OntoThesaurus Web Service API 及其应用研究*

林伟明　　曾新红

(深圳大学图书馆，深圳，518060)

[摘 要] 本文描述了中文叙词表本体共建共享系统(OTCSS)所提供的 Web Service API（OntoThesaurus-API）。以 OntoThesaurus-API 在 OPAC 中的应用为例，详细介绍其使用方法。最后对 OntoThesaurus-API 在其他领域的应用做了进一步的展望。

[关键词] 叙词表；本体；中文叙词表本体；OntoThesaurus-API; Web Service

OntoThesaurus Web Service API and Its Application

Lin Weiming　　Zeng Xinhong

(The Library of Shenzhen University, Shenzhen 518060, China)

[Abstract] This paper presents the Web Service API(OntoThesaurus-API) of OTCSS(OntoThesaurus Co-construction and Sharing System).Taking the application of OntoThesaurus-API in OPAC as an example,the usage of OntoThesaurus-API is expounded. Finally, applying the Ontothesaurus-API in other fields are proposed for further study.

[Keywords] thesaurus　ontology　OntoThesaurus　OntoThesaurus-API　Web Service

1 前言

OntoThesaurus, 中文叙词表本体，是利用本体技术对中文叙词表进行形式化描述和扩展，使中文叙词表成为机器真正可理解的本体知识库，增强了中文叙词表的共享性。本课题运用 Jena、SPARQL 等技术[1]，在文献[2]所定义的中文叙词表 OWL 表示方式基础上，构建了 OntoThesaurus，实现了中文叙词表本体共建共享系统 OTCSS。该系统解决了中文叙词表本体的网络共建共享和动态完善问题，为 OntoThesaurus 提供了检索/获取、一致性检测、修订意见发送、知识提取、词表管理/发布等功能[1-4]。为了实现内嵌式的机辅标引/机辅检索功能，并进一步提供面向应用系统的术语学服务，OTCSS 基于 Apache 的 Axis2[5]开发了一系列 Web Service API（OntoThesaurus-API）。

2 OntoThesaurus-API

Web Service 是一种可以通过网络进行发布、发现、调用的新应用。它使用 WSDL 语言来描述应用操作的接口，通过 XML 消息传递的机制，让其他应用程序通过网络来访问这些操作。其目的是为了不同的程序之间可以进行交互，实现应用程序间的通信。[6] Web Service 的优势在于：1、平台的无关性；2、协议的通用性；3、企业的互操作性；4、功能复用；5、

*本文系国家社科基金项目“基于本体和知识集成实现中文叙词表的升级、共享和动态完善”（05CTQ001）的研究成果之一。

拓展业务；6、服务器的中立性；7、安全的通信。[7] 因此通过 Web Service，可以有效地解决 M2M（Machine to Machine）的互操作问题。如图书馆信息管理系统、数据库、搜索引擎、标签系统等，不管这些系统基于什么平台开发、采用何种开发语言，都可以使用 OntoThesaurus-API 提供的服务，来实现基于中文叙词表本体的智能检索、知识链接和知识服务。

随着 Web Service 的提出与发展，各大程序语言平台（Java, . Net 等）都对 Web Service 提供了支持，并涌现了一批开源的 Web Service 开发框架，如 XFire、Axis2。Axis2 是 Apache 下的第三代 Web 服务引擎，它比前一个版本 Axis 更高效、更模块化、更面向 XML[5]。Axis2 既可以作为独立的 Web 服务平台，也可以内嵌到具体的 Web 应用程序中。OTCSS 系统选用了 Axis2 作为它的 Web Service 开发平台，开发了以下 16 个通用的 Web Service API（另有若干个 API 用于特定的叙词表）。

（1）public Concept[] getConcepts (String key_word,int matching_mode,int search_field)；

该接口的作用是通过各种检索关键字检索并获取叙词。其中，参数key_word为检索关键字（检索值）；参数matching_mode为匹配方式 （0:精确匹配 1:前方一致 2:任意一致 其中任意一致检索只能用于叙词（入口词）项以及英译名项）；参数search_field表示检索项 （0:叙词（入口词） 1:自定义分类号 2:中图分类号 3:英译名）。如果查找结果为空则返回null，否则返回Concept（系统中定义的类，其属性有叙词名、叙词类型）数组。

