《当代语言学》(试刊) 1998 年第 4 期 1 - 11 页 ,北京

标准通用置标语言 SGML 及其在自然语言处理中的应用

冯志伟　国家语言文字工作委员会

提要　本文根据国际标准 ISO 8879 —1986 和我国国家标准 GB 14814 ,介绍标准通用置标语言的原则和方法 ,叙述了从通用置标语言 GML 到标准通用置标语言 SGML 的发展过程 ,详细说明了标准通用置标语言的三个组成部分(SGML 声明、文件类型定义、文件实例) 的具体内容。在标准通用置标语言的基础上 ,又进一步根据国际标准 ISO DIS 12200 ,介绍了机器可读术语数据交换格式(MARTIF) 的数据类目和文件的基本结构 ,以术语工作为例子说明了 SGML 在自然语言计算机处理中的应用价值。

关键词　标准通用置标语言 ,SGML 声明 ,文件类型定义 ,文件实例 ,标记 ,实体引用 ,术语数据交换格式 ,数据类目。

文本语料库和术语数据库是当前自然语言处理(natural language processing ,简称 NLP) 的重要内容 ,特别在语料库语言学兴起之后 ,这方面的研究特别活跃。在文本语料库和术语数据库的研制和应用中 ,如何交流库与库之间的信息资源 ,是一个极为重要而迫切的问题。为了实现文本语料库之间以及术语数据库之间的信息交换(information exchange) 和信息的再利用(information reuse) ,必须首先解决语言信息格式的标准化问题。国际标准化组织 ISO / TC37 / SC3 近年来一直力图把标准通用置标语言(Standard Generalized Markup Language) 应用于文本语料库和术语数据库的研制和开发之中 ,在最近召开的 ISO / TC37 / SC3 的会议上 ,几乎每一次会议都要讨论 SGML 在自然语言处理中的应用问题。本文作者曾经作为中国代表参加过 1993 年(在丹麦哥本哈根召开) 、1994 年(在挪威奥斯陆召开) 和 1995 年(在美国费城召开) 的标准化会议 ,在会议期间积极参与了有关问题的讨论 ,并参加了有关国际标准的投票 ,对于这些标准的情况和细节比较了解。因此 ,我们在本文中介绍一下 SGML 这种重要的标准通用置标语言以及它在文本语料库和术语数据库中的应用 ,希望这些介绍有助于我国语言信息处理中标准化工作的进一步开展。

为了帮助人们理解书面信息的内容 ,书面信息必须显现为便于人们阅读的形式 ,人们便常常在书面文本中加入一些控制标记 ,如文字的大小、标题的字体、插图的位置等。但是 ,每个系统都有自己的控制标记 ,不同的系统之间没有统一的格式 ,标记和格式都没有通用性 ;有的系统尽管也采用了所谓“过程性置标”(procedural markup) 的办法 ,说明在系统中 ,应该选用什么字体 ,标题应该处于版面上的什么位置等 ,但是 ,这种过程性置标无法使人们了解文件的结构 ,而且缺乏灵活性。例如 ,过程性置标告诉人们标题应排在一行的正中 ,但排在一行正中的不一定是标题 ,有时可能是插图或引用的某一内容 ;又如 ,当要使用另一个系统 ,或者使用同一个系统而要改变格式时 ,过程性置标必须逐一地改变以前所置的标记 ,重复处理的过程 ,费时而又费力。这种情况 ,严重地妨碍了信息的交换。因此 ,很有必要设计一种通用的标记 ,以提高信息的利用效率。

美国早在 60 年代就有学者开始研究这个问题。1969 年,Charles Goldfarb 在领导 IBM 公司关于集成法律办公信息系统的研究项目时,就同他的合作者 Edward Mosher 和 Raymond Lorie 共同设计了通用置标语言(Generalized Markup Language,简称 GML 语言)。1978 年,美国国家标准化局(ANSI)的下属单位成立了一个工作组,由 Goldfard 领导,进行文本描述语言标准化的研究,这项研究以 GML 语言为基础,进一步设计标准化的通用置标语言(Standard Generalized Markup Language,简称 SGML 语言)。这项研究得到了美国图形通信协会(Graphic Communication Association,简称 GCA)的支持,1980 年发表了 SGML 第一稿草案。1983 年,GCA 推荐 SGML 第六稿草案作为工业标准,被美国税务署和国防部采纳,1985 年 SGML 作为国际标准草案发布,1986 年,国际标准化组织正式发布了 SGML 国际标准,标准号是 ISO 8879-1986。我国于 1995 年也把 SGML 语言作为国家标准,标准号为 GB 14814。从研究通用的标记到 SGML 语言成为正式的国际标准,前后经过了近 20 年的时间。

