

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

FORMALIZACE A VALIDACE PŘÍSTUPNOSTI WEBOVÝCH STRÁNEK

DIPLOMOVÁ PRÁCE

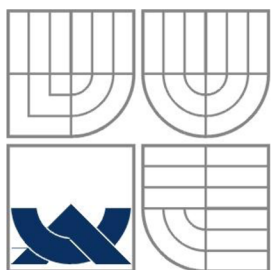
MASTER'S THESIS

AUTOR PRÁCE

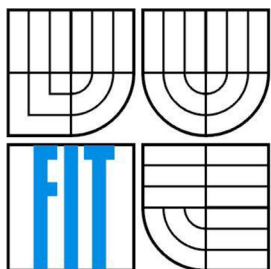
AUTHOR

Ing. Peter Marcely

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

FORMALIZACE A VALIDACE PŘÍSTUPNOSTI WEBOVÝCH STRÁNEK

FORMALIZATION AND VALIDATION OF WEB ACCESSIBILITY

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Ing. Peter Marcely

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Pavel Očenášek, Ph.D.

BRNO 2012

Abstrakt

Uživatelé s různými typy postižení přistupují k webovým stránkám s pomocí různorodých zařízení. Je proto často komplikované vytvořit webové stránky tak, aby byly přístupné pro všechny. Tato práce je zaměřena na vybrané metodiky pro tvorbu přístupného obsahu na webových stránkách, formalizaci jejich pravidel a následnou validaci.

Abstract

Users with different types of handicaps access web pages with wide scale of device. Thus, it is often difficult to develop web sites in order to be accessible for everyone. This thesis takes into consideration specific guidelines for accessible content of web pages as well as formalization of their rules and subsequent validation of these rules.

Klíčová slova

přístupnost, web, formalizace pravidel, validace webové přístupnosti

Keywords

accessibility, web, formalization, web accessibility validation.

Citace

Marcelý Peter: Formalizace a validace přístupnosti webových stránek, diplomová práce, Brno, FIT VUT v Brně, 2012

Formalizace a validace přístupnosti webových stránek

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Pavla Očenáška, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Peter Marcely
20.5.2012

Poděkování

Rád bych poděkoval Ing. Pavlu Očenáškov, Ph.D., za vedení práce a hodnotné podněty při jejím vytváření. Děkuji také rodině a přítelkyni za podporu při studiu.

© Peter Marcely 2012

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

Obsah	1
1 Úvod	4
2 Postihnutia a práca s PC.....	6
2.1 Užívatelia so zrakovým postihnutím.....	6
2.1.1 Slabozrakosť	6
2.1.2 Nevidomosť	6
2.1.3 Farbosleposť	6
2.1.4 Užívatelia s dočasným obmedzením videnia.....	7
2.1.5 Zariadenia pre prácu s PC.....	7
2.2 Pohybovo postihnutí užívatelia	8
2.2.1 Pomôcky pre prácu s PC	8
2.3 Sluchovo postihnutí užívatelia	9
2.4 Užívatelia používajúci alternatívny software.....	9
2.5 Užívatelia s poruchami sústredenia	9
3 Prístupnosť a používané metodiky	10
3.1 Definícia prístupnosti na webe.....	10
3.2 Prečo vytvárať prístupný web?.....	10
3.3 Metodiky prístupnosti	11
3.4 Medzinárodné normy	11
3.4.1 WCAG 1.0.....	11
3.4.2 WCAG 2.0.....	13
3.5 Lokalizované pravidlá prístupnosti.....	14
3.5.1 Česká verzia pravidiel prístupnosti.....	14
3.5.2 Pravidlá Slovenskej republiky	14
3.5.3 Veľká Británia.....	15
3.5.4 Švédsko	15
3.5.5 USA.....	15
3.5.6 Kanada	15
3.5.7 Zhrnutie legislatívnej lokalizácie	15
4 Formalizácia pravidiel.....	17
4.1 Teoretické minimum nutné pre formalizáciu.....	17
4.1.1 HTML.....	17
4.1.2 CSS.....	17
4.1.3 JavaScript	18

4.1.4	XML	19
4.1.5	Gramatiky	19
4.2	Spôsob využitia gramatík pre formalizáciu	20
4.2.1	Jednoduchý príklad využitia gramatiky	20
4.2.2	Príklad na kompletne využitie bezkontextovej gramatiky	21
4.3	Rozbor pravidiel v normách.....	22
4.3.1	Pravidlá prístupnosti Českej republiky	22
4.3.2	WCAG 2.0.....	25
4.3.3	Súhrn rozboru pravidiel.....	26
5	Návrh aplikácie	27
5.1	Analýza existujúcich validátorov	27
5.1.1	W3C validátor	27
5.1.2	Achecker.....	28
5.1.3	WAVE validátor.....	29
5.1.4	Cynthia Says	31
5.1.5	Total Validator	32
5.2	Analýza doplnkových nástrojov	32
5.2.1	Toolbary pre webové prehliadače	32
5.2.2	Kontrola kontrastu farieb.....	33
5.3	Analýza požiadaviek	33
5.3.1	Diagram prípadov použitia	34
5.4	Návrh validátora	35
5.4.1	Voľba frameworku - Django	36
5.4.2	Užívateľské rozhranie	36
5.4.3	Načítanie pravidiel gramatík.....	36
5.4.4	Načítanie webovej stránky.....	37
6	Implementácia.....	39
6.1	Trieda XmlParser.....	39
6.2	Pohľady a šablóny	41
6.3	Trieda PreParser	42
6.4	Trieda GramParser.....	43
6.5	Testovanie.....	44
7	Porovnanie výsledkov s inými validátormi	45
7.1.1	Porovnanie výsledkov pre stránky FIT VUT.....	46
7.1.2	Porovnanie výsledkov pre Seznam.cz.....	47
8	Rozšírenia a možné vylepšenia	48
8.1	Vylepšenie HTML parsera.....	48

8.2	Ukladanie web stránok do databázy.....	48
8.3	Vylepšenie algoritmu spracovávania gramatík	49
8.3.1	Pravidlo č. 22 českej normy	49
8.4	Aplikácia pre tablet a smartphone	50
8.5	Podporný toolbar pre webový browser	50
9	Záver.....	51
	Zoznam príloh.....	54
	Príloha A	55

1 Úvod

Rýchly vývoj Internetu priniesol so sebou masívne rozšírenie nástrojov a služieb, ku ktorým je prístup možný cez webové stránky. Tento, z väčšej časti nekontrolovaný rozvoj, so sebou priniesol rôzne prekážky, ktoré znemožňujú určitým skupinám užívateľom získať prístup k požadovaným webovým službám. Preto sa s postupným rozvojom webových stránok začali vynárať rôzne štandardy, ktoré majú tvorcom webových služieb pomôcť pri ich vývoji.

Ako v reálnom živote nie je žiadané handicapovaných občanov odstrihnúť od prístupu, či už ku komerčným alebo nekomerčným službám, rovnaký princíp platí aj na webe. V prípade nekomerčných aktivít chceme zachovať rovnoprávnosť ľudí a neznemožňovať handicapovaným občanom prístup ku vzdelaniu a kultúre. V prípade komerčných aktivít je na odopretí tejto služby škodný dokonca aj prevádzkovateľ. Handicapovaný zákazník odíde tam, kde mu jeho právo využitia služby nebude odopreté. Rôznymi typmi postihnutí a spôsobmi ako s nimi bojovať pri práci s PC sa zaoberá druhá kapitola.

Samotná webová prístupnosť však nie je len prínosom pre handicapovaných užívateľov. Prehľadnosť a čitateľnosť textu patria k najdôležitejším zásadám prístupnosti, čo v konečnom dôsledku zľahčuje navigáciu aj pre bežných užívateľov. V paralele so skutočným svetom nám napríklad zvukové signály pre chodcov na svetelných križovatkách dávajú možnosť venovať sa inej činnosti - napríklad práci s mobilným telefónom a spoľahnúť sa na sluchový zmysel oznamujúci zmenu stavu križovatky. Metodiky a štandardy, ktoré počas rýchleho vývoja vznikli vo svete a boli prebrané aj do našich krajín, sú popísané v tretej kapitole. Hlavný dôraz sa kladie na medzinárodný štandard WCAG, ktorý bol vo väčšej alebo menšej forme prebraný do metodík viacerých krajín a zároveň zanesený do zákonov s platnosťou pre weby štátnej správy.

V nadväzujúcej štvrtej kapitole sa pozrieme na jednotlivé metodiky podstatne detailnejšie. Dôležitou úlohou je vysvetliť čitateľovi základné postupy a možnosti pri tvorbe webových stránok a zároveň aplikovať ukážky nesprávnych postupov, ktoré narušujú prístupnosť. Následne sú rozobraté možnosti formalizácie a následnej validácie podľa regulárnych výrazov, prípadne bezkontextových gramatík, pričom sú vysvetlené dôvody podporujúce použitie bezkontextových gramatík. Podrobným rozborom českej metodiky pre prístupnosť a medzinárodného štandardu dostane čitateľ kompletný pohľad na tematiku.

Analýzou existujúcich nástrojov určených pre validáciu a nástrojov, ktoré je vhodné používať počas vývoja webových stránok získame také požiadavky validátora, ktoré budú spĺňať základne požiadavky konkurenčných aplikácií. Aplikácia pravidiel gramatík so sebou prináša problémy v tom, že vstup do algoritmu pre ich spracovanie musí byť jednotný. Preto je spracovaný taký návrh riešenia, aby táto podmienka bola splnená.

Implementácia nástroja pre validáciu je popísaná v piatej kapitole. V rámci implementácie sú vysvetlené súvislosti medzi jednotlivými triedami a konštrukciami frameworku, ktorý ponúka využitie pre online aplikácie. V rámci aplikácie je nutné definovať štruktúru XML konfigurácie, ktorá obsahuje pravidlá podporovaných štandardov. Načítanie konfigurácie pravidiel, webovej stránky určenej na validáciu a algoritmus, ktorý tieto dva elementy spojí, tvoria spolu najzákladnejší proces samotnej validácie.

Šiesta kapitola sa zaoberá porovnaním užívateľského rozhrania, funkčnosti a výsledkov validácie s inými validátormi. Referenčnou webovou stránkou určenou pre porovnanie sú webové stránky Fakulty informačných technológií VUT v Brne (FIT VUT) a komerčná stránka spoločnosti Seznam.cz. Výsledky naznačujú podobný prístup k pravidlám a ich vyhodnotenie.

Na možnosti vylepšenia uvedené v šiestej kapitole nadväzuje siedma kapitola s niekoľkými možnosťami rozšírenia. Medzi najpodstatnejšie patria zaistenie lepšieho parsovania zdrojového kódu, optimalizácia algoritmu pre spracovávanie pravidiel gramatík, či ukladanie webových stránok pre opätovné použitie. Tieto vylepšenia by boli zväčša výhodné až pri vyššej záťaži servera a veľkom náraste užívateľov validátora.

Cieľom tejto práce je zanalyzovať súčasný stav štandardov, ktoré sú publikované uznávanými inštitúciami. Následná analýza jednotlivých pravidiel umožňuje formalizáciu týchto konkrétnych pravidiel do gramatiky, ktorá bude postačovať pre formalizáciu problému. Na základe tejto analýzy je vytvorený návrh a následná implementácia aplikácie. Implementovaná aplikácia umožňuje univerzálny vstup formalizovaných pravidiel, ktorý môže mať aj iný účel ako validáciu webovej prístupnosti, hoci vytvorené príklady budú pre jednotlivé štandardy prístupnosti webových stránok. Aplikácia bude zároveň dostupná on-line pre všetkých užívateľov.

2 Postihnutia a práca s PC

Na to, aby sme vedeli pochopiť, pre koho je prístupná webová stránka určená, je potrebné poukázať na rôzne typy handicapov, ktoré ľuďom s postihnutiami zhoršujú, prípadne znemožňujú pohyb po webových službách. V tejto kapitole sa zameriame na konkrétne typy handicapov [1].

2.1 Užívatelia so zrakovým postihnutím

Zrak je pre prácu s počítačom a následne prácu s webovými stránkami kľúčový zmysel. Podľa metodiky webovej prístupnosti je možné zrakové postihnutie rozdeliť do štyroch rôznych skupín.

2.1.1 Slabozrakosť

Slabozrakosť je dočasné alebo trvalé zníženie zrakovej ostrosti. Pre ľudí s menším stupňom tohto zdravotného problému je možné ho riešiť optickými okuliarmi alebo kontaktnými šošovkami.

Riešenie tohto problému je väčšinou vykonávané prostredníctvom zväčšovania textu webovej stránky, prípadne zvyšovaním kontrastu obrazovky. Čitateľný text sa môže prekryvať s inými prvkami stránky alebo blokmi textu. Ideálnym stavom je teda dosiahnutie vysokej čitateľnosti už bez zväčšovania textu, pretože jednak táto možnosť nemusí byť dostupná, alebo ju užívateľ neovláda. Ako ukážkové pravidlo teda vyplýva z týchto skutočností, že je vhodné používať relatívnu veľkosť písma a prvkov na webových stránkach.

2.1.2 Nevidomosť

Nevidomí a inak ťažko zrakovo postihnutí užívatelia patria medzi skupinu užívateľov, ktorí si webové stránky neprehliadajú v bežných prehliadačoch. Web si interpretujú pomocou hlasu alebo hmatu. Čítačky – screen readery slúžia na čítanie výstupu obrazovky, pričom na vstup je využitá bežná alebo upravená klávesnica. Druhou možnosť je braillovský riadok, ktorý spĺňa oba účely – vstup aj výstup z obrazovky.

Na základe uvedených informácií je pochopiteľné, že nie je možné interpretovať grafické prvky pomocou týchto zariadení. Je preto nutné grafickým prvkom určiť ich alternatívny textový význam. Taktiež je vhodné, aby textové bloky nasledovali v logickom poradí a štruktúra stránky nebola zmenená absolútnym pozicovaním technológiou CSS.

2.1.3 Farbosleposť

Samotná farbosleposť sa rozdeľuje na anomálie – zhoršené vnímanie farieb modrého, červeného alebo zeleného spektra a dichromazie – úplnú nemožnosť vnímať konkrétne spektra týchto farieb.

Porucha vnímania farieb komplikuje orientáciu na stránke a môže priniesť problém rozoznať farebne zvýraznené časti textu.

Čiastočným riešením tohto problému je zabezpečenie dostatočného kontrastu pozadia a písma na popredí. Kontrola je možná rôznymi nástrojmi. Najlepší spôsob na zdôraznenie niektorých častí textu je ich zvýraznenie inou formou ako farbou, napríklad podčiarknutím.

2.1.4 Užívatelia s dočasným obmedzením videnia

Medzi týchto užívateľov patria aj zdraví jedinci, ktorých videnie obrazovky je ovplyvnené externými faktormi. Tie faktory zahŕňajú napríklad monitor so zlým zobrazením farieb či zlý obraz, kvôli svetlu dopadajúcemu na monitor. Týmto spôsobom sa teda môže ľahko stať handicapovaným užívateľom aj zdravý užívateľ.

2.1.5 Zariadenia pre prácu s PC

Zrakovo postihnutí užívatelia môžu používať celú škálu zariadení, ktoré im uľahčujú prácu na PC. Medzi tieto zariadenia patria:

- Screen reader – je to zariadenie, ktoré je schopné premeniť vstup z obrazovky na hlasový výstup. Užívateľ ovláda webovú stránku cez virtuálny kurzor na obrazovke, odkiaľ pochádza súčasný časový výstup.
- Braillové riadky (obrázok 2.1) – tieto zariadenia simulujú hmatový výstup obrazovky pomocou špeciálnych ihličiek. Môžu slúžiť aj ako vstupné zariadenie.



Obrázok 2.1: Braillov riadok. Zdroj [2]

- Zväčšovacie lupy – táto softwarová pomôcka je určená pre užívateľov s poškodeným zrakom. Buď to je možné zväčšiť písmo v rámci celej webovej stránky, alebo lupou zväčšiť konkrétnu časť.

2.2 Pohybovo postihnutí užívateľa

Pohybovo postihnutých užívateľov je možné rozdeliť na dve kategórie. Jednak sú ňou užívatelia s trvalými následkami, ktoré vyplývajú z ich chronických chorôb, ako napríklad ochrnutie, Parkinsonova choroba alebo artritída. Druhou kategóriou sú užívatelia zaradení do tejto skupiny dočasne – ľudia, ktorí utrpeli úraz niektorej končatiny.

Z týchto komplikácií vyplýva ľuďom sťažené ovládanie webových stránok myšou. Riešením týchto problémov sú špeciálne upravené klávesnice, ktorých výstup je však rovnaký ako výstup z normálnej klávesnice. Tvorcovia by teda mali optimalizovať svoje webové stránky tak, aby boli použiteľné len klávesnicou. Nespoliehať sa teda na JavaScriptové udalosti ako onmouseover apod., ktoré sú spúšťané interakciou s myšou.

2.2.1 Pomôcky pre prácu s PC

Pohybovo postihnutí užívatelia majú zväčša problém používať klasické vstupné zariadenia – najmä myš, v niektorých prípadoch aj klávesnicu. K dispozícii sú im preto rôzne pomocné zariadenia:

- Trackball (obrázok 2.2) – je to zariadenie, ktorým je možné ovládať kurzor na obrazovke. Samotný trackball je možné dokonca používať aj nohou.



Obrázok 2.2: TrackBall. Zdroj [3]

- Špeciálna klávesnica pre ovládanie jednou rukou (obrázok 2.3) – vystihujúci názov určuje jeho funkciu.



Obrázok 2.3: Klávesnica pre ovládanie jednou rukou. Zdroj [4]

2.3 Sluchovo postihnutí užívateľa

Sluchovo postihnutí užívateľa majú väčšinou bezproblémový prechod navigáciou jednotlivých stránok. Čo však pre nich môže byť komplikované, sú zvukové a video záznamy obsahujúce informačnú hodnotu. Platí teda, že pri informáciách, ktoré sú kritické pre nejakú službu, by mal existovať textový prepis tohto záznamu.

2.4 Užívatelia používajúci alternatívny software

Medzi týchto užívateľov sa radí veľmi rozšírená skupina používateľov mobilných telefónov, ktorí k nemu pristupujú. Taktiež sem patria užívatelia textových prehliadačov v opensource systémoch, ktorí často hľadajú kľúčovú informáciu k rozbehnutiu grafického módu. Rozhodne nie je možné zabezpečiť rovnaké zobrazovanie na všetkých uvedených systémoch, ale podstatné je, aby bola zachovaná funkcionálnosť aj v minimálnom možnom vykresľovacom móde. Preto by malo patriť k práci tvorcu webových aplikácií aj otestovanie minimálnej funkčnosti.

2.5 Užívatelia s poruchami sústredenia

Do poslednej skupiny handicapovaných užívateľov sa radia jedinci s problémami sústredenia spojenou s hyperaktivitou, prípadne problémami učenia – dyslexiou. Ich samotná sústredenosť môže byť rozrušovaná blikajúcimi grafickými prvkami, ktorým je lepšie sa vyhnúť.

Riešením týchto problémov je text správne štrukturovať a písať nenáročne texty. To však platí len v prípade, keď má byť web určený masám a nie vedecky založenej komunite.

3 Prístupnosť a používané metodiky

Ako vyplýva z ukážok typov handicapov v predchádzajúcej kapitole, ku každému typu handicapu môžeme priradiť určitý zákaz pre tvorcov webových stránok. Týmto sa riadili aj inštitúcie pri návrhu jednotlivých metodík a v závislosti od dôležitosti jednotlivým pravidlám nastavujú prioritu. V tejto kapitole sa budeme najskôr venovať samotnej prístupnosti na webe a následne analýzou jednotlivých štandardov.

3.1 Definícia prístupnosti na webe

Celkovo sa počet užívateľov s handicapmi v používaní webových stránok, ktorým webová prístupnosť uľahčuje prístup k informáciám, odhaduje na 35% populácie [5]. Do tejto populácie sa zaraďujú aj užívatelia, ktorí v bežnom živote žiadny problém nemajú, napríklad užívatelia mobilných telefónov prístupujúcich k Internetu.

