

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Tvorba vexilografického modulu s využitím jazyků MetaPost a Lua

Bakalářská práce

Marek Treťák

Vedoucí práce: RNDr. Tomáš Hála, Ph. D.

Brno 2016

NA MÍSTĚ TOHOTO LISTU
SE NACHÁZÍ ORIGINAL
ZADÁNÍ PRÁCE

Na tomto místě bych rád poděkoval panu RNDr. Tomáši Hálovi Ph. D. za vedení bakalářské práce, cenné rady a čas, který mi věnoval.

Čestné prohlášení

Prohlašuji, že jsem práci *Tvorba vexilografického modulu s využitím jazyků MetaPost a Lua* vypracoval samostatně a veškeré použité prameny a informace uvádím v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách, ve znění pozdějších předpisů a v souladu s platnou Směrnicí o zveřejňování vysokoškolských závěrečných prací.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 21. května 2016

.....
podpis

Abstract

TREŠÁK, MAREK. *Creation of vexillographic module with the use of MetaPost and Lua languages*. Bachelor thesis. Brno, 2016.

This bachelor thesis deals with creation vexillographic module for typesetting system ConT_EXt. The first part describes vexillology, its history and terminology and basic patterns of flags. Next part focuses on typesetting system T_EX, its format ConT_EXt and programming languages MetaPost and Lua. The third passage discusses implementation of module. The last section presents capabilities of module.

Key words

flag, banner, vexillology, vexillography, module, T_EX, ConT_EXt, MetaPost, Lua

Abstrakt

TREŠÁK, MAREK. *Tvorba vexilografického modulu s využitím jazyků MetaPost a Lua*. Bakalářská práce. Brno, 2016.

Tato bakalářská práce se zabývá tvorbou vexilografického modulu pro typografický systém ConT_EXt. V první části je stručně popsána vexilologie, její historie, používaná terminologie a základní vzory vlajek. Další část práce popisuje typografický systém T_EX, jeho formát ConT_EXt a programovací jazyky MetaPost a Lua. Následně je v práci popsán postup implementace modulu. Poslední část prezentuje schopnosti vytvořeného modulu.

Klíčová slova

vlajka, prapor, vexilologie, vexilografie, modul, T_EX, ConT_EXt, MetaPost, Lua

Obsah

1	Úvod a cíl práce	9
1.1	Úvod	9
1.2	Cíl práce	9
2	Vexilologie	10
2.1	Historie	10
2.2	Terminologie	11
2.3	Základní vzory dělení plochy listu	13
3	Použité technologie	15
3.1	T _E X	15
3.1.1	Jednotky v T _E Xu	16
3.2	ConT _E Xt	17
3.2.1	Srovnání s konkurencí	17
3.3	MetaPost	18
3.3.1	Datové typy	19
3.3.2	Možnosti MetaPostu	19
3.4	Lua	20
4	Tvorba modulu	21
4.1	Podkladové materiály	21
4.2	Návrh rozhraní modulu	22
4.2.1	Příkaz pro generování vlajky	22
4.2.2	Příkaz pro generování praporu	23
4.2.3	Příkaz pro generování vlajky podle vzoru	23
4.3	Popis implementace příkazů	24
4.3.1	Příkazy <code>\flag</code> a <code>\banner</code>	24
4.3.2	Příkaz <code>\defineflag</code>	24
4.3.3	Příkaz <code>\@CountryFlag</code>	25
4.3.4	Příkazy jednotlivých zemí	26

4.3.5	Příkaz <code>\ratioTest</code>	27
4.4	Tvorba základních vzorů listů	28
4.5	Ohraničení vlajky a praporu	30
4.6	Svislé verze vlajek a praporů	30
4.7	Barvy	30
4.8	Tvorba znaků	32
4.8.1	Použitý software	32
4.8.2	Příprava podkladu	32
4.8.3	Tvorba znaků pomocí MetaPostu	33
5	Ukázka generování vlajek a praporů	36
6	Diskuze	37
7	Závěr	38
	Literatura	38
	Zdroje informací o parametrech vlajek	40
	Přílohy	46

1 Úvod a cíl práce

1.1 Úvod

Vlajky jsou jedním z obecně nejrozšířenějších symbolů, se kterými se setkáváme. Jejich význam spočívá v označení příslušnosti ke státu, organizaci či jiné skupině. Zvolené barvy a obrazce na vlajkách jednotlivých zemí většinou skrývají nějaký význam, úzce souvisí s historií země a vytváří sounáležitost občanů ke své vlasti. Vlajky z celého světa můžeme vidět při mezinárodních sportovních utkáních, při zahájení olympijských her nebo při setkáních vrcholných představitelů států. Některé vlajky jsou velmi známé a bez problémů identifikujeme zemi, ke které je přiřadit, u jiných marně tápeme v paměti. V poslední době se vlajky a prapory kromě v klasických látkových provedeních používají i v jiných podobách. Ve stále větší míře se začaly vyskytovat v elektronických i tištěných dokumentech.

Vytvářením dokumentů vysoké kvality se zabývá typografický systém ConT_EXt, který je možno rozšířit externími moduly. Ty jsou tu proto, aby obohatily jeho základní možnosti o nové specifické funkce. A právě tvorbou modulu pro typografický systém ConT_EXt, který bude generovat vlajky a prapory, se zabývá i tato práce.

1.2 Cíl práce

Hlavním cílem této práce je vytvoření vexilografického modulu pro typografický systém ConT_EXt. Modul bude umožňovat generování vlajek a praporů podle zadaných parametrů a bude obsahovat výchozí sadu zemí Evropské unie. Definované příkazy nového modulu určené běžným uživatelům budou mít snadné a intuitivní ovládání. Při tvorbě vexilografického modulu budou kromě makrojazyka ConT_EXtu využity také programovací jazyky Lua a MetaPost.

2 Vexilologie

Vexilologie je pomocná věda historická, jejímž předmětem studia jsou vlajky a prapory, případně i ostatní vexilologické symboly. Zabývá se jejich popisem, funkcí, výkladem, historickým vývojem, způsoby používání, pravidly navrhování i problematikou vlajek a praporů obecně. Název této vědní disciplíny byl poprvé užit v roce 1959 a je odvozen od slova *vexillum*, názvu prvního prokázaného praporu na evropském území (MUCHA A VALÁŠEK, 1987). Problematika vlajek a praporů je ovšem mnohem starší a již dříve byla součástí jiných vědních disciplín, např. heraldiky, od které se vexilologie později odpoutala.

2.1 Historie

Za jednoho z předchůdců dnešních vlajek a praporů se považují tzv. *vexiloidy*. Šlo o hmotné předměty, které byly nejčastěji připevnovány na vrcholy tyčí, a dalo by se říci, že plnily funkci praporu. Nejstarší zmínky o těchto typech vexilologických symbolů pocházejí z Egypta a Mezopotámie a datují se až k roku 3500 př. n. l., respektive 4000 př. n. l. (MUCHA A VALÁŠEK, 1987).

Z hlediska historického vývoje bylo významné užití vlajek, praporů a případně jiných prevexilologických symbolů jako prostředku sloužícího k předání informace. Tato funkce byla využívána zejména pro vojenské účely. Prvním takto použitým symbolem a také prvním praporem v pravém slova smyslu bylo již zmíněné *vexillum* zavedené roku 105 př. n. l., které bylo vyrobeno z tkané látky a sloužilo jako rozlišovací znamení jízdy římských legií (MUCHA A VALÁŠEK, 1987). Další důležitou etapou, která měla podstatný vliv na rozšíření vexilologických symbolů, bylo období křížáckých výprav, kde byl prapor vztyčený na kopí jediné opravdu rozlišovací znamení v úzkém kontaktu velkého počtu křížáků a orientálních bojovníků (FRAJDL A ZÁLABSKÝ, 1989). Následně si vojenská koordinace vynucovala stále propracovanější tvorbu rozlišovacích prvků, které umožňovaly rozpoznání jednotlivých částí armády a vzájemnou koordinaci na bojišti i mimo něj.

V průběhu času vexilologické symboly začaly mít i jiný než vojenský účel, a to byl počátek vlajek a praporů tak, jak je známe dnes. První vlajkou, která reprezentovala stát, byla vlajka Dánského království, vznikla v roce 1219 a je v nezměněné podobě užívána dodnes (Dánský Dannebrog ..., 2009). Další významnou vlajkou je vlajka Nizozemska, která je tvořena třemi vodorovnými pruhy, a jde o první používanou trikoloru na světě (FRAJDL A ZÁLABSKÝ, 1989). Nizozemská vlajka měla vliv na podobu vlajek

řady zemí, ovlivnila například vlajky Francie, Lucemburska a vlajku Ruska, z které byly odvozeny vlajky řady slovanských národů (FRAJDL A ZÁLABSKÝ, 1989).

Podstatným prvkem při vývoji vlajek bylo jejich užívání v námořní dopravě. Vlajka, pod kterou loď plula, identifikovala zemi, ze které daná loď pocházela. Tento způsob identifikace se používá dodnes a má významné postavení v mezinárodním právu. Postupně se také jednoduché vlajkové signály využívané na dorozumění mezi loděmi vyvinuly v námořní vlajkovou abecedu, která je užívána k vizuálnímu dorozumívání na dálku.

Vyvíjely se také materiály, ze kterých byly vlajky vyrobeny a způsoby manipulace s nimi. Z nepraktických hmotných objektů se postupně stávaly klasické látkové vlajky, které umožňovaly lepší manipulaci a opakované vytahování nebo stahování pomocí šňůry.

Součástí některých vlajek jsou reprezentativní znaky. Dříve se často jednalo o poměrně složité symboly, později však z praktických důvodů docházelo k jejich zjednodušování nebo úplnému nahrazení jednoduššími geometrickými tvary. V mnoha případech se na vlajku dostávaly pouze pruhy v barvách původního znaku, což umožňovalo snadné a rychlé zhotovení většího množství vlajek (FRAJDL A ZÁLABSKÝ, 1989).

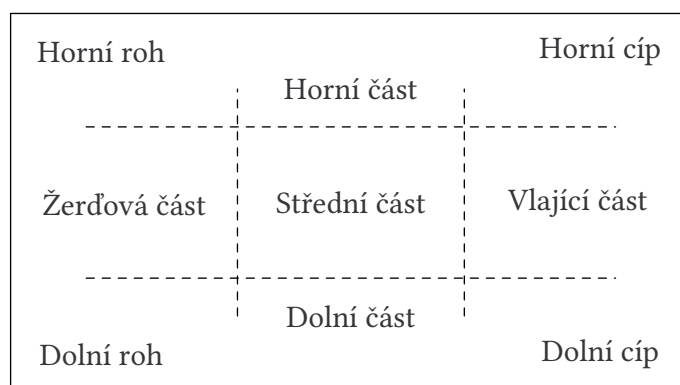
S rozvojem moderních států vznikaly také jejich vlajky a zároveň se kodifikovaly zásady jejich používání. Vlajka vypovídá o historii a kultuře dané země, o jejím státním a společenském zřízení a její hlavní funkcí se stala symbolická reprezentace státu. Mimo vlajek státních a národních došlo k rozvoji mnoha dalších druhů, jde například o vlajky územně správních celků, vlajky sportovní, služební, ale také vlajky firem a organizací. Při jejich užívání se začalo rovněž dbát na tzv. vlajkovou etiku, která určuje hierarchii vlajek, jejich obecné zásady užívání a výjimky, při nichž lze pravidla porušit.

Ve druhé polovině 20. století vzniklo ve světě velké množství nezávislých organizací zabývajících se vexilologií. V České republice jde o Českou vexilologickou společnost, která byla založena v roce 1972 a vydává čtvrtletní zpravodaj Vexilologie. Je členem Mezinárodní federace vexilologických asociací, která sdružuje 52 regionálních, národních a nadnárodních vexilologických spolků a institucí z celého světa (What is FIAV?, 2015).

