

# 大规模本体协同构建框架研究与设计\*

李亚子<sup>1</sup> 钱庆<sup>1</sup> 郭文丽<sup>2</sup> 方安<sup>1</sup> 洪娜<sup>1</sup> 王军辉<sup>1</sup>

中国医学科学院医学信息研究所 北京 100020 <sup>2</sup>北京邮电大学图书馆 北京 100876

(摘要) 阐述本体协同构建研究的热点,分析大规模分布式环境下协同构建本体的用户类型,总结必要的操作及功能,提出分布式加工、集中式存储以及模块式开发的 DCM 框架,并对框架中的元素和功能进行阐述。

(关键词) 协同式本体开发 模块式开发 分布式加工 集中式存储

(分类号) G350

## Research and Design of Collaborative Construction Framework for Large Scale Ontology

Li Yazhi<sup>1</sup> Qian Qing<sup>1</sup> Guo Wenli<sup>2</sup> Fang An<sup>1</sup> Hong Na<sup>1</sup> Wang Junhui<sup>1</sup>

Institute of Medical Information, Chinese Academy of Medical Sciences, Beijing 100020

<sup>2</sup>Beijing University of Posts and Telecommunications Library, Beijing 100876

(Abstract) States the topics of collaborative ontology development, analyzes the types of users which refer to develop ontology in the large-scale distributed environment, and concludes requirements and functions, proposes DCM framework representing distributed implementation, and module development. Finally details the elements and the function of the framework.

(Keywords) collaborative ontology development module development distributed implementation central storage

目前语义网(Semantic Web)是知识工程发展的主要方向,本体是实现语义网的主要途径,本体的设计和维护是一项复杂的社会活动,一方面需要工具支持本体的设计和重用等典型操作,例如叙词表、词表、草根分类法、数据库模式以及文档内知识的重用,需要合适的评估和选择工作,使得本体的功能符合特定的任务;另外,需要支持本体构建和重用的协同工作。大规模本体的开发涉及到领域专家、本体工程师和最终用户等角色,在构建本体的各个环节中每个人都有自己的目标,每个角色从不同的维度思考本体包含的概念,并发地创建概念,已涉及到不同用户对本体的可控性,本体内内容创建的动态性、开发环境的分布性等特点。

目前,对单个本体的创建、发布、浏览、编辑和存储等方法的研究较为成熟,但分布式环境下本体的协同构建聚焦在具体的创建环节,例如协同创建本体的冲突检测,不同用户的视图管理,本体版本的演化等,从已有的研究来看,对大规模本体协同构建的工作模型与框架的系统阐述尚有不足。

## 1 本体协同构建相关研究

本体协同构建既要包括单个本体创建、发布、编辑、检索、浏览与重用等相关的所有功能,又要解决开放环境中不同领域、不同层次、不同角色用户的协同工作问题,支持概念的一致性理解、本体的动态演变、本体认识中不同视角的统一,支持多个团队有效的沟通,使得工作流程中各个环节无缝衔接,能够高效地创建和重用本体。

Tempich 等<sup>[1]</sup>在 DILIGENT 项目中提出协同开发本体的方法,并将 DILIGENT 方法分为 4 个步骤:①参与项目的组选择一个缓和的模式,将讨论和组织的决策结构化;②对讨论的论点达成一致的决策过程有统一的认识;③组织别出决策的论点,并针对论点进行讨论,提出解决方法;④分析讨论的结果,同意最终达成的解决方案,或者推迟解决方案的产生。Yewang Chen 等从冲突检测的角度出发,支持本体的协同开发<sup>[2]</sup>,将冲突分为三类:协同冲突、一致性冲突以及不规则冲突,并提出 CPP(Command PackagePool)算法支持冲突

\* 本文系国家科技支撑计划项目“科技知识组织体系的协同工作系统和辅助工具开发”(项目编号:2011BAH10B02)和医科院信息所基本科研业务费专项项目“国家级新型农村合作医疗信息系统建设关键问题研究”(项目编号:11R0101)研究成果之一。

