

Web 环境下本体管理^①

高 丽¹, 王新房¹, 陈兴虎²

¹(西安理工大学 自动化与信息工程学院, 西安 710048)

²(西安航空职业技术学院, 阎良 710089)

摘 要: 分析了语义 Web 环境中本体的特点和本体管理的功能需求, 提出了针对 web 环境下本体库管理系统体系结构的设计原则, 基于该原则设计 B/S 结构的本体系统. 最后, 给出了一个遵循该体系结构设计和实现的原型系统.

关键词: 语义 web; 本体; 本体管理; 图形化浏览; 图形化编辑

Management of Web-Based Ontology

GAO Li¹, WANG Xin-Fang¹, CHEN Xing-Hu²

¹(School of Automation & Information Engineering, Xi'an University of Technology, Xi'an 710048, China)

²(Xi'an Aeronautical Polytechnic Institute, Yanliang 710089, China)

Abstract: This paper has analysed the character for ontology and the function of ontology management, meanwhile, proposed the design principle for the system structure of ontology management in the semantic web surroundings. The B/S Ontology system is also developed by this policy. Finally, an original system following the ontology structure is also given in this paper.

Key words: semantic web; ontology; ontology management; GraPhleal browsing; GraPhleal editing

1 引言

本体^[1]在语义 Web 得到了广泛的应用, 借助本体, 可以方便地获取、描述和表示相关领域的知识, 提供对该领域知识的共同理解^[2]. 在需要知识共享^[3]的语义网中, 本体是解决语义层次上的信息共享和重用的基础. 本体编辑工具^[4,7]是对领域知识建模的必要工具, 能完成对本体的创建、解析、管理、存储和重用等工作^[8]. 但随着社交网络的普及使“用户贡献价值”的理念深入人心, 对本体的需求不断扩大. 互联网数据的爆炸式增长以及数据内容和形式的多样化, Web 中的本体越来越多. 领域知识的丰富本体的规模越来越大, 因此在 Web 环境下为构建和管理大量、大规模本体^[9]提供一个通用的框架和软件平台, 是很有必要的. 因此本文讨论基于 Web 的本体管理系统的设计与实现.

2 系统总体结构

2.1 功能需求

本体开发是一个反复迭代的过程^[2], 领域专家通

过持续总结相关领域的特点和规律为本体添加组成元素. 这样一来, 随着内容的丰富, 本体产生了不同的阶段性版本, 领域专家将不同阶段的本体应用到实际系统中或与其他专家交流, 根据评价结果对其修正, 直至满意. 这个反复迭代的过程将贯穿本体的整个生命周期. 结合 web 特点, 同时为了保证领域信息的完整性以及语义描述的表达能力和推理能力, 本体开发的内容包括: 类的编辑、属性的编辑、实例的编辑; 类的编辑应包括的内容如表 1 所示.

表 1 类编辑的内容

内容	说明
类的创建和删除	创建类和删除已有类
类的重命名	编辑类的名称
等效类	添加和删除类的等效类
父类	添加和删除类的父类
不相交类	添加和删除类的不相交类
不想交并类	添加和删除类的不相交并类

① 收稿时间:2014-11-22;收到修改稿时间:2015-01-12

属性的编辑应包括的内容如表2所示.

表2 属性编辑的内容

内容	说明
属性的创建和删除	创建属性和删除已有属性
属性的重命名	编辑属性的名称
等效属性	添加和删除等效属性
父属性	添加和删除父属性
逆属性	添加和删除逆属性
定义域	添加和删除对象属性定义域
值域	添加和删除数据属性值域
不相交	添加和删除不相交属性

实例的编辑应包括的内容如表3所示.

表3 实例编辑的内容

内容	说明
实例的创建和删除	创建实例和删除已有实例
实例的重命名	编辑实例的名称
实例的归类	添加, 修改, 删除已有实例的类
实例之间关系	添加和删除实例的对象属性
实例的值	添加和删除实例的数据属性

2.2 非功能需求

非功能需求是系统必须具备的属性, 这些属性可以看作是一些特性, 他们使产品具有吸引力、易用、快速或可靠, 同时为工作赋予特征的. 非功能需求主要包括以下几种.

观感需求: 即产品的外观的精神实质, 也就是与用户界面的感官相关的一组属性.

