第34卷 第15期 计 算 机 工 程 2008年8月
Vol.34 No.15 Computer Engineering August 2008
·软件技术与数据库· 文章编号：1000—3428(2008)15—0095—02 文献标识码：A 中图分类号：TP391

基于 Nutch 的 XML 网站全文搜索引擎实现

吴敏琦，丁岳伟
(上海理工大学计算机工程学院，上海 200093)

摘 要：普通搜索引擎的网页抓取程序只能理解常见 HTML 标签，无法对 XML 网站的内容做有效解析。该文建立一个包含动态自定义标签的纯 XML 网站，提出借助 XSL 样式信息帮助网页抓取程序理解 XML 网页标签含义的方案，实现了基于 Nutch 的 XML 网站全文搜索引擎。

关键词：XML 信息检索；可扩展样式表语言转换；基于 Nutch 的搜索引擎

Implementation of XML Website Complete Text Search Engine Based on Nutch

WU Min-qi, DING Yue-wei
(College of Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093)

【Abstract】 General search engine spiders can understand only common HTML tags, and can't parser information from XML Web sites efficiently. This paper proposes a strategy of using XSL to help spiders to understand the structure of XML pages. Based on this strategy, a pure XML Website is set up, and a search engine based on Nutch which is able to parse XML Website content correctly is realized.

【Key words】 XML information retrieval; eXtensible Stylesheet Language Transformations(XSLT); search engine based on Nutch

1 概述

随着 Web 的迅速发展，互联网上的信息量正快速增长。我国互联网网页数至 2006 年 12 月已达 4.47×10^9 个，网页字节数达 122 305 737 MB。因此，必须利用一定手段从海量信息中获取有价值的信息。目录式搜索引擎、关键词搜索引擎及混合型搜索引擎在传统信息检索系统的基础上得到快速发展。在信息集散地从图书馆和资料室逐渐迁移到互联网的过程中，第 1 代 Web 语言——超文本标记语言(HyperText Markup Language, HTML)发挥了重要作用。作为一种页面描述语言，HTML 的简洁及跨平台特性极大方便了用户获取信息，但 HTML 过于精简的语法导致其难以表现复杂形式，且存在难以扩展、交互性差、语义性差及单向超链接等缺点，难以在电子数据交换、数据库或搜索引擎等领域被深入应用。

万维网联盟(World Wide Web Consortium, W3C)于 1998 年 2 月推出可扩展性标记语言(eXtensible Markup Language, XML)作为因特网上数据表示与交换的标准。XML 自推出以来，在数据交换、数据存储等多个领域得到全面发展，但在 Web 表示方面，XML 仍处于初级阶段。大部分 Web 网站仍使用 HTML 作为客户端描述语言。

W3C 在 2007 年 3 月发表的章程中提到，互联网的发展方向是以 XML 取代 HTML，并提出将加速该进化过程。在互联网从 HTML 向 XML 发展的过程中，现有搜索引擎技术将出现重大变革，适应并充分利用纯 XML 的网络环境将是搜索引擎发展重要方向之一。

2 相关工作及技术

XML 信息检索(Information Retrival, IR)是近期发展最迅速的 IR 领域之一，INEX 是现有最著名的 XML 检索与评价组织。文献[1]论述了 INEX 中的相关课题。INEX 的议题主要包括 XML Ad-hoc 检索、交互式 XML 检索、多媒体 XML 信息检索、XML 相关反馈检索、异构 XML 检索、XML 文档挖掘和基于自然语言处理的 XML 检索。目前大量研究集中在传统信息检索领域，以 INNEX 2005 为例，有近一半的论文集中在 XML Ad-hoc 领域，即对图书馆馆藏静态文档集检索的一种模拟，XML 的作用是实现数据存储。XML 信息检索是实现内容和结构的双重检索，文献[2-3]提出一种基于 XPath 的结构化查询方法，增加类似“about”的模糊查询函数来满足 XML 信息检索的非结构化查询全文检索要求。