（2）public String getConceptAndCLCCode (String concept,String separator)；

该接口的作用是通过指定的叙词获取相应的叙词及其中图法分类号。其中，参数concept为指定的叙词；参数separator为指定的分隔符号。该接口返回的结果是以separator作为分隔的叙词及其中图法分类号字符串。

（3）public String getConceptAndNTerm (String concept,String separator)；

该接口的作用是通过指定的叙词获取相应叙词及其入口词（同义词）。其中，参数concept、separator的说明同（2）。该接口返回的结果是以separator作为分隔的叙词及其入口词字符串。

（4）public String getNTermForConcept(String concept,String separator)；

该接口的作用是通过指定的叙词获取相应叙词的入口词。其中，参数concept、separator的说明同(2)。该接口返回的结果是指定叙词的以separator作为分隔的所有入口词字符串。

（5）public String getConceptAndEngCounterpart (String concept,String separator)；

该接口的作用是通过指定的叙词获取相应叙词及其英译名。其中，参数concept、separator的说明同(2)。该接口返回的结果是以separator作为分隔的叙词及其英译名字符串。

（6）public String getEngCounterpartForConcept(String concept,String separator)；

该接口的作用是通过指定的叙词获取相应叙词的英译名。其中，参数concept、separator的说明同(2)。该接口返回的结果是指定叙词的以separator作为分隔的所有英译名字符串。

（7）public String getConceptAndBroader (String concept,String separator)；

该接口的作用是通过指定的叙词获取相应叙词及其直接上位词。其中，参数concept、separator的说明同(2)。该接口返回的结果是以separator作为分隔的叙词及其直接上位词字符串。

（8）public String getBroaderByLevel(String concept,String separator,int level)；

该接口的作用是通过指定的叙词和指定的级数获取相应叙词的指定级数以内的所有上位词。其中，参数concept为指定的叙词；参数separator为分隔符号；参数level为所获取上位词的级数 ，从1开始（直接上位词级数为1）。该接口返回的结果是指定叙词的指定级数以内的所有上位词，并以separator作为分隔的字符串。

（9）public String getBroaderByPath(String concept,String separator, String subproperty, int level)；

该接口的作用是通过指定的叙词和指定的上位词子关系、级数获取相应叙词的指定级数以内的指

定子关系上位词。其中，参数concept为指定的叙词；参数separator为分隔符号；参数subproperty为指定需要获取的上位词子关系属性（类属/实例/整体_部分）；level为获取子关系上位词的级数 ，从1开始（直接子关系上位词级数为1）。该接口返回的结果是指定叙词、指定级数以内的所有指定子关系上位词，并以separator作为分隔的字符串。

（10）public String getConceptAndNarrower(String concept,String separator)；

该接口的作用是通过指定的叙词获取相应叙词及其直接下位词。其中，参数concept、separator的说明同(2)。该接口返回的结果是以separator作为分隔的叙词及其直接下位词字符串。

（11）public String getNarrowerByLevel(String concept,String separator,int level)；

该接口的作用是通过指定的叙词和指定的级数获取相应叙词的指定级数以内的所有下位词。其中，参数concept为指定的叙词；参数separator为分隔符号；参数level为所获取下位词的级数 ，从1开始（直接下位词级数为1）。该接口返回的结果是指定叙词的指定级数以内的所有下位词，并以separator作为分隔的字符串。

（12）public String getNarrowerByPath(String concept,String separator, String subproperty, int level)；

该接口的作用是通过指定的叙词和指定的下位词子关系、级数获取相应叙词的指定级数以内的指定子关系下位词。其中，参数concept为指定的叙词；参数separator为分隔符号；参数subproperty为指定需要获取的下位词子关系属性（类属/实例/整体_部分）；level为获取子关系下位词的级数 ，从1开始（直接子关系下位词级数为1）。该接口返回的结果是指定叙词、指定级数以内的所有指定子关系下位词，并以separator作为分隔的字符串。