收稿日期:2010-12-28 修回日期:2011-04-02 本文起止页码:96-100 本文责任编辑:杜杏叶

的自动检测。Ceccaroni 等人提出 Visual Ontology Modeler 模型<sup>[3]</sup>,图形化支持本体的协同构建,可满足大规模本体协同开发和管理的需要,易于使用与理解、图形化展示内容,具有可配置能力,集成了版本控制、一致性和完备性检查等功能。Jiménez-Ruiz 和 Berlanga 领域专家、本体工程师和最终用户对本体的认识维度不同,提出基于视图的方法<sup>[4]</sup>,识别大规模本体在演化、使用过程中设计到的必要因素,支持大规模本体分布、协同和模块化的开发。Aldo Gangemi 等构建了协同本体设计模型(Collaborative Ontology-Design Ontology, C-ODO)<sup>[5]</sup>,通过构建 C-ODO 本体开发过程中的关键操作和概念,重点解决两方面问题:

- 需要工具支持典型的操作,例如现有本体和设计模式的重用、叙词表、词表、草根分类法、数据库模式以及文档内知识的重用,需要核实、评估和选择步骤,使得本体的功能符合特定任务。

- 需要工具支持这些动作的协同执行,提供关于本体元素和基本原理的讨论和一致性意见的达成。C-ODO 曾成功运用于 NeOn 项目中<sup>①</sup>。J. Bao 和 Hona-var 提出 Wiki@nt 作为协同构建本体的开发环境<sup>[6]</sup>,支持协同本体开发。Wiki@nt 基于 OSHOQP(D),它是利用 O (partial order on axioms) 和 P (localized axioms in package) 对 SHOQ(D) 扩充,支持在多个语义异构、不一致环境下单独开发的模块化本体的整合,致力于冲突的解决。Sung-Kooc Lim 提出基于 wiki 的本体协同构建环境<sup>[7]</sup>,领域专家可以协同、便捷地组织领域知识,通过定义模板,实现本体的关联,使用户能够在基于 wili 的界面准备、创建知识组件,并以基于本体的模型去存储他们。Natalya F. Noy 等总结了本体协同开发环境所具备的必要元素,包括:支持讨论和达成共识的工具集、数据起源信息、确立可信赖和可行性的途径、不同(经验)层次的用户具有不同的视图、访问的控制、用户角色的管理、工作流的支持等<sup>[8]</sup>。G. Correndo 调研了支持本体协同开发的工具<sup>[9]</sup>,如: Ontolingua、BibSonomy、DBin Hozo、OntoWiki、Collaborative Protege 以及 SOBOLEO 等,工具有的是基于 B/S 结构,有的是基于 C/S 结构,支持的功能各异,例如有的支持分类表的构建,有的支持知识库的创建,有的擅长于本体建模等。因为提供功能方向不同,所以差别很大,如标记 (tagging)、投票 (voting)、讨论 (discuss)、变化控制 (change control) 和可视化 (visualization) 等。

从以上分析来看已有本体协同构建系统、模型或框架各自偏重不同,设计大规模本体协同构建框架的

前提是要分析参与构建本体的用户类型,以及各自的角色,设计(分布式)本体的必要需求,不同角色在构建本体过程中的作用及流程。本文首先分析本体协同构建过程中涉及到的用户类型和分布式环境下开发本体所需的必要元素,最后构建本体协同构建框架。

## 2 大规模本体协同构建模型的设计

### 2.1 用户类型

大规模本体的协同构建涉及到四种类型用户:管理员、本体工程师、领域工程师以及终端用户<sup>[10]</sup>,四种类型的用户统称为系统用户。下面分别阐述各自的角色。