易用性需求: 也就是产品的易用性程度, 以及特殊的可用性考虑, 通常包括用户接受率、因为引入该产品而提高生产效率、错误率, 特殊人群的可用性等指标

性能需求: 也就是关于功能实现要求有多快、多可靠、多少处理量及多精确的约束. **可操作性需求:** 衡量产品的操作环境, 以及对该操作环境必须考虑的问题. **可维护性和可移植性需求:** 期望的改变, 以及完成改变允许的时间.

2.3 总体架构设计

基于上述的需求分析系统主要是基于浏览器/服务器(B/S)三层框架如图1.

2.3.1 表示层

主要职责就是为用户提供信息, 以及用户的指令翻译, 它包括界面外观层、界面规划层. 界面外观层: 主要利用 ASP.NET 的丰富页面组件, 设计一系列的网

页界面, 实现了本体显示、编辑等用户交互的界面.

界面规划层: 根据用户指令调用业务逻辑层中相应的类, 并将数据传递给业务层

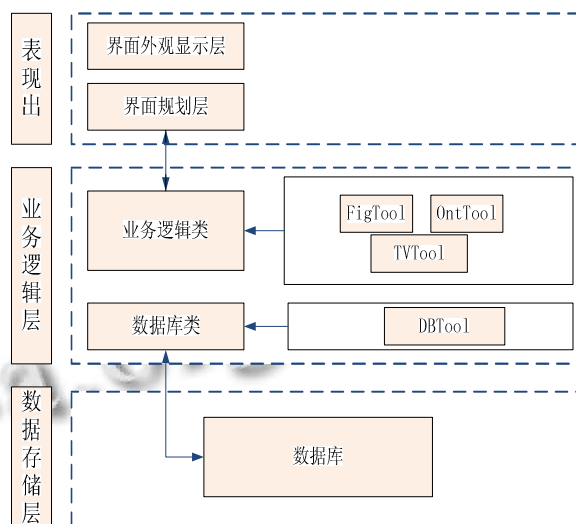


图1 体系结构

2.3.2 业务逻辑层

主要是对用户提交的指令及数据做校验, 再加工后将数据存储到数据存储层或者将数据提交后返回给表示层. 表示层主要包括业务层、数据服务层.

业务层: 提供给表示层指令服务类, 根据用户指令和数据的不同, 将指令划分给不同的类处理, 并将指令操作结果返回. 同时, 业务层还负责向数据服务层发送数据同步请求, 通知数据存储层及时更新数据, 根据业务需求抽象出的类主要有以下几个

(1) **FigTool(绘图类):** 主要显示本体中类、对象属性、数据属性或者个体等各个知识之间的关系, 实现的功能如下表所示

表4 绘图类

方法	说明
setParameters	设置绘图工具需要运行的程序
getGraph	用于绘图中的dot命令字符串
writeGraphToFile	将绘制的图片写入文件中
GetATempFileName	得到一个临时文件名
get_img_stream	调用dot.ext, 根据dot文件, 绘制图形, 结果保存在临时文件中, 并返回图像数据
writeDotSourceToFil	将dot绘制命令存入一临时文件, 并返回临时文件名
Draw_OntInd_OPFig	根据个体标识编号, 绘制个体关系图, 返回图形字节数组

(2) OntTool(本体操作类): 对本体中类、对象属性、数据属性或者个体进行检验和编辑等相关操, 功能如图 2 所示



图 2 本体操作接口函数

(3) TVTool(TreeView 操作类): 对 TreeView 进行各种基本操作, 实现的功能如表 5 所示

表 5 TreeView 操作类

方法	说明
AddSubNode	将 dt 中存放的层次数据, 添加到对应的节点
OWLEntityTreeView	将全部 OWL 类、对象属性或数据类型属性按照层次
ClassIndTreeView	按类的层次结构在 TreeView 中显示各类和个体
AddSubNode	添加类或者属性的子节点

数据服务层: 数据服务层是业务逻辑层支持层, 它通过 ADO.NET 访问数据库, 执行业务层的数据处理要求. 数据服务层可以有效地分离数据访问的相关操作, 如连接管理、事务管理、存取数据集等, 从而保证业务逻辑层可以专注于数据处理. 在该层中我们可以通过本文提出的 DBTool 类来实现, 如表 6 所示.

