●丁晟春　甘利人　陈开浩

本体的图形化可视检索研究与应用*

摘　要　研究了开发语义应用系统过程中的一些关键技术,设计并实现了基于 J2EE 的 B/S 结构的通用本体可视化检索系统。本系统使用 Ajax 技术实现了客户端与服务器端的的异步通信,能够将任一 OWL 本体中的类层次、属性、实例等语义关系以动态二维网状图形浏览器的方式进行直观显示,动态图形浏览器支持全览、聚焦、移动等功能。它还支持关键字、SPARQL 检索与图形检索的过程同步。图 5。参考文献 5。

关键词　语义检索　本体　可视化检索　OWL 本体

分类号　G354

ABSTRACT　The authors make a study of some key technologies in the development of semantic application systems, and design and realize a visualized retrieval system for general ontology, which has a J2EE-based B/S structure. In this paper, the authors also analyze the technologies used in this system and some design conceptions of the system. 5 figs. 5 refs.

KEY WORDS　Semantic retrieval.　Ontology.　Visualized retrieval.　OWL ontology.

CLASS NUMBER　G354

目前国内外主要采用树形列表反映本体结构,检索结果以文本加超链接方式显示类、属性、实例等之间的语义关系。这种方式虽然已经能够较好地展示本体内部的组织结构,但仍不能很直观地显示本体的语义关系,且有些本体的语义关系相当复杂,关系间的交叉组合常形成网状结构,采用文本加超链接的方式显示网状关系往往力不从心[1]。

本文着重探讨开发本体应用系统中的一些关键性技术,并设计、实现基于 B/S 结构的本体可视化检索系统。

1　本体图形化可视的设计

1.1　设计思路

如何将本体中网状结构的知识以图形可视化的方式展示呢? 通过分析本体资源的定义语言 RDF 的语法结构,我们认为 RDF 是由一组组的陈述构成的,一个陈述由一个主体、谓词、对象组成。根据这一特性,由此产生了本体可视化系统设计思路:(1)先利用本体解析 API 从本体中提取出一个个三元组;(2)再以三元组的主体和对象为结点,以它们谓词为边,使用 Java 相关技术绘制出一个网状式连通图;(3)结合 Applet,Ajax,J2EE 技术实现 B/S 模式下的本体可视化检索。如图 1 一个简单的 RDF 文档,抽取其中的所有三元组后,可绘制出如图 2 所示的 RDF 三元组图。一条有向边及两端的结点代表一个三元组,边的起始结点代表主体,终止结点代表对象,边的标签值代表谓词。

```
<?xml version = "1.0"? >
<rdf:RDF xmlns:rdf = "http://www.w3.org/1999/02/22
-rdf-syntax-ns#"
xmlns:contact = "http://www.ijava.com/contact#" >
<contact:Person rdf:about = "http://www.w3.org/Peo
ple/EM/contact#me" >
<contact:fullName > Eric Miller </contact:fullName >
<contact:mailbox rdf:resource = "mailto:em@w3.
org"/ >
<contact:personalTitle > Dr. </contact:personalTitle >
</contact:person >
</rdf:RDF >
```

图 1　一个简单的 RDF 文档

1.2　系统的实现方案

首先通过 Jena,owl api/Pellet 等支持 OWL 解析的 API 完成对任一个本体资源文件(OWL 文档)内部结构解析。

其次根据解析结果将数据(一个个三元组)输出到一个 XML 文件,输出的文件包含了本体内部结构的相关信息,包括类、属性、实例以及它们之间的语义关系。

* 本文系解放军总装备部项目"面向科技数据库网站应用的 Ontology 实体构建研究"的成果。

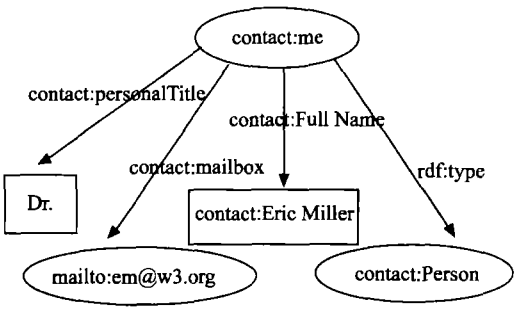

图2　本体可视化中的RDF三元组图

然后利用一定的二维图形编程接口(如TouchGraph[2],Graphviz和Jgraph等)读取上面第二步骤生成的XML数据文件,将本体中的类和实例绘制成结点,其中的语义关系(类之间、类与属性、实例之间、类与实例等)绘制成边,最终以可视化的方式把OWL本体表达的语义信息展示给用户,实现本体的可视化检索。