SGML 语言应满足如下约束条件:

(1) SGML 语言所置标的文本必须能够被广泛使用的文本处理系统和文字处理系统所接受;

(2) SGML 语言能够支持现有的大容量文本输入设备;

(3) SGML 语言不依赖于任何的字符集,可以允许在不同的设备上输入文本;

(4) SGML 语言独立于具体的处理程序、系统、设备;

(5) SGML 语言没有任何的民族语言倾向;

(6) SGML 语言应该适合于人们所熟悉的打印机和文字处理程序的习惯;

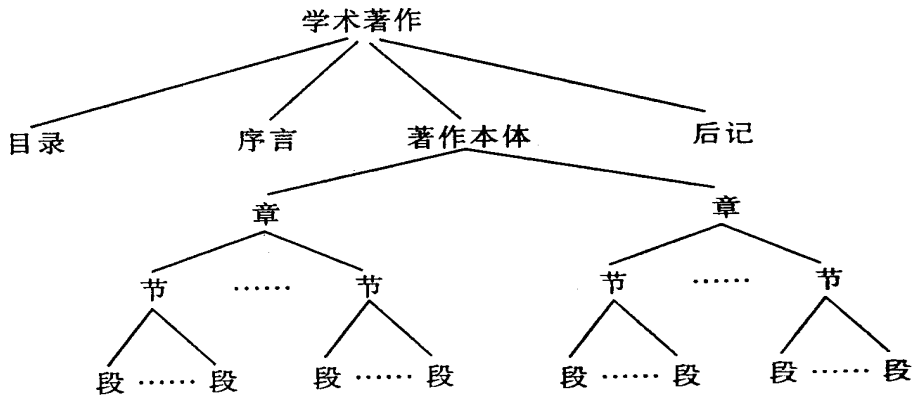
(7) SGML 语言置标后的文本应该能够与其它数据并存;

(8) SGML 语言应该既可以为计算机所使用,也可以为人所使用。

总而言之,SGML 语言的设计目的是要使文件信息与设备无关,与处理系统无关,甚至与所用的语种无关。SGML 语言的目的,就是要在各个孤立的系统之间架起桥梁,使各个孤立的系统彼此联系起来。

SGML 语言通过描述文件逻辑结构的方法,使置标具有通用性,并通过一系列的声明(declaration),使各个系统都能理解文件的信息与置标。

任何文件都有其自身的结构。例如,书信由发信人、收信人、信件本体等部分组成,公文由发文单位、收文单位、题目、文号、公文本体等部分组成,学术著作由目录、序言、学术著作本体、后记等部分组成,等等。学术著作本体又可由若干章组成,每一章又可以由若干节组成,每个节又可以由若干段组成。学术著作的结构可以用下面的树形图来表示:



书信和公文的结构也可以用类似的树形图来表示。SGML 语言在逻辑上将文件表示为树

形图结构。树形图上的每一个非终极结点叫做“元素”,一个元素的子结点,叫做这个元素的“内容”。例如,在学术著作的树形图结构中,“章”是“著作本体”的内容,“节”是“章”的内容,“段”是“节”的内容。树形图上的每一个终极结点没有内容,这些没有内容的终极结点,叫做“数据”。

SGML 语言用描述性置标来区分文件信息中的不同元素,用“文件类型定义”(Document Type Definition,简称 DTD)来描述某一类文件的结构,说明在这一类文件中有哪些元素,这些元素之间存在什么样的关系,SGML 语言还用 SGML 声明(SGML declaration)来说明所用的字符集、置标用字符、语法规则等。

总起来说,SGML 的文件应该包括三个部分:SGML 声明、文件类型定义、文件实例。下面我们分别对这三个部分加以说明。

1. SGML 声明(SGML declaration)

SGML 声明用来定义字符信息、具体语法规则、容量要求以及选用 SGML 的哪些特性。

(1)字符信息:说明置标用的字符。例如,重新定义用以区分文件内容与置标的定界符,定义功能字符,更改保留名,关键字等。SGML 允许指定正文所用的字符集,包括字符表及其对应的编码。SGML 还允许使用多八位编码字符集,这就可以在不同语种上使用 SGML 语言并交换文件。

