

MASARYKOVA UNIVERZITA  
FAKULTA INFORMATIKY



# DVISVG

DIPLOMOVÁ PRÁCE

**Rudolf Sabo**

Brno, jaro 2004

## **Prohlášení**

Prohlašuji, že tato diplomová práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

**Vedoucí práce:** RNDr. Petr Sojka

## **Poděkování**

Chcel by som poďakovať vedúcemu diplomovej práce RNDr. Petrovi Sojkovi za poskytnuté konzultácie a svojmu otcovi za podporu počas štúdia.

## Shrnutí

Práca pojednáva o možnostiach výstupu do formátu SVG v sádzacom systéme  $\text{T}_{\text{E}}\text{X}$ . Zavedené a vysvetlené sú štandardy W3C konzorcia, formáty súvisiace s  $\text{T}_{\text{E}}\text{X}$ om a štandardy firmy Adobe. Skúma sa podpora priameho výstupu vo  $\text{V}_{\text{T}}\text{E}_{\text{X}}$ u, konverzia cez DVI v  $\text{B}_{\text{a}}\text{K}_{\text{o}}\text{M}_{\text{a}}\text{T}_{\text{E}}\text{X}$ u a  $\text{dvi2svg}$ , konverzia cez PostScript a PDF v  $\text{pstoedit}$  a  $\text{pdf2svg}$ . Súčasťou práce je vlastná implementácia programu  $\text{DVISVG}$  pre konverziu z DVI do SVG

## **Klíčová slova**

DVI, SVG, TeX, LaTeX

## Obsah

1	Úvod	1
2	W3C	2
2.1	Špecifikácia XML	2
2.1.1	XML Dokument	2
2.1.2	Logická štruktúra	3
2.1.3	Fyzická štruktúra	4
2.1.4	Gramatika	4
2.2	Špecifikácie rozširujúce možnosti XML	4
2.2.1	XML Base	5
2.2.2	Priestor mien: XML Namespace	5
2.2.3	Adresovanie a linkovanie: Xlink, XPointer, XPath	6
2.3	SVG	7
2.3.1	SVG dokument	7
2.3.2	Text	9
2.3.3	Fonty	10
2.3.4	Údaje pre grafickú cestu	11
2.3.5	Dátové typy	12
2.3.6	Súradnicový systém a transformácie	12
2.4	CSS2	14
2.4.1	Špecifikácia fontu	14
2.4.2	Výber a popis fontu	14
3	TEX	16
3.1	DVI	16
3.2	TFM	17
4	Adobe	19
4.1	PostScript	19
4.2	Type 1	20
4.3	Type 2, CFF, CEF	21
4.3.1	Type 2	21
4.3.2	CFF	21
4.3.3	CEF	22
5	Priamy výstup: V <sub>T</sub> E <sub>X</sub>	23
6	Konverzia cez DVI: BaKoMa <sub>T</sub> E <sub>X</sub> a dvi2svg	26
6.1	BaKoMa <sub>T</sub> E <sub>X</sub>	26
6.2	dvi2svg	27
7	Konverzia cez PS: pstoe <sub>D</sub> it a PDF: pdf2svg	30
7.1	pstoe <sub>D</sub> it	30
7.2	pdf2svg	30
8	DVISVG	31

8.1	Implementácia . . . . .	31
8.2	Integrácia . . . . .	33
8.2.1	teTeX . . . . .	33
8.2.2	Kpathsea . . . . .	33
8.2.3	GhostScript . . . . .	34
8.2.4	font2svg . . . . .	34
8.3	Testovanie . . . . .	35
9	<b>Záver</b> . . . . .	40
10	<b>Skratky</b> . . . . .	41

## Kapitola 1

### Úvod

Práca pojednáva o možnostiach výstupu do formátu SVG v sádzacom systéme  $\text{T}_{\text{E}}\text{X}$ . Je možné zvoliť tri spôsoby:

- priamy výstup do SVG,
- prevod z DVI do SVG,
- prevod cez PostScript a PDF do SVG.

Priamy výstup do SVG má produkt firmy Micropress [34]  $\text{V}_{\text{T}}\text{E}_{\text{X}}$ , na prevod z DVI do SVG existuje  $\text{BaKoMa}_{\text{T}}\text{E}_{\text{X}}$  [28] a `dvi2svg` [27]. Konverzia cez PostScript je možná pomocou `pstoedit` [33] a cez PDF pomocou `pdf2svg` od firiem Pdftron [30], Savage Software [31] a otvorená implementácia od Solid Code [32].

Zavedené a vysvetlené budú pojmy:

- XML, XML Namespace, XLink, XPointer, XPath, SVG, CSS2 od W3C konzorcia [35],
- formáty súvisiace s  $\text{T}_{\text{E}}\text{X}$ om ako DVI, TFM
- a tiež PostScript, Type 1, Type 2, CFF, CEF firmy Adobe [24].

Skúmané budú:

- podpora priameho výstupu vo  $\text{V}_{\text{T}}\text{E}_{\text{X}}$ u kap. 5,
- konverzia cez DVI v  $\text{BaKoMa}_{\text{T}}\text{E}_{\text{X}}$ u kap. 6.1 a `dvi2svg` kap. 6.2,
- konverzia cez PostScript a PDF v `pstoedit` kap. 7.1 a `pdf2svg` kap. 7.2.

Súčasťou práce je ďalej vlastná implementácia konverzie DVI do SVG. Program má názov `DVISVG` a je popísaný v kap. 8 tejto práce.



## Kapitola 2

### W3C

Podľa [36] XML špecifikácia nepredpisuje ani neobmedzuje, ktorý grafický formát používať. SVG kap. 2.3 je vektorový grafický formát napísaný v XML a štylovateľný pomocou CSS kap. 2.4. SVG samotné môže obsahovať rastrové obrázky ako PNG a JPEG. Na poskytnutie linkovej funkcionality používa SVG XLink a XPointer kap. 2.2.3.

## 2.1 Špecifikácia XML

### 2.1.1 XML Dokument

Podľa [23] je dátový objekt XML dokument, ak je dobre utvorený a môže byť navyše validný ak spĺňa isté ďalšie podmienky.

Každý XML dokument `document` má logickú a fyzickú štruktúru. Logicky je dokument zložený z deklarácií, elementov, komentárov, odkazov na znaky a inštrukcií na spracovanie (PI). Fyzicky je tvorený jednotkami tzv. entitami. Entita môže odkazovať na inú entitu, čo spôsobí jej vloženie do dokumentu.

Začneme počiatočným symbolom gramatiky `document`.

```
document      ::=  prolog element Misc*

prolog        ::=  XMLDecl? Misc* (doctypedecl Misc*)?
Misc          ::=  Comment | PI | S
```

Kde `XMLDecl` a `doctypedecl` je možné prepisovať podľa.

```
XMLDecl       ::=  '<?xml' VersionInfo EncodingDecl?
                   SDDDecl? S? '?>'
doctypedecl   ::=  '<!DOCTYPE' S Name (S ExternalID)?
                   S? ('[' intSubset ']' S?)? '>'
```

V `doctypedecl` teda deklarujeme typy buď externe pomocou `ExternalID` alebo interne pomocou `intSubset`.

```
ExternalID    ::=  'SYSTEM' S SystemLiteral
                   | 'PUBLIC' S PubidLiteral S SystemLiteral
intSubset     ::=  (markupdecl | DeclSep)*

markupdecl    ::=  elementdecl | AttlistDecl | EntityDecl
                   | NotationDecl | PI | Comment
```

```
DeclSep ::= PEReference | S
PEReference ::= '%' Name ';'

```

### 2.1.2 Logická štruktúra

Každý deklarovaný element `elementdecl` má typ a je identifikovaný menom. Može mať špecifikovanú množinu atribútov `AttlistDecl`. Každá špecifikácia atribútu `AttlistDecl` má meno a hodnotu. Atribúty sú teda použité na asociáciu meno-hodnota s elementami.

```
elementdecl ::= '<!ELEMENT' S Name S contentspec S? '>'
```

Kde `contentspec` môže byť `'EMPTY'`, `'ANY'`, `Mixed` alebo `children`.

```
AttlistDecl ::= '<!ATTLIST' S Name AttDef* S? '>'
AttDef ::= S Name S AttType S DefaultDecl

```

`AttType` môže byť:

```
'#REQUIRED', '#IMPLIED', '#FIXED' 'CDATA', 'ID',
'IDREF', 'IDREFS', 'ENTITY', 'ENTITIES', 'NMTOKEN',
'NMTOKENS', 'NOTATION', Nmtoken

```

Hranice každého elementu `element` sú určené jedným z vymedzovačov `start-tag` a `end-tag`, pre `empty element` máme `empty-element-tag`.

```
element ::= EmptyElemTag | STag content ETag

EmptyElemTag ::= '<' Name (S Attribute)* S? '/>'
STag ::= '<' Name (S Attribute) S? '>'
ETag ::= '</' Name S? '>'

content ::= CharData? ((element | Reference
| CDSect | PI | Comment) CharData?)*

Attribute ::= Name S? '=' S? AttValue

```

Pre obsah elementu `content` máme pravidlá.

```
CDSect ::= CDStart CData CDEnd
CDStart ::= '<![CDATA['
CData ::= (Char* - (Char* ']]>' Char*))
CDEnd ::= ']]>'

Comment ::= '<!--' ((Char - '-')
| ('-' (Char - '-')))* '-->'

```

Zostáva pravidlo pre notácie.

```
NotationDecl ::= '<!NOTATION' S Name
S (ExternalID | PublicID) S? '>'

```

### 2.1.3 Fyzická štruktúra

XML dokument môže pozostávať z viacerých úložných jednotiek. Tie nazývame **entity**. Tieto majú **obsah** a sú, až na niektoré, identifikované entitou **meno**.