（13）public String getRelated(String concept,String separator)；

该接口的作用是通过指定的叙词获取相应叙词的相关词。其中，参数concept、separator的说明同(2)。该接口返回的结果是指定叙词的以separator作为分隔的所有相关词字符串。

（14）public String getRelatedByPath(String concept,String separator,string subproperty)；

该接口的作用是通过指定的叙词和指定的相关关系子关系获取相应叙词的指定子关系相关词。其中，参数concept为指定的叙词；参数separator为分隔符号；参数subproperty为指定需要获取的相关关系的子关系。该接口返回的结果是指定叙词的所有指定子关系相关词，并以separator作为分隔的字符串。

（15）public ConceptProperty[] getConceptProperties(String concept)；

该接口的作用是获取指定叙词的所有属性-属性值（即叙词款目信息）。其中，参数concept为指定的叙词。该接口返回的结果是：如果叙词不存在或该叙词不存在任何属性-属性值，返回null，否则返回ConceptProperty（系统中定义的类，其属性有Concept数组、Property、String数组，其中Property表示叙词款目具体属性如中图法分类号、上位词；而该属性所对应的多个属性值则根据属性值类型为叙词（入口词）或字符串值分别写入Concept数组或String数组中）数组。

（16）public Property[] getProperties()；

该接口的作用是获取OntoThesaurus中支持的所有属性名称及其comment、label等信息。Property为系统中定义的类，其属性有name、label、comment，分别用于表示属性名、属性标签、属性注释。

3 OntoThesaurus-API 的使用方法

3.1 Web Service 的调用方法简介

Web Service 的使用比较简单。目前一般采用以下三种调用方法（程序员无需了解具体的 SOAP 协议）：

（1）使用程序语言平台本身提供的对 Web Service 调用的支持。这种方法可以使程序员不需要自己写 SOAP 信息，就可以像调用普通函数一样去调用 Web Service 接口。如在. NET 开发平台中，通过添加 Web 引用，填写调用的 Web Service 的 WSDL 的 URL 地址，系统就

会自动创建一个代理类；程序员调用 Web Service 接口时，只需要在程序中像使用自己定义的类一样去调用代理类的方法。

（2）通过其他公司或机构提供的组件进行访问，如微软的 htc 组件，Axis2 开发包等。这些组件可以对服务方提供的描述文件 WSDL 进行解析。调用服务时，生成恰当的 SOAP 请求消息发向服务端，等待服务端返回 SOAP 响应消息并进行解析获取返回值。

（3）直接通过 URL 地址访问。该 URL 地址的格式为：服务地址/函数名?参数 1=参数 1 的值&参数 2=参数 2 的值&…&参数 n=参数 n 的值，其中参数 1，参数 2，…，参数 n 均是函数的形式参数名。当 Web 服务引擎（如 Axis2 引擎）接收到此 URL 地址后，会把该地址交给控制类（如 AxisServlet）进行处理，从中获取客户的 HTTP 请求信息，并调用与地址中函数名同名的服务，最后把调用结果封装成 SOAP 消息流返回到客户端。这样，程序员就可以使用类似调用 Ajax 的方法一样去调用 Web Service。

3.2 OntoThesaurus-API 的应用示例

下面我们以图书馆信息管理系统中的公共联机书目查询系统 OPAC 为例，详细介绍 OntoThesaurus-API 的使用方法。其他应用系统可参照此过程开发自己的具体应用。

在 OPAC 中，可以利用 OTCSS 提供的 API，进行扩展性的智能检索应用。比如规范化检索：对于用户输入的检索词，系统运用 OntoThesaurus-API 进行检测，如果检索词为某一个主题词的入口词，系统将会使用该主题词代替此检索词进行检索，否则就使用原检索词进行检索。下面以此为例来说明如何调用 OntoThesaurus-API 来增强 OPAC 的服务。在本例中，采用直接访问 URL 的方式调用 Web Service。

（1）首先需要得知检索词是否为入口词，并将其对应的叙词作为新的检索词。OntoThesaurus-API 的第一个 API 可以通过各种检索关键字检索并获取叙词，其中有一个检索途径是通过入口词或叙词进行精确检索。该 Web Service API 的 URL 地址为

http://210.39.15.167:8080/ThesaurusProjectForCCT/services/ThesaurusService/getConcepts?key_word={0}& matching_mode=0&search_field=0