David Špinar [1] definuje prístupnosť webových stránok takto: „Pojem prístupnosti webových stránok je oproti pojmu zúžený na funkčnosť webových stránok. Prístupné stránky teda nestavajú svojim užívateľom žiadne prekážky, ktoré by im znemožnili daný web efektívne používať.“

Základom je teda osveta medzi tvorcami webových stránok, aby si osvojili pravidlá vedúce k nevyčleneniu tejto tretinovej Internetovej populácie. Vyčlenenie tejto komunity môže mať ekonomický, prípadne spoločenský dopad na prevádzkovateľa.

3.2 Prečo vytvárať prístupný web?

Vyčlenenie užívateľov tretinovej internetovej komunity z používania vlastných služieb je samozrejme najdôležitejším dôvodom pre prevádzkovateľov komerčných weboch za účelom zisku. Investícia do webovej prístupnosti zvýši užívateľskú bázu bez ďalších nákladov natrvalo. Je vhodné teda v prípade nutnosti prerobiť súčasné webové stránky tak, aby spĺňali pravidlá prístupnosti podľa niektorých z metodík uvedených v nasledujúcej kapitole. Ak sa jedná o vývoj nového produktu, nie je nutná investícia, zadávateľ by mal len klásť dôraz na použiteľnosť do požiadaviek vývoja.

Ďalším dôvodom sa javí pozitívna PR propagácia značky. Je neetické odopierať službu určitej skupine užívateľov. Samotné dodržiavanie pravidiel patrí k dobrej vizitke spoločnosti, ktorá môže byť docenená rôznymi oceneniami, či uznaniami.

Komerčné weby sa snažia získať čo najlepšie postavenie v pozíciách vyhľadávania vo webových vyhľadávačoch. Pravidlá prístupnosti čistia kód od zbytočných komplikovaných štruktúr a tým sa zjednodušuje prístup robotov indexujúcich webové stránky. V podstate môžeme povedať, že aj robot je handicapovaný užívateľ a tým zlepšujeme prístupnosť aj pre neho.

Samotný prístupný web umožňuje lepší prístup a prehľadnosť aj pre bežných užívateľov Internetu, a tým očisťuje Internet od šialených výstrelkov tvorcov webových stránok.

3.3 Metodiky prístupnosti

Metodik sa spolu s rozvojom webového designu vytvorila celá rada. Medzi najuznávanejšie patrí WCAG, WCAG 2.0, a v rámci ČR Pravidlá prístupnosti MV ČR. Metodiky zahŕňajú v sebe pravidlá, ktoré určujú postup pri ich vytváraní tak, aby boli webové stránky prístupné handicapovaným užívateľom. Samotné dogmatické dodržiavanie pravidiel býva spochybniteľné, pretože vývoj technológií handicapovaných užívateľov je rapidný a tým sa z niektorých pravidiel stávajú skôr brzdy technologického pokroku. Postupne sa v rôznych krajinách vytvorili a lokalizovali národné pravidlá, ktoré sú v menšej, či väčšej miere vyžadované.

3.4 Medzinárodné normy

Medzi medzinárodné normy patria WCAG verzie 1.0 a 2.0, ktoré mali významný vplyv pri formovaní tejto oblasti.

WCAG je akronym zo spojenia „Web Content Accessibility Guidelines“ (Pravidlá pre prístupnosť webového obsahu) od konzorcia W3C. Prvá verzia vznikla v roku 1999 a stala sa prvou uznanou metodikou vo svete [6]. Na jej základe vznikli rôzne ďalšie metodiky, v súčasnosti sa však už nepoužíva.

WCAG a vo všeobecnosti každá metodika webovej prístupnosti je určená pre rôzne skupiny ľudí:

- tvorcov webových stránok – web designerov, prípadne tvorcov obsahu webových administrátorov
- tvorcov autorizačných a validačných nástrojov
- tvorcov zákonov v jednotlivých krajinách, ktorí na základe metodiky vypracovávajú vlastné zákonné pravidlá

3.4.1 WCAG 1.0

Prvá verzia WCAG znamenala zdefinovanie prvého medzinárodného štandardu prístupnosti webových stránok. Táto metodika sa stala významným míľnikom v roku 1999 pre webovú prístupnosť. Primárnou funkciou WCAG je pochopiť, ako sprístupniť webové stránky užívateľom s popísanými handicapmi.

3.4.1.1 Priority

Každé pravidlo z WCAG má zadanú prioritu, ktorá určuje, aký má pravidlo vplyv na webovú prístupnosť.

- Priorita 1 – táto priorita určuje povinné pravidlo pre tvorcu webových stránok. Ak nie je dané pravidlo splnené, niektorá zo skupín užívateľov s postihnutím nebude schopná používať konkrétnu webovú stránku.
- Priorita 2 – táto priorita určuje nepovinné, ale silne doporučené pravidlo pre tvorcov webových stránok. Ak nie je pravidlo splnené, niektorá zo skupín handicapovaných užívateľov bude musieť prekonávať značné bariéry pre prístup k webovým stránkam.
- Priorita 3 – táto priorita určuje nepovinné pravidlo. Splnenie tohto pravidla zlepšuje webovú prístupnosť vo všeobecnosti.

3.4.1.2 Úroveň vyhovenia pravidlám

WCAG metodika zároveň určuje konkrétnemu webovému dokumentu jeho úroveň vyhovenia pravidlám. Využitie môže byť v štandardizovaných výstupoch z validátorov webovej prístupnosti. Úrovně sú opäť tri:

- úroveň A – sú splnené všetky pravidlá s prioritou 1
- úroveň AA – sú splnené všetky pravidlá s prioritou 1, 2
- úroveň AAA – sú splnené všetky pravidlá s prioritou 1, 2, 3

3.4.1.3 Smernice a pravidlá

WCAG obsahuje smernice (guidelines), v rámci ktorých sú uvedené jednotlivé pravidlá (checkpoints). Členenie smerníc a pravidiel je dvojvrstvové. Každá smernica obsahuje pravidlá, ktoré logicky patria pod danú smernicu a sú jej podmnožinou.

Konkrétne formulované pravidlá budú popísané v rámci formalizácie. Pre pochopenie súvislosti je vhodné uviesť tie smernice a pravidlá, ktoré boli intuitívne odvodené v predchádzajúcej kapitole o užívateľoch so špecifickými postihnutiami.

- Smernica 1 – Poskytni ekvivalentné alternatívy k audio a vizuálnemu obsahu
 - Pravidlo 1.1 – Poskytnutie alternatívneho textu ku všetkým netextovým elementom.
 - Pravidlo 1.2 – Poskytnutie textového odkazu k aktívnemu výberu interaktívnej mapy.

Smernica 1 obsahuje ešte ďalšie tri pravidlá. Užívatelia so zrakovým postihnutím alebo alternatívnym softwarom nie sú schopní obrázkový materiál vidieť rovnako ako užívatelia so sluchovým postihnutím zvukový materiál počuť. Touto smernicou sa riešia najzákladnejšie problémy, ale ako vidieť, spája rôzne typy handicapov. Jednotlivé smernice teda obsahujú všetky skupiny handicapovaných užívateľov s problémami, ktoré sú pre nich podobné.

3.4.2 WCAG 2.0

WCAG 2.0 bola publikovaná na základe vývoja webových stránok a korekcie prvotných pravidiel, ktorých vyžadovanie sa ukázalo v praxi ako nepotrebné. Samozrejme tým nie je dosiahnutá spätná kompatibilita pravidiel, ak však nejaké webové stránky dodržiavali štandard WCAG 1.0, niekoľko menších úprav by malo stačiť na dodržiavanie určitej úrovne WCAG 2.0. Samotné úrovne vyhovenia pravidlám zostali zachované tak, ako boli navrhnuté v prvej verzii. Priority boli odstránené, pravidlá sa hodnotia rovnakou stupnicou akou sú hodnotené úrovne vyhovenia.

3.4.2.1 Štyri princípy prístupnosti

V novej verzii sú predložené štyri princípy, ktoré web musí spĺňať, aby užívatelia s postihnutiami boli schopní web používať. Medzi tieto princípy patria:

- Vnímateľnosť – Informácie na webovej stránke musia byť prezentované v takej podobe, v akej ich môže užívateľ vnímať.
- Ovládateľnosť – Navigácia a užívateľské rozhranie musí byť ovládateľné, nemôže vyžadovať interakciu, ktorú užívateľ nie je schopný splniť.
- Pochopiteľnosť – Informácie a ovládanie užívateľského rozhrania musí byť pochopiteľné.
- Robustnosť – Obsah musí byť dostatočne robustný, aby bol schopný pokrývať rôzne dostupné technológie, cez ktoré sa k nemu pristupuje. Ak teda príde nový vývoj iných zariadení, ktoré budú používané, je potrebné zabezpečiť prístupnosť webových stránok na týchto zariadeniach.

3.4.2.2 Rozdiely v pravidlách

Vplyvom technologického pokroku a kritiky prvej verzii boli upravené niektoré pravidlá a ich priority. Celá štruktúra smerníc a pravidiel bola reorganizovaná. Pre príklad vo WCAG 2.0 v prvom pravidle uvedenom vyššie, pribudla definícia netextového obsahu, medzi ktorú sa radí aj CAPTCHA. Táto zmena reflektuje technologický pokrok, keď sa ochrana proti spamu, CAPTCHA, stala typickým webovým prvkom.

Úprava priority, prípadne odstránenia celého pravidla môže byť ukázaná na príklade pravidla 5.5 prvej verzie. Pravidlo vyžadovalo zhrnutie tabuliek za každou tabuľkou. Toto pravidlo bolo na základe kritiky odstránené.

Spolu s praxou, pokrokom a kritikou bolo pridaných približne dvadsať nových pravidiel. Presné číslo nie je možné určiť, pretože sa niektoré nové pravidlá prekrývajú, prípadne obsahujú čiastočný význam odstránených pravidiel.

3.5 Lokalizované pravidlá prístupnosti

Lokalizáciou pravidiel prístupnosti sa myslí ich čiastočné alebo úplné zahrnutie do zákonov konkrétnej krajiny. Väčšina zákonodarcov bola ovplyvnená metodikou WCAG, ktorú prebrali a upravili do svojej lokálnej podoby.

3.5.1 Česká verzia pravidiel prístupnosti

Pravidlá prístupnosti Českej republiky [1] sú uvedené v zákone č. 365/2000. Vyhláškou 64/2008 Ministerstva Vnútra ČR boli pravidlá aktualizované tak, aby viac odpovedali požiadavkám technologického pokroku [7]. Zároveň boli odstránené niektoré nepotrebné pravidlá, ktorých vymáhanie bolo zbytočné. Cieľom tohto zákona nie je bazírovať na prístupnosti pre všetkých webových tvorcov, ale zabezpečenie prístupnosti k webovým stránkam štátnej správy. Preto by mala štátna správa brať pravidlá ako svoju dogmu, kým tvorcovia bežných webových stránkam ako dobrý doporučený postup. Výhodou týchto pravidiel je ich dostupnosť v rodnom jazyku. Tým je zabezpečená jasnosť výkladu pravidiel, ktorá sa môže pri prekladoch stratiť.

Pravidlá sú rozdelené do šiestich kategórií:

- Obsah webových stránok je dostupný a čitateľný
- Prácu s webovou stránkou riadi užívateľ
- Informácie sú zrozumiteľné a prehľadné
- Ovládanie webu je jasné a pochopiteľné
- Kód je technicky spôsobilý a štruktúrovaný
- Prehlásenie o prístupnosti webových stránok

Ako vidno, hoci táto metodika vyšla dva roky pred normou WCAG 2.0, niektoré kategórie sú analogické s princípmi normy WCAG 2.0, ako napríklad ovládateľnosť a pochopiteľnosť.

3.5.2 Pravidlá Slovenskej republiky

V rámci informatizácie na Slovensku spracováva Ministerstvo financií SR každoročne monitorovanie webovej prístupnosti štátnej správy. Publikáciou výsledkov monitorovania sa Ministerstvo financií SR snaží zaistiť čo najširšiu prístupnosť webov v štátnej správe. Okrem toho hrozia týmto inštitúciám aj sankcie [8]. Pravidlá prístupnosti pre weby verejnej a štátnej správy sú záväzné a vychádzajú z metodiky WCAG 2.0. Keďže predchádzajúcou metodikou bola zvolená WCAG 1.0, na základe odporúčania EÚ pre jej členské štáty sa referenčnou metodikou stala WCAG 2.0 a tým vznikol dokument popisujúci súčasnú potrebnú harmonizáciu metodík [9].

3.5.3 Veľká Británia

Pravidlá pre webovú prístupnosť sú vo Veľkej Británii pomerne čerstvé, keďže boli aktualizované štandardom BS 8878:2010 v roku 2010. Okrem prístupnosti definujú aj použiteľnosť pre postihnutých a starších ľudí. Samotné legálne vyžadovanie nie je priame, avšak zákon „Equality Act 2010“ zakazuje akúkoľvek diskrimináciu postihnutých občanov vo všetkých sektoroch, teda aj súkromnom. Táto diskriminácia je teda z časti popísaná aj hore uvedeným štandardom. [10]

3.5.4 Švédsko

Vo Švédsku existuje inštitúcia – Švédska administratívna vývojová agentúra, ktorá sa zaoberá všeobecnými pravidlami pre webové stránky štátneho sektoru. Len časť z nich pokrýva prístupnosť, medzi ďalšie patria webové štandardy, použiteľnosť, bezpečnosť údajov, vývoj, architektúra apod. Anglický preklad je k dispozícii v referencii. [11]

3.5.5 USA

V USA platí dodatok zákona nazvaný „Section 508“, ktorý patrí pod zákon „Rehabilitation Act of 1973“. Tento zákon vyžaduje dodržiavanie pravidiel federálnymi inštitúciami za účelom sprístupnenia informácie aj užívateľom s postihnutiami. Samotné znenie obsahuje pravidlá vychádzajúce z WCAG 1.0 s prioritou 1, aktualizované o niektoré pravidlá, ktoré sú obsiahnuté vo WCAG 2.0 [12].

3.5.6 Kanada

Pravidlá prístupnosti v Kanade sú upravené štandardom „Common Look and Feel Standards“ z roku 2007, ktoré vyžadujú od webových stránok štátnych inštitúcií dodržiavanie pravidiel štandardu WCAG 1.0 s prioritami 1 a 2. Tento štandard bol zavedený v roku 2000 a aktualizovaný v roku 2007 [13].

3.5.7 Zhrnutie legislatívnej lokalizácie

Z hore uvedených legislatívnych implementácií vychádza, že WCAG naozaj hrá dôležitú úlohu pri tvorbe zákonov jednotlivých krajín a tým sa z neho stáva skutočný medzinárodný štandard. Druhým poznatkom je vymáhateľnosť týchto pravidiel. Dodržiavanie týchto zásad štátnou správou je záväzné takmer vo všetkých krajinách a účinok je prínosný. V súkromnom sektore je však situácia iná a prípadné vynútenie pravidiel môže byť krokom späť, pretože bude brzdou technologického pokroku a v prípade žalôb vyplývajúcich z diskriminácie na základe neprístupnosti webových stránok môže zbytočne skomplikovať podnikanie firmám.

Keďže sa táto práca zaoberá prevažne českou normou webovej prístupnosti, je vhodné porovnať jej metodiku s ostatnými krajinami. Z uvedenej rešerše vyplýva, že legislatívna vymáhateľnosť v porovnaní s ostatnými štátmi je na rovnakej úrovni – prístupnosť sa explicitne vyžaduje len po webových stránkach štátnej a verejnej správy.

Česká republika je špeciálna v tom, že jej metodika nie je doslovne prevzatá zo štandardu WCAG ako je to u väčšiny skúmaných krajín. Hoci jej aktuálnosť je podobná s ostatnými krajinami, keďže pozmeňujúca novela pochádza z roku 2006, o jej aktuálnosti môžeme pochybovať. Rozhodne by bolo vhodné harmonizovať pravidlá s novým vývojom a metodikou WCAG 2.0, tak ako je to napríklad na Slovensku.

4 Formalizácia pravidiel

Táto kapitola sa zaoberá návrhmi na formalizáciu vybraných pravidiel podľa českej normy a medzinárodnej normy WCAG. Nie všetky pravidlá je možné exaktne formalizovať, pretože ich význam môže často závisieť na subjektívnom pocity užívateľa, a preto sú tieto pravidlá v skutočnosti formulované ako odporúčenia.

4.1 Teoretické minimum nutné pre formalizáciu

Na to, aby sme boli schopní formalizovať pravidlá, je nutná teoretická znalosť technológií, z ktorých je webová stránka zložená.

4.1.1 HTML

HTML znamená „HyperText Markup Language“, v preklade hypertextový značkovací jazyk. Vychádza z univerzálneho značkovacieho jazyka SGML. Účelom jazyka je vytvoriť platformu pre jednoduchú tvorbu dokumentov s možnosťou použitia multimediálnych prvkov, odkazov na iné dokumenty a rýchly prechod medzi týmito dokumentmi [14]. Najpoužívanejšou verziou je HTML 4.01, nová verzia 5 je od roku 2008 vo vývoji a je preto ťažké odhadovať, kedy bude dokončená, odladená a zapracovaná do webových prehliadačov [15].

Syntax HTML jazyka je pomerne jednoduchá. Skladá sa z pevne definovaných párových a nepárových značiek a ich atribútov. Každá značka má svoju sémantiku, napríklad párová značka `<title>Názov stránky</title>` označuje nadpis stránky, ktorý sa zobrazí v hornej lište webového prehliadača.

4.1.1.1 Príklad problému pre handicapovaného užívateľa

Nepárová značka `` vykreslí vo webovom prehliadači obrázok. V prípade, že obrázok nie je možné načítať, webový prehliadač zobrazí alternatívny text k obrázku. Na tomto príklade je možné si uvedomiť zraniteľnosť pre handicapovaných užívateľov, ktorí multimediálny obsah nevidia. Ak alternatívny text chýba, názov obrázku neodpovedá jeho obsahu a zároveň je obrázok nosným prvkom dokumentu, stáva sa tento dokument pre užívateľa nepoužiteľný.

4.1.2 CSS

CSS znamená „Cascading Style Sheets“, vo voľnom preklade kaskádové štýly. Je to jazyk, ktorý sa vyvíjal ako doplnok k jazykom vychádzajúcich z SGML. Keďže možnosti definovania formátovania

jazykom HTML sú obmedzené, kaskádové štýly rozširujú tieto možnosti o širokú škálu formátovacích značiek [14].

CSS sa skladá z dvoch častí. Jednou je selektor, ktorý určuje, pre ktoré značky je daný štýl platný a druhou samotná úprava formátu značky. Selektor ponúka viacero možností, ako vybrať danú značku. Používa sa najmä platnosť pre všetky značky, platnosť pre značky s atribútom id (unikátna značka v html dokumente), alebo platnosť pre značky s atribútom class (viacero značiek s rovnakým názvom triedy).

Pri použití selektoru je možné zapísať CSS buď priamo do hlavičky stránky, alebo do externého súboru a pomocou značky *link* ho načítať. Druhou možnosťou je zapísať štýl priamo do atribútu *style* značky, pre ktorú má byť štýl platný.

4.1.2.1 Problematická konštrukcia CSS pre handicapovaného užívateľa

Pomocou CSS je možné zdefinovať pozíciu akéhokoľvek bloku relatívne k súčasnej pozícii, alebo absolútne k ľavému hornému okraju stránky. Táto technika môže byť nesprávne využívaná napríklad na uprednostnenie hlavného textu stránky. Existujú indície, že text nachádzajúci sa na začiatku dokumentu má pre vyhľadávače väčšiu prioritu a teda znamená optimalizáciu pre vyhľadávacie mechanizmy. Pomocou techniky CSS je posunutý smerom dole, napríklad po menu:

```
<div id="hlavny_text" style="position: absolute; top: 100px;">Hlavny text.</div>  
<div id="menu" style="position: absolute; top: 0px;">...</div>  
<div id="vedlajsi_text" style="position: absolute; top: 200px;">Text navazujuci na hlavny  
text</div>
```

V tomto prípade je hlavný text posunutý pod menu, na ktorý v grafickom výstupe nadväzuje iný blok textu. Čítačky pre užívateľov so zrakovým postihnutím, alebo textové prehliadače nie sú schopné interpretácie css tak, aby správne určili o koľko sa text posunul, či nie je posunutý len kvôli prekryvu s obrázkom, apod. Z toho dôvodu čítačka interpretuje text tak, ako je uvedený v zdrojovom kóde, najskôr hlavný text, potom je kontext prepnutý na menu a začne sa čítať menu. A až na poslednom mieste bude vedľajší text, ktorý môže pre užívateľa byť nepochopiteľný, pretože medzičasom dostával informácie o navigácii stránky. Preto sa css konštrukcie na tak radikálne úpravy textu neodporúčajú a mali by sa využívať len k opodstatneným posuvom, ktoré nebudú mať vplyv na tok textu.

