

一种基于 XML 的软件动态演化方法

李申莉¹, 王振宇¹, 王亮², 方木云^{1,2}

(1. 安徽工业大学 计算机学院, 安徽 马鞍山 243002;

2. 中国科学技术大学 信息科学技术学院, 安徽 合肥 230027)

摘要:随着操作环境与用户需求的不断变化, 现有系统无法满足新的处理需求。如何将现有系统的大量数据提取并转换成新的应用是一个非常迫切需要解决的问题, 而利用 XML 的可扩展性、开放性和互操作性等优点可以将现有系统的数据转化成新的应用。详细介绍了一种基于 XML 的动态演化方案并给出了方案的框架设计。该框架主要由 DOM, XSLT, Xquery, XML, 转换器等几部分构成。通过简单的应用实例展示了该方案可以基本实现系统的动态演化功能。

关键词:可扩展标记语言; 动态演化; 现有系统; 文档对象模型; XSL 转换; XML 查询

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2009)03-0039-03

An Approach Based on XML to Software Dynamic Evolution

LI Shen-li¹, WANG Zhen-yu¹, WANG Liang², FANG Mu-yun^{1,2}

(1. Department of Computer Science, Anhui University of Technology, Ma'anshan 243002, China;

2. Sch. of Info. Sci. & Tech., University of Science and Technology of China, Hefei 230027, China)

Abstract: Legacy system can't satisfy new requirements along with the changing operation circumstance and user's demands. How to convert the useful data in the legacy system and achieve new applications is an important issue. Because XML has expandable, exoteric and mutual manipulation characters, it can convert the legacy system's data to new applications. Introduces a kind of dynamic evolution based on XML in details and offers the design frame. The frame is made up of DOM, XSLT, XQuery, XML and switch part. The design could implement the legacy system dynamic evolution proved by a simple example.

Key words: XML; dynamic evolution; legacy system; DOM; XSL transform; XML query

0 引言

随着时间的推移, 现有系统的用户需求在不同程度上发生改变; 软件的实现技术和计算环境也在不断的改善, 如在数据访问技术方面, 计算资源的增长和数据访问技术能力的增强驱动了更多访问的需求; 新的计算模型的不断推出使人们能更精确地刻画、理解和分析问题域(Problem Domain), 如对象和 Agent 的计算模型; 新的软件工程方法学, 如基于组件的软件工程^[1](CBSE)允许在组件的基础上装配符合需求的系统; 所有这些情况的改变都要求软件工程师在考虑新的用户需求和技术的情况下, 分析现有系统应该扮演什么样的角色以及它与新的需求之间的关系, 而演化的商业

和组织环境也要求软件具有可演化特性以适合未来持续变化的操作环境与用户需求, XML 应运而生。XML 以其自身的可扩展性、开放性和互操作性等特点成为新一代网络语言。利用 XML 将现有系统的数据提取并转换成新的应用, 具有很重大的实际意义。

1 XML 和现有系统

XML^[2]规范提供了使用嵌入式标记描述数据的框架。XML 让用户定义特殊的可以描述领域相关数据的内容和结构。通过使用 XML 标记数据, 计算机能够读取、标识和解释数据, 从而产生工业相关的数据语言、产生中立的数据交换以及能够使用智能客户机的媒体独立的出版技术。

XML 促使了以数据为中心的现有系统集成方法的发展, 它超越了对象包装器或者专用的企业应用程序集成方案。XML 实质上解放了数据对软件体系结构的依赖, 其数据模型的层次关系能够很好地适合于面向对象数据库的存储, 数据库厂商通过添加 XML

收稿日期: 2008-06-05

基金项目: 安徽省教育重大项目计划(ZD2008005-1); 安徽省计算与通信软件重点实验室资助项目

作者简介: 李申莉(1980-), 女, 安徽舒城人, 硕士研究生, 研究方向为软件工程、信息系统和数据库; 方木云, 副教授, 研究方向为软件工程、软件度量及软件可靠性。

存储来扩展功能。WWW 组织已经制定了 XML 查询语言的规范,这使得基于内容的文档搜索变成可能。

将数据从软件依赖的范围中解放出来,可以使用存在的 Web 协议以便将 XML 作为 Web 上数据交换的标准。基于数据的新中间件(而不是组件界面)会允许更多的现有系统集成,因为中间层服务器能够将专用的标记——XML,变换为应用程序需要的任何形式。当前令人们感兴趣的一种趋势是 Web Distributed Data Exchange^[1](WDDX,一种使用 HTTP 在 Web 上交换和存储 XML 数据的标准)。WDDX 支持包括数组、数据词典、散列表、数组、链表、字符串以及 Boolean 值,使得它可以成为基于对象的基础设施(诸如 CORBA、COM 或者 EJB)的可替换方案。和 WAP(与 XML 兼容的 Wireless Application Protocol)结合后,信息可以通过 Web 下载到手持智能电话和 PDA 上显示。