Entity sa deklarujú pomocou EntityDecl.

```
EntityDecl ::= GEDecl | PEDecl
GEDecl ::= '<!ENTITY' S Name S EntityDef S? '>'
PEDecl ::= '<!ENTITY' S '%' S Name S PEDef S? '>'
```

Odkaz na entitu je potom možný pomocou Reference.

```
Reference ::= EntityRef | CharRef
EntityRef ::= '&' Name ';'
CharRef ::= '&\#' [0-9]+ ';' | '&\#x' [0-9a-fA-F]+ ';'

```

### 2.1.4 Gramatika

Formálne je gramatika XML daná v špecifikácii použitím EBNF notácie. Každý symbol je definovaný ako `symbol ::= expression`. Počiatkový symbol gramatiky je `document`.

Literály sú označené "string" alebo 'string'.

Výskyt výrazov:

- $A?$  znamená, že výraz  $A$  je prázdny alebo  $A$ ,
- $A+$  znamená jeden alebo viac výskytov  $A$ ,
- $A^*$  znamená nula alebo viac výskytov  $A$ ;
- označenie  $A B$  pre výraz  $A$  nasledovaný  $B$ ,
- $A | B$  pre výraz  $A$  alebo  $B$ ,
- $A - B$  pre ľubovoľný reťazec ktorý zodpovedá  $A$ , ale nezodpovedá  $B$ .

Ďalšie obmedzenia v špecifikácii sú:

- pre dobre utvorený dokument [ wfc: ... ]
- a pre validný dokument [ vc: ... ].

Pravidlá gramatiky neuvedené v tejto práci je možné nájsť v už citovanej špecifikácii.

## 2.2 Špecifikácie rozširujúce možnosti XML

Uvod do problematiky je v [3], špecifikácie pre XML Base [21], XML Namespace [22], XLink [11], XPointer [12], XPath [9].

### 2.2.1 XML Base

XML Base umožňuje pre dokument explicitne špecifikovať základ URI, aby bolo možné používať relatívne odkazy. Zavádza sa atribút `xml:base`.

### 2.2.2 Priestor mien: XML Namespace

V XML Dokumente musia byť mená tagov mien unikátne. Isté rozšírenie prinášajú tzv. priestory mien. Namiesto XML elementov a atribútov sa používajú tzv. kvalifikované mená.

Dobre utvorený dokument v priestore mien je validný ak je validný podľa XML špecifikácie kap. 2.1 a ak pre elementy a atribúty produkt Name zodpovedá produktu NCName.

V gramatika je zmenené pravidlo pre `doctypeDecl`.

```
doctypeDecl ::= '<!DOCTYPE' S QName (S ExternalID)?
              S? ('['
                (markupDecl | PReference | S)*
                ']' S?)? '>'
```

Na logickej úrovni pre elementy a atribúty.

```
elementDecl ::= '<!ELEMENT' S QName S contentspec S? '>'
AttlistDecl ::= '<!ATTLIST' S QName AttDef* S? '>'
AttDef       ::= S (QName | NSAttName)
              S AttType S DefaultDecl
Attribute    ::= NSAttName Eq AttValue | QName Eq AttValue
```

Hranice elementu sú určené nasledovne:

```
STag ::= '<' QName (S Attribute)* S? '>'
ETag ::= '</' QName S? '>'
EmptyElemTag ::= '<' QName (S Attribute)* S? '/>'
```

Nasleduje samotná deklarácia priestoru mien.

```
NSAttName ::= PrefixedAttName | DefaultAttName
PrefixedAttName ::= 'xmlns:' NCName
DefaultAttName ::= 'xmlns'
NCName ::= NCNameStartChar NCNameChar*
NCNameChar ::= NameChar - ':'
NCNameStartChar ::= NameStartChar - ':'
```

Namiesto mien elementov a atribútov typu name máme tzv. kvalifikované mená QName.

```
QName ::= PrefixedName | UnprefixedName
PrefixedName ::= Prefix ':' LocalPart
UnprefixedName ::= LocalPart
Prefix ::= NCName
LocalPart ::= NCName
```

Pravidlá musia spĺňať ďalšie obmedzenia pre priestor mien [NSC: ... ].

### 2.2.3 Adresovanie a linkovanie: Xlink, XPointer, XPath

Linkovanie v XML je možné pomocou troch podporovaných jazykov:

- XLink, ktorý popisuje ako môžu byť linkované XML dokumenty dohromady;
  - href špecifikuje URI spolu s identifikátorom fragmentu;
  - role indikuje význam linkovaných dokumentov;
  - show špecifikuje čo bude urobené s dokumentom po nahraní:  
embed, show, replace, new, undefined;
  - actuate špecifikuje, kedy sa má akcia indikovaná cez show vyskytnúť:  
onLoad, onRequest, undefined;
- XPointer, ktorý umožňuje adresovanie určitých častí v XML dokumente, zavádza tzv. `xpointer(...)` výrazy;
- XPath, ktorý je použitý XPointerom na popis umiestnenia ciest.

## 2.3 SVG

SVG je aplikácia XML kap. 2.1, je kompatibilné s XML Namespace, utilizuje XLink pre URI, potrebuje podporu base URI z XML Base, odkazy ID sú kompatibilné s odkazmi v XPointer kap. 2.2, obsah SVG môže byť štýlovaný pomocou CSS Level 2 kap. 2.4.

SVG je jazyk na popis dvojrozmernej grafiky v XML. V SVG je možné používať tri typy grafických objektov: grafické cesty, obrázky a text. SVG je modularizované do kolekcie abstraktných modulov, ktoré poskytujú špecifické jednotky funkcionality. Vďaka modularizácii je možné v profiloch popísať zoznam potrebných SVG modulov. Profil "Full" je kolekcia všetkých úplných modulov.

Typ MIME je pre SVG podľa [RFC3023] "image/svg+xml". Podľa [15, kap. 1.2, 1.3] má mať SVG súbor koncovku ".svg" alebo ".svgz", ak ide o gzip-komprimovaný súbor.

- SVG Namespace kap. 2.2.2 je: <http://www.w3c.org/2000/svg>,
- identifikátor kap. 2.1.1 Public: "-//W3C//DTD SVG 1.1//EN",
- identifikátor System: <http://www.w3c.org/Graphics/SVG/1.1/DTD/svg11.dtd>.

Aktuálne sú pre SVG špecifikácie: 1.1 [15] ako verzia doporučená a 1.2 [16] ako verzia pracovná. Súvisiace štandarty s SVG sú SVG Mobile [8] a SVG Print [10].

### 2.3.1 SVG dokument

SVG dokument môže byť súčasťou XML dokumentu kap. 2.1.1, alebo môže ísť o samostatný dokument.

SVG fragment môže byť rozsahovo:

- od prázdneho fragmentu,
- cez jednoduchý s grafickými elementami,
- až po komplexný, kde je hlboko zanorená kolekcia kontajnerových a grafických elementov.

SVG fragment môže byť:

- samoobsahujúci súbor alebo zdroj, v tomto prípade ide o SVG dokument,
- alebo vstavaný fragment v rodičovskom XML dokumente kap. 2.1.1.

**K dispozícií sú elementy:**

- 'svg' koreňový element;  
atribúty:
  - xmlns[:prefix] = "resource-name"  
XML Namespace kap. 2.2.2;
  - version = "version";
  - baseProfile = profilename  
popisuje min. SVG jazykový profil, ktorý je nutný pre korektné vygenerovanie obsahu;
  - x = "<coordinate>", y = "<coordinate>"  
x-ová, y-ová súradnica od jedného rohu rovnobežníkovej oblasti v ktorej je vstavaný element umiestnený;
  - width = "<length>", height = "<length>"  
pre najvonkajší element dĺžka a šírka fragmentu; pre vstavaný element dĺžka a šírka rovnobežníkovej oblasti do ktorej je element umiestnený;
- 'g'  
kontajnerový element pre zoskupovanie vzťahujúcich sa grafických elementov, pri použití s elementami 'desc' a 'title' dáva informáciu o štruktúre dokumentu a sémantike, objekt alebo skupina elementov môže byť daná menom použitím atribútu id;
- 'defs'  
kontajnerový element odkazovaných elementov;
- 'desc' a 'title'  
každý kontajnerový element alebo grafický element môže mať popisujúci textový reťazec 'desc' a/alebo 'title';
- 'symbol'  
používa sa na definíciu graf. šablónových objektov, ktoré môžu byť instanciované elementom 'use';
- 'use'  
v SVG môže byť každý šablónový objekt 'svg', 'symbol', 'g', grafický element alebo iný 'use' znovupoužitý - instanciovaný, avšak nemôže odkazovať na celý súbor;  
atribúty:  
podobne ako element 'svg' má atribúty x, y, width, height,  
naviac má atribút xlink:href = "<uri>" kap. 2.2.3;
- 'image'  
indikuje, že obsah celého súboru bude zobrazený do daného štvoruholníka v aktuálnom užívateľskom súradnom systéme, môže ísť o rastrový obrázok ako je PNG alebo JPEG alebo MIME typ "image/svg+xml";  
atribúty ako predchádzajúci element;

- 'switch'  
atribúty:
  - requireFeature = list-of-features;
  - requireExtensions = list-of-extensions;
  - systemLanguage [RFC3066].

### Všeobecné atribúty pre všetky elementy :

- 'id' a 'xml:base';  
id = "name";  
xml:base = "uri" kap. 2.2.1;
- xml:lang a xml:space;  
xml:lang = "languageID";  
xml:space = "{default|preserve}".

Podrobnosti v [15, kap. 5]

### 2.3.2 Text

Text ktorý je vytvorený ako časť SVG fragmentu, je špecifikovaný použitím elementu 'text'. Znaký ktoré sa majú vykresliť, sú vyjadrené ako znakové údaje v XML [23, dodatok B]. Každý element 'text' vytvorí jeden reťazec textu.

Lámanie riadkov a zalamovanie slov je potom riešené:

- predpočítaním riadkových zlomov a viacnásobným použitím elementu 'text',
- alebo predpočítaním riadkových zlomov a použitím jedného elementu 'text' s jedným alebo viac potomkami 'tspan' so zodpovedajúcimi hodnotami atribútov x, y, dx a dy.

Textový obsah je definovaný ako sekvencia XML znakov (character), kde každý znak je definovaný v UNICODE.

