维普资讯 http://www.cqvip.com

基于扩展标记图的网页信息重组技术*)

朱征宇　王　亮　赵银春　程代杰
(重庆大学计算机学院　重庆400044)

摘　要　本文介绍一种基于扩展标记图[1]ETG(Extended Tag Graph)的网页信息抽取与重组新技术，引入了扩展标记图操作和重构概念，提出了作为用户接口的标记查询语言 TagSQL。用户通过类标准 SQL 的语言描述，即可方便地实现对网页信息的灵活抽取和重组操作。

关键词　HTML，扩展标记图 ETG，标记路径，查询语言，TagSQL

A Technique for Information Reconstruction of Web Pages Based on ETG

ZHU Zheng-Yu　WANG Liang　ZHAO Yin-Chuen　CHENG Dai-Jie
(Computer College of Chongqing University, Chongqing 400044)

Abstract　Based on Extended Tag Graph (ETG)[1], a new technique for information extraction and reconstruction of Web pages has been presented in the paper. We have introduced the concepts of ETG Operations and ETG Reconstruction, and put forward a Tag Structured Query Language (TagSQL) in the design of user interface. By using the language given in the similar form as SQL, a user can describe conveniently the operations for the information extraction and reconstruction of Web pages.

Keywords　HTML, Extended tag graph(ETG), Tag path, Query language, TagSQL

1　概述

在数以亿计的 Web 网页中包含着大量有用信息，探索更加灵活的网页信息查询新途径具有重要的现实意义和应用价值。

目前 Web 上主要的网页信息查询技术是搜索引擎和个性化推荐系统，它们都局限于为用户提供网页级的 URL。虽然已出现多种基于网页结构的查询语言[2~7]，但目前还处在探索和实验阶段。文[4]给出了一种基于规则的 HTML 查询语言 WebQL，能够提供灵活的基于网页内部层次结构的查询，但这种采用具有逻辑语义的高级说明性查询语言不利于理解和使用。文[5]介绍了一种基于标记图的 Web 数据模型，但仅给出了查询语言(TGQL)及功能的初步构思。XPath[6]，尤其是 XQuery[7]，是一种针对 XML 资源的查询语言，虽然也可用于处理网页，但因未考虑到 HTML 的特殊性(标记固定，作用明确，数目较少)，查询优化与实现的技术复杂，执行效率降低。

本文将介绍一种基于扩展标记图[1]ETG(Extended Tag Graph)的网页信息抽取与重组新技术。首先对扩展标记图的操作和重构问题进行讨论，然后作为用户接口将给出一种标记查询语言 TagSQL(Tag Structured Query Language)，使用户通过类标准 SQL 的语言描述即可方便地实现对网页信息的灵活抽取和重组操作。此外，对系统框架及实现进行了简述。

2　ETG 操作与重构

2.1　ETG 简介[1]

文[1]通过引入 ETG 概念，讨论了网页的虚拟表示模型及其在网页模块化设计和网页一致性维护中的作用。ETG 采用虚拟方式描述网页或 HTML 标记段，具有如下特点：

1)ETG 通常是一棵边有序树 $G_r(V,E^d)$，简记 G(r,V,E)，节点对应于 HTML 文件中的 HTML 元素，节点间父子、兄弟关系分别描述 HTML 元素间嵌套、顺序关系。各节点均采用五元组(标识，名称，类型，属性，值地址)表示；