2　系统关键技术的研究

2.1　OWL本体的解析

在此实现方案中,要考虑两个问题。

(1)OWL本体内部结构信息的加载。OWL本体内部结构信息的加载方式可以有两种:一次全部加载和分次加载。根据我们得出的实验结果认为:较小的领域本体适宜采用一次全部加载方案,即用户第一次打开系统时,系统一次性加载本体库中全部内部结构信息,这样可以使得用户在可视化浏览、检索本体的过程中获得较快的响应性;而对于较大的应用本体、领域本体,适合采取分次加载方案,即本体内部结构信息是在用户可视化浏览检索本体的过程中分次加载的,这样即使是大本体也不至于占用过多的客户端机器内存空间,并且较前一种方案,用户第一次进入系统时所需等待时间更短。

(2)OWL本体资源文件的解析。在OWL本体中,所有用户自定义类都继承自owl:Thing,类又可以继承自另外一个已经存在的类,所有类层次关系构成了一个以owl:Thing为根结点的树形结构。要遍历树形结构,一般采用递归算法。本系统根据载入的本体,采用递归算法生成系统所需的含有本体内部结构信息的数据文件,包括:用于显示树形列表的数据文件、用于关键词搜索提示的数据文件、可视化图形边的数据文件和结点属性列表的数据文件。

2.2　本体检索模型

(1)OWL推理引擎。对本体而言,推理机主要有两个方面作用:一个是在建立本体时对本体进行一致性检查,确保本体内部无矛盾性。另外一个是通过推理挖掘本体中蕴涵的隐式知识,这也是语义检索中至关重要的一个作用。

本系统支持Pellet推理机,同时允许通过DIG接口挂接到Racer推理机。Pellet推理机以描述逻辑作为理论基础,采用Tableaux算法,在对小本体的推理上,Pellet的性能非常好。本系统使用Java直接调用Pellet推理机进行本体之间的蕴涵推理,并完成对军用领域飞机本体的一致性检查。对于大的本体,本系统允许切换到Racer推理机。通过推理机在本体解析时找出本体中未申明的隐含的子类关系,同时在本体检索时使用自定义推理规则挖掘其中蕴涵的隐式知识[3-4]。

(2)OWL检索引擎。本系统通过检索模型Model封装了OWL检索引擎及推理引擎,使用功能强大的RDF检索语言SPARQL进行关键字检索。除了提供直观的SPARQL窗口检索,还完成对文献信息资源的检索。

2.3　二维网状图形的绘制与控制

(1)本体中概念和关系的表示。二维网状图形中本体的概念和关系的表示采用TouchGraph组件来完成,以结点表示本体中的类、概念、实例,以边表示概念之间的关系,利用本体解析API提取出的一个个三元组数据文件,以三元组中的Object和Subject作为结点,以它们之间的语义关系为边,绘制出一个网状式的动态图。如图3为绘制结点的关键方法。

(2)结点径度的控制。为了方便说明问题,我们先定义"中心结点"和"径度"的概念:中心结点指用户关心并选择的当前结点;从中心结点出发到最远结点所经过边的条数称为径度。一个本体中的某一个类或实例与其他类,实例的关系是错综复杂的,这种数量众多的语义关系同时在一个有限的区域中全部显示完全没有必要,我们只需显示中心结点(用户关心的当前信息)及其相邻结点(当前信息的背景信息)即可。图形显示结点的层数由径度控制,当径度为0时,可视化图形只显示中心结点;当径度为1时,除中心结点外还显示与中心结点直接相连的结点;当径度为2时,还将显示与中心结点次相邻的结点。通过反复实验,我们认为可视化图形径度最好控制在3以内,超过3时,界面上的边将显得过多过杂,太多的边容易分散用户注意力,因此我们将径度的取值范围设为0到3。用户在浏览本体时,可根据需要随时调整径度。