Fonty pozostávajú z kolekcie znakov a pridružených informácii, ako sú fontové tabuľky. Grafický znak (glyph) je prezentovateľný tvar znaku, pozostáva z identifikátora a inštrukcií pre vykreslenie. Zobrazenie medzi kolekciou znakov a pridruženými informáciami nemusí byť vždy bijektívne, výnimky sú napr. ligatúry, kompozitné znaky, substitúcie grafických znakov a alternatívne špecifikácie znakov.

SVG obsah vyžaduje precízny výber znakov a grafických znakov. Možnosti pre použiť fontov sú SVG fonty alebo WebFonty.

Kombináciu kolekcie znakov a fontových tabuliek nazývame fontové údaje. Fontové tabuľky obsahujú: údaje o zobrazovaní znakov na grafické znaky, vymedzenie veľkosti znakových oblastí a pozície znakových oblastí.



**K dispozícii sú elementy:**

- 'text' element definuje graf. element obsahujúci text; atribúty:
  - x = "<coordinate>+", y = "<coordinate>+"  
ak je daná jedna súradnica, potom hodnota reprezentuje absolútnu hodnotu x, y-vej súradnice aktuálnej pozície textu; ak je daný zoznam súradníc oddelených medzerou alebo čiarkou, potom hodnota reprezentuje novú x, y-ovú súradnicu pre aktuálnu pozíciu textu pre daný grafický znak v tomto elemente alebo v jeho potomkoch;
  - dx = "<length>+", dy = "<length>+"  
posúva aktuálnu pozíciu v texte pozdĺž x, y-vej osi pre znaky v tomto elemente alebo v jeho potomkoch;
  - rotate = "<number>";
  - textLength = "<length>";
  - lengthAdjust = "spacing|spacingAndGlyphs";
  - transform kap. 2.3.6;
- 'tspan'  
v elemente 'text' je možné upravovať aktuálnu pozíciu textu relatívne alebo absolútne pomocou 'tspan', atribúty ako 'text';
- 'tref'  
textový obsah môže byť: priamo vstavané znakové údaje alebo znakové dáta odkazované elementom 'tref'.

**Vlastnosti:**

- pre výber fontu sú normatívne CSS vlastnosti kap. 2.4.1,
- pre výber medzery: 'kerning', 'letter-spacing', 'word-spacing'.

Podrobnosti v [15, kap. 10]

**2.3.3 Fonty**

SVG utilizuje WebFonty definované v CSS level 2 špecifikácii. Na poskytovanie všeobecného formátu fontov v SVG je možné definovať fonty pomocou tzv. SVG fontov.

**K dispozícii sú elementy:**

- 'font' umožňuje definovať nový SVG font; atribúty:
  - horiz-origin-x = "<number>", horiz-origin-y = "<number>";
  - horiz-adv-x = "<number>";
  - vert-origin-x = "<number>", vert-origin-y = "<number>";
  - vert-adv-y = "<number>";

- 'glyph' umožňuje definovať jednotlivé znaky vo fonte;  
atribúty:

```
- unicode = "<string>";
- glyph-name = "<name[, <name>]*>";
- d = "path data"
  údaje pre grafickú cestu kap. 2.3.4 na stvárnenie grafického znaku;
- orientation = "h|v";
- lang = "%LanguageCodes";
```

ostatné ako atribúty pre 'font';

- 'hkern a 'vkern' umožňujú definovať kerning;  
atribúty:

```
- u1, u2;
- g1, g2;
- k = "<number>";
```

- 'font-face'  
vlastnosti pre daný font ako @font-face v kap. 2.4, avšak zapísané ako atribúty a rozšírené o niekoľko, pre SVG špecifických, atribútov.

Podrobnosti v [15, kap. 20]

### 2.3.4 Údaje pre grafickú cestu

Príkazy:

- moveto  
M (absolútne), m (relatívne);
- closepath  
Z, z;
- lineto  
L (absolútne), l (relatívne);
- curveto  
C, c, S, s  
segment kubickej beziérovej krivky, definovaný je začiatočným, koncovým a dvoma kontrolnými bodmi;  
Q, q, T, t  
segment kvadratickej beziérovej krivky, definovaný je začiatočným, koncovým a jedným kontrolným bodom;  
A, a  
segment eliptického oblúka.

Podrobnosti v [15, kap. 8]

### 2.3.5 Dátové typy

Máme uvedené typy:

- `<integer>` celé číslo je špecifikované voliteľným znakom pre znamienko '+' alebo '-', nasledované jedným alebo viacerými číslicami od 0 do 9;
- `<number>` reálne číslo;
- `<length>` ako `<number>` s jednotkou: `em`, `ex`, `px`, `pt`, `pc`, `mm`, `cm`, `in` a percentá;
- `<coordinate>` ako `<length>`;
- `<angle>` ako `<number>` nasledované jednotkou: `deg`, `grad`, `rad`;
- `<color>` model `rgb(x,y,z)`.

Podrobnosti v [15, kap. 4]

### 2.3.6 Súradnicový systém a transformácie

Pre všetky média, SVG plátno popisuje miesto, kde je SVG obsah vytvorený. Ten je nekonečný pre každý rozmer, avšak konečnú pravouhlu oblasť do ktorej sa obsah vytvorí, nazývame SVG viewport.

Rozmery počiatočného viewportu sú určené pre najvonkajšiu 'svg' element kap. 2.3.1 vyjednáváním. Požadované rozmery môžu byť určené atribútmi `width` a `height`. Pre najvonkajšiu 'svg' element sú určené počiatočné súradnicové systémy: viewportový a užívateľský.

Užívateľský priestor môže byť transformovaný pomocou atribútu `transform`. Transformácie je možné použiť na atribút `viewport` pre elementy 'svg', 'symbol', 'view' alebo na kontajnerový a grafický element.

Matematicky sú transformácie reprezentované transformačnou maticou.

$$\begin{pmatrix} x_{prev} \\ y_{prev} \\ 1 \end{pmatrix} = \begin{pmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{new} \\ y_{new} \\ 1 \end{pmatrix}$$

Keďže je použitých len 6 hodnôt je možné použiť vektor: `[a b c d e f]`.

Pre atribút `transform` je možné použiť hodnoty, ktoré majú dátové typy `<number>`:

- `matrix(<a> <b> <c> <d> <e> <f>);`
- `translate(<tx> [<ty>])` ekv. `matrix(1 0 0 1 tx ty);`
- `scale(<sx> [<sy>])` ekv. `matrix(sx 0 0 sy 0 0);`
- `rotate(<rotate-angle> [<cx> <cy>])`  
ekv. `matrix(cos(a) sin(a) -sin(a) cos(a) 0 0);`

- `skewX(<skew-angle>)` ekviv. `matrix(1 0 tan(a) 1 0 0);`
- `skewY(<skew-angle>)` ekviv. `matrix(1 tan(a) 0 1 0 0).`

Atribút `viewBox` špecifikuje štvoruholník v užívateľskom priestore, ktorý má byť namapovaný do hraníc naviazného viewportu.

Hodnoty pre atribút `viewBox` sú "`<min-x> <min-y> <width> <height>`".

Atribút `preserveAspectRatio` slúži na zachovanie uniformity pri transformáciach pri použití `viewBox` atribútu.

```
preserveAspectRatio = "[defer]<align>[<MeetOrSlice>]";  
<align> bud' none alebo xpppYqqq, kde ppp, qq q može byť Min, Mid, Max;  
<MeetOrSlice>: meet alebo slice.
```

Podrobnosti v [15, kap. 7]

## 2.4 CSS2

Popis práce s fontami v CSS Level 2 je v [6, kap. 15].

### 2.4.1 Špecifikácia fontu

O fonty sa nežiada pomocou jedného mena fontu, ale pomocou nastavenia série fontových vlastností.

Font sa špecifikuje pomocou nasledujúcich charakteristík:

- rodina 'font-family',
- štýl 'font-style',
- varianta 'font-variant',
- váha 'font-weight',
- rozťahnutie 'font-stretch',
- veľkosť 'font-size' a 'font-size-adjust'.

Ako skratku je možné použiť 'font'.

### 2.4.2 Výber a popis fontu

Výber fontu je možný pomocou:

- zhody mena,
- inteligentnej zhody,
- syntézy,
- stiahnutia z Webu.

Popis fontu poskytuje most medzi špecifikáciou fontu a fontovými údajmi. Popis fontu je pridaný do databázy fontov a potom použitý na výber relevantných fontových údajov.

Pre popis fontu sa používa:

```
@font-face { <font-description> }
```

kde popis fontu <font-description> tvoria popisovače:

```
descriptor: value;
...
descriptor: value;
```

Každé @font-face pravidlo špecifikuje hodnotu pre každý popisovač fontu, buď implicitne alebo explicitne.

Popisovače máme pre:

- výber fontu, ako v špecifikácii fontu kap. 2.4.1,
- kvalifikáciu fontových údajov 'unicode-range',
- popis numerických hodnôt 'units-per-em',
- odkazovanie 'src',
- zhodu 'units-per-em', 'panose-1', 'stemv', stemh, slope, cap-height, x-height, ascent, descent,
- syntézu 'widths', 'bbox', 'definition-src',
- zarovnávanie 'baseline', 'centerline', 'mathline', 'topline'.

## Kapitola 3

### TEX

Systém  $\TeX$  vyvynul D. E. Knuth a popísal ho v [17] a [18]. Dobrý úvod do problematiky možno tiež nájsť v domácich publikáciách Petra Olšáka [19] a [20].

#### 3.1 DVI

Binárny formát DVI, čo je skratka pre Device Independent, navrhol D. R. Fuchs a jeho popis možno nájsť v [18, časť 31] alebo v [20, kap. 7.4].

Navrhnutý je tak, aby spĺňal:

- čo najmenšia dĺžka,
- možnosť ľahko a rýchlo čítať ovladačom,
- tvorba  $\TeX$ om a prístup ovladačom sekvenčne.

