

统一消息中邮件的 XML 标准化方法^{*})

王守芳 樊闻斌 金 浩 潘金贵

(南京大学计算机软件新技术国家重点实验室 南京大学计算机科学与技术系 南京 210093)

摘 要 原始邮件松散的结构和过于灵活的定义给应用程序分析带来了困难,为了给基于邮件的应用程序开发提供统一的结构化表示和统一管理各种消息,可以选择 XML 来标准化邮件。在对邮件的语法结构进行简要分析的基础上,分别给出了 XML 表示邮件头和邮件体的方法,该方法解决了复杂邮件实体结构的表示问题。还对这种表示方法的性能进行了分析。

关键词 电子邮件, XML, 多用途 Internet 邮件扩展(MIME)

XML Standard Representation of Email in Uniform Messaging

WANG Shou-Fang FAN Wen-Bin JIN Hao PAN Jin-Gui

(State Key Lab for Novel Software, Department of Computer Science and Technology, Nanjing University, Nanjing 210093)

Abstract The relax structure and the flexible definition of the email bring difficulty to the programs to analysis, a uniformed representation is needed now for applications in emails and managing kinds of messages. XML is a good choice for representing email in a uniformed form. In this paper, based on an overview of the syntax structure of email, the method of representing the mail header and the mail body is presented. The method solves the representation of complex email entity structure. We give the performance analysis of the presentation of this method at last.

Keywords Email, XML, Multipurpose Internet Mail Extensions(MIME)

1 引言

随着网络技术的迅速发展和 Internet 的普及, email 已作为商务或朋友之间互相联系的主要手段。现在电子邮件可以传递文本信息、超文本信息和其他各种类型的二进制附件,甚至可以通过邮件调用各种程序,比如访问 FTP/WWW 服务器、订阅电子杂志、查询资料等。这类应用一般是邮件服务器将接收到的邮件转交给被访问的应用程序进行处理,处理结果被组织成邮件再通过邮件服务器发送给访问者。

但是传统的 email 结构松散定义过于灵活, XML 作为一种能够以统一的格式描述信息的文本语言,可以克服这些缺点,提供具有良好语法结构的信息,可以省去第三方对邮件的分析工作。目前, XML 能将不同系统来源的信息按照统一的格式显示,也能够进行格式的互相转化,已经成为当前信息标准化的主流。

另一方面,通讯方式呈多样化发展,但来自不同媒介的信息相互独立,缺乏关联,各种不同的终端设备以不同的方式进行不同信息的交互显得非常不方便,越来越多的通讯方式引发了一个管理的问题,不易于统一管理,这就需要将现行的通讯系统关联起来,使不同媒介的信息相互转换,又能通过一个平台统一管理这些信息。整合各种通讯方式(语音、短消息、电子邮件、Web 访问等),并对各种信息加以智能管理、转换、分发、存储的新一代实时性通讯系统,这就称为“统一消息”^[1]。统一消息使得用户可以在任何时间、地点、设备访问

用户拥有的信息,提高工作效率。要统一化消息,首先必须进行各种消息的标准化,这样不同媒介的信息才可以相互转化。

本文的目的是提出一种统一消息中的邮件的 XML 标准化方法,这种方法兼容于其他格式消息,分别包括邮件头和邮件体,并以邮件的 XML 表示为基础,讨论了 XML 邮件的应用。

2 邮件格式以及 XML 表示

XML(eXtensible Markup Language)是一种包含结构化信息的标记语言^[2],在不同的应用领域 XML 可以按照该应用领域的特点构造自己所需的标记(tag), XML 适合描述文档的内容和结构,而不关心其展现格式,这使得人们阅读更清晰,也使非人类、机器定位文档中的各类信息更加容易,邮件的原始信息复杂,难以理解,对邮件使用 XML 来表示可以使其充分结构化,解析后的 XML 文件结构清晰,适合在不同的结构中进行传输以及处理。

