●王兰成 冯文杰

两种 CNMARC 的 XML DTD 信息描述机制的研究与比较*

摘 要 提出和设计了基于 CNMARC 书目资源和基于 CNMARC 资源框架的两种 XML DTD 解决方案,从而使 MARC 规范格式的数据转换成通用机器可理解的元数据成为可能。这两种方案对于现有 CNMARC 数据在数字图书馆中的利用有现实意义。参考文献 3。

关键词 MARC XML 元数据 数字图书馆
分类号 G250.76

ABSTRACT The authors propose two XML DTD solutions based on CNMARC bibliographical resources and CNMARC resource framework, thus making it possible to convert MARC standard formats to machine intelligible metadata. The two solutions have practical significance for the application of present CNMARC data in digital libraries. 3 refs.

KEY WORDS MARC. XML. Metadata. Digital library.
CLASS NUMBER G250.76

随着网络技术和信息需求的迅速发展,MARC 存在的数据可读性差、不利于理解和数据的管理、显示与检索依赖于特定软件平台的局限性日益突出。比较研究基于 XML 的 CNMARC 信息描述机制已成为当前研究的热点。

1 基于 XMARC 元数据的信息描述技术

1.1 MARC 的局限与 XMARC 元数据的提出

国际图联近年来对 UNIMARC 所作的补充和修改,很大程度上解决了目前应用中的矛盾。然而随着实践的发展,其局限性并没有从根本上完全解决。首先,MARC 格式的复杂性、不易理解性造成了它在信息显示方面的障碍,许多基于 MARC 的系统都有相应的显示程序,但要求编制程序员是一个 MARC 专家,由一个系统满足不同 MARC 信息的显示形式及详略程度需求是不可能也是没必要的。其次,MARC 系统的大部分应用仍然局限在对书目信息的描述,著录效率无法适应海量的网络资源著录需求,对 MARC 信息的处理极度依赖于特定的软件,离开了专用软件很难对 MARC 进行操作。再次,由于 MARC 格式是经过严格限定和检验的数据流格式,目前只有专用的符合 MARC 格式的应用前端才能对 MARC 数据加以利用,这也是大量网上资源目前无法通过通用的搜索引擎向因特网用户提供链接的主要原因。

面向 WWW 的 MARC 信息需求已成为图书馆系统发展的重要课题。XMARC 是本文提出的 CNMARC 资源描述框架,它是一种用 CNMARC 元数据来描述资源并由 XML 扩展定义的信息描述机制。其概念内涵包括:用 XML 语法来阐述 CNMARC,目前规定 XML 文档的逻辑结构通过 DTD(可转换为 XML Schema)方法来实现,有多种 DTD 设计方案来支持 CNMARC 全集,所有元素可自定义但具有核心元素[1]。MARC 格式使用规定,一条书目记录包含 n 个字段,一个字段可包含 0 到 n 个子字段,不同字段和子字段具有不同含义,不同字段和子字段根据格式具有不同的重复属性和必备属性,字段具有不同含义的指示符等等[2],这些规则都应该通过 DTD 表现出来,而 XMARC 的 DTD 恰恰可以表现这些规则,并可以检验 MARC 语法的正确性。

1.2 XML 及其文档类型定义

CNMARC 的优势决定了它在中国图书馆界不可被替代的地位,其不足又在一定程度上制约了它的进一步发展,新兴的 XML 技术可以很好地解决这个问题。XML 具有客户定制标记词表,使得描述数据以及数据片之间的关系成为可能,如以 XML 作为统一的数据交换格式,将极大地方便不同系统之间的信息传输。XML 文档中须包含 DTD,才能识别出其中的标记所代表的意义。DTD有外部和内部之分,

* 本文得到上海市科委课题(项目编号 02JG05031)的支持。

外部 DTD 独立于 XML 文档并可被多个 XML 文档共用,但使用该 DTD 的文件中必须指明 DTD 的位置,内部 DTD 是内嵌于 XML 文件中的 DTD,只能被该 XML 文档使用。