2.2 Terminologie

Stejně jako jiné samostatné disciplíny i vexilologie má své vlastní názvosloví, které slouží k popisu vexilologických symbolů nejrůznějších druhů. Jeho účelem je stručně, jednoznačně a výstižně popsat vlajky a prapory, jejich části, vzory a další prvky, z nichž se skládají. Samotný popis probíhá z hlediska pozorovatele zleva doprava a odshora dolů (SVOBODA, 1972). Dále jsou uvedeny základní pojmy, které se v textu práce používají k detailnímu popisu vlajek a praporů.

List je základní částí praporu či vlajky, označuje se jím celé pole použité pro vyobrazení (MUCHA A VALÁŠEK, 1987). Názvosloví jednotlivých částí listu systematizoval SVOBODA (1972), který rozdělil list následujícím způsobem:



Obrázek 2.1 Názvosloví částí vlajkového listu

Vlajka je list různého tvaru, který má pevně stanovený poměr mezi šířkou a délkou (FRAJDL A ZÁLABSKÝ, 1989). Poměr je většinou určen zákonem příslušné země a je nutné ho dodržovat. Při popisu poměru se vždy nejdříve uvádí šířka a pak délka. Vlajky reprezentující stát mají ve většině případů obdélníkový tvar, výjimku tvoří vlajky Švýcarska a Vatikánu, které mají tvar čtvercový. Ojedinělou zajímavostí je pak vlajka Nepálu, která je složena ze dvou trojúhelníků.

Prapor je stejně jako vlajka list různého tvaru, avšak oproti vlajce nemá pevně stanovený poměr stran (FRAJDL A ZÁLABSKÝ, 1989). Je však doporučováno, aby délka nepřesahovala šířku více jak trojnásobně (Co je to vexilologie?, 2016). Neznamená to však, že lze měnit také poměry všech jeho vnitřních prvků. „Určuje-li tedy zákon, že modrý klín na české vlajce dosahuje do jejího středu, musí tak tomu být bezpodmínečně i u praporu!“ (EXNER, 2004). Naopak, pokud je na vlajce znak a není-li uvedeno jinak, jeho poměr stran musí zůstat nezměněn. Oproti vlajce je rozdíl taktéž v technice vyvěšování, což však v kontextu této práce není důležité. Pojmy prapor a vlajka bývají často u laické veřejnosti zaměňovány.

Státní vlajka slouží ke služebním účelům a používají ji výhradně státní instituce (MUCHA A VALÁŠEK, 1987). Oproti vlajce národní se v některých případech liší tím, že obsahuje navíc státní znak.

Národní vlajka může být využívána každým občanem (MUCHA A VALÁŠEK, 1987). V mnoha zemích, mimo jiné i v České republice, jsou vlajky státní a národní identické.

Šířka vlajky vyjadřuje v základní poloze její svislý rozměr, tj. u klasické obdélníkové vlajky pohledově výšku.

Délka vlajky je stanovena jako kolmice k šířce vlajky, tj. u klasické obdélníkové vlajky se jedná o pohledovou šířku.

Znak je součástí mnoha vlajek. Jedná se o libovolný symbol či jejich soubor umístěný na listu vlajky. Při popisu vlajek a praporů je třeba polohu znaků na listu přesně lokalizovat.

2.3 Základní vzory dělení plochy listu

Existuje několik základních způsobů dělení plochy listu, které se opakovaně vyskytují na mnoha vlajkách a jsou pro ně zavedeny termíny. Nejčastější vzory, které slouží jako inspirace při návrhu vlajek, uvádím v dalším textu. Názvosloví i popis jsou převzaty od SVOBODY (1972) a FRAJDLA A ZÁLABSKÉHO (1989).

Trikolóra je jeden z nejpoužívanějších vzorů. List vlajky je rozdělen rovnoměrně třemi stejně velkými pruhy různých barev. Rozeznáváme dva základní typy trikolór, horizontální a vertikální.

Třípruhový vzor je obdobně jako trikolóra vzor se třemi vodorovnými nebo svislými pruhy, avšak všechny nemusejí mít stejnou velikost a rozdílné barvy. Šíře jednotlivých pruhů se popisuje poměrem.

Bikolóra je vzor, kdy je list vlajky rozdělen rovnoměrně na dvě stejně velké části rozdílných barev. Stejně jako u předchozích vzorů rozlišujeme horizontální a vertikální typ.

Čtvrcení je vzor, kdy list je rozdělen na čtyři vzájemně rozeznatelné obdélníky stejné velikosti.

Kanton je obdélníkové pole umístěné v levé horní části listu. Kanton obsahují např. mnohé vlajky zemí bývalého britského impéria.

Svatojiřský kříž je kříž, který je centrálně umístěný a jeho ramena se dotýkají okrajů listu.

Skandinávský kříž je kříž, jehož střed je umístěn blíže k žerďové části listu vlajky a jehož ramena se dotýkají okrajů listu.

Zkrácený kříž někdy nazývaný také řecký kříž, je typ kříže, který má všechna ramena stejně dlouhá a nedotýká se okrajů listu.

Břevno je název pro užší, nejčastěji úhlopříčné pruhy, které rozdělují list vlajky na dvě části.

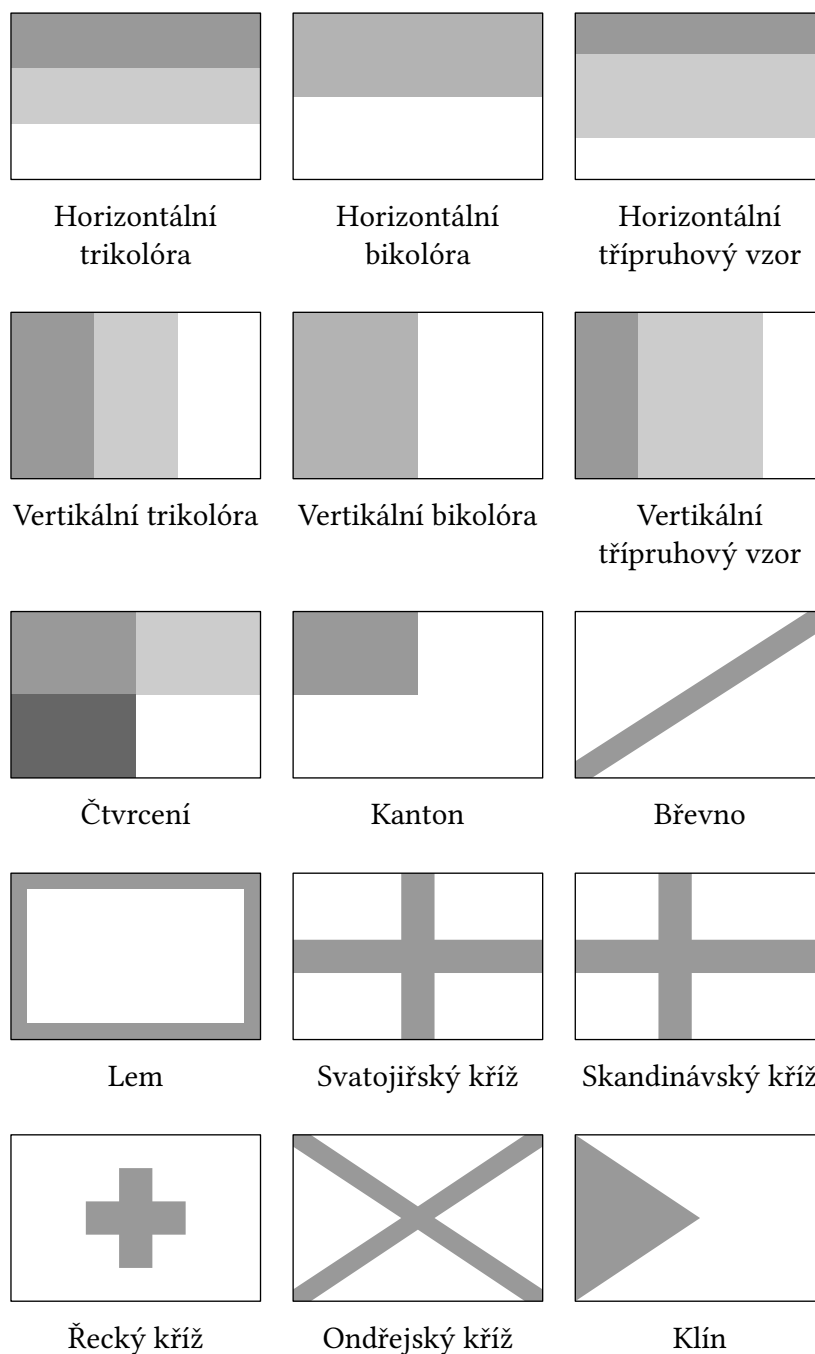
Ondřejský kříž je složen ze dvou břevnen propojujících protilehlé vrcholy.

Klín je obrazec ve tvaru trojúhelníka. Je potřeba přesně popsat umístění, například v případě České republiky se jedná o klín na žerďové části s hrotem v 1/2 délky vlajky.

LEM je pruh odlišné barvy, který vroubí list vlajky.

Vzory nedefinované výše uvedenými způsoby nazýváme obecně pole a při jejich popisu je nutné definovat nejen tvar, ale i polohu. Například japonská vlajka obsahuje uprostřed listu červené kruhové pole. Vyskytují se však také pole kosočtvercové a lichoběžníkové, případně kombinace jiných geometrických obrazců.

Velmi často se také na vlajkách vyskytují hvězdy, nejčastěji s pěti cípy, kdy jeden z cípů směřuje k horní části listu. Výjimečně však mají i jiný počet cípů a odlišný způsob uložení.



Obrázek 2.2 Ukázky základních vzorů dělení plochy listu

3 Použité technologie

3.1 T_EX

T_EX je profesionální a volně šiřitelný typografický systém sloužící k elektronické sazbě dokumentů vysoké kvality. Vytvořil jej koncem 70. let profesor DONALD E. KNUTH ze Stanfordské univerzity, který patří mezi uznávané odborníky v oblasti informačních technologií a mimo jiné je znám jako autor mnohosvazkové monografie Umění programování, na které stále pracuje. KNUTH (1986) uvádí, že prvotní impuls proč na T_EXu začal pracovat, je spjat právě s druhým vydáním druhého svazku této monografie, kdy se musela celá kniha znovu vysázet z důvodu přechodu od klasické „horké“ sazby k fotosazbě a výsledek zejména u matematických výrazů nebyl zdaleka takový, jaký očekával.

Samotný T_EX je vybaven přibližně třemi sty zabudovanými příkazy, kterým říkáme primitivní příkazy nebo primitivy (OLŠÁK, 2000). Z těchto příkazů je možno definovat větší a komplexnější celky, tzv. makra, která jsou uživatelsky mnohem přívětivější než používání samotných primitivů. Uživatel si však nemusí všechna potřebná makra definovat sám, existují makra seskupená do balíků, kterým se říká formáty T_EXu a které umožňují uživatelsky přívětivější sazbu. Velké oblíbenosti se T_EX a jeho formáty těší zejména v akademickém prostředí.

Jedním z formátů je plainT_EX, jehož autor je samotný DONALD E. KNUTH. Jedná se o základní formát, z kterého vycházejí téměř všechny ostatní formáty (OLŠÁK, 2001). Pravděpodobně nejznámější a nejrozšířenější je formát L^AT_EX od LEISLIE LAMPORTA, který se těší velké oblibě, a to i mezi běžnými uživateli bez znalosti programování. Mezi další formáty patří $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX, který byl původně vyvinut pro Americkou matematickou společnost a stal se uznávaným standardem na psaní vědeckých zejména matematických publikací (OLŠÁK, 2000). Formátem T_EXu je také ConT_EXt, kterému je věnována samostatná kapitola.