2)ETG 中，可以不包含网页的内容数据，而仅有内容指针；

3)若限定不使用扩展标记符，则 ETG 与网页或 HTML 标记段存在一对一关系。

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" ...>
<html xmlns="http://www.w3.org/1999/xhtml" ...>
 <head profile="http://www.w3.org/2000/08/w3c-synd/#">
  <meta http-equiv="Content-Type" .../>
  <meta name="generator" .../>
  ......
  <style type="text/css">
   <!--html, body { background: #fff; color: #000; } --> </style>
 </head>
<body>
 <h1 id="logo">
  <img ... height="48" width="315" src="Icons/w3c_main" /> </h1>
 ......
</body>
</html>
```

图1　W3C 主页(http://www.w3c.org/)

*)该项研究得到重庆市科技公关项目(2001.6715)、重庆大学骨干教师资助计划项目(2003A33)的资助。朱征宇　博士研究生，研究方向为电子商务，Web 智能检索；王　亮，赵银春　硕士研究生，研究方向为 Web 智能检索；程代杰　教授，博士生导师，研究方向为并行处理，计算机网络，电子商务。

维普资讯 http://www.cqvip.com

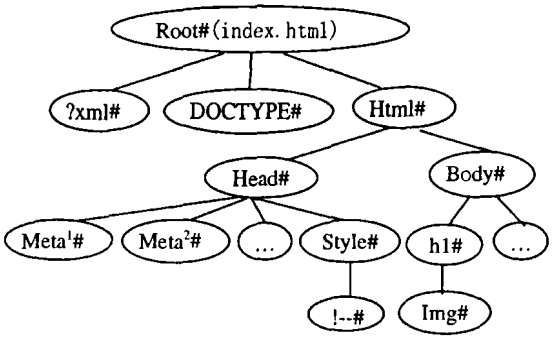

图2 W3C 主页的 ETG

例1 对图1中 W3C 主页，其 ETG 如图2所示。X#代表节点标识，根 root 代表主页文件 index.html，HTML 元素的属性和表示内容数据的内容指针都包含在相应节点的五元组内，内容数据则存放在素材库中。

ETG 不仅描述了网页的标记结构，从模块设计角度看，节点(子树)代表信息模块，ETG 则描述了网页如何由多个信息模块按一定的"逻辑结构"合成。下面，将对网页重组技术进行讨论，首先引入标记路径、约束条件、ETG 操作和重构等概念。

2.2 标记路径和约束条件

定义1(标记对象) ETG 中每一节点 v 都确定一棵以它为根的子树，它描述了网页中相应的 HTML 元素(由首尾标记界定的信息模块)，称这样的子树为标记对象。

显然，标记对象也是一个 ETG。又因 HTML 元素间存在嵌套关系，也可有嵌套标记对象和父、子标记对象之称。

定义2(标记路径) 设 $G=(r,V,E)$ 为 ETG，v_1,v_2 为两节点，v_1 位于从根 r 到 v_2 的路径上，则称从 v_1 到 v_2 的路径所经过的节点的类型序列 $P=type_1|type_2|\cdots|type_k$ 为 G 的一相对路径，并称 v_1 和 v_2 满足 P，记 $P(v_1,v_2)$。当 $v_1=r$ 时称 P 为绝对路径，两者统称标记路径。

注意，在同一 ETG 中，对给定标记路径 P 可能匹配多条实际路径，即满足 $P(v_1,v_2)$ 的 v_2 可能为一集合，记为 P{End}。显然，P 给定后 P{End}是唯一确定的。

例如对图2中 ETG，P_1=index.html|HTML|head|meta 为绝对路径，P_2=head|meta 为相对路径，$P_1\{End\}=P_2\{End\}=\{meta^1\#,meta^2\#,\cdots\}$。

此外，为能从 P{End}中筛选节点，我们引入了约束条件概念。

定义3(约束路径) 设一 ETG 的标记路径为 $P=type_1|type_2|\cdots|type_k$，如果 $f=(f_1,f_2,\cdots,f_k)$，其中 f_i(可为空)是对类型为 $type_i$ 的节点的属性、内容或子孙特征的约束条件($i=1,2,\cdots,k$)，则称 P 为带约束条件 f 的标记路径，简称约束路径，记为 P(f)。

若将满足 $P(f)(v_1,v_2)$ 的节点 v_2 的集合记为 P(f){End}，则显然它是 P{End}的子集。

例如对上 P_2，令 f=(,name="generator")，则 $P_2(f)$ 表示这样的特殊路径集：$\{v_1\rightarrow v_2|$ 节点 v_1 和 v_2 的类型为 head 和 meta，且 v_2 的 name 属性值为"generator"$\}$，显然 $P_2(f)\{End\}=\{meta^2\#\}$。

2.3 ETG 操作

为形式化描述基于 ETG 的网页信息模块抽取与重组技术，引入了7种基本 ETG 操作。ETG 操作的引入借鉴了对象代数[9]思想，但因操作对象和作用不同，其定义也有很大不同。

(1)Π-投影操作 该操作按一组标记路径从 ETG 中抽取标记对象(子树)集，形成新的 ETG。

定义4 设 $G_1=G(r_1,V_1,E_1)$ 为 ETG，$P=\{P_1,P_2,\cdots,P_k\}$ 为 k 个标记路径，将获得 $G_2=G(r_2,V_2,E_2)$ 的过程称为 G_1 关于 P 的投影操作，记为 $\Pi_P(G_1)$。G_2 构成方式分三种情况：

i)当 k=1且 $P_1\{End\}$仅含一个节点 v 时，则 G_2 就是标记对象 v 所表示的子树。

ii)当 k=1但 $P_1\{End\}$含多个节点 $v_1,v_2,\cdots,v_n$ 时，构成 G_2 如下：

根 r_2 下有 n 个儿子节点，依次为 $v_1,v_2,\cdots,v_n$，而每个 v_i 节点又扩展为一棵由标记对象 v_i 所表示的子树($i=1,2,\cdots,n$)。这里，r_2 的类型应使得 G_2 为一 ETG。

iii)其它情况下，将标记对象集按 k * n 表格构成 G_2 如下：

根 r_2 为 table 类型节点，r_2 下有 n 个 tr 类型的儿子节点；而每个 tr 类型节点有 k 个 td 类型的儿子节点，它们分别是满足 $P_1,P_2,\cdots,P_k$ 的标记对象(子树)，可为空。对任何 i ($1\leqslant i\leqslant k$)，若满足 P_i 的所有标记对象为 $v(i,1),v(i,2),\cdots,v(i,n_i)$，则它们依次出现在 r_2 下前 n_i 个 tr 节点中。这里，取 $n=Max(n_i)$。

(2)σ-选择操作 该操作按约束条件从 ETG 中抽取标记对象(子树)集，形成新的 ETG。

定义5 将定义4中标记路径集换为约束路径集 $P(f)=\{P_1(f_1),P_2(f_2),\cdots,P_k(f_k)\}$ 而获得 G_2 的过程，称为 G_1 关于 P(f)的选择操作，记 $\sigma_{P(f)}(G_1)$，它总为 $\Pi_P(G_1)$ 的子图。

