

基于 Web 的多形态新闻信息发布模型

马路 许斌 王克宏

(清华大学计算机系知识工程组 北京100084)

A Web News Information Publish Model With Multi-Format and Multi-Language Capability

MA Lu XU Bin WANG Ke-Hong

(Knowledge Engineering Group, Department of Computer, Tsinghua University, Beijing 100084)

Abstract There are a lot of different terminals existed in internet. Each of them has different demand on format and language when it retrieves information. This paper introduces a Web news information publish model with multi-format and multi-language capability. Within this model, users no longer need specify format and language when he retrieves information by any internet terminal. And information provider no longer need create copies for every format. A discrete model implement based on NewsML, a kind of XML standard used in news area, is presented in the paper.

Keywords XML, NewsML, FOP XSL XSLT FO

1 引言

当今的世界是一个信息爆炸的世界。每天,当人们连入互联网,扑面而来的是无穷无尽的信息。可是这些信息的冗余度非常高。在使用 Google 进行搜索时,大部分人所进行的搜索都会返回上千条结果,但也许其中很少的一部分所包含的是不重复的信息。信息的冗余主要表现在:

(1)不同的版式造成的冗余。也许只是因为字体大小不同,包含同样内容的两个文档将会被计算机程序认为包含两份完全不同的信息。

(2)不同的信息格式造成的冗余。包含同样信息的文本文件和 Word 文件,同样会让程序认为是两份不同的信息。

(3)不同的语言造成的冗余。同一篇英文文章的两种中文翻译,一样会被程序认为是包含两种不同的信息。

要解决信息的冗余问题,完全依赖人工智能的算法是不可行的。在这方面首先要做的就是统一信息的格式,XML 是人们在这个方向上做出的重大努力和成果。

虽然 XML 统一了信息格式,但信息如何表现给用户仍然是一个大问题。随着互联网的进一步发展,越来越多的终端设备开始接入互联网。手机、PDA 以及各种各样的信息家电,每一种终端都有自己对信息的表现格式的要求。同样,各个国家的用户也希望信息以自己熟悉的母语表现出来。你不能期望用户在阅读任何信息之前先在一个长长的下拉列表中去选择自己需要的格式和语言。即使这样可行,那么这个提供选择的界面又采用什么样的格式和语言呢?显然如果你的缺省语言是英语,那么对于大部分不懂英语的人来说就无法去进行这种选择。一个简单的解决方法是将内容复制成一份份的具有不同格式和语言的静态内容,然后将这些静态内容的 URL 告诉那些有特定格式和特定语言需求的用户。显然,要涵盖这么多格式和语言的组合数目是非常大的,这就使得信息的冗余问题仍然不能得到很好的解决。另一方面,信息的发布端要始终维护这么多实质内容一致的副本,也严重地浪费了资源。

综上所述,不仅仅需要统一信息的格式,而且还要提供一种具有多语言多格式表现能力的高效的信息发布模型。本文将从现实世界中最常见的新闻数据着手,研究一种基于 NewsML 这种新闻领域里的 XML 标准下的个性化的多形态信息发布模型。基于我们研究的模型,我们利用 XML 和 J2EE 技术实现了一套演示系统。利用这套系统,用户可以使用 PDA,手机,PC 等各种设备上的浏览器来透明地阅读新闻,即用户不需要指定需要浏览的格式和语言。我们这套系统会自动根据请求设备的能力来发送相应格式和相应语言的新闻。同时,我们这套系统具有良好的扩展性,开发人员可以在这套系统上很方便地添加对特定格式和特定语言的支持,充分地展示了 XML 的内容与显示分离的优势。

2 背景知识

2.1 XML 简介

XML^[1]是 Extensible Markup Language 的简写,一种扩展性标识语言。自从 XML 标准出现以来,它的发展异常迅猛。可以这么说,XML 是自 Java 语言出现以来,在 Internet 领域里最令人激动的发展。XML 是一套定义语义标记的规则,这些标记将文档分成许多部分并对这些部分加以标识。它也是元标识语言,即定义了用于定义其它与特定领域有关的、语义的、结构化的标记语言的句法语言。