Mezi prací s T_EXem, jeho formáty a prací s běžnými většinou komerčními DTP programy je několik podstatných rozdílů. Pro obyčejné uživatele je bezesporu hlavní rozdíl v samotném způsobu ovládání. Největší konkurenti T_EXu pracují způsobem WYSISYG a v průběhu vytváření dokumentu se pokoušejí zobrazovat výsledný dokument tak, jak bude vytištěn. Práce s T_EXem však probíhá odlišným způsobem, který může být pro některé uživatele odrazující. Do souboru se pomocí libovolného textového editoru napíše požadovaný text a také příkazy, kterými se řídí způsob formátování. Následně se provede překlad souboru, jehož výsledkem je obecný soubor s koncovkou dvi (DeVice Independent), který je možné následně zpracovat dalšími programy.

Jednou z možností zpracování je využití programů, které dokáží dvi soubor zobrazit na počítačové obrazovce. Mezi používané varianty zpracování patří také výstup do jazyka PostScript, který se používá u profesionálních tiskáren a osvitových jednotek a lze ho pomocí speciálních programů rovněž zobrazit na obrazovce. V dnešní době se však ve většině případů využívá rozšíření pdf \TeX , které implicitně generuje dokumenty ve formátu PDF, jež lze jednoduše zobrazit pomocí běžně rozšířených PDF prohlížečů.

Posledním krokem při tvorbě dokumentů je kontrola získaného výstupu, případně návrat do editoru, oprava chyb a opakování uvedeného postupu tak dlouho, dokud nejsme s výsledkem plně spokojeni.

3.1.1 Jednotky v \TeX u

\TeX umožňuje při práci používat celou řadu jednotek délky, kromě standardních metrických a imperiálních jednotek lze užít také Didotův a angloamerický měrný systém. Níže v tabulce jsou uvedeny všechny podporované jednotky včetně potřebného vysvětlení.

Tabulka 3.1 Podporované jednotky v \TeX u, zdroj: HAGEN A HOEKWATER (2013)

Zkratka	Název	Poznámka
pt	point	$72,27 \text{ pt} = 1 \text{ in}$
bp	big point	$72 \text{ bp} = 1 \text{ in}$
pc	pica	$1 \text{ pc} = 12 \text{ pt}$
in	inch	$1 \text{ in} = 2,54 \text{ cm}$
cm	centimeter	$2,54 \text{ cm} = 1 \text{ in}$
mm	millimeter	$10 \text{ mm} = 1 \text{ cm}$
dd	didot point	$1 \text{ 157 dd} = 1 \text{ 238 pt}$
cc	cicero	$1 \text{ cc} = 12 \text{ dd}$
sp	scaled point	$65 \text{ 536 sp} = 1 \text{ pt}$

3.2 ConT_EXt

Jak již bylo řečeno, ConT_EXt je obsáhlý balík maker pro typografický systém T_EX, jehož cílem je usnadnit sazbu dokumentů. Je vyvíjen již od roku 1990 nizozemskou firmou PRAGMA Advanced Document Engineering. Vzhledem k faktu, že ve firmě byla řada nepříliš technicky zdatných uživatelů, kteří nebyli zvyklí na cizojazyčné prostředí, byl zpočátku ConT_EXt vyvíjen pouze s rozhraním v nizozemštině (HAGEN, 2000). Nyní však ConT_EXt podporuje vícejazyčné uživatelské rozhraní, které umožňuje uživatelům pracovat v řadě jazyků včetně angličtiny.

ConT_EXt je dostupný ve dvou vývojových verzích Mark II a Mark IV (zkráceně MkII a MkIV). Verze MkII využívá rozšíření X_YT_EX a pdfT_EX. Její vývoj byl pozastaven v roce 2008, avšak nadále pokračuje podpora v opravě nalezených chyb (Support, 2016). Verze MkIV je aktivně vyvíjená a pracuje s rozšířením LuaT_EX, což poskytuje řadu zcela nových možností. Samotný LuaT_EX je rozšíření pdfT_EXu o skriptovací jazyk Lua, který je možno volat přímo z ConT_EXtu a ovlivnit tak výslednou podobu dokumentu. Rozšíření LuaT_EX také nativně podporuje kódování Unicode (UTF-8) a OpenType formát fontů. Uživatelské rozhraní obou verzí ConT_EXtu je stejné, avšak při tvorbě MkIV byla značná část zdrojových kódů přepracována a implementována v jazyku Lua, což způsobuje v některých případech nekompatibilitu mezi oběma verzemi. Rozdíly jsou podrobně popsány na stránce ConT_EXt Garden (2016).

K vlastnostem ConT_EXtu patří například přímé zpracování soborů XML, tato vlastnost byla značně vylepšena ve verzi MkIV. ConT_EXt rovněž umožňuje generovat více variant stejného dokumentu, například jednu verzi určenou pro tisk a druhou pro zobrazení na počítačových obrazovkách včetně interaktivních prvků.

Pro ConT_EXt je dostupná řada externích modulů, kterými se dá rozšířit jeho funkčnost, lze uvést například modul ppchT_EX, který slouží k sazbě chemických vzorců.

V případě potřeby grafických prvků ConT_EXt v základu obsahuje podporu MetaPostu včetně rozšíření nazývaného MetaFun. Jako případná alternativa k MetaPostu může při použití externích modulů sloužit také PsTricks a TiKZ.

3.2.1 Srovnání s konkurencí

Vzhledem k rozšířenosti L^AT_EXu a vzrůstající popularitě ConT_EXtu v posledních letech se nabízí jejich vzájemné srovnání. Cílem obou formátů je zjednodušení sazby pro běžné uživatele, kteří nejsou profesionálové, a používání samotného T_EXu je pro ně náročné.

Výhodu má ConT_EXt oproti L^AT_EXu v tom, že v základu obsahuje vysokou funkcionalitu bez nutnosti importovat rozšiřující balíky, u kterých hrozí potenciálně vzájemná kolize. Za další přednost lze považovat, že je většina příkazů konfigurovatelná pomocí intuitivních voleb klíče a hodnoty, což značně usnadňuje práci. Pro běžné uživatele může být rovněž přínosem automatické určení počtu průchodů při potřebě přesunu informací v dokumentu, například při tvorbě obsahu. Na druhou stranu nevýhodou výše zmíněných přístupů je rychlost kompilace. Porovnání rychlosti překladu 200stránkového dokumentu mezi formáty plainT_EX, L^AT_EX a ConT_EXt provedl HOEKWATER (1998). Čas kompilace u plainT_EXu a L^AT_EXu byly 21, respektive 27 sekund a u nejpomalejšího ConT_EXtu šlo o 2 minuty a 59 sekund.

Za výhody L^AT_EXu lze považovat širokou rozšířenost a s tím související jeho přijetí jako určitého standardu pro psaní akademických publikací. Další výhodou je nepochybně existence řady kvalitních knih, z nichž může uživatel čerpat cenné rady. Naproti tomu prozatímni nedostatečná uživatelská dokumentace je slabinou ConT_EXtu. Vývoj tohoto systému probíhá velmi rychle, což způsobuje, že dokumentace je často zastaralá a mnoho funkcí v ní není zachyceno vůbec, což v případě neobvyklých požadavků značně zpomaluje práci. Pro základní formátování lze doporučit do českého jazyka přeloženou knihu Exkurze do ConT_EXtu (OTTEN A HAGEN, 2006), v případě hlubší potřeby pak portál ConT_EXt Garden (2016), kde se sdílí informace na principu wikipedie.

3.3 MetaPost

MetaPost je programovací jazyk a jeho interpret určený pro tvorbu grafických prvků. Jeho autorem je JOHN D. HOBBY a částečně vychází z METAFONTu, který byl primárně vytvořen jako nástroj sloužící k tvorbě počítačových písem. Autorem METAFONTu je stejně jako v případě T_EXu DONALD E. KNUTH a vytvořil v něm, mimo jiné, známé písmo Computer Modern. MetaPost z METAFONTu přebírá některé základní nástroje pro práci a manipulaci s obrázky (HOBBY, 2014).

Práce s MetaPostem probíhá podobným způsobem jako s T_EXem. Vstupem je obyčejný textový soubor vytvořený v libovolném editoru, který obsahuje posloupnost příkazů. Následuje překlad a prohlížení výsledku, případně opakování uvedeného postupu

tak dlouho, dokud nejsme s daným výsledkem spokojeni. Výstup z MetaPostu je kvalitní vektorová grafika ve formátech PostScript, SVG nebo rastrová grafika ve formátu PNG (HOBBY, 2014).

3.3.1 Datové typy

MetaPost využívá několik specifických datových typů. Datový typ *pair* slouží pro uložení souřadnic, které reprezentují bod v rovině a jsou uloženy jako dvojice čísel. K uložení cest (křivek a lomených čar) vytvořených jedním tahem slouží datový typ *path*. Datový typ *pen* je využíván pro uložení pera, kterým lze následně vykreslit křivku, případně bod. Další datový typ *transform* slouží pro uložení afinní transformace. Pro uložení obrázku a k možnosti skládání prvků do jednoho celku slouží datový typ *picture*. Dalšími běžnými datovými typy jsou *numeric* pro uložení čísla, *string* pro uložení řetězce, *color* pro uložení barvy a logický datový typ *boolean*, který může nabývat dvou hodnot.

3.3.2 Možnosti MetaPostu

MetaPost umí kreslit jak jednotlivé body, tak složité cesty a disponuje celou řadou příkazů, které umožňují ovlivnit jejich výsledný tvar. U křivek jde například o určení kontrolních bodů Beziérových křivek nebo změnu parametru „napětí“ křivky. Jsou možné také parametrické operace s cestami, např. určení průsečíku dvou cest nebo oříznutí části cesty. V případě, že jsou cesty uzavřeny, lze je vyplnit barvou.

Na některé objekty MetaPost umožňuje aplikovat afinní transformace. Jde například o změnu velikosti, otočení, zkosení nebo posunutí celého objektu o vektor.

Repertoár jazyka MetaPost obsahuje také standartní nástroje, jako jsou podmínky, případně několik druhů cyklů. Umožňuje rovněž definici vlastních maker, sazbu textů do obrázku nebo řešení lineárních rovnic.

3.4 Lua

Lua je procedurální skriptovací jazyk vyvíjený už od roku 1993 na Papežské katolické univerzitě Rio de Janeiro v Brazílii a jeho autory jsou ROBERTO IERUSALIMSCHY, WALDEMAR CELES a LUIZ HENRIQUE DE FIGUEIREDO. Lua byla vyvíjena původně pouze pro interní potřebu firmy PETROBRA, kde bylo potřeba zpracovat vstupní data pro simulace, které probíhaly několikrát denně (IERUSALIMSCHY A KOL., 2001). Aktuální verze jazyka Lua je 5.3.2 a je šířena pod svobodnou licencí MIT.

Při vývoji jazyka Lua byl od počátku kladen důraz na schopnost jeho integrace s ostatními jazyky (IERUSALIMSCHY, 2013). Dají se jím rozšířit kupříkladu programy psané v jazycích, jako jsou C/C++ nebo Java, ale i v řadě dalších. Taktéž programy psané v jazyku Lua lze snadno rozšířit pomocí knihoven napsaných v jiných jazycích.

Jedná se o velmi minimalistický jazyk, jehož interpret se základními knihovnamí má velikost pouze 245 kilobajtů (Lua, 2016). Lua je v porovnání s ostatními skriptovacími jazyky velmi rychlá. IERUSALIMSCHY (2013) uvádí, že Lua v nezávislých testech vyšla jako jeden z nejrychlejších skriptovacích jazyků.

