



中华人民共和国国家标准

GB/T 14814—93

信息处理 文本和办公系统 标准通用置标语言(SGML)

Information processing—Text and office systems—
Standard generalized markup language(SGML)

1993-12-24 发布

1994-08-01 实施

国家技术监督局 发布

目 次

0 引言	(1)
0.1 背景	(1)
0.2 目标	(2)
0.3 组织	(3)
1 主题内容	(4)
2 适用范围	(4)
3 引用标准	(4)
4 定义	(5)
5 记法	(26)
5.1 语法记号	(26)
5.2 顺序符号和选择符号	(26)
6 实体结构	(27)
6.1 SGML 文件	(27)
6.2 SGML 实体	(27)
6.3 数据实体	(28)
7 元素结构	(28)
7.1 前导说明部分	(28)
7.2 文件元素	(28)
7.3 元素	(29)
7.4 开始标记	(30)
7.5 结束标记	(30)
7.6 内容	(31)
7.7 文件类型说明	(32)
7.8 通用标识符(GI)说明	(32)
7.9 属性说明表	(32)
8 处理指令	(34)
8.1 数量	(34)
9 公共结构	(34)
9.1 可替换字符数据	(34)
9.2 字符数据	(34)
9.3 名字	(35)
9.4 实体引用	(35)
9.5 字符引用	(37)
9.6 定界符识别	(37)
9.7 置标抑制	(41)
9.8 容量	(41)
10 置标声明:通用的	(41)

10.1	声明的组成部分	(41)
10.2	形式公用标识符	(43)
10.3	注释声明	(45)
10.4	特殊标出节声明	(45)
10.5	实体声明	(46)
11	置标声明:文件类型定义	(47)
11.1	文件类型声明	(47)
11.2	元素声明	(48)
11.3	属性定义表声明	(51)
11.4	记法声明	(53)
11.5	便捷引用映射声明	(53)
11.6	便捷引用使用声明	(53)
12	置标声明:链接处理定义	(54)
12.1	链接类型声明	(54)
12.2	链接集声明	(55)
12.3	链接集使用声明	(56)
12.4	当前链接集	(57)
13	SGML 声明	(57)
13.1	文件字符集	(57)
13.2	容量集	(59)
13.3	具体语法范围	(59)
13.4	具体语法	(59)
13.5	特征使用	(63)
13.6	特定应用信息	(64)
14	基准具体语法和核心具体语法	(64)
15	一致性	(66)
15.1	符合于 SGML 的文件	(66)
15.2	符合于 SGML 的应用	(66)
15.3	符合于 SGML 的系统	(66)
15.4	合法的 SGML 语言分析器	(67)
15.5	文档编制要求	(67)
15.6	系统声明	(68)
附录 A	通用置标介绍(参考件)	(71)
A1	置标过程	(71)
A2	描述性置标	(72)
A3	严格的置标	(74)
A4	结论	(75)
附录 B	基本概念(参考件)	(76)
B1	文件、文件类型定义、过程	(76)
B1.1	文件	(76)
B1.2	文件类型定义	(76)
B1.3	过程	(76)
B2	置标	(77)

B3	从正文中区分置标	(78)
B3.1	描述性置标标记	(78)
B3.2	其他置标	(78)
B3.3	记录边界	(79)
B4	文件结构	(79)
B4.1	文件类型定义	(79)
B4.2	元素声明	(80)
B5	属性	(82)
B5.1	规定属性	(82)
B5.2	声明属性	(83)
B6	实体	(85)
B6.1	实体引用的语法	(85)
B6.2	声明实体	(86)
B7	字符	(87)
B7.1	字符分类	(87)
B7.2	字符引用	(88)
B7.3	将定界符字符作为数据使用	(88)
B8	特殊标出节	(89)
B8.1	忽略一个特殊标出节	(89)
B8.2	单个文件的版本	(90)
B8.3	不可分析的节	(90)
B8.4	临时节	(91)
B8.5	关键字说明	(91)
B8.6	定义特殊标出节为实体	(91)
B9	唯一标识符属性	(92)
B10	内容引用属性	(92)
B11	内容模型例外参数	(93)
B11.1	被包含的元素	(93)
B11.2	被排斥的元素	(93)
B12	文件类型声明	(93)
B13	数据内容	(94)
B13.1	数据内容表示	(94)
B13.2	数据内容记法	(95)
B14	用户剪裁	(97)
B14.1	SGML 声明	(97)
B14.2	用户剪裁的影响	(97)
B15	一致性	(98)
附录 C	附加的概念(参考件)	(98)
C1	置标最简化特征	(98)
C1.1	SHORTTAG:带省略置标的标记	(98)
C1.2	OMITTAG:可省略标记	(100)
C1.3	SHORTREF:便捷引用定界符可以替代完整的实体引用	(103)
C1.4	DATATAG:数据也可是标记	(106)

