

UVOD V SGML

SESTAV ZA OPIS ZGRADBE BESEDIL

Vladimir Batagelj,
Univerza v Ljubljani, Matematika, Jadranska 19, 61111 Ljubljana
e-mail: Vladimir.Batagelj@uni-lj.si
<http://vlado.mat.uni-lj.si/vlado/sgml/sgmluvod.htm>
in <http://vlado.mat.uni-lj.si/ftp.htm#sgml>

Povzetek

SGML - Standard Generalized Markup Language je sestav, namenjen za pripravo zvrsti spisov. Z dopolnitvijo spisov z opisom njihove zgradbe (v od vrste programa ali računalnika neodvisni obliki) omogočimo izdelavo različnih programov za njihovo uporabo (priprava, vzdrževanje, prikaz, iskanje in analiza) in prenosljivost med raznovrstnimi računalniki. V sestavku je podana kratka predstavitev sestava SGML.

Abstract

SGML - Standard Generalized Markup Language is a system for preparing document type definitions. Embedding of descriptive markup within documents enables us to produce different application programs (to prepare, display, maintain, search and analyze documents) and to transfer documents among different platforms. In the paper a short introduction to SGML is given.



Sestavek je sestavljen iz dveh delov:

- splošne predstavitve sestava SGML, namenjene širokemu krogu bralcev - predvsem informatikom, založnikom, knjižničarjem in arhivarjem; ter
- programerske predstavitve, ki naj pomaga pri prvih korakih uporabe SGMLja in vzpodbudi k samostojnemu poglobljanju v njegove tančine.

Na koncu je zbranih še nekaj smerokazov za samostojno brskanje.

1. Zakaj SGML?

1.1. Besedila, računalnik in označevanje

Po nekaterih ocenah v ZDA:

- porabijo 20% narodnega dohodka za ustvarjanje novih informacij;
- 90% teh informacij je na papirju in ne v računalniški obliki;
- pri pripravi spisov se 30% časa porabi za iskanje informacij in nadaljnjih 30% za oblikovanje spisov.

Podobno velja tudi drugod v razvitem svetu.

Zaradi tehnoloških sprememb je potrebno tudi informacijo v računalniški obliki pogosto pretvarjati za nove programe. Kompaktne plošče in omrežja omogočajo hranjenje in razširjanje velikih količin podatkov.

Po drugi strani uporabnik želi informacijo, ki je prilagojena njegovim potrebam.

SGML - Standard Generalized Markup Language je poskus odgovora na te izzive.

V tiskarstvu se *označevanje* pri oblikovanju besedil uporablja že dolgo. Uporabljajo ga oblikovalci, kot navodila stavecem, za določitev *kje* in *kako* naj se dani del besedila prikaže (sprememba pisave in njenih lastnosti, prehod na novo stran,...).

Podobno pri računalniškem oblikovanju besedila oblikovalnik (Word Star, Word, Word Perfect, ...) vključi med besedilo dodatne sestavine, ki določajo prikaz besedila. Taki so tudi opisi besedil v jezikih Postscript, RTF (Rich Text Format), Roff, ... Ker ti opisi natanko določajo, kaj se zgodi s posameznim delom besedila, govorimo o *postopkovni označitvi* besedila.

Korak naprej pomenijo zvrsti besedila v LaTeXu in oblikovanje besedil z uporabo *slogov* (Word, ...), kjer za posamezni del besedila povemo *le kaj je* - opisna označitev, oblikovalnik pa nato sam poskrbi za ustrezno oblikovanje glede na izbrani slog. Prednost opisne označitve se pokaže pri preoblikovanjih besedila zaradi sprememb (članek, prosojnice).

Vendar tudi opisna označitev še ne omogoča zadovoljive računalniške obdelave besedil. To dosežemo šele s *posplošenim* označevanjem, s katerim v besedilo vnese-mo opis njegove zgradbe - določitev značilnih sestavin in njihove medsebojne povezanosti. S tem ustvarimo *znakovno podatkovno bazo*. SGML je sestav, ki to omogoča.

