



ZÁPADOČESKÁ
UNIVERZITA
V PLZNI

Fakulta aplikovaných věd
Katedra matematiky

Online vyhledávač optimální cesty městem
Diplomová práce

2005

Ladislav Červinka

Prohlášení

Předkládám tímto k posouzení a následné obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem zadanou diplomovou práci zpracoval zcela samostatně, pouze s použitím literatury a pramenů, jejichž úplný seznam je její součástí, a za odborného vedení vedoucího diplomové práce, Mgr. et Ing. Otakara Čerby.

V Plzni dne 27. května 2005

podpis diplomanta

Obsah

Seznam zkratk	4
1 Úvod	5
2 Analýza problematiky vyhledávání optimální cesty	7
3 Softwarové prostředky pro realizaci vyhledávače	10
4 Digitální vektorový model	13
5 Realizace digitálního vektorového modelu sítě komunikací	19
6 Dijkstrův algoritmus	27
7 Transformace ze souřadnicového systému SJTSK do WGS 84	31
8 Převod dat do databázové struktury	34
9 Realizace vyhledávače	43
10 Uživatelská příručka	48
11 Závěr	55
Přílohy	56
Seznam použitých pramenů	57

Seznam zkratk

B – byte

CSV – Comma Separated Values (souborový formát)

ČÚZK – Český úřad zeměměřický a katastrální

ER – Entities Attributes

ERA – Entities Relations Attributes

EXE – Executable File (souborový formát)

GIS – Geographic Information System

GPS – Global Positioning System

HTML – Hypertext Markup Language

HTTP – Hypertext Transfer Protocol

PHP – Personal Home Page Tools

px – Picture Element, obrazový element

SJTSK – souřadnicový systém jednotné trigonometrické sítě katastrální

SMO – státní mapa odvozená

SQL – Structured Query Language

TB – trigonometrický bod

VTX – výkres textově (souborový formát)

VYK – výkres (souborový formát)

WGS 84 – World Geodetic System 1984

WML – Wireless Markup Language

WSP – Wireless Session Protocol

ZABAGED – Základní báze geografických dat

ZMČR10 – základní mapa České republiky 1 : 10000

1 Úvod

Když jsem začal v roce 1991 studovat na nižším gymnáziu v Nymburce, stali se mými spolužáky Michal a Šárka, dvě děti, které bydlely na nymburském Zálabí. Shodou okolností oba bydleli ve stejném bloku bytových domů. Vzhledem k tomu, že budova gymnázia se nachází v podstatě na opačném konci města, měli to tihle dva ze všech studentů bydlících v Nymburce do školy nejdál.

Důvodem, proč se o tom zmiňuji, je skutečnost, že mě později překvapilo, když jsem zjistil, že každý z nich chodí do školy jinou cestou. Část cesty přes Zálabí k lávce přes Labe byla u obou stejná, ovšem potom se jejich každodenní trasa, kterou každý z nich nezávisle překonával, zcela rozcházela.

Michal pravidelně za mostem zabočoval doprava na náměstí Přemyslovců, Boleslavskou a Dlouhou ulicí šel k Vodárně, přešel Valy, prošel parkem Hrdinů a Smetanovou a Resslerou ulicí došel až ke gymnáziu.

Šárka směřovala z lávky přes Labe pravidelně do Eliščiny ulice, pak zabočila do Boleslavské ulice a za mostem přes Velké Valy odbočila do parku Svobody, kterým se dostala do Havlíčkovy ulice a dále do ulice Komenského, kterou se došla ke gymnáziu.

Vzhledem k tomu, že značná část jejich cesty vedla přes historické jádro města se složitou uliční sítí, vycházející z původní gotické urbanizace, bylo velmi těžké odhadnout, která ze dvou tras je kratší.

Již tenkrát jsem přemýšlel, jakým způsobem může vlastně člověk ověřit, zda je určitá cesta městem mezi dvěma body opravdu nejkratší. Svým způsobem mě tento problém fascinoval.

Prvním způsobem řešení, který mě napadl, bylo změřit čas nutný k překonání trasy a porovnat jej s alternativními cestami. O něco později jsem se v rámci zeměpisu seznámil se základními kartometrickými technikami pro určování délky cest z mapy a mohl jsem si vyzkoušet práci s křivkoměrem.

Tyto postupy byly jedněmi z mála existujících metod, které dříve obecně používaly. Určení optimální cesty bylo založeno na odhadnutí vhodných kandidátů na optimální cestu pomocí mapy nebo pomocí osobních znalostí terénu a následném vyhodnocování a porovnávání jejich délek. Je zřejmé, že tato metoda není univerzální, neboť s rostoucí vzájemnou vzdáleností cílového a výchozího, narůstá počet cest, které máme na výběr, a narůstá tak i pravděpodobnost, že vhodného kandidáta na optimální cestu zcela opomeneme.

Uliční síť center velkých měst má navíc často svůj základ v původní gotické urbanizaci, která bývá velmi spleťitá a nepřehledná. Typickým příkladem složité uliční sítě je zástavba Nového Města Pražského. V takových lokalitách musíme počítat rovněž s tím, že mapy jsou zde ve zvýšené míře zatíženy generalizací a kresbou přes míru, přičemž oba tyto kartografické fenomény zkreslují skutečné délkové poměry zobrazovaného terénu.

Je zřejmé, že s problémem vyhledávání optimální cesty v městských aglomeracích se pravidelně setkává značná část populace. Přitom současný bouřlivý rozvoj informační společnosti, jehož jsme svědky, již poskytuje dostatečné prostředky pro nový přístup k tomuto problému a nabízí řešení, která mohou daný problém zvládnout lépe a přitom velmi rychle.

Jako vhodné řešení se zde jasně nabízí GIS založený na topologickém vektorovém modelu komunikační sítě města, který by byl vytvořen s odpovídající přesností a který by tvořil sofistikovaný základ pro implementaci algoritmů vyhledávajících optimální cestu. Jako vhodný prostředek komunikace uživatelů s GIS se v současnosti jeví Internet a mobilní telefon. Pomocí těchto prostředků lze poskytnout možnost vyhledávání optimální cesty takřka neomezenému počtu uživatelů a přitom zachovat možnost snadné aktualizace systému.

Vzhledem ke skutečnosti, že má řešení daného problému potenciálně široké použití a vzhledem k tomu, že v České republice dosud neexistuje žádné obdobné řešení daného

problému, když pomineme jistou analogii s bouřlivě se rozvíjejícím odvětvím navigace pomocí profesionálních produktů s podporou GPS pro potřeby motoristů a turistů, přijal jsem rád možnost zpracovat daný problém jako svoji diplomovou práci, kterou završuji své studium geomatiky na fakultě aplikovaných věd Západočeské univerzity v Plzni.

Náplní této diplomové práce je návrh a realizace online vyhledávače optimální cesty městem. Realizace přitom zahrnuje vytvoření optimálního digitálního modelu komunikační sítě a naprogramování vlastního vyhledávače včetně uživatelského rozhraní.

Pro vybudování digitálního modelu jsem si vybral středočeské město Nymburk. Jedná se o bývalé okresní město s 15 tisíci obyvateli a vybral jsem si jej především proto, že v něm žiji a díky tomu jej velmi dobře znám, což je v mnoha ohledech obrovskou výhodou. Chtěl jsem rovněž svou práci, jejíž těžiště spočívá do značné míry v její praktické části, přispět ke zkvalitnění života občanů ve svém rodném městě. V neposlední řadě mi tato volba přinesla výhodnější pozici při zápůjčce rozsáhlého množství cenných dat, které byly nutnými vstupními podklady, bez kterých bych tuto diplomovou práci nebyl schopen vytvořit.

Proto bych chtěl na tomto místě poděkovat úředníkům zastupujícím město Nymburk, kteří mi poskytli data pro moji práci. Dále bych chtěl poděkovat zejména Mgr. et Ing. Otakaru Čerbovi, který vedl moji diplomovou práci, a všem členům akademické obce Západočeské univerzity v Plzni, které jsem v průběhu studia poznal.

2 Analýza problematiky vyhledávání optimální cesty

Náplní této diplomové práce je teoretické zpracování systému vyhledávání optimální cesty městem pro chodce a cyklisty s využitím mobilních technologií a praktická aplikace tohoto návrhu.

Při formulování problematiky pohybu městem jsem byl postaven před zásadní otázku, co je vlastně optimální cesta. Odpověď není jednoduchá. Nabízí se odpověď že optimální cesta je cesta nejkratší, protože se po ní dostaneme do cíle nejrychleji a tím je tato cesta i nejméně namáhavá. To však nemusí být vždy pravda.

Jako příklad bych na tomto místě vybral dvě rozdílné cesty vedoucí ze západu na naši nejvyšší horu, Sněžku. Jedna je opatřena kvalitním asfaltovým povrchem stoupá k vrcholu spirálovitě s mírným sklonem. Druhá stoupá příkře k vrcholu, její značná část je tvořena masivními balvany a v určitých místech je její sklon natolik prudký, že přechází ve schody. Lidé používají obě z těchto cest. Někteří preferují kratší, ale namáhavou a méně bezpečnou cestu, jiní naopak delší, ale bezpečnější a pohodlnější variantu. Lze říci, že kratší cesta v tomto případě znamená zároveň rychlejší cestu pouze pro fyzicky zdatné jedince.

Podobným způsobem je třeba uvažovat při pohybu ve městě. Pokud se pohybujeme městskou aglomerací na jízdním kole, je třeba uvažovat zpoždění, které získáme při odbočování, jízdě přes křižovatky, dále je třeba počítat s jednosměrnými ulicemi a v neposlední řadě je třeba uvažovat, že některé ulice jsou opatřeny povrchy, které nejsou pro jízdní kolo příliš vhodné. V případě pěšího pohybu je třeba analogicky uvažovat zpoždění získané při přechodu rušné komunikace. V obou případech hrají určitou roli výškové poměry cesty a denní doba, která má přímý vliv na hustotu provozu na komunikacích. Nelze zapomínat ani na handicapované občany pohybující se na invalidním vozíku, v jejich případě je nutné volit trasu tak, aby se vyhýbala schodištím a jiným bariérám.

Pokud se budeme snažit tuto problematiku pojmout skutečně komplexně, pak je třeba též uvažovat různé více či méně subjektivní aspekty, které ovlivní vhodnost určité cesty. Pokud existují dvě stejně vhodné varianty optimální cesty a pokud jedna z těchto cest vede například parkem, zatímco druhá vede po celé své délce podél rušné vozovky, pak bude pro naprostou většinu osob vhodnější první varianta. Naopak se parku vyhne alergik, který ví, že v něm právě kvete rostlina, jejíž pyl je pro něj silným alergenem.

Z nastíněných skutečností je patrné, že optimální cesta je závislá na značném množství skutečností, z nichž každá může hrát v souvislosti s konkrétním člověkem různě důležitou roli. Postižení správných funkčních závislostí je proto velmi obtížné. Při řešení jsem se tedy již od počátku omezil pouze na část základních fenoménů ovlivňujících optimálnost cesty.

Pro chodce jsem se rozhodl určit optimální cestu jako skutečně nejkratší cestu, která existuje. Výše zmíněnou problematiku zpoždění při přecházení rušných vozovek jsem se rozhodl zanedbat. Vycházel jsem přitom ze skutečnosti, že v naprosté většině případů jsou přechody přes rušné komunikace řešeny jako značené přechody pro chodce, takzvané zebry, přičemž platí, že chodec má přednost před vozidly, takže zpoždění by v tomto případě mělo být minimální.

Analogicky jsem postupoval v případě pohybu paraplegiků. V tomto případě jsem pouze vyloučil pohyb po schodech.

V případě cyklistů jsem se rozhodl uvažovat kvalitu povrchu a zpoždění získané jízdou přes frekventované křižovatky s nutností dát přednost v jízdě.

Klíčovým problémem bylo nalézt způsob zadávání výchozí a cílové polohy, mezi kterými má být cesta nalezena. Tento problém je závažný především vzhledem ke skutečnosti, že podstatným způsobem ovlivňuje fungování rozhraní, přes které budou k vyhledávacímu přístupu potenciální uživatelé. Přitom základní podmínkou všech softwarových řešení by měla být maximální uživatelská přívětivost.

V existujících navigačních systémech se lze setkat asi se třemi základními přístupy, které je potenciálně možné aplikovat i v daném případě. Polohu lze zadávat jednak pomocí názvů ulic, pomocí zeměpisných souřadnic nebo přímo pomocí adres budov. Všechny tři možnosti lze vzájemně kombinovat.

Vyhledávání optimální cesty založené na zadávání výchozích a cílových bodů pomocí zeměpisných souřadnic je aplikováno především v profesionálních přístrojích pro navigaci pomocí systému GPS, které jsou uživateli k dispozici po celou dobu cesty a průběžně obnovují výchozí pozici podle aktuální polohy. Díky tomu dokáží navigační systémy pružně reagovat na výchyly od naplánované trasy sestavením nové optimální cesty.

Profesionální přístroje pro GPS navigaci při zadávání cílových bodů dokáží těžit především z možnosti zadání cílového bodu přímo do elektronicky zobrazované mapy. Realizovat analogickou funkci ve vyhledávání optimální cesty pomocí mobilních telefonů prostřednictvím WAP je vzhledem k extrémně omezeným možnostem zobrazování grafiky nemožné. Aplikace zadávání polohy výchozího a cílového bodu pomocí zeměpisných souřadnic by byla možná tedy pouze ručně s pomocí dostatečně podrobné mapy daného města s příslušnými rámovými značkami souřadnicového systému WGS84. Přitom by bylo nutné přesné zjištění polohy z mapy, které klade nemalé nároky na schopnosti uživatelů.

Určování polohy pomocí názvů ulic je velice vágní. Bulváry velkých měst dosahují často délky v řádech kilometrů a je jasné, že optimální cesta se může podstatně lišit podle toho, ve které části ulice se výchozí či cílový bod nachází.

Možným zpřesněním by v tomto ohledu mohlo být zadávání výchozích a cílových bodů pomocí křižovatek dvou ulic. V tomto případě je lokalizace již velmi přesná a do značné míry vychází z tradičního způsobu orientace ve městě pomocí mapy. Zásadním limitujícím faktorem je zde situace, kdy výchozím bodem cesty je dům stojící přibližně uprostřed uličního bloku. V takovém případě jsou dva potenciální výchozí body, oba by bylo nutné uvažovat a tedy při vyhledávání uživatelsky zadat. Přitom je nutné respektovat ztíženou možnost zadávání písmenných znaků v mobilních zařízeních, která jsou často vybavena pouze numerickou klávesnicí. Zároveň je třeba počítat se skutečností, že uživatel nemusí znát název ulic, se kterými se výchozí ulice kříží. Pak by byl, tak jako v případě zadávání polohy pomocí zeměpisných souřadnic, odkázán na pomoc klasické mapy.

Jako nejvýhodnější řešení se tedy jeví zadávání polohy pomocí adres. Každá adresa je charakterizována jednoznačnou kombinací názvu ulice a čísla popisného nebo kombinací názvu ulice a čísla orientačního. V případě, že se město nedělí na městské části, je jednoznačným identifikátorem budovy samotné číslo popisné. Číslem orientačním nejsou obvykle vybaveny všechny budovy. V ulicích, kde je orientační číslování budov, je tak usnadněna usnadňují vyhledání konkrétní budovy. Toto číslování je provedeno většinou tak, že při pohybu ulic v jednom směru hodnoty orientačních čísel rostou nebo klesají, přičemž podél jedné strany ulice jsou čísla lichá a podél druhé strany čísla sudá. Orientační číslo je vždy pro danou ulici jednoznačné, takže ve spojení s názvem ulice jej lze použít jako jednoznačný identifikátor stejně jako číslo popisné v kombinaci s názvem ulice. Oba způsoby udávání adresy jsou běžně používány pro identifikaci budov, takže je vhodné umožnit potenciálním uživatelům vyhledávání oběma způsoby.

Zadávání výchozích a cílových bodů pomocí adres přináší tedy výhody v nezávislosti na mapě a zároveň se jedná o jednoznačný a jednoduchý způsob.

V neposlední řadě bylo třeba též rozhodnout o způsobu modelování pohybu po městě, tedy systému, jakým se budou v počítači napodobeny možné trajektorie pohybu sledovaných objektů.

Na tuto problematiku lze nahlížet dvěma způsoby. Pohyb danou ulicí nebo obdobnou komunikací lze zjednodušeně modelovat pohybem po ose komunikace, tedy jakési myšlené čáře, kterou si lze představit jako čáru oddělující na klasické ulici jízdní pruhy.