另一个值得注意的是 Simple Object Access Protocol(SOAP),它由微软支持并且提交到 Internet Engineering Task Force(IETF)作为 Internet 草案,SOAP 定义了应用程序使用 HTTP 和 XML 通信的协议。

2 基于 XML 的动态演化方案

针对现有系统中大量采用的普通数据文件的不足,充分利用 XML 文件模型的自身优点以及 XML 在开源软件和商用软件业界获得的广泛支持,文中提出了一个基于 XML 的针对现有软件进行动态演化的方案。该方案的结构如图 1 所示。

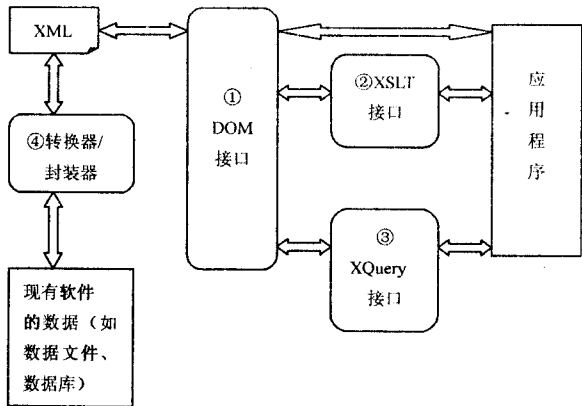


图 1 基于 XML 的动态演化方案结构图

该解决方案的主要组成部分包括:DOM^[3]接口、XSLT 接口、XQuery 接口以及转换器或封装器,它们分别提供了标准的 XML 文档的解析、样式表现接口、查询以及 XML 和现有系统间的转化。

下面逐一介绍一下 4 个接口:

1)DOM 接口。

DOM 的逻辑结构图如图 2 所示。

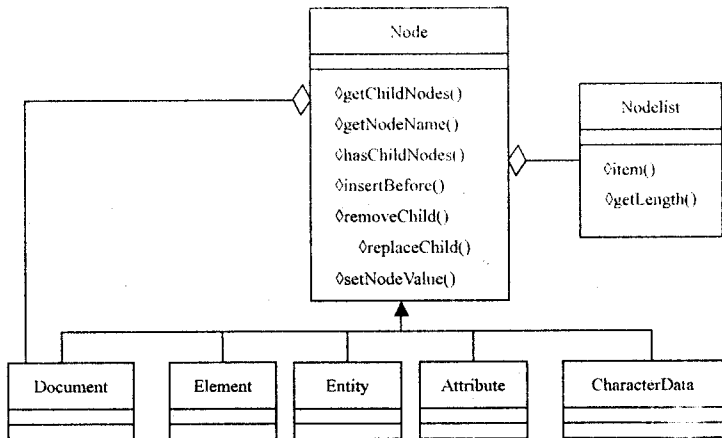


图 2 DOM 的逻辑结构图

图 2 中的“Document”对象和“Node”对象之间形成了一个反向包容环。Document 是一个节点的集合(容器),而它本身又是一个节点。

XML 文件模型对于树形文件模型和对象文件模型都是非常有用的模型。DOM 结合了两者的特点产生了它自己的混合法,能够同时满足应用程序对于接口文件的两个看似相互冲突的要求,操作简单,意义直观,编程灵活;支持广泛,查询方便,搜索效率更高。

2)XSLT^[4]接口。

XSLT 是由 XSL(Extensible Stylesheet Language)发展而来的,一种基于 XML 的语言。它可以用于将一类(class)XML 文档转换成另一类 XML 文档。XML 文件可以指定适用的 XSLT 文件。采用动态 apply 不同的 XSLT 文件,比如:

```

<% @ LANGUAGE = JScript %>
<%
// Set the source and style sheet locations here
var sourceFile = Server.MapPath("simple.XML");
var styleFile = Server.MapPath("simple.xsl");
// Load the XML
var source = Server.CreateObject("MSXML2.DOMDocument");
source.async = false;
source.load(sourceFile);
// Load the XSLT
var style = Server.CreateObject("MSXML2.DOMDocument");
style.async = false;
style.load(styleFile);
Response.Write(source.transformNode(style));
%>
    
```

3)Xquery^[5]接口。

为了使 XML 成为一个真正能实现互操作性的语

言,必须有一个一致的语法在各种类型的应用程序中对 XML 文档进行查询,这就是 XML 查询语言(XQuery)。从最简单的层次来说,XML 文档的查询可以被认为是对 XML 文档内容的查询。更进一步来说,它可以被认为是一种返回 XML 文档信息的方法。从这个意义上来说,XQuery 与结构化查询语言(SQL)类似,均是基于模板匹配技术。