(3)∝-连接操作 该操作按连接条件从两 ETG 中抽取标记对象(子树)集，并按 m * n 表格形成新的 ETG。

定义6 设 $G_1=G(r_1,V_1,E_1)$ 和 $G_2=G(r_2,V_2,E_2)$ 均为 ETG，分别有一组约束路径 $F_1=\{P_{11}(f_{11}),P_{12}(f_{12}),\cdots,P_{1k}(f_{1k})\}$ 和 $F_2=\{P_{21}(f_{21}),P_{22}(f_{22}),\cdots,P_{2t}(f_{2t})\}$，ω 为连接条件，则获得 $G=G(r,V,E)$ 的过程称为 G_1 和 G_2 的连接操作，记为 $\propto_\omega(G_1,G_2)$。按 m * n 表格构成 G 如下：

根 r 为 table 类型节点，r 下有 n 个 tr 类型的儿子节点；而每个 tr 类型节点下均有 m=k+t 个 td 类型的儿子节点，它们分别是 $P_{11}(f_{11}),P_{12}(f_{12}),\cdots,P_{1k}(f_{1k}),P_{21}(f_{21}),P_{22}(f_{22}),\cdots,P_{2t}(f_{2t})$ 的满足 ω 的某个标记对象(G_1 或 G_2 的子树)，而 n 为满足 ω 的不同组合情况数。

(4)Σ-合并操作 该操作合并两 ETG 为一新的 ETG。

定义7 设 $G_1=G(r_1,V_1,E_1)$ 和 $G_2=G(r_2,V_2,E_2)$ 均为 ETG，则称获得 $G=(r,V,E)$ 的过程为合并操作，记为 $\Sigma(G_1,G_2)$。G 构成方式如下：

$$V=\{r\}\cup V_1\cup V_2,E=\{\langle r,r_1\rangle,\langle r,r_2\rangle\}\cup E_1\cup E_2$$

其中，符号$\langle a,b\rangle$表示节点 a 到 b 的有向边，而 r 的类型应使得 G 为一 ETG。

(5)并(∪)、交(∩)、差(－)操作 并操作获得两 ETG 中所有儿子节点(子树)，但去除重复。交操作从两 ETG 中获得公共儿子节点(子树)。差操作(根据另一 ETG)从一 ETG 中剪除部分儿子节点(子树)。

定义8 设 $G_1=G(r_1,V_1,E_1)$ 和 $G_2=G(r_2,V_2,E_2)$ 均为 ETG，且 r_1 和 r_2 类型相同，θ 为比较条件，则称获得图 $G_3=(r_1,V_3,E_3)$、$G_4=(r_1,V_4,E_4)$ 和 $G_5=(r_1,V_5,E_5)$ 的过程分别为并、交和差操作，并分别记为 $G_1\cup G_2$、$G_1\cap G_2$ 和 G_1-G_2。

维普资讯 http://www.cqvip.com

G_3、G_4和G_5构成方式如下：设r_1的儿子节点集为$a_1,a_2,\cdots,a_n$，且对应的标记对象子树分别为$G_{11}=G(a_{11},V_{11},E_{11})$，$G_{12}=G(a_{12},V_{12},E_{12})$，$\cdots$，$G_{1m}=G(a_{1m},V_{1m},E_{1m})$，而$r_2$的儿子节点集为$b_1,b_2,\cdots,b_m$，则

i) $V_3=V_1\cup V_2-\{r_2\}-\{b_j$ 子树的节点集$|b_j\theta(a_1,a_2,\cdots,a_n)$and $(1\leqslant j\leqslant m)\}$

$E_3=E_1\cup E_2-\{(r_2,b_j),b_j$ 子树的边集$|b_j\theta(a_1,a_2,\cdots,a_n)and(1\leqslant j\leqslant m)\}$

ii) $V_4=\{r_1\}\cup\{v\in V_{1i}|a_i\theta(b_1,b_2,\cdots,b_m)$and $(1\leqslant i\leqslant m)\}$

$E_4=\{(r_1,a_i),e\in E_{1i}|a_i\theta(b_1,b_2,\cdots,b_m)$and $(1\leqslant i\leqslant m)\}$

iii) $V_5=\{r_1\}\cup\{v\in V_{1i}|(\text{not } a_i\theta(b_1,b_2,\cdots,b_m))$and $(1\leqslant i\leqslant m)\}$

$E_5=\{(r_1,a_i),e\in E_{1i}|(\text{not } a_i\theta(b_1,b_2,\cdots,b_m))$ and $(1\leqslant i\leqslant m)\}$

其中，$a_i\theta(b_1,b_2,\cdots,b_m)$表示$a_i$与$b_1,b_2,\cdots,b_m$中之一符合$\theta$，前加not表示否定。

注意，这里θ没有限定a_i与$b_1,b_2,\cdots,b_m$之间必须是等式比较，而放松了比较范围。θ允许在a_i子树与b_j子树之间，按指定的属性、内容或子孙标记对象特征进行比较。

2.4 ETG 重构

ETG 操作用于描述基于 ETG 的信息模块抽取行为，接下来将讨论对抽取模块的重组问题，引入了广义 ETG 方法。通过广义 ETG 的设计，能够将多个 ETG 操作结果按照用户需要的形式组织成新的 ETG，并以新网页的形式显示。

定义9 称满足下两条件的有向图 G 为广义 ETG：

i)G 中允许出现代表 ETG 操作的特殊的 ETG 节点；

ii)将 G 中所有 ETG 节点替换为相应操作的结果子树后，得到的新图为一 ETG。

广义 ETG 可以看作是一种针对 ETG 操作子树的构造器，用于描述对多个 ETG 操作的逻辑组织结构。我们可以将每个 ETG 操作(抽取的信息模块)看作一个 ETG 节点，并根据用户需要的网页显示格式来设计相应的广义 ETG。

3 用户接口设计与 TagSQL 语言

根据上述讨论，原则上讲，用户通过 ETG 操作和重构可以实现对网页信息模块的灵活抽取和重组网页。但让用户直接面对 ETG 操作和重构是不现实的，需要提供更为友好的用户接口。为此，我们设计一种便于用户使用的 TagSQL 语言。

TagSQL 的设计参照了 Xpath 语言[6]和基于 Xquery 数据模型[8]的 Xquery 查询语言[7]。虽然 HTML 可看作是 XML 的特例，但 Xquery 毕竟不是针对 HTML 设计的，不能有效利用 HTML 的特点，执行效率低。根据我们的应用目的，基于对 Xpath 和 Xquery 的融合、简化和调整形成了 TagSQL，其描述形式则参照了人们熟悉的标准 SQL 语言。

3.1 约束路径表达式

由于许多 ETG 操作都涉及到约束路径 P(f)，在给出 TagSQL 的形式文法之前，先给出约束路径的形式文法。

定义10 下面是约束路径的形式文法，能够描述对标记路径中任意标记对象的属性约束、子路径约束和文本内容约束。