Daný problém lze ale vyřešit s vyšší přesností, a sice že model se více přiblíží realitě tak, že osa pohybu se bude co nejvíce přibližovat pravděpodobné trajektorii sledovaného objektu.

První způsob lze s výhodou použít při modelování pohybu automobilů, přičemž lze těžit s nižších nároků na objem dat popisujících takový model a tedy i na výpočetní zdroje nutné k výpočtům nad takovým modelem. Zároveň zkrácení trajektorie není v tomto případě velké, neboť automobily se většinou pohybují podél osy vozovky.

V našem případě je ovšem řešen pohyb chodců a cyklistů, takže se nabízí možnost modelovat trajektorii cyklisty pohybujícího se ulicí podél pravé krajnice ve směru jízdy pro každý směr zvlášť. Vhodný je tento způsob též pro pohyb chodců, neboť většina typických městských ulic má chodníky podél obou krajnic vozovky. Tímto způsobem lze tedy dosáhnout věrnějšího modelu pravděpodobné trajektorie pohybu. To jsou důvody, proč jsem se rozhodl použít právě tento druh modelu.

3 Softwarové prostředky pro realizaci vyhledávače

Jeden ze základních požadavků na vyhledávač byl dán již přívlastkem online v samotném názvu této diplomové práce. Je to v podstatě požadavek zpracovat jej tak, aby byl přístupný přes síť Internet.

Jedná se progresivní přístup k předávání informací, který se s postupujícím rozvojem Internetu stále častěji uplatňuje ve všech odvětvích informatiky. Online systémy se využívají zejména v oblastech, kde se výchozí informace velmi rychle mění nebo kde je objem dat tak rozsáhlý, že není hospodárné jimi zatěžovat paměťová média v počítačích uživatelů.

Jádrem online systému je centrální počítač nazývaný server. Server může být pro zvýšení výkonnosti systému nahrazen celou soustavou počítačů nazývanou cluster. Server přitom obsahuje určitá data, ke kterým poskytuje přístup. Uživatelé mohou užívat buď přímo informace uložené na serveru nebo deriváty z nich odvozené, které jsou serverem vytvořeny na základě konkrétních požadavků. Komunikace uživatelských počítačů se serverem probíhá nejčastěji prostřednictvím sítě Internet.

Výhodami služeb poskytovaných online je především snadná aktualizace informací, které jsou uloženy na jednom centrálním počítači a možnost vybudovat systém vysoce odolný proti odcizení dat. Tento systém též umožňuje soustředit náročné výpočty tam, kde je pro ně k dispozici dostatečný výpočetní výkon, tedy na serveru a vzdálené počítače pak pouze slouží ke zobrazení výsledků uživateli.

Aby byl vyhledávač optimální cestou městem snadno použitelný pro širokou veřejnost, je třeba jej vybudovat tak, aby byl přístupný pomocí běžných softwarových prostředků. V daném případě nic nebrání tomu, aby byl přístupný pomocí klasických internetových vyhledávačů jako je masově rozšířený Internet Explorer společnosti Microsoft nebo Mozilla Firefox.

Zároveň byl dán požadavek zpracovat jeho přístupnost též prostřednictvím technologie WAP pomocí mobilních zařízení pracujících na bázi telekomunikační sítě GSM.

WAP je technologie vyvinutá konsorciem společností Nokia, Ericsson, Motorola a Unwired Planet, kterou lze zjednodušeně popsat jako Internet pro mobilní telefony. WAP realizuje pomocí přenosového protokolu WSP možnost permanentního přístupu pomocí mobilních zařízení ke stránkám v souborovém formátu WML. Tyto stránky mohou obsahovat nejrůznější textové a grafické informace.

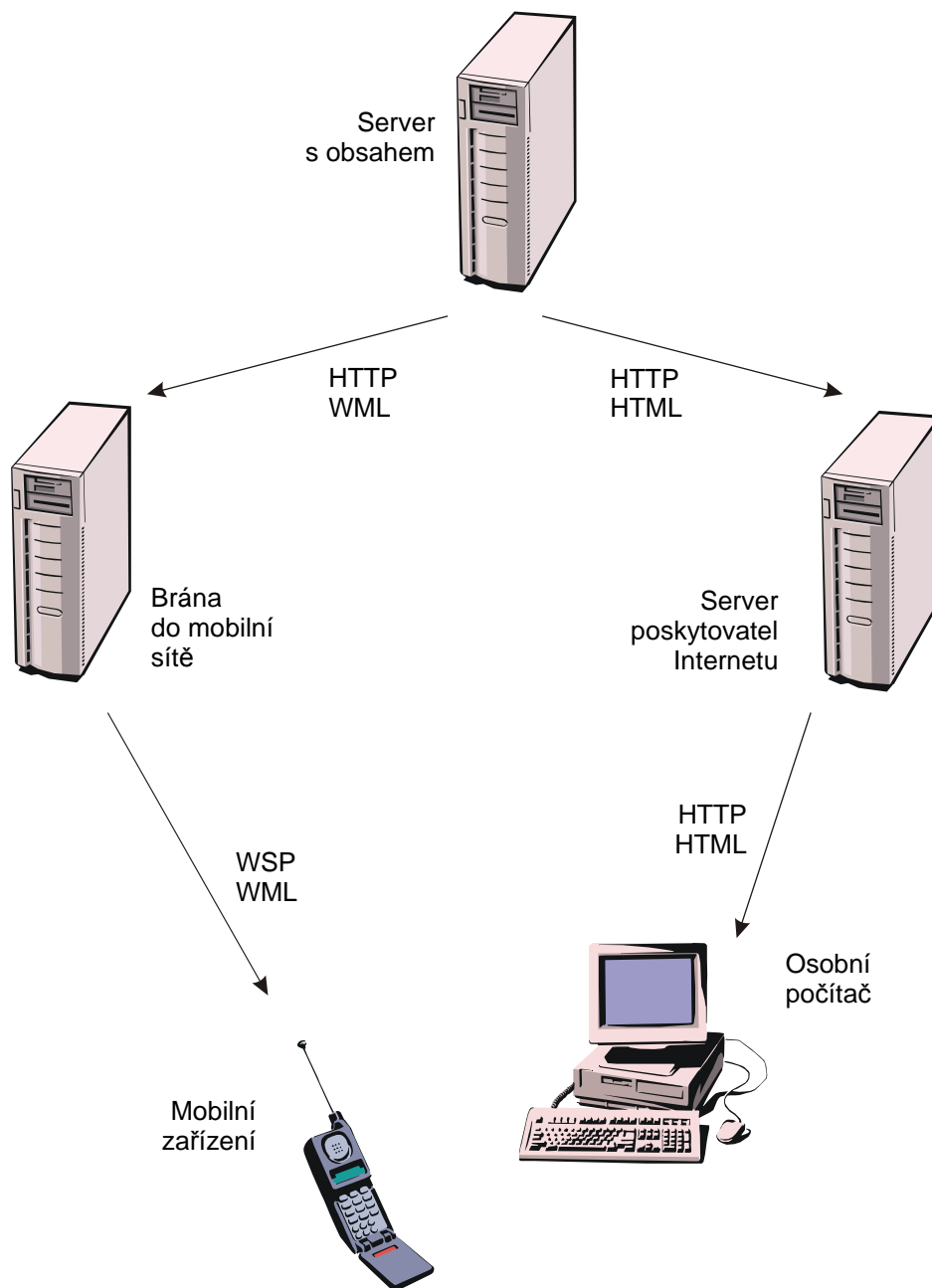
WML stránky jsou přitom stejně jako HTML stránky uloženy na vzdáleném serveru, odkud se přenášejí do brány mobilního operátora protokolem HTTP. Protokol WSP se používá pouze pro komunikaci mobilního telefonu s bránou. Přenos dat je znázorněn na obrázku 3.1.

Technologie WAP je přitom vyvíjena s ohledem na možnosti omezené výpočetní a zobrazovací možnosti mobilních zařízení. Souborový formát WML je odvozen z obdobného formátu HTML, který je používán pro přenos informací v síti Internet. Tak jako se v počítačích připojených k síti Internet používají pro zobrazování stránek prohlížeče, jakými jsou již zmiňovaný Internet Explorer nebo Mozilla Firefox, tak i v mobilních telefonech je nutné používat prohlížeče WML stránek. Většinou se jedná o takzvané mikroprohlížeče, které jsou přímo součástí operačního systému mobilního zařízení.

V současnosti je již drtivá většina běžně používaných mobilních telefonů technologií WAP vybavena. Přestože zažíváme nástup mobilních telefonů vybavených barevnými displeji, které jsou zároveň větší a mají též stále lepší rozlišovací schopnost, je třeba počítat se skutečností, že většina populace využívající mobilní telefony je v současnosti stále vybavena klasickými telefony s monochromatickým displejem, který byl konstruován výhradně pro zobrazování textových informací.

Ačkoliv WAP umožňuje do stránek zobrazovaných v mobilních telefonech vkládat obrázky, je tato funkce u starších telefonů omezena pouze na monochromatické obrázky, tedy obrázky skládající se z elementárních plošek nesoucích buď černou nebo bílou barvu, přičemž maximální šířka obrázku by podle [12] z důvodu kompatibility neměla přesáhnout 96 px a výška by neměla přesáhnout 50 px. Rozumné grafické zobrazení optimální cesty na takto malý formát ve formě srozumitelné mapy je prakticky nerealizovatelné.

Jako jediný schůdný způsob předávání informací o optimální cestě prostřednictvím mobilních telefonů je tedy popis cesty formou ryze textové informace. Výhodou tohoto přístupu je snadná implementovatelnost a nižší nároky na výkon zařízení jak na straně serveru, tak na straně uživatele a zároveň též menší datový tok mezi uživatelem a serverem.



Obrázek 3.1

Jak HTML, tak WML jsou statické formáty, to znamená, že obsah souboru se po přenesení do prohlížeče již nemůže po obsahové stránce měnit. Implementace vyhledávače

však vyžaduje možnost dynamického generování obsahu stránek. Tuto možnost přinášejí do HTML skriptovací jazyky, mezi které patří například ASP, JavaSkript, Perl nebo PHP. Tyto jazyky umožňují do HTML stránek vložit speciální programový kód, který umožňuje dynamicky reagovat na požadavky uživatele.

Pro vytvoření vyhledávače jsem se rozhodl použít skriptovací jazyk PHP. Jedná se o čistě serverový programovací jazyk. To znamená, že všechny příkazy dané skriptem jsou vykonány na straně serveru, který prohlížeči odešle pouze hotový soubor obsahující stránku s HTML obsahem. Soubory dynamicky generované pomocí PHP se odlišují od klasických HTML souborů pouze příponou, která může být podle [7] PHP, PHP3, PHP4 nebo PHTML.

Důvodem pro použití jazyka PHP byla především skutečnost, že se jedná o jeden z nejrozšířenějších skriptovacích jazyků, jedná se o jazyk snadno dostupný tím, že je volně šiřitelný. Skriptovací jazyk PHP lze přitom použít jak pro dynamické generování stránek ve formátu HTML, tak ve formátu WML. Může tedy být zachována určitá dualita mezi vyhledávačem určeným pro prostředí Internetu a pro WAP. Oba vyhledávače mohou pracovat na stejném serveru a nad stejným databázovým systémem

V současnosti je aktuální již pátá generace skriptovacího jazyka PHP, která nově přináší možnosti objektivě orientovaných skriptů. PHP skripty na dynamicky tvořených stránkách vyhledávače byl vytvořeny pro PHP čtvrté generace.

Skriptovací jazyky se většinou používají společně se systémy řízení báze dat, které realizují rychlé a spolehlivé ukládání a manipulaci s daty. V případě PHP je často využíván systém MySQL. Jeho komerční využití zakládá povinnost zakoupení příslušné licence [11], ale pro vývoj a nekomerční použití je celý tento systém k dispozici zdarma. Vzhledem ke skutečnosti, že se jedná o jeden z nejlepších a zároveň nejpopulárnějších systémů řízení báze dat, rozhodl jsem se jej využít i pro vyhledávač.

Konkurencí PHP je skriptovací jazyk ASP vyvíjený společností Microsoft, který se v internetových aplikacích používá často v kombinaci se systémem řízení báze dat SQL Server stejnojmenné společnosti.

Jak PHP, tak MySQL jsou multiplatformní softwarové produkty, které lze provozovat i na populárním operačním systému Linux. Ve spojení s kvalitním serverovým systémem jakým je například Apache lze, oproti řešení založeném na softwaru společnosti Microsoft, získat s minimálními finančními náklady celý softwarový balík potřebný k provozování internetového serveru.

Vyhledávání optimální cesty a navigace se v současnosti skloňuje společně s fenoménem GPS, což je univerzální systém pro určování polohy pomocí specifického signálu vysílaného soustavou družic na oběžných drahách Země.

V současnosti se stále častěji objevují mezi mobilními telefony přístroje integrující v sobě přijímač signálu GPS. Takovým mobilním telefonem je například Nokia 5140 GPS Xpress-On. Tento telefon mimo jiné umožňuje navigaci po trase zadané pomocí významných bodů.

V souvislosti s tím jsem se rozhodl do vyhledávače implementovat možnost výpisu významných bodů trasy v souřadnicích WGS 84 a demonstrovat tak připravenost vyhledávače na příchod nových technologií integrujících GPS do mobilních telefonů.

4 Digitální vektorový model

Základním kamenem pro aplikaci vyhledávající optimální cesty je digitální model sítě komunikací. Je to specificky uspořádaný systém dat, který popisuje městské komunikace, jako jsou chodníky, silnice, stezky, cesty, průchody a podobně. Jeho kvalita zásadním způsobem ovlivňuje kvalitu informací, které lze následně na základě modelu vyhodnocovat.

Digitální modely používané v oblasti GIS se skládají z geometrické a z negeometrické složky. Negeometrická složka je též nazývána složkou atributovou nebo popisnou.

Geometrická data určují tvarové a polohové vlastnosti modelovaných objektů. Jednotlivé objekty se podle zažitého přístupu modelují body, vektory, řetězci vektorů a uzavřenými polygony.

Jako body jsou modelovány objekty, u kterých lze v rámci požadavků na funkčnost GIS zanedbat rozměry objektu. Takovým objektem může být třeba kmen stromu. Bod je charakterizován jednoznačnou polohou v daném souřadnicovém systému.

Vektory modelují liniové objekty, charakterizované přímkou spojnicí počátečního a koncového bodu. Příkladem jsou například osy drátů nadzemního elektrického vedení mezi jednotlivými sloupy.

Řetězec vektorů je posloupnost vektorů, která má počáteční a koncový bod a body vložené. Modelují se jím obecné liniové objekty, jako jsou například osy komunikací. Používají se tam, kde je třeba vystihnout tvar liniového objektu.

Uzavřené polygony jsou vlastně řetězce vektorů, kde platí, že počáteční bod je zároveň bodem koncovým. Takové polygony vymezují plochu a mohou tak modelovat například pozemkové parcely. Častěji jsou však plochy uzavřeny větším počtem vektorů nebo vektorových řetězců.

Některé pokročilé datové modely používají také trojrozměrné geometrické modely objektů, v naprosté většině případů aplikace GIS však lze vystačit s dvourozměrnými modely. Specifickými geometrickými daty jsou též buňky rastrových snímků.

Negeometrická data slouží k připojení popisných informací o objektech. Popisnou informací je například parcelní číslo nebo druh kultury na pozemkové parcele. Jako popisná složka se také většinou realizují výšky objektů.

Digitální model potřebný pro vyhledávání optimální cesty musí obsahovat modely cest, po kterých se lze po městě pohybovat. Cesty jsou typickým lineárním objektem, u kterého převládá jeden rozměr a tedy je vhodné je modelovat ve formě vzájemně navazujících vektorových řetězců.

Pomocí atributových dat pak můžeme doplnit informace sdělující, zda se lze pohybovat po cestách modelovaných vektorovými řetězci oběma směry, nebo pouze jedním směrem, jaká je délka cesty a podobně. Takový model se v teorii geografických informačních systémů nazývá vektorový digitální model.

Digitální vektorové modely se dělí podle [5] na topologické a na takzvané špagetové.

Špagetové vektorové modely jsou charakteristické jednoduchou strukturou ukládání objektů, která je tvořena prostými seznamy bodů tvořících jednotlivé objekty. Takový vektorový model není vhodný pro provádění analýz, jakou je i vyhledávání optimální cesty, neboť nezahrnuje informace o návaznosti jednotlivých vektorů. Špagetový vektorový model se používá například v přípravě tiskových podkladů kartografických děl.

Topologické vektorové modely jsou obohaceny o vyjádření vzájemných vazeb objektů. Pro vazby se též používá pojem relace.