(2)具体语法规则:允许用户按照自己的习惯定义具体语法规则,确定命名的规则,说明哪些字符可用于元素或其他成分的名字及名字开始字等等,改变 SGML 语言中推荐的语法规则。

(3)容量要求:说明各种名字的长度、标记的长度等。如果与 SGML 语言中推荐的容量不一致,用户可自行定义。

(4)可选特性:SGML 语言中有许多可选特性,例如,省略标记特性、链接类型特性等等,这些可选特性增强了 SGML 语言的适应性和灵活性。SGML 声明中,应该说明是否选用这些可选特性,并说明具体选用哪种类型的可选特性。

SGML 语言的核心是抽象语法。抽象语法定义了通用标识符、属性、实体引用等指标的使用方法。对于定界符字符、字符集、关键字、保留名、容量值等的规定,是 SGML 语言中作为向用户推荐的“基准具体语法”(reference concrete syntax)决定的。用户可以根据自己系统的环境、自己的民族语言、自己的键盘特点,使用 SGML 声明的内容来定义自己的具体语法。在使用不同的具体语法的系统之间交换信息时,SGML 声明是必不可少的,它必须出现在 SGML 文件的最前面。

2. 文件类型定义(DTD)

DTD 定义文件的结构和在文件信息中的置标规则。在 DTD 中,要遵照 SGML 声明中规定的字符信息和具体语法规则,对一类文件的结构用一组置标声明(markup declaration)作严格的定义。

在 DTD 中所用的置标声明主要有如下几种:

(1)元素声明(element declaration)

由于 SGML 语言在逻辑上把文件组织为由元素组成的树形图结构,元素是文件信息中的主要成分。元素声明用于说明每种元素的结构,其内容包括元素类型名及其内容结构,一般格式为:

〈! ELEMENT 元素类型名(内容模型)〉

ELEMENT 是元素声明的关键字,内容模型中说明该元素内容中允许出现的子元素,并

说明这些子元素以哪种关系和方式出现,例如,说明这些子元素是不是顺序的,是不是可选的,是不是重复的,等等。

元素声明中,还可说明在文件中置标时,是否可以省略文件的开始标记和结束标记。

(2)属性定义表声明(attribute definition list declaration)

元素具有属性。例如,"插图"这个元素一般具有"尺寸"和"位置"两个属性。由于元素可能具有一个以上的属性,所以,SGML 语言中采用属性定义表来定义元素的各种属性。

属性定义表由各属性定义组成。属性定义包括元素类型名、属性名、属性可取的值、缺省值等。例如,"插图"这个元素的属性定义表包括:元素类型名"插图",属性名"尺寸、位置","尺寸"这个属性名可取的值 NUMBERS,以及缺省值等。其一般格式为:

〈! ATTLIST 元素类型名 属性名 属性值 缺省值〉

ATTLIST 是属性定义表声明中的关键字。

(3)实体声明(entity declaration)

SGML 语言除了在逻辑上把文件组织为元素结构之外,它还在物理上把文件组织为实体结构,并采用实体声明把实体内容与实体名对应起来。

实体声明可以在声明中直接写出实体的内容,其格式为:

〈! ENTITY 实体名 实体正文〉

实体声明也可以不直接定出实体内容,而用实体标识符表示,其格式为:

〈! ENTITY 实体名 SYSTEM 实体标识符〉

ENTITY 是实体声明中的关键字,另一个关键字 SYSTEM 说明此实体正文在系统中,实体标识符用来说明实体正文存在什么地方。

实体声明还可以用来定义不能直接键入的字符和某些常用的特殊字符,如数学符号、希腊字母、法语和德语中的特殊字符、汉语拼音中的特殊字符等。例如,

〈! ENTITY uuml SDATA"[uuml]"= small u,dieresis or umlaut mark - - 〉

这个实体声明定义了德语和汉语拼音中所使用的特殊字符 ü。

(4)其它声明

SGML 语言允许文件的某些部分使用其它比较流行的语言置标。例如,某段复杂的公式可以用 Tex 语言置标,而不是按照 SGML 定义置标。文件中还允许夹用某些非 SGML 语言的字符。这些情况,采用"记法声明"(notation declaration)来说明。此外,SGML 语言中还有"引用声明"(short declaration)、"注释声明"(comment declaration)等。

上面我们说明了文件类型定义 DTD 中所用的置标声明,即元素声明、属性定义表声明、实体声明以及其它声明。文件类型定义 DTD 本身是放在文件类型声明(document type declaration)中的。文件类型声明的格式为:

〈! DOCTYPE 文件类型名 [DTD]〉

DOCTYPE 是文件类型声明的关键字,在方括号内,可以是文件类型定义 DTD 的内容,它们是由上述几类置标声明组成的若干语句,它们也可以是 DTD 的标识符。如果某类文件的 DTD 已经形成共识,成为公共的 DTD,就可以保存在外部实体中,使用时,只要在文件类型声明中直接引用该实体就行了。例如,

〈! DOCTYPE Memo SYSTEM"MEMO.DTD"〉

其中的 SYSTEM 说明文件类型 Memo 的 DTD 在系统中,存于外部实体 MEMO.DTD 中。这种 DTD 成为外部 DTD,直接写在方括号[ ]内的 DTD 是内部 DTD。如果外部 DTD 不完全适

用,允许在外部 DTD 之后,用方括号加入置标声明,以修正或补充已有的 DTD。

3. 文件实例(document instance)

文件实例包含文件信息的正文和置标,它一般由 SGML 声明中允许使用的字符组成,其中夹入的置标遵照这类文件的文件类型定义 DTD 进行。常用的置标有:

(1)标记(tag)

标记用于元素的置标,这是描述性置标。我们使用标记在正文中标出元素的层次结构,它由开始标记(start-tag)和结束标记(end-tag)组成。在开始标记中说明元素类型名(如"标题元素")及其属性(如标题的"位置"),结束标记中说明元素类型名,表示是哪一类元素的结束,因为元素可能有嵌套,在结束标记中指出元素的类别是非常必要的。在开始标记和结束标记之间,是元素的内容。如果省略元素的标记,在元素的置标声明中应该说明。在文件中置标时,应遵照说明进行。标记应该用特殊的符号括起来,这种符号叫做定界符(delimiter),以便把标记同文件的正文和置标区分开来,定界符一般用尖括号◇表示。SGML 语言的基准具体语法中推荐了各类定界符,如果要改变这些定界符,可以在 SGML 声明中重新定义。

例如,在术语交换格式中,用 SGML 语言的标记描述了如下的"文本内容",请注意到其中使用了定界符◇,定界符中所括起来的内容就是元素的标记:

```
〈DOCTYPE tei.2 …〉
〈tei.2〉
    〈tei.header〉
      …
    〈/tei.header〉
    〈text〉
        〈front〉
          …
        〈/front〉
        〈boby〉
            〈termEntry〉
              …
            〈/termEntry〉
            〈termEntry〉
              …
            〈/termEntry〉
              ……
        〈/boby〉
    〈/text〉
〈/tei.2〉
```

这个文本的名字叫做〈tei.2〉,包括文本标题〈tei.header〉和文本〈text〉两部分。文本的前文〈front〉主要是一些说明信息,文本的正文〈body〉中包括若干术语条〈termEntry〉相应部分的结尾都标以符号"/"。例如,文本的结尾标以〈/tei.2〉,文本前文的结尾标以〈/front〉,文本正文的结尾标以〈/body〉,文本中的术语条目结尾标以〈/termEntry〉,等等。采用这样的标记,文本的结构层次便十分清楚地表示出来了。

对元素的置标应该完全遵照文件类型定义 DTD 来进行。例如,如果我们在 DTD 中说明

了这样一类文件:它必须有一个名字,名字之后紧跟着一个标题,标题之后是文本,文本由前文和正文两部分组成,那么,我们在置标时,就必须遵照DTD规定的结构来进行。SGML语言的应用系统,应该对编辑的文件作语法分析,对于不符合DTD所规定的结构,应提出警告。

(2)实体引用(entity reference)

在文件类型定义DTD中用实体声明来定义实体,在文件中,用"实体引用"来引用已定义的实体,将该实体内容嵌入引用处并替换实体引用置标。

实体引用置标由引用打开符(一般为 &)、实体名、引用关闭符(一般为:)构成。

实体引用主要用于如下场合:

A.用名字引用一个很长的或者难以直接键入的字符,例如,DTD中有置标声明:

〈! ENTITY AAP"Association of American Publishers"〉

在文件中如果要用到Association of American Publishers,就可以引用AAP实体来替换它。

B.分开存放的文件,可以用实体引用把它们放在一起,把它们组织为一个实体。

近年来,SGML语言被广泛地应用于出版业中,用DTD来表示书籍、刊物、论文的结构,通过SGML语言置标的出版物,可以被多次地使用,并且便于检索,便于存储,便于通过电子的方式发送,大大地促进了电子出版事业的发展。