2.1.1 管理员 管理员控制协同加工的整个过程,注册加入到协同开发中的每个人,并赋予不同的特征。考虑本体的政策、讨论、标注、变化以及与协同相关的所有活动,不能自己做出决策,必须在群组中达成一致,如果群组没有彼此达成一致,或持续讨论,他必须定制讨论,宣布无法得出决策,因为意见分歧。反之,如果达成一致了,管理员将宣布决定。如果需要通过投票达成一致,要求协同工作人员对现存的不同意见进行投票。这种决策制定的过程的优点有:每个参与者都能对决策贡献自己的力量,新的参与者能够对已有决策进行评论。如果其他参与者破坏了工程的规则和既定政策,还要关注其他角色的活动。另外一个角色是将开发过程中各阶段评估的结果传递给相关人员,还负责检查语义的不一致性。此外负责配置服务器和制定相关政策,将本体上传到服务器,并对协同开发者可视化展示:概念翻译和定义的自然语言表述,本体的树形结构化展现以及 OWL 文件管理。

2.1.2 本体工程师 本体工程师需要熟悉本体语言,并有一定的研究能力。他的重要作用是检查本体组件,尤其是校验公理的正确性,负责本体对不同角色进行不同形式的表现。有些功能只有本体工程师才能完成,例如将公理转化为自然语言,并评估转换的正确性,在协同式本体中仅能进行标注,不能进行编辑。

2.1.3 领域专家 领域专家在本体工程中涉及到知识的获取、维护和评估等工作,可以和其他领域专家协

<sup>①</sup> NeOn 是由欧盟第六框架资助的有 14 个成员国共同参与的項目(基金号: IST-2005-027595),2006 年设立,其宗旨是利用分布式环境中大规模语义的应用,尤其是提高协同处理现有本体的能力,使其具有很高的动态性和不断的演变功能。

同进行领域知识的获取以及知识的本体化。他们不涉及本体语言(例如 owl),但是可以将本体翻译为自然语言,以便于其他领域本体更好地理解内容。经过培训,领域专家可以添加概念及其相关属性和关系。

2.1.4 终端用户 在协同开发本体中,终端用户不能直接和其他角色的人员进行交互。他们熟悉领域知识和本体,因此可以给出本体应用情况的反馈意见。尽管最终用户和领域专家一样不具有本体工程的知识,但领域专家参与知识(概念)的获取阶段,熟悉隐藏在背后各种关系。终端用户可以将发现的 bug 和问题反馈给管理员。

### 2.2 本体协同构建的必要操作

协同构建本体包含构建本体的一般步骤,同时又对单机构建本体的环境提出更高的要求,例如需要进程的并发控制、语义的一致性检测、领域专家意见分歧的协调、版本的管理等。表 1 为本体协同构建需要的操作:

表 1 本体协同构建的必要操作

必要因素	说明
单个本体构建的必要因素	本体的创建、编辑、浏览、导航以及检索等操作
角色管理	为每个用户赋予不同的权限,例如评论、修改等
workflow 支持	可定义不同的编辑人员负责的部分,发起变化 (proposal for change) 的处理过程,达成一致的过程以及用户起到的不同角色等
访问控制	制定一些访问政策,约定用户可操作的对象,和可操作的方法等
讨论、注释及改变等集成、存档和展现	能够将用户的讨论、注释等和概念集成起来,加深对概念的理解和表达
可视化的本体展示	支持不同用户有不同的视图,并提供可视化浏览
持续的编辑	在持续编辑的过程中,即使不保存也能存档
不同版本的存档	对版本进行控制,在必要时可以回滚到某个时间点状态
仲裁决策	成立仲裁委员会管理协同工作,控制决策
语法/语义不一致性检测	当用户更改本体类时,会引起子类的变化,在变更后,应自动检测子类属性
数据的起源信息	查找信息的来源,它的编辑者、改变者、投票者和评论者等
可扩展性	并不是针对工具来说,而是整个方法应具有可扩展性,以便支持本体组件和用户的可扩展性
可信任性	不能丢失数据
分歧达成一致的支持	意见不一致时,怎样使其达成一致
同步/异步控制	当用户同步/异步标注与改变本体时,需要通知相关合作者,并处理好本体组成部分(模块)的协同

