

本体演化研究进展^①

刘金花, 张友华, 李绍稳, 辜丽川, 朱 诚, 朱 麟

(安徽农业大学 信息与计算机学院, 合肥 230036)

摘 要: 知识无时无刻不在变化着, 为了使本体能够吸收和融合新的知识, 就需要对本体中的知识进行及时的更正, 这就涉及到了本体演化的问题。从本体演化的定义、原因、过程、技术、工具等几个方面对本体演化进行一次概述, 重点介绍本体演化过程、关键技术以及支持本体演化的主要工具, 为本体演化的进一步研究奠定理论和技术基础。

关键词: 本体; 本体演化; 知识融合; 本体演化技术; 本体演化工具

Research Progress of Ontology Evolution

LIU Jin-Hua, ZHANG You-Hua, LI Shao-Wen, GU Li-Chuan, ZHU Cheng, ZHU Lin

(School of Information and Computer Science, Anhui Agricultural University, Hefei 230036, China)

Abstract: The knowledge is constantly changing, so it needs a timely correction in order to absorb and integrate new knowledge, which involves the evolution of the ontology. The paper summarized from definition, cause, procedure, technology and evolution tools, and summed up the research of ontology evolution. The procedure of ontology evolution, the important technology at present and main evolution tools are mainly introduced and lay theory and technology foundation for further research on ontology evolution.

Key words: ontology; ontology evolution; knowledge fusion; ontology evolution technology; ontology evolution tools

1 引言

随着语义 Web 的提出, 对于本体的研究越来越受到关注。通过国内外学者对本体定义的归纳, 可以认为本体是关于某些主题的、层次清晰的规范说明, 作为语义 Web 的基础, 本体支持数据、信息和知识的交换、重用与共享; 本体对于知识工程、自然语言处理、信息检索、信息集成、信息交换、知识管理和软件工程等领域而言, 在词法和语义上都比数据库所能表示的信息丰富的多, 并且它不单单是一个存放数据的结构, 而是定位于提供一个领域严谨丰富的理论, 描述事物或概念的运动和变化。

目前, 对本体的研究包括多个方面, 如本体建模、本体学习、本体融合、本体映射等, 其中本体演化的研究也越来越成为一个热点。本体演化即进化, 就是本体的适时调整 and 变化一致性的传播^[1]。在现实社会中

知识并不是一成不变的, 知识的变化就要求本体也要进行及时的变更以满足用户对其需求, 通过对本体演化的研究找出能很好地对本体进行更新和完善的理论和方法。因此对本体演化进行研究就显得极为重要。

2 本体演化理论

本体演化的目的, 就是为了使得本体更好的满足用户的需求, 而对本体做出的适当变化。有多种原因会导致本体的演化, 主要包括如下几个方面^[2]: (1) 领域的变化。领域的改变非常普遍。例如, 当有新的茶树虫害知识添加入茶树虫害本体当中时, 描述该领域的本体就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而引起的, 即上下文语义的变化。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。

① 基金项目: 国家自然科学基金(30971691, 30800663); “十一五” 国家科技支撑计划(2009BAD4B02, 2009BADA6B02)

收稿时间: 2010-11-04; 收到修改稿时间: 2010-11-28

例如,某一本体中“Class”用于表示“开设的课程”,如“Class(alg-Fall-2009)”;当变成表示“一堂课”时,如“Class(Math)”时,就会发生错误。(3)表示的变化。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。

本体演化是一个非常繁琐和复杂的过程,到目前为止还没有形成一套规范的理论和技术体系,也没有一个明确的划分,只要能够解决本体演化问题就是可行的本体演化过程,目前广为接受的本体演化过程,据文献[1]的建议,可分为以下过程:

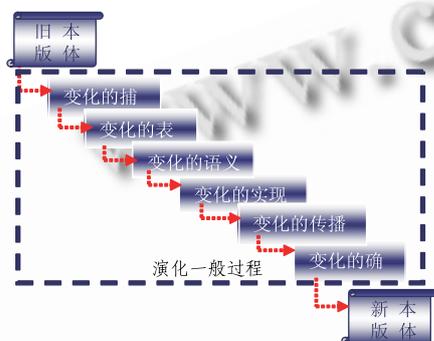


图1 本体演化一般过程

(1) 变化的捕获

主要是发现本体中的变化,我们可以从明确的需求中和从变化发现方法的结果中捕获变化。从需求中捕获变化的方法是自顶向下的,否则是自底向上的。变化的发现可以分为结构驱动的变化发现、用户驱动的变化发现和数据驱动的变化发现这三种。一般在用户修改本体时记录下对本体的变化操作,生成变化日志。

(2) 变化的表示

为了处理变化,需要提供一种合适的形式来标志和表示变化。对一个给定的本体,需要为它定义一种变化的表示方法,一般通过一种规范的语言提供词汇和语法来准确的表达变化。例如,将类C1的子类C2提升为C1的父类C3的子类,我们可以定义为RemoveConcept(C2,C1), AddConcept(C2,C3)。通过这种变化的表示,我们可以知道本体到底要做怎样的变化。

(3) 变化的语义

该过程涉及到变化对本体自身产生的影响,用于检查和维护变化后本体的一致性。本体的一致性包含逻辑一致性、语义一致性以及用户自定义一致性。逻辑一致性指本体的变化要符合本体模型的语法定义,一个本体和它的模型是一致的当且仅当它保持了定义该本体模型的约束^[3];语义一致性指本体的变化在语义上要保持一致,即与本体的自然语言表示的意思保持一致;用户自定义一致性是指用户自己定义的关于本体一致性的约束,比如在本体中不能存在闭环等。

(4) 变化的实现

通知本体工程师发生了一个变化请求以及该请求将产生的所有后果,应用该请求,对变化进行跟踪记录。为了避免不希望的变化,需要归纳出对本体以及基于本体的其他部分的所有可能的变化,由本体工程师来接受或取消这些改变。

(5) 变化的传播

本体通常重用和继承其他的本体,一个本体的变化可能会破坏其他依赖于它的本体和应用程序。变化的传播阶段任务就是在一个本体发生变化后确保和维护和它相关的本体或应用程序的一致性。

(6) 变化的确认

此阶段是对上面本体变化的确认,通过领域专家对本体变化的肯定之后,确认对本体的修改。

值得注意的是,以上所说的本体演化过程通常都需要领域专家的参与,如果领域专家的见解不同,很可能对本体修改意见产生分歧。

3 本体演化技术

从本体演化的过程来看,我们可以将本体演化的技术分为三个部分^[4]:变化发现技术、变化处理技术和变化传播技术。这三个部分分别对应着本体不同的生命周期,所采用的技术和目的也不相同。其中,变化发现管理主要采用一定的技术对本体的不一致性进行侦测,发现不一致的根源并找到本体所需要进行的变化;变化处理主要采用本体进化技术,对本体中需要修改的部分进行修改,并维护本体的一致性;变化结果传播则主要处理本体变化在依赖于本体的应用中的传播,使得相关应用能够不受本体变化的影响。

3.1 变化发现技术

在文献[1]中提出了三种类型的变化发现技术:a)

结构驱动的变化发现；这一变化发现是基于本体工程师的决策知识之上的，在本体结构分析上 L. Stojanovic 等人采用了一组启发式方法集来完善本体，这组启发式方法集是通过对本体中概念的修改操作进行定义以达到发现需要变化的本体结构的目的。b) 用户驱动的变化发现^[5]；这一变化发现是指在用户使用或对本体进行调试的时候发现本体中的变化。Silvana Castano, Alfio Ferrara 和 Guillermo N. Hess 提出一种发现驱动的本体演化方法^[6]。该方法通过以多媒体本体演化案例为关注重点，通过探测多媒体资源和当前版本本体所包含的有用信息和知识来发现新的相关概念，从而展示了本体匹配技术是如何用来提高新的相关概念的发现。c) 数据驱动的变化发现^[7]。这一变化发现是通过考虑本体的实例来精化本体的。注意的是，这种变化发现是不能通过基于描述逻辑 (DL) 的本体语言，例如 OWL 的本体演化系统来实现的，因为 DL 系统依附于开放世界的假设。数据驱动的变化发现是封闭世界的假设，许多符合逻辑的实例是通过推理得知的，本身就是不明确的。这种变化发现可以使用数据挖掘和形式化概念分析技术来实现。