XQuery 能完成下列操作:

(1)从大型 XML 文档中抽取数据,以及对多个 XML 文档进行查询,并合并所得结果。

(2)在不同的 DTD(文档类型定义)之间转换、翻译 XML 数据。

(3)XQuery 采用 XML 路径语言(XPath),支持路径表达式形式的层次查询。

举例说明:XQuery 常用的查询匹配模板如表 1。

表 1 常用的查询匹配模板

匹配模板	含义
eName	匹配名称为 eName 的所有元素
eName1 eName2	匹配名称为 eName1 或者 eName2 的所有元素
eName1 / eName2	匹配名称为 eName2 并且是 eName1 的子元素的所有元素
eName1 // eName2	匹配名称为 eName2 并且是 eName1 的后代的所有元素
@ aName	匹配属性为 aName 的所有元素
*	匹配所有的元素
@ *	匹配所有的属性
eName1 [eName2]	匹配名称为 eName1 并且包含子元素 eName2 的所有元素
eName [@ aName]	匹配名称为 eName 并且具有属性 aName 的所有元素
.	匹配当前正在处理的节点(通常是一个元素)
..	匹配当前正在处理的节点的父节点
comment()	匹配当前节点中的所有批注

XQuery 的优势在于:

①能够索引同属元素的次序。

例如,在部队建制中,建制的次序通常是传递着一定意义的。而在记录集中,记录的次序是没有意义的。当然,次序没有意义是关系理论所要求的(这样,通过建立索引,搜索记录就变得更为容易,在性能上也得到了优化),关系表当中的这种次序关系必须通过额外的附加字段进行描述。

②能够描述父/子、祖先/后代这样的次序关系。

③能够返回一个新的、从旧文档中构造出来的结构良好的 XML 文档。

4)转换器或封装器。

使用某一种编程语言(比如 C#)将现有系统的数据和 XML 相互进行转换,以满足两者处理的需要。

3 简单应用实例分析

基于上述方案的流程图可简化如图 3 所示。

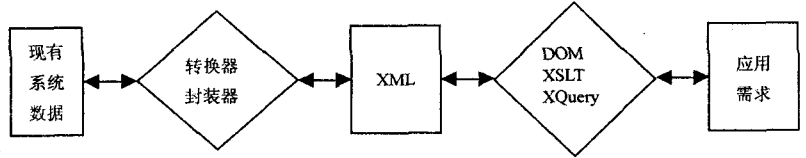


图 3 方案简化流程图

下面给出一个 XML 和现有系统数据集 DataSet 互相转换的 C# 源代码:

```

<script runat="server">
void page_load(object o,EventArgs e)
{
    zfwyldb db=new zfwyldb("db.mdb");//创建数据库对象
    Response.Write(db.geterrormsg());
    System.Data.DataSet ds=null;//初始化系统数据集
    System.Data.DataView dv=db.select("select * from test");//
    定义系统数据集对象并加载方法
    Response.Write(db.geterrormsg());
    string dbstring="";
    string data="";
    string structmsg="";
    if (dv!=null)
    {
        ds=dv.DataViewManager.DataSet;//接收管理器数据集
        dbstring=ds.GetXmlSchema()+ds.GetXml();//获取
        XML 模式字符串
        data=ds.GetXml();//获取 XML 对象数据
        structmsg=ds.GetXmlSchema();//设置结构体信息为
        XML 的模式
        System.IO.TextReader tr=null;
        System.IO.StringReader sr=new System.IO.StringReader(da-
        ta);//创建系统 IO 字符串对象
        System.Data.DataSet nd=new System.Data.DataSet();//创
        建系统数据集对象
        nd.ReadXml(sr);//获取 XML 转换后的信息
        sr=new System.IO.StringReader(structmsg);//读取系统 IO
        中获得的信息
        nd.ReadXmlSchema(sr);//系统数据集读取 XML 转换后的信
        息
        int i=0;
        Response.Write(dbstring.Replace("<","<"));
    }
</script>
  
```

4 结束语

现有软件当中大量采用的普通数据文件给软件的演化带来了极大的难度,这是由它的文件模型所决定的。基于 XML 的现有软件演化解决方案充分利用了

(下转第 45 页)

文件例化到顶层文件中的程序存储器 ROM 中去。最后,将生成的配置数据文件(*.bit)采用 JTAG 模式下下载到开发板中的 XC3S500E-4FG320C 芯片中去,开发板中的八个 LED 灯从右到左循环点亮,验证了其正确性,如图 5 所示。