### 2.3 DCM 模型的设计

本文提出 DCM 模型,即分布式加工(Distributed implication)、集中式存储(Central storage)、模块化开发

(Module development)。DCM 借鉴了 DILIGENT 本体的开发阶段,DILIGENT 本体工程中包含了 5 个主要活动:本体的建立、本地化、分析、校正以及更新等<sup>[1]</sup>。DILIGENT 方法中,专家委员会先行构建初始化的本体,用户可以免费使用,并能够在本地根据自己的需要进行修改。专家委员会可以很好地调节和展示参与本体构建的各方面的利益,确保这些可被共享的核心本体的质量,并负责更新这些核心本体的决策。

DCM 框架分为三个层次:数据层、功能层和用户层。数据层存储本体、工作流数据、文档和用户讨论与标注信息等,标准文档以 XML 形式存储,本体以 RDF 或 OWL 语言存储,本体集中式存储在数据中心。图 1 为多个用户在分布式环境中协同构建本体的概念框架图:

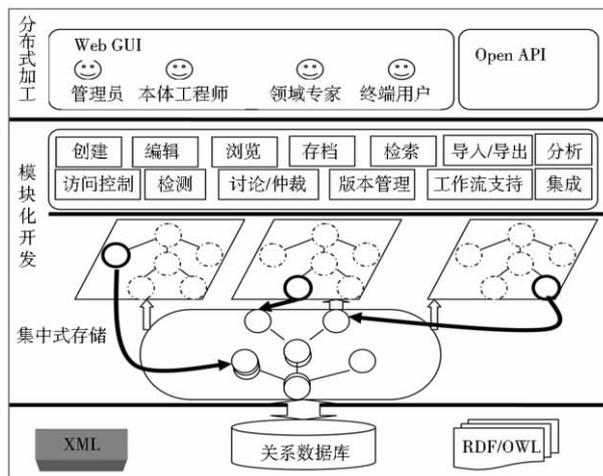


图 1 大规模本体协同构建 DCM 框架

所谓分布式加工是指在技术上支持多用户在分散的物理环境下实现本体的加工,不同领域的专家、不同角色的用户可以同时对目标本体(DCM 框架中待要加工的本体)模块进行创建与编辑,DCM 框架可以确保相关操作的并发执行;所谓集中式存储包括在技术上实现本体的集中存储与备份,合理处理本体模块之间的依赖性,在管理上有统一的机制,协调纠纷,达成最终意见的一致;模块化开发指目标本体可以按照领域或者功能进行划分,并将其指派给相关的建设者,实现本体模块中对象创建、概念定义、属性赋值以及概念关系构建等操作,开发完毕的本体模块具有良好的封装性,便于其他本体模块的调用,最大化地实现本体的重用。

2.3.1 本体描述语言和存储 本体描述语言可采用当前被广泛接受的 RDF 语言与 OWL,即构建的本体符

合 Ontology Lite 层次的本体,支持 OWL 语言描述的本体和关系数据库中存储的本体进行互操作。本体的存储和描述是大规模本体协同构建过程中首先要解决的技术问题。

本体的描述语言和存储涉及到的用户主要是本体工程师。在 DCM 框架中,物理上分散的本体模块具有统一的本体描述语言,由本体工程师统一定义,各模块需要共同遵守。

2.3.2 本体的编辑、维护 创建、编辑和维护本体或本体片段,本体自身会发生变化,需充分考虑每种变化的原因和结果,提供易于编辑本体的界面。这里的编辑定义为创建和修改本体的任务,在编辑本体时有可视化的展示界面对于用户来说有很大的帮助作用,最佳的状态是用户可以与可视化界面进行直接交互。

浏览本体是本体编辑和维护过程中必不可少的部分,对于高效的编辑和维护本体来说至关重要,这里定义浏览为:支持用户理解概念的层次关系,展现实例和属性。在查找特定的概念、实例时浏览功能必不可少。

分布式开发环境下本体模块的删除是确保各用户能够连续工作,实现目标本体稳定建设的前提。因为目标本体被分为粒度不等的模块,模块之间存在相互调用关系,具有很强的依赖性,各本地用户在上传自己负责的内容时,需要检查自己模块调用的其他模块的概念是否存在或者发生变化,并根据实际情况,调整上传本体加工内容的策略。

