XML查询语言XQuery的研究与实现

华珊珊[1],谢铉洋[2]
(1.合肥学院 计算机科学与技术系,安徽 合肥 230601;
2.中国人民银行合肥中心支行 科技处,安徽 合肥 230022)

摘 要:XQuery是一种对XML结构的文档和数据进行查询的语言。在对该查询语言规范体系分析、理解和研究的基础上,提出了支持W3C的XQuery语言的查询处理引擎的体系结构。针对各个输入输出和处理模块按数据流的方式逐一进行分析,对整个系统的运行状态做了一个总体的介绍。按照这个体系结构,一个XQuery查询处理引擎已经被实现。

关键词:XML;XQuery;查询;处理引擎
中图分类号:TP312 **文献标识码**:A **文章编号**:1673-629X(2009)04-0048-03

Research and Implementation of XQuery

HUA Shan-shan[1],XIE Xuan-yang[2]
(1.Department of Computer Science and Technology at Hefei University,Hefei 230601,China;
2.Science and Technology Department,Hefei Central Sub-Branch,People's Bank of China,Hefei 230022,China)

Abstract:XQuery is a descriptive language which can query any XML-structured data,physically store in medium or viewed as XML. Based on the research of XQuery,illustrate the structure of our implement query engine. Give out a whole description of the running state of the system according to the analysis of the different modules. An XQuery processing engine has been realized according to this architecture.

Key words:XML;XQuery; Query; processing engine

0 引 言

XML[1]日益成为Internet上数据交换的标准,人们需要一种能够方便交换XML数据的手段。XML具有自描述性且与平台无关,所以交换XML数据的途径也应该是与平台无关的。虽然XPath2和XSLT在一定程度上解决了这个问题,但是其书写的复杂程度和功能的不完善远远不能达到人们的要求。为此W3C(World Wide Web Consortium,万维网协会)于2001年12月提出了XML查询语言规范(工作草案)—XQuery语言。

XQuery是一种功能强大的数据查询语言,它能够从XML文档中选择并抽取出复杂的模式,进而把查询结果重构成用户需要的新的XML文件结构。XQuery虽然作为一种语言规范被提出,但是其涉及的内容相当广泛。首先,作为一种描述型的语言和函数型语言,它在编译阶段就具有过程型语言的不同特点和要求;其次,作为一个对数据进行查询的语言,它需要有严格的数据模型支持。一个复杂的XML查询可由模式匹配、选择过滤、结果构造等构成。模式匹配用于在一个给定文档中按照给定的路径表达式匹配出所有满足条件的路径集合;选择过滤操作在模式匹配的基础上,从满足条件的路径集中选出符合条件的路径;结果构造用于将查询结果根据要求进行XML格式化。由于目前XQuery语言规范自身还是处于工作草案状态,通过对该语言的编译实现,从实践的角度出发对这些问题做一个探讨,很有理论的研究价值。

收稿日期:2008-08-04
基金项目:安徽省青年教师科研资助计划(2007JQ1153)
作者简介:华珊珊(1979-),女,安徽合肥人,讲师,研究方向为专家系统;谢铉洋,博士,研究方向为数据库、应用系统集成。

1 XQuery处理器的设计方案

1.1 从用户的角度

图1是从使用者角度观察XQuery处理程序的输入和输出。

作为用户,理想的处理过程是这样的:

用户只需要提供XQuery查询的源程序,由XQuery处理器负责处理其余的事务,并在一个可接受的时间内向用户提供查询的处理结果。

但是实际上,上述理想目标只能在有限的前提下

得到满足:

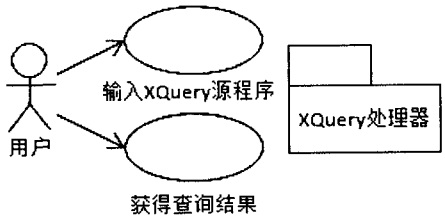

图 1　XQuery 处理器的用户视图

首先,XQuery 处理器拥有一个已经存储好可供用户查询 XML 文档的库。其次,XQuery 处理器在处理源程序之前,知道被查询的 XML 文档的结构类型。最后,由用户输入 XQuery 查询程序中的数据类型信息。这里所描述的是查询代码中的数据类型信息而不是被查询的 XML 结构类型信息。它说明 XQuery 查询中可能遇到的数据类型信息。该条件可选。

1.2　实际的输入和输出

图 2 为开发人员眼中的查询处理过程,是 XQuery 处理器实际的输入与输出。

从图中了解到,从开发人员的角度看,一个 XQuery 处理器的实际输入由 4 个部分组成:

(1)待查询的 XML 文档,这个 XML 从物理存储的文档或者由其他程序生产的 XML 数据流组成。

(2)待查询 XML 文档的格式说明,这个输入有物理存储的和其他应用生成的数据流两种方式。格式说明有 Document Type Define(DTD)和 XML Schema[2]。