Mezi programy, jejichž části byly napsány v jazyku Lua, patří například Adobe Photoshop Lightroom. Mnoho programů rovněž umožňuje ovlivnit své chování prostřednictvím skriptů napsaných v jazyce Lua, patří mezi ně například programy WireShark, VLC player nebo textový editor Vim. Jazyk Lua využívá také celá řada populárních počítačových her, jejichž větší či menší části v něm byly napsány. Lze uvést například velmi populární hry World Of Warcraft a Angry Birds. Za zmínku stojí, že díky své velikosti a jednoduchosti najde uplatnění také jako skriptovací jazyk v grafických kalkulačkách od firmy Texas Instrument nebo v programovatelných robotických stavebnicích vyráběných firmou Lego.

Jak již bylo zmíněno v jedné z předchozích kapitol, jazyk Lua lze volat přímo z Con-TeXtu. Je však potřeba mít na paměti, že veškerý Lua kód se nejdříve expanduje podle pravidel T_EXu, a až po té je kód předán interpretu jazyka Lua. Tudíž je nutné, aby veškerý Lua kód byl zapsán ve validní syntaxi T_EXu.

4 Tvorba modulu

Tvorba modulu probíhala v několika fázích. Nejdříve proběhlo seznámení s programovacími jazyky MetaPost, Lua a také s typografickým systémem $\text{T}_{\text{E}}\text{X}$ a jeho formátem $\text{ConT}_{\text{E}}\text{Xt}$. Po prozkoumání dostupných prostředků a jejich možností jsem přistoupil k tvorbě samotného modulu. V první fázi šlo o vyhledání vhodných podkladových materiálů a určení všech požadovaných funkcí modulu s ohledem na technické možnosti zvolených prostředků a v teoretické části popsanych vlastností vlajek a praporů. V dalších fázích pak bylo přistoupeno k implementaci jednotlivých částí modulu.

4.1 Podkladové materiály

Na začátku práce bylo nutné vyhledat vhodné předlohy, které sloužily k následné digitalizaci. Informace byly převážně čerpány ze zákonů či jiných oficiálních dokumentů dané země, kde jsou popsány parametry státních symbolů, v případě České republiky jde například o zákon č. 3/1993 Sb., o státních symbolech České republiky. Výjimečně také některé země poskytují poměrně přesný grafický manuál, který obsahuje popis jednotlivých částí vlajek včetně ukázek způsobů použití. Práci značně komplikoval fakt, že je většina těchto informací dostupná pouze v úředních jazycích dané země a také to, že řada těchto oficiálních dokumentů byla publikovaná před všeobecným rozšířením digitálních technologií, a tudíž v doporučeních není reflektován technologický vývoj. Jako doplňkový zdroj sloužil dokument *Flags and Anthems Manual London (2012)*, jde o oficiální manuál doporučených praporů pro olympijské hry v Londýně, kam každá národní organizace zastupující danou zemi poslala vzor praporu, tudíž se dá předpokládat, že tento manuál poskytuje věrohodné informace. V některých případech byly informace, zejména o používaných barvách, hledány také na webových stránkách institucí, které používaly národní vlajky.

Ne vždy je součástí oficiálních zdrojů kromě popisu vlajek také vyobrazení, což způsobuje zejména u složitějších vzorů špatné reprodukce vlajek. Problém je rovněž následné přebírání nepřesných reprodukcí ostatními uživateli, a to mnohdy i oficiálními institucemi dané země. Chybná reprodukce se pak často díky své přítomnosti na internetu velmi rychle rozšíří.

4.2 Návrh rozhraní modulu

Při návrhu způsobu ovládání modulu bylo rozhodnuto, že modul bude obsahovat tři jednoduché a intuitivní příkazy určené pro běžné uživatele. Při návrhu byly brány v potaz již existující příkazy ConT_EXtu, aby se zachovala jednotná linie ovládání. Přestože ConT_EXt obsahuje vícejazyčné rozhraní, bylo z praktických důvodů rozhodnuto, že všechny příkazy a jejich parametry budou v anglickém jazyce.

4.2.1 Příkaz pro generování vlajky

První příkaz má na starosti generovat národní vlajky jednotlivých zemí a má následující tvar:

```
\flag[kód země] [volby]
```

Parametr v prvních hranatých závorkách určuje zemi, jejíž vlajka se bude generovat. Vyhledávání probíhá podle kódů, které má přiřazena každá země. Jedná se o normu ISO 3166-1, kterou zveřejnila Mezinárodní organizace pro normalizaci. Tato norma definuje trojici kódů pro státy a závislá území. Jedná se o dvojmístné a trojmístné písmenné kódy a trojmístný číselný kód. Pro identifikaci země byly vybrány oba dva písmenné kódy z následujících důvodů: dvojmístné kódy jsou dobře známé, protože se až na pár výjimek využívají i jako kódy národních domén nejvyššího řádu, a trojmístné písmenné kódy umožňují nejsnadnější vizuální přiřazení mezi kódem a názvem země. Přestože Evropská unie není formálně státem, a z toho důvodu není do normy zařazena, je z praktického důvodu kombinace písmen EU rezervována (ISO, 2016), takže při volání její vlajky nedojde ke kolizi.

Parametry v druhých hranatých závorkách obsahují volby, které ovlivní výslednou podobu vlajky. Všechny parametry jsou typu klíč–hodnota a jsou oddělené čárkou.

Tabulka 4.1 Volby z druhého parametru

Parametr	Význam
width	šířka vlajky
length	délka vlajky
scale	měřítko
frame	ohraničení vlajky
orientation	orientace vlajky

Parametry `width` a `length` určují rozměr generované vlajky. Při zadávání šířky a délky vlajky lze užít všechny podporované rozměrové jednotky \TeX u, které byly zmíněny v kapitole 3.1.1.

Parametr `scale` je číselný koeficient, kterým se násobí implicitní velikost vlajky, nejsou-li parametry `width` a `length` zadány.

Parametr `frame` slouží k možnosti ohraničení vlajky například z důvodu splývání bílé barvy na vlajce s pozadím. Parametr má dvě volby `yes` a `no`. Pokud není zadán, generuje se vlajka bez ohraničení.

Parametr `orientation` určuje, jestli se bude generovat horizontální nebo vertikální verze vlajky. Parametr má dvě volby a to `landscape` pro horizontální polohu a `portrait` pro vertikální polohu vlajky. Pokud není zadán, generuje se horizontální verze.

4.2.2 Příkaz pro generování praporu

Druhý příkaz slouží ke generování praporů jednotlivých zemí a má následující tvar:

```
\banner [kód země] [volby]
```

Funkcionalita příkazu je stejná jako v případě příkazu na generování vlajek. Rozdíl je pouze ve zpracování rozměrových hodnot z druhého parametru vzhledem k vlastnostem praporu, který nemá pevně definován poměr šířky a délky.

4.2.3 Příkaz pro generování vlajky podle vzoru

Poslední uživatelsky přístupný příkaz je určen k možnosti vygenerovat uživatelem vlajku na základě zvoleného vzoru. Příkaz umožňuje generovat mimo jiných i vzory zmíněné v teoretické části práce a má následující tvar:

```
\defineflag [vzor] [volby] [] . . .
```

První parametr příkazu je číselný identifikátor, který určuje vzor generované vlajky. Druhý parametr obsahuje stejné volby typu klíč–hodnota jako předchozí dva příkazy, ale navíc přidává volbu `mirrored` se stavby `yes` a `no`, která slouží k zrcadlovému otočení vlajky.

Počet dalších parametrů závisí na konkrétně zvoleném vzoru. Jedná se o parametry určující počet prvků v daném vzoru. Jako poslední parametry jsou zadávány barvy výplní jednotlivých částí vlajky.

4.3 Popis implementace příkazů

Následuje popis implementace jednotlivých příkazů včetně ukávek konkrétních částí zdrojových kódů, které pomohou pochopit jejich význam.

V názvech některých příkazů, které nejsou určeny k volání uživatelem, se vyskytuje znak @. Užití tohoto znaku slouží k zakrytí příkazu, takže ho uživatel nemůže přímo použít nebo měnit. Předtím, než byly tyto vnitřní příkazy definovány, bylo třeba změnit kategorii znaku @. K tomu byl použit příkaz `\unprotect`, který přenastavil kategorie speciálních znaků využívaných v názvech příkazů. Po ukončení definování vnitřních maker byl užit příkaz `\protect`, který změnil kategorii znaků zpět na původní.

4.3.1 Příkazy `\flag` a `\banner`

Definice příkazů `\flag` a `\banner` je prakticky stejná, oba mají dva parametry a zavolají příkaz `\@CountryFlag`, kde předají první i druhý parametr a jako třetí parametr uvedou, zda se jedná o vlajku nebo prapor. Kód příkazu `\flag` je zobrazen níže.

```
\def\flag[#1][#2]{%
    \@CountryFlag[#1][#2][flag]
}
```

4.3.2 Příkaz `\defineflag`

O definici příkazu `\defineflag`, který generuje vlajku podle zadaného vzoru, se nebudu podrobněji rozepisovat, jelikož jeho funkcionalita je velmi podobná příkazu `\@CountryFlag`, který je popsán v následující kapitole. Za zmínku pouze stojí, že bylo nutné zajistit volitelný počet argumentů vzhledem k tomu, že každý vzor vyžaduje rozdílný počet parametrů. Pro tuto potřebu byl využit příkaz `\doseventupleargument`.

4.3.3 Příkaz \@CountryFlag

Tento příkaz slouží k vyhledání země podle zadaného kódu z prvního parametru a ke zpracování voleb typu klíč–hodnota, které obsahuje parametr druhý.

V těle příkazu \@CountryFlag se jako první volá příkaz \@getparameters, který nabízí ConTeXt ke zpracování parametrů typu klíč–hodnota.

Příkaz má dva parametry, první parametr označuje kategorii, ke které patří jednotlivé hodnoty z druhého parametru, díky čemuž lze použít stejný klíč při definici u více příkazů.

Druhý parametr přebírá seznam voleb typu klíč–hodnota oddělené čárkami. Vytvoří se sady příkazů, které vracejí zadané hodnoty a v případě, kdy uživatel nic nezadá, je možnost nastavit výchozí hodnotu. Například při zadání volby frame=yes se vytvoří příkaz \@ZZframe, který vrací yes a v případě, kdy není frame zadáno, je vrácena hodnota no, viz ukázka kódu níže.

Dále byly definovány dva vnitřní příkazy. První je \@errorCountry a volá se v případě, že není zadán žádný kód země. Druhý příkaz je \@checkCountry a zavolá se, když zadaný parametr neodpovídá žádnému z definovaných ISO kódů. Oba příkazy v případě jejich zavolání přeruší překlad a oznámí chybu na terminál.

Jako poslední je volán příkaz \@processaction. Jedná se o příkaz, který na základě jednoho předaného parametru zavolá příkaz dané země podle shody předaného parametru s ISO kódem. V případě chyby se zavolá jeden z příkazů, které jsou uvedeny v předchozím odstavci.

Následující ukázka obsahuje část zdrojového kódu příkazu \@CountryFlag.

```
\def\@CountryFlag[#1][#2][#3]{%
\getparameters[@@ZZ][frame=no,orientation=landscape,#2]
  \def\@errorCountry{%
    \errmessage{You must enter ISO country code}
  }
  \def\@checkCountry##1{%
    \errmessage{##1 is not supported country}
  }
  \@processaction[#1][
    eu=>\@EuropeanUnion[#2][#3]},
    alb=>\@CountryAlbania[#2][#3]},
    al=>\@CountryAlbania[#2][#3]},
    ...
    ua=>\@CountryUkraine[#2][#3]},
    \s!default=>\@errorCountry},
    \s!unknown=>\@checkCountry{#1}] }
```

4.3.4 Příkazy jednotlivých zemí

Dále bylo definováno celkem 43 příkazů pro jednotlivé země a příkaz pro Evropskou unii. Příkazy obsahují všechny potřebné informace nutné k vygenerování požadované vlajky či praporu.