Informácie sú uložené vo forme povel, parametre, povel, parametre ... Povel má vždy dĺžku jeden byte, je ich definovaných 250; každý má určené koľko parametrov za ním nasleduje, avšak nemusí byť žiadny.

- preambula  
`pre i[1] numerator[4] denominator[4] magnification[4] k[1] x[k]`
- začiatok stránky  
`bop c0[4]c1[4]c2[4]c3[4]c4[4]c5[4]c6[4]c7[4]c8[4]c9[4]p[4]`
- koniec stránky  
`eop`
- postambula  
`post p[4] numerator[4] denominator[4] magnification[4]  
l[4] u[4] s[2] t[2]  
<deklarácia fontov>  
post_post q[4] i[1] <koniec>`
- definícia fontu (i=1..4)  
`fnt_def_i k[1] c[4] s[4] d[4] a[1] l[1] n[a+1]`
- sadzba znaku  
`set_char_i, put_char_i`

- použitie fontu  
fnt\_num\_i
- uloženie stavu sadzby  
push, pop
- posun bodu sadzby  
right\_i, down\_i, w0, w\_i, x0, x\_i, y0, y\_i, z0, z\_i

Aktuálny stav sadzby je určený šiestimi premennými:  $h$ ,  $v$ ,  $w$ ,  $x$ ,  $y$ ,  $z$ . Na začiatku každej strany je  $h$  aj  $v$  nulové a pomocou uvedených príkazov sa tento stav mení. Stav je možné ukladať a vyberať zo zasobníka podľa potreby.

### 3.2 TFM

Popis možno nájsť v [20, kap. 7.3]. Informácie sú v TFM členené do štvorbajtových slov  $w$ . V každom slove  $w = abcd$  je uložené číslo v tvare  $w = 2^4a + 2^{-4}b + b^{-12}c + 2^{-20}d$ . Záporné číslo je uložené ako dvojkový doplnok. Táto hodnota je prenasobená hodnotou `design_size` v jednotke pt.

TFM má isté obmedzenia: pre maximálny počet znakov vo fonte 256 a maximálnu dĺžku TFM súboru  $(2^{16} - 1)w$  teda 256kB .

- prvá sekcia, informácia o veľkosti ostatných sekcií:  
po dvoch bajtoch:  
1w: dĺžka metriky, lh hlavička fontu  
2w: bc, lc - najnižší, najvyšší kód znaku  
dĺžky dátových sekcií:  
3w: nw o šírkach, nh o výškach  
4w: nd o hĺbkach, ni o italicických korekciách  
5w: nl ligačný a kerningový program, nk kerningové páry  
6w: ne vetvenie type extensible, np parametre fontu
- ostatné sekcie:
  - hlavička fontu, dĺžka lh  
1w: kontrolný súčet fontu  
2w: design\_size fontu  
ostatné slová sú nepovinné
  - znaky vo fonte, dĺžka  $ec - bc + 1$   
kódy znakov medzi bc a ec  
1B: odkaz na šírku  
2B: odkaz na výšku (4 bity) a hĺbku (4 bity)  
3B: odkaz na it. korekciu (6 bitov) a tag (2 bity)  
ak je tag=0 remainder nie je použitý  
4B: remainder (8 bitov)  
pre všetky nepoužité znaky majú v odkazoch nulové hodnoty
  - šírky znakov, dĺžka nw
  - výšky znakov, dĺžka nh



- hĺbky znakov, dĺžka nd
- italické korekcie, dĺžka ni  
všetky štyri predchádzajúce začínajú nulovým údajom.  
nulový odkaz znamená, že príslušný rozmer je nulový  
organizované su ako  $w$ .
- ligačný a kerningový program, dĺžka nl
- kerningové páry, dĺžka nk
- vetvenie typu extensible, dĺžka ne
- parametre fontu, fontdimen 1 až 6 v  $\TeX$ u, dĺžka np

## Kapitola 4

### Adobe

#### 4.1 PostScript

Podľa [13] je PostScript programovací jazyk na popis obsahu grafickej stránky. Pri programovaní sa používa tzv. postfixová notácia t.j. nad objektami uloženými na zásobníku je možné vykonávať pomocou operátorov isté akcie.

Všetky položky v zásobníku je možné zmazať `clear`, vrchnú položku duplikovať `dup`, zameniť vrchné dve položky `exch` a vybrať vrchnú položku `pop`. Aritmetické operátory sú napr. `add`, `sub` pre sčítanie a odčítanie a `mul`, `div` pre násobenie a delenie. Interaktívnou obdobou `clear` je `pstack` a pre `pop` zase `==`. V podmienkach `if` sa používajú operátory pre relácie napr. `eq`, `gt`. Cykly sa realizujú pomocou operátora `for`.

Premenné v jazyku PostScript majú prefix `'/'` a definujú sa operátorom `def`. Reťazce sú uzavreté medzi `'(, ')` a ich grafické stvárnenie je možné vykresliť pomocou `show`; číslo sa dá previesť na reťazec operátorom `cvs`. Polia sú identifikované pomocou otváracieho operátora `'['` a operátora pre ukončenie `']'`. Nového pole sa vytvára operátorom `array` a k jednotlivým prvkom poľa sa pristupuje pomocou operátorov `aload` a `astore`. Polymorfne operácie sa vykonávajú pomocou `forall`, `get` a `put`. V jazyku PostScript je možné vytvoriť slovník pomocou operátorov `dict`, `begin` a `end`.

Font je možné zvoliť pomocou:

```
/meno-fontu findfont  
vel'kost'-fontu scalefont  
setfont
```

S každým fontom je asociovaná transformačná matica, ktorú je možné zmeniť uložením poľa `[a b c d e f]` na zásobník a použitím, namiesto `scalefont`, operátora `makefont`.

Nová grafická cesta sa vytvorí operátorom `newpath` a uzavrie operátorom `closepath`.

Grafická cesta sa potom skonštruováva pomocou operátorov:

- `moveto`, `rmoveto` pre posun;
- `lineto`, `rlineto` pre čiaru;
- `rcurveto` pre beziérove krivky.

Grafický stav je možné ukladať operátorom `gsave` a vyberať operátorom `grestore`. Grafické stvárnenie sa vykreslí operátormi `stroke` a `fill`, podľa toho či chceme len krivky alebo aj vyplniť vymedzenú oblasť.

Na transformácie súradnicového systému slúžia operátory `rotate`, `scale`, `translate`.

Samotná grafická stránka sa nakoniec zobrazí operátorom `showpage`.

## 4.2 Type 1

Podľa [7, kap. 2] je Type 1 font v podstate fontový program, ktorý je organizovaný ako kolekcia procedúr popisujúca tvary jednotlivých znakov. K elementom tejto kolekcie sa pristupuje operátorom `show` kap. 4.1. Type 1 fontový program používa internú procedúru PS interpretu `BuildChar`. Tá začína pri použití indexu v poli `Encoding` fontového slovníka pri získaní mena znaku ktorý má byť vybudovaný.

Fontový program je organizovaný nasledovným spôsobom:

```
%!FontType1-1.0
...
eexec
/Private dictionary
...
/OtherSubrs
/Subrs 43 array
dup 0 155 RD ~15~binárnych~bajtov~ ND
...
/CharString 190 dict dup begin
/.notdef 9 RD ~9~binárnych~bajtov~ ND
/A 186 RD ~186~binárnych~bajtov~ ND
...
end end readonly put noaccess put
dup /FontName get exch definefont pop
mark currentfile closefile
000000000000000000
...
000000000000000000
cleartomark
```

Podľa [7, kap. 6] slovník `CharString` obsahuje kolekciu párov meno-procedúra. Procedúry na ktoré tieto mená odkazujú produkujú grafické stvárnenie znakov.

Príkazy pre znakový reťazec `CharString`:

- začiatok a koniec grafického stvárnenia
  - `hsbw`, `sbw` začína každý znak,
  - `seac` pre štandardné kódovanie akcentovaného znaku,
  - `endchar` ukončuje znak;
- konštrukciu grafickej cesty
  - `rmoveto` ako v PS, varianty `hmoveto`, `vmoveto`,

- rlineto ako v PS, varianty hlineto, vlineto,
- rrcurveto ako rcurveto v PS ale relatívne, varianty hvcurveto, vhcurveto,
- closepath;
- hintovanie
  - hstem, htstem3, vstem, vstem3, dotsection;
- aritmetiku
  - div ako v PS;
- podprogramy
  - setcurrentpoint pre nastavenie aktuálneho bodu,
  - callsubr volá znakovoreťazcový podprogram v poli Subrs,
  - return vráti z poľa Subrs znakovoreťazcový podprogram,
  - callothersubr volá PS interpreter,
  - pop vyberie číslo z PS interpreteru a uloží ho do Type 1 BuildChar operačného zásobníka.

Ide v podstate o upravenú sadu príkazov v jazyku PostScript kap. 4.1.

### 4.3 Type 2, CFF, CEF

#### 4.3.1 Type 2

Formát Type 2 [2] poskytuje metódu pre kompaktné kódovanie procedúr pre znak (glyph) vo fontovom programe popisujúcom font. Type 2 znakové reťazce musia byť použité s CFF alebo s OpenType fontovým súborom na vytvorenie úplného fontového programu.

Znakové reťazce Type 2 sú založené na konceptoch Type 1. Pri porovnaní s Type 1 formátom, kódovanie Type 2 poskytuje menšiu veľkosť a lepší výkon. Operátory v Type 2 sú nadmnožinou operátorov v Type 1.

#### 4.3.2 CFF

Compact Font Formát [1] skátene CFF je formát fontu pre kompaktnú reprezentáciu Type 1 fontov alebo CID kľúčovaných fontov.

Formát CFF umožňuje uloženie viacerých fontov do jednej jednotky pomenovanej FontSet. Šetrenie priestorom je výsledkom kompaktnej binárnej reprezentácie pre väčšinu informácií a zdieľania všeobecných údajov medzi fontami. Formát CFF je navrhnutý pre konjukciu s Type 2 znakovými reťazcami procedúr popisujúcich fonty.

Konceptuálne sú binárne údaje organizované ako čísla oddelených dátových štruktúr.