DTD 的语法包括定义元素及其后代和定义元素属性。定义元素及其后代即元素类型声明、定义元素及其子元素和父元素与子元素之间关系,DTD 通过元素之间的父子关系,描述了整个文件的结构关系,元素定义由它们的元素内容模型来描述,即由紧跟元素后面的括号中内容(子元素)来定义,DTD 尽管要求严格,但有它的灵活性,它使用正则表达式来描述父元素与子元素之间非常复杂的关系[3]。上述采用的结构分别是:

<! ELEMENT　元素名　元素内容描述>

<! ELEMENT　元素名(子元素名)>

DTD 中定义元素属性使用的格式:

<! ATTLIST　元素名　(属性名　属性类型　缺省值)* >

2　基于 CNMARC 书目资源的 XMARC 信息描述

我们在研究设计基于 CNMARC 书目资源的 XMARC DTD1 时,解决了以下关键问题。

(1)对(子)字段说明及字段和子字段之间关系的处理。把字段说明作为 DTD 中一个文档元素的名字,同时把该字段所包含的子字段说明作为该元素所包含的子元素名字。定义语法如下:<! ELEMENT 字段说明　(子字段 1 的字段说明(元字符),子字段 2 的字段说明(元字符),……)>;不包含子字段的字段其相应的元素定义是:<! ELEMENT 字段说明(#PCDATA)>。

(2)对(子)字段标识符的处理。利用 DTD 中文档元素的属性来表现(子)字段的标识符。字段或子字段标识符用定义该字段或子字段的文档元素的属性来表示,属性名为 tag 或 subtag。如 005 字段的 $a 子字段"ISBN",其定义是:<! ELEMENT ISBN(#PCDATA)>,<! ATTLIST ISBN subtag CDATA #FIXED"a">。

(3)对字段指示符的处理。字段指示符用定义该字段的文档元素属性来表示,指示符 1 的属性名为 indicator1,指示符 2 的属性名为 indicator2。如 101 字段"作品语种"的元素定义:<! ELEMENT 作品语种(正文声道等语种*,中间语种*,原作语种*,提要或文摘语种*,目次页语种*,与正文语种不同的题名页语种*,正题名语种?,歌词等的语种*,附件语种*,字幕语种*)>,<! ATTLIST 作品语种 tag CDATA #FIXED"101">,<! ATTLIST 作品语种 indicatorl(0|1|2)#REQUIRED>,<! ATTLIST 作品语种 indicator2 CDATA #FIXED"　">。对于没有设指示符的字段,其相应的文档元素不具有 indicator1、indicator2 属性。

(4)对(子)字段必备属性和可重复属性的处理。在元素定义时通过元字符来体现字段和子字段的这些属性。它们之间不同的组合结果与元字符之间的关系如下:{(必备/可重复+),(必备/不可重复),(可选择/可重复*),(可选择/不可重复?)}。字段和子字段的必备与可重复属性分别表现在 MARC 的元素定义语句和所属字段所对应的文档元素定义语句中,如<! ELEMENT MARC(头标区,记录标识号,记录处理时间标识?,国际标准书号*,……,家族名称-次要知识责任*,记录来源+,ISDS 中心?,馆藏信息*)>。

(5)对头标区的处理。MARC 的头标区用一文档元素来表示,该元素包含于元素 MARC 中,它是必备且不可重复的该元素的内容为字符数据。<! ELEMENT 头标区(#PCDATA)>。

(6)根元素 MARCS 及其子元素 MARC。该 DTD 的根元素为 MARCS,包含一子元素 MARC。MARC 子元素是可重复的,其元字符为"*"。MARC 元素包含的子元素为头标区及 MARC 中包含的所有字段所对应的文档元素。根元素 MARCS 定义:<! ELEMENT MARCS(MARC)* >

下面是一条根据该 DTD1 定义所制定的 XMARC 记录:

<? xml version="1.0"encoding="gb2312"? >

<marcs>

<marc>

<头标区>00691nam0 2200217　45 0</头标区>

<记录控制号 tag="001">6010290088</记录控制号>(记录标识号?)

<国际标准书号 tag="010"indereatorl="　"indereator2="　">

<ISBN subtag="a">1007-0052</ISBN>

<获得方式-定价 subtag="d">¥198.00</获得方式-定价>

</国际标准书号>

<一般处理数据 tag="100"indereator1="　"indereator2="　">

<一般处理数据 subtag="a">20020315d2001 kemy0chia0121 ea</一般处理数据>(空处用??? 符号显示?)