Nejdříve jsou definovány všechny barvy, které se vyskytují v dané vlajce, případně v jejím znaku. Následuje volání příkazu `\ratioTest`, který má na starosti výpočty a ověření zadaných rozměrů. O barvách i příkazu `\ratioTest` dále pojednávají samostatné kapitoly.

Následuje vykreslení vlajky či praporu dané země pomocí `MetaPostu`, které se odehrává mezi dvojicí příkazů `\startMPcode` a `\stopMPcode`. Jsou zde volány všechny části, které vlajka obsahuje. Jde o vzor, z kterého vlajka vychází, a také znak, pokud je součástí dané vlajky. Dále jsou volány příkazy, které mají na starost ohrazení vlajky a změnu orientace.

Ukázkový příkaz Slovinska znázorňuje následující příklad:

```
\def\@CountrySlovenia[#1] [#2] {%
\definecolor[SloveniaBlue] [c=1,m=.8,y=0,k=0,r=.058,g=.2157,b=.549]
\definecolor[SloveniaRed] [c=0,m=.95,y=1,k=0,r=.902,g=.1373,b=.098]
\definecolor[SloveniaYellow] [c=0,m=.2,y=1,k=0,r=.98,g=.823,b=.078]

\ratioTest[#1] [#2] [1/2]

\startMPcode
  \@HorizontalTricolor[\the\FLAWIDTH] [\the\FLALENGTH] [white]
  [SloveniaBlue] [SloveniaRed]
  \@CoatOfArmsSlovenia
  \@FlagFrame[\@@ZZframe]
  \@FlagManipulation[\@@ZZorientation] [yes]
\stopMPcode
}
```

4.3.5 Příkaz `\ratioTest`

Kvůli omezeným schopnostem \TeX u provádět numerické výpočty byl tento příkaz vytvořen ve srozumitelnějším procedurálním jazyku Lua. Kód příkazu je zapsán v prostředí, které se ohraničuje povely `\startluacode` a `\stopluacode`.

Následující ukázka kódu přibližuje způsob, jakým bylo naprogramováno rozhraní příkazu `\ratioTest`, který lze následně volat mimo toto prostředí úplně stejným způsobem jako příkaz definovaný v $\text{Con}\TeX$ tu.

```
interfaces.definecommand {
name = "ratioTest",
arguments = {
{ "option", "string" },
{ "option", "string" },
{ "option", "string" },
{ "option", "string" },
},
macro = u.ratioTest,
}
```

Jako první parametr příkaz `\ratioTest` přijímá předané klíče a hodnoty, které obsahují uživatelem zadané rozměry. Druhý parametr identifikuje, jestli se budou počítat a ověřovat rozměry pro prapor či vlajku. V třetím parametru je předán předepsaný poměr šířky a délky vlajky. Ve výjimečných případech, většinou z historických důvodů, mají vlajky definovány dva možné poměry šířky a délky, jde například o vlajku Lucemburska nebo Monaka. Je-li tomu tak, je tento druhý poměr předán jako čtvrtý parametr.

Funkce `ratioTest` zpracuje předané hodnoty a zavolá funkci `flagTest` v případě, že se mají vypočítat rozměry vlajky nebo funkci `bannerTest`, pokud se počítají rozměry praporu.

Význam funkce `flagTest` je následující. Pokud uživatel zadal šířku i délku vlajky, ověří se, jestli tyto hodnoty odpovídají oficiálnímu předepsanému poměru stran požadované vlajky. Pokud tomu tak není, je přerušena kompilace a uživatel je informován na terminál, že zadané rozměry neodpovídají oficiálnímu poměru stran. V případě, že uživatel zadá jen jeden rozměrový údaj, je druhý dopočítán podle předepsaného poměru. V případě, kdy nebyl zadán ani jeden rozměrový parametr, je výsledná velikost vlajky ovlivněna zadaným parametrem `scale`, kterým se násobí implicitní šířka vlajky, stanovená na 1 cm, a dopočítá se délka vlajky podle předepsaného poměru.

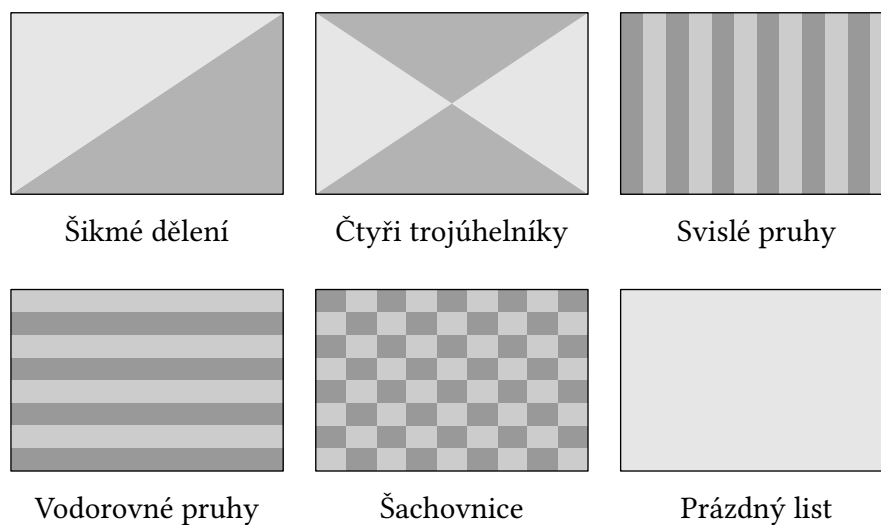
V případě praporu a funkce `bannerTest` dojde pouze k informačnímu ověření zadaných rozměrů. Pokud je zadán nevhodný poměr šířky a délky praporu, je uživatel informován na terminál, a to v případech, pokud je délka praporu kratší než jeho šířka,

anebo když délka praporu přesahuje šířku více jak třikrát. Pokud nejsou rozměry zadány, je velikost opět vypočítána z implicitního rozměru na základě zadaného parametru `scale`.

Pro uložení výsledných rozměrů vlajky či praporu byly v TeXu deklarovány registry `FLAWIDTH` a `FLALENGTH`, které jsou následně v TeXu přístupné pomocí primitivu `\the`.

4.4 Tvorba základních vzorů listů

Další částí práce bylo vytvoření celkem dvaceti základních vzorů dělení plochy listu, z nichž ve vexilologii nejčastěji užívané byly popsány v teoretické části práce a jsou znázorněny na obrázku 2.2. Ukázkou zbývajících přináší obrázek 4.1. Pro každý vzor byl vytvořen vlastní příkaz, který obsahuje kód v jazyce MetaPost. Byl vytvořen také příkaz pro prázdný list, který slouží k obarvení pozadí u některých vzorů.



Obrázek 4.1 Další vytvořené vzory dělení plochy listu

Kromě řeckého kříže se u všech příkazů jako první dva parametry předávají šířka a délka listu. Počet dalších parametrů je individuální u každého vzoru. Jedná se o parametry určující vnitřní poměry jednotlivých prvků, které daný vzor obsahuje a také barvy, které vyplňují jednotlivé části vzoru. V případě řeckého kříže jsou pak jako první dva parametry předány výška a šířka kříže, dále je rozdíl oproti příkazům jiných vzorů v tom, že jsou jako parametr předány souřadnice, které určují polohu kříže na listu. Princip zbývajících parametrů je totožný s ostatními vzory.

Při programování vzorů bylo nejprve nutno určit souřadnice bodů, kterými procházejí jednotlivé části vzoru. Pro zápis bodů byly využity lokální proměnné $z_{\langle index \rangle}$, které reprezentují bod o souřadnicích $x_{\langle index \rangle}$ a $y_{\langle index \rangle}$ a umožňují přistupovat ke každé složce zvlášť. Zápis x -ové a y -ové souřadnice bodu lze provést například následujícím způsobem:

```
x1:=0; y1:=1;
```

Tento zápis umožní používat lokální proměnnou $z1$, která je typu `pair` a jejíž souřadnice reprezentují vzdálenost bodu od referenčního bodu – počátku souřadnic.

Při zápisu souřadnic bylo rovněž využito schopnosti MetaPostu řešit lineární rovnice obsahující páry souřadnic. Ke zjištění požadovaného bodu ležícího na přímce procházející dvěma body byl využíván následující zápis:

```
z3=1/3[z1,z2];
```

který v tomto případě určí souřadnice bodu $z3$ tak, aby byly v jedné třetině vzdálenosti mezi body $z1$ a $z2$.

V případě potřeby určit průsečík dvou přímek, kde je každá určena dvěma body, bylo při řešení soustavy lineárních rovnic využíváno pomocné makro `whatever`, které jak uvádí HOBBY (2014) vygeneruje při každém zavolání novou anonymní proměnnou. Následující část kódu:

```
z9=whatever[z20,z21]=whatever[z3,z4];
```

vypočítá průsečík dvou přímek a jeho souřadnice uloží do proměnné $z9$.

Po získání všech potřebných bodů proběhlo jejich spojení do lomených čar, pomocí uvedení příkazu `--` mezi jednotlivými body. Části vzorů byly následně vyplněny barvou pomocí příkazu `fill`. Aby bylo možno příkaz použít, musely být cesty nejdříve uzavřeny, čehož bylo dosaženo pomocí uvedení příkazu `--cycle` na konci výčtu spojených bodů.

Následující příkaz obsahuje kód v jazyce MetaPost pro vykreslení horizontální trikolóry:

```
\def\@HorizontalTricolor[#1][#2][#3][#4][#5]{%
    flagWidth:=#1;
    flagLength:=#2;
    x1:=x6:=x7:=x8:=y1:=y2:=0;
    x2:=x3:=x4:=x5:=flagLength;
    y5:=y6:=flagWidth;
    z3=1/3[z2,z5];z4=2/3[z2,z5];
    z7=2/3[z1,z6];z8=1/3[z1,z6];
    fill z7--z4--z5--z6--cycle withcolor \MPcolor{#3};
    fill z8--z3--z4--z7--cycle withcolor \MPcolor{#4};
    fill z1--z2--z3--z8--cycle withcolor \MPcolor{#5};
}
```

4.5 Ohraničení vlajky a praporu

Při tvorbě příkazu `\@FlagFrame`, který zajišťuje ohraničení vlajky a praporu, musela být vzata v potaz skutečnost, že příkaz `draw` standardně pracuje s perem, které obsahuje proměnná `currentpen`. Tato proměnná má jako výchozí nastaveno pero s kruhovým průřezem (`pen_circle`), které se vzhledem k výsledným zaobleným rohům nehodí k vytvoření ohraničení.

Uvedením volby `withpen pensquare` za vykreslovanou cestou při použití příkazu `draw` bylo docíleno, že jsou cesty vykreslovány perem, jehož stopa odpovídá jednotkovému čtverci, díky čemuž jsou výsledné rohy ohraničení ostré.

4.6 Svislé verze vlajek a praporů

Jestliže chceme získat svislou vlajku či prapor, v některých případech stačí otočit její standardní horizontální verzi o 90° po směru hodinových ručiček. Jsou ale i jiné případy, kdy je nutné navíc celou vlajku či prapor zrcadlově obrátit, např. svislá verze vlajky České republiky musí mít bílou barvu vždy vlevo.

Z uvedených důvodů byly vytvořeny příkazy, které zajišťují požadovanou podobu svislé vlajky pomocí aplikace vhodných transformací na speciální proměnnou s názvem `currentpicture`, která obsahuje výsledky předešlých příkazů pro kreslení.

Další problém u svislých verzí byl, že vlajky a prapory některých států, např. Slovenska, mají i u svislé verze znak v přirozené poloze a s rozdílným umístěním na listu. Bylo tedy nutné vytvořit transformovaný znak i pro svislé verze vlajek a praporů.