Rozloženie ukazuje nasledujúca tabuľka.

Položka	Komentár
Header	-
Name INDEX	-
Top DICT INDEX	-

String INDEX -  
Global Subr INDEX -  
Encodings -  
Charsets -  
FDSelect len pre CID Fonty  
CharStrings INDEX pre každý font  
Font DICT INDEX pre každý font, len pre CID Fonty  
Private DICT pre každý font  
Local Subr INDEX pre každý font  
alebo pre každý Private DICT pre CID Fonty

### 4.3.3 CEF

Compact Embedded Format, skrátene CEF, je varianta CFF fontov kap. 4.3.2, ktoré podporuje Adobe SVG Viewer. Špecifikácia nie je momentálne dostupná.

## Kapitola 5

### Priamy výstup: V<sub>T</sub>E<sub>X</sub>

Existuje pre OS Windows, Linux a OS/2 a od verzie 8.x má, okrem podpory priameho výstupu do DVI, PDF, PostScript, HTML, navyiac aj výstup do formátu SVG. Avšak momentálne je podpora SVG len vo verzii pre OS Windows. V<sub>T</sub>E<sub>X</sub> je produkt firmy MicroPress(R) [34].

Verzia pre OS Windows má navyiac integrované vývojové prostredie IDE, kde je možné zvoliť výstup do SVG "Typeset to SVG" v menu s nasledujúcimi možnosťami.

```
General  TeX File: file.tex          Use Gex: None
                                                Level 1, 2
                                                Fonts 14, 35
                                                [x] Suport Virtual Fonts

Output   Text:   [x] Compresion          Resolution: 128-4096
         Fonts: [x] External
         PaperType          [x] Produce HTML Wrapper
                                                [x] Autosize of SVGimage
```

Tiež je možné využívať verziu spustiteľnú z príkazového riadku `vtex3h`, pre výstup do `svg` je voľba `-sg`. Formát sa volí pomocou `@format`. Podrobnejšie informácie o výstupe do SVG v IDE a voľbách `vtex3h` možno nájsť vo V<sub>T</sub>E<sub>X</sub> user guide `ugd.pdf`.

Pre každú stranu súboru s názvom `file.tex` sa vygeneruje jeden súbor vo formáte SVG uvedeným spôsobom: `file-i.svg`, kde `i` je číslo strany.

SVG dokument kap. 2.3.1 začína XML preambulou kap. 2.1.1.

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

Nasleduje poznámka.

```
<!-- Generator: VTeX/W 8.10 -->
```

Ďalej DOCTYPE pre `svg` je typu PUBLIC. Pre odkazy na fonty z dcérijných elementov ko-reňového elementu '`svg`' sa zavádzajú nové entity, ktorých mená sú `sti`, kde `i` je číslo fontu.

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20000303 Stylable//EN"
  "http://www.w3.org/TR/2000/03/WD-SVG-20000303/DTD/
  svg-20000303-stylable.dtd"
[
  <!ENTITY st0 "font-family:'mvr10';font-size:10;">
  <!ENTITY st1 "font-family:'mvr12';font-size:12;">
]>
```

Nasleduje koreňový element 'svg' kap. 2.3.1.

```
<svg width="360pt" height=" 360pt" viewBox="0 0 360 360"
  xml:space="preserve">
```

Pri voľbe Autosize of SVGimage sa do dokumentu pridá element 'script' s odkazom xlink kap. 2.2.3 typu href na súbor file.js.

```
<script type="text/ecmascript" xlink:href="file.js"/>
```

a element 'svg' bude mať naviac nasledujúce atribúty:

```
onload="onload(evt)" onresize="fitToWindow();
```

Štandardne vytvorený súbor file.js potom obsahuje JavaScriptový kód týchto funkcií: onload, getDoc, fitToWindow.

```
function onload(evt) {
  getDoc(evt);
  fitToWindow();
}
```

```
function getDoc(evt){
  doc = evt.getTarget().getOwnerDocument();
}
```

```
function fitToWindow() {
  doc.rootElement.setAttribute('width', window.innerWidth);
  doc.rootElement.setAttribute('height', window.innerHeight);
}
```

Následujú fonty. Používajú sa tzv. WebFonty kap. 2.3.3 vo formáte CEF kap. 4.3.3. Pri použití voľby Fonts: [ ] External, sú fonty súčasťou SVG dokumentu a sú kódované pomocou base64 [RFC3548].

```
<style type="text/css">
  <![CDATA[
    @font-face{font-family:'mvr10';src:url("data:;base64,\
      .....skrátené.....")}
    @font-face{font-family:'mvr12';src:url("data:;base64,\
      .....skrátené.....")}
  ]]>
</style>
```

Pri použití voľby Fonts: [x] External sa vytvoria fonty vo formáte CEF kap. 4.3.3 ako súbory file-i.cef, kde i je číslo fontu.

```
<style type="text/css">
<![CDATA[
  @font-face{font-family:'mvr10';src:url(./file-4.cef)}
  @font-face{font-family:'mvr12';src:url(./file-8.cef)}
]]>
```

Ďalej nasleduje element 'g' pre danú stranu Pagei. Ktorý obsahuje text samotný. Jeho poloha je určená pomocou elementov text a tspan.

```
<g transform="matrix(1 0 0 1 72 72)" id="Pagei"
  style="fill-rule:nonzero;
    clip-rule:nonzero;
    stroke-miterlimit:10;
    stroke-width:0.1;">
<text transform="matrix(1 0 0 1 76.71 62.76)"
  stroke="none;" fill="#000000"
  style="&st0;">test</text>
<text transform="matrix(1 0 0 1 61.77 74.72)"
  stroke="none;" fill="#000000" style="&st1;">
  <tspan>test</tspan>
  <tspan x=" 169.36" y=" 555.92">1</tspan></text>
</g>
```

Pri použití voľby Produce HTML Wrapper sa vytvoria súbory file-i.htm, kde i je číslo strany, pre každý SVG súbor. V každom HTML súbore je potom pomocou elementu 'embed' vytvorený odkaz na daný SVG dokument.

```
<embed src="file-i.svg" name="SVGEmbed"
  height="100%" width="100%"
  type="image/svg+xml"
  pluginspage="http://www.adobe.com/svg/viewer/install/">
```

V L<sup>A</sup>T<sub>E</sub>Xu je možné možné využívať v SVG dokumentoch tzv. navigáciu pomocou balíčkov svginav a svgenav.

Pomocou balíčka svgenav je možné vytvoriť externú navigáciu. Vzniknú tak naviac štyri súbory file-{b,l,r,t}.htm, ktoré túto navigáciu realizujú pomocou HTML stránok.

Pomocou balíčka svginav sa upraví obsah SVG dokumentov pridaním internej navigácie. Tá je realizovaná pomocou panelu id="navpanel" s položkami First, Previous, Next, Last a odkazmi xlink kap. 2.2.3 typu href na súbor "file-i.svg".

Pri použití voľby Use GeX: Level 1, 2 Fonts 14, 35, je možné používať integrovaný interpret PostScriptu, ktorý pridáva tzv. Graphics Extensions skratene GeX.



## Kapitola 6

### Konverzia cez DVI: BaKoMaTeX a dvi2svg

#### 6.1 BaKoMaTeX

Autorom je Basil K. Malyshev, ktorý distribuje BaKoMaTeX ako shareware a je ho možné stiahnuť na [28]. Existuje vo verzii, narozdiel od VTeXu kap. 5, len pre OS Windows a od verzie 6.x má podporu výstupu do SVG.

Po vygenerovaní DVI sa vytvoria dva súbory s koncovkami `.dvi` a `.dvv`. Tie sa líšia v postambule kap. 3.1 a v čísle `k` priradenom vo `fnt_def_1` a tiež pri použití vo `fnt_num_1`.

V DView po zvolení výstupu do SVG "Generating SVG" je možné zvoliť nasledujúce voľby.

Page Range: All, OneCurrent, Specify by Range

Compresion: None, Quick, Best

Use Frames [x]

Adding Navigation Bar to: Left Top, Right Top,

Left Bottom, Right Bottom

Margins[...][...]      Panel Color [...]

Height[...]            Button Color [...]

Pre každú stranu sa vytvoria pre DVI súbor `name.dvi`, v podadresáry `svg`, súbory podľa nasledujúcej schémy: `name.svg` pre prvú, pre ostatné `name_pi.svg`, kde `i` je  $(i-1)$ -vá stránka.

SVG dokument kap. 2.3.1 začína XML preambulou kap. 2.1.1.

```
<?xml version="1.0" standalone="no"?>
```

Ďalej DOCTYPE pre `svg` je typu PUBLIC.

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN"
"http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
```

Nasleduje koreňový element '`svg`' kap. 2.1.1. Oproti VTeXu kap. 5 a `dvi2svg` kap. 6.2 je tu definovaný aj priestor mien XML Namespace kap. 2.2.2.

```
<svg width="100%" height="100%" viewBox="0 0 595 842"
xmlns="http://www.w3.org/2000/svg"
xmlns:a="http://www.adobe.com/svg10-extensions">
```

Fonty sú definované ako SVG Fonty kap. 2.3.3. Element 'defs' obsahuje definíciu fontov pomocou elementov 'font' a 'font-face'. Atribút `units-per-em` je, ako je bežné, nastavený pre Type 1 fonty na 1000 a pre TrueType na 2048.

```
<font horiz-adv-x="1000">
  <font-face font-family="BKM-cmr10"
            font-weight="400" font-style="normal"
            units-per-em="1000">
  </font-face>
  ...
</font>
```

Jednotlivé znaky v elemente 'font' sú definované pomocou elementu 'glyph'. Peknou vlastnosťou je, že do SVG sú zaradené len tie znaky, ktoré sú použité v DVI.

```
<glyph unicode="1" horiz-adv-x="500" d="....."/>
```

Nasleduje text samotný.

```
<text>
  <tspan fill="#000000"
        font-family="BKM-cmr10" font-weight="400"
        font-style="normal" font-size="9.9632">
  <tspan x="148.72" y="134.77">test</tspan>
  <tspan x="303.15" y="702.67">1</tspan>
  </tspan>
</text>
```

V L<sup>A</sup>T<sub>E</sub>Xu je podpora balíčkov prosper, pdfscreen, pdfslide. Špecifický je aj SVG.STY, ktorý je primárne určený pre SVG.