(3)图形的移动和转动。为了能够在一个有限的空间显示庞大的图形界面,系统支持图形全览、聚焦、缩放、移动、旋转等功能。用户在可视化浏览检索本体的

过程中,可根据需要自由变换图形,获得更清晰的图形视觉效果。

```
public void paint(Graphics g,TGPanel tgPanel){
if(! intersects(tgPanel. getSize()))
    return;
        paintNodeBody(g,tgPanel);
int ix = (int)drawx;
    int iy = (int)drawy;
    int h = getHeight();
    int w = getWidth();
    if(visibieEdgeCount() < edgeCount()) {
    int tagX = ix(w -6)/2 -2 + W%2;
    int tagY = iy - h/2 -3;
    String hiddenEdgeStr = String. valueOf(edgeCount() -
visibieEdgeCount());
    g. setColor(Color. red);
    g. fillRect(tagX,tagY,3 +5 * hiddenEdgeStr length(),8);
    g. setColor(Color. white);
    g. setFont(SMALL_TAG_FONT);
    g. drawString(hiddenEdgeStr,tagX +2,tagY +7);}
if(tgPanel. getMouseOverN() = this&&Config. showPrope
rty)}
```

图3 绘制表示本体关系的边

(4)结点及边的过滤。本体可视化图形界面上的有向边表示本体中的两个类(实例)之间的关系,边上的标签值表示它们的具体关系名。本体中的知识内容呈网状结构,它们之间语义关系成网状结构。如果不对本体可视化的边进行过滤,那么整个图形界面将布满各种各样的边,太多的边会导致一些相对次要的边干扰相对重要的边,使用户在搜索时不知所措。因此我们有必要对图形上的边进行控制,决定是否展开某一结点的边。系统对于拥有超过n条边的结点默认不予展开,这个n值是用户可控制的。用户可自由选择隐藏或显示图形上的任一结点及边。

2.4 多个检索的过程同步

本体可视化系统提供多个检索入口,包括树形列表检索入口、图形界面检索入口、关键词检索入口。树形列表展示本体内部的类层次关系,实际上向用户提供了对整个本体的全览。图形界面检索由客户端的一个Applet实现,用户通过鼠标改变图形中心结点达到可视化浏览本体目的。可视化图形界面实际上向用户展示当前结点所代表的类(实例)与本体中其他类(实例)具体的语义关系。用户在树形列表选中单击具体的一个类时,图形显示区域将以该类名为中心重绘整个图形。当用户使用关键词检索功能时,若键入的关键词是类名(实例名),客户端向服务器提交查询的同时,图形显示区域也将以该类名(实例名)为中心重绘整个图形。这样树形列表搜索、文本搜索、可视化搜索可相互配合,达到同步搜索目的。

2.5 Ajax DWR引擎

传统的Web应用程序一般是由客户端浏览器向服务器提交数据,服务器返回新的页面在浏览器再次显示,这意味着每次数据往返都要刷新整个浏览器页面。本系统中为了解决浏览器端与服务器端的及时通信问题,使用Ajax技术[5],采用Ajax框架DWR作为Ajax引擎。Ajax引擎在客户端和服务器端交流的过程中担负着中间层的任务。用户界面要向服务器提交数据时,它负责收集数据并通过Http request(XML Http Request)向服务器发送数据,服务器处理完成后返回XML,Ajax引擎将XML处理为便于用户界面显示的XHTML和CSS数据,并刷新用户界面相应部分的显示,而非刷新整个页面,从而避免了不必要的数据往返,只有必要的数据在必要的时刻才在浏览器和服务器之间传输。最重要的是,用户甚至不知道浏览器正在与服务器通信:Web站点看起来是即时响应的。

结果表明整个系统就像是一个Windows桌面程序一样,没有明显的抖动效果且响应性很高。

3 本体可视化实现

系统使用Java开发,基于B/S结构。该可视化本体检索系统总体结构见图4。系统采用Web2.0关键技术Ajax,实现客户端与服务器异步通信,客户端提交查询请求时,页面不会产生抖动效果,因为用户与服务器端的交互完全交给JavaScript在后台进行,当服务器返回结果时,浏览器只是刷新局部页面而不是整个页面。

客户端是一个浏览器,主要由一个树形列表、一个图形显示区域、一个结果输出区域完成本体的可视化检索,同时提供关键字和SPARQL检索入口。客户端封装了Ajax框架DWR,所有查询请求都通过DWR与服务器端交互。由于Ajax框架DWR封装了客户端与服务器端交互的所有细节,客户端浏览器调用服务器端的Java代码就如同在客户端直接调用一般。