4.7 Barvy

V materiálech, ze kterých jsem čerpal, nebyly barvy vždy přesně definovány nebo definovány byly, ale v nevhodných barevných prostorech. Například v případě Polska je použit barevný prostor CIE 1976 (Ustawa..., 1980) a naprostá většina zemí používá k definici barev pouze přímé barvy Pantone. Dále řada zemí například i Česká republika a Slovensko mají barvy popsány pouze slovně bez jakýchkoliv upřesňujících informací, což vede k tomu, že jsou vlajky často prezentovány v různých barevných odstínech.

Pro definici barev tedy bylo potřeba provést převod mezi jednotlivými barevnými prostory a vyhledat alternativy k přímým barvám Pantone.

Na internetu je dostupná celá řada nástrojů sloužících pro převod mezi barevnými prostory RGB a CMYK, avšak nejsou známy všechny technické parametry těchto převodů. Vzhledem ke skutečnosti, že barevné prostory jsou založeny na jiných způsobech míchání barev a mají rozdílný gamut, nemůže být konverze nikdy zcela přesná. Rozhodl jsem se tedy, že převody budu realizovat v programu Inkscape, ve kterém jsem rovněž pracoval při tvorbě znaků a který touto funkcí disponuje.

Při práci s přímými barvami Pantone byly informace čerpány z oficiálních stránek (Pantone, 2016), kde jsou doporučeny alternativy v barevných prostorech CMYK a RGB.

ConTeXt umožňuje definovat barvy pomocí příkazu `\definecolor` v barevných prostorech CMYK i RGB. Rozhodl jsem se definovat barvy v obou modelech. Barvy definované v ConTeXtu jsou následně v MetaPostu přístupné prostřednictvím příkazu `\MPcolor`, kde se jako parametr předá název barvy. Při testování modulu však byla zjištěna nepříjemná skutečnost.

Jestliže v ConTeXtu pracujeme v barevném modelu RGB (nastaven jako výchozí), nejsou žádné problémy. Jinak je tomu při práci v modelu CMYK. Opakovaným pokusy bylo zjištěno, že pokud je barva definovaná pomocí příkazu `\definecolor` v obou barevných modelech, při práci v barevném modelu CMYK s barvami nastavenými příkazem `\setupcolors`, příkaz `\MPcolor` stále čerpá informace o barvě pouze z definovaného barevného modelu RGB.

Další problém byl u bílé barvy. V případě, že byly v definici bílé barvy všechny složky barevného modelu CMYK rovny nule a všechny složky barevného modelu RGB jedné, tak při nastavené práci v modelu CMYK se bílá barva, předaná pomocí příkazu `\MPcolor`, chybně konvertovala na černou. Provizorní řešení spočívá ve zmenšení hodnoty jedné složky modelu RGB při definování o minimální hodnotu. Následný výstup poskytuje bílou barvu bez známky pigmentu.

Při pátrání po příčinách chybného chování příkazu `\MPcolor` byla prostudována řada zdrojů a chybné chování pravděpodobně vychází z historického vývoje MetaPostu. HOBBS (2014, s. 15) uvádí, že současná verze MetaPostu podporuje barevný model CMYK. Dřívější verze ho však nepodporovaly (GROGONO, 2014, s. 12). V manuálu MetaFun HAGEN (2010, s. 40) uvádí, že MetaPost podporuje pouze barevný model RGB. Dále u popisu příkazu `\MPcolor` HAGEN (2010, s. 120) uvádí, že vzhledem ke skutečnosti, že MetaPost nepodporuje barevný prostor CMYK, se o konverzi barevných modelů stará příkaz `\MPcolor`.

Přestože tedy byly barvy v rámci rozhraní nastavovány korektně, implementačně není správa barev v systému dořešena a vznikají výše popsané problémy.

4.8 Tvorba znaků

Některé národní vlajky obsahují znak, proto bylo nutné tyto znaky vytvořit. Celkem bylo potřeba vytvořit patnáct znaků, od jednoduchých jaké obsahují například vlajky Evropské unie a Slovenska, až po velmi komplikované znaky, které se vyskytují například na vlajkách zemí Španělska, Portugalska či Chorvatska. Tyto znaky se skládají z mnoha geometrických obrazců, heraldických a obecných figur a v některých případech obsahují také textové části.

Tvorba znaků probíhala ve třech fázích. V první řadě bylo třeba připravit znaky jednotlivých zemí tak, aby jejich následná práce s nimi byla co nejpohodlnější. Dalším krokem byl postupný popis znaků v programovacím jazyku MetaPost a poslední fází bylo jejich začlenění do vlajek.

4.8.1 Použitý software

Nejprve muselo být vybráno vhodné programové vybavení. Pro práci s grafikou byl zvolen program Inkscape. Jedná se o volně šiřitelný a multiplatformní program, který je primárně určen pro práci s vektorovou grafikou, ale v omezené míře nabízí i základní možnosti práce s grafikou rastrovou.

Při psaní kódů v jazyce MetaPost byl pro usnadnění práce využit program MetaPost Editor and Previewer (MEPer). Jde také o volně šiřitelný program, který je napsán v jazyce Java, takže je rovněž multiplatformní. Uživatelské rozhraní tohoto programu je rozděleno na dvě části. Levá část je určena pro psaní kódu v jazyce MetaPost včetně zvýraznění syntaxe a v pravé části se po stisknutí tlačítka *Preview*, které spustí překlad kódu, zobrazí výsledný obrázek.

4.8.2 Příprava podkladu

U vlajek, které jsem vytvořil, byl přesný geometrický popis všech prvků obsažených ve znaku dostupný pouze u vlajek Ázerbájdžánu, Turecka, Gruzie a vlajky Evropské unie. V ostatních případech byl proveden import znaku do programu Inkscape, díky čemuž bylo možné následně zjistit rozměry a souřadnice jednotlivých grafických prvků. Zdroje, ze kterých jsem při práci čerpal, jsou uvedeny v seznamu zdrojů informací o parametrech vlajek na konci práce.

V případech, kdy nebyl k dispozici samostatný znak v dostatečné kvalitě nebo nebyla k dispozici identická verze znaku, která se vyskytuje na vlajce, bylo nejprve nutné provést separaci znaku od zbytku vlajky.

U zdrojových podkladů ve vektorovém formátu byla použita možnost *Zrušit seskupení*, která umožní manipulaci s jednotlivými částmi vektorového obrázku. Následně byla upravena velikost znaku se zapnutou volbou *Zamčení stavu*, díky čemuž se šířka i výška měnily ve stejném poměru. Nakonec byl znak umístěn k počátku souřadnic.

4.8.3 Tvorba znaků pomocí MetaPostu

Další částí práce byl zápis zdrojových kódů jednotlivých znaků. Na začátku kódu každého znaku je přiřazena do proměnné `u` hodnota, která se vypočítá na základě požadovaného rozměru výsledné vlajky, případně i měřítka, pokud byly rozměry prvků či souřadnice bodů opisovány z programu Inkscape. Tato proměnná byla následně používána při zápisu souřadnic v MetaPostu jako převodní faktor, díky čemuž je možné ovlivňovat velikost výsledného obrázku pouze změnou hodnoty dané proměnné `u`. V několika případech byl rozměr znaku závislý jak na šířce, tak i na délce výsledné vlajky, bylo proto nutné zavést i druhou proměnnou `g`, se kterou se pracovalo totožně jako s proměnnou `u`.

Následně byly provedeny deklarace proměnných, případně i jejich inicializace. Zdrojové kódy jednotlivých znaků neobsahují deklarace zmíněných proměnných `u` a `g`, jelikož je-li použita proměnná bez deklarace, MetaPost automaticky předpokládá, že jde o proměnnou typu `numeric` (GROGONO, 2014).

V případech, kdy znak obsahoval mnoho grafických prvků, probíhala práce na znaku postupně a pro jednotlivé části znaku byly vytvořené proměnné typu `picture`, což zpřehlednilo práci a usnadnilo následnou manipulaci s částmi obrázku.

Konstrukce lomených čar probíhala stejným postupem, jaký byl použit při tvorbě vzorů dělení listu. Dále byly využity předdefinované cesty `fullcircle`, `halfcircle` a `quartercircle`, které slouží k definici jednotkové kružnice a jejích částí.

Křivky složitých grafických prvků bylo nutné popsat jednotlivými body. Například křivku, která prochází čtyřmi body, vytvoří následující příkaz:

```
z1..z2..z3..z4
```

Jestliže jsou uvedené body definovány, MetaPost vytvoří křivku z na sebe navazujících Bézierových křivek a sám vypočítá jejich kontrolní body (HOBBY, 2014). Vznikne tedy hladká křivka, která prochází zadaným sledem bodů.

MetaPost poskytuje několik možností jak upravit výsledný tvar cesty. Byla využívána schopnost ovlivnit zakřivení cesty pomocí nastavení parametru `curl`. Nabízela se také možnost opsat souřadnice kontrolních bodů Bézierových křivek přímo z programu Inkscape nebo zdrojových kódů podkladů, které byly ve vektorovém formátu,

avšak tvar výsledných křivek vykreslený MetaPostem byl odlišný a zcela neodpovídal křivkám původním. Podobný problém popsala i FIŠEROVÁ (2008) při práci s META-FONTEM.

Pro další operace s cestami byly využívány příkazy pracující s parametrem t , tzv. časem. Jedná se o parametr udávající počet částí křivky. Je-li například křivka zadána třemi body, tak hodnota času křivky v prvním bodě je 0 a v posledním 2. Po provedení následujícího příkazu:

```
(t1,t2)= w4 intersectiontimes w3;
```

obsahuje proměnná $t1$ čas na cestě $w4$, ve kterém tato cesta protíná cestu $w3$ a analogicky čas na cestě $w3$ obsahuje proměnná $t2$. Mezi další využívané příkazy pracující s časem patří příkaz `subpath` a na něm založené příkazy `cutbefore` a `cutafter`. Příkaz `subpath` slouží k získání části cesty mezi dvěma časy a příkazy `cutbefore` a `cutafter` k získání částí úseku cesty před nebo za průsečíkem dvou cest.

U osově souměrných cest nebo částí znaku uložených v proměnné typu `picture` byla využívána transformace `reflectedabout`, která vytvoří osově souměrný objekt, což výrazně zjednodušilo práci. Následující ukázka kódu:

```
e6:=e5 reflectedabout (up,down);
```

přihadí do proměnné $e6$ transformovanou proměnnou $e5$ podle přímky zadané dvěma body. V uvedeném příkladu byly využity MetaPostem předdefinované body a uvedený zápis je s následujícím ekvivalentní:

```
e6:=e5 reflectedabout ((0,1),(0,-1));
```

Kromě transformace `reflectedabout` byly využívány i ostatní transformace, které MetaPost nabízí. Následující kód:

```
a96 rotatedaround ((0,776.64u),90);
```

otočí proměnnou $a96$ o 90° proti směru hodinových ručiček kolem středu zadaného bodem $(0,776.64u)$.

Změny velikosti objektu bylo dosaženo transformací `scaled`, pokud byla potřeba změna pouze ve směru jedné osy, byly využity transformace `xscaled` a `yscaled`. Dále byly využívány transformace `rotated` pro otočení kolem počátku souřadnic, `slanted` pro zkosení a `shifted` pro posunutí objektu.

Několikrát byla využita možnost definice vlastní transformace pomocí zadání třech bodů a jejich obrazů, jak je uvedeno v následující ukázce:

```
transform T;
(10.79u,23.83u) transformed T=(9.03u,76.14u);
(54.69u,30.76u) transformed T=(45.08u,80.48u);
(57.02u,80.26u) transformed T=(44.39u,121.86u);
```

Sestavení uzavřených cest u lomených čar probíhalo pomocí již zmíněného příkazu `--cycle`, v případě křivek šlo o příkaz `. . cycle` a pokud na sebe jednotlivé úseky cest bezprostředně navazovaly, byl použit příkaz `&cycle`. Při uzavírání dvojice cest, kde

jedna vznikla transformací `reflectededabout`, bylo pro korektní výsledek nejprve nutné u jedné z křivek změnit orientaci pomocí příkazu `reverse`.