表 6 数据库访问类

方法	说明
SetDBConnectionString	获得 Web.Config 中数据库联接配置节的名字
GetOpenDBConn()	得到一个数据库连接对象
ExcNonQuery	执行指定的 SQL 语句输入
TongJiChaXun	根据规定的 SQL 语句进行统计, 返回统计结果

GetNewId	当向表中增加一行时, 得到该行标识编号
GetIndentityId	向表中插入一行后, 获得新增行的标识编号
GetColumnsOfATable	获取一个表的全部列
DeleteRowById	根据标识编号, 删除表中的一行
ToStrin	将数据库表中获取的数据转换为字符串
ToInteger	将从数据库表中获取的数据转换为整数

2.2.3 数据存储层:

该层的建立是通过数据库管理系统来实现的. 负责整个系统的信息数据维护, 提供信息数据的储存与管理. 使用 DBMS 来管理数据的好处是可以协助数据的处理, 提高数据的使用效率. DBMS 处理数据除了完成数据库本身的插入数据、修改数据、执行系统函数等 SQL 操作外, 还可以通过存储过程和触发器来自动地在数据库上存储应用程序的业务逻辑和函数模块. 存储过程不需要每次都重新分析和优化, 而是一次性编译成执行计划缓存在数据库中, 以后可以直接调用, 避免了重复的解析过程, 节省了时间. 特别是当查询任务由一系列 SQL 命令组成时, 应该考虑将其编写成存储过程. 这样在数据层上就可以对数据做出相关的功能强大的处理, 而不是仅仅把请求结果返回给客户机.

2.4 系统主界面布局设计

系统针对本体中的类、对象属性、数据属性、个体等特性设计四个界面, 在相应的界面上用户可以查看本体信息和操作方便编辑模块. 将每个界面总体设计为两大功能模块: 功能树和操作界面使系统维护比较简单, 用户在编辑信息时比较有针对性.

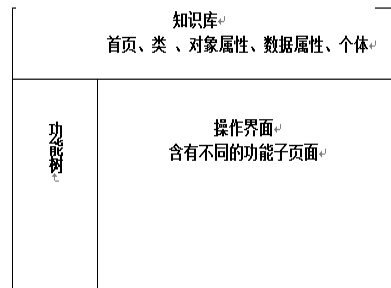


图 3 主界面

类层次, 对象属性层次, 数据属性层次, 以功能树的形式显示, 方便用户查看他们之间的关系, 在实

现类、属性、个体的添加,和删除本体维护时操作简单,时效性比较好.同时会及时在树状层次中显现编辑模板中本体的编辑信息保证数据的一致.

操作界面含有不同的功能子页面是对本体的描述信息添加,修改、删除,同时为了方便用户编辑对已存在本体信息通过 web 导航可以直接以客户端脚本显现,供用户选择.通过调用本体操作函数在 web 可视化界面上实现以下对本体编辑.

2.4.1 类和属性编辑块

类和属性编辑块以直观的形式展示本体中的类和属性的相关信息.类编辑块为用户对本体进行管理时与类信息的交互提供以下子模块:

- ① 类的添加和删除;
- ② 类以树状层次结构显示;
- ③ 祖先类和子孙类以树状层次结构显示;
- ④ 类的重命名服务;
- ⑤ 类描述信息以列表显示
- ⑥ 等效类、父类、成员、目标键、不相交类、不相交并的添加和删除;

属性编辑块为用户对本体进行管理时与属性信息的交互提供以下子模块:

- ① 属性的添加和删除;
- ② 属性以树状层次结构显示;
- ③ 父属性和子孙属性以树状层次结构显示;
- ④ 属性的重命名;
- ⑤ 属性描述信息以列表显示
- ⑥ 属性的等效属性、父属性、逆属性、定义域、值域、不相交属性的添加和删除;

为了实现对本体中类和属性的协同编辑功能,类和属性编辑块支持类和属性信息的显示动态更新.每个子模块本分别监听相应的事件,一旦相应事件发生,则订阅了该事件的模块则采取事先约定的动作执行显示更新.