Natančneje, SGML zagotavlja:

- ločitev opisa zgradbe od opisa oblike prikaza - znakovna podatkovna baza;
- uporabo gradiv v različne namene, vključevanje dodatnih informacij;
- prenosljivost: neodvisnost od vrste računalnika (znakovni opis v izbrani kodi - praviloma koda ASCII);
- povečanje učinkovitosti priprave, uporabe, vzdrževanja in razpečevanja informacije;
- trpežnost in trdoživost gradiv: nevezanost na posameznega proizvajalca programske opreme;
- odprtost: pestrost orodij različnih proizvajalcev;
- nedvoumnost, zmanjšanje števila napak v podatkih;
- izboljššan nadzor: povezave med posameznimi sestavinami, omejen dostop do posameznih podatkov.

1.2. Razvoj sestava SGML

Septembra leta 1967 je William Tunnicliffe na sestanku na Kanadskem državnem uradu za tisk predstavil zamisel o potrebi po ločitvi vsebine in oblike spisov.

Konec šestdesetih let je Stanley Rice, oblikovalec knjig iz New Yorka, predlagal nabor značk za opis zgradbe spisov. Ta nabor je bil ob podpori združenja *Graphic Communications Association* razvit v prvi opisni nabor *GenCode*.

Na teh osnovah so leta 1969 Charles Goldfarb, Edward Mosher in Raymond Lorie za podjetje IBM ustvarili GML - *Generalized Markup Language*. Vanj so vpeljali tudi pojem zvrsti spisa in gnezdenja sestavin. GML je postal osnova IBMove programske podpore založništva. Goldfarb je nadaljeval raziskave o zgradbi spisov in razvil vrsto dodatnih sestavin.

Leta 1978 je bila pri ANSI (*American National Standards Institute*) ustanovljena skupina za pripravo standarda jezika za opis besedil, ki bi izhajal iz jezika GML. Prvi osnutek standarda je bil predstavljen leta 1980; šesta različica pa je že dobila vlogo industrijskega standarda (GCA 101-1983).

Leta 1984 se je začelo sodelovanje med ANSI in ISO (*International Standards Organization*) pri pripravi mednarodnega standarda. Osnutek je bil objavljen leta 1985, naslednje leto pa tudi sam standard SGML (ISO 8879:1986 *Information processing - Text and office systems - Standard Generalized Markup Language*).

1.3. Projekti

SGML je bil že med nastajanjem in ob samem začetku podprt z dvema večjima projektoma:

1.3.1. Electronic Manuscript Project

V letih 1983-1987 je delovna skupina pri *Association of American Publishers* pripravila v SGML opise zvrsti *knjiga, časopis in članek*, s čimer naj bi poenotila in olajšala izmenjavo rokopisov med pisci in založbami. Te rešitve je sprejelo tudi porajajoče se založništvo kompaktnih plošč.

1.3.2. CALS

CALS - spočetka (1987) *Computer-aided Acquisition and Logistic Support*, sedaj *Continuous Acquisition and Lifecycle Support* je projekt ameriškega obrambnega ministrstva (US Department of Defense). Že februarja leta 1988 je bil izdan standard MIL-M-28001, ki temelji na SGML. Cilj projekta je zagotoviti, v računalniški obliki, enotno tehnično dokumentacijo vseh naročenih izdelkov. To naj zagotovi pocenitev, sprotno obnovo in hitrejši dostop do iskane informacije. O pomenu projekta marsikaj pove podatek, da ameriška bojna ladja nosi tudi 20 do 25 ton priročnikov o njej. Odslej naj bi vsi dobavitelji DoD pripravljali dokumentacijo po pravilih CALS.

V devetdesetih letih je število projektov zelo naraslo. Med njimibomo omenili le tri:

1.3.3. HTML

HTML (*HyperText Markup Language*) se uporablja na WWW (*World-Wide Web*) od leta 1990 dalje. To je preprosti jezik za označevanje, ki temelji na SGML in omogoča pripravo sestavkov (besedilo, slika, zvok, video,...) prenosljivih med različnimi vrstami računalnikov. HTML je v trajnem razvoju, za katerega skrbi IETF (*Internet Engineering Task Force*). Trenutno večina pregledovalnikov podpira HTML 2.0; marca 1995 pa je že izšel osnutek za HTML 3.0.