其中{0}处填写检索词。

（2）.NET 平台中提供了 System.Net.HttpWebRequest 来模拟页面访问 URL 的方法，并通过 System.Net.HttpWebResponse 类取得响应的页面内容。在本例中，就是使用这种方法对（1）中的 URL 地址进行访问。在 OTCSS 系统中，利用 Axis2 作为它的 Web 服务引擎，提供的 Web 服务对中文的处理默认是 ISO8859-1 编码模式。因此在使用(1)前需要对检索词进行 ISO8859-1 转码：

```
byte[] tmp = Encoding.Default.GetBytes(value);
string tmpvalue = Encoding.GetEncoding("ISO8859-1").GetString(tmp);
```

再把 tmpvalue 的值作为 key_word 的参数值，利用 HttpWebRequest 类进行访问：

```
HttpWebRequest reqforsearch = (HttpWebRequest)WebRequest.Create(url);
```

接着利用 HttpWebResponse 类获得页面响应：

```
HttpWebResponse resforsearch = (HttpWebResponse)reqforsearch.GetResponse();
```

（3）判断页面的响应是否成功。接着利用.NET 平台中提供的 System.IO.Stream,System.IO.StreamReader 来获取(2)中得到的页面响应的数据流：

```
Stream stream = resforsearch.GetResponseStream();
```

```
srforsearch = new StreamReader(stream, System.Text.Encoding.GetEncoding("utf-8"));
string s = srforsearch.ReadToEnd();
```

（4）Web Service 返回的结果是以 XML 格式进行展示的。所以接下来根据具体需求对这些 XML 返回结果做解析，这里我们主要采用正则表达式的方法去做解析：

```
string concept_regex = "<ax21:label>(.*?)</ax21:label>";
Match concept_match = Regex.Match(s, concept_regex);
if (concept_match.Success)
{
        result = concept_match.Groups[1].Value;
}
```

解析出来的结果为本例中需要使用的正式主题词。

（5）利用这个正式主题词代替输入的检索词，调用 OPAC 中的检索方法进行主题词检索，实现规范化检索功能。

在本例中，如果能找到检索词的正式主题词，或者该检索词就是一个正式主题词，还可以运用 OntoThesaurus-API 的第 15 个 API ：public ConceptProperty[] getConceptProperties (String concept)，把这个主题词的所有款目信息输出到检索结果页面，供用户浏览，用户还可以利用这个主题词的上下位词等款目信息继续进行检索。

调用的步骤与上述步骤几乎一致，不同之处在于调用的 URL 地址以及对返回结果的解析过程。调用的 URL 地址为：

http://210.39.15.167:8080/ThesaurusProjectForCCT/services/ThesaurusService/getConceptProperties?concept={0}

其中{0}处填写叙词。

解析返回结果时也是通过正则表达式进行解析，以下是解析用的正则表达式：

```
string concept_regex = "<ns:return type=\"service.ConceptProperty\">(.*?)</ns:return>";
string property_label_regex = "<ax21:property type=\"service.Property\"><ax21:comment>.*?</ax21:comment>
<ax21:label>(.*?)</ax21:label>";
string property_value_regex_fdp = "<ax21:literal>(.*?)</ax21:literal>";
string property_value_regex_fop = "<ax21:concepts type=\"service.Concept\"><ax21:label>(.*?)</ax21:label>.*?
</ax21:concepts>";
```

最后根据解析出来的结果得出叙词的完整款目。

例如，检索“公断”这个词，系统利用 OntoThesaurus-API 找出“公断”的正式主题词为“仲裁”，并使用“仲裁”进行主题词检索，显示检索结果如图 1 所示。读者可点击“完整叙词款目”查看 “仲裁”的叙词款目信息，如图 2 所示。

图 1 使用“公断”进行规范化检索后的结果

图 2 点击图 1 的“完整叙词款目”后显示的叙词款目

在上述的例子中，调用 Web Service 时也可以使用 3.1 中所提到的其他方法进行实现，如在.NET 平台中使用 Web 引用的方法。首先点击“添加 Web 引用”，填写 URL 地址为：

http://210.39.15.167:8080/ ThesaurusProjectForCCT /services/ThesaurusService?wsdl