```
CTPath ::=(URL"|"RelativePath)| RelativePath
RelativePath ::= Paths["|"Paths]…
Paths ::= *|?|?(n)|Tag["("ConditionExpr")"]
ConditionExpr ::= [not] BaseConditionExpr [(and | or) [not] BaseConditionExpr]…
BaseConditionExpr ::= SubPath (like |=)RelativePath | Subid= integer | AttributeExpr
AttributeExpr::= (AttributeName | Tagtxt)(< | <= | = | > | >= | <> | like)Constant
```

其中，Tag(ConditionExpr)表示标记节点及相应约束条件，Constant 表示字符串常量，Subid=integer 用于指明该节点在父节点中的位置序号(integer 为正整数)，AttributeExpr 为基本的属性比较运算，and、or 和 not 表示逻辑运算，SubPath 表示对标记对象内的子孙节点特征进行约束(采用约束路径方式)，Tagtxt 为代表标记对象内文本内容的特殊属性，通配符 *、?和?(n)分别表示任意级、下一级和第 n 级子节点。此外，可引入替换符@将冗长的约束条件单独表示，"|"、")"和"("等区分于文法分隔符|、)和(。

3.2 TagSQL 语言的形式文法

(1)信息模块抽取文法 针对网页信息模块抽取的3种基本 ETG 操作：Π、σ 和∝，我们为用户设计了易于描述和功能较强的 Select 语句，并扩展了分组计算和排序等常用功能。

定义11 Select 语句用于网页信息模块的抽取。其形式文法如下：

