第34卷 第20期 计 算 机 工 程 2008年10月
Vol.34 No.20 Computer Engineering October 2008
·软件技术与数据库· 文章编号：1000—3428(2008)20—0055—03 文献标识码：A 中图分类号：TP393

基于模板流程配置的Web信息抽取

刘 辉，陈静玉，徐学洲
(西安电子科技大学软件工程研究所，西安 710071)

摘 要：针对Web信息抽取中存在的包装器构造复杂及抽取精度等问题，提出并实现了一种基于模板流程配置的Web信息抽取框架。将用户请求、访问和获取Web页面的动作进行分解，抽取其中的动作模式，并映射到流程配置模板中的节点。通过流程解析器对用户创建的流程配置XML描述文档进行解析，抽取感兴趣的信息。试验结果表明，系统可快速、准确地实现抽取。

关键词：Web信息抽取；模板流程配置；包装器；框架

Web Information Extraction Based on Template Flow Configuration

LIU Hui, CHEN Jing-yu, XU Xue-zhou
(Software Engineering Institute, Xidian University, Xi'an 710071)

【Abstract】 To solve the existing problems such as the complexity to constructing wrappers and extracting precision, a Web extraction framework based on template flow configuration is presented and accomplished. Decompose the actions of requiring, accessing and obtaining of users, and extract those action patterns, reflecting them into the flow configuration template as nodes. Flow interpreter will interpret the flow configuration description XML document which is created by users, and then extract the information which is interesting to them. Experimental result indicates that the framework can quickly and correctly realize the extraction.

【Key words】 Web information extraction; template flow configuration; wrapper; framework

1 概述

Web信息抽取的本质是从Web页面所包含的无结构或半结构的信息中识别出感兴趣的信息，并将其转化为更为结构化、语义更为清晰的格式。目前对网页进行信息抽取有许多种技术手段，主要包括基于归纳学习的信息抽取、基于HTML结构解析的信息抽取、基于Web查询的信息抽取、基于自然语言处理的信息抽取、基于模型的信息抽取和基于本体的信息抽取[1]。其中，基于HTML结构解析的信息抽取的特点是将Web文档转换成反映HTML文件层次结构的解析DOM树，通过自动或半自动的方式产生抽取规则，典型的系统有W4F[2]，XWRAP[3]，Lixto[4]等。W4F具有较强的表达能力，但是需要理解复杂的描述抽取规则及语法。XWrap的抽取规则相对来说要简单一些，但表达力较弱，语义项的抽取使用过程控制的方式，缺乏灵活性。Lixto使用Elog，具有较好的表达能力，但实现较为困难。

近年来基于HTML的抽取的研究非常多，也取得了很多成果。其中文献[5]具有较高的回召率和准确率，可以简化构造Wrapper的操作，但由于采用多达150多条规则，其构造本身复杂性及规则维护是一个需要考虑的问题。文献[6]基于一些固定的模板，利用基于实例的方法来获取包装器，但模板是不确定的，因此，具有一定的约束性。

针对以上抽取方法的不足，本文提出了基于模板流程配置的Web信息抽取方法，通过在模板流程中构造对欲抽取的Web页面请求、访问及获取的动作及页面内容DOM树的路径抽取等动作，由解析器来解析流程配置动作，生成XSLT并执行抽取。其主要特点是将Web信息抽取时的动作流程进行分解，抽象出各个动作的子动作，然后把这些动作封装成标准流程模板节点，并且对各子动作预定义相应的解析操作。用户对要抽取的数据项简单地创建配置抽取流程描述XML文档，然后流程解析器分解并解析流程配置描述的抽取动作，对页面逐个进行抽取并将结果整合。

2 流程配置的模板设计

根据用户访问Web数据源时的动作，一个典型的Web信息抽取过程由以下活动构成：获取Web查询页面，填写查询表单，发送查询请求，返回结果页集，格式化返回的结果。考虑到数据流的转化，详细的数据流程为：

(1)由初始的URL获取所有以分页的形式显示的Web页面结果集，将这些页面下载到本地并进行HTML页面的清洗，转化为更规范的XHTML(Extensible HTML)文档。

