含领域关注点的信息结构体一样，相关信息可能会简单地由于关注点发生变化而改变。即使不现实地假设存在一个一成不变的领域，有时也需要改变观察领域的视角，视角变了，本体内容自然不同。(2) 另外一个可能性是发现忽视了用户的某个隐性需要，原始领域概念模型中包含了一个设计错误，需要根据用户的需求综合更多的功能，于是需要改正本体错误，扩充本体内容。(3) 类似地，像不相容或语义不连贯这样的矛盾可能暴露出来，这种情况下也会采用行动应对这个矛盾。(4) 更进一步，以前是未知的、未分类的新信息，现在变成可用的了，或者是该领域的不同特性变得已知和重要了。此外，本体变化的原因还与本体开发和语义网的特性有关。

本体开发是一个复杂的系统工程，过程上需要本体工程师分工协作，多个部分本体组合在一起而形成最终的本体。这个过程中需要对子本体进行修改以达到一个相容的正确的结果状态。即便如此，所谓的“最终”状态也很少是最终结果，因为本体开发通常是一个持续演进的过程。

语义网的特性也决定了本体动态变化的特点。语义网上的本体通常依赖于其他本体，但本体工程师可能对它们根本无法完全控制。如果由于某些原因，这些“远程”本体被修改了，依存本体也就需要修改，以便体现术语和表达方面的变化。其他变化的情形还包括：某个智能体、服务或应用可能需要使用一个本体，它的术语或表示与它所理解的不同，这时就需要对引入本体进行某种转换（变化）才能使用；再就是也许需要从两个或更多本体中综合信息以便生成一个更适用于某个应用的本体。

#### 1.2.1.2 本体演化的定义

本体演化[12,25]是为适应变化而对一个本体做出的及时修正，要求是保持结构上的相容性、用户自定义的相容性和逻辑一致性。一个变化可能导致本体其他部分的不一致，还会影响到相关的智能体、服务和应用。所以说，本体演化是一系列一致性传播过程，它包含技术上和管理上的多种活动[26]，这些活动的目的是有效地保证修改后的本体依旧能够满足结构上一致性和用户的需求。

#### 1.2.1.3 本体演化与数据库模式演化

数据库中的模式演化一直是热点的研究课题[27~30]，本体演化和数据库的模式演化有密切的联系[27]，尤其是面向对象的数据库[31,32]。表 1-1 对比了数据库模式和本体在实例、查询上的不同特点，二者的演化条件有所不同，但在变化传播方面，数据库模式变化影响的控制要么是通过维持相容性规则完成，要么是通过基于公理的推理机制完成[33]，而本体变化要将这两种方法结合起来使用。

表 1-1 数据库模式演化与本体演化

| 数据库模式 | 本体 |
| --- | --- |
| • 数据库模式是数据的模型，一定有具体的实例 | • 本体可以没有具体的实例 |
| • 查询的结果是实例的集合，不包含模式本身 | • 查询结果可能也包含本体本身，如查询概念的子概念 |
| • 数据库模式演化是集中式开发过程[34,35] | • 本体演化大部分是非集中式的协同过程 |
| • 演化在数据库系统的生命周期中频繁发生<br>• 演化的管理不是很规范；数据库管理者必须进行维持数据库的一致性<br>• 手动修改实例 | • 用户获取了变更需求可能大幅度增加变更的频度<br>• 本体工程师对变化的检测和管理进行控制<br>• 自动变化并且将变化影响通知给本体工程师 |

（续表）

| 数据库模式 | 本体 |
| --- | --- |
| • 将变化传播给实例的方法有三种[12]：①数据迁移和使实例适应于模式；②模式-实例同步机制；③整合以上两种方法 | • 本体变化的传播有两种完全不同的情况：①基于数据迁移的集中式方式；②基于延迟变化的分散式方式。两种方式的选择依赖于两个标准：全局一致性和运行时间 |
| • 变化传播局限到实例 | • 变化传播可能扩展到所有依赖的本体和应用等 |

相对于关系数据库，本体演化更接近面向对象数据库的模式演化，表 1-2 对比了两者的不同之处，需要指出的是有几个本体演化改编自面向对象数据库模式演化的方法，但两者模型之间的区别使本体演化方法不仅是对已有方法的改写，而且是一种扩充[36]。

表 1-2 面向对象的数据库模式演化与本体演化

| 面向对象的数据库模式 | 本体 |
| --- | --- |
| • 面向对象的数据库模式演化反映了数据和代码的结构，并且还考虑了对象的行为（包含在模型中的方法）。另外，它还整合了数据的物理表示（整数、实数等） | • 本体通过概念、关系和它们之间的约束反映特定领域的机构 |
| • 实例和类不在同一个层次 | • 术语层（概念和属性）和断言层（实例）没有明显的界线。观念和实例在本体查询中可以一起使用[33] |
| • 模式的整合是不合理的 | • 本体重用时可以整合本体或者本体的一部分，因此，一个变化可以是从一个本体包含或者删除另外一个本体[12] |
| • 变化定义与模型本身 | • 除了表示本体模型本身的变化，还需为验证定义变化语义，并且为保持本体一致性进行额外操作，结构的变化会导致本体不一致，解决的策略也相当复杂 |
| • 演化需定义一致性模型和具体的变化应用规范 | • 本体包含更加丰富的语义，一致性模型的定义也就包含更多的约束。另外，应用变化的可能性也更大。（例如，本体中增加一个实例时不一定要实例化一个概念，或者可以增加一个不依赖于概念的属性） |
| • 数据库中的模式语义不足以用于一致性验证的推理机制 | • 本体语义丰富，可以用于检测不一致的推理机制 |