1.3.4. TEI - Text Encoding Initiative

TEI je mednarodni projekt, ki ga podpirajo *Association for Computing in Humanities, Association for Literary and Linguistic Computing in Association for Computational Linguistics*. Usmerjen je na označevanje gradiv s področja umetnosti in družbenih ved in naj bi ustvaril priporočila za pripravo in izmenjavo besedil v računalniški obliki za raziskovalne in založniške namene.

1.3.5. UTF

Delovne skupine pri *International Press Telecommunications Council in Newspaper Association of America* so za osnovo novega standarda UTF (*Universal Text Format*), ki bo nadomestil standarda IPTC 7901 in ANPA 1312, izbrale SGML.

2. Osnove SGML

2.1. Osnovne sestavine

Pri opisu osnov sestava SGML bomo uporabljali *običajni zapis*. Njegova pravila (uporabljeni koda, ločila, dolžina imen, ...) lahko po potrebi spremenimo.

2.1.1. Značke

Opis zgradbe vnašamo z značkami (tag, ELEMENT). Posamezna značka je obdana z znakoma *začetek značke* <; in *konec značke* >;. Na primer <odstavek>. Poznamo:

- *samostojne značke ali določila*, pri katerih začetku značke sledi *ime značke*; in
- *oklepajne značke*, ki jih sestavlja par značk. Z oklepajnimi značkami omejujemo *gradnike* besedila. Prva značka para - *začetek* je po obliki enaka samostojni znački; druga značka para - *konec* pa ima po začetku značke znak /, ki mu sledi *ime značke*. Pravzaprav je </ sestavljen znak za začetek konca značke. Na primer, z oklepajno značko kraj povemo, da uklenjeni del besedila

pripeljali smo se v <kraj>Ljubljano</kraj>, glavno mesto

predstavlja nek kraj.

V običajnem zapisu je ime značke sestavljeno iz črk, števč in znakov . ter -; začeti mora s črko; ni razlike med velikimi in malimi črkami; dolgo je lahko največ 8 znakov.

2.1.2. Delci

V besedilo lahko vključujemo tudi *delce* besedila (ENTITY), katerih vsebina se nahaja izven samega spisa - v pomnilniku ali na datoteki. Delci so lahko od znaka do vsebine cele datoteke. Vključitev nekega delca zahtevamo tako, da zapišemo njegovo ime predznačeno z znakom & in zaključeno s podpičjem ;. Za imena delcev veljajo v običajnem zapisu ista pravila kot za imena značk, le da razlikujemo med velikimi in malimi črkami. Delci nam omogočajo:

- zapisati posebne znake, ki jih ni v kodi, v kateri pripravljamo opis spisa;
- vpeljati okrajšave za dolge in ponavljajoče se dele besedila;
- pripraviti opis spisa na več datotekah;
- vključevanje lokalnih podatkov - datum, ustanova, vrsta računalnika ...

Delec oblike &#n; označuje znak s kodo n (desetiško). SGML pozna tudi delce &#RS; (record start - začetek zapisa), &#RE; (record end - konec zapisa), &#SPACE; (presledek), &#TAB; (predelčnik) in imena za znake, ki nastopajo v značkih. <;, >;, &;, "; dajo zaporedoma znake <, >, &, ".

Kako sami določimo delce, bomo razložili v nadaljevanju.

2.1.3. Stavki

Določitev delcev in značk ter opis medsebojnih zvez med posameznimi značkami omogočajo stavki. Ti so nekakšne samostojne značke, ki začenjajo s sestavljenim znakom *začetek stavka* <! . Temu sledi ime vrste stavka (ELEMENT, ENTITY, ATTRIBUTE, NOTATION, LINK, SHORTREF, ...) in za njim, glede na vrstostavka, v pred-

pisanem vrstnem redu in obliki zahtevane sestavine. V stavku ima del od začetnega -- do končnega -- (oba vključno) vlogo pojasnila. To vlogo ima tudi stavek <!>.

2.1.4. Ukazi

Tudi *ukazi* so nekakšne samostojne značke, ki začenjajo s sestavljenim znakom *začetek ukazov* <? . Temu sledi zaporedje ukazov namenjenih programu, ki obdeluje spis.