本体的编辑、维护涉及到的用户主要是领域专家和管理员。在 DCM 框架中,本体的编辑和维护由物理上分布的领域专家从事,管理员进行审核和最终发布。

2.3.3 版本管理 描述创建和维护本体不同元素时,管理本体变化和变化带来影响的能力。每个领域都定义了自己特有的概念,这些概念在现实中是会变化的,相当于数据库中的 Schema 版本。版本管理需要解决如下三个问题:

- 如果一个概念/关系在新的本体中不存在了,意味着该概念/关系已被删除。
- 如果概念/关系在旧的本体中没有,而在新的本体中有,则该概念/关系是被新建的。
- 如果新的本体中概念/关系的定义描述和旧的版本中有差别,则表明该概念/关系被修改了。

版本管理应该能够追溯数据的起源、数据变化的路径,在必要的时候可以通过事物的回滚,还原到历史的某个状态,同时用户对已有成果进行验证。分布式本体协同构建中注意用户标注、评论以及各种意见的

整合,有助于对概念的理解与表达,在实践中,对这些内容进行及时存档,相互关联。

版本的管理涉及到的用户主要是管理员和本体工程师。版本管理确保本体模块之间依赖性的完整性,保证目标本体建设的稳定性。

2.3.4 本体的映射和集成 构建本体需要实现已有本体相互之间的映射,并能形成一个相互兼容一致的本体。集成、映射和合并描述了采用和映射已有本体的能力。本体协同构建平台中的本体是一个多用户参与构建、跨领域的本体。本体管理应该能够映射合并其他领域开发人员开发的本体,甚至能够集成外部本体,最后可以形成一个完整的本体。同时,集成、映射和合并也是解决异构问题的有效途径。

本体的映射与集成涉及到的用户主要是领域专家和本体工程师。

2.3.5 本体的查询和推理 支持通用的本体查询语言进行本体的查询,例如 SPARQL 等。推理组件告诉开发者关于本体一致性的内容。如果没有一定的推理功能,很容易产生错误。另外推理功能可以利用本体得到事件结果产生的原因,可被用来提高语义检索结果的准确性和召回率。

本体的查询和推理几乎涉及到所有类型的用户,不同层次的用户需求有所差异。

2.3.6 导入/导出 支持符合标准规范描述的已有本体的导入,并能导出符合业界本体描述语言的领域本体。

导入导出涉及到的用户主要是管理员和本体工程师,终端用户可根据实际的需要对已发布的本体进行取舍,导出自己所需的本体片段。

2.3.7 工作流支持 协同本体构建模型中工作流的作用是描述项目参与者怎样在定义上达成一致,谁能够发起改变,谁能执行改变,当这些改变公布后,谁能对这些改变进行评论。

工作流的操作主要由管理员控制,涉及到所有类型的用户。

2.3.8 用户访问控制 访问控制和用户角色管理与分配密切相关,访问控制通过对用户的授权决定其具有权限的操作对象和操作方式。例如并不是所有具有写权限的人对所有本体都能进行编辑,尤其是大规模协同加工过程中,用户可能仅对某些领域的局部概念熟悉,系统允许其编辑该用户熟悉的部分,而对其不熟悉的内容只能浏览或者链接。用户的角色是通过系统预定义的一系列的政策规约,通过建模和程序的功能

实现,不同的用户具有不同的权限,每个项目对用户角色权限的划分各具特色,例如有的用户可以发起修改的请求(proposal for changes),而自己无权限修改,而有的用户可以对请求进行评论,但是没有发起请求的权利。同时不同项目对角色权限控制的粒度要求差别也较为明显。大规模本体协同构建模型中需要充分考虑不同用户对已有本体和数据认识的角度差别。

访问控制的操作主要由管理员控制,涉及到所有类型的用户。

2.3.9 仲裁和意见分歧的统一 在协同本体构建模型中对意见分歧的处理是区别于单机构建本体的重要因素,从现有分析来看分为两种方式,一种是成立专家委员会,对分歧进行评估与决策;另外一种是通过相关人员投票决定。不管选择哪种方式,只要达成一致就必须执行,否则无法协同工作。