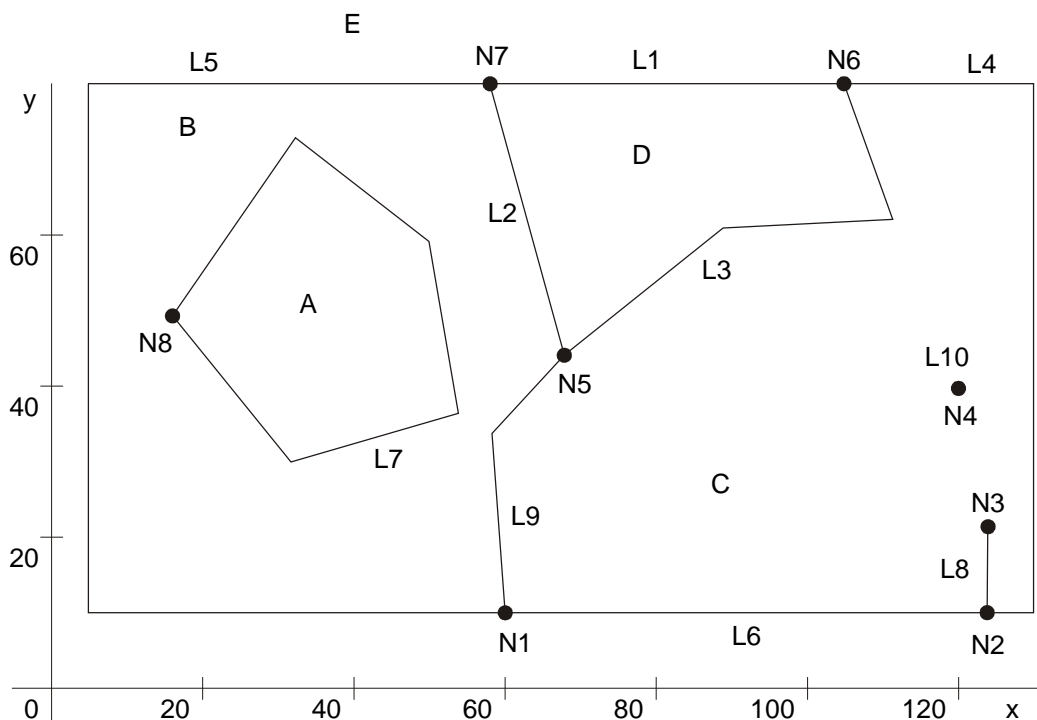
Pro popis topologických modelů se terminologie podle [5] str. 34 rozšiřuje o pojmy uzel, spoj, směrový spoj, směrový řetězec, hraniční oblouk, okruh, oblast.

Uzel je bod, který ve kterém dochází ke styku spojů. Přitom mohou existovat osamělé uzly stejně tak jako uzly, které incidují pouze s jedním spojem.

Spoj je úsečka nebo řetězec úseček, někdy též označovaný jako linie, spojující dva uzly. Relace, kterou realizujeme pomocí spoje může být platná stejně tak oběma směry jako pouze jedním směrem, potom se jedná o směrový spoj. Příkladem může být situace, kdy uzly modelujeme křižovatky komunikační sítě, spoji modelujeme silnice a účelem má být model pohybu automobilu. Pokud se mezi dvěma křižovatkami nachází jednosměrná silnice, je vhodné takovou relaci modelovat směrovým spojením.

Hraniční oblouk nebo hranice je soubor bodů spojený křivkou, která je určena matematickou funkcí, případně je definován řetězcem nebo směrovým řetězcem. Okruh je uzavřená sestava vektorů. Jednoduchá oblast je plocha daná svou vnější hranicí a je bez vnitřních okruhů. Komplexní oblast je dána vnějším okruhem s možností vnitřních okruhů, nazývaných též ostrovy.

Pokud se vektorový model skládá z uzlových bodů, vektorů a ploch uzavřených polygonů, lze podle [5] topologické vztahy obecně realizovat pomocí čtyř tabulek: tabulky polygonů, uzlové tabulky, tabulky souřadnic a tabulky spojů. Příklad vektorového modelu je uveden na obrázku 4.1.



Obrázek 4.1

Tabulka polygonů obsahuje identifikační klíče jednotlivých ploch a pro každou plochu jsou vyjmenovány její hraniční polygony. Příkladu na obrázku odpovídá tabulka 4.1.

polygon	spoj
A	L7
B	L2, L5, L7, L9
C	L3, L4, L6, L9
D	L1, L2, L3
E	L1, L4, L5, L6

Tabulka 4.1

Klíč je přitom specifický kód, který jednoznačně identifikuje daný řádek v tabulce. Klíče mohou být realizovány například očíslováním řádků tabulky.

Tabulka uzlů obsahuje identifikační klíče uzlů a spoje, které vycházejí z daného uzlu. Příkladu z obrázku odpovídá tabulka 4.2.

uzel	spoj
N1	L5, L6, L9
N2	L4, L6, L8
N3	L8
N4	L10
N5	L2, L3, L9
N6	L1, L3, L4
N7	L1, L2, L5
N8	L7

Tabulka 4.2

Tabulka souřadnic obsahuje klíče jednotlivých spojů a jednotlivé body ve formě souřadnic, pomocí kterých jsou dané spoje realizovány. Příkladu z obrázku odpovídá tabulka souřadnic 4.3.

spoj	souřadnice bodu
L1	58,80 105,80
L2	68,44 58,80
L3	68,44 89,61 111,52 105,80
L4	105,80 130,80 130,10 124,10
L5	58,80 5,80 5,10 60,10
L6	60,10 124,10
L7	16,49 32,73 50,60 54,36 32,30 16,49
L8	124,10 124,21
L9	60,10 58,34 68,44
L10	120,40 120,40

Tabulka 4.3

Tabulka spojů obsahuje identifikační klíče jednotlivých spojů, počáteční a koncové body a identifikační klíče. Tabulka 4.4 odpovídá tabulce spojů v uvedeném příkladu.

spoj	poč. uzel	konc. uzel	levý polyg.	pravý polyg.
L1	N7	N6	E	D
L2	N5	N7	B	D
L3	N5	N6	D	C
L4	N6	N2	E	C
L5	N7	N1	B	E
L6	N1	N2	C	E
L7	N8	N8	B	A
L8	N2	N3	C	C
L9	N1	N5	B	C
L10	N4	N4	C	C

Tabulka 4.4

Pro ukládání a práci s rozsáhlými digitálními vektorovými modely se velmi dobře osvědčuje použití systémů řízení báze dat, což je speciální software určený k ukládání a správě rozsáhlých dat.

Data jsou v databázových systémech uspořádána v pomoci datových modelů. Datový model je způsob ukládání informací o objektech reálného světa v digitální podobě. V současnosti se v oblasti GIS uplatňuje relační datový model, nově se používá též objektový přístup.

Hlavním přínosem používání systémů řízení báze dat je snížení nároků na ukládání dat, rychlejší vyhledávání, přehlednost, snadná editace a údržba dat. K úspoře paměti dochází v daném příkladě tím, že názvy ulic

Relační model organizuje data do tabulek. Tabulky se skládají z entit a atributů. Entity jsou přitom jednotlivé položky v tabulce. Jedna entita přitom odpovídá jednomu řádku tabulky. Atributy odpovídají sloupcům tabulky a popisují vlastnosti entit. Jeden atribut nebo skupina atributů musí být v tabulce vždy jednoznačně určovat danou entitu, tak jako v reálném životě například rodné číslo jednoznačně určuje občana České republiky. Takový atribut nebo skupina atributů se nazývá klíč. Tabulka může obsahovat více klíčů, pak se jeden klíč používá jako hlavní a označuje se jako primární klíč.

Například v tabulce 4.4 může být klíčem číslo popisné. Klíčem zde však může být složen z atributů ulice a číslo orientační.

V tabulce 4.5 může být klíčem jak číslo tak název. Obecně je však vhodné klíč volit tak, aby byl co možná nejkratší, neboť se tak urychluje vyhledávání v databázi.

BUDOVY			
ČP	ČO	parcela	ulice
939	33	1013	2
568	6	617	4
560	9	617	4
518	36	555	1
1683	9	2252	3
591	31	630	1

Tabulka 4.4

ULICE	
číslo	název
1	Komenského
2	Maršála Koněva
3	Raisova
4	Zbožská

Tabulka 4.5

Pomocí klíčů se uvádějí entity z různých tabulek do vzájemných vztahů, relací. Entita může být obecně v relaci s libovolným množstvím entit v libovolných tabulkách datového modelu.

V terminologii relačních databází se často používá pojem kardinalita, což je určitý popis relační vazby. Relace může nabývat kardinality 1 : 1, 1 : N nebo M : N. Pokud je kardinalita relace mezi tabulkami 1 : 1, pak může být jedna položka z dané tabulky v relaci vždy jen s jednou položkou v druhé tabulce. Relace může být povinná nebo nepovinná. Pokud je relace například v druhé tabulce nepovinná, označuje se kardinalita 1 : 1(0).

Příkladem relace 1 : N může být vztah tabulky 4.4 a 4.5, kde je v tabulce budovy pomocí klíče ulice realizován vztah budovy k ulici, ve které se nachází. Budova náleží svou adresou vždy do jedné ulice a v jedné ulici je obvykle více budov. Rozdělením informací o adrese do tabulek v tomto případě uspoříme místo v paměti, neboť kdyby byly názvy ulic přímo v tabulce číslo 4.4 došlo by k opakování názvů Komenského a Zbožská.

Rozdělení do tabulek je výhodné také pro aktualizaci dat. V případě, že se změní název ulice, bude nutné přejmenovat pouze jednu položku. V opačném případě by bylo nutné projít celou tabulku budov a změnit původní název za nový právě tolikrát, kolik budov se v ulici nachází.

Rovněž se tak urychluje vyhledávání. Pokud je třeba například vyhledat všechny budovy, které se nacházejí v ulicích, jejichž jméno obsahuje řetězec Ko, vyhledají se klíče všech ulic, které odpovídají požadavku a pak se vyberou z tabulky budov ty položky, které mají jako atribut ulice příslušnou hodnotu klíče.

V opačném případě by bylo nutné procházet opět všechny položky tabulky a u každého názvu ulice zjistit, zda obsahuje požadovaný řetězec.

OSOBY		
rodné číslo	příjmení	jméno
5610061013	Acsay	Emil
6808210999	Svoboda	Josef
8911091097	Haman	Jan
8112130859	Waleski	Jacek

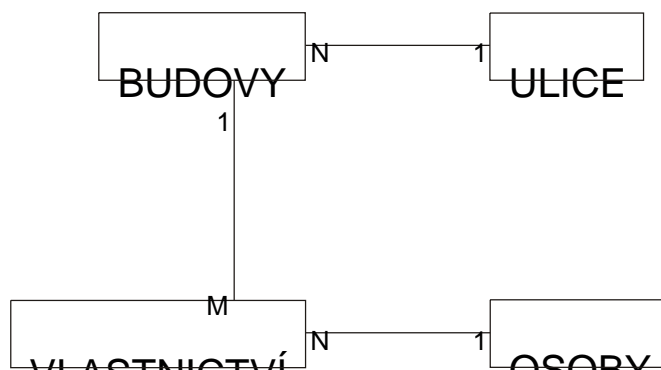
Tabulka 4.6

VLASTNICTVÍ	
rodné číslo	budova
5610061013	939
5610061013	568
6808210999	560
8911091097	939
8911091097	1683
8112130859	591

Tabulka 4.7

Příkladem relace M : N může být vztah vlastnictví k budovám. Budova může být vlastněna více subjekty a každý subjekt může být vlastníkem více budov.

Relace M : N se obvykle implementují pomocí zvláštní tabulky, kde entity odpovídají vždy jedné relaci a atributy jsou klíče příslušných tabulek, které jsou v relaci. Tabulka 4.7 uvádí do relace vlastnictví tabulku osob 4.6 a tabulku budov 4.4.



Obrázek 4.2

Datové modely se zakreslují pomocí specifických schémat, které zachycují třídy entit, což jsou vlastně jednotlivé tabulky, a relace včetně kardinalit, které mezi nimi existují. Tato

schémata se nazývají ER modely. ER model uvedeného příkladu je na obrázku 4.2. Někdy se též ER modely obohacují o atributy, pak se jedná o ERA modely.

Kvalita digitálního modelu je závislá na kvalitě zdrojových dat, která byla pro jeho tvorbu použita. Vektorová data v současnosti vznikají buď na základě přímých měření v terénu nebo se odvozují z existujících mapových děl, výsledků fotogrammetrického snímkování nebo dálkového průzkumu Země.

Tvorba vektorů přitom může probíhat ručně, automaticky nebo poloautomaticky. Poloautomatická tvorba vektorů se používá zejména při digitalizaci analogových map tvořených čarami, kde je vyžadována vysoká přesnost provedení digitalizace a automatické rozpoznávání vektorů je usnadněno kvalitním monochromatickým podkladem. Tato metoda je závislá na dohledu operátora, který kontroluje správný průběh vektorizace prováděné počítačem a přitom řeší nejrůznější singularity, jako je přerušení čáry a podobně. Příkladem rozsáhlého použití poloautomatické tvorby vektorového modelu je například původní vrstevnicová vrstva v ZABAGED, vytvořená na základě čárových předloh ZMČR10.

Plně automatická tvorba vektorových modelů se používá v případech, kdy jsou digitalizované plochy obrazovou funkcí dobře vymezeny. Obdobně jako poloautomatická digitalizace se používá především pro získávání geometrických dat z analogových map.

Ruční tvorba vektorového modelu je z vyjmenovaných metod nejnáročnější. Je náročná časově a klade vysoké nároky na soustředění operátora. Je to však jediná metoda, která dokáže účinně odstranit složité singularity v digitalizovaných datech. Při rozpoznávání objektů vykazujících značnou složitost obrazové funkce je to prakticky jediný použitelný způsob tvorby digitálního vektorového modelu.

Ruční vektorizace se v minulosti často prováděla pomocí speciálních polohovacích zařízení nazývaných tablety nebo pomocí velkoformátových digitalizačních stolů, v současnosti je však upřednostňována ruční kresba na vrstvě rastrového obrazu. Tento přístup bylo možné realizovat až s nástupem výkonných počítačů poskytujících dostatečné prostředky pro ukládání a práci s objemnými rastrovými daty.

5 Realizace digitálního vektorového modelu sítě komunikací

Pro tvorbu digitálního vektorového modelu jsem se rozhodl použít ortofotomapu Nymburka. Výhodou použití ortofotomapy je skutečnost, že se jedná při dostatečně velkém rozlišení o kartografické dílo, které není zatíženo kartografickou generalizací ani kresbou přes míru, takže je věrně zobrazuje realitu. Naopak nevýhodou lze spatřovat v zakrytí některých důležitých objektů korunami stromů, zvláště pokud bylo snímkování provedeno ve vegetačním období. Kvalita ortofotomapy přitom velmi úzce závisí na digitálním modelu terénu, na jehož základě se za pomoci výpočetní techniky odvozuje z digitalizovaných, popřípadě digitálních, fotogrammetrických snímků. V České republice se pro tvorbu ortofotomap široce využívají výškové informace ze ZABAGED.

Ortofotomapy České republiky s rozlišením 0,5 m/px jsou jedním z mapových děl v kladu listů SMO, které jsou v pravidelných tříletých cyklech obnovovány ČÚZK na základě nově provedeného snímkování. Přitom cyklická obnova je podle [16] plánována tak, že je území České republiky rozděleno směrem od východu na západ na tři pásma, a v daném roce probíhá snímkování vždy jedné z těchto částí. Pásma jsou příznačně nazývána Východ, Střed a Západ. V roce 2005 se snímkuje pásmo Západ, v roce 2004 se snímkovalo pásmo Střed a v roce 2003 pásmo Východ. V současnosti probíhá první cyklus barevného snímkování. V minulosti byly ortofotomapy vyhotovovány v šedé škále. Vzorový výřez ortofotomapy je na obrázku číslo 5.1. Výřez je zobrazen v měřítku 1 : 10000.