DView má tiež vstavaný, podobne ako V<sub>T</sub>E<sub>X</sub> má GeX, interpret PostScriptu, takže pomocou príkazov special v DVI je možné vytvoriť grafický výstup z PostScriptu do SVG.

## 6.2 dvi2svg

Dvi2svg [27] je naprogramovaný v jazyku Java, takže je do istej miery nezávislý na platforme. Autorom je Adrian Frischauf, ktorý dvi2svg uvoľnil pod licenciou MPL, takže je prístupný aj zdrojový kód. Využíva ANT, SAX a JDOM, čo sú štandardné knižnice pre prácu s XML v prostredí Javy.

Po spustení shellového skriptu dvi2svg s menom DVI súboru name.dvi sa vytvoria súbory namei.svg, kde i je i-tá stránka. Cesty k metrikám TFM sú nastavené v adresáry conf v súbore tfmlist a samotné SVG fonty sú v adresáry svgfonts.

SVG dokument kap. 2.3.1 začína XML preambulou kap. 2.1.1. Oproti V<sub>T</sub>E<sub>X</sub>u kap. 5 a BaKoMa<sub>E</sub>T<sub>E</sub>Xu kap. 6.1 je použité kódovanie UTF-8.

```
<?xml version="1.0" encoding="UTF-8"?>
```

Ďalej DOCTYPE pre svg je typu PUBLIC.

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
  "http://www.w3.org/TR/2001/PR-SVG-20010719/DTD/svg10.dtd">
```

Nasleduje koreňový element 'svg' kap. 2.3.1.

```
<svg width="513" height="728" viewBox="0 0 513 728">
```

Element 'defs' obsahuje definíciu fontov.

```
<font id="CMR">
  <font-face font-family="CMR" />
  ...
</font>
```

Jednotlivé znaky v elemente 'font' sú definované pomocou elementu 'glyph'.

```
<glyph unicode="..." glyph-name="t" horiz-adv-x="388.9">
  <path style="fill-rule=evenodd; stroke:none" d="....." />
</glyph>
```

Nasleduje text samotný. Oproti V<sub>TeX</sub>u kap. 5 a BaKoMa<sub>TeX</sub>u kap. 6.1 je použitý element 'rect'. Pomerne zvláštna je použitá presnosť v horizontálnom smere na šesť desiatinných miest a vo vertikálnom pätnásť. Znak v riadku sú umiestňované jedným elementom 'tspan' pomocou argumentu x, takže sa nevyužíva atribút horiz-adv-x z elementu 'glyph'.

```
<rect x="0" y="0" width="513" height="728"
  stroke="none" fill="none" />
<g>
  <text fill="Black" font-family="CMR" font-size="9.8201065">
    <tspan y="46.2201080"
      x="45.307056 49.129272 53.497505 57.374313">
      ...unicode text...
    </tspan>
  </text>
  <text fill="Black" font-family="CMR" font-size="9.8201065">
    <tspan y="691.599975" x="254.04286">
      ...unicode text...
    </tspan>
  </text>
</g>
```

Dvi2svg využíva SVG fonty, ktoré boli vytvorené podľa [14, kap. 3] pomocou t<sub>1</sub>disas a perlového skriptu t<sub>1</sub>svg.pl. Na prevod Type 1 fontov kap. 4.2 do príkazov pre SVG fonty kap. 2.3.3 v atribúte d bola použitá nasledujúca prevodná tabuľka.

Type 1	SVG
a1 a2 hsbw	horiz-adv-x = a2 M a1 0
a1 hlineto	h a1
a1 hmoveto	m a1 0
a1 a2 a3 a4 hvcurveto	c a1 0 (a1+a2) a3 (a1+a2) (a3+a4)
a1 a2 rlineto	l a1 a2
a1 a2 rmoveto	m a1 a2
a1 a2 a3 a4 a5 a6 rrcurveto	c a1 a3 (a1+a3) (a2+a4) (a1+a3+a5) (a2+a4+a6)
a1 vlineto	v a1
a2 vmoveto	m 0 a1
a1 a2 a3 a4 hcurveto	c 0 a1 a2 (a1+a3) (a2+a4) (a1+a3)
closepath	z

V  $\LaTeX$ u má dvi2svg podporu pre balíčky `graphicx`, `hyperref` a `color`.

## Kapitola 7

### Konverzia cez PS: pstoedit a PDF: pdf2svg

#### 7.1 pstoedit

Dvitoedit [33] prevádza vstup v PostScripte do rôznych výstupných formátov. Pomocou špeciálneho pluginu je možný výstup aj do SVG.

Program nevyužíva SVG fonty, ale všetky znaky sú vygenerované ako grafické cesty. To je možné zvoliť pomocou voľby `-dt`.

Podľa [14, kap. 4.1] je možné v `pstoedit` používať SVG fonty pomocou stylesheetu `ins-saxon.xsl` a skriptu `saxon6.sh`.

```
pstoedit -f svg:-texmode -nfr in.ps out.svg
saxon6.sh out.svg ins-saxon6.xsl out1.svg
```

#### 7.2 pdf2svg

Konverzia cez PDF je možná pomocou implementácií `pdf2svg` konvertorov od firiem Pdftron [30], Savage Software [31] a otvorená implementácia od Solid Code [32].

Prvý z uvedených existuje pre OS Windows aj Linux a je šírený ako shareware. Druhý je len pre OS Windows. Sľubne vyzerá tretí z uvedených, avšak ide o vývojovú verziu.

## Kapitola 8

# DVISVG

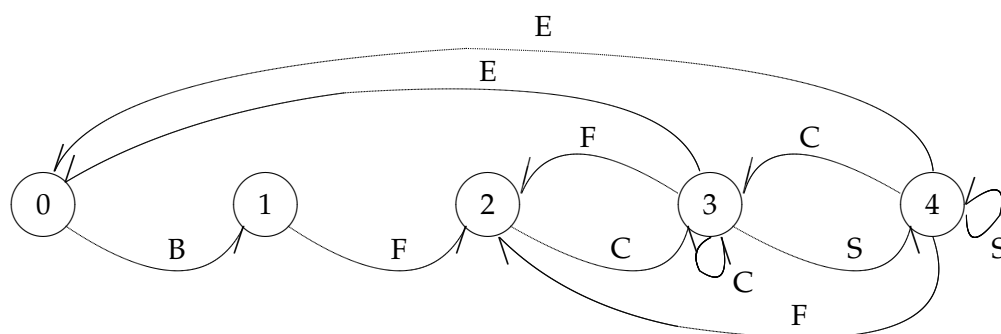
### 8.1 Implementácia

Funkcie sú organizované do sekcií, každá sekcia je samostatný C súbor. Závislosti medzi objektovými súbormi resp. sekciami sú v súbore `Makefile` a spracované sú pomocou `make`.

Základná sekcia je `dvivsvg`. Tá využíva kód z nasledujúcich sekcií:

`dvibase`, `dvipage`, `dvitfm`, `svgout`, `svgfonts`.

Sekcia `dvibase` načíta základné údaje z DVI súboru. V sekcii `dvipage` sa spracovávajú jednotlivé stránky DVI súboru. Využívajú sa pritom metrické informácie z TFM súborov, načítané v `dvitfm` sekcii, použitých fontov v DVI súbore. Výstup do SVG je v sekcii `svgout`. Princíp fungovanie tejto sekcie popíšeme stavovým automatom na nasledujúcom obrázku.



Automat má päť stavov 0 až 4 a päť návěstí B, E, F, C, S.

- B: DVI\_BOP
- E: DVI\_EOP
- F: DVI\_FNT\_NUM
- C: DVI\_SET\_CHAR, DVI\_PUT\_CHAR
- S: DVI\_RIGHT\_i, DVI\_W0, DVI\_W\_i  
DVI\_X0, DVI\_X\_i  
DVI\_DOWN\_i, DVI\_Y0, DVI\_Y\_i  
DVI\_Z0, DVI\_Z\_i

SVG fonty, sú vytvorené kódom v sekcii `svgfonts`.

V jednotlivých sekciách sa využívajú:

- DVI štruktúry zo súboru `dvistr.h` s prefixom `dvi_` a sufixom `_struct`:  
`pre`, `post`, `post_post`, `bop`, `fnt` a `state`.
- DVI príkazy kap. 3.1 zo súboru `dvicom.h` s prefixom `DVI_`.

Súbor `xmlsvg.h` obsahuje elementy XML a SVG definované pomocou symbolických konštánt.

Funkcie `dvibase` a `dvipage` je treba volať s deskriptormi pre vstup `dvi_inp` a výstup `svg_out`, `log_out`.

Pre načítanie metrických informácií pomocou sekcie `dvitfm` je treba urobiť inicializáciu pomocou funkcie `dvitfminit`, načítať metrické informácie `dvitfm` pre daný font s daným číslom `k` a nakoniec uvoľniť alokovanú pamäť pomocou `dvitfmfree`. Sekcia `dvitfm` interne používa štruktúru `tfm_header_struct` a funkcie `tfminit`, `tfmmetr`. Pomocou funkcií `tfmprintD` a `tfmprintF` je možné vypísať načítané metriky.

V sekcii `svgout` zabezpečuje zmenu kódovania funkcia `conv`.

Program `dvsvg` sa spúšťa s príkazovej riadky, kde ako parameter treba uviesť meno DVI súboru `file.dvi` alebo `file`. Po spustení sa štandardne vytvorí SVG výstup do súboru `out.svg` a LOG výstup do `out.log`. Toto je možné zmeniť pomocou volieb `-o` a `-O`.

Pre znaky v jednotlivých fontoch použitých vo vygenerovanom SVG súbore sa vytvorí súbor `fonts.svg`, ktorý obsahuje tzv. SVG fonty kap. 2.3.3. Meno `font.svg` je možné zmeniť pomocou `-f`.

Ako každý program GNU, `dvsvg` tiež pozná voľby `--version` a `--help`.

K programu bola vytvorená manuálová stránka `dvsvg.1`, ktorá obsahuje stručné zoznamenie s DVISVG a popis uvedených a tiež ďalších volieb.

## 8.2 Integrácia

### 8.2.1 teTeX

Ide o voľne šíriteľnú distribúciu TeX-u pre OS UNIX a Linux. Distribúcia zahŕňa 3 súbory: `teTeX-xxx-ver.tar.gz` kde `xxx` je `src`, `texmf`, `texmfsrc` a `ver` je verzia. V čase písania tejto práce bola aktuálna verzia 2.0. Distribúcia zahŕňa `tex`, `pdftex`, `e-tex`, `omega` s priamym výstupom pre `tex`, `omega` do DVI, `pdftex`, `e-tex` do PDF. a DVIér: `dvihp(k)`, `dvips(k)`, `dvipdfm`, `xdvi(k)`, kde `(k)` znamená, že sa využívajú služby knižnice Kpathsea kap. 8.2.2.

Na zostavenie programu treba najskôr spustiť skript `configure`. Pre `configure` je možné zmeniť prefix pre inštalčné cesty pomocou `--prefix`. Pre nastavenie použitého prekladača pre zdrojové kódy napísané v C alebo C++ sú premenné `CC`, `CXX` a pre voľby `CFLAGS`. Pre `make` potom parametre `all`, `install`, `strip`. Po nainštalovaní `install` sa do adresárov `include` a `lib` nahrajú hlavičkové súbory a statická knižnica Kpathsea. Adresáre `info` a `man` potom obsahujú dokumentáciu.

### 8.2.2 Kpathsea

Knižnica funkcií primárne určená na vyhľadávanie súborov v stromovej štruktúre `texmf`. V distribúcii teTeX-u ju využívajú všetky programy z Web2c napr. `tex`, `mf`, `dvitype` alebo `omega`. Ich zdrojový kód je vo `.web` súboroch, kde podporu pre Kpathsea pridávajú zmenové súbory `.ch`. Ostatné programy sú napísané v jazyku C. Medzi nich patria: `dvips`, `dvilj`, `xdvi`, ktoré sú upravené pre použitie s Kpathsea a v názve resp. verzii majú príponu `k` resp. `(k)`.

Podľa [4] je pre použitie Kpathsea vo svojich programoch doporučený uvedený postup.