```
Select [distinct] CTPathExpr [$alias] [,CTPathExpr [$alias]]…
       [format col | row]
From CTPath [$alias]"|" [,CTPath [$alias]"|"]…
[Where ConditionExpr]
[Order By CTPathAttr | [,CTPathAttr]… ]
[Group By CTPathAttr]
[Having HavingCondition];
```

其中出现的表达式应满足下述子文法：

```
CTPathExpr ::= CTPathOrAttr | function"("CTPathOrAttr")"
CTPathOrAttr ::= CTPath | CTPathAttr
CTPathAttr ::= CTPath. Attribute
HavingCondition ::= (CTPathExpr(<|<=|=|>|>=|<>) Constant)
Function ::= Count | Sum | Max | Min | Avg
```

这里，CTPath 和 ConditionExpr 如定义10。Function 为五种常用聚集函数；Attribute 为属性名，"."表示取标记对象的属性值，特别将 Tagtxt 也作为特殊的属性看待，表示取标记对象的文本内容；$alias 定义别名变量 alias，可在该语句中的任何(包括条件表达式中的)标记路径的首部使用。

Select 语句中各子句的作用与标准 SQL 中 Select 语句情形十分类似。Select 子句为查询结果构建器，由多个约束路径构成，[format col | row]选项用于对结果输出格式的简单控制(参3.2节(5))；From 子句指定标记对象抽取范围，为以"|"结尾的多个约束路径，可来自多个网页；Where 子句给出对 Select 子句中标记对象的约束，如对结果集的筛选条件或多个标记对象间连接条件；Order By 子句对结果元组排序；Group By 子句用于聚集函数的分组计算；Having 子句用于对结果元组作筛选。后四个子句均为可选项。

大多数应用情形，From 子句中仅包含一个约束路径，这时 Select 语句中所有约束路径都针对该约束路径指定范围进行对象抽取，可以不使用别名；但当 From 子句中包含多个约束路径时，每个路径后都应定义别名，并需在 Select 语句中各约束路径首部指明引用何别名。

(2)集合运算文法 ETG 操作中除投影、选择和连接三种基本操作外，还涉及到对基本操作结果进行并构的∪、∩、－和Σ四种集合操作。为此，我们为用户设计了相应的集合运算文法。

定义12 两个 Select 语句(结果元组同构)可以作并(Union)、交(Intersect)、差(Except)集合运算，其形式文法如下：

维普资讯 http://www.cqvip.com

〈Select1〉〈Union | Intersect | Except〉[Constraint]〈Select2〉;
其中:Constraint ::=〈CTPathOrAttr1〉with〈CTPathOrAtt2〉
〈CTPathOrAtt1〉::= CTPathOrAttr
〈CTPathOrAtt2〉::= CTPathOrAttr

Constraint 用于指定 θ 比较条件,〈CTPathOrAtt1〉和〈CTPathOrAtt2〉分别指明 Select 语句〈Select1〉和〈Select2〉中用于比较的对象,而 CTPathOrAttr 如定义11。

应当指出,与标准 SQL 不同,为使集合操作还能在结构不尽相同的网页上进行,集合运算允许通过比较条件指明比较对象。而且比较方式不是直接针对两标记对象进行,而推广到对它们内部的指定子孙对象特征的比较。注:Union 后带 Constraint 时为∪,否则为Σ。

(3)*嵌套文法* 为能对抽取结果中元组进行筛选,TagSQL 扩展了 Select 语句功能,允许在 Where 子句中嵌套 Select 语句。

定义13 将原 Where 子句中的 AttributeExpr 形式文法扩展如下:

AttributeExpr ::=(AttributeName(<|<=|=|>|>=|<>|like
| in)〈Select〉)
| Tagtxt like <Select> |exists〈Select〉

(4)*结果构造器文法* Select 语句的查询结果一般为由 m 列×n 行组成的标记对象集合,m 为 Select 子句中用','分隔的标记路径数,若满足标记路径 P_i 的标记对象数为 n_i($1 \leqslant i \leqslant m$),则 $n=\max(n_i)$。为对这些对象在新网页中的显示格式进行简单控制,Select 子句后允许使用[format col|row]选项。缺省时,所有对象按查找出的次序线性排列;指定 col 时,采用1列、m×n 行的表格控制线性排列结果的显示;指定 row 时,采用 m 列、n 行的表格控制对象集合的显示。

更一般地,针对广义 ETG 构造器,我们为用户设计了简单实用的结果构造器文法,可用于根据多个 Select 语句结果构成新网页时对复杂显示格式的说明。

定义14 嵌入 Selecet 语句的 HTML 文件被称为广义 HTML 文件(EHTMLFile)。一个 EHTMLFile 描述了一个广义 ETG 构造器,广义 HTML 语言即是为用户提供的结果构建器文法。

总之,Select 语句和 EHTMLFile 的查询结果均是一个描述新网页的 HTML 文件。

(5)*视图文法* 与标准 SQL 类似地,也可采用 Select 语句或 EHTMLFile 来定义网页视图。系统仅保存视图定义,而不保存其执行结果。视图被看作是虚拟 ETG,可在视图上使用 Select 语句。

定义15 设〈Select_Ehtm〉为 Select 语句或 EHTMLFile,视图定义的形式文法如下:

Create View viewname As〈Select_Ehtm〉

最后指出,无论是 Select 语句还是 EHTMLFile,其描述的有效性(运行结果为一有效的 HTML 文件)需要用户进行精心设计和调试。因语法的复杂性和涉及到 HTML 语言,TagSQL 语言不一定适合最终用户使用,查询语句、EHTMLFile 和网页视图的设计可由网站管理者等专业人员负责,然后将调试完成的设计结果提供给最终用户使用。

虽然任意的 Select 语句并不具有特别的应用价值,但 TagSQL 语言确实为我们提供了一种网页信息模块的灵活抽取与重组能力,通过专业人员的精心设计和调试,而不需对网页做任何修改,就能够为最终用户提供满足不同需求的新网页视图,这是目前各种搜索引擎或网页信息检索系统都难以办到的。

3.3 TagSQL 语言应用举例

例2 从 IEEE 分会网页抽取"会员人数多于100 的专业分会"信息的语句为:

```
Select html|body|table|tr@1 $z
From  http://www.cie-china.org/ieee-beijing/newsletter/ieee2002-2.htm|
@1=(Subpath Like td(Subid=2 and Tagtxt>100))
Order By z.td(Subid=2).Tagtxt;
```

其中,subid=2用于限定子节点在父节点中的位置(序号),而替换符@将冗长约束条件单独表示使语句可读性好,系统会自动实现替换。网页视图如图3,结果按会员数排序显示。

查询结果

Society	会员数目	是否建立Chapter
Electron Devices	101	已建立
Microwave Theory and Techniques	103	已建立
Signal Processing	170	已建立
Communications	597	已建立
Circuits & System	634	已建立
Computer	641	已建立

图3

查询结果

许乐平	电气01级1班	65119070	1-407
周 军	计算机02级自动化	67917017	1-117
李代华	计算机通信工程	65119037	1-210
田 阳	计算机通信工程	65119035	1-206
周 浩	计算机通信工程0201班	65119645	5-614
曾正华	计算机通信工程0202班	65119035	1-206
苏旺杰	计算机维修0102班	65119065	1-320
陈旭旺	计算机系通信工程	65119550	5- 622
崔称云	计算机系网络02级1班		3-412

图4

查询结果

不同地区摩托罗拉T720手机报价表

〔北京〕北京国美手机报价 - 手机报价系统 - 手机之家					
摩托罗拉 T720	¥2166	比	平	04-17	详细
〔西安〕西安国美手机报价 - 手机报价系统 - 手机之家					
摩托罗拉 T720	¥2378	比	降	(新装)	06-04
〔上海〕上海国美手机报价 - 手机报价系统 - 手机之家					
摩托罗拉 T720	¥2280	比	-	06-04	详细
〔重庆〕重庆国美手机报价 - 手机报价系统 - 手机之家					
摩托罗拉 T720	¥2480	比	升	06-04	详细

统计源: http://www.imobile.com.cn/

图5

例3 要从两结构相同的学生会网页和科协网页中查找同是两会成员的学生信息,可采用集合运算语句(网页视图从略):