Vykreslení a vyplnění cest probíhalo již zmíněnými příkazy `draw` a `fill`. Pokud byly cesty přidávány do proměnné typu `picture`, byly využívány příkazy `contour` a `doublepath`. Při provedení následujícího kódu:

```
addto crown3 contour a32 withcolor \MPcolor{white};
```

se do proměnné `crown3` uloží vyplněná plocha uzavřené křivky `a32` a příkazem

```
addto crown3 doublepath a32 withcolor \MPcolor{black};
```

se do téže proměnné uloží obrys křivky `a32`.

Princip tvorby jednotlivých znaků a jejich umístění na vlajce demonstruje následující příklad programování znaku, který obsahuje vlajka Evropské unie. V příkladu jsou jednotlivé prvky vytvářeny kolem referenčního bodu a následně uloženy do proměnné `coatOfArms`, která je nakonec pomocí transformace `shifted` posunuta na požadované umístění znaku na vlajce a vykreslena pomocí příkazu `draw`.

```
u:=(1/2*flagWidth);
pair a[],c[]; path b[],d;
picture coatOfArms;
coatOfArms := nullpicture;
a1:=(0,1/9u);
for i:=2 step 1 until 6:
  a[i]:=a[i-1] rotated 72;
endfor;

a7:= whatever [a1,a3]= whatever [a2,a4];
for i:=8 step 1 until 11:
  a[i]:=a[i-1] rotated 72;
endfor;

d:= a1--a11--a2--a7--a3--a8--a4--a9--a5--a10--a6--cycle;
c1:= (2/3u,0);
for i:=2 step 1 until 12:
  c[i]:=c[i-1] rotated 30;
endfor;

for i:=1 step 1 until 12:
  b[i]:= d shifted (c[i]);
endfor;

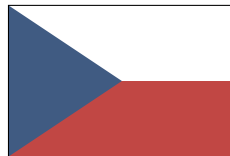
for i:=1 step 1 until 12:
  addto coatOfArms contour b[i] withcolor \MPcolor{EUGold};
endfor;
draw coatOfArms shifted ((1/2*flagLength),(1/2*flagWidth));
```

5 Ukázka generování vlajek a praporů

V této kapitole se nachází ukázky několika vlajek a praporů, které byly vygenerovány pomocí vytvořeného modulu.

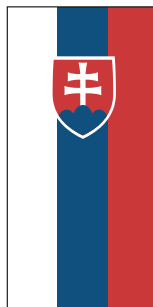
První ukázka kódu vygeneruje vlajku České republiky s délkou 3 cm a ohraničením.

```
\flag[cze] [length=3cm, frame=yes]
```



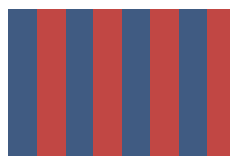
Druhá ukázka kódu vygeneruje ohraničený svislý prapor Slovenska s délkou 4 cm a šířkou 2 cm.

```
\banner[sk] [length=4cm, width=2cm, orientation=portrait, frame=yes]
```



Poslední ukázka demonstruje doplňkovou možnost vygenerování jednoduché vlajky podle některého z 20 předdefinovaných vzorů.

```
\defineflag[18] [width=2cm, length=3cm] [8] [blue] [red]
```



6 Diskuze

Modul najde uplatnění samozřejmě všude tam, kde je potřeba, aby součástí dokumentu byla vlajka. V běžných případech jsme nuceni vyhledávat externí soubory v dostatečné kvalitě a následně ověřovat, zda se jedná o aktuální verzi vlajky a se správným poměrem stran. S využitím modulu lze místo toho jediným příkazem vygenerovat požadovanou vlajku ve vektorovém formátu, správném poměru stran, požadované velikosti, případně i s ohraničením.

Funkce modulu, která umožňuje generovat prapory, může najít uplatnění všude tam, kde je třeba vytvořit sadu praporů několika států se stejnými rozměry. Vygenerovanými prapory lze například doplnit a zpřehlednit tabulky, které porovnávají ekonomické ukazatele několika zemí, případně prezentaci výsledků z mezinárodních soutěží, kde jsou pro lepší orientaci u jmen účastníků uvedeny i prapory států, z kterých pocházejí. Vzhledem ke skutečnosti, že vlajky států mají různé poměry stran, vznikají při jejich použití jako praporů se stejným poměrem stran deformace jejich vnitřních prvků. Jde zpravidla o deformace znaku, který vlajka obsahuje. Dalším typem deformace, která může nastat při změně originálního poměru stran jsou například rozdílné šířky jednotlivých ramen kříže obsaženého na praporu. Najít takovou sadu praporů s požadovaným poměrem stran a bez deformovaných vnitřních prvků by bylo bez použití modulu obtížné.

V průběhu řešení práce jsem musel čelit celé řadě problémů, které se postupně objevovaly, od získání relevantních podkladů až po problém s barvami, který byl popsán v kapitole 4.7. Nejtíživější problém však nastal při rekonstrukci znaků, které se vyskytují na vlajkách. Obecně je ve vexilologii doporučováno zjednodušovat znaky obsažené na vlajce kvůli jejich snadné reprodukci. Znaky obsažené na vlajkách některých zemí, které bylo třeba vytvořit, jsou však značně složité, obsahují velké množství geometrických obrazců, heraldických a obecných figur. Implementace těchto znaků se v MetaPostu ukázala jako obtížná a velmi časově náročná. Mezi nejsložitější patřily bezesporu znaky na vlajkách Španělska, Srbska, Chorvatska, Portugalska či Malty, jejichž vytvoření trvalo velmi dlouhý čas.

Modul je koncipován obecně a umožňuje rozšiřování o vlajky a prapory dalších zemí, případně i nové vzory dělení plochy listu vlajky, což je požadavkem na moduly toho typu.

7 Závěr

Cílem této bakalářské práce byla tvorba vexilografického modulu pro typografický systém ConT_EXt. Pro úspěšné zvládnutí tohoto cíle bylo nutné provést celou řadu kroků.

V první části práce proběhlo seznámení s vexilologií, byla stručně popsána její historie a používaná odborná terminologie. Dále byly popsány nejpoužívanější způsoby dělení plochy vlajkového listu.

Druhá část práce obsahuje základní informace o typografickém systému T_EX a jeho formátu ConT_EXt, který byl srovnán s T_EXovou konkurencí. V této části jsem se rovněž seznámil s programovacími jazyky MetaPost a Lua.

Další část práce popisuje postup implementace modulu, která se skládala z několika částí. Byly vytvořeny všechny potřebné příkazy a podprogramy. Dále bylo v MetaPostu vytvořeno dvacet základních způsobů dělení plochy listu a všechny potřebné znaky, které obsahují vlajky některých zemí.

Podařilo se vytvořit funkční modul pro ConT_EXt, který umožňuje generovat nejen vlajky a prapory 28 členských států Evropské unie včetně vlajky a praporu Unie, ale nad rámec stanoveného cíle byly definovány vlajky a prapory pro dalších 14 geograficky blízkých zemí. Dále modul obsahuje doplňkovou funkci, která umožňuje vygenerovat jednoduchou vlajku z 20 nabízených vzorů.

V budoucnu je plánováno postupné rozšiřování modulu o vlajky a prapory dalších států, vyřešení správy barev, doplnění o možnost nastavení typu ohraničení vlajky a vytvoření anglické dokumentace.

Literatura

- Co je vexilologie?* [on-line]. 2015. [cit. 2016-03-17]. Dostupné na: http://www.vlajky.cz/informace/informace_vexilologie_co_je_vexilologie.php.
- CONTEXT GARDEN. *MkIVDifferences* [on-line]. 2016. [cit. 2016-04-13]. Dostupné na: http://wiki.contextgarden.net/MkIV_Differences.
- Dánský Dannebrog: nejstarší stále užívaná vlajka státu světa* [on-line]. 2009. [cit. 2016-01-15]. Dostupné na: <http://www.ceskatelevize.cz/ct24/archiv/1402576-dansky-dannebrog-nejstarsi-stale-uzivana-vlajka-statu-sveta>.
- EXNER, P., FOJTÍK, P., SVOBODA, Z. *Vlajky, prapory a jejich používání: základní pravidla pro vyvěšování vlajek na území České republiky*. Liberec : Libea, 2004. 63 s. ISBN 80-239-2873-2.
- FIŠEROVÁ, Š. *Rekonstrukce a digitalizace florentské bastardy*. (Bakalářská práce.) Brno : Mendelova univerzita v Brně, 2008. 42 s.
- FRAJDL, J., ZÁLABSKÝ, J. *Vexilologie*. Seč : Pedagogická fakulta Hradec Králové, Ústřední politická škola SSM Seč, 1989. 127 s.
- GROGONO, P. *MetaPost: A Reference Manual* [on-line]. 2014. [cit. 2016-03-03]. 72 s. Dostupné na: <https://users.encs.concordia.ca/~grogono/Writings/mpref.pdf>.
- HAGEN, H. *ConTeXt the manual* [on-line]. 2000. [cit. 2016-03-21]. 370 s. Dostupné na: <http://www.ctex.org/documents/context/cont-enp.pdf>.
- HAGEN, H. *MetaFun* [on-line]. 2010. [cit. 2016-04-11]. 376 s. Dostupné na: <http://www.pragma-ade.com/general/manuals/metafun-p.pdf>.
- HAGEN, H., HOEKWATER, T. *ConTeXt reference manual* [on-line]. 2013. [cit. 2016-03-16]. 376 s. Dostupné na: <http://pmrb.free.fr/contextref.pdf>.
- HOBBY, J. H. *MetaPost: A user's manual* [on-line]. 2014. [cit. 2016-02-25]. 112 s. Dostupné na: <https://www.tug.org/docs/metapost/mpman.pdf>.
- HOEKWATER, T. *Comparing ConTeXt and L^AT_EX* [on-line]. 20/1998. [cit. 2016-02-12]. Dostupné na: <https://www.ntg.nl/maps/20/42.pdf>.
- IERUSALIMSCHY, R. *Programming in Lua*. 3. vyd. Rio de Janeiro : Lua.org, 2013. 348 s. ISBN 978-85-903798-5-0.
- IERUSALIMSCHY, R., FIGUEIREDO, L. H., WALDEMAR, C. *Proceedings of V Brazilian Symposium on Programming Languages* [on-line]. *The evolution of an extension language: A history of Lua*, 2001/5, s. 14–28.. [cit. 2015-12-11]. Dostupné na: https://www.researchgate.net/profile/Waldemar_Celes/publication/2486970_The_Evolution_of_an_Extension_Language_A_History_of_Lua/links/5465e5170cf2052b50a0745a.pdf.
- Inkscape* [on-line]. Ver. 0.91. 2016. Dostupné na: <https://inkscape.org/cs/>.