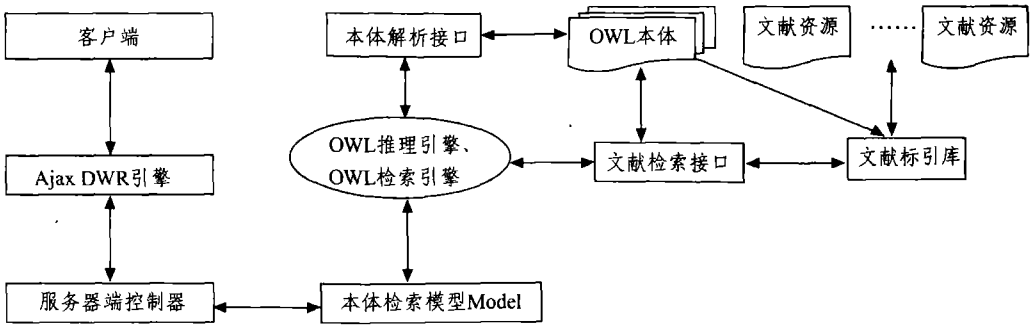

图4　系统总体结构

服务器端由一个 Servlet 控制器负责监听所有客户端请求,主要是来自客户端的本体查询请求,Servlet 再将请求转交给如图 4 检索模型 Model 类。Model 类是系统的核心类,它封装了 OWL 文档解析接口、SPARQL 查询接口、本体推理引擎、本体检索引擎。客户端用户在可视化浏览本体的过程中,通过鼠标单击图形或键入关键字将本体查询请求提交服务器,服务器则通过核心推理引擎和查询引擎得到查询结果,再通过 Servlet 控制器将其返回给客户端。

4　系统实验结果

本体可视化检索系统界面如图 5 所示,整个客户端可以分 5 个模块。

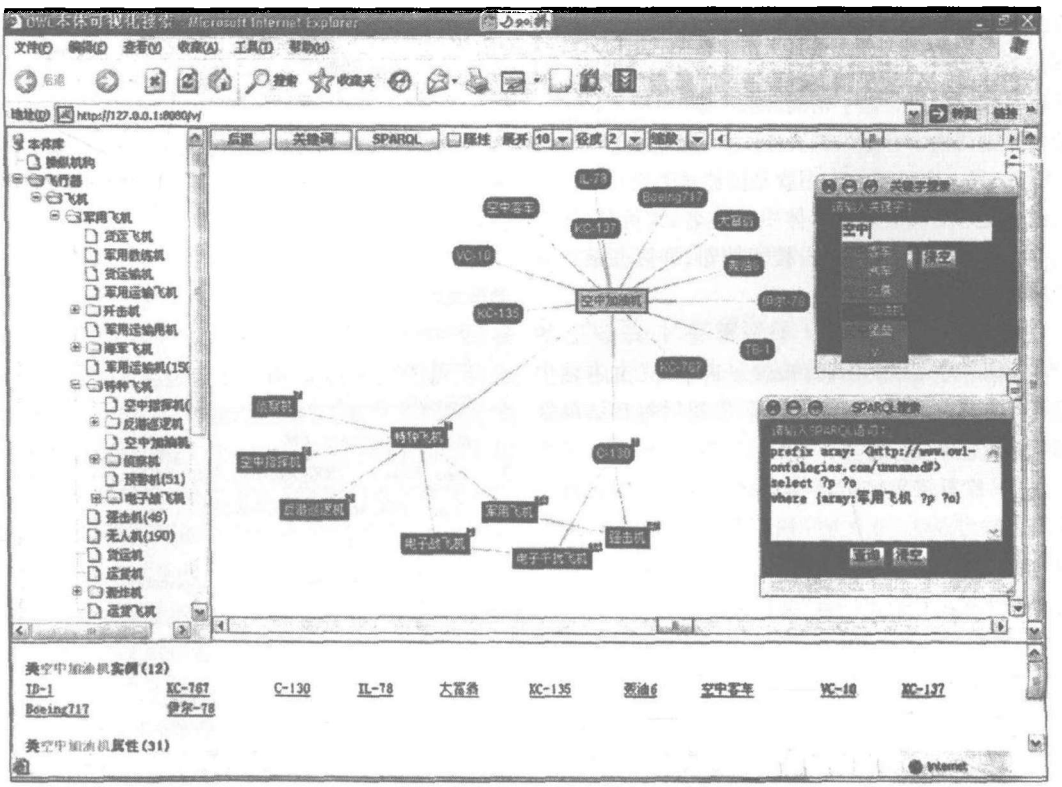

图5　本体可视化检索总界面

(1)树形列表检索入口。左边的树形列表展示本体库的类层次关系,根结点代表整个军用飞机领域本体。除根结点以外,每个结点都代表本体中的一个类,类与子类关系用目录树层级结构显示,文件夹图标表示该类拥有子类,文件图标表示该类除 owl:Nothing 外没有其他子类,同时图标右边的标签值表