2.2 NewsML 简介

NewsML^[2]是 XML 的产物之一,它能让新闻工作者及其他出版商以任何语言从金融服务桌上型电脑、网页到手机上,载取影像、文稿、图像、图片及视讯来制造和组合新闻。随着新闻业越来越数字化和数据库档案的不断增加,NewsML 旨在使新闻组织可轻松地自如地收集多媒体新闻报道,更方便地对这些多媒体新闻进行改编和存储,并且更容易在今后再次使用时找到这些新闻。NewsML 有以下几点特征,(1)具有良好的可扩展性和灵活性。(2)支持元数据和新闻条目内容的身份验证和签名。(3)允许新闻内容支持各种语言、格式和编

马路 硕士研究生,研究方向为知识工程与网络计算。许斌 讲师,博士研究生,研究方向为知识工程与网络计算。王克宏 教授,博士生导师,研究方向为知识工程与网络计算。

码。(4)使新闻内容可重用。(5)支持管理和开发过去的新闻内容。

2.3 NewsML 文件结构^[3]

NewsML 文件主要由三大块组成,首先是 TopicSet,里面描述了这篇新闻所涉及的主题,这将用于新闻的分类和搜索。第二块是 NewsEnvelope 标签,正如 Envelope 所表现出的意思,这里描述的是整块新闻所具有的一些属性。最后就是 NewsItem,显然这里是组成新闻的内容。注意内容并不需要完全嵌入到 NewsML 文件中,内容可以是链接,等显示的时候再取回来。图1是 NewsML 文件框架。

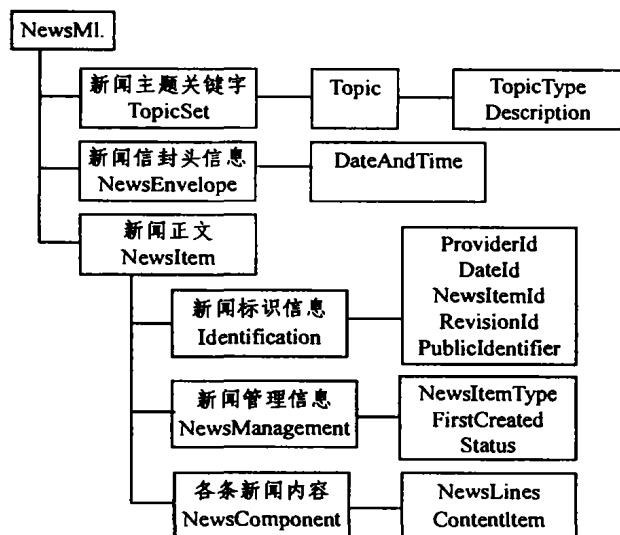


图1 NewsML 文件结构图

3 多形态新闻信息发模型的体系结构

利用 NewsML 的优秀特性,我们在 Tomcat^[4]上构建了一个采用 JSP 和 Servlet 技术的个性化的多形态新闻信息发模型。利用这个模型,系统将会根据用户终端设备的请求来智能地判断用户所需要的新闻格式和语言,然后调用相应的引擎来将存储在 NewsML 里的新闻内容转换为相应的格式,从而发送给用户。同时平台也提供了让用户显式地指定输出格式的功能。另一方面,该模型具有很好的可扩展性,可以很方便地加入对其它格式及语言的支持。目前该模型支持的格式有 HTML、CHTML^[10]、WML^[9]、PDF 以及 MIF,支持的语言包括中文和英文。图2是系统的整体架构。