```
#include kpathsea/kpathsea.h
kpse_set_program_name(argv[0], NULL)
kpse_init_prog(prefix, dpi, mode, fallback)

#include kpathsea/tex-file.h
kpse_find_FORMAT, kpse_find_tfm

#include kpathsea/tex-glyph.h
kpse_find_pk, gf, glyph

FILE *kpse_open_file("name")
```

Pomocou `kpse_set_program_enabled`, je možné nastaviť automatické generovanie metrik. Ako parametre treba zvoliť `kpse_tfm_format`, `MAKE_TEX_TFM_BY_DEFAULT` a `kpse_src_compile`. Na začiatku potom treba vložiť hlavičkový súbor `c-auto.h`.

Ľadiace informácie je možné zapnúť nastavením premennej `KPATHSEA_DEBUG` na hodnotu `-1`.



### 8.2.3 GhostScript

Ghostscript [25] je voľná implementácia interpreteru jazyka PostScript, ktorého stručný popis možno nájsť v kap. 4.1 tejto práce.

Interpreter sa spúšťa príkazom `gs` s parametrom `meno.ps`, kde `meno.ps` je program v jazyku PostScript resp. v jazyku Ghostscript.

Na príkazovom riadku je možné definovať premenné, s ktorými potom program v jazyku PostScript pracuje a to pomocou argumentov `-dname` alebo `-dname=token` a pre reťazce pomocou argumentu `-sname=string`.

Pre neinteraktívne spracovanie dávok treba nastaviť pre interpreter PostScriptu argumenty `-DNODIPLAY`, `-dBATC`H, `-dNOPAUSE`, `-dQUIT` a argument `-q`.

### 8.2.4 font2svg

Výhodné je použiť z `font2svg` [29] program v PostScripte kap. 4.1 `font2svg.ps`, ktorý načíta font v Type 1 kap. 4.2 a vytvorí SVG font kap. 2.3.3. Spúšťame ho pomocou interpreteru Ghostscript kap. 8.2.3 s parametrom `font2svg.ps` a uvedenými premennými, ktoré nastavíme podľa požiadaviek na font, ktorý sa má vygenerovať.

premenná	typ	význam
FONTFILE	s	meno súboru Type 1 fontu
THEFONT	d	meno Type 1 fontu
ENCODING	s	kódovanie, napr. Default
ENCFILE	s	súbor s kódovaním, štandardne obsahuje Default
SYMBOLS	d	znaky ktoré sa majú do SVG fontov vygenerovať
DVIENC	s	v DVISVG nepoužité
DVIPSFILE	s	v DVISVG nepoužité
HOME	s	cesta k <code>font2svg.ps</code>

### 8.3 Testovanie

Na testovanie použijeme DVI súbor `test.dvi`.

Jeho obsah, vytvorený programom `DVItype`, je nasledovný:

```
beginning of page 1
push 88: down3 -917504
pop
down4 42152922
push
down4 -41497562
push [ ]
fntdef1 0: cmr10---loaded at size 655360 DVI units
fntnum0 [test]
pop
y3 786432
push [ ]
fntdef1 36: cmti10---loaded at size 655360 DVI units
fntnum36 [test]
pop
y0 786432
push [ ]
fntdef1 23: cmbx10---loaded at size 655360 DVI units
fntnum23 [test]
pop
pop
down3 1572864
push [ ]
fntnum0 [1]
pop
eop
```

```
beginning of page 2
... skrátene ...
eop
```

```
Postamble starts at byte 349.
maxv=43725786, maxh=30785863, maxstackdepth=2, totalpages=2
Font 36: cmti10
Font 23: cmbx10
Font 0: cmr10
```

Zo súboru `test.dvi` dostaneme pomocou programu `dvisvg` nasledujúci výstup. Použijeme argumenty `-o test.svg -O test.log` a parameter `test`.

```
SVG output is now: test.svg
LOG output is now: test.log
Page: 1
This is DVISVG version 0.4
Opening files: test.dvi (r) test.log test.svg (w).
' TeX output 2004.02.17:1517'
Fonts:***
Pages:...
SVG Fonts:***
```

Ako vidieť, pri nezadaní koncovky `.dvi` si ju program DVISVG sám doplní a pri nezadaní čísla strany sa vytvorí strana číslo jeden.

SVG súbor `test.svg` obsahuje:

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<!-- Generator: DVISVG version 0.4 -->
<!DOCTYPE svg SYSTEM "-//W3C//DTD SVG 1.1//EN" [
  <!ENTITY st0 "font-family:cmr10;font-size:10.00pt;">
  <!ENTITY st23 "font-family:cmbx10;font-size:10.00pt;">
  <!ENTITY st36 "font-family:cmti10;font-size:10.00pt;">
]>
<svg width="542.02pt" height="739.47pt" version="1.1"
  viewBox="0 0 542.02pt 739.47pt"
  xmlns="http://www.w3c.org/2000/svg"
  xml:space="preserve">
<style type="text/css">
<![CDATA[
  @font-face{font-family:'cmr10';font-size:'10.00pt';
    src:url(fonts.svg#cmr10)}
  @font-face{font-family:'cmbx10';font-size:'10.00pt';
    src:url(fonts.svg#cmbx10)}
  @font-face{font-family:'cmti10';font-size:'10.00pt';
    src:url(fonts.svg#cmti10)}
]]>
</style>
<text id="page1">
<tspan style="&st0;">
<tspan x="92.27pt" y="82.27pt">test</tspan>
</tspan>
<tspan style="&st36;">
<tspan x="92.27pt" y="94.27pt">test</tspan>
</tspan>
<tspan style="&st23;">
<tspan x="92.27pt" y="106.27pt">test</tspan>
```

```

</tspan>
<tspan style="&st0;">
<tspan x="304.65pt" y="739.47pt">1</tspan>
</tspan>
</text>
</svg>

```

Súbor `fonts.svg` obsahuje použité fonty v SVG dokumente.

```

<?xml version="1.0" encoding="utf8"?>
<svg>
<defs>
<font id="cmr10"
<font-face font-family="cmr10" />
  <glyph unicode="#101;" glyph-name="e"
    horiz-adv-x="444.444" d="..skrátené.." />
  <glyph unicode="#115;" glyph-name="s"
    horiz-adv-x="394.444" d="..skrátené.." />
  <glyph unicode="#116;" glyph-name="t"
    horiz-adv-x="388.889" d="..skrátené.." />
  <missing-glyph horiz-adv-x="250" d="M0,0" />
</font>
  <font id="cmti10">
  <font-face font-family="cmti10" />
  ...skrátené...
</font>
  <font id="cmbx10">
  <font-face font-family="cmbx10" />
  ...skrátené...
</font>
</defs>
</svg>

```

Súbor `test.log` obsahuje údaje o činnosti sekcií `dvibase` a `dvipage`.

```

#### dvibase
PRE: i:2 num:25400000 den:473628672 mag:1000
      x:' TeX output 2004.02.17:1517'
Padding:7
POST_POST: q:349 i:2
POST: p:222 num:25400000 den:473628672 mag:1000
      1:43725786 u:30785863 s:2 t:2
FNT_DEF_1: k:36, c:4244645690, s:655360, d:655360, a:0,
l:6 n:cmti10
FNT_DEF_1: k:23, c: 452076118, s:655360, d:655360, a:0,
l:6 n:cmbx10
FNT_DEF_1: k: 0, c:1274110073, s:655360, d:655360, a:0,
l:5 n:cmr10

```

dvibasefontfree: 0 23 36

BOP: c0:2 c1:0 c2:0 c3:0 c4:0 c5:0 c6:0 c7:0 c8:0 c9:0 p:42

BOP: c0:1 c1:0 c2:0 c3:0 c4:0 c5:0 c6:0 c7:0 c8:0 c9:0 p:-1

V ďalšej sekcii vidieť fungovanie stavového automatu SM. Stav  $SM=\langle state \rangle$  sú popísané v kap. 8.2. Spracovávaný znak resp. reťazec má tvar  $charSM=(string)$ .