国际互联网Internet的出现,为全世界范围内的电子信息发送提供了强有力的手段,Internet上有一种环球网WWW(World Wide Web)服务,WWW是基于超文本(hypertext)方式的信息查询工具,它可以帮助用户在全世界任何地方查询到所要的信息。WWW的文件格式采用统一的超文本置标语言(Hyper Text Markup Language,简称HTML语言),HTML语言有一个固定的SGML声明和一个DTD,这种语言实质上是SGML语言的一个应用。在WWW上的文件信息,大部分采用HTML语言格式。其版面信息丰富多彩,具有超文链接(hyperlink)的功能,便于实现文件之间的链接,使用户在计算机前能读遍他感兴趣的所有信息。HTML语言使用SGML语言定义它的语法,同时还定义了语义,说明各个元素及其属性在实际应用中的含义。HTML语言定义了题头(TITLE)、主体(BODY)、插图(IMG)、各级标题(H1～H6)、段落(P)、列表(有序表OL、无序表UL、表项I)、定义表(题目DT、定义DD)等元素。HTML语言中还有一个锚元素(anchor),这个锚元素可以在HTML文件之间,或者在HTML文件与其它数据资源之间,建立起超文链接,锚元素的属性值可区分链接的起点和终点。当计算机的光标链接到起点时,HTML的浏览器(browser)便可把屏幕转而显示该链接的终点,这样,用户便能够快速地浏览他自己感兴趣的内容。现在,国外的文本语料库都采用了SGML标准,我国的语料库建设如果不考虑SGML标准,就不能与世界接轨。

国际计算语言学界还把SGML标准应用到术语数据库的开发之中,其最重要的成果就是术语数据交换格式MARTIF的制定和应用。

术语数据库中术语数据的收集、存储、管理的方式是多种多样的,因此,机器可读术语数据的交换就成为了一个十分迫切需要解决的问题,为此,国际标准化组织ISO/TC37/SC3提出了机器可读术语数据交换格式(Machine-Readable Terminology Interchange Format,简称MARTIF)的国际标准。

MARTIF是SGML语言在术语数据库中的一个应用,MARTIF使得在不同硬件和软件格式中的术语可以互相交换,实现术语资源共享。

在MARTIF文件中使用的术语数据管理的基本单元是术语条目,MARTIF的文件是由术语条目构成的。由于术语数据库是多种多样的,为了进行术语数据的交换,把单个的术语条

目结构映射到 MARTIF 的结构中去,是一种行得通的做法。因此,我们不以术语数据库作为术语数据管理的基本单元,而以术语条目作为术语数据管理的基本单元。但是,在术语数据交换时,如果源术语数据库的结构比目标术语数据库丰富,交换时就会导致信息的丢失。为了避免信息丢失,有必要重新构造目标术语数据库,或者对目标术语数据库重新置标。

术语条目中可包含若干数据类目(data category)。MARTIF 的目的是对任何结构的术语数据库中的术语数据进行交换,因此,在术语条目中的数据类目应该具有一致性,这些数据类目之间的关系也必须在术语条目的范围之内来进行编码,这样,它们才有可能被重新分配到目标术语数据库中去。

在 MARTIF 文件中,数据类目是用类标识符(Generic Identifiers,简称为 GIs)和属性来描述和置标的。

类标识符也就是 MARTIF 中的标记(tag),主要有:

(1)〈termEntry〉(术语条目)

包括在一种语言中表达单一概念的、单独而完整的术语条目以及和它相关的描述性数据,在双语言或多语言术语的工作中,包含每种语言中的一个或多个术语及相关的描述性数据。

(2)〈tig〉(terminological information group,术语信息组)

在一个〈termEntry〉元素中,包含着与一个单独的术语相关的若干个信息元素,这些信息元素将在同一平面上发挥其功能,在〈tig〉的下位元素中,不容许嵌入别的信息。

(3)〈ntig〉(nested teminological information group,巢状术语信息组)

在一个〈termEntry〉中,当某些信息元素同内部元素相关,而不是同整个的 tig 相关的时候,使用巢状术语信息组。

为了便于在〈ntig〉中嵌入新信息,可以使用〈termGrp〉,〈termNoteGrp〉,〈descripGrp〉和〈adminGrp〉等元素。

(4)〈term〉(术语)

包括单词型术语、多词型术语(词组型术语)以及可以看作术语的符号。

(5)〈termGrp〉(术语组)

包括一个〈term〉以及至少一个附加到该术语上的巢状元素。

(6)〈termNote〉(术语说明)