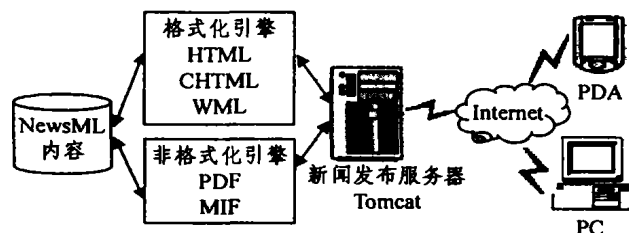


图2 系统结构图

在转换引擎的设计上,我们主要采用 XSL 技术,对于 HTML、CHTML、WML 这类结构化信息来说,只需采用 XSL 中的 XSLT^[6]部分,这类格式化信息转换引擎的结构如图3。

对于 PDF、MIF 这类非结构化信息处理起来就要稍微麻烦一些,这里要用到 XSL 里的另一部分 XSL-FO^[7]。这些非

格式化信息转换引擎的结构如图4。

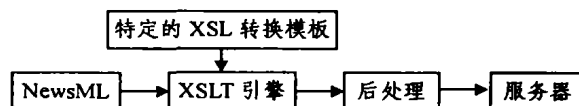


图3 格式化信息转换引擎结构图



图4 非格式化信息转换引擎结构图

4 系统实现

4.1 系统开发环境

新闻发布平台的服务器软件采用 Apache 开发的 Tomcat v3.2。开发语言采用的是 Java。在 XML 的解析和 XSLT 的引擎方面,利用了 Sun 公司的 JAXP^[5]包,版本是1.1。产生 PDF 和 MIF 格式的 FO 处理引擎采用的也是 Apache XML Project 里的 FOP^[6],版本是0.20.2。WML 开发上采用的是 Ericsson 公司的模拟器。

4.2 关键技术

4.2.1 个性化发布 为了实现对用户透明的新闻发布,平台对所有用户提供一个缺省的访问接口。用户在任何浏览器下只需浏览这个 url,然后通过截获用户 HTTP 请求头中的信息来得知用户的请求偏好。通过分析请求头中的 Accept-Language 信息,可以得知用户浏览器所用的语言。通过分析请求头中的 Accept 信息,可以得知用户所需要格式的 Mime 类型。通过得到上述信息,系统便可以自动返回用户所需要的新闻格式和语言。当然,可能用户不希望以它的浏览器的缺省偏好来获取新闻,例如用户用中文版的 Internet Explorer 来看英文新闻,另外还有可能不存在能缺省请求某种格式的浏览器,例如还没有缺省请求 pdf 格式的浏览器。对于这两种情况,用户就必须显式地指定所需要的格式和语言,一方面可以在缺省返回页面中提供相应的按钮,另一方面用户也可以直接在 url 中输入调用参数。

4.2.2 系统流程 整个系统的处理流程图如图5所示。

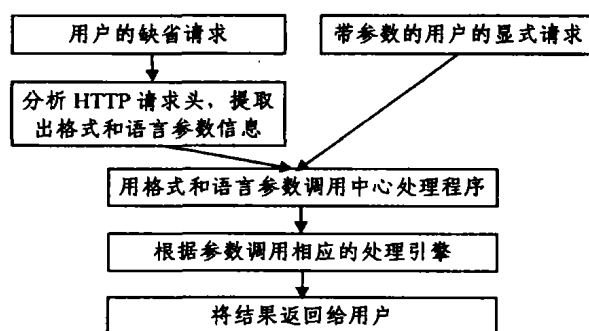


图5 系统处理流程图

4.2.3 格式化处理引擎 对于每一种特定格式化信息的处理引擎,其核心是一个 xsl 文件。每一种格式都对应一个 xsl 文件,在用户请求相应的格式时,首先通过使用相应的 xsl

来提取出 NewsML 文件中想要的信息。对于各种格式有着不同提取策略,例如对于 WML 格式来说,xsl 就无需理会 NewsML 中的新闻图片信息,因为绝大部分图片都无法在手机上显示。