4.1.3 JavaScript

HTML značkovací jazyk sám o sebe neponúka žiadnu asynchrónnu interakciu s užívateľom, a preto vznikol priestor pre skriptovacie jazyky, ktoré sú podporované ich interpretom priamo vo webovom prehliadači.

Najpoužívanejší z týchto jazykov je JavaScript, ktorého syntax vznikla inšpiráciou z konštrukcií jazyka Java. Tento jazyk využíva rôzne udalosti značiek, ktoré sú spúšťačom jeho kódu. JavaScript je podobne ako CSS možné načítavať z externého súboru, z hlavičky zdrojového kódu alebo priamo ako inline funkcia v značke. Interpretácia z pohľadu formalizácie pravidiel je veľmi náročná, pretože by vyžadovala vlastný interpret jazyka. Preto sa vo formalizácii obmedzíme na jednoduché prípady, ktoré budú skôr ako jednoduchý prototyp riešenia a nebudú zahŕňať všetky prípady [16].

4.1.3.1 Problematická konštrukcia JavaScriptu pre handicapovaného užívateľa

Presmerovanie stránky je možné buď štandardným riešením pomocou párovej značky `<a>` alebo pomocou JavaScript konštrukcie `window.open`. V kombinácii s udalosťou spustenou na základe pohybu myši môže viesť k nemožnosti k presunu na inú stránku:

Prejdite myšou na nasledovný obrázok a budete presmerovaný na stránku s detailmi!
``

Užívateľ je presmerovaný na detail zájazdu len v prípade, ak prejde myšou nad obrázok. V skutočnosti je na podobný prípad ťažké naraziť, ale pre ilustráciu postačuje. Oveľa bežnejšie je možné nájsť na webe rôzne rolovacie menu, ktoré využívajú JavaScript a neuvažujú o použití iných zariadení. Tým sú tvorcovia webových stránok schopní odrezat' handicapovaných užívateľov, ako aj užívateľov mobilných zariadení, od reálnej navigácie po stránke. Tým sa pre nich stáva webová stránka nepoužiteľná.

4.1.4 XML

XML (extensible markup language) je ďalší značkovací jazyk vychádzajúci z SGML. Na rozdiel od popisovaného HTML, umožňuje voľné použitie značiek bez preddefinovaných názvov a atribútov. Tým sa stáva jazykom vhodným na prenos dát medzi rôznymi platformami alebo na ukladanie perzistentných dát na disk [17].

V tomto konkrétnom prípade bude použitý na konfiguráciu pravidiel, ich nastavení, dôležitosti a značiek, ku ktorým patria.

4.1.5 Gramatiky

Táto podkapitola je prevzatá zo študijných materiálov predmetu Teoretická Informatika [18], ktorý sa vyučuje na Fakulte informačných technológií VUT v Brne. Jazyk je možné reprezentovať prostredníctvom gramatik. Gramatika G je štvorica $G = (N, \Sigma, P, S)$ kde:

- N je konečná množina nonterminálnych symbolov

- Σ je konečná množina terminálnych symbolov
- P je konečná podmnožina kartézkeho súčinu $(N \cup \Sigma)^* N (N \cup \Sigma)^* x (N \cup \Sigma)^*$ nazývaná množina prepisovacích pravidiel
- $S \in N$, je štartovací symbol gramatiky
- Prvok $(\alpha, \beta) \in P$ je prepisovacie pravidlo a zapisuje sa v tvare $\alpha \rightarrow \beta$, kde α je ľavá strana, β pravá strana pravidla $\alpha \rightarrow \beta$

4.1.5.1 Bezkontextové gramatiky

Gramatiky sa líšia podľa tvaru prepisovacích pravidiel. Bezkontextové gramatiky sú podľa Chomského hierarchie typom 2 s pravidlom:

$$A \rightarrow \alpha \quad A \in N, \alpha \in (N \cup \Sigma)^*$$

4.2 Spôsob využitia gramatík pre formalizáciu

Z vyššie uvedených poznatkov je možné odvodiť, že bezkontextový jazyk ako HTML je možné reprezentovať a generovať pomocou pravidiel tak, aby bol validný a splňal jeho definície.

Rovnako je možné upraviť pravidlá tak, aby čiastočne vyhovovali norme, ktorá použitie HTML jazyka upresňuje. Nie všetky pravidlá normy je však možné exaktne interpretovať strojom a preto nie je možné vytvoriť gramatiku, ktorá by kompletne generovala HTML kód pre danú normu. Príkladom môže byť pravidlo číslo 31 z českej normy, *Obsah všetkých tabuliek musí dávať zmysel čítaný po riadkoch zľava doprava*. Sme schopní generovať tabuľku tak, aby jej značky boli validné pre jazyk HTML, avšak nie sme schopní zdefinovať pravidlá, ktoré by generovali tabuľku, ktorej obsah bude dávať zmysel pre človeka. Analogicky nie je možné toto pravidlo kontrolovať, keď obsah, ktorý by mal byť validovaný, nie je ani možné vygenerovať pomocou gramatiky.

Ako už bolo načrtnuté, gramatika generujúca jednotlivé pravidlá normy nám spätne poslúži na kontrolu toho, či časť kódu splňa pravidlo normy, alebo sa v nej nachádza chyba. Ak sme teda schopní generovať validný kód pravidlami gramatiky, budeme schopní s pomocou prioritizácie pravidiel dosiahnuť štartovací symbol gramatiky S . Vďaka tomuto procesu môžeme overiť platnosť alebo neplatnosť daného pravidla.

4.2.1 Jednoduchý príklad využitia gramatiky

Pravidlo č. 28 z českej normy hovorí: „V zdrojovom kóde musí byť určený hlavný jazyk obsahu webovej stránky.“ [7] Toto pravidlo jednoduché na interpretáciu očakáva atribút *lang* alebo *xml:lang* v značke *html*. Značka `<html lang="en">` je teda validná, kým samotná značka `<html>` validná nie je. Pravidlo normy budeme kontrolovať podľa nasledovných pravidiel zjednodušenej gramatiky, kde S je štartovací symbol, veľké písmena sú nonterminály, malé terminály:

$$\begin{aligned}
S &\rightarrow \langle html F \\
F &\rightarrow +F \\
F &\rightarrow lang="A \\
F &\rightarrow xml:lang="A \\
A &\rightarrow +A \\
A &\rightarrow \rangle
\end{aligned}$$

Keďže sa pravidlo týka značky *html*, musíme obmedziť kontrolu len na tieto značky. Okrem toho je vhodné vylepšenie gramatiky inšpiráciou z regulárnych výrazov, keď za + interpret gramatiky nahradí akýkoľvek iný znak. Formálne by mala teda gramatika oveľa viac pravidiel, pretože namiesto + by sme museli vypísať všetky znaky, ktoré chceme nahradiť.

Algoritmus prechádza vstupný kód značky sprava a hľadá pravidlo, ktorým by sa dal nahradiť. Ak ho nájde, text nahradí a opäť sa vráti na koniec vstupu a postupuje smerom doľava. Mohlo by sa teda stať, že niektoré pravidlá preskočí pravidlo s +. V tomto algoritme majú však dlhšie pravidlá výhodu, pretože sú vyhodnocované skôr, teda podľa poradia, v ktorom sú uvedené. Vďaka tomu nám pravidlo $F \rightarrow +F$ nezačne postupne pohlcovať značku *html*, ale dostane sa naozaj do stavu *S*. Ak nám z celého vstupného reťazca zostane len *S*, vieme, že táto gramatika generuje kód, ktorý skúmame a je teda validný.

Správna prioritizácia poradia pravidiel gramatiky je teda veľmi dôležitá pri zostavovaní gramatiky pre pravidlo normy. Gramatika by mohla nesprávne vyhodnotiť vstupný text ako neplatný a tým by nebolo pravidlo vyhodnotené korektne.

Algoritmus môžeme tiež využiť na opačnú situáciu, a to takú, že v kóde hľadáme chybu. Môžeme takto skontrolovať, že atribút *lang* je prázdny a v prípade, že sa dostaneme do stavu *S*, vypísať chybu pravidla, ktorého sa to týka.

4.2.2 Príklad na kompletne využitie bezkontextovej gramatiky

Gramatika z predchádzajúceho príkladu by mohla byť prepísaná na pravú lineárnu gramatiku a teda byť validovaná regulárnymi výrazmi, avšak neboli by sme nimi schopní pokryť prípady, keď potrebujeme kontrolovať párové značky ako v nasledujúcom príklade. Preto je na validáciu pravidiel využívaná bezkontextová gramatika, ktorá vlastne patrí k bezkontextovému jazyku, ako je HTML.

Pravidlo č. 27 z českej normy znie: „Prvky značkovacieho jazyka, ktoré sú párové, musia mať vždy uvedení počiatočnú a koncovú značku. Značky musia byť správne zanorené a nesmie dochádzať

k ich križeniu.“ [7] Keďže má HTML pevne definovaný počet značiek a zároveň bezkontextové gramatiky umožňujú generovať tento jazyk, môžeme jednoducho spätne validovať platnosť tohto pravidla, ak dostaneme na vstup „čistý“ kód skladajúci sa len zo značiek. Skutočný počet pravidiel je oveľa väčší, pre ukážku si vystačíme s dvomi nepárovými a jednou párovou značkou:

$$S \rightarrow \langle div \rangle \langle /div \rangle$$
$$S \rightarrow \langle div \rangle S \langle /div \rangle$$
$$S \rightarrow \langle h1 \rangle \langle /h1 \rangle$$
$$S \rightarrow \langle h1 \rangle S \langle /h1 \rangle$$
$$S \rightarrow \langle img \rangle$$
$$S \rightarrow SS$$

Pomocou tejto gramatiky môžeme generovať jazyk, ktorého značky sa nekrížia a zároveň sú správne zanorené. Spätne teda ak sa zo vstupného kódu algoritmom dostaneme k jednému štartovaciemu symbolu S, dokážeme, že je toto pravidlo normy splnené.

Naopak pre kód $\langle div \rangle \langle h1 \rangle \langle /div \rangle \langle /h1 \rangle$ nebude možný návrat k symbolu S a teda pravidlo normy nie je splnené.

4.3 Rozbor pravidiel v normách

V tejto podkapitole sa budeme venovať pravidlám a ich rozboru, významu a na aké značky sa vzťahujú. Samotné pravidlá gramatík vo formáte XML vytvorené z týchto pravidiel je možné nájsť v prílohe, keďže ich rozsah je veľmi obsiahly. Konkrétne pravidlo je možné nájsť v rámci značky *gramrule* s atribútom *idRule*, ktoré predstavuje číslo pravidla podľa konkrétnej normy. Zároveň sú uvedené len pravidlá normy, ktoré sú možné strojového spracovania, ako bolo odôvodnenie vyššie. Pravidlá sú uvedené vo voľnom preklade do slovenčiny [19].

4.3.1 Pravidlá prístupnosti Českej republiky

Vyššie spomínaná vyhláška 64/2008 obsahuje 33 pravidiel. V tejto podkapitole uvedieme tie, ktoré je možné formalizovať.

4.3.1.1 Pravidlo 1 – Každý ne textový element nesúce významový obsah musí mať svoju textovú alternatívu.

Hneď prvé pravidlo naráža na podstatu zmyslu celého procesu validácie prístupnosti a teda nemožnosť interpretácie multimedialného obsahu pre handicapovaných užívateľov. V tomto prípade môžeme teda zabezpečiť, aby značky $\langle img \rangle$, $\langle input type = "image" \rangle$ a $\langle area \rangle$, ktoré reprezentujú obrázok, obrázkové tlačítko vo formulári a obrázkovú mapu, obsahovali atribút alt, ktorý reprezentuje alternatívny obsah týchto grafických elementov stránky.

Môžeme tiež skontrolovať, či sa v alternatívnom texte nevyskytuje len generický obsah, ktorý obrázok nenahradí. Zameriame sa na tieto texty: image, obrázek, foto.

4.3.1.2 Pravidlo 3 – Ak to charakter webových stránok nevyklučuje, informácie obsiahnuté prostredníctvom skriptov, objektov, apple tov, kaskádových štýlov, cookies a iných doplnkov na strane užívateľa, musia byť dostupné aj bez ktoréhokoľvek z týchto doplnkov a stránky musia byť štandardne ovládateľné.

Z tohto pravidla je možné vybrať zákaz ovládať stránky prostredníctvom skriptov. Ako bolo spomenuté, bez interpretu JavaScriptu bude táto kontrola len prototyp, ktorý nie vždy musí fungovať. Nájdem teda odkazy, ktoré sú závislé na JavaScripte a vypíšeme varovanie pre užívateľa.

4.3.1.3 Pravidlo 4 – Informácie prezentované vizuálnou podobou webových stránok, tvary jednotlivých prvkov, ich veľkosťou, poradím alebo umiestnením musia byť dostupné aj v prípade, že užívateľ nemôže tieto aspekty vnímať.

Tomuto pravidlu sme sa venovali v podkapitole č. 4.1.2.1. Nie sme schopní detekovať, či sa úpravou pozície pomocou CSS zmenil aj informačný tok v stránke, avšak môžeme vypísať varovanie, ktoré zdôvodní tento problém.

4.3.1.4 Pravidlo 7 – Veľkosť písma musí byť možné zväčšiť aspoň na 200 % a zmenšiť aspoň na 50 % pôvodnej hodnoty pomocou štandardných funkcií prehliadača. Pri takej zmene veľkosti nesmie dochádzať k strate obsahu alebo funkcionality.

Veľkosť písma môže byť definovaná prostredníctvom CSS v rôznych jednotkách. Jednotky reprezentujúce veľkosť ako pt, pc, in, cm, mm a px sú v prehliadačoch zobrazené napevno a pri zväčšovaní a zmenšovaní stránky sa ich zmena neprejaví. Toto sa však týka len starších prehliadačov, nové prehliadače sú schopné zväčšovať celé stránky ako jeden celok. Preto toto pravidlo od návrhu normy stratilo na dôležitosť. Každopádne však budeme normu dodržiavať, a preto vypíšeme varovanie pre užívateľa.

4.3.1.5 Pravidlo 10 – Načítanie novej webovej stránky či presmerovania musí byť možné len po aktivácii odkazu alebo po odoslaní formuláru.

Toto pravidlo čiastočne dopĺňa pravidlo č. 3. Opäť vytvoríme gramatiku, ktorá bude hľadať presmerovania na základe pohybu myši a v prípade, že takýto jav objaví, vypíše varovanie.

4.3.1.6 Pravidlo 11 – Načítanie novej webovej stránky do nového okna prehliadača musí byť možné len v odôvodnených prípadoch a užívateľ na to musí byť dopredu upozornený.

HTML jazyk umožňuje prostredníctvom značky *a* odkazovať na iné dokumenty. S pridaním atribútu *target* s hodnotou *_blank* alebo *_new* umožňuje otvoriť odkaz v novom okne, respektíve tabe

prehliadača. Keďže nie je možné strojovo rozoznať, či je odkaz otvorený v novom okne graficky odlišený, môžeme na to užívateľa upozorniť, aby skontroloval túto skutočnosť.

4.3.1.7 Pravidlo 12 – Na webovej stránke nesmie dochádzať rýchlejšie ako trikrát za sekundu k výrazným zmenám farebnosti, jas u, veľkosti alebo umiestneniu prvkov.

Toto pravidlo zamerané na multimedialne prvky nie je možné gramatikou kompletne kontrolovať, môžeme tak aspoň skontrolovať, či sa v texte nenachádza značka *blink*, ktorú toto pravidlo v podstate zakazuje.

4.3.1.8 Pravidlo 17 – Bloky obsahu, ktoré sa opakujú na viacerých webových stránkach daného orgánu verejnej správy, je možné preskočiť. Ak webové stránky nemajú veľký rozsah, nemusí byť zaistené preskočenie opakujúcich sa blokov obsahu.

Bloky obsahu je možné skrývať, avšak neodporúča sa CSS konštrukcia *visibility: hidden* a *display:none*, pretože niektoré zariadenia môžu mať problém s jej interpretáciou.

4.3.1.9 Pravidlo 20 – Každá webová stránka musí mať výsťažný názov odpovedajúci jej obsahu.

Toto pravidlo je jednoduché na zdefinovanie. V rámci tohto pravidla budeme kontrolovať existenciu značky *title* a toho, či sa v ňom nenachádza generická hodnota, ako napríklad *Neznáma stránka* alebo *Unknown title*.

4.3.1.10 Pravidlo 24 – Text odkazu alebo jeho priamo súvisiaci text musí výsťažne popisovať cieľ odkazu. Ak odkaz vedie na iný typ súboru, ako je webová stránka, musí byť odkaz doplnený informáciou o type, prípadne o veľkosti tohto súboru.

V rámci tohto pravidla opäť nemôžeme dôsledne kontrolovať jeho dodržiavanie. Môžeme však skontrolovať, či je v tele odkazov nevyskytujú klasické generické názvy ako „zde“, „tento odkaz“ apod.

Taktiež môžeme čiastočne pokryť druhú polovicu pravidla. Nájsť odkazy, ktoré smerujú evidentne na súbor s typickou príponou, ako napríklad pdf, doc apod. Ak odkaz smeruje na súbor, vypíšeme pripomienku pre užívateľa, aby skontroloval tieto informácie.

4.3.1.11 Pravidlo 25 - Každý rám musí mať vhodné meno či popis vyjadrujúci jeho zmysel a funkčnosť.

Značka *frame* musí na základe tohto pravidla obsahovať atribút *title*, ktorý popíše účel, ktorý v rámci stránky spĺňa.

4.3.1.12 Pravidlo 27 – Prvky značkovacieho jazyka, ktoré sú párové, musia mať vždy uvedenú počiatočnú a koncovú značku. Značky musia byť správne zanorené a nesmie dochádzať k ich kríženiu.

Tomuto pravidlu sme sa venovali v ukázkovom príklade v kapitole č. 4.2.2. Kontrolujeme správne zanorenie a vypíšeme chybu, ak k nemu v syntaxi jazyka dochádza.

4.3.1.13 Pravidlo 28 – V zdrojovom kóde musí byť určený hlavný jazyk obsahu webovej stránky.

Tomuto pravidlu sme sa taktiež podrobne venovali v ukázkovom príklade v kapitole č. 4.2.1. Budeme kontrolovať existenciu a obsah atribútu *lang* v značke *html*.

4.3.2 WCAG 2.0

Keďže niektoré pravidlá z českej normy vychádzajú z normy WCAG, prípadne sa vzájomne prekrývajú, budú uvedené odkazy na predchádzajúcu podkapitolu, ktoré riešia rovnaký problém. Opäť je uvedený voľný preklad do slovenčiny z neoficiálneho českého prekladu [20].

4.3.2.1 Pravidlo 1.1.1 - Ne textový obsah: Každý ne textový obsah, ktorý je užívateľovi prezentovaný, má svoju textovú alternatívu, slúžiacu rovnakému účelu.

Toto pravidlo je zhodné s pravidlom č. 1 českej normy. Dôležitým rozšírením je však spomínanie prístupnosti CAPTCHA, teda bežné overenie interakcie s človekom vo formulároch. CAPTCHA by mala obsahovať rôzne varianty, aby ju bolo možné zadať použitím iných zmyslov ako zraku.

4.3.2.2 Pravidlo 1.2.3 - Audiopopis či alternatíva pre predtčený multimedialny prvok musí existovať.

Takýmto prvok je značka *object*, ktorá by ako svoje telo medzi párovými značkami mala obsahovať alternatívny popis multimedialneho prvku.

4.3.2.3 Pravidlo 1.3.1 – Informácie a vzájomné vzťahy: Informácie, štruktúra a vzájomné vzťahy obsiahnuté v prezentácii je možné programovo určiť alebo sú dostupné vo forme textu.

Toto pravidlo v sebe zahŕňa napríklad neodporúčanie používania niektorých starších značiek tak, aby boli nahradené novšími konštrukciami. Je vhodné pri datových tabuľkách využiť atribút *summary*, ktorý bude popisovať obsah tabuľky.