包括与术语相关的信息。

(7)〈termNoteGrp〉(术语说明组)

包括一个〈termNote〉以及至少一个附加到与术语相关的信息上的巢状元素,可用来在〈termGrp〉之中嵌入新的信息。

(8)〈descrip〉(描述性信息)

包括诸如定义、上下文、描写概念和术语的解释等描述性信息。

(9)〈descripGrp〉(描述性信息组)

包括一个〈descrip〉以及至少一个附加到描述性信息中的巢状元素。

(10)〈admin〉(行政性信息)

包括行政数据。

(11)〈adminGrp〉(行政信息组)

包括一个〈admin〉元素以及至少一个附加到行政信息中的巢状元素。

(12)〈note〉(说明)

包括对〈termEntry〉、〈tig〉、〈ntig〉或者对〈…Grp〉等元素的说明或意见。

(13)〈ptr〉(pointer,指针)

包括一个指针,它指向当前文件中的另一个位置,〈ptr〉不能作为元素的内容与附加的文本相联系。

(14)〈ref〉(reference,参照)

在当前文件中,把一个参照指定到另一个位置,〈ref〉可以作为元素的内容与附加的文本相联系。

(15)〈xref〉(x-reference,x-参照)

对图形、图式、表格、其它外部文件指派一个参照。

(16)〈hi〉(highlighting,高亮度显示)

用高亮度显示的方法来标识一个词或一个短语,以便突出它在周围文本中的地位。

(17)〈biblist〉(bibliographical list,文献表)

在文件的后面或者在另一个外部文件中,包含一个或多个由〈bibl〉引入的参考文献条目所构成的表。

(18)〈bibl〉(文献)

包括一个由参考文献信息构成的条目,它一般置于文件的后面,或者置于外部的文件中。

(19)〈item〉(项目)

包括文献信息的一个实例。

(20)〈itemGrp〉(项目组)

包括一个或多个〈term〉。

(21)〈foreign〉(外语)

说明一个单词或一个词组属于某种外语。

MARTIF 中使用的属性主要有:

(1)id(identifier,等同)

用于表示与某一个元素等同。

(2)lang(language,语言)

用于指出元素内容的语言。使用 ISO 639 中的代码。

〈tig〉、〈ntig〉等标记要使用 lang 这个属性。

(3)type(类别)

用于把一个类标识符 GI 同一个属性值联系起来,以便在数据类目中形成完整的标记名。

MARTIF 中标记名的完整形式如下:

(1)开始标记

〈descrip type = ‘definition’〉

其中,“〈”是开定界符,“descrip”是类标识符,“type”是属性,“=”是值标识符,“‘definition’”是 type 的属性值,“〉”是闭定界符。

(2)结束标记

〈/descrip〉

其中,“〈”是开定界符,“/”是表示结束的符号,“descrip”是类标识符,“〉”是闭定界符。

例如,opacity(阻光度)这个英语术语条目可用标记和属性描述如下:

```
〈termEntry id = ‘ ID00073578 ’〉
  〈descrip type = ‘ subjectField ’〉appearance of materials〈 / descrip〉
  〈ntig lang = en〉
    〈termGrp〉〈term type = ‘ preferredTerm ’〉opacity〈 / term〉
      〈termNote type = ‘ partOfSpeech ’〉n〈 / termNote〉〈 / termGrp〉
    〈descripGrp〉〈descrip type = ‘ definition ’〉degree of obstruction
      to the transmission of visible light〈 / descrip〉〈prt type =
      ‘ sourceIdentifier ’target = ‘ ASTM. E284 ’〉〈 / descripGrp〉
    〈adminGrp〉〈admin type = ‘ responsibility ’〉E12〈 / admin〉
    〈 / adminGrp〉
  〈 / ntig〉
〈 / termEntry〉
```

其中,〈termEntry id = ‘ ID00073578 ’〉表示术语条目的号码为‘ ID00073578 ’。描述性信息〈descrip type = ‘ subjectField ’〉带有属性 type,这个 type 的属性值为‘ subjecField ’(专业领域),该专业领域为 appearance of materials(材料的外观),〈 / descrip〉是描述性信息的结束标记。关于专业领域的描述性信息涉及到整个的术语条目,置于〈termEntry〉之后,其它的〈tig〉或〈ntig〉之前。