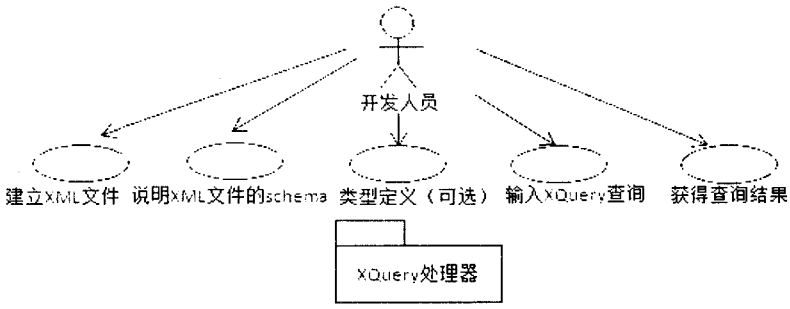

图 2 XQuery 处理器的开发人员视图

(3)用户输入的 XQuery 源程序。

(4)用户对自己输入的 XQuery 源程序中的数据类型进行说明,这个输入是可选的。

XQuery 处理器的输出由用户所要的查询结果和查询结果的结果类型两部分构成。

1.3　总体设计

综合上面的分析,提出了如图 3 所示的 XQuery 处理器的设计方案[3]。大致由 5 个部分组成:数据模型构造器,语法分析和语法转换器,类型处理器,运行环境和查询执行。

数据模型构造器将待查询的 XML 文档和文档格式说明转换为查询处理器课识别的格式。这个可识别的格式称为该文档的数据模型。数据模型的构造分为 XML 信息集的构造和 XQuery 数据模型。

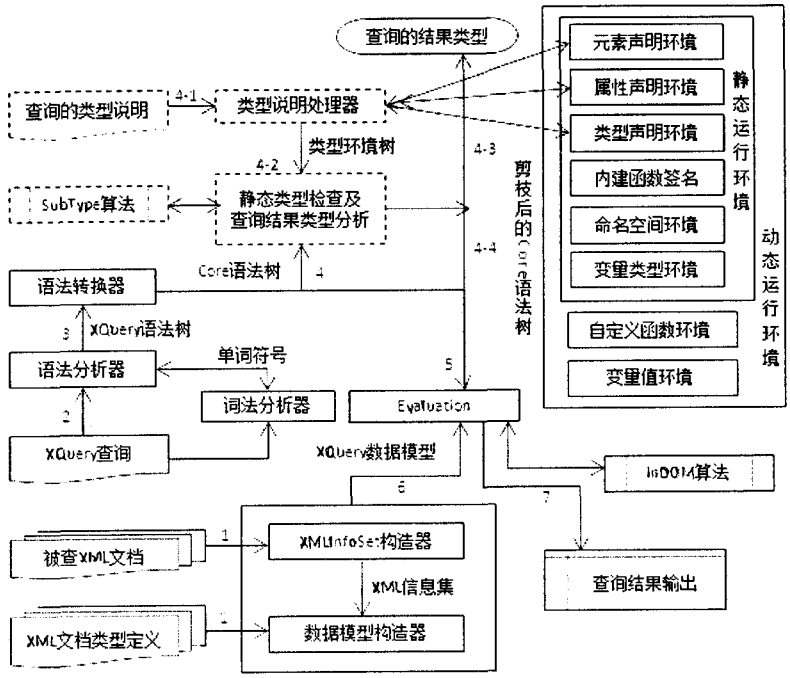

图 3　XQuery 处理器的构造

语法分析器和语法转换器负责将 XQuery 查询代码转换为语法树。语法树的形成分为两个阶段:第 1 阶段形成的是 XQuery 的语法解析语法树;第 2 阶段形成一个规范化的语法树。这两种语法树分别称为表层语法和核心语法,它们的转换是由语法转换器负责。

类型处理部分主要包括 XQuery 的类型说明处理器、静态类型检查和查询结果类型分析模块。它的主要任务是负责将 XQuery 源程序的类型说明转换为类型环境供类型分析模块进行查询的类型检查和分析。该部分的另外一个重要的功能是将 Core 语法树进行剪枝操作,删除那些计算时不可能经过的语法树分支,从而形成一个经过剪枝简化后的 Core 语法树。

运行环境部分:XQuery 的运行环境包括静态环境和动态环境。静态类型环境主要负责处理类型信息和运行时保持不变的映射表等。它包括下面 6 个部分:

元素声明环境负责处理元素声明的名称到元素类型体的映射关系,并由此可以提供元素类型供 XQuery 引用;属性声明环境负责处理属性声明的名称到属性类型体的映射关系,并由此可提供属性类型供 XQuery 引用;类型声明环境管理类型的名称到类型体的映射关系;内建函数签名负责 XQuery 内建函数的函数签名的管理,其信息供类型分析和赋值计算时查找以检查函数的参数是否匹配和从哪里加载该函数的实现;命名空间环境负责处理命名空间名称与实际地址的映射关系;变量类型环境负责管理变量的名称与该量类型的映射关系,用于类型检查和分析。

动态运行环境除了包含静态环境的所有组件外,添加了两个模块:自定义函数环境和变量值环境,分别用于维护自定义函数的信息和自定义的函数体等信息。从本质上说,运行环境是由一个个的系统表组成的。

查询执行部分是 XQuery 查询被真正计算出结果的模块。它的输入由两部分组成:一个是由语法转换器直接输出的 Core 语法树;另一个是由静态类型检查和查询结果分析器输出的经过剪枝操作后的 Core 语法树。

2 数据流分析

下面对图 3 中的各个输入输出和处理模块按数据流的方式逐一进行分析,以便对整个系统的运行状态有一个总体的了解。

图中标注 1 表示待查询的 XML 文档被交给 XML 信息集处理器。该处理器获得 XML 文档并对文档做解析处理后形成 XML 信息集(InfoSet)输出给数据模型构造器。XQuery 数据模型构造器将根据同时在 1 中输入的 XML 模式信息和 XML 信息集构造出支持 XQuery 查询的数据模型。该数据模型在 XQuery 值计算时被使用。

图中标注 2 表示用户输入的 XQuery 查询源程序被交给词法分析器和语法分析器。这里并不是按照首先词法分析再语法分析的两个阶段完成的,而是同时被交给语法分析和词法分析:当语法分析需要一个单词符号的时候向词法分析"索取"下一个单词符号,并最终由语法分析器输出一个 XQuery 查询的语法树。这个语法树称为 XQuery 语法树。语法转换器得到 XQuery 语法树并在这棵树上进行语法转换,将 XQuery 语法树转换为 Core 语法树。这里的语法转换是必需的,正如同编译执行的程序需要产生中间代码一样。在这里产生的是中间代码的语法树。注意这个语法树不是从语法分析得来的,而是由语法转换器直接转换来的。语法转换器生成的 Core 语法树被交给静态类型检查和查询结果分析处理器。注意这个步骤是可选的。XQuery 查询的类型声明被提交给类型说明处理器。在那里,类型说明被分解为 3 个部分:元素声明、属性声明和类型声明,分别被存贮到静态运行环境中的元素声明环境、属性声明环境和类型声明环境。并且该处理器在这三种声明的基础上构造出一棵类型环境树供静态类型检查使用。静态类型检查和查询结果类型分析器根据已有的类型环境树和 Core 语法树,在 SubType 算法的支持下进行类型的检查和推导。

经过上面的处理步骤后,可以在 4-3 得到查询结果的类型和在 4-4 得到一个一个被剪枝后的 Core 语法树。

由于 4 的可选性,在值计算模块 Evaluation 中将会有两个输入:3 中语法转换器输出的 Core 语法树和 4-4 输出经过剪枝的 Core 语法树。这两个 Core 语法树之间的差别在于类型分析处理器可以将一些根本不会被执行的语法树分支删除。值计算的另外一个输入就是在 1 中数据模型构造器输出的待查 XML 文档的数据模型。Evaluation 模块将在这个数据模型的基础上进行值的查找、过滤和输出等操作。Evaluation 模块完成在 Core 语法树上的计算并将计算结果返回给用户。

3 结束语

文中研究了一种 XML 查询语言 XQuery,提出了支持 W3C 的 XQuery 语言的查询处理引擎的体系结构。在该体系结构的指导下,定义 XQuery 处理引擎的 4 个主要步骤:词/语法解析、语法转换、类型检查和分析、赋值计算。并针对各个输入输出和处理模块按数据流的方式逐一进行分析,对整个系统的运行状态做了一个总体的介绍。按照这个体系结构,一个 XQuery 查询处理引擎已经被实现[3~5]。

参考文献:

[1] World Wide Web Consortium. Extensible Markup Language (XML)[EB/OL]. 2004-02-04. http://www.w3.org/XML/.

[2] World Wide Web Consortium. XML Schema[EB/OL]. 2004-09-22. http://www.w3.org/XML/Schema.

[3] 谢铉洋. XML 查询语言 XQuery 的编译实现[D]. 合肥:安徽大学计算机系,2002.

[4] 姚 吉. 支持 XQuery 的 XML 数据库研究[D]. 合肥:安徽大学计算机系,2002.

[5] 刘政怡. XQuery 查询语言的规范化[J]. 微机发展(现更名:计算机技术与发展),2003,13(6):50-52.