4.3.2.4 Pravidlo 1.4.4 – Zmena veľkosti textu

Pravidlo je totožné s pravidlom č. 7 českej normy.

4.3.2.5 Pravidlo 2.4.1 – Preskakovanie blokov

Pravidlo je totožné s pravidlom č. 17 českej normy.

4.3.2.6 Pravidlo 2.4.2 – Stránka obsahuje titulok.

Pravidlo je totožné s pravidlom č. 20 českej normy.

4.3.2.7 Pravidlo 2.4.3 – Poradie prechádzanie prvkov.

Pravidlo je totožné s pravidlom č. 4 českej normy.

4.3.2.8 Pravidlo 2.4.9 – Odkaz musí mať je dnoznačný účel.

Pravidlo je totožné s pravidlom č. 24 českej normy.

4.3.2.9 Pravidlo 3.1.1 – Jazyk stránky musí byť nastavený.

Pravidlo je totožné s pravidlom č. 28 českej normy.

4.3.2.10 Pravidlo 3.1.1 – Zmeny kontextu sa prevádzajú v dôsledku akcie užívateľa.

Toto pravidlo je komplikovanejšie si vysvetliť. Hovorí však o potrebe informovania užívateľa o zmenách kontextu, napríklad pri odkaze na nové okno je nutné, aby bol odkaz odlišený. Pravidlo sa teda prekrýva s pravidlom č. 11 českej normy.

4.3.2.11 Pravidlo 4.1.2 – U užívateľských rozhraní je možné strojovo určiť názov.

Toto pravidlo informuje o používaní atribútu *title* u značiek *frame* a *iframe*. Pravidlo sa prekrýva s pravidlom č. 25 českej normy, v odporúčaniach sa však pridáva aj novšia značka *iframe*.

4.3.3 Súhrn rozboru pravidiel

Ako vidno na konkrétnych pravidlách, česká norma vychádza z WCAG verzie 1.0, a tým pádom sú niektoré pravidlá neaktualizované, na rozdiel od normy WCAG 2.0. Tá pridáva niekoľko nových pravidiel. Nič však nemení na skutočnosti, že dané pravidlá už nemusia odpovedať skutočnosti, keď si uvedomíme, že česká norma je z roku 2006 a WCAG 2.0 z roku 2008. Pri takom enormnom rozmachu Internetu, by ju rozhodne nebolo na škodu jemne aktualizovať a vydať nové odporúčania.

Druhou témou je subjektívny a rozsiahly výklad pravidiel. Niektoré pravidlá z normy WCAG sú veľmi obsiahle a človek si bez oficiálnych odporúčaní [21] k nim veľa informácii neodnesie. Preto by som tiež navrhol sprehľadniť a zoštíhliť titulky pri jednotlivých názvoch u WCAG. Sám som ich pre účely tejto diplomovej práce v minulej kapitole skracoval, ich zoznam je však možné nájsť v prílohe. Správca webových stránok, ktorý má snahu vylepšiť svoje stránky, môže byť odradený po otvorení kvanta textu týkajúceho WCAG 2.0. Veľké množstvo informácií sa týka multimediálnych stránok, ktoré nie sú pre všetkých užitočné. Bolo by prípadne vhodné vytvoriť súhrn aplikovateľných pravidiel podľa typu webového obsahu.

5 Návrh aplikácie

Aplikáciou navrhnutou na implementáciu v rámci diplomovej práce je validátor prístupnosti webových stránok. Aby bol validátor jednoduchý na použitie a dostupný širokému obecnstvu, bude aplikácia bežať online na webovom serveri.

5.1 Analýza existujúcich validátorov

V tejto podkapitole budú zanalyzované najčastejšie používané validátory a pomocné nástroje určené k korektnému vývoju webových stránok.

5.1.1 W3C validátor

Hoci W3C validátor¹ nepatrí do oblasti webovej prístupnosti, je vhodné spomenúť ho na prvom mieste. Tento validátor sa zaoberá sa validáciou HTML a XHTML webovej stránky, ktorá by mala byť považovaná za prvotný krok pri validácii stránky. Po overení čistoty zdrojového kódu môže tvorca webových stránok pristúpiť k validácii ostatných vlastností stránky, medzi ktorými je aj skúmaná validácia prístupnosti.

Validátor poskytuje nasledovné vlastnosti, ktoré môžu byť použité v navrhnutom validátore prístupnosti webových stránok:

- možnosť validovať stránku priamo z URL adresy
- možnosť validovať ručne vložený HTML kód
- možnosť validovať vložený súbor s HTML kódom
- zapnutie dlhšieho popisu výstupných hodnôt
- nastavenie prípadne chýbajúceho kódovania
- nastavenie prípadne chýbajúcej verzie HTML

Samotné výsledky validátora sú v textovej podobe, rozdelené na chyby, upozornenia a poznámky zároveň s odporúčaniami na zmenu tak, aby pri následnej kontrole všetko prebehlo v poriadku. Užívateľské prostredie validátora, zobrazené na obrázku 5.1, je veľmi príjemné s možnosťou rýchlej zmeny nastavení zobrazených výsledkov a opätovného validovania stránky.

¹ <http://validator.w3.org/>



Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

Jump To: [Validation Output](#)

Error found while checking this document as HTML 4.01 Strict!

Result:	1 Error	
Address :	<input type="text" value="http://www.fit.vutbr.cz/"/>	
Encoding :	iso-8859-2	<input type="text" value="(detect automatically)"/>
Doctype :	HTML 4.01 Strict	<input type="text" value="(detect automatically)"/>
Root Element:	HTML	



The W3C validators rely on community support for hosting and development.

3186

[Donate](#) and help us build better tools for a better web.



Options

Show Source

Show Outline

List Messages Sequentially Group Error Messages by Type



Validate error pages



Verbose Output

Clean up Markup with HTML-Tidy

[Help](#) on the options is available.

Revalidate

Obrázok 5.1: HTML validácia webovej stránky www.fit.vutbr.cz

5.1.2 Achecker

Novší validátor Achecker² webovej prístupnosti, ktorý umožňuje validáciu HTML aj CSS podľa viacerých dostupných štandardov, medzi inými aj novší WCAG 2.0. Oproti iným validátorom je jeho prínos v týchto oblastiach:

- príjemné užívateľské prostredie
- dostupnosť štandardov ako Section 508, WCAG 1.0, WCAG 2.0
- možnosť nastavenia, aké priority má výstupná správa obsahovať
- možnosť separátnej validácie CSS a HTML

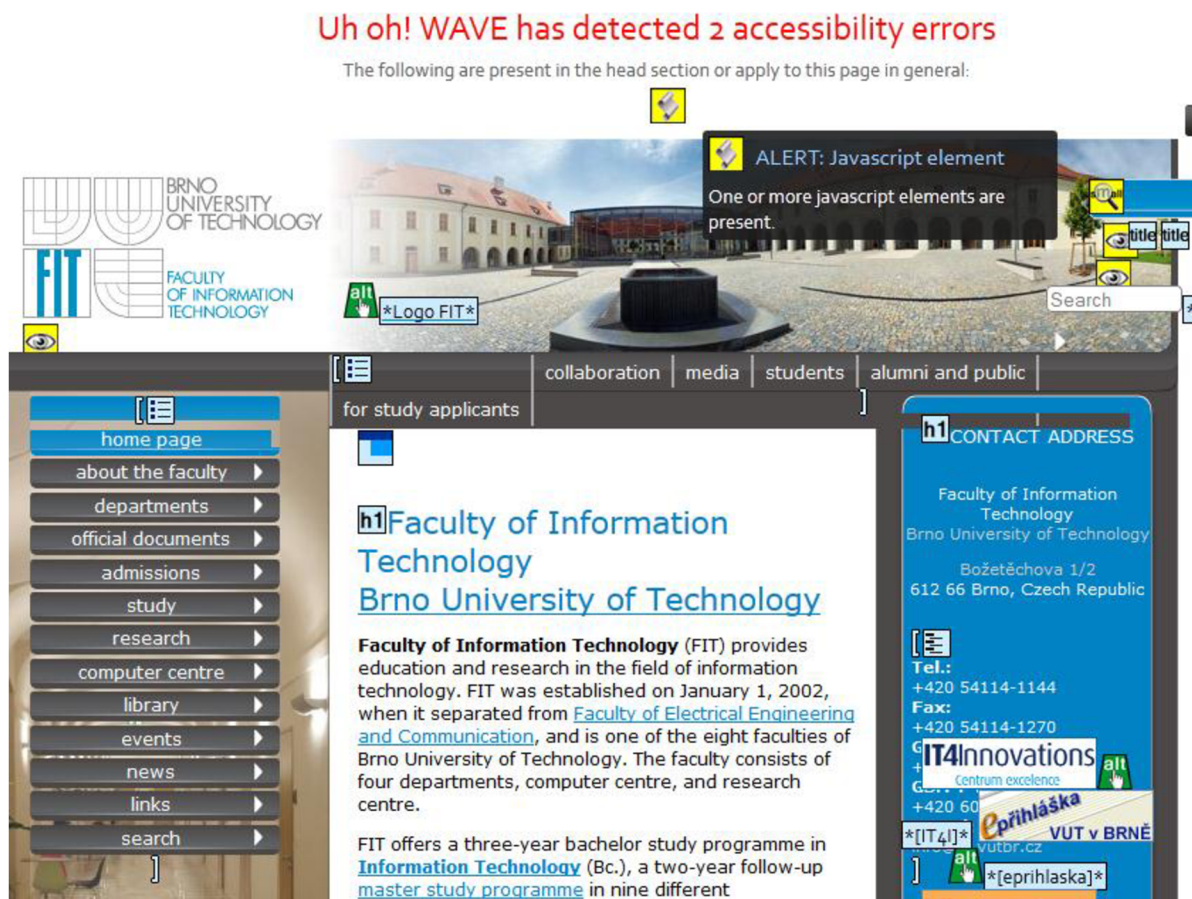
² <http://achecker.ca>

5.1.3 WAVE validátor

WAVE validátor³ sa radí medzi popredne používané validátory prístupnosti. Oproti základnému W3C validátoru neposkytuje toľko možností na vstupe, avšak výstup validátora je grafický s prekryvom, ktorý obsahuje podrobný popis jednotlivých prvkov na stránke a odporúčania na ich prípadné vylepšenia.

Nasleduje popis vlastností programu, ktoré vynikajú medzi inými validátormi:

- Grafický výstup (obrázok 5.2) s popisom jednotlivých prvkov.
- Možnosť vypnúť zobrazovanie kaskádových štýlov (CSS).
- Možnosť využiť ako plugin do webového prehliadača – okamžité hodnotenie prehliadaných stránok. Táto vlastnosť môže byť využitá medzi webovými grafikmi.



Obrázok 5.2: Grafický výstup validácie webovej stránky www.fit.vutbr.cz

³ <http://wave.webaim.org>

5.1.3.1 Vypnutie zobrazenia kaskádových štýlov

Bez zobrazených kaskádových štýlov (obrázok 5.3) je možné veľmi dobre si predstaviť, ako vnímajú webové stránky ľudia so zrakovým postihnutím. Webová stránka je pre nich iba text plynúci zhora nadol.

[IMAGE: "Logo FIT"]

[TEXT BOX: "Search:"] [IMAGE BUTTON: "Search"]

- [For Study Applicants](#)
- [Collaboration](#)
- [Media](#)
- [Students](#)
- [Alumni and Public](#)

- [Home page](#)
- [About the Faculty](#)
- [Departments](#)
- [Official Documents](#)
- [Admissions](#)
- [Study](#)
- [Research](#)
- [Computer Centre](#)
- [Library](#)
- [Events](#)
- [News](#)
- [Links](#)
- [Search](#)

Contact Address

[Faculty of Information Technology](#)
[Brno University of Technology](#)

Božetěchova 1/2
612 66 Brno, Czech Republic

Obrázok 5.3: Webová stránka www.fit.vutbr.cz bez kaskádových štýlov

5.1.4 Cynthia Says

Cynthia Says⁴ od spoločnosti HiSoftware patrí rovnako ako Wave medzi popredné validátory webovej prístupnosti. Na rozdiel od prechádzajúceho validátora umožňuje vybratie štandardu, podľa ktorého chceme webové stránky hodnotiť. Bohužiaľ, medzi podporovanými metodikami chýba nový štandard WCAG 2.0. Pozitívne ale je, že je možné vybrať prioritu pravidiel WCAG, ktoré má program považovať za vyžadované. Na druhú stranu však neobsahuje grafický výstup, ale len pomerne neprehľadnú tabuľku s výsledkami a odporúčaniami, ako je možné vidieť na obrázku 5.4.

Vlastnosti, ktorými sa Cynthia odlišuje od ostatných validátorov:

- výber z viacerých štandardov
- možnosť hodnotenia podľa rôznych typov priorít
- výsledky na základe emulácie konkrétneho webového browseru
- test na pohybujúce sa elementy obrazovky
- upozornenia k pravidlám na ručnú kontrolu

Napriek tomu, že je tento validátor pomerne neudržiavaný, niektoré nápady môžu byť využité v nových aplikáciách nadväzujúcich na vývoj. Negatívom je, že tento validátor bez podpory novej verzie WCAG je stále na viacerých weboch považovaný za jeden z najlepších nástrojov.

Verification Checklist			
Checkpoints	Passed		
	Yes	No	Other
Basic Settings			
<p>1.1 / (a) Provide a text equivalent for every non-text element (e.g., via "alt", "longdesc", or in element content). <i>This includes:</i> images, graphical representations of text (including symbols), image map regions, animations (e.g., animated GIFs), applets and programmatic objects, <u>ascii</u> art, frames, scripts, images used as list bullets, spacers, graphical buttons, sounds (played with or without user interaction), stand-alone audio files, audio tracks of video, and video.</p> <ul style="list-style-type: none"> ○ Rule: 1.1.1 - All IMG elements are required to contain either the alt or the longdesc attribute. <ul style="list-style-type: none"> ○ Warning - IMG Element found at Line: 151, Column: 9 contains the 'alt' attribute with an empty value. Please verify that this image is only used for spacing or design and has no meaning. ○ Rule: 1.1.2 - All INPUT elements are required to contain the alt attribute or use a LABEL. <ul style="list-style-type: none"> ○ No invalid INPUT elements found in document ○ Rule: 1.1.3 - All OBJECT elements are required to contain element content. <ul style="list-style-type: none"> ○ No OBJECT elements found in document body. ○ Rule: 1.1.4 - All APPLET elements are required to contain both element content and the alt attribute. <ul style="list-style-type: none"> ○ No APPLET elements found in document body. ○ Rule: 1.1.6 - All IFRAME elements are required to contain element content. <ul style="list-style-type: none"> ○ No IFRAME elements found in document body. ○ Rule: 1.1.7 - All Anchor elements found within MAP elements are required to contain the alt attribute. 	Yes		

Obrázok 5.4: Jednoduchý výstup z validátoru Cynthia Says

⁴ <http://www.contentquality.com>

5.1.5 Total Validator

Na rozdiel od prechádzajúcich online validátorov, sú na Internete k dispozícii aj desktopové aplikácie. Konkrétne Total Validator⁵ je určený ako desktopová aplikácia pre Windows, Linux, OS X. Je však dostupný aj ako rozšírenie pre Firefox.

Jeho veľkou výhodou je implementácia rôznych štandardov v jednom. Umožňuje validovať základnú konštrukciu stránky, HTML zdrojový kód ako W3C validátor, WCAG 1.0, WCAG 2.0 a Section 508 ako validátor prístupnosti, CSS validáciu, vyhľadávanie mŕtvych odkazov vedúcich zo stránky, kontrolovať gramatiku angličtiny, nemčiny, francúzštiny, taliančiny a španielčiny. Jeho vývoj je veľmi živý a každý rok obsahuje viacero dlhých opráv chýb a pridaných vylepšení. V súčasnej dobe je vo verzii 7.4.0. Jeho vývoj je pravdepodobne umožnený platenou profesionálnejšou verziou, ktorý pridáva nové funkcie, ako napríklad spracovanie celého stromu webovej stránky v rámci jednej validácie. Tento nástroj spolu s integráciou do webového prehliadača sa javí ako jeden z najprofesionálnejších a najprepracovanejších nástrojov pre tvorcov webových stránok.

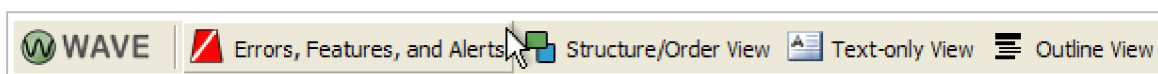
5.2 Analýza doplnkových nástrojov

Okrem validátorov existujú rôzne nástroje, ktoré pomáhajú tvorcom webových stránok pri dodržiavaní základných princípov tvorby prístupného webu.

5.2.1 Toolbary pre webové prehliadače

Podobne ako ponúka Total Validátor vlastný integrovaný toolbar pre Firefox so všetkými možnosťami ako jeho desktopová aplikácia, existujú viaceré toolbary pre rôzne prehliadače.

Príkladom je WAVE toolbar (obrázok 5.5) opäť pre Firefox, ktorý je schopný validovať stránky, prípadne ich zobrazit' bez CSS štýlov čisto v textovej podobe.



Obrázok 5.5: Ukážka WAVE toolbaru

Podobným pomocníkom je aj Web Accessibility Toolbar⁶, ktorý je určený pre Internet Explorer. Umožňuje podobné možnosti ako predchádzajúce nástroje, umožňuje teda možnosť využitia pre tvorcov webových stránok, ktorých obľúbeným prehliadačom je Internet Explorer.

⁵ <http://www.totalvalidator.com>

⁶ <http://www.visionaustralia.org.au/ais/toolbar/>

5.2.2 Kontrola kontrastu farieb

Keďže rozdiel medzi kontrastom farby textu a farby pozadia je témou pravidiel všetkých štandardov prístupnosti, je táto pomôcka⁷ veľmi užitočná pri vytváraní, alebo revidovaní stránky. Po zadaní hodnôt farby textu a pozadia v hexadecimálnom tvare, zobrazí pomôcka ukážku (obrázok 5.6) pre menší a väčší text, a zároveň výsledky splnenia pre WCAG kategórie AA a AAA. Táto pomôcka by mala byť bežný nástroj pre každého správcu webových stránok.