3.2 变化处理技术

变化处理是整个本体演化过程中最重要的环节，该阶段主要是对本体的变化做出修改，以及对本体产生的边际效应进行处理，从而使整个本体达到一致性。在变化处理阶段最主要的是本体演化方法的使用，不同的演化方法的使用会产生不同的效果，文献[4]中给出了四种本体演化方法，分别是基于进化策略的本体演化方法、基于事务的本体演化方法、基于推理的本体演化方法和基于信念修正的本体演化方法。但在实际应用中本体演化处理技术是多种多样的，只要能够达到理想的效果不管用什么方法都是可行的。

Muhammad Javed, Yalemisew M. Abgaz 和 Claus Pahl 提出一种基于变化操作框架的本体演化模式^[8]。变化操作是本体演化的基石。该方法将变化操作分层处理，将变化表示成基于分层操作框架的模式，根据变化粒度、变化领域特殊性和变化的抽象性确定四个不同层次的变化操作。前两个层次是基于通用结构的变化操作，后两个层次是基于领域特殊性的变化模式。这些变化模式层次捕获选定领域的真正变化并进行处理。

Natalya F. Noy 等人提出一种单一的统一本体演

化框架^[9]，这个框架能灵活地运用于同步或异步模式，在这些同步或异步模式下，当没有对变化步骤作出记录的时候，用 Protégé 和插件来跟踪变化。该系统的核心是变化和注释本体 (Change and Annotation Ontology: CHAO)，它用实例来记录具体变化和一些原信息(包括：作者、时间戳、注释、接受状态等)。当 Protégé 和变化管理插件被用于本体编辑时，CHAO 实例就会成为编辑过程的副产物而被创建，否则，CHAO 则是通过对两个不同版本本体结构进行比较而产生。然后，通过 CHAO 实例的驱动用户界面将版本间的变化呈现给用户，允许用户接受和拒绝变化、查看历史概念、生成一个新的基准或发布一个其他应用能够使用的历史变化等等。用户可以通过这个框架在不同的模式之间进行切换。

鲍爱华、姚莉和张维明提出一种基于变化生成图的 OWL 本体的协同演化方法^[10]。引入了一种有向的与或图——变化生成图来表示知识工程师的本体改进意图，通过发现变化生成图之间的冲突来找出多个知识工程师本体改进意图之间潜在的矛盾，并通过变化生成图的冲突消解来解决本体改进意图之间的矛盾，使得多个本体改进意图能够并存，从而实现本体协同进化的目标。

刘晨、韩燕波等提出 MINI 本体演化算法^[11]，这种演化算法可减小本体变更的影响范围。该算法首先分析了本体实体和服务之间的依赖关系并提出了量化变更影响范围的数学公式。根据这一公式，MINI 算法将本体演化过程转变为图的启发式搜索过程，通过搜索一条影响值最小的变更路径来减小本体演化的影响范围。实验结果表明，MINI 算法导致的平均变更影响范围大大小于现有的本体演化算法。该算法已在某实际项目中得以应用和验证。

宋奂寰、张志祥提出基于概念代数的本体演化方法^[12]。该方法提出基于概念代数的本体演化 3 层结构，包括原子概念层、原子概念关系层、应用层。分析概念代数与本体表示语言 OWL 之间的对应关系。在应用层次中，利用概念代数根据应用的需要在原本体模型基础上构建面向应用的概念，并维护原有本体模型。