2.2. Opis spisa v SGML

Opis spisa v SGML je sestavljen iz treh delov:

1. določitev načina zapisa
2. opis zvrsti spisa
3. spis

V posameznih programih oziroma sestavih sta pogosto prvi in drugi razdelek stalna - vgrajena v program. Tedaj mora uporabnik pripraviti le sam spis. To je uporabljeno v HTML, ki uporablja običajni zapis z dolžino imen podaljšano na 72. Opis jezika HTML (zvrst spisa) pa je tudi vgrajen v pregledovalnike (Netscape, Mosaic, Lynx, ...). Stavki lahko nastopajo le v prvih dveh delih opisa.

2.2.1. Določitev načina zapisa

SGML daje precej svobode pri določitvi načina zapisa, vendar, kakor rečeno, se bomo v tem sestavku držali običajnega zapisa. Tega napovemo na začetku opisa spisa s stavkom

```
<!SGML "ISO 8879:1986">
```

ki ga ponavlja lahko opustimo.

2.2.2. Opis zvrsti spisa

Zvrst spisa je določena z značkami in delci, ki so značilni za neko skupino spisov. Te so običajno zbrane na eni ali več datotekah s podaljškom DTD (*Document Type Definition*).

```
<!DOCTYPE zvrst SYSTEM "pot\zvrst.DTD">
```

V stavku DOCTYPE lahko v oglatih oklepajih vnesemo popravke in dopolnila opisa zvrsti spisa in poskrbimo za posamezne delce.

Praviloma zvrst spisa sestavimo za večjo skupino uporabnikov, ki uporabljajo in si izmenjujejo istovrstne spise. Najpogosteje v opisu zvrsti uporabljamo naslednje stavke:

2.2.2.1. Določitev delcev

Najpreprostejša oblika določitve delca je vpeljava okrajšave, ki ima obliko

```
<!ENTITY ime "niz znakov">
```

Na primer

```
<!ENTITY UI "Uporabna Informatika">
```

Del spisa na datoteki proglasimo za delec s stavkom

```
<!ENTITY ime SYSTEM "pot\datoteka">
```

Na primer

```
<!ENTITY uvod SYSTEM "D:\KNJIGA\UVOD.SGM">
```

Z opisom spisa lahko povežemo (NOTATION) tudi ne-SGMLjevske datoteke (bitne slike, spise oblikovane v Wordu, ...).

Delce, ki vsebujejo ukaze, vpeljemo s stavkoma oblike

```
<!ENTITY ime PI "niz znakov">
<!ENTITY ime SDATA "niz znakov">
```

Prvo obliko PI (Processing Instructions) uporabljamo takrat, ko gre za čiste ukaze programu za obdelavo

```
<!ENTITY Stran PI "newpage">
```

druga oblika pa vrne programu za obdelavo niz znakov. Tako je v dodatkih k ISO 8879, v opisu nabora znakov Latin 2, znak č podan kot

```
<!ENTITY ccaron SDATA "[ccaron]"--=small
c, caron-->
```

Delec č vrne v program za obdelavo niz [ccaron]. Če pa v stavku [ccaron] nadomestimo z ~, bomo na starejših prirejenih tiskalnikih in zaslonih dobili željeni izpis.

Kadar želimo delec ime uporabiti v definiciji nekega drugega delca, ga vpeljemo s stavkom oblike

```
<!ENTITY % ime "niz znakov">
```

uporabimo pa z zahtevo % ime;. Na primer, kadar je delec na datoteki širše uporaben (del knjižnice), to poveemo z obliko

```
<!ENTITY % ime PUBLIC "javno ime">
```

Tako delec ISolat2 povežemo z opisom nabora znakov Latin 2, v dodatkih k ISO 8879, s stavkom

```
<!ENTITY % ISolat2 PUBLIC
"ISO 8879-1986//ENTITIES Added Latin 2//EN">
%ISolat2;
```

S stavkom

```
<!ENTITY #DEFAULT "niz znakov">
```

lahko določimo vrednost vključitev nenajavljenih delcev.