2.1 邮件的基本组成

邮件的格式最初在 RFC822 中提及^[3],目的是定义邮件系统之间(邮件客户代理(MUA, Mail User Agent)和邮件传输代理(MTA, Mail Transport Agent)之间或者 MTA 和 MTA 之间)传输的由 7bit(或 8bit)组成的字符流的含义。每封邮件都必须由邮件头部和邮件正文组成,头部和正文之间由一个空白行(CRLF)分割,这也是程序区别邮件头部和正文的唯一标志。

^{*})基金项目:微软亚洲研究院“长城计划”资助。王守芳 硕士研究生,研究方向为 Web 信息检索与挖掘,嵌入式系统。樊闻斌 硕士研究生,研究方向为软件工程、设计模式。金 浩 博士研究生,研究方向为人工智能,模式识别。潘金贵 教授,博士生导师,研究方向为多媒体信息处理、远程网络教育。

邮件的头部由称为“域”(field)的实体构成,其语法定义很松散,除部分包含结构化信息的域具有比较严格的语法定义外,其他域的语法要求主要是域以域名“:”域体的形式出现,即 field::=field-name“:”[field-body];各个域之间出现的次序没有限制,域包含了对于邮件传输系统(MUA 和 MTA)具有意义的信息,涵盖发送者身份、传递路径信息、邮件传输版本信息、邮件来源等各个方面的内容^[4]。

从语法分析的角度看,我们从信息是否具有特定的语法结构分类进行处理。地址信息、类型信息和表示信息的域体部分由更小的语法单位组成,对这些粒度更小的语法单位的分析和提取更有利于 XML 表示的结构化,称之为“结构化信息”,其余各类信息(跟踪信息、标识信息等)则不包含更小的语法结构,或者说,域体的字符串是一个不可再分解的语法单位,属于另一类,称之为“无结构化信息”。当然,这里的划分是从 XML 标准化方面考虑的。一个对邮件进行过滤的防火墙系统可能认为“Received”域的信息更具有分析的必要。

在很长一段时间内,正文部分是最为简单的一个部分,由一个或者多个包含属于 us-ascii 字符集的字符的行组成,既不需要解码操作,也不需要正文的结构进行分析,只需要取出第一个空白行之后的文本即可。但是随着 email 应用范围的扩大,为了传输声音、图像以及其他二进制数据,IETF 提出了多媒体互联网邮件扩展(Multipurpose Internet Mail Extensions, MIME, RFC2045^[5])。这样,邮件可以看作是由一个或者多个的 MIME 实体组成,这些实体之间还可以组合和嵌套。

2.2 头部信息的处理

近年来在这个方面已经有了很多的尝试,用 XML 表示邮件的研究尚处于起步阶段。目前可借鉴的成果有 Digital Preservation Testbed 组织的研究^[6],在其中提到了两种方法:All-In-One 方法和 Split-Files 方法。前者将传输文件直接转化为 XML 文件,头部的每个域的域名和域体被分别标注,域名的元素标记名是“name”,域体的元素标注名是“value”,邮件内容和附件保留原有的传输格式,直接表示成 CDATA 的形式,这种方法类似于“XML Wrapper”方法。而 Split-Files 方法,传输文件被分成几个部分,域名作为 XML 文档的元素标注名,域体作为 XML 文档的元素标注值,邮件内容保存为它们的原始格式,通过在域的名字和 XML 文档的元素标注名之间建立简单的对应关系得到邮件的 XML 表示。事实上,域的名字和 XML 文档的元素的标注名之间有自然的对应关系。All-In-One 方法的缺点是 XML 文件冗长,查询不方便,但优点是扩展性好,格式统一;Split-Files 很灵活,域直接变成 XML 标记,这种方法适合于有着复杂内容的邮件,弥补了第一种方法的缺点,XML 文件长度相对较小且结构清晰,查询方便,可以根据域的名字直接读取同名标注的 XML 元素即可得到相应信息,但缺点是可扩展性差,格式不统一。这两种方法足以指导非结构化信息的表示,但对结构化信息的处理则不够,因而对于实际应用来说,还有很大的不足。