Color Contrast Checker

Foreground color: #3e3eff  [lighten](#) | [darken](#)

Background color: #e1e1e1  [lighten](#) | [darken](#)

Contrast Ratio: **4.8:1**

Normal Text

WCAG AA: **Pass**

WCAG AAA: **Fail**

Sample: I am normal text

Large Text

WCAG AA: **Pass**

WCAG AAA: **Pass**

Sample: I am big text

Obrázok 5.6: Pomôcka na validáciu kontrastu

5.3 Analýza požiadaviek

Na základe analýzy príbuzných validátorov je možné vybrať podstatnú funkcionality a navrhnuť, ktoré požiadavky budú dostupné v navrhovanom programe. Okrem nich sú pridané niektoré vylepšenia:

- možnosť validovať stránku priamo z URL adresy
- možnosť validovať ručne vložený HTML kód
- možnosť validovať vložený súbor s HTML kódom
- možnosť pridávať metodiky hodnotenia ako formáty XML
- a následná možnosť výberu z viacerých štandardov
- možnosť hodnotenia podľa rôznych typov priorít

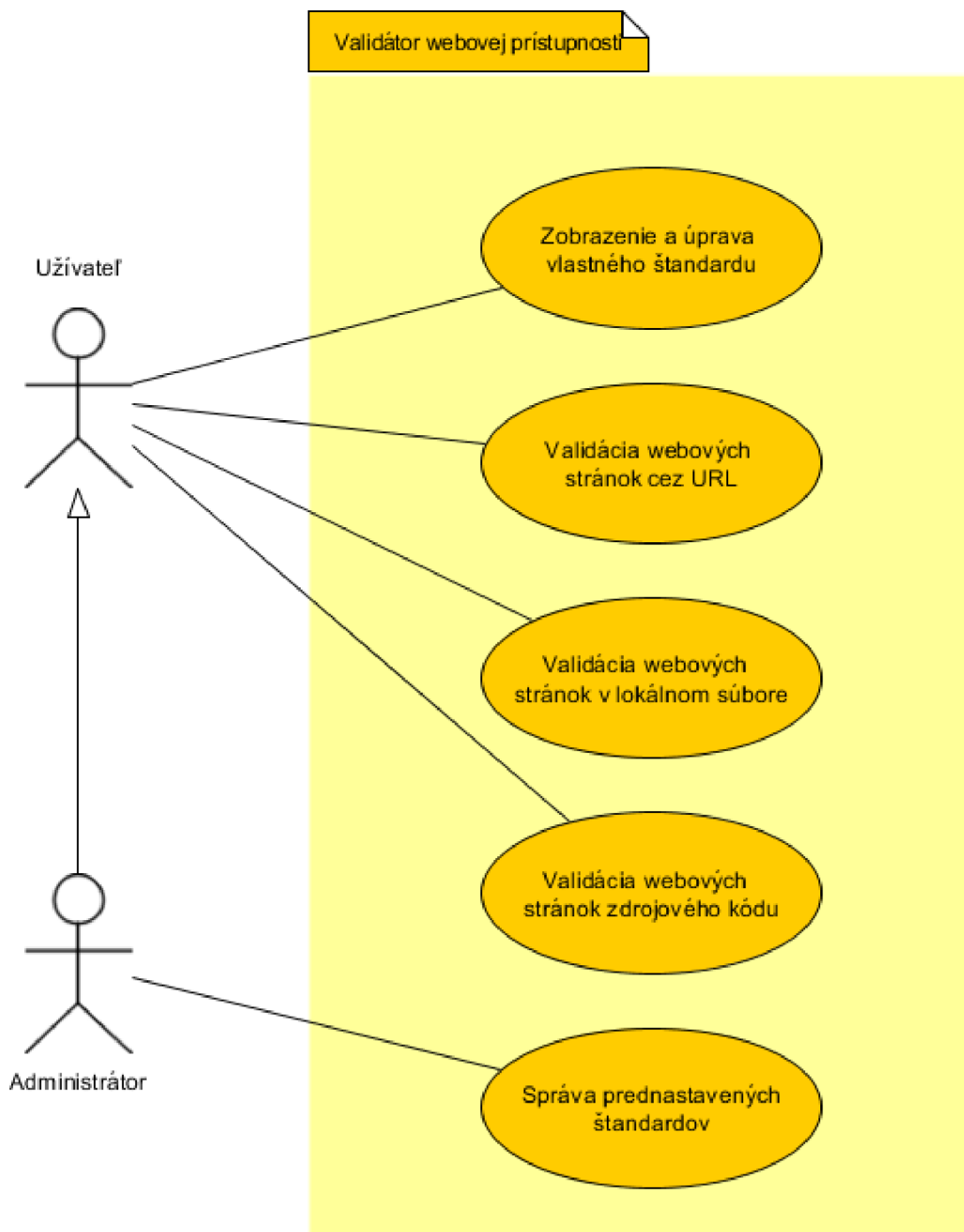
⁷ <http://webaim.org/resources/contrastchecker/>

Validátor by mal byť dostupný čo najširšiemu množstvu užívateľov, bez nutnosti inštalácie ako desktopovej inštalácie, ktorá užívateľov zbytočne zdržuje. Zároveň môžu mať výhrady voči spúšťaniu necertifikovaných binárnych súborov. V súčasnej situácii sa javia tri možnosti, jednou je online webové rozhranie, druhou aplikácia pre tablet, ktorej inštalácia je veľmi rýchla a užívateľa nijako neobťažuje. Aplikácia pre tablet by bola pravdepodobne svetovým priekopníkom, nevýhodou však zostáva, že vývoj webových aplikácií prebieha na zariadenia s klávesnicou a tým pádom nie je pre užívateľa vhodné kontrolovať prístupnosť na tablete. Tretou možnosťou je integrovaný toolbar pre niektorý z webových prehliadačov.

Keďže má autor väčšie znalosti týkajúce sa vývoja webových aplikácií, vhodnou formou aplikácie bude, keď bude dostupná online na verejnej doméne. Tým bude prínos tejto diplomovej práce zachovaný a v prípade pozitívneho ohlasu na aplikáciu môže byť táto ľahko šírená.

5.3.1 Diagram prípadov použitia

Diagram prípadov použitia, zobrazený na obrázku 5.7, vychádza z predchádzajúcej analýzy požiadaviek.



Obrázok 5.7: Diagram prípadov použitia aplikácie

5.4 Návrh validátora

Programovacím jazykom validátora bol zvolený Python pre dobrú a jednoduchú použiteľnosť parsera potrebného k spracovaniu HTML, alebo XML a iných potrebných nástrojov vhodných pre tento projekt. Na prepojenie pythonu s webovým serverom je k dispozícii veľká škála opensource softvéru, ako napríklad Django, TurboGears, CherryPy alebo Web2py.

5.4.1 Voľba frameworku - Django

Pre implementáciu tohto projektu bol vybraný framework pre prepojenie webového serveru s pythonom Django, z dôvodu silnej podpory, množstva dokumentácie a v neposlednej rade dostupného webhostingu.

Django vyžaduje dodržiavanie softwarovej architektúry MVC (model, viewer, controller), ktorý sprehľadňuje aplikáciu a jej kód. Model umožňuje návrh databázy a následne zjednodušené operácie nad ňou prostredníctvom vstavaných metód frameworku. V našom prípade však model potrebovať nebudeme. Druhou zložkou princípu zobrazovania je viewer (pohľad), ktorého úloha je spracovávať hrubé dáta z modelu, prípadne z iných zdrojov ako XML dokumentov v našom prípade. Tieto spracované a upravené dáta sú následne posielané do controlleru (radič), ktorý ich dosadí do šablóny a zobrazí užívateľovi. Týmto spôsobom je zabezpečený správne rozložený kód, na ktorom môže spolupracovať viac ľudí bez komplikácií. Napríklad grafik, ktorý nepotrebuje ovládať znalosť programovacieho jazyka, môže vytvárať šablóny a vsádzať do nich spracované dáta, za ktorých správnosť ručí programátor [22].

K dispozícii sú všetky knižnice Pythonu a zároveň rozšírenia, ktoré ponúka Django. V skutočnosti však väčšinu z nich nebudeme potrebovať, pretože aplikácia nebude využívať databázu. Bude jej stačiť načítať webovú stránku, pravidlá z dokumentu XML a pre potreby tohto projektu nie je nutné výsledky ukladať do databázy.

5.4.2 Užívateľské rozhranie

Užívateľské rozhranie je v rámci frameworku Django špecifikované využitím radiča a šablón presne tak, ako je popísané v predchádzajúcej podkapitole. Konkrétne užívateľské rozhranie pre túto prácu bude vytvorené z jednoduchej webovej šablóny dostupnej na internete. Užívateľ bude mať možnosť rozhodnúť sa, ktorú z možností vstupu (zdrojový kód, URL adresa alebo lokálny súbor) bude preferovať. Zároveň si užívateľ bude môcť vybrať, ktorý štandard bude chcieť aplikovať na svoj zdrojový kód.

Po odoslaní formulára sa vyhodnotia pravidlá na základe vstupného zdrojového kódu a zobrazí sa prehľadná tabuľka so sumárom chýb v záhlaví. Každý riadok tabuľky bude obsahovať identifikátor pravidiel, na základe ktorého bol prehrešok spáchaný. K identifikátoru bude prislúchať detailná správa o chybe a odporúčanie ako ju napraviť. Okrem týchto informácií budú užívateľovi zobrazené detaily ako číslo riadku, na ktorom sa chyba v zdrojovom kóde vyskytuje, konkrétny riadok a priorita, ktorej vyhodnotenie náleží.

5.4.3 Načítanie pravidiel gramatík

Aby bola aplikácia univerzálna, bude ako vstup požadovať XML súbor s gramatikou pre jednotlivé pravidlá. Rovnako bude možné nadefinovať prioritu konkrétneho pravidla.

Validátor bude teda schopný využiť akýkoľvek zadaný XML súbor. Tým sa stane nástrojom nielen pre validáciu webovej prístupnosti, ale v podstate pre validáciu akýchkoľvek štandardov, ktorých pravidlá je možné presne nadefinovať gramatikou.

5.4.4 Načítanie webovej stránky

Užívateľ bude mať na úvodnej stránke tri možnosti ako určiť stránku k validácii a to: vložením súboru, URL adresy, alebo priamo zdrojového kódu. Tu však cesta pre spracovanie HTML kódu ešte len začína. Keďže webové prehliadače sú schopné načítavať aj nie kompletne validný zdrojový kód stránok, správcovia webových stránok nie sú nútení zabezpečiť svojim stránkam kompletnú HTML validáciu. Tým ale môže byť narušené vyhodnotenie pravidiel podľa zadaných gramatík, čo by viedlo k nekorektnej výslednej správe.

Aby sme predišli týmto problémom a zároveň zjednodušili zapisovanie gramatík, zavedieme formát HTML kódu, do ktorého budeme vstupný kód spracovávať. Budeme využívať nasledovné postupy:

- Atribúty u jednotlivých značiek môžu mať formát *atr=hodnota*, *atr='hodnota'* alebo *atr="hodnota"*. HTMLparser z knižnice pythonu nám tieto hodnoty načíta do jednej štruktúry, a preto ju môžeme na výstupe uložiť do jedného formátu *atr="hodnota"*. Tým ušetríme pravidlá, ktoré by museli počítat s rôznymi formátmi.
- Značky znamenajúce komentáre budeme zahadzovať.
- Celý výstup zmeníme na malé písmená, aby sme odlišili nonterminály od terminálov pri spracovávaní gramatík.

5.4.4.1 Načítanie kaskádových štýlov

CSS je možné definovať v externom súbore, v hlavičke kódu alebo priamo v atribúte *style* tagu. Posledný spôsob je z hľadiska spracovania kódu gramatikami najlepší, pretože nie je nutné dohľadávať štýl v inej štruktúre, alebo na inom mieste kódu. Preto bude súčasťou aplikácie aj CSS parser, ktorý načíta externé súbory a hlavičkové definície a podľa niektorých jednoduchších selektorov naplní inline štýly značiek, ku ktorým hodnoty kaskádových štýlov patria. Tu sa opäť dostávame k problému s možnou nevalidnosťou CSS kódu, avšak budeme predpokladať, že program rozparsuje CSS kód správne. CSS parser je jedným z návrhov v kapitole o možných rozšíreniach programu. Týmto spôsobom bude možné jednoduchšie validovať pravidlá noriem, týkajúce sa použitia kaskádových štýlov.

Externý súbor môže byť v hlavičke definovaný buď podľa absolútnej, alebo relatívnej cesty. V prípade, že sa jedná o relatívnu cestu (adresa na začiatku neobsahuje *http* alebo *https*), pridá sa táto adresa zo zadaného nastavenia. Ak však užívateľ spúšťa validátor so zadaním zdrojového kódu alebo lokálneho súboru, nemal by externé súbory používať. Pri svojom návrhu však nemusí nič

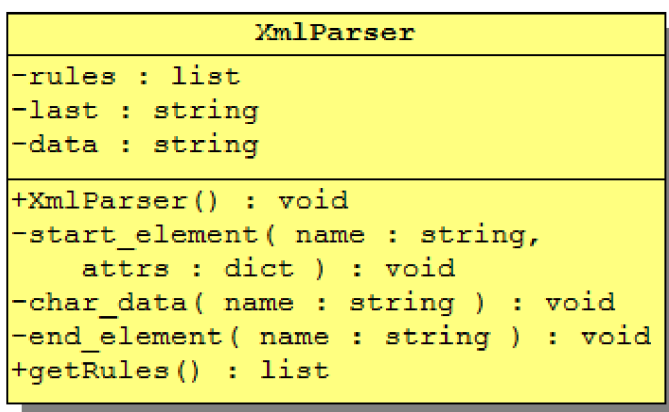
meniť, stačí keď si uloží svoju stránku niektorým webovým prehliadačom alebo programom tak, aby obsahoval aj všetky CSS definície priamo v zdrojovom kóde.

6 Implementácia

Implementácia aplikácie bude prebiehať v jazyku Python a frameworku Django tak, aby k nej bolo možné pristupovať z online rozhrania. Nasleduje popis jednotlivých kľúčových tried, ktoré tvoria štyri základné piliere aplikácie:

1. Načítanie XML súboru s konfiguráciou, jej spracovanie a uloženie pravidiel gramatík do vnútorného nastavenia programu.
2. Spracovanie typu vstupu do aplikácie – url adresa, zdrojový kód alebo súbor.
3. Parsovanie HTML a CSS kódu, jeho preformátovanie na validný kód očakávaný gramatikami.
4. Aplikácia algoritmu, ktorého vstupnými parametrami sú súbory pravidiel normy a pravidiel gramatiky. Výsledok je zapísaný do štruktúry.

6.1 Trieda XmlParser



Obrázok 6.1: Diagram triedy XmlParser

Trieda *XmlParser* (diagram triedy na obrázku 6.1) využíva štandardnú knižnicu jazyka Python *xml.parsers.expat*, ktorá umožňuje jednoduché načítanie xml súborov. Inštancii triedy *expat* sú nastavené objekty funkcií, ktoré spracovávajú značky. Metóda *start_element* triedy *XmlParser* tak získava na vstupe úvodnú značku a atribúty značky, následne metóda *char_data* načítava dáta a metóda *end_element* dostáva na vstup uzavretú párovú značku. Stačí si teda zapamätať úvodnú značku a na základe nej uložiť dáta do správnej štruktúry.

Koreňová značka XML konfigurácie *normrules* v sebe obsahuje jednotlivé súbory pravidiel prístupnosti pod značkou *normrule* s atribútmi, ktoré určujú ich identifikačné číslo a názov. Pod týmito súbormi sa nachádzajú gramatiky pod značkou *gramrule*, ktoré sú časťami validačných pravidiel. Každé validačné pravidlo má nastavený typ chyby a zároveň môže obsahovať viacero

týchto gramatík, ktoré kontrolujú rôzne typy značiek, alebo ich obsahov. Podľa typu značky v nastavení je možné validovať tieto typy:

- Ak je uvedený názov značky, validujú sa len značky s týmto názvom.
- Ak je uvedená hviezdička, validujú sa všetky značky.

Je potrebné rozlišovať podľa typu výsledku algoritmu, aj podľa toho, či vráti pravdu alebo nepravdu. Možnosti pre typ funkcie sú nasledovné:

- Hodnota *true* – validuje sa obsah značky, ak je jazyk generovaný gramatikou, tak vyhlási chybu (napríklad značka obsahuje prázdny atribút).
- Hodnota *false* – ak nie je generovaný gramatikou, vyhlásenie chyby (neobsahuje prázdny atribút).
- Hodnota *content-true* a *content-false* obdobne ako u predchádzajúcich typov len s tým rozdielom, že sa kontroluje obsah párovej značky.
- Hodnota *cleantags-true* a *cleantags-false* obdobne ako u predchádzajúcich typov, avšak kontroluje sa celá html kostra kódu, ktorá neobsahuje žiadne dáta, ani atribúty u značiek.
- Hodnota *exists*, ktorej funkcia kontroluje prítomnosť značky v html kóde.

Nasleduje ukážka XML súboru s konfiguráciou jednej normy a jednej gramatiky:

```
<normrules>
  <normrule normId="1" description="text" title="Czech legal guidelines">
    <rules>
      <gramrule function="false" message="Attribute for alternative description of image is
        missing." typeError="error" tag="img" idRule="2">
        <rules>
          <singlerule>S-><imgF</singlerule>
          <singlerule>F->+F</singlerule>
          <singlerule>F->alt="A"</singlerule>
          <singlerule>A->+A</singlerule>
          <singlerule>A->></singlerule>
        </rules>
      </gramrule>
    </rules>
  </normrule>
```

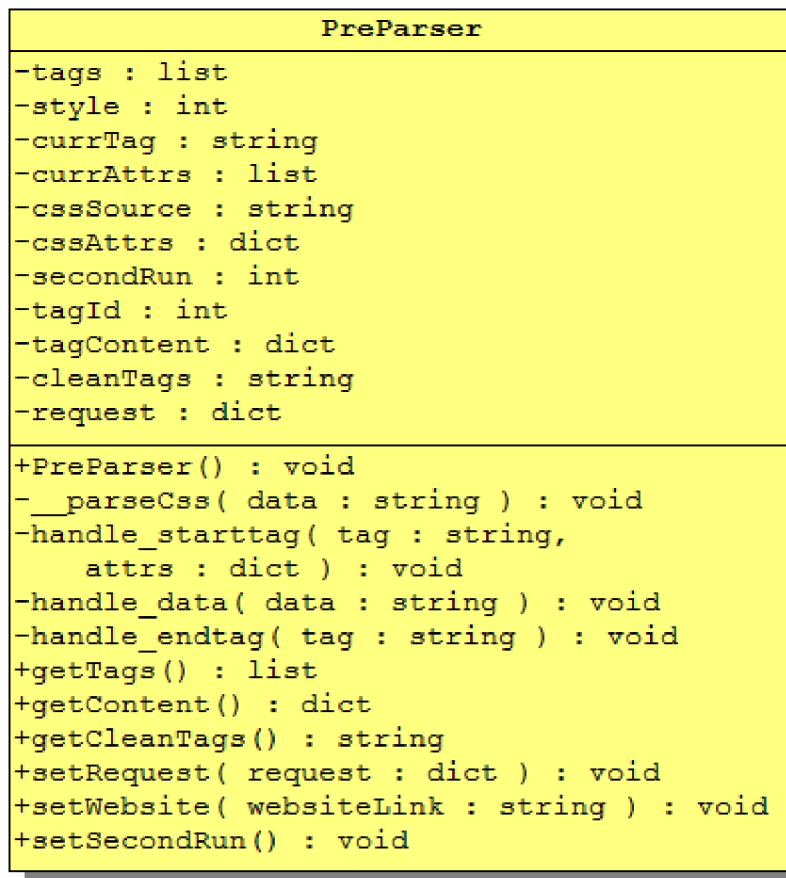
6.2 Pohľady a šablóny

Ako už bolo spomenuté v návrhu, framework Django využíva princíp Model-Viewer-Controller (MVC). Pre pohľad sa vytvárajú funkčné bloky spracovania dát, z ktorých najpodstatnejší je pohľad *validate* určený pre spracovanie parametrov zo vstupu, vytvorenie inštancií tried, nastavenie výstupu do štruktúry a odoslanie týchto dát do šablóny *validate.html*.

Samotný postup užívateľa po stránke je umožnený viacerými šablónami. Na úvodnej stránke má užívateľ možnosť sa rozhodnúť, ktorú voľbu vstupu kódu zvolí a podľa toho je presmerovaný na adekvátnu šablónu. Po odoslaní formulára zo všetkých z týchto možností sa vráti tok programu do pohľadu *validate*, kde sa rozhodne podľa skrytého parametru vo formulári, aký typ načítania sa má spracovať. Výstup výsledkov je potom farebne štruktúrovaný a vypisovaný v šablóne za pomoci Django syntaxe pre šablóny. Tým pádom sú využité dve zložky princípu MVC, viewer a controller, model ako databázu v súčasnom prípade nepotrebujeme.

Táto funkcia – pohľad zároveň zachytáva výnimky, ktoré môžu vyplývať z nesprávneho použitia niektorej z nasledovných tried. V prípade zachytenia výnimky je uložené chybové hlásenie spolu s detailným textom výnimky a toto hlásenie je poslané do šablóny, kde je zobrazené užívateľovi. Odporúčaným postupom je zväčša kontrola správnej štruktúry HTML kódu.

6.3 Trieda PreParser



Obrázok 6.2: Diagram triedy PreParser

Trieda *PreParser* (diagram triedy na obrázku 6.2) je zodpovedná za predprípravu HTML kódu do preformátovanej podoby, ktorá je vhodná k spracovávaniu gramatík. Táto trieda dedí triedu *HTMLParser* z rovnomennej štandardnej knižnice, ktorá má predpripravený spôsob parsovania html kódu.

Po vytvorení inštancie tejto triedy sa v prvom kole skontrolujú a uložia do štruktúry všetky externé CSS súbory a CSS definície v hlavičke webovej stránky tak, aby ich bolo možné priradiť k inline štýlom jednotlivých značiek a následne validovať. Jednoduché parsovanie CSS kódu je implementované v privátnej metóde *_parseCss*.