```
Select x|?|td(Subid=1).Tagtxt,?|td(Subid=2).Tagtxt
From @1|html|body| * table(width="75%") $x
Intersect x|?|td(Subid=1).Tagtxt With y|?|td(Subid=1).Tagtxt
Select y|?|td(Subid=1).Tagtxt, ?|td(Subid=2).Tagtxt
From @2|html|body| * table(width="75%") $y
@1 = http://www.hebiat.edu.cn/jxx/xsh/xueshenhui/ganbumingdan.htm
@2=http://www.hebiat.edu.cn/jxx/xsh/kx/mingdan.htm;
```

例4 将计算机协会会员按专业分组显示的语句为(网页视图如图4):

```
Select $x | * | table | tr @1) From http://202.202.2.59/jsjxh/xhhy.asp|html| $x
Group By y.Tagtxt
@1=(Subpath=td $y(Subid=2));
```

例5 查找 http://www.529buy.com 网站上 T720手机报价中比 http://www.szmobile.com 网站上相同手机报价低的手机信息,可采用嵌套的 Select 语句(网页视图从略):

```
Select tr $t1
From @1|html|body|?|table(Tagtxt Like "T720i")
Where t1|td(subid=2)|s.Tagtxt <
```

维普资讯 http://www.cqvip.com

```
    (Select Min(tr(Tagtxt Like "T720i")|td(subid=4).Tagtxt)
     From @2|html|body|?|table(width="100%"))
@1= http://www.529buy.com/class.asp?aid=20&nid=213
@2= http://www.szmobile.com/market/price.ASP;
```

例6　可通过 Select 语句从多个网页中抽取 motoT720手机信息，并按照下述广义 HTML 文件 motoT720.Ehtm 构造为一新网页(网页视图如图5)：