在结构化信息中包含很多常用的信息,尽可能地分离这些信息以方便读取,在本文的 XML 标准化方案中,我们克服了上述两种方法的缺点,对非结构化和结构化这两类信息转换为 XML 进行分别处理:

(1)对于非结构化域,如“Received”那么将域体部分直接作为元素的内容。

(2)对于结构化域,类型信息和表示信息都以“Content”开头,我们采用以下的步骤变换,域名作为元素的标注名,域体直接作为元素的内容,如果域体还包含属性,那么属性的名字作为元素的属性名,域体属性的值作为元素的属性值,例如:

Content-Type: text/plain; charset="gb2312"被表示成
<Content-Type charset="gb2312">text/plain</Content-Type>。

但是这些信息和邮件主体的内容相关,我们把它们放入邮件体的 XML 标准化中,并未放入邮件头的标准化中。

(3)地址信息是比较特殊的一类,语法结构最为复杂,有以下几种:

- somebody@domain.com, 这是最基本的邮件地址格式;

- Some Body<somebody@domain.com>, 其中“Some body”在 RFC822 中被称为 phrase, MUA 常常显示邮件地址中的 phrase 部分,而不是“<”中的 email 地址平;

- somebody@domain.com (Some Body), 括号引进的字符串被称为 comment, 它可以包含任意的说明信息。

对于以上三种情况,我们将地址部分直接作为 XML 文档的元素的内容,而 phrase 或 comment 部分应该是 email 地址的补充,而不是邮件传输中起决定作用的元素。因此将它作为元素的属性更适当和简洁。例如:

From: “wsf”<wsf@mes.nju.edu.cn>则对应地转换为
<From name=“wsf”>wsf@mes.nju.edu.cn</From>

对于统一的消息模式来说,一般都具有收件人、发件人、时间、消息长度、主题这些信息,而邮件和其他类型的消息一般都会有自己的特别的一些头域,比如邮件可能有 Message-ID、优先级等信息,相对短消息头部信息相对少些。考虑到可扩展性,将各种格式的消息中都有的域列出,其余的则插入 Additional 标记中,这样就克服了文[6]中 Split-Files 的缺点,EBNF^[7]表示邮件头如表 1。

表 1 邮件头的 EBNF 表示法

```
mailHeader="<Header">headerInfo</Header">
headerInfo=from tolist date subject content-Length additional
from="<From">[attributeInfo]">Value</From">
to="<To">[attributeInfo]">Value</To">
tolist=to{to}
date="<Date">Value</Date">
content-Length="<Content-Length">Digital</Content-Length">
additional={("<" tagName{attributeInfo}">Value</tagName">)}
tagName=["Received"|"Message-ID"|"Cc"|"MIME-Version"|"
Message-ID"|extension-field|user-defined-field]
attributeInfo="name="Value
extension-field=( * Any field which is defined in a document published as a formal extension to RFC822; none will have names beginning with the string "X-" * )
user-defined-field=( * Any field which has not been defined in RFC822 or published as an extension to RFC822; names for such fields must be unique and may be pre-empted by published extensions * )
```

对于头部信息进行语法分析转换得到的 XML 文档给应用程序带来了很大的方便,应用程序只要根据域名即可得到相应的信息,原来混杂在域体中的属性可以直接作为元素的属性存取。

2.3 正文部分的处理

在 RFC2045 中提到 MIME 的概念,它是当前广泛应用的一种电子邮件规范, MIME 实体实际上成为组成邮件的基础,从最简单的邮件到复杂的邮件,其邮件组成单位都是