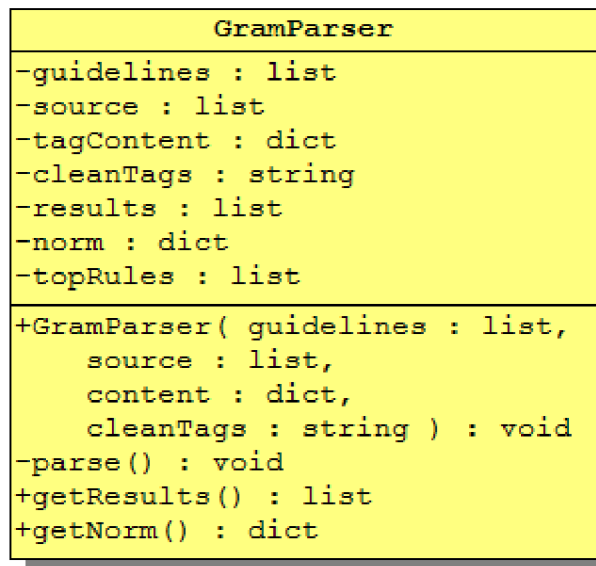
V druhom behu sú tieto hodnoty podľa CSS triedy alebo identifikátora značky vyhľadávané a pridávané k súčasným inline štýlom značiek. Bolo by pravdepodobne možné zvládnuť tento úkon počas jedného behu, ale umožníme takto mať definované CSS vlastnosti až za samotným odkázaním na CSS atribút.

Parsovanie HTML kódu prebieha podobne ako parsovanie XML uvedené vyššie. Po sputení metódy *feed* sú volané jednotlivé metódy podľa vstupného kódu:

- Metóda *handle_starttag* dostáva na vstupe úvodnú značku a atribúty v štruktúre.
- Metóda *handle_data* dostáva na vstupe dáta vnútri párovej značky vo formáte reťazca.
- Metóda *handle_endtag* dostáva na vstupe koncovú značku.

Je teda výhodné si vždy pamätať aká párová značka nám prišla na vstupe a podľa nej vykonať požadovanú akciu. Výstupom z tejto triedy je pole so základnou kostrou HTML kódu bez atribútov a dát. Ďalej je to pole všetkých html značiek s identifikátorom a číslom riadku, na ktorom sa v originálnom kóde nachádzajú. Posledným výstupom je štruktúra s textami, ktoré sa nachádzajú vnútri značiek a identifikátorom značky, aby mohli byť validované podľa značiek. Každá z výstupných štruktúr má svoju verejnú metódu, ktorá sprístupní jej obsah pohľadom *validate*.

6.4 Trieda GramParser



Obrázok 6.3: Diagram triedy GramParser

Trieda *GramParser* (diagram triedy na obrázku 6.3) je určená pre spracovanie gramatík pravidiel a ich vyhodnotenie. Na vstupe očakáva konštruktor tejto triedy pravidlá určitej normy, zdrojovú štruktúru značiek, obsahu a kostru HTML kódu tak, ako boli načítané cez inštancie tried *XmlParser* a *PreParser*.

Metóda *parse* sa stará o validovanie pravidiel podľa typu funkcie, ktorý je popísaný v kapitole 6.1. Do štruktúry *results* sú potom zapisované jednotlivé prehrešky proti prístupnosti spolu s číslom riadku, prioritou chyby, číslom porušeného pravidla a detailnou správou o prehrešku. Táto štruktúra je následne vrátená do pohľadu prostredníctvom verejnej metódy *getResults*. Dáta obsahujúce

informácie o prehreškoch voči prístupnosti sú načítané zo štruktúry uchovávajúcej konfiguráciu zo vstupného XML súboru.

6.5 Testovanie

Počas implementácie prebiehalo zároveň testovanie výsledkov na ukázkových stránkach obsahujúcich všetky prehrešky voči prístupnosti. Následne bolo možné odladiť webové stránky pre rôzne typy chýb. Medzi chyby patrilo problém s kódovaním webových stránok, ktorý bol vyriešený s prečítaním typu kódovania z hlavičky HTTP. Jedným z ďalších problémov bol aj nesprávne štruktúrovaný zdrojový kód webovej stránky, kedy oficiálny HTML parser vrátil výnimku. Táto výnimka bola správne ošetrená a užívateľovi doporučené, aby svoj kód opravil a následne opätovne vyhodnotil. Konkrétne ukážky testovania sú obsiahnuté v nadväzujúcej kapitole 7. Podrobné testovanie je možné nájsť na priloženom CD.

7 Porovnanie výsledkov s inými validátormi

Ďalším krokom pre správne validovanie pravidiel bola krížová kontrola s výsledkami iných validátorov. Samozrejme nie je možné zabezpečiť rovnaké výsledky medzi validátormi, keďže žiadny validátor nemá absolútne presné výsledky. Rôzne pravidlá štandardov môžu byť spracovávané rôznymi prístupmi, a preto nie je možné rozsúdiť, ktorý z týchto prístupov je správny. Z tohto tvrdenia teda vyplýva aj skutočnosť, že nemôžeme určiť správnosť výsledkov validátorov tak, aby sme ich vedeli exaktne porovnať. Môžeme ich len subjektívne ohodnotiť podľa počtu výsledkov, prípadne odporúčaní, ako naše webové stránky vylepšiť.

Jednou zo základných výhod oproti iným validátorom je možnosť upravovať pravidlá jednoducho v rámci XML konfigurácie. Užívateľ má možnosť nahliadnutia do zdrojového kódu pravidiel a zaslať administrátorovi upravenú verziu pravidiel. Táto výzva je uvedená na hlavnej stránke validátora. Samozrejme je k dispozícii aj zdrojový kód aplikácie s konfiguráciou, avšak na nainštalovanie aplikácie potrebuje užívateľ dostupnosť python webhostingu. Prítomnosť tejto konfigurácie očakáva technickú zdatnosť. Na druhú stranu má však užívateľ XML konfiguráciu pod svojou plnou kontrolou. Otvorený prístup k XML konfigurácii by mohlo byť jedno z rozšírení, kde podobne ako pri slobodnom software by sa o správnosť pravidiel štandardov starala komunita.

Pri priamom porovnaní s inými validátormi je potrebné vyzdvihnúť možnosť validácie podľa metodiky WCAG 2.0 na rozdiel od veľkého množstva validátorov, ktoré túto možnosť nemajú, ako napríklad Cynthia Says. Je to však spôsobené pravdepodobne historickým vývojom, za ktorý v súčasnej dobe nikto neodpovedá a validátor žije bez údržby svojim životom. Rovnako oproti Cynthia Says je dostupný prehľadnejší výstup z validátora.

Na druhú stranu existujú možnosti zlepšenia najmä v pridaní funkcionality, ako napríklad grafický výstup podobný výstupu validátora WAVE, či možnosť zobrazenia webovej stránky bez CSS vlastností. Ďalšou oblasťou možného zlepšenia je užívateľské rozhranie vstupného formulára, kde by sa za ukázkový príklad mohol zobrať W3C validátor. Ak bude webová aplikácia úspešná, nie je postupný vývoj týchto vylepšení a rozšírení uvedených v nasledujúcej kapitole nemožný.

Ako referenčný validátor je vybraný Achecker, ktorý umožňuje validovať podľa WCAG 2.0.

7.1.1 Porovnanie výsledkov pre stránky FIT VUT

Check Accessibility By:

Web Page URL HTML File Upload Paste HTML Markup

Address:

Check It

[Options](#)

Accessibility Review

Accessibility Review (Guidelines: [WCAG 2.0 \(Level AA\)](#)) Export Format: PDF Report t

Known Problems(51) Likely Problems (0) Potential Problems (152) HTML Validation

1.4 Distinguishable: Make it easier for users to see and hear content including separating foreground from background.

Success Criteria 1.4.3 Contrast (Minimum) (AA)

Check 302: [The contrast between the colour of visited link text and its background for the element](#)
[WCAG2.0 Level AA.](#)

Obrázok 7.1: Ukážka validácie webu FIT VUT validátorom Achecker podľa WCAG 2.0

Web Accessibility Check

Home WCAG Czech nom Contact

Results for <http://www.fit.vutbr.cz> according to WCAG

Errors: 19, Warnings: 0, Notices: 54
Total: 73

Priority	Rule ID	Reason
error	1.1.1	Attribute for alternative description of image is empty.
<i>Line number: 280</i> <i>Source: </i>		
error	1.1.1	Attribute for alternative description of image is empty.
<i>Line number: 281</i> <i>Source: </i>		

Obrázok 7.2: Ukážka validácie webu FIT VUT vlastným validátorom podľa WCAG 2.0

Ako je možné vidieť na obrázkoch 7.1 a 7.2, počet nájdených problémov majú obe skúmané webové aplikácie podobný. Nie je možné určiť správny presný počet chýb, keďže to závisí od

nastavenia validátora, jeho spracovania pravidiel a spájania podobných chýb do skupín. Výpis problémov samozrejme u oboch validátorov pokračuje nižšie.

7.1.2 Porovnanie výsledkov pre Seznam.cz

Check Accessibility By:

Web Page URL **HTML File Upload** **Paste HTML Markup**

Address:

► [Options](#)

Accessibility Review

Accessibility Review (Guidelines: [WCAG 2.0 \(Level AA\)](#)) Export Format: Report t

Known Problems(22) **Likely Problems (5)** **Potential Problems (639)** **HTML Validation**

1.1 Text Alternatives: Provide text alternatives for any non-text content

Success Criteria 1.1.1 Non-text Content (A)

Check 1: [img element missing alt attribute.](#)

Obrázok 7.3: Ukážka validácie webu Seznam.cz validátorom AChecker podľa WCAG 2.0

Web Accessibility Check

[Home](#) [WCAG](#) [Czech norm](#) [Contact](#)

Results for <http://www.seznam.cz> according to WCAG

Errors: 13, Warnings: 0, Notices: 105
Total: 118

Priority	Rule ID	Reason
error	1.1.1	Attribute for alternative description of image is missing.

Line number: 137
Source: ``

Obrázok 7.4: Ukážka validácie webu Seznam.cz vlastným validátorom podľa WCAG 2.0

V rámci ukážky funkčnosti aj pre iné ako referenčné riešenie obsahuje táto podkapitola porovnanie validácie komerčných webových stránok spoločnosti Seznam.cz. Opäť je možné vidieť na obrázkoch 7.3 a 7.4 podobné výsledky s veľkým počtom možných vylepšení webových stránok tak, aby spĺňal pravidlá.

8 Rozšírenia a možné vylepšenia

Aplikácia umožňuje viaceré možnosti rozšírenia do budúcnosti. Hlavným jadrom je správne a kompletne spracovanie pravidiel jednotlivých noriem. Zvyšná užívateľská funkcionálnosť môže prospieť rastu bázy užívateľov.

8.1 Vylepšenie HTML parsera

V súčasnom stave aplikácie, kedy je využívaná štandardná knižnica Pythonu na parsovanie HTML kódu, dochádza často k výnimkám tejto knižnice, keď HTML kód nie je dostatočne validný. To sa prejavuje nemožnosťou celého validačného procesu. Na jednej strane je to pochopiteľný problém, keď webová stránka obsahuje neexistujúce značky a rôzne preklepy v značkách. Na druhú stranu by ich však mal vedieť HTML parser prekonať, vypísať ako upozornenie a nezastavovať proces validácie.

Z uvedených skutočností vyplýva, že by ako možné rozšírenie programu bolo výhodné naprogramovať vlastný parser, ktorý predpripraví vstup do triedy spracovávajúcej pravidlá gramatik jednotlivých štandardov. Bolo by prípadne možné aj nájsť iný parser ako je súčasný zo štandardnej knižnice Pythonu a zistiť, či spĺňa túto vlastnosť.

8.2 Ukladanie web stránok do databázy

Validátor vytvorený v rámci tejto diplomovej práce umožňuje validovanie web stránok po zadaní URL adresy. Aplikácia celú webovú stránku stiahne, skontroluje a vypíše výsledky. Problém môže nastať v prípade, keď sa užívateľská základňa rozrastie, prípadne by spúšťal validáciu externý robot. Prenos dát a vyťaženie servera môže byť neúnosný pre firmu poskytujúcu webhosting. V tom prípade by hrozila buď migrácia na iný server, alebo veľmi vysoké poplatky.

Riešením tohto problému by bolo použitie databázy, do ktorej by sme ukladali zdrojové adresy webových stránok a ich výsledky. Keďže protokol http vo svojej hlavičke posiela aj čas zmeny, mohli by ukladať zároveň aj túto informáciu a použiť ju ako kľúč pri následnej validácii. Najskôr by sme skontrolovali len hlavičku, či zdrojový kód stránky naozaj potrebujeme. V prípade, že nie, mohli by sme výsledky načítať z databázy a rovno zobrazit' výsledky. Takto by sa celý proces sťahovania väčšieho množstva dát zoptimalizoval.

Druhou možnosťou je ukladanie celých často sťahovaných webových stránok a ich CSS externých súborov, pretože aj tie tvoria veľkú časť dát. Ak správca webovej stránky pri vývoji upravuje stránku a priebežne ju kontroluje validátorom, je veľká pravdepodobnosť, že nie všetky externé súbory sa budú meniť.

V ideálnom prípade by sme teda najskôr skontrolovali hlavičku podľa http protokolu webovej stránky, či bola zmenená. Ak bola zmenená, skontrolovali by sme časy zmeny u externých súborov tak, aby sme ich nemuseli sťahovať, ale načítali ich z lokálneho úložiska. V prípade, že by bola už úvodná stránka nezmenená, vypísali by sme rovno výsledky z databázy. Týmto postupom by sme boli schopní ušetriť dátový prenos.

8.3 Vylepšenie algoritmu spracovania gramatík

Ďalším bodom k možnému zlepšeniu rýchlosti spracovania gramatík je optimalizácia algoritmu spracovania bezkontextových gramatík. V súčasnom riešení je vstupný kód spracovaný sprava doľava a vždy, keď nájde hľadané pravidlo a nahradí ho za nonterminál, je posunutá pozícia spracovávaného textu opäť úplne doprava. Tento posun by pravdepodobne nebol vždy nutný, mohla by sa kontrolovať dĺžka pravých strán pravidiel a pozícia posunúť len o maximálnu veľkosť pravej strany pravidla smerom doprava. Implementované riešenie postačuje pre potreby tejto diplomovej práce, avšak pri väčšej záťaži a využití validátora by bola táto úprava vhodná.

Druhým problémom je spracovanie kontroly križenia a vnárania značiek. Pozícia je rovnako vždy posúvaná do stredu vstupného kódu, aby bola opäť posunutá úplne doprava. Algoritmus pri zadaní nastavenia by mohol vyhľadávať zo stredu oboma smermi. Bolo by však nutné overiť funkčnosť a to, či by sa necyklil, prípadne sa naozaj dostal k správny výsledkom. Inou možnosťou tohto riešenia je plne zapojiť funkčnosť programu priamo do vynoveného HTML parsera a kontrolovať toto pravidlo už pri predpríprave kódu určeného k spracovaniu.

8.3.1 Pravidlo č. 22 českej normy

Vyhodnocovanie tohto pravidla súvisí opäť s potrebou vylepšenia algoritmu, prípadne nového postupu na jeho spracovanie. Samotné pravidlo hovorí o existencii popisu vystihujúceho podstatu pre každý formulárový prvok. Konkrétne ide o previazanie značky *label* a *input* prostredníctvom atribútov *for* (u *label*) a *id* (u *input*) jednoznačným identifikátorom pre danú stránku.

Toto pravidlo znamená v súčasnom algoritme problém, hoci je podobné s križením a zanáraním značiek, keďže na oboch stranách od terminálu na pravej strane pravidla potrebuje rovnaké znaky. Problém je však v tom, že jednoznačný identifikátor nemá preddefinované značky ako je to u pravidla križenia značiek. Preto by bolo naozaj nutné vytvoriť gramatiku s veľkým množstvom pravidiel, ktorá by pokrývala všetky znaky. Okrem toho je však ďalším problémom to, že sa značky môžu nachádzať napríklad v tabuľke a medzi nimi bude väčšie množstvo textu a značiek. Rovnako sa môžu jednotlivé previazané značky *input* a *for* križiť, a tým by bolo vyhodnotenie ešte viac sťažené.

V každom prípade by však kontrola tohto pravidla musela byť podrobená väčšiemu výskumu, či je možné ju vôbec spracovávať bezkontextovými gramatikami, alebo je siahnuť po inom prístupe. Hoci generovanie kódu – jazyka by malo byť bezkontextovými gramatikami možné, spätná kontrola jeho platnosti je komplikovaná.

8.4 Aplikácia pre tablet a smartphone

Ďalšou možnosťou ako podporovať vývoj inováciami by bolo vytvorenie špeciálnej verzie pre tablet alebo smartphone. Táto aplikácia by využívala ako backend (server pre spracovanie vstupov z užívateľského rozhrania) naprogramovanú aplikáciu. Bolo by len nutné vytvoriť rozhranie pre komunikáciu medzi backendom validátora a aplikáciou bežiacou pod iOS alebo Androidom. Pred vývojom by ale bolo vhodné získať pomocou dotazníkov reálny záujem užívateľov, alebo zmapovať konkurenciu a jej počty užívateľov.

8.5 Podporný toolbar pre webový browser

Posledným zmieným rozšírením je vytvorenie samostatného toolbaru, ktorý by ako toolbary podobných projektov existoval nezávisle od servera. Tým pádom by uľahčil prácu správcov webových stránok, ktorí by mohli validovať svoje stránky aj pri práci offline a zároveň by preniesol záťaž zo servera na užívateľské počítače. Takýto prístup by však vyžadoval portovanie aplikácie do programovacieho jazyka podporujúceho tvorbu toolbarov, ktorým je napríklad vo webovom prehliadači Firefox skriptovací jazyk JavaScript.

9 Záver

V rámci tejto diplomovej práce som sa detailne zamerlal na súčasnú situáciu štandardov webovej prístupnosti. Samotná webová prístupnosť je pozitívnym odrazom v celosvetovom dianí v snahe o potlačenie diskriminácie postihnutých občanov. Okrem toho webové prístupné stránky sú prehľadné a majú pozitívny vplyv na použiteľnosť pre všetkých užívateľov Internetu. Prístupné webové stránky sú tiež kompatibilnejšie v rámci veľkej škály webových prehliadačov.

Prvou časťou prínosu v rámci tejto diplomovej práce tkvie v rešerši súčasných internetových štandardov, keď sa metodika WCAG stala akousi dogmou prístupnosti webových stránok. Z WCAG vychádzajú jednotlivé, z časti upravené štandardy jednotlivých krajín, ktoré sú legálne vyžadované pre webové portály v štátnej sfére. Vo Veľkej Británii však požiadavky zasahujú aj do privátnej sféry.

Ukážkou nesprávneho použitia kódu HTML, CSS alebo JavaScript získa čitateľ základný prehľad jednak o týchto jazykoch, ako aj pochopenie akých prehreškov je možné sa dopustiť voči prístupnosti webových stránok. Následná formalizácia jednotlivých pravidiel webovej prístupnosti pre českú a pre medzinárodnú normu WCAG sa považuje za základnú a najnáročnejšiu časť tejto práce. Skĺbením teoretických znalostí zo štúdia teoretickej informatiky a vlastných nápadov som získal algoritmus schopný spracovávať zdrojový vstupný kód na základe konfigurácie. Opätovnou formou ukážky dokáže čitateľ pochopiť princíp, na ktorom je algoritmus pre spracovávanie gramatík postavený. Nasledujúci výpis pravidiel spolu s konkrétnou analýzou zhrnie všetky predchádzajúce poznatky a jeho výsledkom je kompletná konfigurácia validátora uvedená v prílohe.

Návrh aplikácie rozoberá jednotlivé konkurenčné validátory a ich funkcionality tak, aby sme na základe nej mohli určiť funkcionality pre bežný validátor. V návrhu sa tiež venujem pokročilejšej funkcionalite, ktorá by v budúcnosti v prípade úspechu mohla byť využitá ako rozšírenie aplikácie medzi užívateľmi. Konkrétnymi problémami spojenými s implementáciou sa zaoberám v posledných podkapitolách. Podstatným faktom je výber frameworku Django pre programovací jazyk Python ako východzí systém určený k implementácii. Pri implementovaní validátora som narazil na množstvo problémov, ktoré sú v týchto kapitolách popísané. Triedy sú vytvorené s ohľadom na funkčnosť a rozširiteľnosť tak, aby spĺňali všetky podmienky objektového programovania.