(2)依照系统规定的流程配置模板Schema，由用户来创建抽取流程XML描述文件。

(3)通过抽取流程解析器来完成对XML文件的解析，将抽取的结果合并并以XML文件形式存储。

2.1 流程配置模板的节点内容

根据以上分析，系统将数据抽取的动作抽象出来，通过预定义模板流程配置，设置模式Schema文件中的节点来定义各个抽取动作以及子节点中描述的更为详细的抽取子动作，抽取动作可以嵌套使用。基于此，对系统的抽取流程配置模板Schema文件的定义尤为关键，用户正是根据此文件来创建符合Schema规范的抽取流程XML描述文件，通过

作者简介：刘 辉(1983－)，男，硕士研究生，主研方向：数据库，信息抽取；陈静玉，讲师、博士研究生；徐学洲，教授
收稿日期：2007-12-24 **E-mail：**hickey.liu@gmail.com

XML 抽取流程解析器完成解析抽取的任务来获得其感兴趣的信息。Schema 模板包含的主要节点信息有：

(1)Web 页面数据内容的类型(text 和 file)

对于 Web 页面，一般在 html 的 head 标签内描述内容为 text 或 file 类型信息。若为 text 类型，说明内容信息是一些文本或 html 信息；如果是 file，则说明信息内容为二进制文件。如<meta http-equiv= Content-Type content="text/html>，说明内容类型为 text 或 html。而对于一些内嵌的文件，如 pdf 等格式文件则以二进制文件格式存储。

(2)URL 解析(http 和参数 http-param)

对于用户给定的 URL，相应的页面可能会以 HTTP 或 HTTPS 协议的形式进行传输。由于两种传输协议的 URL 字符串中，“ :// ”后面对应的字符串没有区别，因此系统可以进行相同的处理。当 Web 页面存在 HTML FORM 时，由于爬虫不能处理使用 GET 方法的链接，可采用 GET-to-POST 方法来转化问题。在 HTTP 会话的过程中，通过客户端代理发送 GET/POST 类型的 HTTP 请求，服务器端程序执行请求，发送 HTTP 响应给客户端。在 Web 信息抽取系统中，查询参数封装在 HTTP 请求中，通过响应来获取页面数据。

(3)查询表达式(XPath, XQuey)

有些包装器语言(如 W4F 中 HTML 使用的抽取语言)需要使用 HTML 绝对路径路径来指向需要抽取的数据项。XPath 和 XQuey 经常用在基于 XML 的查询处理中，能够避免绝对路径的弱点，准确定位到节点位置，以获取所需信息。对于 XPath 表达式，由于采用轴处理的方式，能方便地查询对应的 DOM 树节点信息。使用 XQuery 时，还需增加查询条件信息，以进行 XML 查询。

(4)匹配模式正则表达式(RegExp)

由于 XHTML 结构是基于 DOM 树结构，Web 页面中数据项与 DOM 树中的节点相对应。页面中数据项信息的查询即可转化为 DOM 树节点内容的查询。XPath 及 XQuey 中对 DOM 树中节点内容信息经常要做匹配操作，采用正则表达式可以快速地完成匹配任务。对于正则表达式活动节点，还要记录要处理的源信息，以及经过正则表达式处理后的结果。

2.2 文法推导流程配置

流程配置模板 Schema 文档信息，包含上面所分析的主要抽取活动内容，然后系统将这些抽取活动组织并设计为对应的节点元素。通过文法表达式对 Schema 进行推导，合理地进行组织，以形成抽取活动模板节点。将整个抽取活动元素作为一个文法表达式，进行抽取活动的推导：

```
CONFIG-STMT -> URL + TEMPLATE-STMT
TEMPLATE-STMT -> [EXTRACT-STMT1
EXTRACT-STMT2 EXTRACT-STMT3 ...]
| TEMPLATE-STMT
```

不同的 EXTRACT-STMT，分别对应不同的子句，包括抽取过程中的主要逻辑。

下面以条件分支 IF ELSE 语句为例，详细地说明其文法表达式及其对应的 Schema 设计：

```
IF-STMT -> IF STMT
        | IF EXPR STMT ELSE STMT
```