MIME 实体。MIME 实体可以看作是一个由 Content-Type 域、Content-Transfer-Encoding 域以及其他可选域组成头部,任意多行以 CRLF 结束的文本组成内容的语法单位。

邮件头相当于传统邮件中的信封,和邮件体是逻辑上独立的,这样就要求邮件体的处理能够和整个邮件头无关,而邮件头中的 content-type 域表明了 MIME 内容类型以及编码信息等,Content-transfer-Encoding 说明了 MIME 内容的传输编码格式,所以在邮件头的 XML 化中,我们没有把这些类型信息和表示信息直接放入邮件头的 XML 文件中,将邮件体的 XML 设计成:整个邮件正文 Body 由一个 MIME 组成,MIME 由 Info 和 Body 组成,Info 中放的是这个 MIME 部分的类型信息和表示信息,如 Content-Type 信息等,Body 由若干个 MIME 或纯文本组成,这样就形成了一个嵌套定义。上述的邮件 XML 格式有利于分开处理,因为邮件正文的处理(编码、解码等)除了和 Content-type、Content-Transfer-Encoding 这些有关,与信封信息无关,所以将 content-type 等信息放在体中,虽然 Content-Type 这些信息在原始邮件中是显示在头部中的。将 Content-Type、Content-Transfer-Encoding 常用的信息列出,而将其余的插入 Additional 标记中,这样的格式方便扩充,使用 EBNF 定义邮件体如表 2 所示。

表 2 邮件体的 EBNF 表示法

```
mailBody="<Body>" mime "<Body>"
mime="info body
info="<Info>" headerInfo "</Info>"
body={mime}|Value
headerInfo=content-Type content-Transfer-Encoding additional
content-Type="<Content-Type>{attributeInfo}" Value "</Content-Type>"
content-Transfer-Encoding="<Content-Transfer-Encoding>" value
"</Content-Transfer-Encoding>"
additional="{<tag>{attributeInfo}</tag> value "</tag>{attributeInfo}</tag>" value "</tag>{attributeInfo}</tag>"
attributeInfo=attributeName "=" value
tag="Content-ID" | "Content-Disposition" | "Content-Description" | "Content-location"
attributeName="boundary" | "name" | "charset" | "type" | "filename"
```

表 3 基本邮件及其转换后的 XML 文件

原邮件
Date: Fri, 8 Oct 2004 09:33:48+0800 From: "wangshoufang" (wsf@mes.nju.edu.cn) To: "tester" (tester@mes.nju.edu.cn) Subject: test X-mailer: Foxmail 5.0[cn] Mime-Version: 1.0 Content-Type: text/plain; charset="iso-8859-1" Content-Transfer-Encoding: 7bit This is a test mail
转换后的 XML 文件
<Message> <Header> <From name="wangshoufang">wsf@mes.nju.edu.cn</From> <To name="tester">tester@mes.nju.edu.cn</To> <Date>Fri, 8 Oct 2004 09:33:48+0800</Date> <Subject>test</Subject> <Content-Length></Content-Length> <Additional><X-mailer>Foxmail 5.0[cn]</X-mailer> <Mime-Version>1.0</Mime-Version> </Additional> </Header> <Body>(MIME) <INFO> <Content-Type charset="iso-8859-1">text/plain</Content-Type> <Content-Transfer-Encoding>7bit</Content-Transfer-Encoding> </INFO><Body>this is a test mail</Body></MIME> </Body> </Message>

个 MIME 实体,但是形式仍然和复杂的邮件是统一的。邮件可以在 MIME 头部附加上一些域“From”,“To”,“Subject”等,下面的例子中斜体部分表示了附加在 MIME 实体的域。在附加了这些域之后仍然是一个有效的 MIME 实体,因为按照 RFC2045 的规定,不以“Content-”开头的域出现在 MIME 实体中可以忽略,对斜体的处理和前面我们介绍的邮件头部的处理方法完全一样,原邮件和转换后的 XML 文件见表 3。