2.2.2.2. Značke

Značke vpeljemo s stavki oblike:

```
<!ELEMENT ime zac kon sestava>
```

Pri tem je ime ime značke. zac in kon lahko zavzameta vrednost - ali 0. V opisu spisa omogočata opuščanje posameznih značk, ki so določene z drugimi. Vrednost - pomeni, da je značka obvezna; vrednost 0 pa, da jo lahko opustimo. SGML pozna še več vrst okrajšav, vendar moramo biti pri njih uporabi previdni, ker lahko privedejo do dvomij.

sestava opisuje zgradbo in medsebojne povezanosti gradnikov in določil. Če uporabimo za sestavo besedico EMPTY, je pripadajoča značka določilo. Sicer je sestava določena z izrazom, ki pove, katere značke in v kakšnem vrstnem redu se lahko pojavijo znotraj gradnika. Izraz zgradimo postopno iz členov oblike:

- #PCDATA - (parsed character data) zaporedje znakov,
- ime - ime značke,
- clen? - člen je lahko prisoten ali odsoten,
- clen+ - člen se mora pojaviti vsaj enkrat,
- clen* - člen se lahko pojavi poljubnokrat - lahko je tudi odsoten,
- (clenⁱ) skupina: zaporedje členov povezanih z:
 - , - zaporedje,
 - | - ali, ali,
 - & - vsi, v poljubnem vrstnem redu.

Tako na primer

```
<!ELEMENT spis 0 0 ((naslov&pisec?),odstavek+)>
```

določa, da se spis (začetno in končno značko lahko opustimo) začne z naslovom in neobveznim piscem ali pa piscem in naslovom. Sledi neprazno zaporedje odstavkov.

Pri opisu sestave gradnikov, lahko opise gradnikov z enako sestavo združimo v en opis, ki ima namesto imena značke skupino imen, ločenih z |, združenih gradnikov.

Opis sestave lahko nadaljujemo tudi z znakom + predznačeno skupino imen, kar pomeni, da se te značke lahko pojavljajo kjerkoli znotraj dane značke. Predznak - ima nasprotni učinek.

Posamezni znački ime lahko pripišemo lastnosti s stavkom:

```
<!ATTLIST ime lastnost zaloga vrednost ...>
```

Trojica lastnost zaloga vrednost se ponovi za vsako lastnost, ki jo lahko pripišemo znački. Lastnost podamo z njenim imenom. Za zalogo vrednosti napišemo seznam vseh možnih vrednosti (naštete) ali pa navedemo eno od vrst vrednosti: ID, IDREF, CDATA, NUMBER, NUMBERS, NAME, NAMES, NMTOKENS, NUTOKENS, ... Za vrednost lahko napišemo izbrano vrednost ali eno od gesel:

- #REQUIRED - pri uporabi značke mora biti določena tudi lastnost;
- #IMPLIED - če vrednost ni podana, dobi lastnost vgrajeno vrednost;
- #CURRENT - če vrednost ni podana, dobi lastnost zadnjo uporabljeno vrednost.

Zaradi omejenega prostora, bomo tu naš opis sestava SGML prekinili. Seveda SGML omogoča še vrsto stvari, ki pa presejajo okvire uvoda.

3. Primer

Za primer si oglejmo poenostavljen opis zvrsti, ki ustrezno zbirkni vesti. Shranimo ga na datoteki VESTI.DTD:

```

<!--
% Document Type Definition for VESTI
% Vladimir Batagelj, september 1995-->
<!ENTITY % kk "kdo|komu">
<!ELEMENT vesti O O (vest*)>
<!ELEMENT vest - O (glava,telo) +(vrsta)>
<!ELEMENT glava - O ((%kk;)*&datum&sklic*)>
<!ELEMENT (%kk;) - O (oseba,naslov?)*>
<!ELEMENT (datum|clen|oseba|naslov|odstavek)
- O (#PCDATA)>
<!ELEMENT (sklic|vrsta)
- O EMPTY>
<!ELEMENT telo - O (odstavek|seznam)*>
<!ELEMENT seznam - O (clen)*>

<!ATTLIST vest
oznaka ID #REQUIRED
tajnost (zaupno|javno)
zaupno

>
<!ATTLIST sklic
oznaka IDREF #REQUIRED >
<!ATTLIST seznam
znak (crta|pika|crke|stej)
stej
>