示该类在本体中的本地名称，括号内的数字表示该类拥有的实例数目。点击树形列表的某个结点，该结点将高亮显示，同时在右边的图形显示区域中将以该结点为中心，动态重绘一个图形界面。

(2)本体动态图形显示区域。图形显示区域提供本体的可视化导航检索功能。从图5可以看出，图形由一系列的结点和有向边组成，每个结点代表一个类或一个实例，结点的标签值表示类名或实例名，类与实例用不同的形状区分开来，其中直角矩形代表类，圆角矩形代表实例。当单击或双击某一结点时，该结点获得焦点并高亮显示，用以区别其他结点，同时将以该结点为中心，绘制出新的导航图。

有向边用来表示类与类之间、类与实例之间、实例与实例之间的语义关系。鼠标滑过某一边时，该边也将高亮显示，并在边的中部绘制出该边的标签值，指示该边代表的语义关系。

(3)关键词检索入口。单击工具条上"关键字"将弹出"关键字搜索"窗口。该窗口拥有输入提示和自动完成功能，键入关键词时，在文本框的下方即刻弹出下拉列表，列出可能与用户输入相匹配的词汇。目前系统只提供关键字精确匹配搜索，可以说提示功能对关键词检索有至关重要作用，因为它除了提示用户输入，另一个重要的作用就是能够规范用户输入的词汇。提示的词汇来自本体中的类名、实例名、属性名等。输入关键词后，单击搜索按钮，即可在结果显示区域中输出搜索结果。

(4) SPARQL检索入口。单击工具条上的"SPARQL"，弹出SPARQL检索子窗口，在文本域中输入SPARQL语句，单击"查询"按钮后将在结果区域中输出搜索结果。

(5)检索结果输出区域。用户界面的底部是检索结果输出区域，用户选中快捷菜单的"详细信息"，执行关键词检索和SPARQL检索都将在该区域中输出查询结果。在输出的查询结果中，还可单击其中的超链接继续检索，若超链接的文本值是类名(实例名)，单击后将在图形显示区域以该类名(实例名)为中心重绘整个图形区域，这样文本搜索和可视化搜索可相互配合，达到同步搜索目的。而若用户搜索的是本体实例，系统还能检索出与该实例相关的文献并在结果输出区域显示。

5 结论

本可视化检索系统能够将本体中的类层次、属性、实例等语义关系以图形化方式直观显示，实现可视化语义检索，在此基础上又支持关键字、SPARQL的本体检索与文献资源的检索，同时系统生动的动态二维网状图形也给用户带来了不一样的视觉效果。

(1)针对本体推理问题，探讨了Racer和Pellet本体推理机，在此基础上设计了OWL推理引擎，完成了军用飞机领域本体一致性检查和推理。

(2)解决了本体可视化递增性导航浏览难题，动态绘制了本体内部结构及包含的语义关系，实现本体的二维网状的图形可视化检索，动态图形浏览器支持全览、聚焦、移动等功能，这是本文的主要创新点所在。

(3)给出了基于B/S结构本体可视化检索系统的实现方案。系统运行于J2EE平台之上，不但易于将本体与现有的互联网资源整合，而且系统更具实用性，用户只需一个浏览器，便可使用该系统。

(4)本系统具有很高的扩展性和灵活性。对于推理要求不高的本体，可直接实现对任何一个OWL本体的可视化检索。这也是本系统的主要创新点所在。

(5)深入探讨了Ajax技术，借助DWR框架，实现客户端与服务器端的异步通信，使得系统具有很高的响应性。

(6)系统提供的关键词检索、树形列表检索、可视化图形检索等多个检索过程同步，使得系统的互动性更高，用户界面更友好方便。

参考文献

1 Semantic Web. [2006-02-05]. http://www.w3.org/2001/sw
2 TouchGraph Development. [2006-03-10]. http://touchgraph.sourceforge.net
3 Jena Project. [2006-02-16]. http://jena.sourceforge.net
4 Pellet OWL Reasoner. [2006-05-01]. http://www.mindswap.org/2003/pellet
5 Ajax: A New Approach to Web Applications. [2006-02-08]. http://adaptivepath.com/publications/essays/archives/000385.php

丁晟春 南京理工大学经济管理学院讲师。通信地址：南京。邮编210094。

甘利人 南京理工大学经济管理学院教授、博导。通信地址同上。

陈开浩 南京理工大学情报学硕士专业在读研究生。通信地址同上。

(来稿时间：2006-07-31)