2.3.2 复合的 MIME 实体 Content-Type 中的 type (主类型)为 multipart 的 MIME 实体,称之为复合实体,它由 MIME 实体的嵌套组成,是附件和多格式邮件的基础,email 藉此传递多种信息,同时,这也是对 email 分析的最大难点。越来越多的增强功能不可避免地导致格式种类繁多,语法也不甚严格。邮件处理软件对邮件正文的分析大量依赖于字符串的顺序搜索,效率十分低下,按照 content-Type 的主类型来分,最常用的邮件的格式有以下几种:

- ① 没有使用 multipart 扩展的邮件,一般具有类型 text/plain 或者 text/html 或者 text/xml;
- ② 夹带附件的邮件,具有类型 multipart/mixed;
- ③ 包含相同的信息不同展现形式的邮件,具有类型 multipart/alternative;
- ④ 存在内嵌资源的,具有类型 multipart/related;

可以看出,如果在邮件中要添加附件,必须定义 multipart/mixed 段;如果存在内嵌资源,至少要定义 multipart/related 段;如果纯文本与超文本共存,至少要定义 multipart/alternative 段。“至少”的意思是:如果只有纯文本与超文本正文,那么在邮件头中将类型扩大化,定义为 multipart/related,甚至 multipart/mixed,都是允许的^[8]。内嵌资源是 MIME 的一个亮点,它能使邮件内容变得生动活泼。可在邮件的 multipart/related 框架内定义一些与正文关联的图片。动画等,通常是在 HTML 正文中,使得超级链接与内嵌资源相联系。

复合实体有多种子类型,我们以 mixed 为例介绍,假设有一个最简单的 mixed 子类型 MIME 实体,它包含一行“This is an example”文本和一个包含一个同样内容的文本附件。为了节省篇幅,我们只给出邮件正文和正文转换后的 XML 文件,邮件头部处理和前面介绍的处理方法一致。对于下表左边的邮件正文,我们对应的 XML 文档使用元素的嵌套来表示,假设按照前文所述的方法将外层 MIME 实体称为 A,内层两个实体分别称为 B 和 C,那么 B 和 C 成为 A 的 Body 元素的子元素,这种结构非常清晰。参见表 4。

2.4 性能分析

邮件表示成 XML 文档之后,一个不可避免的结果是邮件的尺寸增加,标注给文档带来结构化特性的同时,也带来了信息冗余,因为 XML 文档的标注必须成对出现,为了对邮件大小的增加有一个准确的了解,包括日常生活中接收的邮件以及科研中与他人联系的邮件进行了分析,实际转换结果的对比参见表 5。

可以看出,按照我们的转换方法邮件的大小变化并不明显,幅度一般在 5% 之内,转换后的大小基本上和转换前相当。可以看出,随着邮件的增大,增长越来越不明显。增长的主要原因是标注,但是在 XML 文件中我们消除了原来邮件中一些空格、注释等,两者起了一些抵消作用,增量并不明显,因此还是可以接受的。

表 4 原邮件正文以及正文转换的 XML 文件

原邮件正文	正文转换的 XML 文件
<pre>Content-Type: multipart/mixed; boundary="example-boundary" -- example-boundary Content-Type: text/plain; charset="UTF-8" Content-Transfer-Encoding: 7bit This is an example -- example-boundary Content-Type: application/octet-stream; name="demo.txt" Content-Transfer-Encoding: 7bit This is an example -- example-boundary--</pre>	<pre><Body><MIME><INFO> <Content-Type boundary="example-boundary">multipart/mixed</Content-Type> <Content-Transfer-Encoding></Content-Transfer-Encoding> </INFO><Body> <MIME><INFO> <Content-Type charset="UTF-8"> text/plain </Content-Type> <Content-Transfer-Encoding>7bit</Content-Transfer-Encoding> </INFO><Body>This is an example</Body></MIME> <MIME><INFO> <Content-Type name="demo.txt">application/octet-stream</Content-Type> <Content-Transfer-Encoding>7bit</Content-Transfer-Encoding> </INFO><Body>This is an example</Body></MIME> </Body></MIME></Body></pre>