C1.5	RANK;可以从标记中省略层号	(109)
C2	LINK 特征;SIMPLE、IMPLICIT 以及 EXPLICIT	(110)
C2.1	链接处理定义	(110)
C3	其他特征	(111)
C3.1	CONCUR;可同时出现多个文件实例.....	(111)
C3.2	SUBDOC;可以出现嵌套的子文件实体	(111)
C3.3	FORMAL;形式化的公用标识符	(112)
附录 D	公用文本(参考件).....	(112)
D1	元素集合	(112)
D1.1	普通元素类型	(112)
D1.2	形式上的元素类型	(112)
D2	数据内容记法	(113)
D3	变异的语法	(113)
D3.1	多代码具体语法	(113)
D4	实体集	(114)
D4.1	一般性考虑	(115)
D4.2	字母表字符	(116)
D4.3	一般用途	(125)
D4.4	技术用途	(131)
D4.5	附加的数学符号	(135)
附录 E	应用举例(参考件).....	(143)
E1	文件类型定义	(143)
E2	计算机图形元文卷	(147)
E3	与设备无关的代码扩充	(148)
E3.1	代码扩充方式	(149)
附录 F	实现考虑(参考件).....	(151)
F1	一个 SGML 语言分析器模型	(151)
F1.1	物理输入	(151)
F1.2	识别状态	(152)
F1.3	置标最简化	(152)
F1.4	转换	(153)
F1.5	命令语言的类比	(153)
F2	初始化	(153)
F2.1	初始的过程映射	(153)
F2.2	链接处理说明	(153)
F2.3	共存文件实例	(153)
F3	动态过程映射	(153)
F4	错误处理	(154)
附录 G	一致性的分类和确认(参考件).....	(154)
G1	分类代码	(154)
G1.1	特征代码	(154)
G1.2	合法性代码	(156)
G1.3	语法代码	(156)

G2 确认方面的考虑	(156)
附录 H SGML 内容模型的理论基础(参考件)	(156)
H1 模型组记法	(157)
H2 自动机理论的应用	(157)
H3 偏离自动机理论	(157)
附录 I 非一致性的变化(参考件)	(158)
I1 定长通用标识符	(158)
I2 单个定界符	(158)

中华人民共和国国家标准

信息处理 文本和办公系统 标准通用置标语言(SGML)

GB/T 14814—93

Information processing—Text and office systems—
Standard generalized markup language(SGML)

本标准等效采用国际标准 ISO 8879—1986《信息处理 文本和办公系统 标准通用置标语言(SGML)》及修正 1—1988。

0 引言

本标准规定了一种用于文件表示的语言,称之为“标准通用置标语言”(SGML)。就其最广泛的定义而言,可以将 SGML 用于排版,其范围包括从传统的单一媒体数据的排版直到多媒体数据的排版。此外,也可以将 SGML 用于办公文件的处理,以满足人们阅读和在排版系统之间进行文件交换的需要。

0.1 背景

可以抽象地将文件看成是由多种类型的元素组成的一种结构。例如,作者可以将若干章组成一本书,每章中又含有段落和附有文字性注解的插图。又如,编辑人员可以将若干篇文章编成一本刊物,而每篇文章中又可以含有若干段,段中又可以含有文字,等等。

处理程序采用不同的方法来处理上述这些元素。例如,格式化程序就可用醒目的字体打印标题,在段首或各段之间留空,这就从视觉上将文件的结构和属性展示给了读者。而当为信息检索系统建立标题字典时,则可能对标题中的文字赋予特别的含义。

虽然在文件的属性与其处理之间的这种联系现在看来是清晰的,然而就早期的文本处理方法而言,这种联系是模糊的。在自动化排版出现之前,编辑人员先使用特殊的处理指令对底稿进行“置标”,然后再由排版工人按照这些指令排出所要的格式。指令和文件结构之间的所有联系完全存在于编辑人员的脑中。

早期计算机排版继承了上述方法,用与处理相关的“置标”加入计算机可读的文件卷中。虽然加入的置标依然由特定的处理指令组成,但这些处理指令采用的是格式化程序的语言,而不是排版工人使用的语言。然而,这种文卷如果不改变其中的全部置标,就很难再用于其他不同的目的或不同的计算机系统中。

随着用户水平日益提高及文本处理程序功能日益增强,人们开发出许多解决上述问题的方法。例如,使用“宏调用”(或“格式调用”)将文件中需要处理的地方标识出来,而将实际的处理指令放在文件外部的“过程”之中(或“宏定义”或“存储格式”之中),这样就能够比较容易地对它们进行修改。

虽然可以将宏调用放在文件中的任意位置上,但是用户开始逐步认识到大部分宏调用都是放在文件元素的开头或结尾处。于是,很自然地选择表示元素类型的“通用标识符”来命名这些宏调用,而不是用特定处理的名字命名(例如,用“heading”代替“format—17”,即用“标题”代替“格式 17”),这是实行“通用编码”(或“通用化的标记”)的开始。

通用编码使自动文本处理系统迈出重要的一步,它反映了文件属性与处理之间的自然关系。七十年代初,“通用置标语言”的出现为通用编码技术提供了一种形式化语言的基础,从而进一步推动了这种发

国家技术监督局 1993-12-24 批准

1994-08-01 实施