4.2.4 非格式化处理引擎 对于用户所需要的非格式化信息来说,无法通过 xsl 来一步达到目标。例如显然无法通过 xsl 直接从 NewsML 中提取出 pdf 格式的信息。要描述这类信息,必须借助 XSL-FO 来实现这一点。对每种非格式化信息,同样用相应的 xsl 文件来从 NewsML 文件中提取需要的信息,同时把这些信息组织在 FO 文件中。然后通过对产生的 FO 文件进行再处理来产生相应的格式。

4.2.5 FOP 对于格式化信息来说,要自己实现一种新的格式处理引擎并不困难,但对于非格式化信息来说,要完全自己实现一种新的格式就很困难了,例如从 FO 到 PDF 的转换引擎,工作量是非常大的。可以利用 FOP 来实现这一点。

FOP 是由 James Tauber 发起的一个开源项目,最初的目的是利用 XSL-FO 将 XML 文件转换成 PDF 文件。目前 FOP 已经归并到 Apache 的 XML Project 中,最新版本是2002年7月4日发布的0.20.4,它可以将 XML 文件转换成 PDF, MIF, PCL, TXT 等多种格式以及直接输出到打印机,并且支持使用 SVG 描述图形。FOP 的开发目的是希望借助这种 XML To PDF 的巨大需求,来推动 XSL-FO 规范的发展。

在本系统的实现中,利用了 FOP 来作为 FO 到 PDF 和 MIF 格式的引擎。对于那些不希望采用 FOP 的用户来说,也可以去选择其它的 FO 转 PDF 的商业产品。

5 系统性能的改进

在上述模型中,系统的可扩展性和信息冗余度的减少是以系统的性能为代价的。XML 虽然好,但不可能直接把原始的 XML 文件展现给用户。用户需要看到的是经过格式化处理的新闻。在本模型中,XSL 被大量应用,这样对于每一个用户请求,系统都要进行解析 XML,解析 XSL 以及进行 XSLT 转换工作。对于新闻这种被大量访问的信息来说,上述工作构成了这个模型系统的瓶颈。至于如何优化和提高处理 HTTP 请求的速度,那是 Web Server 的工作。下面将讨论针对前面提到的任务瓶颈的解决方案。

首先需要分析新闻数据的特性。显然,可以得出以下几个特性:(1)经常被重复读取。(2)内容很少被改动,修改已经发布的新闻是很不常见的。(3)时效性比较强,比较旧的新闻很少再被阅读。

通过分析上述新闻数据的特点,可以引进 Cache 机制来解决性能问题。对于每次 XSL 转换后的结果,将它放入 Cache,这样当另一个用户请求同样的格式和语言的内容时,就无需再进行转换工作。图6是 Cache 机制的结构图。

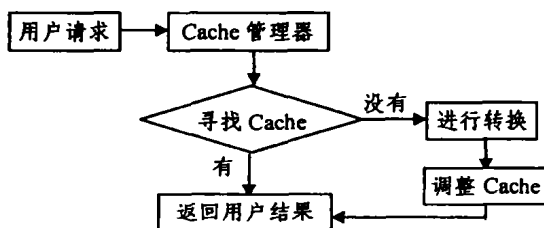


图6 Cache 机制处理流程

注意这同本文一开始提到的那种浪费资源和增加冗余的

简单复制方法是两回事。首先这种解决方法对用户是透明的,信息对外仍然只是一个 URL,不同于前面方法需要每份拷贝一个 URL,这样就避免了信息的冗余问题。其次这里的 Cache 是有生存时间的,Cache 的内容将会访问频率和时间进行调整,不同于前面方法的永久保存。

至于 Cache 的调整策略,采用的是类似操作系统中的调度策略。Cache 中的每一项具有一个初始 nice 值,用户每访问一次 nice 值将加1,每过一天 nice 值将除2,通过 nice 值的大小来决定是否保留在 Cache 中。实际应用时,用户可以根据系统运行的具体情况来调整上述数值以获得最优的效率。

最后需要提到的是,上述性能改进方案是跟特定类型的新闻数据相关的,因此对于不同于新闻的其它种类信息,如何进行性能的改进需要单独的分析。虽然如此,这里给出的针对新闻这种特定信息的性能改进方案,将对大多数其它类型信息的发布性能改进具有指导作用。