在这个术语条目中,因为需要描述若干个信息,因此,使用巢状元素〈ntig〉,〈ntig lang = en〉表示用英语写的巢状术语信息组。这个〈ntig lang = en〉包括三个组元素:〈termGrp〉,〈descripGrp〉和〈adminGrp〉。它们构成了巢状术语信息组〈ntig〉下面的第一个层次。第一个组元素〈termGrp〉(术语组)又包括两个元素:〈term〉(术语)和〈termNote〉(术语说明),它们构成了巢状术语信息组的第二个层次。这两个元素都带有属性,〈term type = ‘ preferredTerm ’〉表示〈term〉的 type 属性的属性值为‘ preferredTerm ’(优先术语),这个优先术语就是 opacity(阻光度),〈 / term〉是〈term〉的结束标记;〈termNote type = ‘ partOfSpeech ’〉表示术语说明〈 termNote 〉的 type 属性的属性值为‘ partOfSpeech ’(词类),术语 opacity 的词类为名词 n,〈 / termNote 〉是〈termNote〉的结束标记,〈 / termGrp〉是第一个组元素〈termGrp〉的结束标记。第二个组元素〈descripGrp〉(描述性信息组)也包括两个元素〈descrip〉(描述性信息)和〈prt〉(指针),它们也构成巢状术语信息组的第二个层次,这两个元素都带有属性,〈descrip type = ‘ definition ’〉表示〈descrip〉的 type 属性的属性值为‘ definition ’(定义),这个定义就是 degree of obstruction to the transmission of visible light(可见光传播的阻碍程度),〈 / descrip〉是〈descrip〉的结束标记;〈prt type = ‘ sourceIdentifier ’target = ‘ ASTM. E284 ’〉表示指针的 type 属性的属性值为‘ sourceIdentifier ’,说明该指针指向‘ sourceIdentifier ’(源标识符),其目标文件是‘ ASTM. E284 ’(美国材料实验学会. E284),指针不作为元素内容,故无结束标记;〈/ descripGrp〉是第二个组元素〈descripGrp〉的结束标记。第三个组元素是〈adminGrp〉(行政信息组),这个组元素只有一个元素〈admin〉(行政信息),它也是巢状术语信息组的第二个层次,〈admin〉也带有属性,〈admin type = ‘ responsibility ’〉表示 type 属性的属性值为‘ responsibility ’(责任者),这个责任者是 E12(责任者的代号),〈 / admin〉是〈admin〉的结束标记,〈 / adminGrp〉是〈adminGrp〉的结束标记,〈 / ntig〉是巢状术语信息组〈ntig〉的结束标记,〈 / termEntry〉是术语条目的结束标记,整个的术语条目描述完毕。

MARTIF 规定了术语条目〈termEntry〉应该遵循的如下规则:

(1) 每一个单独的概念设立一个〈termEntry〉,一个〈termEntry〉表示一个单独的概念。

(2) 每个术语、同义词、术语变体等都有它们各自的〈tig〉或〈ntig〉,必要时,可以允许交叉参照。

(3) 要使用标准化的数据类目。

(4) 如果某个元素涉及到整个的〈termEntry〉而不仅仅涉及到一个〈tig〉或一个〈ntig〉,那么,就要把这个元素置于标记〈termEntry〉之后,第一个〈tig〉或〈ntig〉的标记之前。

(5) 如果一个〈note〉或〈termNote〉涉及到个别的元素(例如〈term〉这样的元素),而不涉及到整个的术语信息组,就使用〈ntig〉元素,再带上适当的组元素(如〈termGrp〉,〈descripGrp〉或〈adminGrp〉),并把这个〈note〉或〈termNote〉嵌入到组元素中。哪怕是只有一个元素要用〈ntig〉,也必须使用巢状元素。不要求嵌入信息的简单的术语信息组就使用〈tig〉元素;在同一个〈termEntry〉中,可以同时使用〈tig〉和〈ntig〉。

(6) 如果为了说明某个数据类目而须引入标记〈termNote〉,那么,就要在元素〈termGrp〉中,引入〈termNoteGrp〉以及〈termNote〉和相应的〈note〉。

(7) 如果嵌入到一个组元素〈…Grp〉中的某一元素必须与另外的元素相参照,但是这个元素并不涉及到所有的组元素,那么,就使用标记〈ref〉,把要参照的信息包括到元素〈ref〉的内容中。

(8) 每一个〈tig〉或〈ntig〉都应该具有属性 lang(语言)。属性可用于任何元素之中,也可以用于一切的子元素中,除非该子元素也有自己的属性 lang,或者属性 lang 已经包含在其它的元素之中。

(9) 日期的写法应该遵照 ISO 8610 的规定,按"年 - 月 - 日"的顺序书写,例如,1995 年 10 月 30 日应写为 1995 - 10 - 30。