#### dvipage

Stack depth: 2

bop: Page 1

$SM=\langle 1 \rangle \langle push \rangle$   $SM=\langle 1 \rangle \langle d3 \rangle$   $SM=\langle 1 \rangle \langle pop \rangle$   $SM=\langle 1 \rangle \langle d4 \rangle$   $SM=\langle 1 \rangle \langle push \rangle$

$SM=\langle 1 \rangle \langle d4 \rangle$   $SM=\langle 1 \rangle \langle push \rangle$   $SM=\langle 1 \rangle \langle r3 \rangle$   $SM=\langle 1 \rangle$

[fnt\_def\_0] c:1274110073, s:655360, d:655360, a:0, l:5

s/d: 1.000000

n:cmr10

Font metrics cmr10:

/usr/local/teTeX/texmf/fonts/tfm/public/cm/cmr10.tfm

$SM=\langle 1 \rangle$

[fnt\_num\_0]  $SM=\langle 2 \rangle tSM=(t)$   $SM=\langle 3 \rangle eSM=(te)$   $SM=\langle 3 \rangle sSM=(tes)$

$SM=\langle 3 \rangle tSM=(test)$   $SM=\langle 3 \rangle \langle pop \rangle$   $SM=\langle 3 \rangle \langle y3 \rangle$   $SM=\langle 4 \rangle \langle push \rangle$

$SM=\langle 4 \rangle \langle r3 \rangle$   $SM=\langle 4 \rangle$

[fnt\_def\_36] c:4244645690, s:655360, d:655360, a:0, l:6

s/d: 1.000000

n:cmti10

Font metrics cmti10:

/usr/local/teTeX/texmf/fonts/tfm/public/cm/cmti10.tfm

$SM=\langle 4 \rangle$

[fnt\_num\_36]  $SM=\langle 2 \rangle tSM=(t)$   $SM=\langle 3 \rangle eSM=(te)$   $SM=\langle 3 \rangle sSM=(tes)$

$SM=\langle 3 \rangle tSM=(test)$   $SM=\langle 3 \rangle \langle pop \rangle$   $SM=\langle 3 \rangle \langle y0 \rangle$   $SM=\langle 4 \rangle \langle push \rangle$

$SM=\langle 4 \rangle \langle r3 \rangle$   $SM=\langle 4 \rangle$

[fnt\_def\_23] c: 452076118, s:655360, d:655360, a:0, l:6

s/d: 1.000000

n:cmbx10

Font metrics cmbx10:

/usr/local/teTeX/texmf/fonts/tfm/public/cm/cmbx10.tfm

$SM=\langle 4 \rangle$

[fnt\_num\_23]  $SM=\langle 2 \rangle tSM=(t)$   $SM=\langle 3 \rangle eSM=(te)$   $SM=\langle 3 \rangle sSM=(tes)$

$SM=\langle 3 \rangle tSM=(test)$   $SM=\langle 3 \rangle \langle pop \rangle$   $SM=\langle 3 \rangle \langle pop \rangle$   $SM=\langle 3 \rangle \langle d3 \rangle$

$SM=\langle 4 \rangle \langle push \rangle$   $SM=\langle 4 \rangle \langle r4 \rangle$   $SM=\langle 4 \rangle$

[fnt\_num\_0]  $SM=\langle 2 \rangle lSM=(1)$   $SM=\langle 3 \rangle \langle pop \rangle$   $SM=\langle 3 \rangle$

eop

$SM=\langle 0 \rangle$

bop: Page 2

```
<push><d3><pop><d4><push><d4><push><r3>  
[fnt_num_0]nexttest<pop><y3><push><r3>  
[fnt_num_36]nexttest<pop><y0><push><r3>  
[fnt_num_23]nexttest<pop><pop><d3><push><r4>  
[fnt_num_0]2<pop>  
eop
```

dvitfmfree: 0 23 36

## Kapitola 9

### Záver

Problematika výstupu do SVG je oveľa širšia ako na prvý pohľad zdá. Problematická je aktuálna podpora XML, tiež súvisiacich štandardov a štandardu SVG samotného. Najprijateľnejšie sa javí SVG Viewer od firmy Adobe [24], avšak ani ten nepodporuje napr. externé SVG fonty. Podpora vstavaných fontov je však dobrá a tiež potešujúca je aj dostupnosť prezerača SVG Viewer pre rôzne platformy.

Ako bolo v práci ukázané, existencia nástrojov s výstupom do SVG nie je okrajová, a v budúcnosti ich počet a kvalita bude rásť. Nedoriešený je však výstup viacerých strán vyprodukovaných  $\TeX$ om do jedného SVG súboru, čo je dôležité pre priblíženie sa k samostojnosti dokumentov vo formátoch PostScript a PDF, avšak využitia vlastností špecifických pre publikovanie na Internete.

Dúfame, že vlastná implementácia výstupu do SVG v DVISVG sa uplatní v praxi a bude sa ďalej rozvíjať.

## Kapitola 10

### Skratky

W3C	World Wide Web Consortium
SVG	Scalable Vector Graphics
XML	Extensible Markup Language
CSS2	Cascading Style Sheets Level 2
DVI	DeVice Independent
TFM	TeX Font Metric
CFF	Compact Font Format
CEF	Compact Embeded Format
PDF	Portable Document Format
PS	PostScript



## Literatúra

- [1] Adobe Systems Inc.: *The Compact Font Format Specification, Version 1.0*. Technical Note #5176, 2003.
- [2] Adobe Systems Inc.: *The Type 2 Charstring Format*. Technical Note #5177, 2000.
- [3] Bergholz, A.: *Extending your markup: An XML Tutorial*. IEEE Internet Computing, July-August 2000.
- [4] Berry, K.: *Kpathsea: A Library for path searching*. [kpathsea.info](http://kpathsea.info).
- [5] Berry, K., Weber, O.: *Web2C: A T<sub>E</sub>X implementation*. [web2c.info](http://web2c.info).
- [6] Bos, B., Lie, H. W., Lilley, Ch., Jacobs I. (Edts.): *CSS2 Level 2, Recommendation*. W3C, 1998.
- [7] Brotz, D., Paxton, B., Walden, J.: *Adobe Type 1 Font Format*. Adobe Systems Inc., Addison-Wesley, Reading, MA, USA, 1990.
- [8] Capin, T. (Edt.): *Mobile SVG Profiles: SVG Tiny and Basic, Recommendation*. W3C, 2003
- [9] Clark, J., DeRose, S. (Edts.): *XPath Version 1.0, Recommendation*. W3C, 1999.
- [10] Danilo, A. (Etd.): *SVG Print, Working Draft*. W3C, 2003
- [11] DeRose, S., Maler, E., Orchard, D. (Edts.): *XLink Version 1.0, W3C Recommendation*, 2001.
- [12] DeRose, S., Maler, E., Daniel, R. (Edts.): *XPointer xpointer() Scheme, Working Draft*. W3C, 2002.
- [13] Geshchke, Ch.: *PostScript Language Tutorial and Cookbook*. Adobe Systems Inc., Addison-Wesley, Reading, MA, USA, 1985.
- [14] Goossens, M., Sivunen, V.: *L<sup>A</sup>T<sub>E</sub>X, SVG, Fonts*. TUGboat Volume 22 No. 4, 2001.
- [15] Jackson, D. (Edt.): *SVG 1.1, Recommendation*. W3C, 2003.
- [16] Jackson, D. (Edt.): *SVG 1.2, Working Draft*. W3C, 2004.
- [17] Knuth, D. E.: *The T<sub>E</sub>Xbook*. Addison-Wesley, Reading, MA, USA, 1986.
- [18] Knuth, D. E.: *T<sub>E</sub>X: The Program*. Addison-Wesley, Reading, MA, USA, 1986.
- [19] Olšák, P.: *Typografický systém T<sub>E</sub>X*. Československé sdružení uživatelů T<sub>E</sub>Xu, Praha 1995.
- [20] Olšák, P.: *T<sub>E</sub>Xbook naruby*. KONVOJ, spol. s r.o., Praha, 1997.
- [21] Marsh, J.: *XML Base, Recommendation*. W3C, 2001.

- 
- [22] Tobin, R. (Edt.): *Namespaces in XML 1.1, Recommendation*. W3C, 2004.
- [23] Yergeau, F. (Edt.): *XML 1.0 (3rd edition), Recommendation*. W3C, 2004.
- [24] Adobe Systems Inc.: <http://www.adobe.com/>.
- [25] Aladdin Enterprises.: *Ghostscript*. <http://www.cs.wisc.edu/~ghost/>.
- [26] Esser, T.: *teTeX*. <http://www.tug.org/tetex/>.
- [27] Frischauf, A.: *dvi2svg*. <http://www.active-math.org/~adrianf/dvi2svg/>.
- [28] Malyshev, B. K.: *BaKoMaTeX*.  
<http://www.ctan.org/tex-archive/nonfree/systems/win32/bakoma/>.
- [29] Peinecke, N.: *font2svg*. <http://www.cg.uni-hanover.de/~peinecke/font2svg/>.
- [30] Pdftron.: *dvi2svg*. <http://www.pdftron.com/pdf2svg/>.
- [31] Savage Software.: *dvi2svg* <http://www.savagesoftware.com/products/pdf2svg/>.
- [32] Solid Code.: *dvi2svg*. <http://www.solidcode.net/pdf2svg/>.
- [33] Glunz, W.: *pstoedit*. <http://www.pstoedit.net/pstoedit/>.
- [34] MicroPress.: <http://www.micropress-inc.com/>.
- [35] W3C.: <http://www.w3c.org/>.
- [36] W3C.: *Graphics*. <http://www.w3c.org/Graphics>.