鲍爱华、张维明等提出一种基于规则的 OWL-S 本体语法一致性维护方法^[13]。该方法首先定义抽象出 OWL-S 本体基本变化，提出变化关联矩阵分析基本变

化间的依赖关系,基于关联矩阵提出额外变化序列生成算法分析基本变化对全局的影响。

3.3 变化传播技术

本体是语义 web 的基础,很多应用和本体都依赖于其他本体,因此在对一个本体进行修改的同时也要将这种变化传播到其他的依赖本体以及应用中去。在实际的应用中,当某个本体进行变化的时候,很难及时的将变化传播到其他依赖本体及应用中去,这就需要不同的本体版本具有兼容的访问接口,以适应用户对本体的一致性访问。

当然,在研究的过程中也有从本体演化的整个过程来考虑的,何扬帆、何克清就提出一种支持可靠语义互操作的本体演化管理框架 MFI-3^[14]。该框架的基本思想是:由标准组织发布参考本体;应用系统在参考本体的基础上,根据自身要求对参考本体实施一定的变换,得到本地本体;建立本体变化的类型系统;建立本体演化的约束系统;根据本体的基本结构,类型系统和约束系统详细记录变换过程的相关信息;根据本体的演化信息,以及演化传播策略对依赖于它的构建进行相应的演化;在本体注册信息和演化信息的基础上得到可靠的本体映射。

4 支持本体演化的工具

目前有很多工具和平台都在一定程度上支持本体的演化,例如:Protégé、KAON、OntoLingua、OilEd、OntoEdit 等,下面作简要介绍。

Protégé是一个由斯坦福大学开发的基于 Java 语言的开源本体编辑工具,用来帮助本体开发人员和领域专家对领域知识进行建模,使构建本体知识库的过程易于操作和管理,降低本体构建的高昂成本和维护代价。Protégé有 GUI 界面和 API 接口两种应用形式,Protégé-OWL API 支持所有 OWL 建模原语,并提供语法检查、一致性检测、推理机调用等。

KAON 是德国 Karlsruhe 大学编制的一套用于语义网和本体研究的工具。KAON API 提供了本体的编辑和储存机制,支持本体的演化、本体模块化、本体的访问和处理程序。KAON API 提供了 KAON 本体程序使用界面,包含概念(Concept)、属性(Property)和实例(Instance)等种类。API 结构反映了 KAON 语言的性能。根据使用者的需求,通过添加树形结构策略,来实现本体变化管理。例如,从本体中删除一个概念,

则 API 自动删除其下位概念,也可以将下位概念接入到上位概念中。在实际对本体操作应用之前,使用者对每步操作都可以预先观测执行以后的结果^[15]。

OntoLingua 是 1995 年 2 月由斯坦福知识系统人工智能实验室的网络服务中心(Stanford Knowledge Systems, AI Laboratory(KSL)Network Services)开发并提供使用,该本体编辑工具提供一种分布式协作的环境,使用面向对象的框架视图表示和浏览知识,浏览器使用超链接,可以方便地从一个术语转到另一个术语,它使用类/子类的方式展现类层次,以便浏览、构建、编辑、修改和使用本体。

OilEd 由英国 Manchester 大学开发,是创建和编辑 OIL 本体的图形工具,这个工具结合了框架表示和描述逻辑表示两者的长处。OilEd 的初创意图是提供一个简单的编辑工具来示范 OIL 语言的使用,并激发人们对 OIL 语言的兴趣。其最新版本没有提供充足的本体开发环境,不具备合作开发的能力,不支持大规模的本体开发、综合、立论以及其他很多有关本体构建的活动。OilEd 提供源代码,是本体编辑工具的“记事本”,为用户建造本体提供足够的功能,并示范应如何利用事实推理来检查本体的一致性。

OntoEdit 由德国 Ontoprise 公司开发,所用的软件包是“Java Standard Application”。OntoEdit 是一种通过使用图形工具来支持本体开发和维护的本体工程环境。它将本体开发方法论(骨架法)与合作开发和推理功能相结合,分层构建本体,系统的基本公理有不相交的概念、对称性关系和传递性关系。OntoEdit 可以通过插入分析器进行一致性检验,即考察本体的句法、关系以及逻辑正确性能被自动地检验的程度。

5 结语

目前,国内外研究者对本体演化已有了很多新的贡献,诸如有减少演化代价的,有提高执行效率的,有从变化的发现来考虑的,有从变化的表示来考虑的,还有从变化传播方面来考虑的等。这些方法都是从本体演化的共性问题来考虑的,缺少特定领域本体演化的研究,因此特定领域本体演化研究可以作为一个新的研究项目。

参考文献

- 1 Stojanovic L. Methods and tools for ontology evolution.