表 5 邮件转换前后的大小比较

原始数据集	邮件数量	原始 总字节数	转换后 字节数	增加量/%
singlepart 邮件 (接收)	100	4569764	4853085	6.20%
singlepart 邮件 (发送)	100	4298436	4665952	8.55%
multipart 邮件 (接收)	100	5076304	5203408	2.5%
multipart 邮件 (发送)	100	5325788	5472012	2.7%

3 应用举例

转换后的 XML 文件是结构化的文档,易于被计算机处理,从我们的邮件的 XML 标准化方法来看,我们分别详细对邮件头和邮件体进行了 XML 转换,而头和体的处理充分考虑了独立性,力求在进行体(头)的处理时不需要查询头(体)的信息,由于无线设备容量和网络速度收到限制,这种方法能适用于无线设备上的邮件客户端,将邮件表示成 XML 文件,根据用户设定的不受欢迎主题和邮件地址等,自动将其删除邮件或拒收邮件内容。如果附件很大,使用转换化的 XML 文件可以方便地只查看邮件正文中的文本信息。甚至可以通过转换后的 XML 文件调用各种应用程序,比如将邮件附件转发到最近的传真机。这样当用户出差在外,他们只想收看那些紧急的邮件,而不是所有的冗长而又复杂的邮件,用户使用移动电话阅读小数据量的邮件,大而复杂的邮件就转发到最近的传真机,然后在纸上处理。

结束语 XML 技术的应用范围越来越广泛,各种应用对它的支持也在逐步加强,目前,我们实现了使用协议将原始邮件转换为 XML 文件,可以弥补原始邮件格式的随意性的缺点,如果在简单邮件传输协议(Simple Mail Transfer Protocol, SMTP^[9])应用环境中 XML 成为传输 email 的基本格式,

MUA/MTA 可以节省很多分析工作。我们在邮件标准化时,不丢弃任何信息,以便信息能够重新构造和查询,这样就可以更大程度地适合于其他应用,比如转化成其他格式的消息进行转发等。在未来的情况下,可以设想,人们可以通过电子邮件发送命令控制各种终端,比如要求打印,这些程序通过终端设备将接收到的邮件转交给被访问的应用程序进行处理。使用 XML 统一表示各种信息,可以使不同媒介的信息能相互转换,用户可以随时随地通过任意一种终端,比如智能电话、计算机、PDA 来收发任何消息,这样就方便了人们的工作和生活。

参 考 文 献

- 1 <http://www.microsoft.com/china/solutions/calling/serve/top-call.asp>
- 2 Walsh N. A technical introduction to XML. Ann Arbor, MI: ArborText, Inc, 1998. <http://www.xml.com/pub/a/98/10/guide0.html>
- 3 Network. Standard for the format of ARPA Internet text messages, RFC822. Department of Electrical Engineering, University of Delaware, 1982
- 4 Palme J. Common Internet Message Headers, RFC2076. Stockholm University/KTH, 1997
- 5 Freed N, Borenstein N. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, RFC2045. Internet Engineering Task Force, 1996
- 6 Potter M. XML For Digital Preservation: XML Implementation Options for E-Mails. Digital Preservation Testbed, 2002. <http://www.digitaleduurzaamheid.nl/bibliotheek/docs/email-xml-imp.pdf>
- 7 Information technology-Syntactic metalanguage-Extended BNF. 1996. ISO/IEC 14977:1996(E)
- 8 <http://blog.csdn.net/bhw98/archive/2003/05/20/19670.aspx>
- 9 Postel J B. Simple Mail Transfer Protocol, RFC821. Aug. 1982