```
<html>
<head><title>不同地区摩托罗拉 T720手机报价表</title></head>
<body>
<p>不同地区摩托罗拉 T720手机报价表</p>
<TagSQL>
Select $P1|head|title,$P1|*|table|tr@5, $P2|head|title,$P1
       |*|table|tr@5,
       $P3|head|title,$P1|*|table|tr@5, $P4|head|title,$P1
       |*|table|tr@5
From @1|html $P1, @2|html $P2, @3|html $P3, @4|html
       $P4
@1=http://www.imobile.com.cn/price.php?jid=27
@2=http://www.imobile.com.cn/price.php?jid=38
@3=http://www.imobile.com.cn/price.php?jid=28
@4=http://www.imobile.com.cn/price.php?jid=30
@5=(Subpath=*|a(Tagtxt="T720"));
</TagSQL>
<p>统计源:http://www.imobile.com.cn/</p>
</body>
</html>
```

4　系统框架与实现

针对上述讨论，我们在虚拟网页技术[1]基础上开发了一种能够支持 TagSQL 语言的实验原型系统 PowerSearcher，系统框架如图6，实现了基本的网页信息抽取与重组功能。该系统虽然主要针对虚拟网站的情形开发，但也适用于处理 Web 上任何 HTML 网页，只是增加了根据 HTML 文件自动抽取相应 ETG 的处理环节。

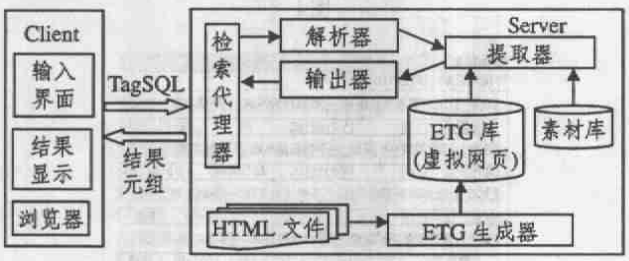

图6　PowerSearcher 系统框架

PowerSearcher 的设计和实现充分借助了关系数据库技术的处理效率，也降低了处理难度。ETG 树采用关系数据模式存储，关系以节点(标记对象)为记录，并将常用的 HTML 属性都作为该关系模式的固定属性，同时还增加了 Subid、标记对象在虚拟网页中的起止地址等属性。此外，为能便于判定嵌套关系还增加了必要的路径索引信息。Select 语句的处理则是通过编译程序转化为标准 SQL 语句来执行，再对 SubPath 或 Tagtxt 约束条件补充处理。

4.1　客户端

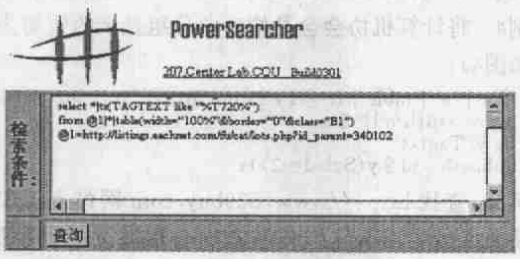

图7　TagSQL 输入界面

主要由浏览器、输入界面(图7)和结果显示模块组成。输入界面提供 TagSQL 语言输入环境，结果显示模块可提供多级查询结果并以 HTML 文件在浏览器中显示。

结果显示模块支持四种显示方式：1)仅显示结果元组涉及的相关网页的 URL 列表(图8)；2)仅显示结果元组涉及的相关标记路径列表；3)显示最终查询结果(图3、4、5)；4)在 URL 列表(图8)中点击 URL 链接可显示相应网页。

图8　相关网页的 URL 列表

4.2　服务器端

由检索代理器、解析器、ETG 生成器、提取器、输出器等模块组成。

检索代理器(Searcher)：接检索请求，把 TagSQL 语句传递给解析器，将查询结果返回给客户端。

解析器(Translator)：将 Select 语句从 EHTMLF 中分离，并对 Selcct 语句进行语法解析，并提交给提取器。

提取器(Extractor)：执行各 Selcct 语句，提取相应信息模块，并送输出器。

输出器(Exporter)：根据 EHTMLF 构造查询结果，生成相应 HTML 文件。

ETG 生成器(Collector)：仅供处理实际 HTML 网页时使用，用于自动提取相应 ETG 结构，存于 ETG 库，供提取器使用。由于许多网页都存在语法错误，在抽取 ETG 前应对 HTML 文档进行清理[10]。

结束语　针对当前各种 Web 搜索引擎和信息推荐系统都是以网页级 URL 作为输出、对网页查询语言的研究还处于实验探索阶段的情况，本文对网页信息重组技术进行了研究。我们在前期扩展标记图[1]ETG 研究基础上，借鉴对象代数[9]思想引入了 ETG 操作和重构概念，并给出了一种基于 ETG 的多网页信息模块抽取和网页重组技术。进而在用户接口设计中，参照 Xpath[6]和 XQuery[7]语言，提出了一种 ETG 操作和重构语言 TagSQL，用户通过类标准 SQL 的语言描述，即可实现对网页信息模块的灵活抽取和重构操作。TagSQL 语言的基本功能，在我们设计开发的基于虚拟网页技术的实验原型系统 PowerSearcher 中得到初步实现和验证。

参 考 文 献

1　朱征宇，朱庆生，王茜. 基于扩展标记图的虚拟网页技术. 计算机科学，2001，28(11)：80～82

2　Abiteboul S，Quass D，McHugh J，Widom J，Wiener J L. The Lorel query language for semistructured data. Int J Digit Libr，1997，1：68～88

3　Spertus E，Stein L A. Squeal：a structured query language for the Web，Computer Networks，Volume 33，Issues 1-6，June 2000. 95～103

4　Liu M，Ling T W. A Rule-based Query Language for HTML. In：Proc. of the Seventh Intl. Conf. on Database Systems for Advanced Applications (DASFAA™01). IEEE，2001

(下转第64页)

维普资讯 http://www.cqvip.com

效率的提高产生很大帮助。

1. 并行性：由于目前对检索速度的要求越来越高，单纯从模型的检索算法上来提高速度已经不能满足需要，这就迫切地要求索引模型提供并行处理机制。有两种并行，一种是检索请求间的并行，对每个检索请求串行处理，但是同时处理多个请求；另一种是请求内部并行。处理请求的过程一般包括搜索索引库和处理合并搜索结果。如果是署名文件还需要一个查询文档消除误匹配的过程，这三部分之间可以并行处理。索引模型的并行机制应该充分利用到这两种并行方式。

2. 相关度排序(Ranking)：很多时候，无需将查询结果全部返回给用户，而是对查询结果进行相关度排序，仅把相关度最高的 N 条记录返回。若能够在不检索全部文档的情况下就把相关度最高，或者近似最高的 N 条记录返回，则可使查询效率大大提高。实际上，我们可以预先设定关键词的权重，首先处理那些具有较大权重的关键词。

3. 对不同长度文档的处理：有些全文检索模型的性能会受到文档长度变化的影响，为了获得比较好的性能，我们可以有两种方法来解决这个问题：一、把需要索引的文档分类，相同长度的放在一起处理作为一个子索引库，查询的时候，分别查询所有子索引库，而后合并查询结果。二、将长文档切分为多个部分，作为多个短文档来索引。

结束语 全文检索技术的出现，导致了信息检索领域的一场革命。它不仅可以实现情报检索的绝大部分功能，而且还能直接根据数据资料的内容进行检索，实现了多角度、多侧面地综合利用信息资源。但是目前的全文检索模型并不完善，最广泛使用的倒排表模型膨胀比小，创建和检索速度都比较快，但要受到索引数据格式的限制，查询功能不完备。而 Pat 树和 Pat 数组能够解决倒排表的问题，但是膨胀比又很大，动态性能也很差。ISTree 兼顾了倒排表和 Pat 树的优点，但是对 ISTree 的研究还刚刚起步。所以，对 ISTree 模型的研究与完善应该成为我们今后研究工作的重点。

参 考 文 献

1 Zeng Haiquan, Shen Zhan, Hu Yunfa. Mining Sequence Pattern from Time Series Based on Inter-Relevant Successive Trees Model. In:Proc. of 9th. Intl. Conf. on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC'2003), LNCS/LNAI, Spring-Verlag, Chongqing, China, 2003

2 曾海泉，刘永丹，宋扬，胡运发. 基于互关联后继树的多时间序列关联模式挖掘. 计算机研究与发展，(已录用)2003,1