其中，EXPR 可以是系统中使用的表达式，在此作为判断条件表达；STMT 可以为所有子句的文法。由此可看出，子句可嵌套。将条件分支语句的 Schema 对应于的配置节点，对应的 Schema 设计如下：

```
<xs:element name="IF_ELSE_SENTENCE">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="IF" maxOccurs="unbounded">
<xs:complexType>
  <xs:group ref="Processors"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:attribute name="condition"
    use="required"/>
    <xs:attribute name="id"/>
  </xs:complexType>
 </xs:element>
  <xs:element name="ELSE" minOccurs="0">
   <xs:complexType>
    <xs:group ref="Processors"
        minOccurs="0" maxOccurs="unbounded"/>
     <xs:attribute name="id"/>
    </xs:complexType>
   </xs:element>
  </xs:sequence>
 <xs:attribute name="id"/>
</xs:complexType>
</xs:element>
```

对应的 XSD(XML Schemas Definition)显示片断见图 1。

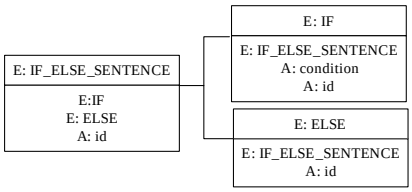

图 1　IF-STMT 文法对应的 XSD 显示

其中，E 表示 Element；A 表示 Attribute。

条件 IF_ELSE_SENTENCE 对应的 XML 配置如下：

```
<IF_ELSE_SENTENCE>
    <IF CONDITION='CONDITION'>
<!—PROCESS OF OTHER SENTENCE --!>
</IF>
</IF_ELSE_SENTENCE >
```

2.3 流程配置模板的设置

在进行 Web 信息抽取时，使用 Web 信息抽取过程描述文件来保存抽取规则定义等信息。该文件是结构良好的 XML 文档，主要记录的信息是依照 Schema 中规定的格式内容来创建，并符合此 Schema。XML 文档中不仅包括用户指定的 URL，还可能需要配置用户的登录信息以供 Proxy 登录 Web 站点操作使用。其中核心内容还是根据 Schema 创建的抽取活动，包括对待抽取的信息块的搜索、定位、匹配及转换等操作。用户也可对待抽取信息的显示进行灵活的组织。

3 抽取框架及核心解析算法

在以上模板流程配置的基础上，笔者设计了基于模板流程配置的 Web 信息抽取的流程框架，如图 2 所示。

系统执行的具体过程如下：

(1)通过抽取流程 Schema 模板构造抽取流程配置 XML 描述文档。

(2)抽取流程解析器通过解析抽取流程配置文档，获取

Web 页面的请求动作，通过客户端代理获取数据并进行预处理，解析预处理后的页面 DOM 树并存储其结果。

(3)通过解析流程配置文档中其他节点的配置信息，执行相应抽取动作，并将抽取结果保存为 XML 格式的数据文件。

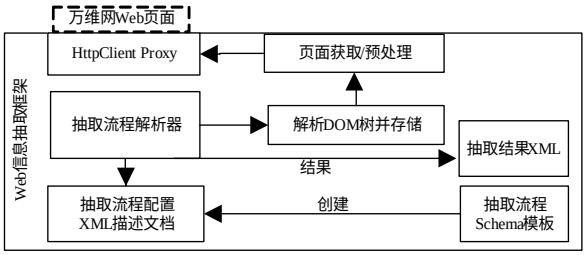

图 2　Web 信息抽取框架

抽取流程解析器是系统中一个关键的处理部分，其主要功能是解析用户根据流程模板 Schema 配置的 XML 流程抽取描述文档。解析器要对 XML 文档进行遍历，提取 XML 中节点所描述的抽取活动。将这些活动转化为相应的类实体，节点元素的内容及属性以类实体的属性进行存储。解析时将 XML 文档中的变量用实体代替，用哈希表来存储变量与实体信息。对于抽取活动的节点，分别处理活动对应的语义操作。根据前面解析的抽取模式，对 Web 页面内容进行抽取，抽取的数据以 XML 格式写入磁盘中。其中核心的抽取活动解析算法为：