6 演示系统

图7~图9是我们实现的系统的几个输出界面。



图7 HTML 输出

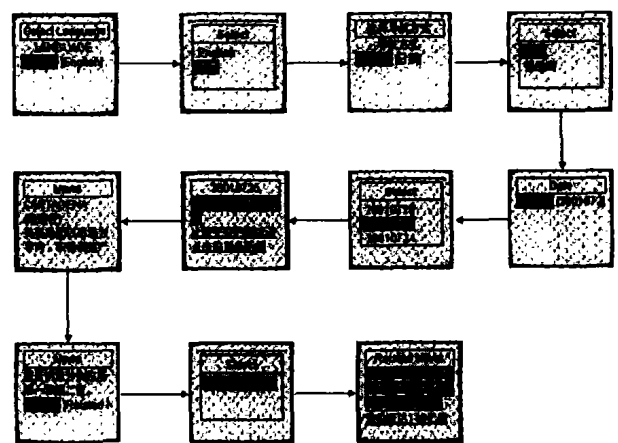


图8 WML 输出

结束语 本文通过以 NewsML 为例,提出了一种基于 XML 的多形态新闻信息发布模型。经过实验系统的验证,随着接入互联网的终端设备类型的增多,这种基于 XML 多形态信息发布系统对于新闻信息的多格式、多语言的发布具有很好的实用性和扩展性,对于格式化信息和非格式化信息提

出了相应的处理方法。系统的实现采用的是业界成熟的 J2EE 技术和 XML 技术,具有良好的跨平台性,并提供了提高性能

的办法。用户可以根据自己的具体情况,参考本文中的模型上实现自己的系统。

参考文献

- 1 XML 规范: <http://www.w3.org/xml>
- 2 NewsML 主站点: <http://www.newsml.org/>
- 3 NewsML v1.0 规范: <http://www.iptc.org/site/NewsML/specification/NewsMLv1.0.pdf>
- 4 Apache Tomcat: <http://jakarta.apache.org/tomcat>
- 5 Sun JAXP: <http://java.sun.com/xml>
- 6 FOP 主站点: <http://xml.apache.org/fop/>
- 7 FO 规范: <http://www.w3.org/TR/xsl>
- 8 XSLT 规范: <http://www.w3.org/TR/xslt>
- 9 WML 规范: www.wapforum.org/what/technical/SPEC-WML-19990616.pdf
- 10 CHTML: <http://www.w3.org/TR/1998/NOTE-compactHTML-19980209/>

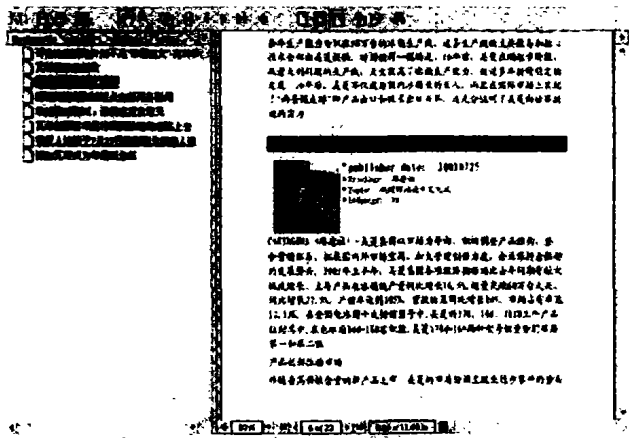


图9 PDF 输出

(上接第75页)

等,以提高系统性能。

3.2 编程语言和数据库对象元数据冗余的处理

根据体系结构中的元数据管理将编程语言中的类加入数据库中,如果数据库本身提供的类与编程语言提供的类功能相近时,会导致数据库中的冗余不断增加。解决的方法是在两者之间建立对应关系。这种方法需要人为地建立对象及其成员方法之间的对应关系,显然这样做的工作量很大。