Obrázek 5.1

Přestože jsou tato data pro zpracování daného modelu postačující, podařilo se mi pro účely této práce sjednat bezplatné zapůjčení ortofotomapy, která je ve vlastnictví města Nymburk a která poskytuje vyšší rozlišení 0,2 m/px. Ukázkový výřez této ortofotomapy je na obrázku 5.2 a je zobrazen stejně jako předchozí výřez v měřítku 1 : 10000

Zdrojová data, která jsem pro svoji diplomovou práci použil, jsou spravována odborem informatiky městského úřadu v Nymburce. Ze souboru dat, která mi úřad poskytl jsem použil především již zmiňovaný soubor ortofotomap vycházejících z kladu SMO v podobě bitmapových souborů datového formátu TIF. Jedná se o 14 mapových polí v

souvislém kladu o rozměrech 6250×5000 pixelů. Jedna ortofotomapa tedy odpovídá území 1250×1000 metrů. Čtyři ortofotomapy potom odpovídají přímo kladu SMO v měřítku 1 : 5000. Od tohoto mapového díla je odvozeno i označování jednotlivých souborů s ortofotomapami, jejichž pojmenování vždy odpovídá formátu NymbXXY, kde NymbXX odpovídá předepsanému označení mapových listů SMO 1 : 5000 ve tvaru NYMBURK X-X. Y je pořadí příslušné čtvrtiny SMO 1 : 5000, která odpovídá dané ortofotomapě. Severozápadní čtvrtina má číslo 1, severovýchodní má číslo 2, jihozápadní má číslo 3 a jihovýchodní má číslo 4.

Ortofotomapy mají barevnou hloubku 24 bitů, jsou nekomprimovány a všechny mají díky tomu shodnou velikost 93 871 498 B. Části, které nezobrazují intravilán města jsou zamaskovány souvislou bílou plochou. Ke každé ortofotomapě náleží též georeferenční soubor typu TFW udávající orientaci mapového listu v souřadnicovém systému SJTSK.



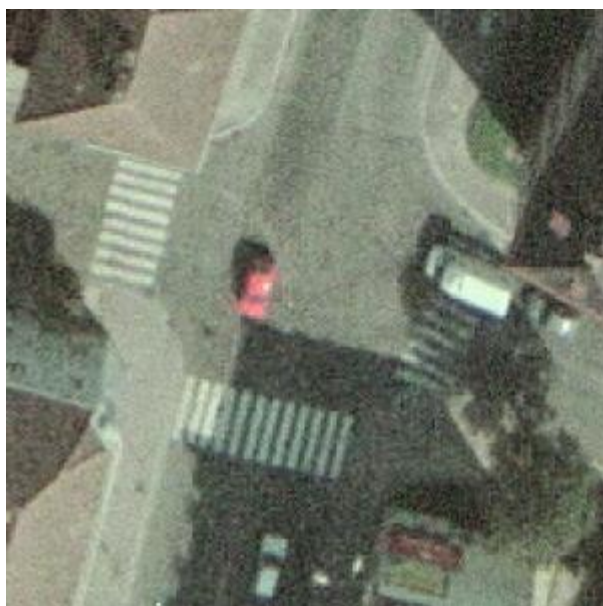
Obrázek 5.2

Při zevrubném srovnání ortofotomapy z majetku města Nymburk s ortofotomapou ČÚZK, která je volně poskytována přes Internet jako vzor [16], jsem ocenil podstatně snadnější identifikaci obrubníků tvořících přechod mezi krajnicí vozovky a chodníkem u ortofotomapy s větším rozlišením. Rovněž bylo možné lépe rozeznat další objekty jako jsou dělící čáry na vozovkách a přechody pro chodce. Kvalitu rozlišení lze porovnat na výřezech ortofotomap v obrázcích 5.3 a 5.4. Na obou je zachycena křižovatka silnic a oba zachycují prostor 40×40 m. Na prvním z dvojice obrázků lze rozpoznat přechod pro chodce, který splývá v jednolitou bílou plochu, kdežto na kvalitnější ortofotomapě jsou patrné jednotlivé pruhy takzvané zebry a obrubníky chodníků.

Společně s ortofotomapou Nymburka se mi podařilo bezplatně získat zapůjčení vektorových dat ve formě ShapeFile, které reprezentují osy ulic a které jsou opatřeny atributovými daty obsahujícími aktuální názvy jednotlivých ulic. Bohužel přesnost zakreslení vektorů a topologická čistota je natolik nízká, že možnost využít tuto vektorovou vrstvu pro vytvoření digitálního modelu je mizivá a spíše by takový pokus vedl k zavlečení obtížně odstranitelných chyb. Původní účel vzniku této vrstvy mi není znám, ale patrně byl vyhotoven pouze pro informaci o názvech ulic v kombinaci s ortofotomapou.



Obrázek 5.3



Obrázek 5.4

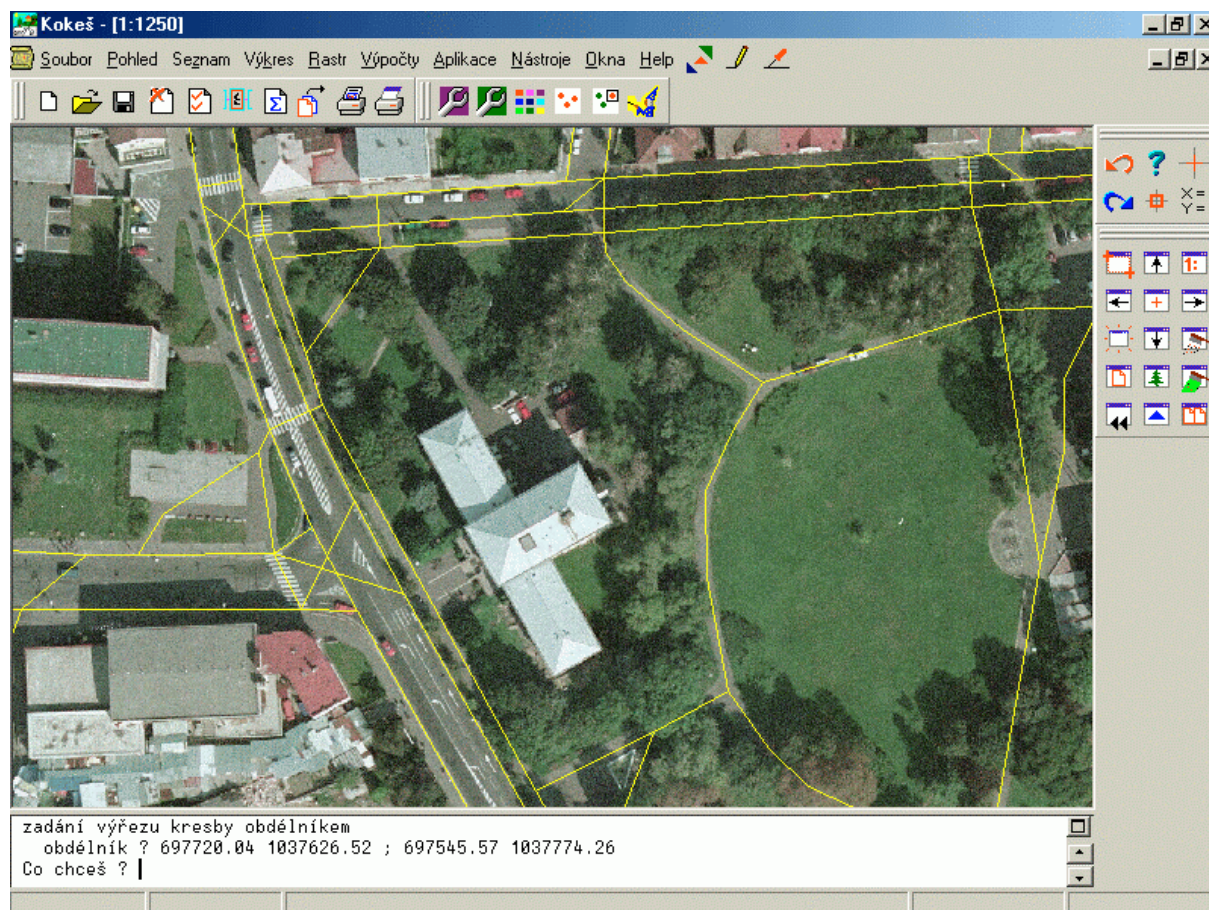
Z tohoto zdroje jsem do své práce tedy převzal pouze názvy ulic, které jsem převedl do textového formátu s kódováním Windows CP 1250. Jedná se o zásadní zdroj atributových dat použitých v modelu.

Digitální model komunikační sítě jsem se rozhodl vytvořit v programu Kokeš 5.62 vyvinutém společností GEPRO. Ačkoliv je tento software pro takový úkol není ideálním nástrojem, byla zásadním momentem, který mě vedl k jeho využití, skutečnost, že mám kopii tohoto programu zapůjčenu společností GEPRO pro studijní účely na svůj osobní počítač. Možnost pracovat na tvorbě modelu mimo univerzitní počítačovou laboratoř byla přitom prioritní, neboť tvorba modelu byla mimořádně časově náročná.

Využití tohoto programu znamenalo především nutnost naprogramovat aplikaci, která převádí specifický výkres vytvořený v programu Kokeš do formy databázových tabulek. Výhodou tohoto způsobu je dokonalá kontrola nad způsobem vzniku digitálního modelu.

Kresba geometrické složky modelu byla provedena ručně nad vrstvou georeferencované ortofotomapy a skládá se z vektorů, respektive z vektorových řetězců, které se stýkají v uzlech. Vektorový řetězec je v rámci programu Kokeš chápán jako samostatný objekt charakterizovaný číslem, které je dáno pořadím, ve kterém vznikl, a je uložen jako posloupnost lomových bodů řetězce.

V části Analýza problematiky vyhledávání optimální cesty byl stručně nastíněn způsob, který jsem zvolil pro tvorbu modelu po stránce geometrických dat. Rozhodl jsem se přiblížit vektorový model co nejvíce reálnému způsobu pohybu cyklistů a chodců. Nevýhodou tohoto řešení je nárůst geometrických dat a tedy i nárůst nároků na analýzy prováděné nad takovým modelem. V případě klasické ulice, kde je podél obousměrně průjezdné vozovky po obou stranách chodník, by bylo nutné trajektorie pohybu chodců a cyklistů modelovat čtyřmi vektory.



Obrázek 5.5

Z tohoto důvodu jsem provedl při určení způsobu digitalizace zjednodušení spočívající ve sjednocení vektoru pro pohyb cyklisty v jednom směru s vektorem modelujícím možnost pohybu po chodníku. Při aplikaci tohoto zjednodušení pak vektor kopíruje rozhraní chodníku a vozovky, které je většinou dáno obrubníkem. Klasická ulice je tak modelována dvěma vektory. Pouze v místech, kde je mezi krajnicí vozovky a chodníkem zatravněný pás jsem volil zvláštní vektor pro chodník. Přesnost modelu lze empiricky charakterizovat maximální odchylkou 3 m, ve vzdálenosti skutečné trajektorie pohybu chodce, respektive cyklisty, a příslušného vektoru v modelu.

Zvlášť jsou modelovány značené přechody pro chodce a zvlášť jsou modelovány též křižovatky. Pokud se kříží dvě silnice, přičemž po každé se lze pohybovat v obou směrech, je

taková křižovatka modelována čtyřmi vektory znázorňujícími jízdu rovně přes křižovatku. Dva vektory modelují odbočení vlevo a na klasické křižovatce tak vytvářejí kříž spojující diagonálně vrcholy pomyslného čtverce tvořeného hranicemi křižovatky. Příklad vektorové kresby nad vrstvou ortofotomapy v prostředí Kokeš je uveden na obrázku 4.5.

Při digitalizaci geometrické složky modelu byla paralelně pořizována též složka popisná. Každý vektor nese celkem 7 popisných atributů. Základním atributem je název vektoru. Tím je většinou úplný název ulice, ve které se příslušný vektor, respektive vektorový řetězec nachází. V případě, že se jedná o vektor modelující přejezd křižovatky, je použit název ulice, kterou vektor překonává. V určitých případech se modeluje cesta, která nekopíruje žádnou ulici, jedná se například o cestu parkem nebo cestu přes náměstí, nepojmenovanou stezku pro chodce nebo cyklisty. V takovém případě je použit název příslušného parku, náměstí, nebo je použito pojmenování, které co nejlépe vystihuje danou cestu. Byl použit například název podél Labe, podél kostela, k zimnímu stadionu a podobně. V případě, že se jedná o cestu, kterou nelze pojmenovat, je použit místo názvu ulice příznak ne. Nepojmenovanými vektory jsou především vektory modelující jízdu přes křižovatku vlevo.

Dále byly pořizeny atributy charakterizující typ komunikace, směr pohybu, druh povrchu komunikace a penále pro pohyb po hraně. Vzhledem ke skutečnosti, že některé hrany modelují jak trajektorii pohybu chodce tak trajektorii pohybu cyklisty, byly první dva jmenované atributy pořizovány jak pro pohyb na jízdním kole, tak pro pohyb pěšky.

Komunikace zahrnuté v modelu jsou charakterizovány celkem třinácti různými typy: chodník, krajnice vozovky, cesta parkem, značený přechod pro chodce a podobně. Při digitalizaci byly používány znakové kódy, které jsou uvedeny v tabulce 5.1. Mezi tyto atributy bylo též nutné zahrnout variantu, kdy vektorový řetězec modeluje pouze komunikaci pro chodce nebo naopak pouze pro cyklisty, takže atribut typu komunikace je vlastně zbytečný. K tomuto účelu slouží atribut s kódovým označením ne.

číslo	kód	typ komunikce
0	ne	nerozlišený typ komunikace
1	kr	krajnice vozovky
2	ps	přechod nebo přejezd kolmo přes vozovku
3	pk	přejezd vlevo přes křižovatku
4	ch	chodník
5	cp	cesta parkem
6	sc	schodiště
7	st	stezka pro chodce nebo cyklisty
8	ze	značený přechod pro chodce nebo cyklisty
9	mo	po mostě
10	po	podchod nebo podjezd
11	pr	průchod nebo průjezd
12	nc	neupravená stezka

Tabulka 5.1

Směr pohybu udává vždy jednu ze čtyř obecně možných variant směru pohybu po modelované komunikaci. Je možné se po komunikaci pohybovat oběma směry, to je typický atribut pro pohyb chodců po chodníku. V případě, že vektorový řetězec modeluje krajnici vozovky pro pohyb cyklistů, je pohyb možný pouze v jednom směru, neboť na vozovce se musí cyklista pohybovat při pravé krajnici ve směru jízdy. Tento směr může být souhlasný nebo nesouhlasný s orientací vektoru. Orientace vektorového řetězce je v programu Kokeš dána implicitně pořadím digitalizace jednotlivých bodů, ale lze ji dodatečně měnit. Čtvrtou variantou je úplný zákaz pohybu po komunikaci modelované daným vektorovým řetězcem.

Typickým příkladem jsou vektory modelující odbočování cyklistů na křižovatkách vlevo, které by v případě chodců odpovídalo diagonálnímu přechodu křižovatky, který je vyloučen. Atributy směru pohybu a příslušného kódování jsou v tabulce 5.2.

číslo	kód	směr
0	n	pohyb zakázán oběma směry
1	a	souhlasný s orientací spojnice
2	b	opačný k orientaci spojnice
3	o	pohyb oběma směry

Tabulka 5.2

Druh povrchu komunikace charakterizuje materiál, jakým je daná komunikace pokryta. Tento atribut platí pouze pro pohyb na jízdním kole. V modelu se nachází celkem sedm druhů povrchu, přičemž nejfrekventovanějším materiálem je asfaltový beton. Je zde vyhrazena též hodnota atributu charakterizující nerozlišený povrch, která je použita v případech, kdy je vektorový řetězec určen pouze pro pohyb chodců. Vzhledem ke skutečnosti, že závislost optimální cesty na druhu povrchu je uvažována pouze pro pohyb na jízdním kole, je informace o povrchu relevantní pouze v případech, kdy vektorový řetězec modeluje pohyb cyklisty. Druhy povrchu a jejich kódování je uvedeno v tabulce 5.3.

číslo	kód	povrch
0	ne	nerozlišený povrch
1	as	asfaltový beton
2	dl	dláždění z kamenných kostek
3	pa	betonové silniční panely
4	za	betonové dlaždice
5	hl	nezvevněný povrch
6	dr	dřevo

Tabulka 5.3

Pro účely zavedení pravděpodobného čekání při přejezdu rušných komunikací jsem se rozhodl rozdělit vektorové řetězce empiricky do čtyř tříd označených čísly od 0 do 3. Tato klasifikace se opět týká pouze pohybu na jízdním kole, neboť zpoždění chodců při přechodu frekventovaných vozovek jsem se rozhodl zanedbat. Vektorové řetězce modelující pohyb podél krajnice nejsou nijak penalizovány a příslušný atribut je tak 0. Obdobně je tomu při přejíždění křižovatky s vedlejší silnicí.