Následnými testami a porovnaním s ostatnými známymi a dostupnými validátormi webovej prístupnosti som sa utvrdil v tom, že pri dostatočnej vôli nie je nemožné aplikáciu vypilovať do štádia, v ktorom môže konkurovať najlepším validátorom. Overenie prebiehalo na testoch referenčnej webovej stránky FIT VUT.

Počas implementácie som taktiež získal množstvo nápadov, ktorými je možné aplikáciu vylepšiť a v budúcnosti ďalej rozvíjať tak, aby sa dostala medzi najpoužívanejšie aplikácie vo svojom odbore. Verím teda, že táto diplomová práca mala teoretický význam pre mňa a bude mať praktický význam pre užívateľov.

Literatúra

- [1] ŠPINAR, David. Tvoříme přístupné webové stránky. Brno: Zoner Press, 2004. 360 s. ISBN 80-86815-11-0.
- [2] Resources in the ATC. IT Connect [online]. [cit. 2011-12-20]. Dostupné z WWW: <<http://www.washington.edu/itconnect/accessibility/at/resources.html>>.
- [3] All About Trackballs. Geekhack forums [online]. [cit. 2011-12-20]. Dostupné z WWW: <<http://geekhack.org/showwiki.php?title=All+About+Trackballs> >.
- [4] One Handed Keyboards. Maltron [online]. [cit. 2011-12-20]. Dostupné z WWW: <<http://www.onehandedkeyboard.com/maltron.html> >.
- [5] Kolik vlastně je handicapovaných uživatelů webu? POSLEPU [online]. [cit. 2011-12-08]. Dostupné z WWW: <<http://poslepu.blogspot.com/2010/04/kolik-vlastne-je-handicapovanych.html> >.
- [6] Web Content Accessibility Guidelines (WCAG) Overview. W3C [online]. [cit. 2011-12-08]. Dostupné z WWW: <<http://www.w3.org/WAI/intro/wcag>>.
- [7] Vyhláška č. 64/2008 Sb. MVČR [online]. [cit. 2012-03-20]. Dostupné z WWW: <<http://www.mvcr.cz/clanek/vyhlaska-c-64-2008-sb-o-forme-uverejnovani-informaci-souvisejicich-s-vykonem-verejne-spravy-prostrednictvim-webovych-stranek-pro-osoby-se-zdravotnim-postizenim-vyhlaska-o-pristupnosti-10.aspx>>
- [8] Monitorovanie prístupnosti webových stránok [online]. [cit. 2012-01-05]. Dostupné z WWW: <<http://informatizacia.sk/monitorovanie-pristupnosti-webovych-stranok/2824s>>.
- [9] Štandardy pre informačné systémy verejnej správy [online]. [cit. 2012-01-05]. Dostupné z WWW: <http://www.informatizacia.sk/standardy-is-vs/596s#std_dokumenty>.
- [10] BS8878 web accessibility standard [online]. [cit. 2011-12-28]. Dostupné z WWW: <<http://www.hassellinclusion.com/bs8878/>>.
- [11] Swedish National Guidelines for Public Sector Websites [online]. [cit. 2011-12-28]. Dostupné z WWW: <<http://www.eutveckling.se/static/doc/swedish-guidelines-public-sector-websites.pdf>>.
- [12] Section 508 Homepage [online]. [cit. 2011-12-30]. Dostupné z WWW: <<http://www.access-board.gov/508.htm>>.
- [13] Common Look and Feel for the Internet 2.0 [online]. [cit. 2011-12-30]. Dostupné z WWW: <<http://www.tbs-sct.gc.ca/clf2-nsi2/index-eng.asp>>.
- [14] SCHAFER, Steven M. HTML, XHTML a CSS Bible pro tvorbu WWW stránek. 4.vydanie. Praha: Grada, 2009. 647s. ISBN 978-80-247-2850-6
- [15] Dive Into HTML5 [online]. [cit. 2012-04-28]. Dostupné z WWW: <<http://kniha.html5.cz>>
- [16] ŠKULTÉTY, Rastislav. JavaScript: Programujeme internetové aplikace. 2. aktualizované vydání. Brno: Computer Press, 2001. 228 s. ISBN 80-7226-457-5.
- [17] SKONNARD, Aaron; GUDGIN, Martin: XML pohotová referenční příručka. Praha : Grada. 2006. 342 s. ISBN 80-247-0972-4.
- [18] Jazyky a jejich reprezentace, algebra formálních jazyků [online]. [cit. 2012-05-05]. Dostupné z WWW: <<http://www.fit.vutbr.cz/study/courses/TIN/public/Prednasky/tin-pr01-rj1.pdf>>
- [19] Pravidla tvorby přístupného webu [online]. [cit. 2012-04-06]. Dostupné z WWW: <<http://www.pravidla-pristupnosti.cz>>

- [20] Český překlad části metodiky Web Content Accessibility Guidelines (WCAG) 2.0 [online]. [cit. 2012-04-28]. Dostupné z WWW: <<http://blindfriendly.cz/wcag20>>
- [21] Techniques and Failures for Web Content Accessibility Guidelines 2.0 [online]. [cit. 2012-03-10]. Dostupné z WWW: <<http://www.w3.org/TR/WCAG20-TECHS>>
- [22] PECINOVSKÝ, Rudolf. Návrhové vzory: 33 vzorových postupů pro objektové programování. Brno: Cpress, 2007. 527 s. ISBN 978-80-251-1582-4.

Zoznam príloh

Príloha A: XML konfigurácia pravidiel.

Príloha B: CD so zdrojovým kódom aplikácie

Príloha A

<normrules>

Česká norma pravidiel

Pravidlo č. 2

```
<normrule normId="1" description="text" title="Czech legal guidelines">
  <rules>
    <gramrule function="false" message="Attribute for alternative
description of image is missing." typeError="error" tag="img" idRule="2">
      <rules>
        <singlerule>S-><imgF</singlerule>
        <singlerule>F->+F</singlerule>
        <singlerule>F->alt="A</singlerule>
        <singlerule>A->+A</singlerule>
        <singlerule>A->></singlerule>
      </rules>
    </gramrule>
    <gramrule function="true" message="Attribute for alternative
description of image is empty." typeError="error" tag="img" idRule="2">
      <rules>
        <singlerule>S-><imgF</singlerule>
        <singlerule>F->+F</singlerule>
        <singlerule>F->alt=""A</singlerule>
        <singlerule>A->+A</singlerule>
        <singlerule>A->></singlerule>
      </rules>
    </gramrule>
    <gramrule function="true" message="Alternative text of image is not
supplied correctly. It should describe the content of image." typeError="warning"
tag="img" idRule="2">
      <rules>
        <singlerule>S-><imgF</singlerule>
        <singlerule>F->+F</singlerule>
        <singlerule>F->alt="obrazek"A</singlerule>
        <singlerule>F->alt="obrazok"A</singlerule>
        <singlerule>F->alt="image"A</singlerule>
        <singlerule>A->+A</singlerule>
        <singlerule>A->></singlerule>
      </rules>
    </gramrule>
```

Pravidlo č. 3

```
<gramrule function="true" message="Not correct redirect. Redirect
should not rely on javascript." typeError="error" tag="*" idRule="3">
  <rules>
    <singlerule>S-><F</singlerule>
    <singlerule>F->+F</singlerule>
    <singlerule>F->onclick="C</singlerule>
    <singlerule>F->onsubmit="C</singlerule>
    <singlerule>A->="B</singlerule>
    <singlerule>A->="C</singlerule>
    <singlerule>C->+C</singlerule>
    <singlerule>C->>window.locationB</singlerule>
    <singlerule>B->+B</singlerule>
    <singlerule>B->"A</singlerule>
    <singlerule>A->+A</singlerule>
    <singlerule>A->></singlerule>
  </rules>
</gramrule>
```

Pravidlo č. 4

```
<gramrule function="true" message="Note that information on website
should be available in same order as in source code. Check if the block are not
shuffled by CSS positioning." typeError="notice" tag="*" idRule="4">
  <rules>
    <singlerule>S-><F</singlerule>
    <singlerule>F->+F</singlerule>
    <singlerule>F->style="C</singlerule>
    <singlerule>A->="B</singlerule>
    <singlerule>A->="C</singlerule>
    <singlerule>C->+C</singlerule>
    <singlerule>C->position:B</singlerule>
    <singlerule>B->+B</singlerule>
    <singlerule>B->"A</singlerule>
    <singlerule>A->+A</singlerule>
    <singlerule>A->></singlerule>
  </rules>
</gramrule>
```

Pravidlo č. 7

```
<gramrule function="true" message="Font size should not be set by px,
cm, in, mm, pc or pt." typeError="error" tag="*" idRule="7">
  <rules>
    <singlerule>S-><F</singlerule>
    <singlerule>F->+F</singlerule>
    <singlerule>F->style="C</singlerule>
```

```

    <single-rule>A->="B</single-rule>
    <single-rule>A->="C</single-rule>
    <single-rule>C->font-size:D</single-rule>
    <single-rule>B->:D</single-rule>
    <single-rule>D->pxB</single-rule>
    <single-rule>D->ptB</single-rule>
    <single-rule>D->pcB</single-rule>
    <single-rule>D->inB</single-rule>
    <single-rule>D->cmB</single-rule>
    <single-rule>D->mmB</single-rule>
    <single-rule>D->+D</single-rule>
    <single-rule>B->+B</single-rule>
    <single-rule>B->"A</single-rule>
    <single-rule>A->+A</single-rule>
    <single-rule>A->></single-rule>
  </rules>
</gram-rule>

```

Pravidlo č. 10

```

<gram-rule function="true" message="Not correct redirect. Redirect
should not rely on javascript and its events." typeError="warning" tag="*"
idRule="10">

```

```

  <rules>
    <single-rule>S-><F</single-rule>
    <single-rule>F->+F</single-rule>
    <single-rule>F->onmouseover="C</single-rule>
    <single-rule>F->onmouseout="C</single-rule>
    <single-rule>F->onmousemove="C</single-rule>
    <single-rule>F->onmousedown="C</single-rule>
    <single-rule>F->onmouseup="C</single-rule>
    <single-rule>A->="B</single-rule>
    <single-rule>A->="C</single-rule>
    <single-rule>C->+C</single-rule>
    <single-rule>C->>window.locationB</single-rule>
    <single-rule>B->+B</single-rule>
    <single-rule>B->"A</single-rule>
    <single-rule>A->+A</single-rule>
    <single-rule>A->></single-rule>
  </rules>
</gram-rule>

```

Pravidlo č. 11

```

<gram-rule function="true" message="Please check if hyperlink opening in
new window is visually different from other hyperlinks." typeError="notice" tag="a"
idRule="11">

```

```

  <rules>

```

```

        <singlerule>S-><aF</singlerule>
        <singlerule>F->+F</singlerule>
        <singlerule>F->target="_new"A</singlerule>
        <singlerule>F->target="_blank"A</singlerule>
        <singlerule>A->+A</singlerule>
        <singlerule>A->></singlerule>
    </rules>
</gramrule>

```

Pravidlo č. 12

```

<gramrule function="true" message="Tag blink is not permitted to use."
typeError="error" tag="blink" idRule="12">

```

```

    <rules>
        <singlerule>S-><blinkF</singlerule>
        <singlerule>F->+F</singlerule>
        <singlerule>F->></singlerule>
    </rules>
</gramrule>

```

Pravidlo č. 17

```

<gramrule function="true" message="Hyperlinks should not be hidden by
CSS, because some devices may show them anyway." typeError="warning" tag="a"
idRule="17">

```

```

    <rules>
        <singlerule>S-><aF</singlerule>
        <singlerule>F->+F</singlerule>
        <singlerule>F->style="C</singlerule>
        <singlerule>A->="B</singlerule>
        <singlerule>A->="C</singlerule>
        <singlerule>C->display: noneB</singlerule>
        <singlerule>C->visibility: hiddenB</singlerule>
        <singlerule>B->+B</singlerule>
        <singlerule>B->"A</singlerule>
        <singlerule>A->+A</singlerule>
        <singlerule>A->></singlerule>
    </rules>
</gramrule>

```

Pravidlo č. 21

```

<gramrule function="exists" message="Tag title does not exists."
typeError="error" tag="title" idRule="21">

```

```

    <rules></rules>
</gramrule>

```

```

<gramrule function="content-true" message="Title should not contain
generic names, but real title of website." typeError="error" tag="title"
idRule="21">

```

```

    <rules>
      <singlelerule>S-><title</singlelerule>
      <singlelerule>S-><Unknown title</singlelerule>
      <singlelerule>S-><Neznama stranka</singlelerule>
    </rules>
  </gramrule>
  <gramrule function="content-true" message="Title should not be empty."
  typeError="error" tag="title" idRule="21">
    <rules>
      <singlelerule>S-><</singlelerule>
    </rules>
  </gramrule>

```

Pravidlo č. 24

```

  <gramrule function="content-true" message="Hyperlink should not contain
  generic name, but real title of website where hyperlink leads." typeError="error"
  tag="title" idRule="24">
    <rules>
      <singlelerule>S-><zde</singlelerule>
      <singlelerule>S-><odkaz</singlelerule>
      <singlelerule>S-><klikni zde</singlelerule>
      <singlelerule>S-><tento odkaz</singlelerule>
      <singlelerule>S-><kliknete zde</singlelerule>
    </rules>
  </gramrule>
  <gramrule function="content-true" message="Hyperlink should not be
  empty." typeError="error" tag="title" idRule="24">
    <rules>
      <singlelerule>S-><</singlelerule>
    </rules>
  </gramrule>
  <gramrule function="false" message="Attribute for title of hyperlink is
  missing." typeError="warning" tag="a" idRule="24">
    <rules>
      <singlelerule>S-><aF</singlelerule>
      <singlelerule>F->+F</singlelerule>
      <singlelerule>F->title="A"</singlelerule>
      <singlelerule>A->+A</singlelerule>
      <singlelerule>A->></singlelerule>
    </rules>
  </gramrule>
  <gramrule function="true" message="Attribute for title of hyperlink is
  empty." typeError="warning" tag="a" idRule="24">
    <rules>
      <singlelerule>S-><aF</singlelerule>
      <singlelerule>F->+F</singlelerule>

```

```

        <singlerule>F->title=""A</singlerule>
        <singlerule>A->+A</singlerule>
        <singlerule>A->></singlerule>
    </rules>
</gramrule>
<gramrule function="true" message="Check if hyperlink to file is
visually different than ordinary hyperlink." typeError="notice" tag="a"
idRule="24">
    <rules>
        <singlerule>S-><aF</singlerule>
        <singlerule>F->+F</singlerule>
        <singlerule>F->href="B</singlerule>
        <singlerule>A->="B</singlerule>
        <singlerule>B->+B</singlerule>
        <singlerule>B->zip"A</singlerule>
        <singlerule>B->rar"A</singlerule>
        <singlerule>B->tar"A</singlerule>
        <singlerule>B->tgz"A</singlerule>
        <singlerule>B->pdf"A</singlerule>
        <singlerule>B->doc"A</singlerule>
        <singlerule>B->docx"A</singlerule>
        <singlerule>B->xls"A</singlerule>
        <singlerule>B->xlsx"A</singlerule>
        <singlerule>B->ppt"A</singlerule>
        <singlerule>A->+A</singlerule>
        <singlerule>A->></singlerule>
    </rules>
</gramrule>

```

Pravidlo č. 25

```

<gramrule function="false" message="Attribute for title of frame is
missing." typeError="error" tag="frame" idRule="25">
    <rules>
        <singlerule>S-><frameF</singlerule>
        <singlerule>F->+F</singlerule>
        <singlerule>F->title="A</singlerule>
        <singlerule>A->+A</singlerule>
        <singlerule>A->></singlerule>
    </rules>
</gramrule>
<gramrule function="true" message="Attribute for title of frame is
empty." typeError="error" tag="frame" idRule="25">
    <rules>
        <singlerule>S-><frameF</singlerule>
        <singlerule>F->+F</singlerule>
        <singlerule>F->title=""A</singlerule>
    </rules>

```



```

    <singlerule>A->+A</singlerule>
    <singlerule>A->></singlerule>
  </rules>
</gramrule>

```

Pravidlo č. 27

```

<gramrule function="cleantags-false" message="Tags shall not cross each
other." typeError="error" tag="*" idRule="27">

```

```

  <rules>
    <singlerule>S-><blink></blink></singlerule>
    <singlerule>S-><blink>S</blink></singlerule>
    <singlerule>S-><!DOCTYPE></singlerule>
    <singlerule>S-><a></a></singlerule>
    <singlerule>S-><a>S</a></singlerule>
    <singlerule>S-><abbr></abbr></singlerule>
    <singlerule>S-><abbr>S</abbr></singlerule>
    <singlerule>S-><acronym></acronym></singlerule>
    <singlerule>S-><acronym>S</acronym></singlerule>
    <singlerule>S-><address></address></singlerule>
    <singlerule>S-><address>S</address></singlerule>
    <singlerule>S-><applet></applet></singlerule>
    <singlerule>S-><applet>S</applet></singlerule>
    <singlerule>S-><b></b></singlerule>
    <singlerule>S-><b>S</b></singlerule>
    <singlerule>S-><bdo></bdo></singlerule>
    <singlerule>S-><bdo>S</bdo></singlerule>
    <singlerule>S-><big></big></singlerule>
    <singlerule>S-><big>S</big></singlerule>
    <singlerule>S-><blockquote></blockquote></singlerule>
    <singlerule>S-><blockquote>S</blockquote></singlerule>
    <singlerule>S-><body></body></singlerule>
    <singlerule>S-><body>S</body></singlerule>
    <singlerule>S-><button></button></singlerule>
    <singlerule>S-><button>S</button></singlerule>
    <singlerule>S-><caption></caption></singlerule>
    <singlerule>S-><caption>S</caption></singlerule>
    <singlerule>S-><center></center></singlerule>
    <singlerule>S-><center>S</center></singlerule>
    <singlerule>S-><cite></cite></singlerule>
    <singlerule>S-><cite>S</cite></singlerule>
    <singlerule>S-><code></code></singlerule>
    <singlerule>S-><code>S</code></singlerule>
    <singlerule>S-><colgroup></colgroup></singlerule>
    <singlerule>S-><colgroup>S</colgroup></singlerule>
    <singlerule>S-><dd></dd></singlerule>
    <singlerule>S-><dd>S</dd></singlerule>

```

<singlerule>S-></singlerule>
<singlerule>S->S</singlerule>
<singlerule>S-><dfn></dfn></singlerule>
<singlerule>S-><dfn>S</dfn></singlerule>
<singlerule>S-><dir></dir></singlerule>
<singlerule>S-><dir>S</dir></singlerule>
<singlerule>S-><div></div></singlerule>
<singlerule>S-><div>S</div></singlerule>
<singlerule>S-><dl></dl></singlerule>
<singlerule>S-><dl>S</dl></singlerule>
<singlerule>S-><dt></dt></singlerule>
<singlerule>S-><dt>S</dt></singlerule>
<singlerule>S-></singlerule>
<singlerule>S->S</singlerule>
<singlerule>S-><fieldset></fieldset></singlerule>
<singlerule>S-><fieldset>S</fieldset></singlerule>
<singlerule>S-></singlerule>
<singlerule>S->S</singlerule>
<singlerule>S-><form></form></singlerule>
<singlerule>S-><form>S</form></singlerule>
<singlerule>S-><frameset></frameset></singlerule>
<singlerule>S-><frameset>S</frameset></singlerule>
<singlerule>S-><h1></h1></singlerule>
<singlerule>S-><h1>S</h1></singlerule>
<singlerule>S-><h2></h2></singlerule>
<singlerule>S-><h2>S</h2></singlerule>
<singlerule>S-><h3></h3></singlerule>
<singlerule>S-><h3>S</h3></singlerule>
<singlerule>S-><h4></h4></singlerule>
<singlerule>S-><h4>S</h4></singlerule>
<singlerule>S-><h5></h5></singlerule>
<singlerule>S-><h5>S</h5></singlerule>
<singlerule>S-><h6></h6></singlerule>
<singlerule>S-><h6>S</h6></singlerule>
<singlerule>S-><head></head></singlerule>
<singlerule>S-><head>S</head></singlerule>
<singlerule>S-><html></html></singlerule>
<singlerule>S-><html>S</html></singlerule>
<singlerule>S-><i></i></singlerule>
<singlerule>S-><i>S</i></singlerule>
<singlerule>S-><iframe></iframe></singlerule>
<singlerule>S-><iframe>S</iframe></singlerule>
<singlerule>S-><ins></ins></singlerule>
<singlerule>S-><ins>S</ins></singlerule>
<singlerule>S-><kbd></kbd></singlerule>
<singlerule>S-><kbd>S</kbd></singlerule>