XML 网站全文搜索引擎实现的相关资料很少，主要原因如下：(1)一些商业性 XML 搜索引擎的相应技术暂时没有被公布；(2)由于各种原因(如一些浏览器对 XML 网页及 XSL 的支持滞后等)，当前的纯 XML 网站较少，因此多数实践是基于类似图书馆馆藏静态 XML 文档等实验集开展的。

3 纯 XML 网站模拟

由于浏览器只能识别 HTML 标签，无法识别 JSP 或 ASP 的标签，因此无论 JSP, ASP 或 PHP，服务器端接收用户端 request 后，参数化页面变量并返回给客户端的仍然是 HTML 页面。而 HTML 的根本缺陷是它既包含需要显示的数据，又包含这些数据应如何展示的页面设计，造成数据和外观混合，直接影响了一些非桌面互联网终端，如手机、PDA 等，网页浏览效果很差。

XML 网站可以很好地实现文档内容和外观设计的完全分离。XML 文档只负责存储数据，外观显示由可扩展样式表语言(eXtensible Stylesheet Language, XSL)或 CSS 负责。不同

作者简介：吴敏琦(1980－)，男，硕士，主研方向：信息检索，信息安全；丁岳伟，教授

收稿日期：2007-10-12 **E-mail：**wuminqi@163.com

上网终端接收到相同的 XML 文档，只是针对不同客户端的样式文件不同。如果要打印一份网页清单，只要替换一份样式文档，无须重新排版一个新的网页。

3.1 XSL 和 XSLT

XSL 最早由 W3C 于 1999 年提出，在 XSL 标准的发展过程中，原始 XSL 标准被划分为 2 个单独的规范文档：可扩展样式表语言转换(eXtensible Stylesheet Language Transformations, XSLT)和 XSL。XSL 是一种高级格式化语言，用于定义如何显示数据；XSLT 提供一套规则，用于将一组元素描述的 XML 数据转换为另一组元素描述的文档，或将该数据转换为一种自定义文本格式(如需要打印的工资单)。

XSLT 的根本设计目的是转换文档的词汇表，如图 1，将 XSLT 样式表应用于 XML 源文档上，将产生一个结果文档。

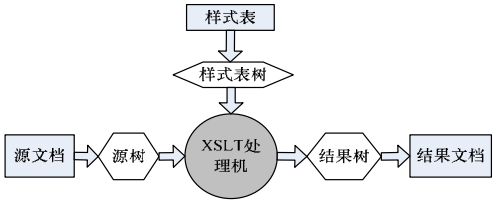

图 1　XSLT 处理机在创建输出树之前数据树的创建

3.2 XML 网站

本次实验模拟的纯 XML 网站是一个部署在 Tomcat 6.0.10 上的静态 Web 网站，包括 33 个相互链接的 XML 网页、1 个 XSL 样式表、1 个 DTD 文档和 1 幅 gif 图片。

XML 网页的内容来自 SUN 的 Java EE 5 的教材，首页的部分内容如下：

```
<?xml version="1.0" encoding="iso-8859-1"?>
<?xml-stylesheet href="site.xsl" type="text/xsl"?>
<spec w3c-doctype="rec">
<header>
<title>The Java(tm) EE 5 Tutorial</title>
<version>1.0 (Second Edition)</version>
<w3c-designation>REC-xml-&iso6.doc.date;</w3c-designation>
<w3c-doctype>Sun MicroSystems</w3c-doctype>
<abstract>
<p>This tutorial is intended for programmers who are interested in developing and deploying Java EE 5 applications on the Sun Java System Application Server Platform Edition 9.</p>
</abstract>
…
```

通过给源文件指定相应的 XSL 样式表，用户可以使用浏览器看到一个由标题、摘要和正文 3 部分组成的首页，页面风格与 http://java.sun.com/javaee/5/docs/tutorial/doc/相似。其他 XML 页面可通过首页链接访问，它们具有相似外观。

4　Nutch 的全文检索机制

Nutch 是基于全文检索模块 Lucene 的一个开源搜索引擎。其中，Lucene 是 Apache 软件基金会下的一个开源全文检索引擎工具包，提供了查询引擎和索引引擎。Nutch 可分为如下 3 个部分[4]：