2.4.2 实例数据输入模板

实例数据输入模板简化了对本体中类的实例的管理,以直观的形式展示本体中的实例的相关信息,为用户对本体进行管理时与属性信息的交互提供以下子模块:

- ① 实例以列表层次显示;
- ② 实例在特定类下的添加和删除;
- ③ 实例的查询;

④ 实例之间以关系图形式直观显示;

⑤ 实例属性值的添加和删除;

为了实现对本体中实例的协同编辑功能,实例数据输入模板,实例信息的动态更新.每个子模块本分别监听相应的事件,一旦相应事件发生,则订阅了该事件的模块则采取事先约定的动作执行显示更新

3 系统详细实现

鉴于需求分析中类、对象属性、数据属性实现的功能基本相同,在实现类、对象属性、数据属性各种功能算法也大体相仿,所以下列主要针对类和个体实现进行详细的描述.

3.1 类的实现

类的实现主要是通过以下几大模块实现,树形图显示模块:主要是显示类名称及子类与父类的关系.列表显示模块,主要实现类描述信息查看及信息的编辑. JQuery miniUI 模块:主要在添加已有类,实现对类信息查看.图 4 是类树层次和局部关系图的主界面查看,在界面的右边最上方有“对象属性”、“数据属性”、“个体”导航按钮,可以定位到他们各自对应的主界面区.

图 5 是编辑类信息时提供给用户输入接口,在图 4 左边树层次区正上方点击“增加”按钮会自动弹出添加类信息文本输入框,当输入信息完成以后点击“确定”按钮,会出现“文本信息提示”对添加操作成功与否进行说明,如果添加成功,文本框自动消失同时新增加的信息自动更新到树层次中.删除类信息时首先在树层次中选中类,点击“删除”按钮,会出现“信息提示框”再次提醒用户对删除的内容进行确认,如果删除成功,树层次自动删除该信息.右边区域有局部关系图查看按钮和类描述信息添加按钮.当选中树节点类,点击“类信息”按钮可以查看选中类的直接子类和父类关系图.点击其他按钮会出现添加类描述信息输入框以及对应该操作中已存在信息显示框.在输入框最右边“...”导航可以查看类、对象属性、数据属性树层次和个体列表显示,在该显示区选中某一信息点击“确定”树层次和列表框自动消失,同时把该信息直接传递到输入框中.当添加成功,输入框下方列表自动显示新增内容,同时会出现“添加成功”文本提示信息.删除描述信息只需点击列表中删除就能实现.

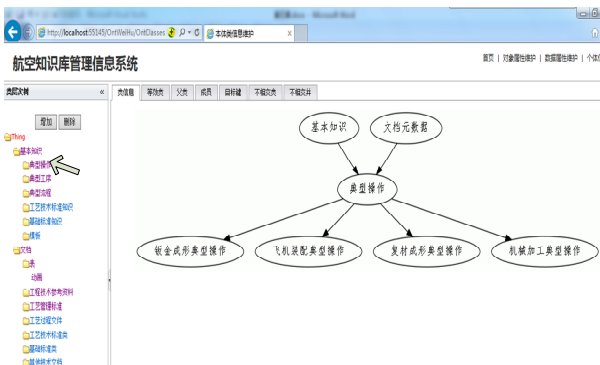


图 4 类主界面浏览

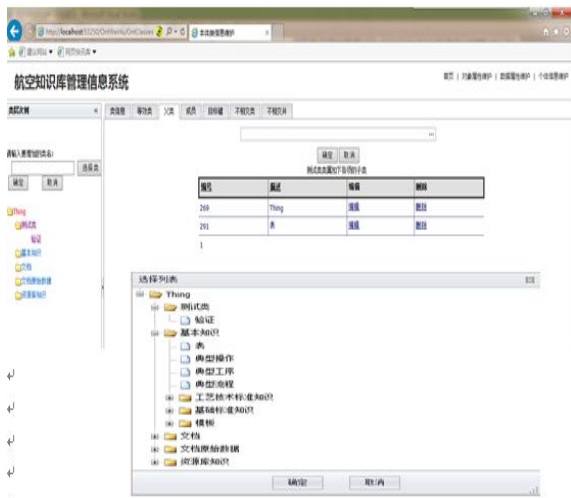


图 5 类编辑

3.2 个体实现

个体主要通过以下模块实现，列表显示模块，实现个体信息查看及信息的编辑。JQuery miniUI 模块：主要实现对个体添加已有类，对象属性，数据属性显示。图 6 是个体主界面浏览：