并填写 Web 引用名（如 WebReference）；系统会自动根据这个 WSDL 文件生成相应的代理类；接着在程序中通过这些代理类来调用 Web Service 接口，如调用第 2 节中所述的第一个 API：

```
byte[] tmp = Encoding.Default.GetBytes(value);
string tmpvalue = Encoding.GetEncoding("ISO8859-1").GetString(tmp);
TestWSClient.WebReference.ThesaurusService ts = new TestWSClient.WebReference.ThesaurusService();
TestWSClient.WebReference.Concept[] result = ts.getConcepts(tmpvalue, 0, true, 0, true);
……
```

在使用这种方法时，同样需要先对传入的中文参数进行转码。

3.3 OntoThesaurus-API 在其他方面的应用

利用 OntoThesaurus-API, 可以在 OPAC 中实现规范化检索，分类号、英译名扩展检索，上位词、下位词、相关词（或它们的子关系词）的扩展检索等，为读者提供更准确、更强大的检索服务。

在图书馆信息管理系统的编目子系统中，OntoThesaurus-API 可以为标引员提供内嵌式的辅助标引功能，帮助标引员使用更为准确的主题词进行更快捷的标引。

此外，OntoThesaurus-API 还可以应用在其他领域。如在 CNKI 等数据库检索系统中，在主题词、关键词的检索功能上，可以使用它来辅助完成规范化检索、上下位扩展检索等功能；在搜索引擎中，可以利用此服务获取更多的检索词形态；在众多支持 Tag 的应用程序中，可以使用 OntoThesaurus-API 为用户输入的标签进行规范化的提示，使用户输入的标签更为规范化；在面向主题词的数据挖掘中，运用 OntoThesaurus-API 可以对主题词及其款目信息进行挖掘、统计；在机器学习中，通过 OntoThesaurus-API，可以提供同义词辨析机制和主题词分析机制，进行与主题词相关的信息抽取等等。

上述所有的应用都可以通过 Web Service 调用的标准方法，在不同的应用平台上进行实现。

4 结　语

基于 Web Service 技术开发的 OntoThesaurus-API 可以非常广泛地应用到其他应用系统中，促进新信息环境下知识服务的发展。

本文通过 OPAC 应用实例详细阐述了这些 OntoThesaurus Web Service API 的使用方法。相关领域应用系统的开发者可以参考这些方法，根据具体的 OTCSS 系统提供的 Web Service API 开发自己的具体应用。OntoThesaurus-API 今后还可以针对新的需求，扩展提供其他服务，如支持返回 JSON 数据格式，支持叙词修订信息采集 Web Service 等。

参考文献：

[1] 曾新红等. 中文叙词表本体共建共享系统研究[J]. 情报学报, 2008, 27(3):386-394.

[2] 曾新红. 中文叙词表本体——叙词表与本体的融合[J]. 现代图书情报技术, 2009(1):34-43.

[3] 曾新红, 林伟明, 明仲. 中文叙词表本体的检索实现及其术语学服务研究[J]. 现代图书情报技术, 2008(2):8-13.

[4] 曾新红, 林伟明, 明仲. 中文叙词表本体一致性检测机制研究与实现[J]. 现代图书情报技术, 2008(5):1-9.

[5] Axis2[EB/OL]. [2009-07-07]. http://ws.apache.org/axis2/index.html.

[6] 柴晓路, 梁宇奇. Web Services 技术、架构和应用[M]. 北京:电子工业出版社, 2003:12-18.

[7] Ashish Banerjee 等著. C#Web 服务高级编程[M]. 康博译. 北京：清华大学出版社, 2002:5-8.

林伟明，男，深圳大学图书馆，研究方向：计算机应用技术。
曾新红，女，深圳大学图书馆，研究方向：知识组织与知识管理，数字图书馆相关技术。
E-mail：zengxh@szu.edu.cn
通讯地址：广东深圳南山区 深圳大学图书馆　邮政编码：518060