(1)网页收集(fetch)。网页收集程序通过定期收集方式或增量收集方式从 URL 列表中选择要收集页面的 URL，通过此 URL 访问网页并将网页抓取到本地。

(2)建立索引(index)。索引建立程序将抓取的网页进行分词和过滤，将文档分隔成一个词干的集合，以关键词作为索引建立或动态维护倒排文档，即关键词 A 出现在哪些文档中的对应关系。在 Nutch 中，这样的索引文档由很多个小索引文档组合而成。

(3)查询(searcher)。查询模块接收用户的查询输入，通过分词和过滤，分隔成查询关键词组合，根据这些关键词到索引库中匹配相应网页，并按排序算法对匹配结果进行排序，返回结果。

Nutch 的整体框架如图 2 所示。

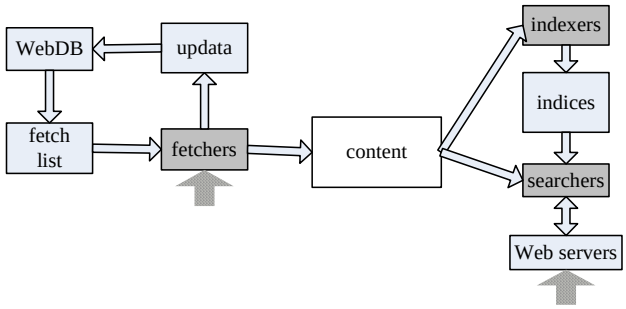

图 2　Nutch 的整体框架

5　基于 Nutch 的 XML 搜索引擎

虽然 Nutch 自 2003 年推出以来发展迅速且受到开源社区的广泛关注，但 Nutch 本身不能直接完成纯 XML 网站的检索。主要原因是纯 XML 网页的各个标签是自定义的一组标签。传统网页搜集模块(又称为 Spider 或 Bot 等)通过 HTML 标签建立网页的 DOM Tree。当网页变成了 XML 后，DOM Tree 仍然可以建立，但其中的标签无法被网页搜集模块识别。

笔者在使用 Nutch 0.9 对之前建立的纯 XML 模拟网站进行检索实验时，以首页 http://localhost:8080/xml/javaeetutorial.xml 作为检索初始页，只能获得首页，网页搜集模块无法识别首页中<loc href=”somepage.xml”>元素的含义，因此，无法自动爬行到下一链接中，导致抓取失败。文献[5]提出一种基于文档类型定义(Document Type Define, DTD)的 XML 内容检索方法，通过 DTD 的上下文关系帮助用户提高检索效率。本文借助 XML 之外的信息(如 XSL 中的信息)帮助网页抓取程序理解 XML 网页中自定义标签的含义。任何 XML 网页最终都需要结合一定样式来生成浏览器可以理解的视图。

有如下 3 种方式可以使网页抓取程序正确理解 XML 网页中的标签含义：

(1)人工分析＋硬编码。Nutch 的 HTMLParser 采用此方式，因为 HTML 的标签是固定的，所以哪些标签代表链接关系，哪些标签代表显示逻辑，可以硬编码到代码中。但用这种机制处理 XML 网页会使代码难以被维护，特别是当样式表发生变化时。

(2)人工分析＋配置文件。使用配置文件方式维护 XML 网页标签的词汇表，可以使程序更灵活地应对标签的增删变更，且无须重新编译源代码。但这种方案依赖人工分析，当 XML 网页采用相同的 XSL 样式文件时可以应对，但如果各个 XML 网页采用不同 XSL 样式文件和不同标签体系，则人工分析基本不可能实现，而对于 XML 搜索引擎而言，这种情况相当普遍。

(3)程序动态解析。程序动态解析是最灵活的处理方式，即在网页抓取程序解析 XML 网页的同时解析 XSL 样式文件，从而获得各个标签的语义。比如，在本文模拟网站的样式文件中，可以找到标签<loc>的含义，代码如下：

（下转第 107 页）