MARTIF 还规定了自己的文件类型定义 DTD,包括文件类型声明、元素声明、属性定义表声明、实体声明等,这里就不再细说了。

MARTIF 的 DTD 规定了 MARTIF 文件的基本结构。

一个完整的 MARTIF 文件包括序言(prologue)和紧跟在序言后面的 MARTIF 文件实例(document instance)两部分。MARTIF 文件实例又包括 martif 文件头(〈martifHeader〉)和正文(text)两部分。正文又包括正文前部(front)、正文主体(body)和正文后部(back)三部分,其中,正文前部和正文后部是随选的,正文主体就是一系列的术语条目,可表示如下图:

Ⅰ. 序言
Ⅱ. 文件实例(〈martif〉)
  A. 文件头(〈martifHeader〉)
  B. 正文
    1. 正文前部(随选)
    2. 正文主体
      a. 第一个术语条目〈termEntry〉
        (至少一个)
      b. 第二个术语条目〈termEntry〉
      c. 其它的术语条目等等
    3. 正文后部(随选)

MARTIF 文件实例的结构可以用 DTD 的格式描述如下:

```
〈martif〉
〈martif Header〉
  …〈文件头置于此〉
〈 / martif Header〉
〈text〉
    〈body〉
      …〈术语条目置于此〉
    〈 / body〉
    〈back〉
      …(参考文献〈xref〉置于此)
      …(其它任何的外部参照也置于此)
    〈 / back〉
〈 / text〉
〈 / martif〉
```

在对一个 MARTIF 文件进行转换之前,首先必须用 SGML 语言的分析程序对其进行结构分析,确保 MARTIF 文件完全符合 SGML 语言的规定。用 MARTIF 置标后的机器可读术语数据,由于遵从了统一的术语交换格式,可以方便地进行术语数据的交换,推动了术语数据在全世界范围内的传播。

参 考 文 献

冯志伟(1988),国外术语数据库研制概况。《自然科学术语研究》第 2 期,—页。

冯志伟(1996),《自然语言的计算机处理》,上海外语教育出版社。

冯志伟(1995),《自然语言机器翻译新论》,语文出版社。

冯志伟(1997),《现代术语学引论》,语文出版社。

陈 球,标准通用置标语言 SGML 语言概况。《国际电子报》1996 年 6 月 17 日,第 79 版。

ISO DIS 12200.2:Terminology-Computer Application-Machine-readable Terminology Interchange Format(MARTIF),1996.

ISO 8879:Information Processing-Text and Office System-Standard Generalized Markup Language(SGML),1986.

Goldfarb,C. F.,(1990), *The SGML Handbook*,Oxford:Clarendon Press.

作者通讯地址:100010 北京朝内南小街 51 号 国家语言文字工作委员会

哀悼叶蜚声先生

本刊编委、北京大学中文系教授叶蜚声先生因病于 1998 年 8 月 23 日在北京逝世。享年 72 岁。

叶蜚声先生长期任本刊编委,为本刊的组稿、审稿、撰稿等各项工作都作出了贡献。他为人正派,乐于助人,工作勤奋。这些都值得我们学习。对于他的逝世本刊表示沉痛的哀悼。

本刊编辑部

Contents

Abstracts of major articles

Feng, Zhiwei, **SGML and its application in natural language processing** It discusses the principle and method of Standard Generalized Markup Language (SGML) based on ISO 8879 - 1986: SGML declaration, Document Type Definition, and Document Instance. Also addressed is the Machine - Readable Terminology Interchange Format (MARTIF) based on ISO DIS 12200, an application of SGML in terminological database.

Xu, Yiping, **A survey of contemporary studies of Japanese grammar** It presents a survey of 50 year studies of Japanese grammar by Japanese scholars. Grammatical thoughts and schools are analyzed. The latest trends in syntax, verb and particle studies are also discussed.

Zou, Chongli, **A review of discourse representation theory** It first examines the historical root of discourse representation theory. It then focuses on the theoretical framework of DRT, on its differences from Montague Grammar in particular. It concludes with a discussion of the strength and weakness of DRT.

Dong, Xiufang, **Zero anaphora as prepositional complement in classical Chinese** Zero anaphora is a grammatical feature unique to classical Chinese, that is, not found in modern Chinese. This paper, drawing data from The Book of History, analyzes the syntactic features of zero anaphora. It also explores the historical factors that contribute to its disappearance in modern Chinese.