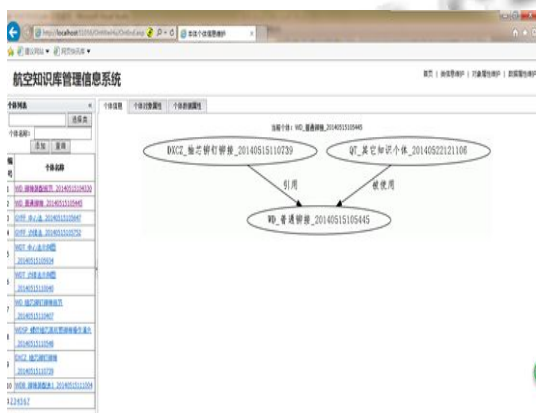


图 6 个体主界面浏览

个体编辑主要提供给用户输入信息的接口如图 7

所示，个体增加设计在特定类下实现，所有左边正上方设置类信息输入框，在该输入框中只能输入已存在的类信息，为了方便用户输入旁边还专门设计对已有类查看的导航“选择类”按钮，主要实现对已有类的自动输入。当用户在类和个体输入框完成输入信息，点击“添加”会出现“文本信息提示”对添加操作成功与否进行说明，如果成功，个体列表框中自动新增加一行显示新增内容。个体查询在个体名称输入框中，输入要查询个体的关键字点击“查询”按钮，在个体列表显示区中按照相似度高低由上到下显示。个体删除首先在列表中通过“选择”键选中要删除的个体点击“删除”按钮就实现个体的删除，如果删除操作成功，个体列表中自动删除该行信息。右边区域有局部关系图查看按钮和类描述信息添加按钮。当选中树节点类，点击“类信息”按钮可以查看选中类的直接子类和父类关系图。点击其他按钮会出现添加类描述信息输入框以及对应该操作中已存在信息显示框。在输入框最右边“...”导航可以查看类、对象属性、数据属性树层次和个体列表显示，在该显示区选中某一信息点击“确定”树层次和列表框自动消失，同时把该信息直接传递到输入框中。当添加成功，输入框下方列表自动显示新增内容，同时会出现“添加成功”文本提示信息。删除描述信息只需点击列表中删除就能实现。

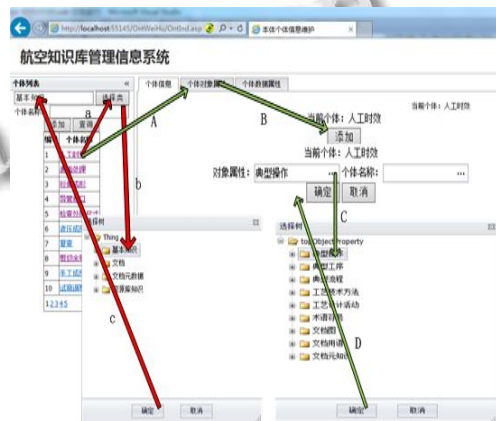


图 7 个体编辑

4 结论

本系统实现了在 web 环境下，对本体中类，对象属性，数据属性，及个体不同概念角色的管理。同时对数据库操作时运用事务思想实现了多用户对本体进行添加，删除，修改等本体维护，保证了数据的一致性，完整性的前提下实现了系统的时时更新。

参考文献:

- 1 邓志鸿,唐世渭.ontology 研究综述.北京大学学报,2002,38(5):730-73.
- 2 李勇,张志刚.领域本体构建方法研究.计算机工程与科学,2008,30(5):129-131.
- 3 Studer R, Benjamins VR, Fensel D. Knowledge engineering: principles and methods. Data & Knowledge Engineering, 1998, (1-2).
- 4 OntoEdit. www.semtalk.com/semnet_files/POntoEdit.htm.
- 5 WebOnto. <http://kmi.open.ac.uk/projects/webonto/>.
- 6 OilEd. <http://oiled.man.ac.uk/>.
- 7 Protégé. <http://protege.stanford.edu/>.
- 8 陶皖,廖述.当前本体编辑工具的分析与研究.计算机工程与设计,2005(26).
- 9 NF Noy, Chugh A, Liu W, Musen MA. A framework for ontology evolution in collaborative environments. Fifth International Semantic Web Conference(ISWC). Athens, GA. Springe. 2006. LNCS 4273r.

WWW.C-S-A.ORG.CN

WWW.C-S-A.ORG.CN