3 Knuth D E. The Art of Computer Programming, Sorting and Searching. 1st edition. Addision-Wesley Pub. Co., 1973

4 Weiner P. Linear pattern matching algorithm. In: Proc. 14th IEEE Symposium on Switching and Automata Theory, 1973. 1～11

5 Manber U, Myers E. Suffix arrays: A new method for on-line string searches. In: Proc. of the FISTREE Ann. ACM-SIAM Symp. on Discrete Algorithms, 1990. 319～327

6 Hu Yunfa, Zhou Shuigeng. A New Model of Chinese Full-text databases. In: Proc. World Multiconference on Systemics, Cybernetics and Informatics, Florida, USA, 2001. 528～533

7 Tao Xiaopeng, Hu Yunfa, Zhou Shuigeng. Subsequent Array: A New Full Text Index. In: Proc. World Multiconference on Systemics, Cybernetics and Informatics, Florida, USA, 2001. 551～556

8 胡运发. 互关联后继树—— 一种新型全文数据库数学模型：[技术报告]. 复旦大学计算机与信息技术系，2002. 3

9 曾海泉，宋扬，申展，胡运发. 基于互关联后继树的时间序列相似性查询. 计算机研究与发展，(已录用)2002. 12

10 杜莉. 基于互关联后继树的关联规则挖掘研究：[复旦大学硕士学位论文]. 2003. 5

11 Zobel J, Moffat A, Ramamohanarao K. Inverted files versus signature files for text indexing. Transactions on Database Systems, 1998,23(4):453～490

12 Grossi R, Vitter J S. Compressed suffix arrays and suffix trees with applications to text indexing and string matching (extended abstract). STOC 2000. 397～406, 1999

13 申展，王建会，吴爱华，胡运发. 互关联后继树模型——一种新颖的全文检索模型. 见：全国数据库学术会议，湖南长沙，2003

14 Moffat A, Zobel J. Self-Indexing Inverted Files for Fast Text Retrieval. ACM Transactions on Information System, 1996, 14(4):349～379

15 Gonnet G H, Baeza-Yates R. Lexicographical indices for text: Inverted files vs pat trees: [Technical Report TR-OED-91-01]. University of Waterloo, 1991

16 陶晓鹏. 面向(中文)全文数据库的全文索引的研究：[复旦大学博士学位论文]. 1999

17 Zobel J, Moffat A, Davis S. An Efficient Indexing Technique for Full-Text Database Systems. In: Proc. of the 18th Inter. Conf. on VLDB, 1992. 352～362

18 Golomb. Run-Length Encodings. IEEE Transactions on Information Theory, 1996, IT-12(3): 399～401

19 Faloutsos C. Access methods for text. Computing Survey, 1985, 17(1)

20 Roberts C. Partial-match retrieval via the method of superimposed code. Proceeding of the IEEE, 1979, 67(12):1624～1642

21 Christodoulakis S, Faloutsos C. Design Considerations for a Message File Server. TSE, 1984, 10(2):201～210

22 Farach M, Ferragina P, Muthukrishnan S. Overcoming the Memory Bottleneck in Suffix Tree Construction. In: Proc. of IEEE FOCS, 1998. 174～183

23 McCreight E. A space-economical suffix tree construction algorithm. Journal of the ACM, 1976, 23(2):262～272

24 Muthukrishnan S. Efficient Algorithms for Document Retrieval Problems. In: Proc. ACM-SIAM SODA, 2002. 657～666

25 Manber U, Myers G. Suffix arrays: a new method for on-line string searches. SIAM journal on Computing, 1993, 22(5):935～948

26 Boss B, Helmer S. Indexing a Fuzzy Database Using the Technique of Superimposed Coding - Cost Models and Measurements: [Technical Report TR-96-002]. Department for Mathematics and Computer Science, University of Mannheim, 1996

27 Gonnet G H, Baeza-Yates R A, Snider T. New Indices for Text: PAT Trees and PAT Arrays. In Information Retrieval: Data Structures And Algorithms, chapter 5, Prentice-Hall, 1992. 66～82

28 Baeza-Yates R, Ribeiro-Neto B. Modern Information Retrieval. Addison-Wesley, 1999

29 刘学文，陶晓鹏，于玉，胡运发. 一种全新的全文索引模型——后继数组模型. 软件学报，2002,(13)2:150～158

(上接第60页)

5 陈滢，徐宏炳，王能斌. 基于标记图的 Web 数据模型. 计算机学报，1999,22(3): 306～312

6 XML Path Language (XPath) 2.0. W3C Working Draft 02 May 2003. http://www.w3.org/TR/2003/WD-xpath20-20030502/

7 XQuery 1.0: An XML Query Language. W3C Working Draft 02 May 2003. http://www.w3.org/TR/2003/WD-xquery-20030502/

8 XQuery 1.0 and XPath 2.0 Data Model. W3C Working Draft 02 May 2003. http://www.w3.org/TR/2003/WD-xpath-datamodel-20030502/

9 王宁，徐宏炳，王能斌. 基于带根连通有向图的对象集成模型及代数. 软件学报，1998,9(12): 894～898

10 HTML Tidy. http://www.w3.org/MarkUp/#tidy