</一般处理数据>

<题名与责任说明 tag="200" indereator1="1" indereator2=" ">
<正题名 subtag="a">世界经济年鉴 1999/2000</正题名>
<正题名汉语拼音 subtag="A">Shi Jie Jing Ji Nian Jian 1999/2000</正题名汉语拼音>
<第一责任说明 subtag="f">世界经济年鉴编辑委员会</第一责任说明>
</题名与责任说明>
<记录来源 tag="801" indereatorl=" " indereator2="0">
<国家代码 subtag="a">CN</国家代码>
</记录来源>
<馆藏信息 tag="905" indereatorl=" " indereator2=" ">
<馆藏机构代码 subtag="a">601</馆藏机构代码>
<分类号 subtag="d">F11-54</分类号>
<书次-种次号 subtag="e">1/1999-2000</书次-种次号>
<复本数 subtag="f">1</复本数>
</馆藏信息>
</marc>
</marcs>

3 基于CNMARC资源框架的XMARC信息描述

我们研究设计的基于CNMARC框架的XMARC DTD2,处理并解决了以下的关键问题。

(1)对(子)字段说明及字段和子字段之间关系的处理。把字段说明作为文档元素"字段"的一个名为"字段说明"的属性,把子字段说明作为文档元素"子字段"的一个名为"子字段说明"的属性。字段和子字段之间的包含和被包含关系也通过父元素和子元素之间的关系来体现。元素"字段"可以包含子元素"子字段"或纯文本数据(如字段001等不包含子字段的字段)。子元素"子字段"可以在父元素"字段"中出现1到多次,这根据该字段的具体情况而定。

(2)对字段(子字段)标识符的处理。标识符处理与前面的DTD处理大体一致,不同的是属性的缺省值为"#REQUIRED"而非"#FIXED",含义为在XML文件中必须为这个属性给出一个属性值,但这个值不是固定的。

(3)对字段指示符的处理。字段指示符用元素"字段"的属性来表示,指示符1的属性名为indicator1,指示符2的属性名为indicator2。与前一种类型DTD不同的是,属性的缺省值为"#IMPLIED"而非"#FIXED",其含义是不强行要求在XML文件中给该属性赋值,这里这样定义的原因是MARC中有个别字段是不设指示符的(如001字段),在实际使用过程中除不设指示符的字段外都应该给该属性赋值。

(4)对头标区的处理方法同前一形式的DTD。

(5)根元素MARCS及其子元素MARC。这种形式DTD的根元素为MARCS,包含一子元素MARC。MARC子元素是可重复的,其元字符为"*"。MARC元素包含两个子元素:"头标区"和"字段"。

元素MARC的现实含义为一条书目信息。DTD2具体是:

```
<? xml version="1.0"
encoding="GB2312"? >
<! ELEMENT MARCS(MARC*)>
<! ELEMENT MARC(头标区,字段+)>
<! ELEMENT 头标区(#PCDATA)>
<! ELEMENT 字段(子字段*)>
<! ATTLIST 字段 字段说明 CDATA #REQUIRED>
<! ATTLIST 字段 tag CDATA #REQUIRED>
<! ATTLIST 字段 indicatorl CDATA #IMPLIED>
<! ATTLIST 字段 indicator2 CDATA #IMPLIED>
<! ELEMENT 子字段(#PCDATA)>
<! ATTLIST 子字段 子字段说明 CDATA #REQUIRED>
<! ATTLIST 子字段 subtag CDATA #REQUIRED>
```

下面是一条根据该DTD所制定的XMARC记录:

```
<? xml version="1.0"encoding="gb2312"? >
<marcs>
<marc>
<头标区>00691nam0 2200217 45 0</头标区>
<字段 字段说明="记录控制号" tag="001">6010290088</字段>
<字段 字段说明="一般处理数据" tag="100" indereator1=" " indereator2=" ">
<子字段 子字段说明="一般处理数据" subtag="a">20020315d2001 kemy0chia0121ea</子字段>
</字段>
<字段 字段说明="题名与责任说明" tag=
```