<single rule>S-><label></label></single rule>
<single rule>S-><label>S</label></single rule>
<single rule>S-><legend></legend></single rule>
<single rule>S-><legend>S</legend></single rule>
<single rule>S-></single rule>
<single rule>S->S</single rule>
<single rule>S-><map></map></single rule>
<single rule>S-><map>S</map></single rule>
<single rule>S-><menu></menu></single rule>
<single rule>S-><menu>S</menu></single rule>
<single rule>S-><noframes></noframes></single rule>
<single rule>S-><noframes>S</noframes></single rule>
<single rule>S-><noscript></noscript></single rule>
<single rule>S-><noscript>S</noscript></single rule>
<single rule>S-><object></object></single rule>
<single rule>S-><object>S</object></single rule>
<single rule>S-></single rule>
<single rule>S->S</single rule>
<single rule>S-><optgroup></optgroup></single rule>
<single rule>S-><optgroup>S</optgroup></single rule>
<single rule>S-><option></option></single rule>
<single rule>S-><option>S</option></single rule>
<single rule>S-><p></p></single rule>
<single rule>S-><p>S</p></single rule>
<single rule>S-><pre></pre></single rule>
<single rule>S-><pre>S</pre></single rule>
<single rule>S-><q></q></single rule>
<single rule>S-><q>S</q></single rule>
<single rule>S-><s></s></single rule>
<single rule>S-><s>S</s></single rule>
<single rule>S-><samp></samp></single rule>
<single rule>S-><samp>S</samp></single rule>
<single rule>S-><script></script></single rule>
<single rule>S-><script>S</script></single rule>
<single rule>S-><select></select></single rule>
<single rule>S-><select>S</select></single rule>
<single rule>S-><small></small></single rule>
<single rule>S-><small>S</small></single rule>
<single rule>S-></single rule>
<single rule>S->S</single rule>
<single rule>S-><strike></strike></single rule>
<single rule>S-><strike>S</strike></single rule>
<single rule>S-></single rule>
<single rule>S->S</single rule>
<single rule>S-><style></style></single rule>
<single rule>S-><style>S</style></single rule>

<singlerule>S-></singlerule>
<singlerule>S->_S</singlerule>
<singlerule>S-></singlerule>
<singlerule>S->^S</singlerule>
<singlerule>S-><table></table></singlerule>
<singlerule>S-><table>S</table></singlerule>
<singlerule>S-><tbody></tbody></singlerule>
<singlerule>S-><tbody>S</tbody></singlerule>
<singlerule>S-><td></td></singlerule>
<singlerule>S-><td>S</td></singlerule>
<singlerule>S-><textarea></textarea></singlerule>
<singlerule>S-><textarea>S</textarea></singlerule>
<singlerule>S-><tfoot></tfoot></singlerule>
<singlerule>S-><tfoot>S</tfoot></singlerule>
<singlerule>S-><th></th></singlerule>
<singlerule>S-><th>S</th></singlerule>
<singlerule>S-><thead></thead></singlerule>
<singlerule>S-><thead>S</thead></singlerule>
<singlerule>S-><title></title></singlerule>
<singlerule>S-><title>S</title></singlerule>
<singlerule>S-><tr></tr></singlerule>
<singlerule>S-><tr>S</tr></singlerule>
<singlerule>S-><tt></tt></singlerule>
<singlerule>S-><tt>S</tt></singlerule>
<singlerule>S-><u></u></singlerule>
<singlerule>S-><u>S</u></singlerule>
<singlerule>S-></singlerule>
<singlerule>S->S</singlerule>
<singlerule>S-><var></var></singlerule>
<singlerule>S-><var>S</var></singlerule>
<singlerule>S-><xmp></xmp></singlerule>
<singlerule>S-><xmp>S</xmp></singlerule>
<singlerule>S-></singlerule>
<singlerule>S-></singlerule>
<singlerule>S-><area></area></singlerule>
<singlerule>S-><area></singlerule>
<singlerule>S-><base></base></singlerule>
<singlerule>S-><base></singlerule>
<singlerule>S-><basefont></basefont></singlerule>
<singlerule>S-><basefont></singlerule>
<singlerule>S->
</br></singlerule>
<singlerule>S->
</singlerule>
<singlerule>S-><col></col></singlerule>
<singlerule>S-><col></singlerule>
<singlerule>S-><frame></frame></singlerule>
<singlerule>S-><frame></singlerule>

```

    <singlerule>S-><hr></hr></singlerule>
    <singlerule>S-><hr></singlerule>
    <singlerule>S-><input></input></singlerule>
    <singlerule>S-><input></singlerule>
    <singlerule>S-><link></link></singlerule>
    <singlerule>S-><link></singlerule>
    <singlerule>S-><meta></meta></singlerule>
    <singlerule>S-><meta></singlerule>
    <singlerule>S-><param></param></singlerule>
    <singlerule>S-><param></singlerule>
    <singlerule>S->SS</singlerule>
  </rules>
</gramrule>

```

Pravidlo č. 28

<gramrule function="false" message="Attribute for lang in HTML tag is missing." typeError="error" tag="html" idRule="28">

```

  <rules>
    <singlerule>S-><htmlF</singlerule>
    <singlerule>F->+F</singlerule>
    <singlerule>F->lang="A</singlerule>
    <singlerule>F->xml:lang="A</singlerule>
    <singlerule>A->+A</singlerule>
    <singlerule>A->></singlerule>
  </rules>

```

</gramrule>

<gramrule function="true" message="Attribute for lang in HTML tag is empty." typeError="error" tag="html" idRule="28">

```

  <rules>
    <singlerule>S-><htmlF</singlerule>
    <singlerule>F->+F</singlerule>
    <singlerule>F->lang="A</singlerule>
    <singlerule>F->xml:lang=""A</singlerule>
    <singlerule>A->+A</singlerule>
    <singlerule>A->></singlerule>
  </rules>

```

</gramrule>

</rules>

</normrule>

WCAG

<normrule normId="2" description="text" title="WCAG">

<rules>

Pravidlo 1.1.1

```
<gramrule function="false" message="Attribute for alternative
description of image is missing." typeError="error" tag="img" idRule="1.1.1">
  <rules>
    <singlerule>S-><imgF</singlerule>
    <singlerule>F->+F</singlerule>
    <singlerule>F->alt="A</singlerule>
    <singlerule>A->+A</singlerule>
    <singlerule>A->></singlerule>
  </rules>
</gramrule>
<gramrule function="true" message="Attribute for alternative
description of image is empty." typeError="error" tag="img" idRule="1.1.1">
  <rules>
    <singlerule>S-><imgF</singlerule>
    <singlerule>F->+F</singlerule>
    <singlerule>F->alt=""A</singlerule>
    <singlerule>A->+A</singlerule>
    <singlerule>A->></singlerule>
  </rules>
</gramrule>
<gramrule function="true" message="Alternative text of image is not
supplied correctly. It should describe the content of image." typeError="warning"
tag="img" idRule="1.1.1">
  <rules>
    <singlerule>S-><imgF</singlerule>
    <singlerule>F->+F</singlerule>
    <singlerule>F->alt="obrazek"A</singlerule>
    <singlerule>F->alt="obrazok"A</singlerule>
    <singlerule>F->alt="image"A</singlerule>
    <singlerule>A->+A</singlerule>
    <singlerule>A->></singlerule>
  </rules>
</gramrule>
<gramrule function="false" message="Attribute for alternative
description of area image is missing." typeError="error" tag="area" idRule="1.1.1">
  <rules>
    <singlerule>S-><areaF</singlerule>
    <singlerule>F->+F</singlerule>
    <singlerule>F->alt="A</singlerule>
    <singlerule>A->+A</singlerule>
    <singlerule>A->></singlerule>
  </rules>
</gramrule>
```

```

    <gramrule function="true" message="Attribute for alternative
description of area image is empty." typeError="error" tag="area" idRule="1.1.1">
    <rules>
        <singlerule>S-><areaF</singlerule>
        <singlerule>F->+F</singlerule>
        <singlerule>F->alt=""A</singlerule>
        <singlerule>A->+A</singlerule>
        <singlerule>A->></singlerule>
    </rules>
</gramrule>
<gramrule function="false" message="Attribute for alternative
description of applet is missing." typeError="error" tag="applet" idRule="1.1.1">
    <rules>
        <singlerule>S-><appletF</singlerule>
        <singlerule>F->+F</singlerule>
        <singlerule>F->alt="A</singlerule>
        <singlerule>A->+A</singlerule>
        <singlerule>A->></singlerule>
    </rules>
</gramrule>
<gramrule function="true" message="Attribute for alternative
description of applet is empty." typeError="error" tag="applet" idRule="1.1.1">
    <rules>
        <singlerule>S-><appletF</singlerule>
        <singlerule>F->+F</singlerule>
        <singlerule>F->alt=""A</singlerule>
        <singlerule>A->+A</singlerule>
        <singlerule>A->></singlerule>
    </rules>
</gramrule>
<gramrule function="true" message="Tag blink is not permitted to use."
typeError="error" tag="blink" idRule="1.1.1">
    <rules>
        <singlerule>S-><blinkF</singlerule>
        <singlerule>F->+F</singlerule>
        <singlerule>F->></singlerule>
    </rules>
</gramrule>

```

Pravidlo 1.2.3

```

<gramrule function="content-true" message="Content of object should not
be empty." typeError="error" tag="object" idRule="1.2.3">
    <rules>
        <singlerule>S-><</singlerule>
    </rules>
</gramrule>

```

Pravidlo 1.3.1

```
<gramrule function="true" message="It is recommended to use CSS form
aligning document." typeError="warning" tag="*" idRule="1.3.1">
  <rules>
    <singlerule>S-><F</singlerule>
    <singlerule>F->+F</singlerule>
    <singlerule>F->align=""A</singlerule>
    <singlerule>A->+A</singlerule>
    <singlerule>A->></singlerule>
  </rules>
</gramrule>
<gramrule function="exists" message="Tag strong should be used to
emphasize text." typeError="notice" tag="b" idRule="1.3.1">
  <rules></rules>
</gramrule>
<gramrule function="exists" message="Tag em should be used to emphasize
text." typeError="notice" tag="i" idRule="1.3.1">
  <rules></rules>
</gramrule>
<gramrule function="false" message="For data tables, you may use
attribute summary to describe the content of table." typeError="notice" tag="table"
idRule="1.3.1">
  <rules>
    <singlerule>S-><tableF</singlerule>
    <singlerule>F->+F</singlerule>
    <singlerule>F->summary="A</singlerule>
    <singlerule>A->+A</singlerule>
    <singlerule>A->></singlerule>
  </rules>
</gramrule>
```

Pravidlo 1.3.1

```
<gramrule function="true" message="Font size should not be set by px,
cm, in, mm, pc or pt. It is recommende to use em instead." typeError="error"
tag="*" idRule="1.4.4">
  <rules>
    <singlerule>S-><F</singlerule>
    <singlerule>F->+F</singlerule>
    <singlerule>F->style="C</singlerule>
    <singlerule>A->="B</singlerule>
    <singlerule>A->="C</singlerule>
    <singlerule>C->font-size:D</singlerule>
    <singlerule>B->:D</singlerule>
    <singlerule>D->pxB</singlerule>
    <singlerule>D->ptB</singlerule>
```



```

    <singlelerule>D->pcB</singlelerule>
    <singlelerule>D->inB</singlelerule>
    <singlelerule>D->cmB</singlelerule>
    <singlelerule>D->mmB</singlelerule>
    <singlelerule>D->+D</singlelerule>
    <singlelerule>B->+B</singlelerule>
    <singlelerule>B->"A</singlelerule>
    <singlelerule>A->+A</singlelerule>
    <singlelerule>A->></singlelerule>
  </rules>
</gramrule>

```

Pravidlo 2.4.1

```

<gramrule function="true" message="Hyperlinks should not be hidden by
CSS, because some devices may show them anyway." typeError="warning" tag="a"
idRule="2.4.1">

```

```

  <rules>
    <singlelerule>S-><aF</singlelerule>
    <singlelerule>F->+F</singlelerule>
    <singlelerule>F->style="C</singlelerule>
    <singlelerule>A->="B</singlelerule>
    <singlelerule>A->="C</singlelerule>
    <singlelerule>C->display: noneB</singlelerule>
    <singlelerule>C->visibility: hiddenB</singlelerule>
    <singlelerule>B->+B</singlelerule>
    <singlelerule>B->"A</singlelerule>
    <singlelerule>A->+A</singlelerule>
    <singlelerule>A->></singlelerule>
  </rules>
</gramrule>

```

Pravidlo 2.4.2

```

<gramrule function="exists" message="Tag title does not exists."
typeError="error" tag="title" idRule="2.4.2">

```

```

  <rules></rules>
</gramrule>

```

```

<gramrule function="content-true" message="Title should not contain
generic names, but real title of website." typeError="error" tag="title"
idRule="2.4.2">

```

```

  <rules>
    <singlelerule>S-><title</singlelerule>
    <singlelerule>S-><unknown title</singlelerule>
    <singlelerule>S-><neznama stranka</singlelerule>
  </rules>
</gramrule>

```

```

    <gramrule function="content-true" message="Title should not be empty."
typeError="error" tag="title" idRule="2.4.2">
    <rules>
        <singlerule>S-><</singlerule>
    </rules>
</gramrule>

```

Pravidlo 2.4.3

```

    <gramrule function="true" message="Note that information on website
should be available in same order as in source code. Check if the block are not
shuffled by CSS positioning." typeError="notice" tag="*" idRule="2.4.3">
    <rules>
        <singlerule>S-><F</singlerule>
        <singlerule>F->+F</singlerule>
        <singlerule>F->style="C</singlerule>
        <singlerule>A->="B</singlerule>
        <singlerule>A->="C</singlerule>
        <singlerule>C->+C</singlerule>
        <singlerule>C->position:B</singlerule>
        <singlerule>B->+B</singlerule>
        <singlerule>B->"A</singlerule>
        <singlerule>A->+A</singlerule>
        <singlerule>A->></singlerule>
    </rules>
</gramrule>

```

Pravidlo 2.4.9

```

    <gramrule function="content-true" message="Hyperlink should not contain
generic name, but real title of website where hyperlink leads." typeError="error"
tag="title" idRule="2.4.9">
    <rules>
        <singlerule>S-><zde</singlerule>
        <singlerule>S-><odkaz</singlerule>
        <singlerule>S-><klikni zde</singlerule>
        <singlerule>S-><tento odkaz</singlerule>
        <singlerule>S-><kliknete zde</singlerule>
        <singlerule>S-><here</singlerule>
        <singlerule>S-><link</singlerule>
        <singlerule>S-><click here</singlerule>
        <singlerule>S-><this link</singlerule>
    </rules>
</gramrule>
    <gramrule function="content-true" message="Hyperlink should not be
empty." typeError="error" tag="title" idRule="2.4.9">
    <rules>
        <singlerule>S-><</singlerule>

```

```

        </rules>
    </gramrule>
    <gramrule function="false" message="Attribute for title of hyperlink is
missing." typeError="notice" tag="a" idRule="2.4.9">
        <rules>
            <singlerule>S-><aF</singlerule>
            <singlerule>F->+F</singlerule>
            <singlerule>F->title="A</singlerule>
            <singlerule>A->+A</singlerule>
            <singlerule>A->></singlerule>
        </rules>
    </gramrule>
    <gramrule function="true" message="Attribute for title of hyperlink is
empty." typeError="notice" tag="a" idRule="2.4.9">
        <rules>
            <singlerule>S-><aF</singlerule>
            <singlerule>F->+F</singlerule>
            <singlerule>F->title=""A</singlerule>
            <singlerule>A->+A</singlerule>
            <singlerule>A->></singlerule>
        </rules>
    </gramrule>
    <gramrule function="true" message="Check if hyperlink to file is
visually different than ordinary hyperlink." typeError="notice" tag="a"
idRule="2.4.9">
        <rules>
            <singlerule>S-><aF</singlerule>
            <singlerule>F->+F</singlerule>
            <singlerule>F->href="B</singlerule>
            <singlerule>A->="B</singlerule>
            <singlerule>B->+B</singlerule>
            <singlerule>B->zip"A</singlerule>
            <singlerule>B->rar"A</singlerule>
            <singlerule>B->tar"A</singlerule>
            <singlerule>B->tgz"A</singlerule>
            <singlerule>B->pdf"A</singlerule>
            <singlerule>B->docx"A</singlerule>
            <singlerule>B->xls"A</singlerule>
            <singlerule>B->xlsx"A</singlerule>
            <singlerule>B->ppt"A</singlerule>
            <singlerule>A->+A</singlerule>
            <singlerule>A->></singlerule>
        </rules>
    </gramrule>

```

Pravidlo 3.1.1

```
<gramrule function="false" message="Attribute for lang in HTML tag is missing." typeError="error" tag="html" idRule="3.1.1">
```

```
<rules>  
  <singlerule>S-><htmlF</singlerule>  
  <singlerule>F->+F</singlerule>  
  <singlerule>F->lang="A"</singlerule>  
  <singlerule>F->xml:lang="A"</singlerule>  
  <singlerule>A->+A</singlerule>  
  <singlerule>A->></singlerule>
```

```
</rules>
```

```
</gramrule>
```

```
<gramrule function="true" message="Attribute for lang in HTML tag is empty." typeError="error" tag="html" idRule="3.1.1">
```

```
<rules>  
  <singlerule>S-><htmlF</singlerule>  
  <singlerule>F->+F</singlerule>  
  <singlerule>F->lang=""A</singlerule>  
  <singlerule>F->xml:lang=""A</singlerule>  
  <singlerule>A->+A</singlerule>  
  <singlerule>A->></singlerule>
```

```
</rules>
```

```
</gramrule>
```

Pravidlo 3.2.5

```
<gramrule function="true" message="Please check if hyperlink opening in new window is visually different from other hyperlinks." typeError="notice" tag="a" idRule="3.2.5">
```

```
<rules>  
  <singlerule>S-><aF</singlerule>  
  <singlerule>F->+F</singlerule>  
  <singlerule>F->target="_new"A</singlerule>  
  <singlerule>F->alt="_blank"A</singlerule>  
  <singlerule>A->+A</singlerule>  
  <singlerule>A->></singlerule>
```

```
</rules>
```

```
</gramrule>
```

Pravidlo 4.1.2

```
<gramrule function="false" message="Attribute for title of frame is missing." typeError="error" tag="frame" idRule="4.1.2">
```

```
<rules>  
  <singlerule>S-><frameF</singlerule>  
  <singlerule>F->+F</singlerule>  
  <singlerule>F->title="A"</singlerule>  
  <singlerule>A->+A</singlerule>  
  <singlerule>A->></singlerule>
```

```

        </rules>
    </gramrule>
    <gramrule function="true" message="Attribute for title of frame is
empty." typeError="error" tag="frame" idRule="4.1.2">
        <rules>
            <singlerule>S-><frameF</singlerule>
            <singlerule>F->+F</singlerule>
            <singlerule>F->title=""A</singlerule>
            <singlerule>A->+A</singlerule>
            <singlerule>A->></singlerule>
        </rules>
    </gramrule>
    <gramrule function="false" message="Attribute for title of frame is
missing." typeError="error" tag="iframe" idRule="4.1.2">
        <rules>
            <singlerule>S-><iframeF</singlerule>
            <singlerule>F->+F</singlerule>
            <singlerule>F->title="A</singlerule>
            <singlerule>A->+A</singlerule>
            <singlerule>A->></singlerule>
        </rules>
    </gramrule>
    <gramrule function="true" message="Attribute for title of frame is
empty." typeError="error" tag="iframe" idRule="4.1.2">
        <rules>
            <singlerule>S-><iframeF</singlerule>
            <singlerule>F->+F</singlerule>
            <singlerule>F->title=""A</singlerule>
            <singlerule>A->+A</singlerule>
            <singlerule>A->></singlerule>
        </rules>
    </gramrule>
</rules>
</normrule>
</normrules>

```