Přejíždění křižovatky s hlavní silnicí rovněž je vždy charakterizováno penalizací s hodnotou od 1 do 3. Penalizace 1 přitom odpovídá přejezdu vozovky s malým dopravním zatížením. Hodnoty 2 a 3 pak odpovídají středně a vysoce frekventovaným vozovkám. Obdobným způsobem je řešeno odbočování na křižovatkách vlevo, příslušné vektory jsou opatřeny stejnými hodnotami atributu penalizace. Nerozlišuje se přitom, zda se jedná o odbočování vlevo z hlavní silnice na silnici vedlejší či naopak, pro oba směry pohybu platí jedna hodnota atributu.

Přiřazení hodnot penalizace jsem prováděl empiricky. Hodnota 3 odpovídá nejfrekventovanější vozovce ve městě, která prochází severojižním směrem ulicemi Boleslavská a Kolínská. Hodnota 2 odpovídá ostatním vozovkám tvořícím hlavní tepny provozu ve městě. Jedná se o ulice Poděbradská, Zbožská, V Kolonii a další. Vektory překračující ostatní ulice jsou klasifikovány hodnotou 1.

Aby bylo možné zavést do modelu odhadované zpoždění získané při přejezdu frekventovaných silnic, bylo třeba provést studii reálného provozu ve městě. Problematika

modelování dopravy ve městě je nesmírně obsáhlou vědní disciplínou, která překračuje rámec této diplomové práce, takže jsem se omezil na elementární způsob zavedení těchto vlivů do určování optimální cesty.

Provedl jsem měření průměrného času čekání na křižovatce. Studii jsem provedl na vzorku deseti křižovatek v úterý 29. března 2005. Pět křižovatek přitom bylo zatížených vysoce frekventovaným automobilovým provozem a pět křižovatek bylo středně frekventovaných. Provedl jsem též měření hustoty provozu, abych alespoň přibližně formalizoval rozmezí provedené klasifikace hustoty provozu.

Na ulici Boleslavská poblíž křižovatky s ulicí Zbožská jsem v daný den napočítal po 8. hodině ranní během časového intervalu 15 minut 291 motorových vozidel, což odpovídá frekvenci asi 19 vozidel za minutu.

Na ulici Zbožská jsem stejným způsobem došel k počtu 174 vozidel, což odpovídá asi 12 vozidlům za minutu.

Daný provoz odpovídá dopravní špičce, času, kdy frekvence vozidel dosahuje svého denního maxima. Stejnou studii jsem provedl na stejných místech a stejnou metodikou na Velikonoční pondělí 28. března 2005 po 20. hodině. Provoz odpovídal 116 vozidlům v ulici Boleslavská a 53 vozidlům v ulici Zbožská. To je po přepočtení přibližně 8 a 4 vozidel za minutu.

Vodítkem k výpočtu zpoždění získaného při přejezdu frekventovaných křižovatek však vychází pouze z experimentálního měření, jehož výsledky jsou v tabulce 5.4.

doba čekání u vysoce frekventované křižovatky (s)	doba čekání u středně frekventované křižovatky (s)
58	0
0	11
11	12
14	0
0	9
38	7
19	0
0	0
17	15
19	5
28	0
0	10
0	14
27	8
33	0
průměrná doba (s)	
17,6	6,1

Tabulka 5.4

Popisné vlastnosti lze v programu Kokeš k příslušným vektorovým řetězcům zapsat pomocí tabulky vlastnosti objektů. Tato tabulka má dva sloupce a variabilní počet řádků. První sloupec je přitom určen pro název vlastnosti, tedy vlastně typ atributu. Druhý sloupec se pak naplňuje hodnotami atributů. Pro hladký průběh digitalizace jsou atributy zapsány pouze do dvou řádků. První řádek s označením 1 nese název ulice nebo pojmenování cesty. Druhý řádek s označením 2 nese zbývající atributy, které jsou řazeny sekvenčně za sebou, takže tvoří alfanumerický řetězec o délce 8 až 9 znaků. První dvojice znaků udává typ komunikace pro cyklisty, následuje typ komunikace pro chodce, dále jsou znaky charakterizující směr pohybu

cyklisty a chodce, na sedmé a osmé pozici je dvojice znaků udávajících povrch komunikace. Poslední znak je vyhrazen pro číslici udávající penalizaci za průjezd křižovatky z hlavní silnicí. Pokud má řetězec pouze 8 znaků, pak je penalizace implicitně nulová.

Příkladem častého atributového řetězce je *krchaoas*. Tento řetězec charakterizuje vektor modelující krajnici asfaltové vozovky pro pohyb cyklistů, kteří se mohou pohybovat v směru souhlasném s orientací vektoru. Vektor zároveň modeluje chodník po kterém se mohou chodci pohybovat oběma směry. Jedná se tedy o typický soubor atributů modelujících klasickou ulici s vozovkou a chodníky při okrajích.

Dalším příkladem atributového řetězce může být *pspsaodl2*. Tento řetězec modeluje přejezd, respektive přechod rovně přes křižovatku s hlavní silnicí, která je zatížena středně silným provozem. Pohyb na jízdním kole je opět pouze v jednom směru, chodci se mohou pohybovat obousměrně. Jedná se o křižovatku, v jejíž blízkosti se nenachází značený přechod pro chodce, jinak by podél tohoto vektoru nebyl pohyb chodců možný. Vozovka je dlážděna kamennými kostkami.

6 Dijkstrův algoritmus

Pro účely vyhledání optimální cesty se nejčastěji využívá Dijkstrův algoritmus. Tento algoritmus je pojmenován po nizozemském matematikovi Edsgeru Wybu Dijkstrovi, který jej popsal jako první.

Definice Dijkstrova algoritmu je založena na zavedení pojmu grafu z oblasti diskrétní matematiky.

Graf se podle [3] str. 11 skládá z vrcholů a hran. Hrana vždy spojuje dva vrcholy a je buď orientovaná, nebo neorientovaná. U hran orientovaných rozlišujeme počáteční a koncový vrchol a říkáme, že hrana vede z počátečního do koncového vrcholu. Neorientované hrany chápeme jako symetrické spojení dvou vrcholů.

Pokud hrana spojuje nějaký vrchol se sebou samým, nazýváme ji smyčkou. Orientovaný graf má všechny hrany orientované, neorientovaný graf má všechny hrany neorientované. Smíšenými grafy se nazývají takové grafy, které mají oba druhy hran.

Orientovaný graf je podle [3] str. 11 trojice $G = (V, E, \varepsilon)$ tvořená neprázdnou konečnou množinou V , jejíž prvky nazýváme vrcholy, konečnou množinou E , jejíž prvky nazýváme orientovanými hranami, a zobrazením $\varepsilon: E \rightarrow V^2$, které nazýváme vztahem incidence. Toto zobrazení přiřazuje každé hraně $e \in E$ uspořádanou dvojici vrcholů (x, y) . Prvý z nich, x , nazýváme počátečním vrcholem hrany a značíme jej $PV(e)$. Druhý nazýváme koncovým vrcholem hrany a značíme jej $KV(e)$.

O hraně e říkáme, že vede z vrcholu x do vrcholu y a také, že spojuje vrcholy x a y . O vrcholech x, y pak říkáme, že jsou incidentní nebo že incidují s hranou e a také naopak e je incidentní s vrcholy x, y . Oba vrcholy x, y také souhrnně nazýváme krajními vrcholy hrany e . Jestliže $PV(e) = KV(e)$, pak hranu e nazýváme smyčkou, která může být jak orientovaná, tak neorientovaná. Vrchol, který není incidentní s žádnou hranou, nazýváme izolovaným vrcholem.

Je možné, aby několik hran mělo stejné počáteční a koncové vrcholy, tj. aby pro různé hrany e_1, e_2 platilo $PV(e_1) = PV(e_2)$ a $KV(e_1) = KV(e_2)$ nebo, zapsáno jinak, $(e_1) = (e_2)$. O takových hranách říkáme, že jsou rovnoběžné nebo též násobné. Množina hran grafu může být prázdná.

Posloupnost vrcholů a hran $v_0, e_1, v_1, e_2, v_2, \dots, e_k, v_k$ nazýváme podle [3] str. 19 orientovaným sledem, jestliže pro každou hranu e_i z této posloupnosti platí $PV(e_i) = v_{i-1}$ a $KV(e_i) = v_i$.

Posloupnost vrcholů a hran nazýváme neorientovaným sledem, jestliže každá hrana e_i z této posloupnosti spojuje vrcholy v_{i-1}, v_i .

Vrchol v_0 v obou případech nazýváme počátečním a vrchol v_k koncovým vrcholem sledu. O sledu říkáme, že vede z vrcholu v_0 do vrcholu v_k , nebo také, že spojuje vrcholy v_0, v_k .

V orientovaném i neorientovaném sledu na sebe vrcholy i hrany navazují. U orientovaného sledu však navíc požadujeme, aby všechny hrany byly orientovány „vpřed ve směru sledu“. U neorientovaného sledu na orientaci nezáleží. Proto má pojem neorientovaného sledu smysl pro orientované i neorientované grafy. V obecných sledech se mohou vrcholy i hrany opakovat.

Orientovaný sled, v němž se žádná hrana neopakuje, nazýváme podle [3] str. 20 orientovaným tahem. Orientovaný sled, v němž se neopakuje žádný vrchol, nazýváme orientovanou cestou. Každá cesta je zároveň též tahem. Tah nemusí být cestou.

Mějme podle [3] str. 85 orientovaný graf G , jehož každá hrana $E(G)$ je ohodnocena reálným číslem $a(e)$, které nazýváme délkou hrany. Délka cesty je součet délek jednotlivých hran tvořících cestu. Jsou-li x, y dva vrcholy grafu, pak vzdálenost $u(x, y)$ z vrcholu x do

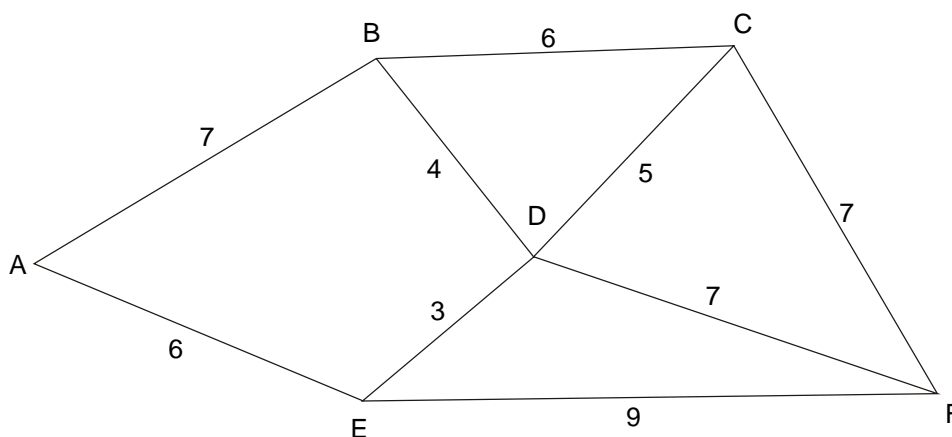
vrcholu y definujeme jako délku nejkratší cesty z x do y , pokud vůbec nějaká cesta z x do y existuje. Jestliže neexistuje, definujeme vzdálenost $u(x, y) = \infty$.

Pokud je ohodnocení hran nezáporné, pak se pro výpočet nejkratší cesty v grafu nejčastěji používá Dijkstrův algoritmus.

Podle [3] str. 101 je znění Dijkstrova algoritmu následující: Necht' je dán graf G , výchozí vrchol r a ohodnocení hran $a: E(G) \rightarrow \mathbb{R}$ takové, že pro všechny hrany platí $a(e) \geq 0$. Výsledkem Dijkstrova algoritmu jsou hodnoty $U(v)$ a $O(v)$, kde $U(v)$ odpovídá délce cesty a $O(v)$ odpovídá předchozímu uzlu cesty. Jako pomocné proměnné se používá množina vrcholů D taková, že pro všechny vrcholy $v \in D$ bude hodnota $U(v)$ rovna vzdálenosti $u(r, v)$.

Algoritmus se skládá ze tří kroků

1. Inicializace. Položíme $U(r) = 0$, pro ostatní vrcholy položíme $U(v) = \infty$. D je prázdná množina.
2. Výběr vrcholu. Jestliže pro všechny vrcholy $v \in V \setminus D$ platí $U(v) = \infty$, výpočet končí. Jinak z množiny $V \setminus D$ vybereme vrchol x , který má nejnižší hodnotu $U(x)$, zařadíme jej do množiny D a pokračujeme krokem 3.
3. Zpracování hran vycházejících z vrcholu x . Pro každou hranu $e \in E'(x)$ provedeme krok 4 a po zpracování všech hran pokračujeme podle kroku 2. E' je množina všech hran, pro které platí $PV(e) = x$ v případě orientovaného grafu. Pokud se jedná o graf neorientovaný, může platit $PV(e) = x$ nebo $KV(e) = x$.
4. Pokud $x = PV(e)$ necht' platí $y = KV(e)$. Pokud $x = KV(e)$ necht' platí $y = PV(e)$. Jestliže platí $U(x) + a(e) < U(y)$, pak položíme $U(y) = U(x) + a(e)$ a položíme $O(y) = x$. Po vyhodnocení hrany se pokračuje krokem 3.



Obrázek 6.1

Algoritmus lze snadno ozřejmit na následujícím příkladu. Na obrázku 6.1 je znázorněn neorientovaný graf, jehož vrcholy jsou označeny písmeny A až F a hrany jsou číselně ohodnoceny. Za ohodnocením si přitom lze představit délku hrany, nebo čas nutný k jejímu překonání. Pokud Dijkstrovým algoritmem chceme vyhledat nejkratší cestu z vrcholu A do vrcholu F, postupujeme následujícím způsobem:

V prvním kroku položíme hodnotu $U(A) = 0$, což vlastně neznamená nic jiného, než skutečnost, že z uzlu A se lze do uzlu A vést cestou dlouhou 0 jednotek. Ostatním uzlům je přiřazena hodnota $U(B) = \infty$, $U(C) = \infty$, $U(D) = \infty$, $U(E) = \infty$, $U(F) = \infty$. To odpovídá skutečnosti, že po úvodním kroku algoritmu není známa žádná cesta z výchozího uzlu do jiných uzlů.

Ve druhém kroku vybereme vrchol v s nejnižší hodnotou $U(v)$. Tím je právě výchozí vrchol A. Tento vrchol zařadíme do množiny D .

Ve třetím kroku vybereme podmnožinu E' . Jedná se o neorientovaný graf, takže vybereme vlastně všechny hrany incidující s vrcholem A. Jedná se o dvě hrany ohodnocené čísly 6 a 7.

S touto podmnožinou otestujeme hodnoty $U(B)$ a $U(E)$. Protože platí $0 + 7 < \infty$, přiřadíme $U(B) = 0 + 7 = 7$ a $O(B) = A$. Protože též platí $0 + 6 < \infty$, přiřadíme $U(E) = 0 + 6 = 6$ a $O(E) = A$.

Po prvním průchodu cyklem Dijkstrova algoritmu tedy platí $U(A) = 0$, $U(B) = 7$, $U(C) = \infty$, $U(D) = \infty$, $U(E) = 6$, $U(F) = \infty$. Přitom $D = \{A\}$, $O(B) = A$, $O(E) = A$.

Dále opakujeme druhý krok. Tentokrát zařadíme do množiny D vrchol E, neboť vrchol A s hodnotou $U(A)$ je již prvkem množiny D a nejmenší z hodnot $U(x)$ pro $x \in V \setminus D$ tak odpovídá právě vrcholu E.

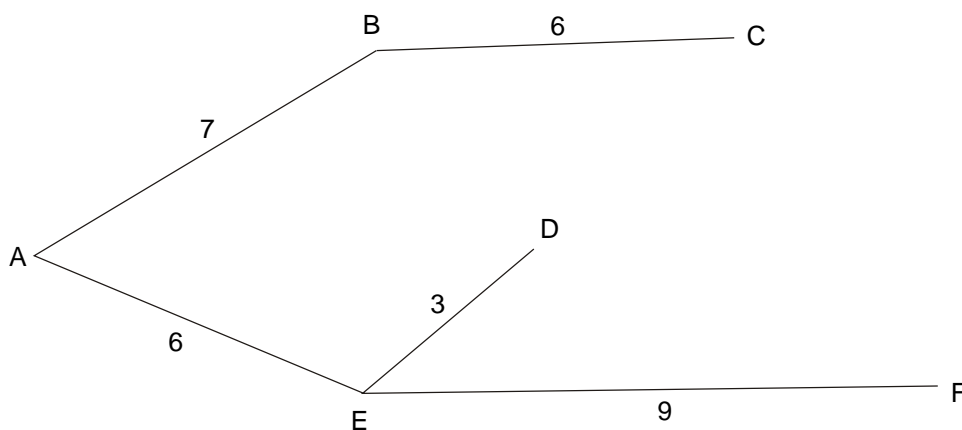
S vrcholem E incidují 3 hrany. Protože neplatí $6 + 6 < 0$, zůstane hodnota $U(A) = 0$. Pro vrcholy D a F však platí $6 + 3 < \infty$ a $6 + 9 < \infty$, takže budou zapsány nové hodnoty $U(D) = 6 + 3 = 9$ a $U(F) = 6 + 9 = 15$.