成立专家委员会仲裁分歧的意见,在调研的系统里,多数由管理员来承担,或者将专家委员会成员看成具有部分权限的管理员。

通过 DCM 框架搭建本体协同构建平台实现的分布式模块化的协同开发,创建、编辑和浏览本体,并能实现已有本体内容的导入,按照特定格式实现本体内容的导出,为了重用已有本体,提供映射集成功能,具备一定的推理实现一致性检查,跟踪本体变化的动态性实现本体的版本管理,通过访问控制使不同用户在各自的责任范围内对本体实现相应操作。本文旨在阐述模型的构成元素,模型中的每个元素及功能的实现不再延伸阐述,例如本体语义与语法一致性检查等有专门的工具支持或项目研究。

### 3 总结

本文阐述本体协同构建研究的热点和方向,构建了 DCM 框架:分布式加工、集中式存储与模块化开发,支持分布式环境下大规模本体的协同构建,并对框架中主要元素和功能点进行了阐述,尤其是对 workflow、访问控制和仲裁的支持等。DCM 模型的主要思想在 Ne-On 项目和 Hozo 工具中都有应用,对项目的管理和流

程的设计起到支撑作用,但是模块间的依赖关系处理在一定程度上约束了开发的进程,需要手工干预,这也是模块化开发本体的瓶颈之一,有待于进一步解决。

参考文献:

- [1] Tempich C, Simperl E, Luczak M, et al. Argumentation-based ontology engineering. *IEEE Intelligent Systems*, 2007, 22(6): 52-59.
- [2] Chen Y. A collaborative ontology construction tool with conflicts detection, semantics, knowledge and grid//Fourth International Conference on Issue Date. SKG, 08, 2008: 12-19.
- [3] A semantically-rich, graphical environment for collaborative ontology development in agenticities. [2010-11-28]. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.83.3237>.
- [4] Jimenez-Ruiz E, Berlanga R. A view-based methodology for collaborative ontology engineering: An approach for complex applications (VIMethCOE). [2010-11-25]. <http://www.computer.org/portal/web/csdl/doi/10.1109/WETICE.2006.13>.
- [5] Gangemi A, Lehmann J. CODO: An OWL metamodel for collaborative ontology design. WWW 2007, Banff Canada. [2010-12-01]. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.127.1742>.
- [6] Bao J, Honavar V. Collaborative ontology building with wiki@nt-A multi-agent based ontology building environment. [2010-04-02]. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.59.1285>.
- [7] Sung-Kooc L, In-Young K. Collaborative ontology construction using template-based wiki for semantic web applications//Computer Engineering and Technology. International Conference on Issue Date ICCET'09 2009: 171-175.
- [8] Noy N F, Tudorache T. Collaborative ontology development on the (semantic) Web. [2010-10-28]. <http://www.aaai.org/Papers/Symposia/Spring/2008/SS-08-07/SS08-07-009.pdf>.
- [9] Correndo G. Survey of tools for collaborative knowledge construction and sharing. [2010-12-28]. <http://portal.acm.org/citation.cfm?id=1339264.1339647>.
- [10] Reza Kalbasi Khoramdashti. Collaborative ontology development for geo-information sharing. [2011-01-02]. [http://www.itc.nl/library/papers\\_2009/msc/gfm/kalbasi.pdf](http://www.itc.nl/library/papers_2009/msc/gfm/kalbasi.pdf).
- [11] Vrandečić D, Pinto H S, Sure Y, et al. The diligent knowledge process. *Journal of Knowledge Management* 2005 9(5): 85-96.

(作者简介) 李亚子,男,1979年生,博士,助理研究员,发表论文20余篇;钱庆,男,1970年生,研究员,主任,发表论文10余篇;郭文丽,女,1968年生,助理研究员,发表论文20余篇;方安,男,1976年生,副主任,发表论文6篇;洪娜,女,1980年生,助理研究员,发表论文15篇;王军辉,男,1982年生,实习研究员,发表论文8篇。