```

Na datoteki ZNAKI.ENT pripravimo nekaj različnih rešitev za obravnavo naših črk čžš:

```

<!ENTITY ccaron SDATA "c"--small c, caron-->
<!ENTITY Ccaron SDATA "C"--capital C, caron-->
<!ENTITY scaron "&#123;"--small s, caron-->
<!ENTITY Sscaron "[ "--capital S, caron-->
<!ENTITY zcaron SDATA "z"--small z, caron-->
<!ENTITY Zcaron "Z"--capital Z, caron-->

```

Primer vesti je shranjen na datoteki JANEZ.SGM:

```

<vest oznaka=janez01 tajnost=javno>
<glava>
<kdo>
<oseba> X Y </oseba>
<naslov> aaaa <vrsta> bbbb </naslov>
</kdo>
<komu>
<oseba> U V </oseba>
<naslov> cccc <vrsta> dddd </naslov>
</komu>

<datum> 10. januar 1995 </datum>
</glava>
<telo>
<odstavek> &ccaron;ira &Ccaron;ara </
odstavek>
<odstavek> &zcaron;iga &Zcaron;aga </
odstavek>
</telo>
</vest>

```

Vse skupaj povežemo z datoteko VLADO.SGM:

```

<!-- VESTI - Vlado -->
<!doctype vesti SYSTEM "../vesti.dtd"
[ <!entity janez SYSTEM "../janez.sgm">
<!entity % znaki SYSTEM "../znaki.ent">
%znaki;]
>
<vesti>
<janez>
<vest oznaka=vlado01>
<glava>
<kdo> <oseba> U V <komu> <oseba> X Y
<naslov> cccc <vrsta> &Scaron;i&scaron;ka

```

```

<sklic oznaka=janez01>
<datum> 23. januar 1995
<telo>
<odstavek> bla <vrsta> bla
<odstavek> ble ble
</vesti>

```

Za pregled pripravljenega opisa bomo v DOSu uporabili Clarkov razčlenjevalnik nsgmls (SP parser). Da nam ne bo potrebno vsakič navajati vseh stikal, si jih pripravimo na datoteki SP.BAT:

```

SET DOS4G=quiet
nsgmls -deguv -f%1.ERR %1.SGM > %1.PRS

```

Sedaj lahko preprosto zahtevamo SP vlado. Na datoteki vlado.ERR dobimo izpis sporočil o napakah v opisu - v našem primeru ostane prazna. Na datoteki vlado.PRS pa dobimo razčlenjeni opis (prikazan je v dveh stolpcih - najprej preberite celoten prvi stolpec):

(VESTI	AOZNAKA TOKEN VLADO01
AOZNAKA TOKEN JANEZ01	ATAJNOST TOKEN ZAUPNO
ATAJNOST TOKEN JAVNO	(VEST
(VEST	(GLAVA
(GLAVA	(KDO
(KDO	(OSEBA
(OSEBA	- U V
- X Y	)OSEBA
)OSEBA	(KDO
(NASLOV	(KOMU
- aaaa	(OSEBA
(VRSTA	- X Y\n
)VRSTA	)OSEBA
- bbbb	(NASLOV
)NASLOV	- cccc
(KDO	(VRSTA
(KOMU	)VRSTA
(OSEBA	- [i{ka\n
- U V	)NASLOV
)OSEBA	(KOMU
(NASLOV	AOZNAKA TOKEN JANEZ01
- cccc	(SKLIC
(VRSTA	)SKLIC
)VRSTA	(DATUM
- dddd	- 23. januar 1995\n
)NASLOV	(DATUM
(KOMU	)GLAVA
(DATUM	(TELO
- 10. januar 1995	(ODSTAVEK
)DATUM	- bla
)GLAVA	(VRSTA
(TELO	)VRSTA
(ODSTAVEK	- bla\n
- \ c\ ira \ C\ ara	)ODSTAVEK
)ODSTAVEK	(ODSTAVEK
(ODSTAVEK	- ble ble
- \ z\ iga "Zaga	)ODSTAVEK
)ODSTAVEK	(TELO
(TELO	)VEST
)VEST	)VESTI
	C

Znak C na koncu razčlenitve pomeni *conforming* - opis je skladen z vrstjo spisa. Podrobneje si oglejte, kaj je razčlenjevalnik naredil s posameznimi delci za naše črke. Primerjajte tudi vest iz datoteke JANEZ.SGM (polni opis) z vestjo z datoteke VLADO.SGM (okrajšani opis).