Po druhém průchodu cyklem tedy budou hodnoty $U(A) = 0$, $U(B) = 7$, $U(C) = \infty$, $U(D) = 9$, $U(E) = 6$, $U(F) = 15$. Přitom $D = \{A, E\}$, $O(B) = A$, $O(D) = E$, $O(E) = A$, $O(F) = E$.

V dalším cyklu bude analogicky $U(A) = 0$, $U(B) = 7$, $U(C) = 13$, $U(D) = 9$, $U(E) = 6$, $U(F) = 15$. Přitom $D = \{A, B, E\}$, $O(B) = A$, $O(C) = B$, $O(D) = E$, $O(E) = A$, $O(F) = E$.

V následujících cyklech se bude již měnit pouze množina D, do které přibudou vrcholy D, C a jako poslední vrchol F. Paralelně se budou měnit též hodnoty $O(x)$. Délky cesty a předchozí uzly se již měnit nebudou, neboť nebude pro žádný vrchol splněna nerovnost ve čtvrtém kroku.

Takovýmto způsobem lze tedy získat nejkratší cesty z výchozího uzlu do všech dostupných uzlů grafu. Pokud je uzel nedostupný, projeví se to hodnotou ∞ v délce cesty. Jako výsledek známe jak délku cesty, tak jednotlivé vrcholy cesty. Posloupnost vrcholů a hran tvořících nejkratší cestu se zachovává ve formě stromové struktury, jejímž kořenem je výchozí uzel. Tento strom je vlastně podmnožinou původního grafu, která vznikla vypuštěním množiny hran, přes které nevede žádná z cest. Strom odpovídající uvedenému příkladu je uveden v obrázku číslo 6.2. Pokud si zvolíme libovolný uzel x_1 , lze sestavit posloupnost vrcholů tak, že postupně zapisujeme uzly $x_1, x_2, x_3 \dots x_n$, kde $x_{n+1} = O(x_n)$ a n odpovídá výchozímu uzlu.



Obrázek 6.2

Je zřejmé, že algoritmus závisí na volbě výchozího uzlu. Pro každý uzel tak existuje jiný strom nejkratších cest.

Zásadní charakteristikou algoritmu je jeho složitost. Informatika rozlišuje dva typy složitosti. Složitost prostorovou a složitost časovou. Prostorová složitost zohledňuje náročnost

algoritmu na použití operační paměti. Časová složitost udává závislost doby výpočtu na objemu zpracovávaných dat. Pokud označíme počet těchto dat n , pak časová složitost je obecně funkcí tohoto čísla. Příkladem je může být časová složitost vyjádřená rovnicí $T(n) = an + b$, což je příklad lineární závislosti, kde a , b jsou určité konstanty. Jako časová složitost algoritmu se často uvádí pouze typ funkční závislosti bez ohledu na konstanty. Hovoří se pak o lineární, kvadratické, kubické či exponenciální časové složitosti.

Pokud vyhledáváme nejkratší cestu v grafu, který má n vrcholů, v každém cyklu vyhledáváme mezi n vrcholy uzel x s minimální hodnotu $U(x)$. Dobu nutnou k vyhodnocení jednoho uzlu označíme konstantou a . Pokud dále označíme konstantou b dobu nutnou k posouzení možností přiřazení nových hodnot $U(y)$, která závisí na počtu hran incidujících s konkrétním uzlem, náročnost jednoho cyklu je dána vztahem $an + b$.

Pokud platí, že pro každou dvojici uzlů grafu existuje alespoň jedna cesta mezi těmito uzly, pak počet cyklů algoritmu odpovídá počtu uzlů, neboť v každém cyklu můžeme do množiny D přiřadit pouze jeden nový uzel.

Výsledný vzorec je tedy $T(n) = n(an + b) = an^2 + bn$. Jedná se tedy o kvadratickou časovou náročnost.

Při určitých aplikacích Dijkstrova algoritmu však postačuje nalezení nejkratší cesty pouze mezi dvěma vrcholy, takže výpočet lze zastavit ve chvíli, kdy strom nejkratších cest obsahuje cílový uzel cesty. Průměrný počet cyklů se tak sníží na polovinu.

Souvislost grafu a modelu sítě komunikací je zřejmá, pokud se nahradí pojem vektorových řetězců termínem hrana a body styku dvou a více samostatných řetězců termínem uzel. Síť komunikací lze modelovat i duálním způsobem, kdy jednotlivé úseky mezi křižovatkami jsou uzly grafu a hrany mezi uzly modelují existenci křižovatek mezi úseky. První jmenovaný model je však pro lidské vnímání přirozenější.

Topologický vektorový model je tak vlastně orientovaným grafem. V elementárním případě může být ohodnocením délka úseků mezi uzly. Tak je realizováno vyhledávání optimální cesty městem pro chodce, kdy slovo optimální lze skutečně nahradit slovem nejkratší.

7 Transformace ze souřadnicového systému SJTSK do WGS 84

Vzhledem ke skutečnosti, že geometrická data digitálního modelu jsou založena na lokalizaci lomových bodů vektorových řetězců pomocí souřadnic systému SJTSK, nabízí se příležitost nabídnout potenciálním uživatelům vyhledávače možnost výpisu významných bodů na vyhledané trase ve formě souřadnic podporovaných přijímači GPS.

Současné turistické přístroje pro příjem signálu vysílaného družicemi systému GPS vyhodnocují polohu v geocentrickém souřadnicovém systému WGS 84, který byl vytvořen jako systém univerzální, vhodný pro použití po celé Zemi. Poloha v systému WGS 84 se udává obvykle v zeměpisných souřadnicích vztažených ke stejnojmennému elipsoidu, výška je dána k povrchu elipsoidu.

Pro rovinný souřadnicový systém SJTSK je referenčním tělesem Besselův elipsoid. JTSK byla budována v letech 1920 až 1957. Vzhledem k metodě vyrovnání polohy bodů JTSK vykazuje síť určité deformace, takže pro přesný převod mezi systémy nelze použít univerzálně platný transformační klíč.

Pro účely využití technologie GPS v geodézii byl vytvořen na území České republiky soubor geodetických bodů ZPBP označovaný jako body vybrané údržby. Body, které jsou v něm zahrnuty, mají kromě polohy v systému SJTSK určenu též polohu v systému WGS 84. Tyto geodetické body jsou stejně jako ostatní body ZPBP udržovány za státní prostředky Zeměměřickým úřadem. Body vybrané údržby jsou rozloženy s hustotou asi 3 až 5 bodů na jeden triangulační list a jsou tedy vhodnými identickými body pro výpočet klíčů pro obousměrný převod mezi danými souřadnicovými soustavami.

Turistické navigační přístroje jsou v současnosti schopny standardně poskytovat údaje o poloze se střední chybou v řádech jednotek metrů. Například přístroj Garmin GPS 60 podle [17] poskytuje při standardním měření údaje o poloze s chybou nepřesahující s pravděpodobností 95% 7,5 m. Vzhledem k této skutečnosti lze použít zjednodušený postup výpočtu lokálního transformačního klíče analogický s postupem podle [2] str. 109.

Rovinné souřadnice v systému SJTSK se převedou na polární souřadnice a provedou se inverzní konformní zobrazení na kouli a na Besselův elipsoid.

Pro převod mezi zeměpisnými souřadnicemi na Besselově elipsoidu a na elipsoidu WGS 84 lze pak nalézt klíč pomocí afinní transformace v obecném tvaru:

$$B_W = B_B + a + bX + cY$$
$$L_W = L_B + d + eX + fY$$

kde a , b , c , d , e , f jsou neznámé konstanty. B a L jsou hodnoty zeměpisných souřadnic, index W značí elipsoid WGS 84, index B značí Besselův elipsoid. X a Y jsou rovinné souřadnice daných bodů v systému SJTSK.

Pro zjištění neznámých konstant je třeba použít alespoň 3 identické body. V případě většího počtu bodů lze hodnoty konstant vyrovnat nebo použít místo afinní transformace polynomiální transformaci vyššího řádu.

Pro sestavení klíče jsem vyhledal v blízkém okolí Nymburka pomocí [15] 3 body ZPBP vybrané údržby tak, aby tvořily trojúhelník. Zájmové území je rozloženo přibližně kolem těžiště trojúhelníku. Jedná se o následující geodetické body:

TB 0915240110 na katastrálním území Bobnice se souřadnicemi SJTSK $Y = 694946,02$ m, $X = 1034580,95$ m a souřadnicemi ETRS89 $B = 50^\circ 13' 09,962''$ a $L = 15^\circ 04' 10,5407''$.

TB 0915240280 na katastrálním území Drahelice se souřadnicemi SJTSK $Y = 699092,92$ m, $X = 1037748,30$ m a souřadnicemi ETRS89 $B = 50^\circ 11' 11,0974''$ a $L = 15^\circ 01' 03,6005''$.

TB 0915250070 na katastrálním území Chvalovice u Nymburka se souřadnicemi SJTSK $Y = 695603,85$ m, $X = 1041544,15$ m a souřadnicemi ETRS89 $B = 50^\circ 09' 23,7110''$ a $L = 15^\circ 04' 22,4845''$.

Pro každý z těchto bodů jsem vypočítal polohu v zeměpisných souřadnicích na Besselově elipsoidu. Výpočet byl proveden podle [1] str. 67-70. Rovinné pravoúhlé souřadnice X, Y byly převedeny rovinné polární souřadnice ρ, ε podle vzorce:

$$r = \sqrt{X^2 + Y^2}$$

$$e = \arctan \frac{Y}{X}$$

Rovinné polární souřadnice ρ, ε byly zobrazeny konformně na kartografické souřadnice \check{S}, D :

$$n = \sin S_0$$

$$S = 2 \left(\arctan \left(\tan \left(\frac{S_0}{2} + \frac{p}{4} \right) \cdot \sqrt{\frac{R}{r \tan S_0}} \right) - \frac{p}{4} \right)$$

$$D = \frac{e}{n}$$

kde R je poloměr referenční koule $6380703,6105$ m a S_0 je základní kartografická rovnoběžka s hodnotou $78^\circ 30'$.

Kartografické souřadnice byly převedeny na zeměpisné souřadnice U, V na kulové ploše:

$$U = \arcsin(\sin U_K \sin S - \cos U_K \cos S \cos D)$$

$$V = V_K - \arcsin \frac{\cos S \sin D}{\cos U}$$

kde $U_K = 59 42 42,6969$ a $V_K = 42 31 31,41725$.

Souřadnice U, V byly zobrazeny konformně na Besselův elipsoid pomocí vzorců:

$$\Delta U = U - U_0$$

$$B = B_0 + 1,001416022789 \Delta U - 8,687150417 \cdot 10^{-5} \Delta U^2 +$$

$$+ 1,670197 \cdot 10^{-7} \Delta U^3 + 1,175089 \cdot 10^{-8} \Delta U^4$$

$$L = \frac{V}{a}$$

kde B_0 je základní rovnoběžka s hodnotou $49^\circ 30'$, U_0 je $49^\circ 27' 35,84625''$ a α je konstanta o velikosti $1,000597498372$. Vzorec pro B je pouze aproximativní a platí pro ΔU ve stupních.

Výsledné zeměpisné souřadnice na Besselově elipsoidu jsou následující:

TB 0915240110 $B = 50^\circ 13' 12,7563''$, $L = 32^\circ 44' 14,8437''$

TB 0915240280 $B = 50^\circ 11' 13,8845''$, $L = 32^\circ 41' 7,8686''$

TB 0915250070 $B = 50^{\circ} 9' 26,4797''$, $L = 32^{\circ} 44' 26,7709''$

Ze zeměpisných souřadnic identických bodů byly vypočítány příslušné konstanty. Pro souřadnice v radiánové míře a rovinné souřadnice v metrech jsou jejich hodnoty následující:

$a = -0,00002791486184$, $b = 1,794766423 \cdot 10^{-11}$, $c = -6,039821140 \cdot 10^{-12}$,
 $d = -0,3083949407$, $e = 8,316018643 \cdot 10^{-12}$, $f = 3,449812111 \cdot 10^{-11}$.

Pro posouzení přesnosti transformačního klíče jsem provedl převod dalších identických bodů nacházejících se v blízkosti Nymburka. Vyhodnocení přesnosti je v tabulce 6.1. U každého bodu je udána vzdálenost od středu zájmového území, pro které byl klíč sestaven a odchylky polohy vypočítané od vyrovnané polohy deklarované v údajích o trigonometrickém bodu.

TB	vzdál. od zájm. území (m)	vyrovnaná poloha v systému WGS 84						vypočtená poloha v systému WGS 84						ΔB (vt.)	ΔL (vt.)
		B (st. min. vt.)			L (st. min. vt.)			B (st. min. vt.)			L (st. min. vt.)				
0915240110	3,842	50	13	9,9627	15	4	10,5407	50	13	9,9627	15	4	10,5407	0,0000	0,0000
0915240280	1,729	50	11	11,0974	15	1	3,6005	50	11	11,0974	15	1	3,6005	0,0000	0,0000
0915250070	4,361	50	9	23,7110	15	4	22,4845	50	9	23,7110	15	4	22,4845	0,0000	0,0000
0915240050	5,587	50	14	10,5218	15	0	37,2818	50	14	10,5251	15	0	37,2801	0,0033	0,0017
0914040380	7,075	50	10	58,0125	14	56	33,9964	50	10	58,0134	14	56	33,9938	0,0009	0,0026
0915190270	8,177	50	12	28,6120	15	9	8,2845	50	12	28,6141	15	9	8,2869	0,0021	0,0024
0915200010	8,158	50	10	34,1504	15	9	12,0501	50	10	34,1524	15	9	12,0502	0,0020	0,0001
0915250210	7,856	50	7	54,2657	15	6	11,3700	50	7	54,2651	15	6	11,3714	0,0006	0,0014

Tabulka 6.1

Maximální dosažené odchylky v šířce i délce jsou v řádech tisíců vteřiny, přitom v zeměpisné délce odpovídá tato odchylka skutečné délce podle vzorce:

$$d = \Delta L \left(\frac{a}{\sqrt{1 - e^2 \sin^2 B}} + h \right) \cos B$$

kde B je zeměpisná šířka, ΔL difference v zeměpisné délce v radiánech, a je délka hlavní poloosy elipsoidu WGS84 6378137 m, e^2 je excentricita elipsoidu 0,006694379990141, h je převýšení nad elipsoidem.

Pro zeměpisnou šířku 50° , převýšení 200 m odpovídá ΔL 0,005'' délce 0,100 m, což je dostačující přesnost pro daný účel.

Postup převodu souřadnic do systému WGS 84 je tedy realizován zpětným zobrazením z rovinných souřadnic SJTSK na zeměpisné souřadnice na Besselově elipsoidu a dále převodem do systému WGS 84 pomocí vypočítaného transformačního klíče.

8 Převod dat do databázové struktury

V prostředí Kokeš bylo provedeno zakreslení vektorových řetězců a jejich opatření popisnými daty. Ukládání dat v tomto programu je orientováno souborově přičemž mezi datové formáty vyvinuté společností GEPRO pro tento účel patří VYK a VTX. VYK je proprietární binárně orientovaný formát. VTX je formát duální k formátu VYK, data jsou v něm zapsána textově. V následujícím odstavci je vzorový soubor typu VTX.

```
&V DP YX CM 0 0
&R 696787.16 1036425.15 697491.44 1036995.31 1000
&C W
&O 0 1
&A 1=Za Žoskou
&A 2=krkrbapa
&L P 697105.83 1036494.55 K=614
    L 697026.06 1036507.37
    L 697012.52 1036523.74
    L 697011.19 1036544.50
    L 697029.57 1036622.91
    L 697042.01 1036698.86
    L 697040.26 1036720.65
    L 696883.66 1036938.48
    L 696846.46 1036962.31
&O 0 2
&A 1=Maršála Koněva
&A 2=krchaoas
&L P 696901.30 1037546.03 K=614
    L 696965.88 1037540.08
&O 0 3
&A 1=Boženy Němcové
&A 2=pspsaoas
&L P 696965.88 1037540.08 K=614
    L 696937.93 1037331.58
&K
```

Struktura formátu VTX je velmi přímočará, úvodní tři řádky slouží k určení pořadí složek souřadnic SJTSK, přesnosti souřadnic a typu použité znakové sady. Dále již následuje popis vektorové kresby. V daném případě se vektorová kresba skládá pouze z vektorů nebo vektorových řetězců.