Karlsruhe: University of Karlsruhe, 2004.

- 2 张子振,王存刚.本体进化关键技术研究.科技信息(学术版),2008,(8):202-204.
- 3 Sure Y, Tempich C. State of the art in ontology engineering methodologies. Karlsruhe: Semantic Knowledge Technologies (SEKT), 2004.
- 4 鲍爱华,姚莉,刘芳,张维明.本体变化管理技术研究综述.计算机科学,2007,34(9):151-155.
- 5 Stojanovic L, Maedche A, Stojanovic N, Motik B. User-driven ontology evolution management. Benjamins R V, eds. Proc. of the 13th European Conference on Knowledge Engineering and Knowledge Management. Madrid, Spain: Springer-Verlag, 2002. 285-300.
- 6 Castano S, Ferrara A, Hess NG. Discovery-driven ontology evolution. Pisa: Scuola Normale Superiore, 2006.
- 7 Plessers P, Troyer O D. Ontology change detection using a versioning log. Yolanda Gi, eds. Proc. of the 4th International Semantic Web Conference. Ireland: Springer-Verlag, 2005. 578-592.
- 8 Javed M, Yalemisew M, Abgaz, Pahl C. A pattern-based framework of change operators for ontology evolution. In: Shvaiko P, eds. The 4th International Workshop on Ontology Content. Algarve: OTM Workshops, 2009. 544-553.
- 9 Musen AM, Chugh A, Liu W, Noy FN. A framework for ontology evolution in collaborative environments. In: Isabel F, eds. Proc. of the 5th International Semantic Web Conference. Heidelberg: Springer-Verlag, 2006.544-558.
- 10 鲍爱华,姚莉,张维明.基于变化生成图的 OWL 本体协同进化方法研究.计算机科学,2007,34(3):186-191.
- 11 刘晨,韩燕波,陈旺虎,王建武.MINI——一种可减小变更影响范围的本体演化算法.计算机学报,2008,31(5):711-720.
- 12 宋夙寰,张志祥.基于概念代数的本体演化.计算机工程,2009,35(13):23-25.
- 13 鲍爱华,张维明,袁金平,姚莉.一种基于规则的 OWL_S 本体语法一致性维护方法.国防科技大学学报,2009,31(3):97-103.
- 14 何扬帆,何克清.一种支持可靠语义互操作的本体演化管理框架.计算机工程,2007,33(18):26-30.
- 15 常春.大型 Ontology 构建工具 KAON 的使用和评价.现代图书情报技术,2004,20(8):14-17.

(上接第 194 页)

顶点序列所构成的轨迹平面将变得更加平滑,如图 10 所示。

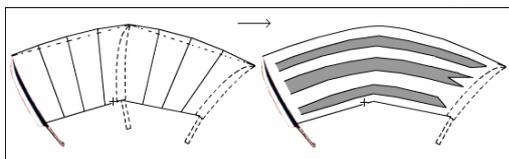


图 10 改进算法后的效果

以上便是“拖影”特效的全部算法,这种算法的计算量不大,但实际的显示效果却十分理想。

参考文献

- 1 Lake A. Game Programming Gems 8. America: Course Technology PT, 2010: 140-203.
- 2 Jacobs S. Game Programming Gems 7. America: Charles River Media, 2008: 255-336.
- 3 Walsh P. Advanced 3D Game Programming with DirectX 9.0. Texas: Wordware Publishing, 2003: 413-481.
- 4 黄蓝泉.游戏引擎中特效的实时渲染技术研究[硕士学位论文].成都:电子科技大学,2008.
- 5 赵明.基于 DirectX 的三维游戏特效技术的研究与实现[硕士学位论文].哈尔滨:哈尔滨工程大学,2009.