- ISO. *Online Browsing Platform (OBP)* [on-line]. [cit. 2016-03-25]. Dostupné na: <https://www.iso.org/obp/ui/#search/code/>.
- KNUTH, D. Remarks to celebrate the publication of Computers & Typesetting [on-line]. *TUGboat*, 1986, s. 95–98.. [cit. 2015-12-02]. (ISSN 0896-3207.) Dostupné na: <http://www.tug.org/TUGboat/tb07-2/tb15knut.pdf>.
- LONDON ORGANISING COMMITTEE OF THE OLYMPIC GAMES AND PARALYMPIC GAMES LIMITED. *Flags and anthems manual London 2012* [on-line]. 2012. [cit. 2016-04-03]. Dostupné na: http://doc.rero.ch/record/32076/files/LOC2012_ITR1201_Flags_Manual_Supplement.pdf.
- LUA. *Lua: about* [on-line]. 2016. [cit. 2016-02-10]. Dostupné na: <https://www.lua.org/about.html>.
- MUCHA, L., VALÁŠEK, S. *Vlajky a znaky zemí světa*. 3. vyd. Praha : Geodetický a kartografický podnik, 1987. 215 s.
- OLŠÁK, P. *T_EXbook naruby*. 2. vyd. Brno : Konvoj, 2001. 468 s. ISBN 80-7302-007-6.
- OLŠÁK, P. *Typografický systém T_EX*. 2. vyd. Brno : Konvoj, 2000. 300 s. ISBN 80-85615-91-6.
- OTTEN, T., HAGEN, H. *Exkurze do ConT_EXtu* [on-line]. 2006. [cit. 2016-03-22]. Dostupné na: <http://bulletin.cstug.cz/pdf/ma-cb-cz-print-bw.pdf>.
- PANTONE. *Find a Pantone Color* [on-line]. [cit. 2016-04-24]. Dostupné na: <https://www.pantone.com/color-finder/>.
- SHENGJUN P. *MetaPost Editor and Previewer* [on-line]. Ver. 1.0. 2009. Dostupné na: <http://cseweb.ucsd.edu/~span/MEPer/>.
- Support* [on-line]. 2016. [cit. 2016-04-13]. Dostupné na: <http://www.luatex.org/support.html>.
- SVOBODA, Z. *Vexilologické názvosloví*. Jílové u Prahy : Heraldická sekce při MTZZ, 1972. 11 s.
- Ustawa z dnia 31 stycznia 1980 r. o godle, barwach i hymnie Rzeczypospolitej Polskiej*. 1980.
- What is FIAV?* [on-line]. 2015. [cit. 2016-02-09]. Dostupné na: <http://fiav.org/what-is-fiav/>.
- Zákon č. 3/1993 Sb., o státních symbolech České republiky*. 1993.

Zdroje informací o parametrech vlajek

Albánie

FLAGS OF THE WORLD. *Albania* [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: <https://flagspot.net/flags/al.html>.

Arménie

[*General Specifications of the Republic of Armenia flag*] [on-line]. 2012. [cit. 2016-05-10]. Dostupné na: <http://yeraguyn.com/wp-content/uploads/2010/04/HST-50-2012.pdf>.

Ázerbájdžán

AZƏRBAYCAN RESPUBLİKASININ MILLI MƏCLISI. *Azərbaycan Respublikası Dövlət bayrağının istifadəsi qaydaları haqqında* [on-line]. 2013. [cit. 2016-05-10]. Dostupné na: <http://www.meclis.gov.az/?/az/legislation/view/2485>.

Belgie

Les drapeaux de la Belgique [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: http://www.belgium.be/fr/la_belgique/connaitre_le_pays/la_belgique_en_bref/symboles/drapeaux.

Bělorusko

WIKIPEDIA. *Flag of Belarus* [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: https://en.wikipedia.org/wiki/Flag_of_Belarus.

Bosna a Hercegovina

CRW FLAGS. *Bosnia and Herzegovina* [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: <http://www.crwflags.com/fotw/flags/ba.html>.

Bulharsko

COUNCIL OF MINISTERS OF THE REPUBLIC OF BULGARIA. *About Bulgaria* [on-line]. 2005. [cit. 2016-05-10]. Dostupné na: <http://www.government.bg/cgi-bin/e-cms/vis/vis.pl?s=001&p=0159&n=000006&g=>.

Dánsko

NORDEN. *The Danish flag* [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: <http://www.norden.org/en/fakta-om-norden-1/the-nordic-flags/the-danish-flag>.

Estonsko

RIIGIKANTSELEI. *Eesti lipp* [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: <https://riigikantselei.ee/et/valitsuse-toetamine/eesti-riigi-sumboolika/eesti-lipp>.

Evropská unie

Graphics guide to the European emblem [on-line]. 2015. [cit. 2016-05-10]. Dostupné na: <http://publications.europa.eu/code/en/en-5000100.htm>.

Finsko

NORDEN. *The Finnish flag* [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: <http://www.norden.org/en/fakta-om-norden-1/the-nordic-flags/the-finnish-flag>.

Francie

CRW FLAGS. *France* [on-line]. 2013. [cit. 2016-05-10]. Dostupné na: <http://www.crwflags.com/fotw/flags/fr.html>.

Gruzie

CRW FLAGS. *Georgia* [on-line]. 2013. [cit. 2016-05-10]. Dostupné na: <http://www.crwflags.com/fotw/flags/ge.html>.

Chorvatsko

HRVATSKI SABOR. *Grafički standardi zastave Republike Hrvatske* [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: <http://www.sabor.hr/Default.aspx?sec=4317>.

Irsko

Constitution of ireland [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: http://www.taoiseach.gov.ie/eng/Historical_Information/The_National_Flag/The_National_Flag.pdf.

Island

NORDEN. *The Icelandic flag* [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: <http://www.norden.org/en/fakta-om-norden-1/the-nordic-flags/the-icelandic-flag>.

Itálie

Manuale di utilizzo del logotipo [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: <http://www.bsnstrategies.com/files/manualebita.pdf>.

Kypr

ΠΡΟΕΔΡΟΣ ΤΗΣ ΔΗΜΟΚΡΑΤΙΑΣ. *Η Κυπριακή Σημαία* [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: http://www.presidency.gov.cy/presidency/presidency.nsf/prc24_gr/prc24_gr?OpenDocument.

Litva

Lietuvos respublikos valstybinės vėliavos ir herbo vektorizuoti etalonai [on-line]. 2007. [cit. 2016-05-10]. Dostupné na: https://lr.v.lt/uploads/main/documents/files/Naudinga_informacija/Valstybes_ir_vyriausybes_simboliai/LR_veliava_ir_herbas.pdf.

Lotyšsko

National Symbols of Latvia [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: <http://www.consulate.lv/national-symbols.html>.

Lucembursko

EMBLEMES DU GRAND-DUCHE DE LUXEMBOURG [on-line]. 1993. [cit. 2016-05-10]. Dostupné na: <http://www.legilux.public.lu/leg/a/archives/1993/0073/a073.pdf#page=3#page=3>.

Makedonie

CRW FLAGS. *Macedonia* [on-line]. 2015. [cit. 2016-05-10]. Dostupné na: <http://www.crwflags.com/fotw/flags/mk.html>.

Malta

OFFICE OF THE PRIME MINISTER MALTA. *Graphical Specifications for the National Flag of Malta* [on-line]. 2008. [cit. 2016-05-10]. Dostupné na: <https://www.gov.mt/en/About\%20Malta/Documents/National\%20Flag\%20Graphical\%20Specifications.pdf>.

Německo

Bundesregierung Corporate Design [on-line]. 2007. [cit. 2016-05-10]. Dostupné na: https://web.archive.org/web/20070926233554/http://styleguide.bundesregierung.de/index_de.html?Content=basiselemente/farben/primaerfarben_de.html.

Norsko

NORDEN. *The Norwegian flag* [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: <http://www.norden.org/en/fakta-om-norden-1/the-nordic-flags/the-norwegian-flag>.

Portugalsko

WIKIPEDIA. *Bandeira de Portugal* [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: https://pt.wikipedia.org/wiki/Bandeira_de_Portugal.

Rumunsko

CRW FLAGS. *Romania* [on-line]. 2013. [cit. 2016-05-10]. Dostupné na: <http://www.crwflags.com/fotw/flags/ro.html>.

Řecko

WIKIPEDIA. *Flag of Greece* [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: https://en.wikipedia.org/wiki/Flag_of_Greece.

Slovensko

VLADA REPUBLIKE SLOVENIJE. *Državni simboli* [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: http://www.vlada.si/o_sloveniji/politichni_sistem/drzavni_simboli/.

Srbsko

Zakona o izgledu i upotrebi grba, zastave i himne Republike Srbije [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: http://www.srbija.gov.rs/extfile/sr/142564/uredba_izvornik_grba_zastave_himne0355_cyr.zip.

Španělsko

WIKIPEDIA. *Flag of Spain* [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: https://en.wikipedia.org/wiki/Flag_of_Spain.

Švédsko

NORDEN. *The Swedish flag* [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: <http://www.norden.org/en/fakta-om-norden-1/the-nordic-flags/the-swedish-flag>.

Švýcarsko

Corporate Identity Manual Switzerland [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: http://www.image-schweiz.ch/fileadmin/user_upload/pdf/f/Marke_Schweiz/Manual_marque_Suisse.pdf.

Turecko

CRW FLAGS. *Turkey* [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: <http://www.crwflags.com/fotw/flags/tr.html>.

Velká Británie

WIKIPEDIA. *Union Jack* [on-line]. 2016. [cit. 2016-05-10]. Dostupné na: https://en.wikipedia.org/wiki/Union_Jack.

Informace o vlajkách a praporech zemí, které nejsou uvedeny v seznamu, byly čerpány z dokumentu *Flags and anthems manual London 2012*, na který je uvedena reference v kapitole Literatura.

Přílohy

A Ukázky vygenerovaných vlajek pomocí modulu



Albánie



Arménie



Ázerbájdžán



Belgie



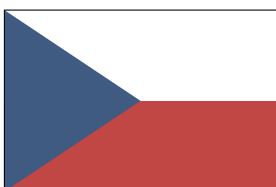
Bělorusko



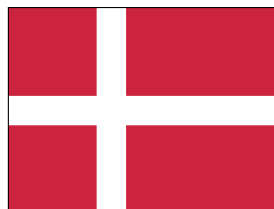
Bosna a Hercegovina



Bulharsko



Česká republika



Dánsko



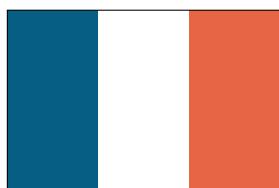
Estonsko



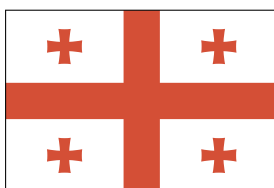
Evropská unie



Finsko



Francie



Gruzie



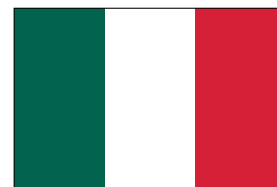
Chorvatsko



Irsko



Island



Itálie



Kypr



Litva



Lotyšsko



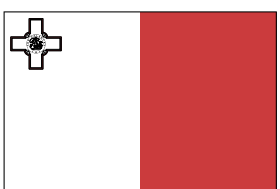
Lucembursko



Maďarsko



Makedonie



Malta



Monako



Německo



Nizozemsko



Norsko



Polsko



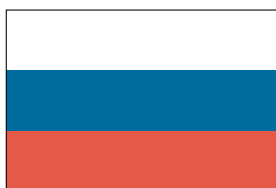
Portugalsko



Rakousko



Rumunsko



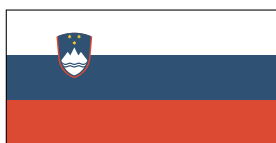
Rusko



Řecko



Slovensko



Slovinsko



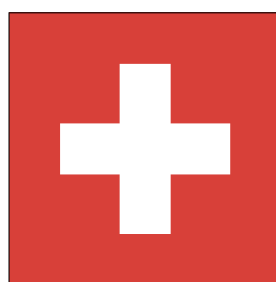
Srbsko



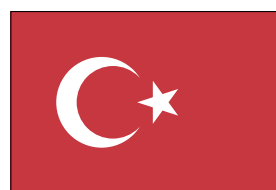
Španělsko



Švédsko



Švýcarsko



Turecko



Ukrajina



Velká Británie

B Příložené CD

Přílohy nacházející se na přiloženém optickém disku:

- modul
- návod na připojení modulu
- seznam podporovaných zemí a jejich ISO kódy
- soubor s ukázkami vygenerovaných vlajek a praporů
- popis příkazů