3.3 对象方法的运行管理

对象方法可以运行在客户端,也可运行在服务器端。如果全部放入 ORDBMS 中运行,数据库服务器将成为运行瓶颈。如何确定方法的运行是否放在服务器上要根据下列各种因素:方法所需参数的容量,方法需要从数据库中获取的数据量的大小,网络带宽的大小,服务器/客户机的性能差距和当时的负荷情况,方法的代码量等。针对不同的情况,ORDBMS 要提供对不同的两种运行方式的支持和判断。

对象方法运行于服务器端时安全性控制是个关键,目前解决的方法有^[5,9]:

(1)DBMS 进程与函数运行进程分开,函数运行进程由用户 ID 标识。这种方法的缺点是引进了远程过程调用及其相关的开销,降低了性能。

(2)单独开辟一个进程,用来检查函数代码中的所有的跳转语句,可以采用在每个跳转语句的前后插入一些检查代码,或者对于每个跳转命令,进行系统调用两种方法。由于这种方法将程序运行于 DBMS 的查询处理进程中,开销比较小。

(3)采用解释型语言如 Java 来编写函数,将边界检查的工作交给语言解释器来完成。

(4)在体系结构上采取硬件措施,缺点是不能保证所有的机器都有这样的安全机制。

展望 本文在回顾了传统数据库模型的不足和 ORDBMS 的发展后,针对 ORDBMS 的实现提出了一个新的结合 Java 的体系结构。其主要目的是:

(1)实现编程语言对象和 ORDBMS 元数据之间的透明转化,简化了应用开发的难度,提高了编程语言和 ORDBMS 中对象的集成度。

(2)系统具有很好的可扩充性:从数据库元数据的生成上看,采用类库元数据生成器,方便了数据库元数据的扩充;从接口转换上看充分利用 CORBA/J2EE 技术,便于系统的应用集成;从数据库对象的实现语言上看采用 Java 具有很好的跨平台性。

基于上述的 ORDBMS 体系结构,还可以扩展出以下的一些特性:

(1)易于与中间件集成。由于结合了 Java 和 CORBA 中的 IDL,加上 JAVA 本身对于 CORBA 的支持,就可以很容易地集成到中间件平台中。

(2)可以拓展成分布式数据库。因为数据库类库具有对 CORBA 的支持,只要解决分布的对象之间的同步更新(即一致性)和事务处理问题,就可成为一个具有良好的垂直划分分布式特性的数据库。

(3)这种体系结构的 ORDBMS 也可扩展到其他编程语言,只要它对数据库交互接口和数据库元数据导入工具支持,就可以实现与 DBMS 结合。

参考文献

- 1 宛延恺.工程数据库系统.清华大学出版社,1999
- 2 Peter M, et al. Object-Oriented DataBases: A Semantic Data Model Approach. Prentice Hall, 1992
- 3 冯玉林, 黄涛, 倪彬. 对象技术导论. 科学技术出版社, 1998
- 4 Stonebraker M, Brown P. Object-Relational DBMSs: tracking the next great wave. Morgan Kaufmann Publishers, Inc., 1999
- 5 鞠时光. 对象关系型数据库管理系统的开发技术. 科学出版社, 2001
- 6 Fortier P J. SQL-3: Implementing the Object-Relational Database. McGraw-Hill, 1999
- 7 Saracco C M. Universal Database Management: A Guide to Object-Relational Technology. Morgan Kaufmann Publishers, Inc., 1999
- 8 Date C J, Darwen H. Foundation for Future Database Systems. Addison-Wesley, 2000
- 9 Brown P. Object Relational Database Development: A Plumber's Guide. Prentice Hall PTR, 2001
- 10 Oracle 用户手册. Application Developer's Guide - Object-Relational Features. Oracle Corporation, 2001
- 11 <http://www.sqlstandards.org/SC32/WG3/Progression-Documents/CD/cdlr1-olb-2001-06.txt>, SQL/J 标准, 2001