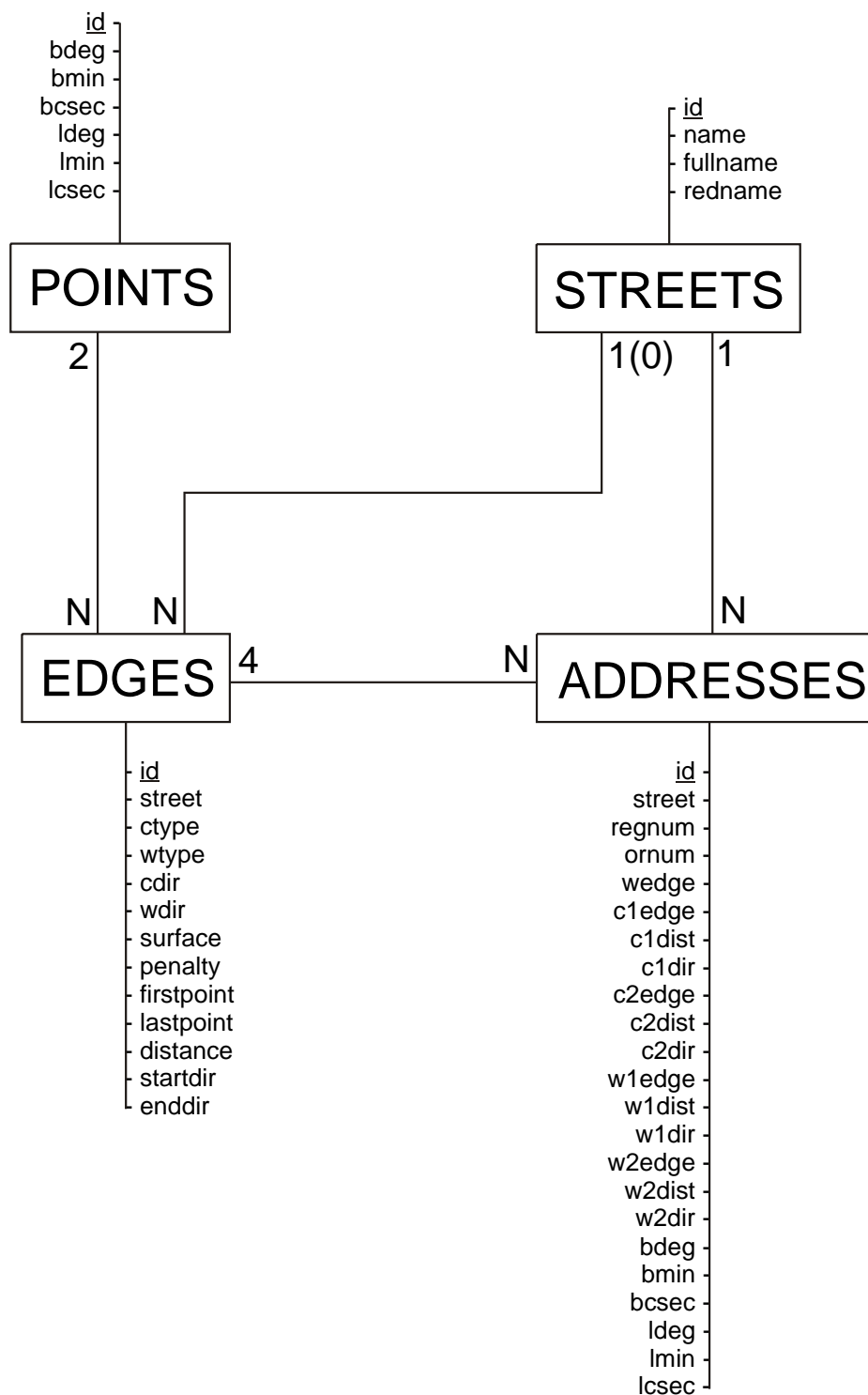
Znakem &O je uvozena dvojice čísel udávajících vrstvu kresby a číslo objektu. Všechny vektorové řetězce jsou v daném případě uloženy ve vrstvě 0 a jsou číslovány vzestupně od čísla 1.

&A označuje řádek nesoucí popisné informace objektu. Každý objekt má dva atributy, první atribut zahrnuje úplný název ulice, druhý atribut nese znakový řetězec udávající zbývajících šest atributů, jak byly popsány v části Realizace digitálního vektorového modelu komunikací.

&L P uvozuje první bod vektorového řetězce, který je dán souřadnicemi v systému SJTSK v pořadí YX a s přesností na centimetry. Na konci řádku je informace o kresebném klíči, která je nedůležitá. Následující body vektorového řetězce jsou již uvozeny pouze znakem L. Řetězec se skládá vždy minimálně ze dvou bodů definujících počátek a konec úsečkového vektoru. Počet vnitřních lomových bodů řetězce není omezen. Informace o objektu jsou ukončeny začátkem popisu nového objektu.

V uvedeném příkladu je jako objekt číslo 1 polygonový řetězec skládající se z osmi vektorů, daných sedmi vnitřními lomovými body. Jedná se o řetězec modelující krajnici

silnice v ulici Za Žoskou, která je určena pro pohyb chodce souhlasně s orientací řetězce a pro pohyb cyklisty v opačném směru. Daná cesta je opatřena povrchem ze silničních panelů.



Obrázek 8.1

Všechny počáteční a koncové body vektorových řetězců jsou zároveň uzlovými body. Koncový bod vektoru definovaného v příkladu jako objekt 2 je totožný s počátečním bodem vektoru 3, což znamená, že mají společný uzel, jedná se tedy o sousedné spojnice. Tato skutečnost je však dána nepřímo rovností příslušných souřadnic. Soubor VTX je tedy takzvaný špagetový vektorový model.

Pro efektivní vyhledávání optimální cesty je třeba zavést do modelu topologické vazby. Model přitom může být oproti standardnímu způsobu realizace, jak byl popsán v kapitole Digitální vektorový model komunikací, zjednodušený. V daném případě není třeba realizovat žádný popis ploch ani jejich topologii.

Databázový model jsem se proto rozhodl uspořádat způsobem, který je patrný z ERA diagramu na obrázku 8.1. Model se skládá ze čtyř tabulek: edges (hrany), points (uzlové body), streets (ulice), addresses (adresní body).

Pro převod vektorového modelu jsem vytvořil speciální program VTX2CSV, který dávkově produkuje dané tabulky ve formátu CSV. Jedná se o konzolový program, který byl naprogramován v jazyce ANSI C. Pracuje v dávkovém režimu, přičemž vstupem jsou tři specifické soubory. Názvy souborů se zadávají jako vstupní parametr při spouštění programu pomocí příslušného EXE souboru. Zdrojový kód programu VTX2CSV je v příloze této diplomové práce.

Program ke své činnosti vyžaduje jednak již zmiňovaný soubor výkresu ve formátu VTX. Jako vedlejší zdroje dat přitom používá tento program samostatný soubor úplných názvů ulic, respektive názvů vektorových řetězců. Jedná se o textový soubor typu CSV s kódováním Windows 1250, jednotlivé názvy jsou vždy na novém řádku. Souborový formát CSV je přitom jednoduchý textový formát sloužící k záznamu tabulkových dat. Položky jsou v něm uspořádány po řádcích a vzájemně jsou odděleny speciálním znakem, většinou středníkem. Příklad položek v tomto souboru je uveden v následujícím odstavci:

```
1. Máje
2. května
28. října
Akátová
Alej Esperanto
Alešova
Alfonse Muchy
Azalková
B. Němcové
Baráčnická
Bílkova
Blahoslavova
Bobnická
Boleslavská
Boučkova
```

Dále je pro převod použit CSV soubor adresních bodů. Jedná se o body v souřadnicovém systému SJTSK charakterizující polohu budovy s číslem popisným, popřípadě s číslem evidenčním. V souboru jsou postupně zaznamenány pro každou položku úplný název ulice nebo označení cesty, číslo popisné nebo číslo evidenční, dále číslo orientační a souřadnice adresního bodu SJTSK v pořadí Y, X. Souřadnice jsou uvedeny s přesností na celé metry. Pokud nemá budova číslo orientační, je místo něj uvedena hodnota 0. Příklad souboru adresních bodů je uveden v následujícím odstavci.

```
Maršála Koněva;939;33;696753;1037568
Raisova;1683;9;696992;1037756
Komenského;518;36;697819;1037475
Dlabačova;2165;0;697235;1038442
Tyršova;224;16;697729;1038149
Malé Valy;353;20;697512;1037831
Husova;387;0;697476;1037322
Boleslavská;2090;56;697450;1036485
Petra Bezruč;360;0;697002;1037269
Máchova;453;2;697979;1038113
```

Zbožská;497;13;697896;1037726
 Na Bělidlech;291;1;697405;1038448
 Jičínská;1677;25;697630;1037011
 Eliščina;470;5;697538;1038004
 Boleslavská;368;15;697721;1037625

Klíčovou složkou datového modelu je tabulka edges obsahující hrany, respektive spojnice uzlových bodů tvořící vlastní topologický digitální model komunikační sítě. Tyto hrany program VTX2CSV generuje na základě vektorové kresby v souboru VTX. Každá hrana tak odpovídá právě jednomu vektoru nebo vektorovému řetězci v původní vektorové kresbě. Příklad obsahu tabulky edges je uveden v tabulce 8.1.

EDGES												
id	street	ctype	wtype	cdir	wdir	surface	penalty	firstpoint	lastpoint	distance	startdir	enddir
1	294	1	1	2	1	3	0	1	2	61465	310	136
2	294	1	1	2	1	1	0	3	4	32726	312	98
3	294	2	2	3	3	1	1	1	4	2027	133	333
4	294	1	1	1	2	1	0	5	6	3970	256	110
5	294	0	2	0	3	0	0	6	3	812	205	5

Tabulka 8.1

Atribut id je zde klíčem a odpovídá inkrementálnímu číslování objektů v souboru výkresu.

Street je cizí klíč z tabulky ulic. Pomocí tohoto klíče je každé hraně přiřazeno jméno, které většinou odpovídá názvu ulice, kterou příslušná hrana prochází. V určitých případech nemá hrana žádné jméno, to je implementováno hodnotou atributu street rovnou 0.

Atribut ctype je číselný kód typu hrany pro pohyb cyklistů. Typy hran jsou popsány v části Realizace digitálního vektorového modelu komunikací a číselné kódy odpovídají číselnému označení v tabulce 5.1. Obdobně atribut wtype definuje typ hrany pro pohyb chodců.

Atributy cdir a wdir udávají směr, jakým se po komunikaci modelované danou hranou mohou pohybovat cyklisté, respektive chodci. Číselné kódování odpovídá tabulce 5.2. Atribut surface definuje materiál, kterým je opatřen povrch cesty modelované danou hranou. Číselné kódy odpovídají označení v tabulce 5.3.

Atribut penalty charakterizuje klasifikaci hrany do jedné ze čtyř tříd pro účely zavedení pravděpodobného čekání při přejezdu rušných komunikací. Atribut penalty může nabývat celočíselných hodnot 0 až 3.

Atributy firstpoint a lastpoint jsou cizí klíče z tabulky uzlových bodů. Pokud mají dvě hrany shodnou hodnotu jednoho z těchto atributů, znamená to, že se jedná o sousedné hrany. Existence cesty přes tyto vektory však závisí ještě na atributu cdir, respektive wdir.

V uvedené tabulce jsou sousedními dvojice hran 1 a 3, 2 a 3, 2 a 5, 4 a 5. Cesta z uzlu 5 do uzlu 3 přitom není možná ani pro cyklisty ani pro chodce, neboť cyklisté se nesmí pohybovat po hraně 5 a chodci se po hraně 4 mohou pohybovat pouze opačným směrem. Cesta z uzlu 3 do uzlu 5 je možná pouze pro chodce.

Atribut distance udává délku hrany. Délka je definována v centimetrech, neboť tak lze používat celočíselné hodnoty a zároveň zachovat přesnost této veličiny. Tento postup je vhodný především kvůli úspoře kapacity nutné pro uchování digitálního modelu. Většina systémů řízení báze dat umožňuje používat široké spektrum celočíselných datových typů, aby bylo možné databáze navrhnout v tomto ohledu co nejúspěšněji.

V daném případě byl použit datový typ unsigned mediumint, který umožňuje ukládat kladná celá čísla v rozmezí 0 až 16777215, přitom zabírá paměť o velikosti 3 B. Nejmenší datový typ umožňující destinný rozvoj vyžaduje paměť o velikosti 4 B.

Délky hran byly získány součtem délek jednotlivých úseků vektorového řetězce určených početně z rovinných souřadnic. Zásadním zjednodušením, které bylo učiněno, je vypuštění veškerých délkových redukcí.

Přítom správně by bylo třeba zohlednit délkovou redukci způsobenou zobrazením v SJTSK. Podle nomogramu je v oblasti Nymburka zkreslení asi -7 cm/km. To znamená, že je nutné zvýšit vodorovnou délku odsunutou z mapy v systému SJTSK na každý km délky o 7 cm.

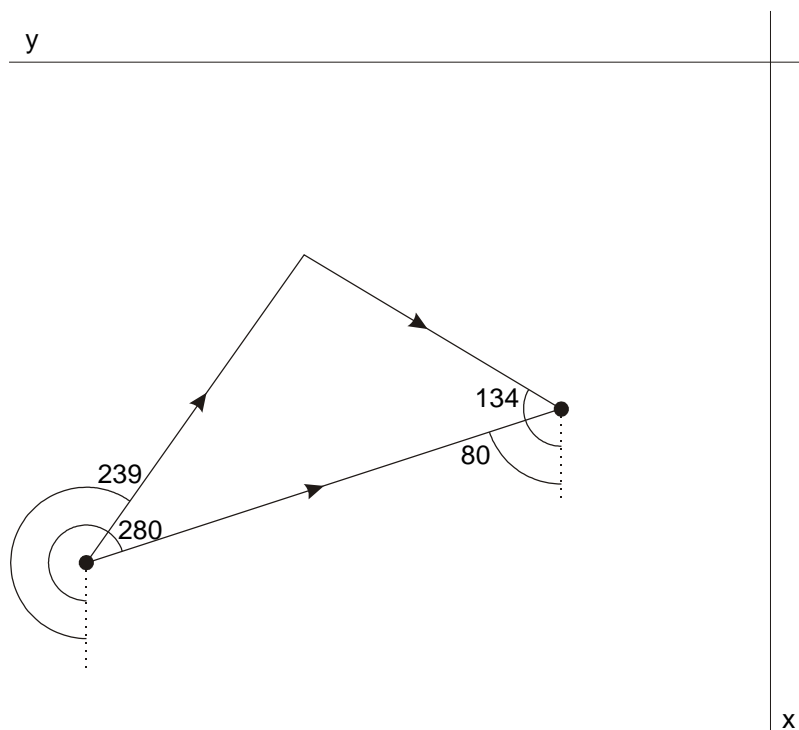
Dále je nutné též uvažovat redukci délky do skutečného horizontu, neboť tato délka je vztažena k nulovému horizontu. Pokud je terén ve výšce 200 m nad nulovým horizontem, zvětší se vodorovná délka podle vzorce:

$$s = \frac{s_0 R}{R - h}$$

kde s_0 je délka vztažená k nulovému horizontu, to je v našem případě délka vypočtená ze souřadnic, R je poloměr referenční koule a h je výška skutečného horizontu. Pro $s_0 = 1000,07$ m a $h = 200$ m je $s = 1000,10$ m.

Především je však třeba uvažovat zvětšení délky vlivem výškových poměrů reliéfu. Pokud má povrch konstantní sklon 1 %, pak vodorovné délce 1000,10 m ve směru největšího spádu odpovídá šikmá délka 1000,15 m.

Vzhledem ke skutečnosti, že terén Nymburka je rovinatý, přičemž průměrná nadmořská výška je asi 185 m, lze délkové redukce zanedbat s ohledem na relativně malou velikost, která je řádově nižší než chyby vzniklé generalizací digitálního modelu.



Obrázek 8.2

Poslední dvojicí atributů v tabulce edges jsou startdir a enddir. To jsou směrníky počátečního a koncového vektoru vektorového řetězce vyčíslené v gonech. Pokud se hrana skládá z jediného vektoru, je rozdíl hodnot počátečního a koncového směrníku roven 200 gónů. Tyto informace se při popisu optimální cesty městem používají pro určení směru v uzlových bodech. Způsob určení těchto směrníků je patrný z obrázku 8.2.