4. Programska in druga podpora

Čeprav lahko opise spisov pripravimo s poljubnim znakovnim urejevalnikom, je to precej lažje početi s prilagojenimi urejevalniki, ki upoštevajo zvrst spisa. Taki so na primer *Emacs*, *Author/Editor (SoftQuad)* in dodatek *Word-Perfectu*. Med proizvajalci programske podpore za SGML sta najbolj znani podjetji *SoftQuad* in *ArborText*.

Za preverjanje pravilnosti spisov je na voljo več razčlenjevalnikov. Med njimi sta najbolj znana *Clarkova sgmls* in novejši *nsghmls (SP)*, ki smo ga tudi mi uporabili v našem primeru. Razčlenjeni spis lahko z lastnimi programi naprej obdelamo glede na naše potrebe. Za oba razčlenjevalnika je mogoče dobiti tudi izvirne programe napisane v Cju, kar nam omogoča izdelati učinkovitejše (brez vmesne datoteke *.PRS) programe za obdelavo spisov.

Obstaja tudi več splošno dostopnih opisov zvrsti (DTD): *Association of American Publishers*, *Elsevier*, *ISO 12083*, *IBM Information Development document type*, *TEI*, *CALS*, ...

S splošnim označevanjem ločimo zgradbo spisa od njegove oblikovanosti pri prikazih. Tudi za opis oblike je bilo pripravljenih nekaj standardov. Taka sta na primer *DSSSL (Document Style Semantics and Specification Language, ISO10179:1991)* in *FOSI (Formatted Output Specification Instance, CALS)*.

5. Viri

5.1. Pisna gradiva

Charles F. Goldfarb: *The SGML Handbook*. Clarendon Press, Oxford 1990.

Eric Van Herwijnen: *SGML Pratique*. International Thomson Publishing France, Paris 1995 (izpopolnjen prevod v francoščino knjige *Practical SGML*, Kluwer AP, 1994).

Ian S. Graham: *HTML Sourcebook*. John Wiley, New York, 1995.

Guidelines for Electronic Text Encoding and Interchange, C.M. Sperberg-McQueen, Lou Burnard, eds., 16. maj 1994:
<http://www.uic.edu/orgs/tei/info/guide.html>

Getting Started with SGML

<http://www.arbortext.com/wp.html>

SGML Users' Group History

<http://www.sil.org/sgml/sgmlhist0.html>

5.2. Naslovi na Internetu

SGML Web Page

<http://www.sil.org/sgml/sgml.html>;

Department of Informatics, University of Oslo, *SGML Repository*:

<ftp://ftp.ifi.uio.no/pub/SGML>;

The <SGML> Project at Exeter Home Page

<http://www.ex.ac.uk/SGML/newsgml.html>;

FTP: <ftp://info.ex.ac.uk/pub/SGML>;

Computer Science Department, Technical University of Darmstadt, *SGML Archive*

<ftp://ftp.th-darmstadt.de/pub/text/sgml>;

James Clark's Home Page;

<http://www.jclark.com/>

in njegovi *SGML Parsers (SGMLS, SP)*:

<ftp://ftp.jclark.com/pub>;

TEI - Text Encoding Initiative Home Page

<http://www.uic.edu/orgs/tei/>;

FTP: <ftp://ftp-tei.uic.edu/pub/tei/>;

CALS / navy

<http://navysgml.dt.navy.mil/cals.html>;

glej še The NTIS CALS Information Center

<http://www.fedworld.gov/edicals/calsinf2.html>;

Oxford Text Archive:

<ftp://ota.ox.ac.uk/pub/ota>;

BNC - British National Corpus Project Text Archive:

<http://info.ox.ac.uk/bnc/index.html>;

Project Gutenberg Home Page:

<http://jg.cso.uiuc.edu/PG/>;

IADS - Interactive Authoring and Display System:

ftp://ftp.ifi.uio.no/pub/SGML/IADS/v2_0/

SoftQuad Inc. Home Page:

<http://www.sq.com/>.

5.3. Slovarček

attribute	lastnost
document	spis
element	gradnik, značka
font	pisava
entity	delec
markup	označevanje
parser	razčlenjevalnik
style	slog
tag	značka
text	besedilo, znakovni
type	zvrst