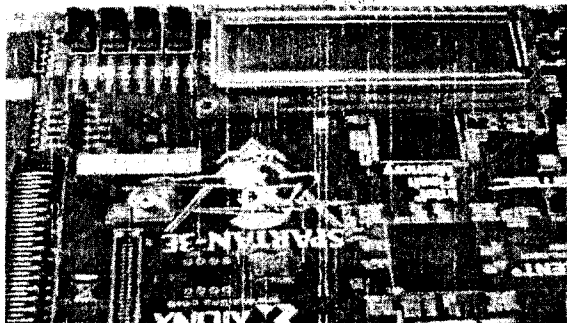


图 5 LED 灯运行结果

3 结束语

在对传统 8051 体系结构进行系统分析的基础上,结合实际应用对软核的控制逻辑进行优化设计,提高了软核运行效率,优化设计了多时钟周期指令,减少了所需的执行时钟周期。因此,所设计出的 8051 内核的最高工作频率及指令执行效率都有相当的提高。由于整个 8051 内核都是采用可综合的 VHDL 语言,使用 RTL 级的语法进行描述,使得该内核具有很好的可移植性、可重复利用性和实用性。

通过对 8051IP 核的研究,明晰了其内部模块结构

和各个模块之间的具体协作细节,特别是对其内部的指令执行控制过程有了深入的认识。同时,也掌握了可编程逻辑器件设计的一般方法和开发流程,为以后进行更复杂的 SOPC 设计积累了经验。

参考文献:

- [1] 陆重阳,卢东华. IP 技术在 SOPC 中的地位及应用[J]. 微电子技术,2002(8):20-23.
- [2] 潘 松,王国栋. 基于 EDA 技术 CPLD/FPGA 应用前景[J]. 电子与机械自动化,1999(3):3-6.
- [3] 余 翔,熊光泽. SOC 芯片的 TOP-DOWN 设计方法[J]. 电子科技大学学报,2002,31(6):585-589.
- [4] 张 凯,汤志忠. 通用 16 位 CPU 的设计与实现[J]. 计算机工程与应用,2002(32):116-118.
- [5] Oregano Systems. MC8051 IP Core Synthesizable VHDL Microcontroller IP Core[J]. Journal of Systems Architecture, 2002(9):11-50.
- [6] 李亚民. 计算机组成原理与系统结构[M]. 北京:清华大学出版社,1999:126-177.
- [7] 尚 笠,葛元庆,周润德. 80C51 微处理器嵌入式内核的设计研究[J]. 微电子技术,2000,30(1):28-30.
- [8] Halverson R Jr, Lew A. FPGAs for expression level parallel processing[J]. Microprocessors and Microsystems,1995(19):533-541.
- [9] 孟宪元,钱伟康. FPGA 嵌入式系统设计[M]. 北京:电子工业出版社,2007:178-212.
- [10] 周宁宁,刘 胜. 基于 FPGA 技术的 CPU 的设计与实现[J]. 南京邮电学院学报,2003,23(1):77-79.

(上接第 41 页)

XML 文件模型的优点以及 XML 在开源软件和商用软件业界获得的广泛支持,可以较好地解决现有软件的演化难题。基于 XML 的现有软件演化的解决方案具有如下的这些优点:

* 带有自描述、可扩展、富含语义特征的标记,使得数据的含义十分直观。

* 若是标记的含义还不够明确和充分,可自由注释,而不会影响文档的解析。

* 在 DTD 的帮助下可以创建出更加规范、容错和智能的文档。

* 具有标准、统一、通用的解析、查询和样式显示接口,避免了特定的文档格式的限制,以及接口程序的重复开发和分发,提高了开发效率。

然而随着 XML 越来越多地作为信息交换标准,持久、验证和查询 XML 文档的功能也越来越重要,因此要不断丰富 XQuery 的语义。同时,随着 Web 服务

和 mash-up 应用程序的泛滥,Web 应用程序也需要更多地转换 XML 信息,这就需要不断完善 XML 的模式。

参考文献:

- [1] 詹剑锋. 因特网环境下的软件演化与动态性研究[D]. 北京:中国科学院软件研究所,2002:2-4.
- [2] 马淑娇,李 晓,周俊林. 异构数据库集成中的 XML 技术探讨[J]. 计算机应用研究,2004,21(1):94-96.
- [3] 张 华,董 慧. JDOM 解析 XML 文档及其在数据转换上的应用[J]. 现代图书情报技术,2005(11):86-87.
- [4] W3C World Wide Web Consortium. Extensible Markup Language (XML) [EB/OL]. 2007-05-08. www.w3.org/XML.
- [5] Choi M J, Hong J W, Ju H T. XML-Based Network Management for IP Networks[J]. ETRI J, 2007, 25(6):3-4.