"200"indereator1="1"indereator2="　">
　　<子字段　子字段说明="正题名"subtag="a">世界经济年鉴 1999/2000</子字段>
　　<子字段　子字段说明="正题名汉语拼音"subtag="A">Shi Jie Jing Ji Nian Jian 1999/2000</子字段>
　　<子字段　子字段说明="并列正题名"subtag="d">The Yearbook of World Economy</子字段>
　　<子字段　子字段说明="第一责任说明"subtag="f">世界经济年鉴编辑委员会</子字段>
　　</字段>
　　<字段　字段说明="记录来源"tag="801"indereator1="　"indereator2="0">
　　<子字段　子字段说明="国家代码"subtag="a">CN</子字段>
　　</字段>
　　<字段　字段说明="馆藏信息"tag="905"indereator1="　"indereator2="　">
　　<子字段　子字段说明="馆藏机构代码"subtag="a">601</子字段>
　　<子字段　子字段说明="分类号"subtag="d">F11-54</子字段>
　　<子字段　子字段说明="书次-种次号"subtag="e">1/1999-2000</子字段>
　　<子字段　子字段说明="复本数"subtag="f">1</子字段>
　　</字段>
　　</marc>
　　</marc>
　　……
　　</marc>
　　<marc>
　　……
　　</marc>
　　</marcs>

4　两种类型 DTD 的研究比较

(1)对 MARC 内容的体现。DTD1 对 MARC 内容体现得十分完整,每个字段的含义是什么、对应的字段标识符是什么、它包含哪些子字段、各个字段子字段的必备属性、可重复属性是什么,等等,DTD1 中都定义得一清二楚,可以把 DTD1 看做是它对应的机读目录格式的缩略版。DTD2 则对机读目录格式进行了一定程度的抽象,对其内容进行了概括,把不同的字段统一用"字段"元素表示,不同的子字段统一用"子字段"元素表示,所以在定义时无法对每个具体字段的相应信息进行揭示,这也是 DTD2 中对"tag"、"subtag"、"indereator"等属性的缺省值无法定义为"#FIXED"的原因。

(2)数据的可理解性。在数据的可理解性上,DTD1 表现得更好,因为它把字段和子字段说明直接作为一个元素的元素名,这样人们只要看到这个标记就知道这个元素记录的内容是什么。而 DTD2 任何字段都用"字段"这个元素名来表示,只有通过元素属性"字段说明"的内容才能知道该元素的具体含义。

(3)程序处理的方便性。DTD1 把每个字段和子字段都定义为一个元素,并把相应的说明作为元素名,方便了用户理解,但却给计算机程序的处理带来了麻烦,同时对于计算机处理的时间存在很大浪费。而 DTD2 仅有 5 个元素,定义格式简单明了,字段(子字段)相应信息的不同都通过属性值的不同来区分,对数据进行处理时只需分析元素名就可以对字段和子字段进行识别,极其方便程序处理。

(4)开放性和兼容性。DTD1 根据 MARC 中规定的字段来确定所定义元素的数量并使用不同的标记,当格式标准发生变化时,DTD1 须作相应调整,且根据某一版本的 MARC 格式制定的 DTD 只能应用于该版本。DTD2 对 MARC 中的字段进行了抽象概括,分析出各字段的共性,这不仅使它可以适应格式标准修订等发生的变化,更重要的是对于其他领域的 MARC 格式(如档案机读目录格式 CNMARC AMC),同样是兼容的。

目前,我们已设计实现了一个基于 XMARC 的书目信息管理系统,并研究实现了 CNMARC 到 XMARC 的数据转换课题。XMARC 是一种定义了用 CNMARC 元数据来描述网络目录资源,在继承 MARC 优点基础上弥补了 MARC 的局限,具有深入研究和应用开发的良好前景。

参考文献

1　胡小菁.书目记录等级与核心记录标准的发展.中国图书馆学报,2003(2)
2　北京图书馆自动化发展部.中国机读目录通用格式.北京:书目文献出版社,1991
3　黄伟红,张福炎.基于 XML/RDF 的 MARC 元数据描述技术.情报学报,2000(4)

王兰成　解放军南京政治学院上海分院信息管理系教授,东华大学信息学院在职博士生。通讯地址:上海市。邮编 200433。

冯文杰　解放军南京政治学院上海分院信息管理系硕士研究生。通讯地址同上。　(来稿时间:2003-05-12)