(1)读取 XML 配置文件并利用 SAX 解析，将解析结果存储于 HashMap 数据结构中。

(2)遍历 SAX 解析生成的 HashMap 的元素，如果元素为字符串，则直接生成常量类型元素，将生成的元素加入到 Entities 列表中；否则根据实体名来匹配，以确定实体类型，生成对应的元素类的实体并加入到 Entities 中。

(3)遍历 Entities 列表中的所有元素，判断每个实体识别出封装的一系列操作是否有操作项，如果有，则设置其操作默认值等操作。将生成的对象加入到 Operation 列表中。

(4)遍历 Operation 列表中的所有元素，对每个操作元素根据操作名来生成对应的处理类。针对不同类的实体，处理如下：

1)如果类实体有操作，则为其生成对应的操作类实体并启动执行。

2)否则，生成对应的变量类实体。

(5)操作实体运行时，利用 OGNL(Object Graph Navigation Language)，将所处理的元素内的变量用注册值替换，生成新字符串。

(6)调用各操作相应的运行函数，来执行对应的语句。

4　抽取实例及结果

4.1　抽取实例

以淘宝网(http://www.taobao.com)为例，用户所需要抽取的页面为输入关键词后，按用户要求的排序方式返回的查询结果页面，其结果以多条记录的形式显示于多个页面上。用本文构建的系统对用户感兴趣的数据项进行抽取，用户只需编写符合流程抽取 Schema 的 XML 文档，然后解析器对 XML 文档进行解析，得到所有以分页形式显示的抽取结果，抽取结果的 XML 文件片断如下：

```
<Item><Name>图书管理系统-JAVA
</Name><Price>0.01</Price>
<Place>江苏南京<Place></Item>
<Item> <Name>*程序员*JAVA 之父
</Name><Price>1.00</Price>
<Place>辽宁大连<Place></Item>
```

4.2　实验结果

利用本文构建的系统，分别对不同类型网站页面的大量记录进行了抽取实验，内容涉及搜索引擎网站的搜索结果、高等院校网站的新闻、电子商务网站的交易商品等不同类型网站的不同类型数据记录，实验数据如表 1 所示。

表 1　网站记录的抽取实验结果

网站	内容	记录数	召回率	准确率
Google	查询结果	205	0.97	1.00
XiDian	校内新闻	520	1.00	1.00
Taobao	电子商务	350	0.98	1.00

由表 1 可以看出，本文提出的基于模板流程配置的抽取方法具有相当高的召回率和准确率。对各种不同类型的网站，流程抽取包装器均能够快速、稳定、正确地抽取用户所需的信息。

5　结束语

本文针对 HTML 结构解析方法的不足，提出并设计了基于模板流程配置的 Web 信息抽取系统，通过模板流程的配置来实现对 Web 信息的抽取。实验结果表明，本系统具有相当高的查全率和查准率，适用于高精度的网站页面数据信息的抽取。另外，采用流程配置的策略来构造抽取规则，可以处理“隐藏网”信息的抽取。下一步的工作是增强 Wrapper 构造的半自动化程度，提高配置 XML 文档的智能化。另外包装器的维护等也将是后续工作的重点。

参考文献

[1] Alberto H F. A Brief Survey of Web Data Extraction Tools[J]. SIGMOD Record, 2000, 31(2): 84-93.

[2] Sahuguet A, Azavan F. Building Intelligent Web Applications Using Lightweight Wrappers[J]. Data and Knowledge Engineering, 2001: 36(3), 283-316.

[3] Liu Ling, Calton P. XWRAP: An XML-enabled Wrapper Construction System for Web Information Sources[C]//Proc. of the 16th IEEE International Conference on Data Engineering,. San Diego, California, USA: [s. n.], 2000: 611-621.

[4] Baumgartner R, Flesca S, Gottlob G. Visual Web Information Extraction with Lixto[C]//Proc. of the 27th VLDB Conference. Roma, Italy: [s. n.], 2001.

[5] 高　军, 王腾蛟, 杨冬青, 等. 基于 Ontolgoy 内容二阶段半自动提取方法[J]. 计算机学报, 2004, 27(3): 310-319.

[6] Zhai Yanhong, Liu Bing. Extracting Web Data Using Instance-based Learning[J]. World Wide Web, 2005, 10(2): 113-132.