Soubor adresních bodů, který je jedním ze vstupních souborů programu VTX2CSV, obsahuje informace o množině všech potenciálních výchozích a cílových bodů. Tyto adresní body musí být ovšem svázány s vektory realizujícími síť komunikací. Jinými slovy je třeba realizovat v modelu odpověď na otázku, co je to za komunikaci, která je před budovou odkud se chci pohybovat.

V tabulce 8.2 je uveden příklad obsahu tabulky addresses.

ADDRESSES 1/2									
id	street	regnum	ornum	c1edge	c1dist	c1dir	c2edge	c2dist	c2dir
1	126	939	33	346	14083	-1	347	1641	1
2	196	1683	9	265	2959	-1	264	6108	1
3	93	518	36	644	5400	-1	645	1441	1
4	27	2165	0	1352	2529	-1	1350	4711	1
5	241	224	16	1206	1131	-1	1194	825	1

ADDRESSES 2/2												
id	w1edge	w1dist	w1dir	w2edge	w2dist	w2dir	bdeg	bmin	bcsec	ldeg	lmin	lcsec
1	347	1641	1	347	1641	1	50	11	1678	15	2	330
2	264	6108	1	264	6108	1	50	10	5564	15	2	3251
3	645	1441	1	645	1441	1	50	11	4084	15	2	1198
4	1350	4711	1	1352	2529	-1	50	11	935	15	2	2299
5	1194	825	1	1194	825	1	50	11	2075	15	2	1139

Tabulka 8.2

Klíčem v této tabulce je opět atribut id, dále je jako atribut regnum uvedeno číslo popisné nebo číslo evidenční a číslo orientační jako atribut ornum. V případě, že budova nemá přiděleno orientační číslo, je hodnota implicitně nulová.

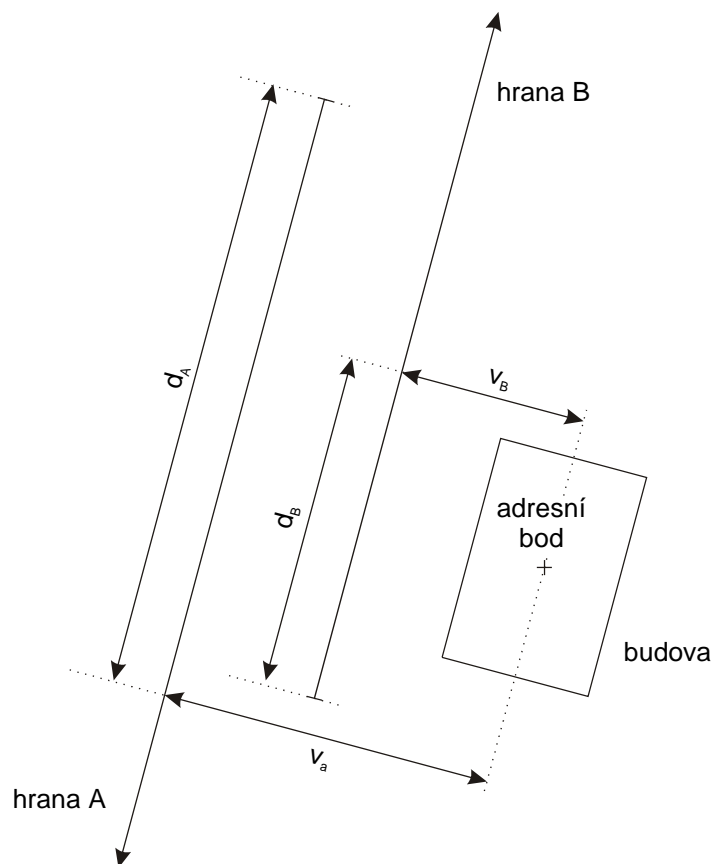
Atribut cledge v tabulce addresses označuje klíč hrany, po které se bude pohybovat cyklista, pokud na ulici před budovou zabočí vlevo, c1dist je délka od počátku hrany k místu, ze kterého se cyklista po hraně pohybuje. Atribut c1dir udává polohu budovy vůči danému vektoru. Pokud je budova vlevo od vektoru ve směru orientace vektoru, pak je hodnotou atributu c1dir -1. Pokud je napravo, pak je hodnota atributu rovna 1.

Obdobným způsobem popisují návaznost adresního bodu na model komunikační sítě atributy c2edge, c2dist a c2dir. V tomto případě je ovšem popsána relace s hranou, po které se bude cyklista pohybovat, pokud se vydá od budovy vpravo.

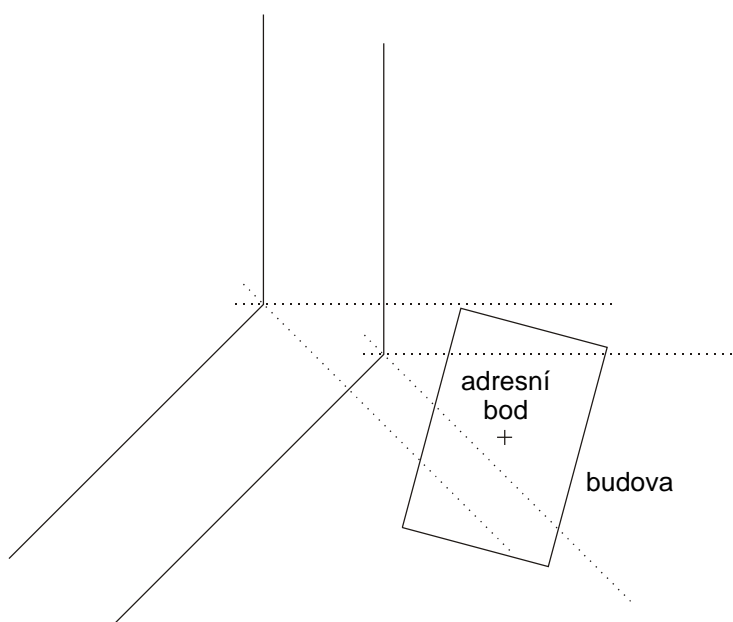
Na obrázku 8.3 je znázorněna budova se svým adresním bodem umístěným přibližně v geometrickém středu zastavěné plochy a dvě hrany. Každá z obou hran je tvořena pouze jedním vektorem. Pokud dané hrany modelují krajnice ulice s obousměrným provozem, pak je možné se na jízdním kole pohybovat pouze ve směru orientace hran. Klíč hrany A pak odpovídá atributu cledge pro danou budovu, vzdálenost d_A v centimetrech odpovídá hodnotě atributu c1dist a c1dir má hodnotu -1. Analogicky bude klíč hrany B odpovídat atributu c2edge, c2dist bude rovno hodnotě d_B a c2dir bude mít hodnotu 1.

Obdobným způsobem jsou realizovány v tabulce addresses relace s hranami pro pohyb chodců pomocí atributů wledge, w1dist, w1dir, w2edge, w2dist, w2dir. Pokud se před budovou nachází chodník, mají wledge i w2edge stejnou hodnotu, neboť po chodníku se lze vždy pohybovat oběma směry.

Příslušné relace vyhledává program VTX2CSV samočinně na základě jednoduché prostorové analýzy. Pro každý adresní bod je vyhledána mezi hranami taková, která má shodný název ulice s ulicí v adrese, odpovídá požadavku na směr pohybu a má minimální vzdálenost v od dané hrany. Takovýmto způsobem se vyhledají postupně relace v obou směrech jak pro pohyb na jízdním kole, tak pro chůzi.



Obrázek 8.3



Obrázek 8.4

Vzdálenost adresního bodu od hrany se přitom získá spuštěním kolmice ke hraně úsečky tvořící příslušný vektorový řetězec, která prochází adresním bodem. V určitých případech však nastává situace, kdy žádná taková kolmice neexistuje. Taková situace je zachycena na obrázku číslo 8.4.

Z tohoto důvodu je každá úsečka pro účely této analýzy pomyslně prodloužena o určitou hodnotu, čímž se zamezuje vzniku dané singularity. Tato hodnota je v programu stanovena na 10 m.

Volba tohoto postupu budování relací mezi adresami a sítí komunikací nahrazuje extrémně náročnou práci, která by vznikla v případě, že by byly tyto relace budovány manuálně. Zároveň lze těžit ze skutečnosti, že adresní body jsou data, která lze získat z jiných zdrojů. Tvorbou, údržbou a prodejem těchto dat se zabývá podle [14] například společnost Central European Data Agency.

Jako poslední položky jsou v tabulce addresses uvedeny pomocí atributů bdeg, bmin, bcsec, ldeg, lmin, lcsec souřadnice zeměpisné šířky a délky na elipsoidu WGS 84 ve tvaru rozloženém na stupně, minuty a setiny vteřin. Tyto souřadnice jsou transformovány z původních souřadnic adresních bodů, které byly v rovinném souřadnicovém systému SJTSK, postupem popsaným v části Transformace bodů se souřadnicového systému SJTSK do souřadnicového systému WGS 84.

Tabulka streets obsahuje opět klíč tvořený inkrementálním očíslováním entit v atributu id. Dále obsahuje názvy ulic a pojmenování hran ve třech modifikacích. Atribut fullname nese vždy plný název v kódování Windows 1250. Atribut name je název zbavený mezer a interpunkce, všechna písmena jsou převedena na malá. Tento tvar je používán pro vyhledávání v názvech ulic. Konečně atribut redname obsahuje úplný název, který je zbaven české diakritiky.

Tabulka points nese informace o poloze uzlových bodů. Klíčem je opět atribut id, který je následován atributy bdeg, bmin, bcsec, ldeg, lmin, lcsec, které udávají polohu bodů na elipsoidu WGS 84.

STREETS			
id	fullname	name	redname
1	1. Máje	1maje	1. Maje
2	2. května	2kvetna	2. kvetna
3	28. října	28rijna	28. rijna
4	Akátová	akatova	Akatova
5	Alej Esperanto	alejesperanto	Alej Esperanto

Tabulka 8.3

POINTS						
id	bdeg	bmin	bcsec	ldeg	lmin	lcsec
1	50	11	5959	15	2	3485
2	50	11	4565	15	2	5084
3	50	12	29	15	2	1746
4	50	11	5984	15	2	3391
5	50	11	5951	15	2	1583

Tabulka 8.4

V programu VTX2CSV je rovněž implementována kontrola konzistence dat výkresu. Program kontroluje formální správnost jmen ulic a dokáže odhalit též diskrepance v ostatních

atributových datech. Rovněž kontroluje správný průběh prostorové analýzy vytvářející relace mezi adresními body a hranami. Pokud je nalezena pouze jedna relace s hranou pro pohyb cyklistů, je vydáno varovné hlášení, které sděluje nutnost zkontrolovat, zda se příslušná budova skutečně nachází v ulici s jednosměrným provozem, což je jediný případ, kdy je taková situace přípustná. Dále je také kontrolováno překročení kritické hodnoty vzdálenosti adresního bodu od příslušné hrany. Tato kritická hodnota je nastavena na 25 m. Všechna varovná hlášení jsou zapisována do souboru Log.txt, který je ukládán automaticky do adresáře, ve kterém se program nachází.

9 Realizace vyhledávače

Vyhledávač optimální cesty městem pro přístup přes Internet je složen z databázového systému, internetových stránek ve formátu HTML tvořících uživatelské rozhraní a konečně stránky ve formátu PHP, která implementuje vlastní proces vyhledávání.

Databázový systém se skládá z dat, jejichž struktura byla popsána v části Převod dat do databázové struktury, spravovaných systémem řízení báze dat MySQL. Data byla importována z příslušných CSV souborů do připravených databázových tabulek. Při přípravě databázových tabulek jsem nadefinoval názvy tabulek a názvy všech entit. U každé entity je přitom nutné definovat též datový typ. Všechna data jsou celočíselná, proto i všechny datové typy v databázi jsou celočíselné a byly vždy voleny tak, aby zahrnovaly celý potenciální rozsah příslušných hodnot a přitom měly co nejnižší paměťové nároky. V každé tabulce byly též definovány atributy *id* jako primární klíče. Výpis souboru SQL příkazů pro tvorbu tabulek je uveden v příloze a je obsažen též v souboru *Plain.sql* na datovém kompaktním disku, který je součástí této diplomové práce.

HTML soubory tvořící uživatelské rozhraní jsou *index.htm*, *smart.htm*, *help.htm*, *smarthelp.htm*, *about.htm*. Součástí rozhraní je též obrázek s logem vyhledávače v souboru *kudy.gif*. Všechny uvedené soubory se nacházejí též na přiloženém kompaktním disku v adresáři WWW.

Při tvorbě HTML stránek jsem používal editor Microsoft FrontpageExpress 2.0 ve spojení s manuální úpravou kód v editoru PSPad 4.3. Výsledný kód odpovídá standardu HTML 4.01, který je v současnosti platnou normou. Daný souborový formát byl vyvinut právě pro potřeby předávání textových, obrazových i zvukových informací prostřednictvím sítě Internet. Jedná se o jazyk obsahující vlastní textovou složku určenou pro uživatele a dále soustavu zvláštních textových značek nazývaných anglickým termínem tag. Pomocí těchto značek lze do stránky snadno vkládat obrázky, které přitom zůstávají v původním souborovém formátu. Nejvíce se přitom používají rastrové obrázky ve formátech JPEG, GIF, BMP a PNG. Pomocí tagů lze realizovat celou řadu interaktivních konstrukcí. Často se používají hypertextové odkazy, to jsou ve výsledném HTML dokumentu znakové řetězce nebo obrázky, po jejichž stisknutí je uživatel přesměrován na jinou internetovou stránku. Dále se používají formuláře umožňující interakci mezi uživatelem a vzdáleným serverem formou uživatelsky vkládaného textu, přepínáním tlačítek či zaškrťáváním políček. Všechny vyjmenované vlastnosti byly při tvorbě internetového rozhraní vyhledávače Kudy v Nymburce použity.

Soubor *index.htm* obsahuje úvodní HTML stránku vyhledávače. Vizuální vzhled této stránky je zachycen na obrázku 10.1. Stránka je určena pro zjednodušený typ vyhledávání a zároveň poskytuje odkazy na další stránky.

Ve střední části úvodní stránky se nacházejí pole pro uživatelské zadávání výchozí a cílové adresy, mezi kterými má být vyhledána optimální cesta. Rovněž je zde sada tří prepínacích tlačítek, pomocí kterých uživatel rozhoduje, pro jaký druh pohybu má být cesta optimalizována. Tyto interaktivní prvky jsou v rámci HTML kódu zahrnuty v elementu formulář. Součástí formuláře je rovněž tlačítko Vyhledat sloužící k odeslání zadaných dat přes síť Internet na vzdálený server ve formě takzvané datové kolekce. Pro zasílání dat z formulářů lze použít metodu *get* nebo metodu *post*.

Metoda *get* přitom připojuje proměnné do specifického řetězce za adresu stránky. Pokud například odešleme proměnnou *language* s hodnotou *cz* a proměnnou *item* s hodnotou *1* tak, aby byla zpracována při přípravě stránky *main.php*, bude adresa končit řetězcem *main.php?language=cz&item=1*.

Metoda *post* předává proměnné skrytě v hlavičce předcházející každému HTML dokumentu.

Metoda `get` se používá zejména v případech, kdy je vhodné částečně uživatelům poodhalit funkcionalitu stránek. Naopak se nehodí v případech kdy je soubor odesílaných parametrů velký. Objem dat u této metody je omezen podle [13] velikostí 4 kB. V daném případě byla implementována metoda `post`, která posílá data stránky *go.php*. Zasílají se přitom zadané řetězce. Volby provedené pomocí tlačítek se přenášejí pomocí číselných kódů parametru *who*, kde hodnota *1* odpovídá chodci, *2* cyklistovi a *3* osobě na invalidním vozíku.

Úvodní stránka obsahuje ve své spodní části hypertextové odkazy směřující na stránku *about.htm* obsahující základní informace o vyhledávači a účelu jeho vzniku. Další odkaz je na stránku obsahující nápovědu k vyhledávání v souboru *help.htm*. Jedná se o stránku vycházející funkčně ze stránky úvodní, kde jsou navíc obsaženy textové údaje nápovědy.

Na úvodní stránce je též hypertextový odkaz na inteligentní vyhledávání. Jedná se o vyhledávání s pokročilými funkcemi, které vyžaduje jednak zadání standardních údajů o výchozí a cílové adrese a typ pohybu, pro který má být vyhledávání optimalizováno, ale je zde obsažena i možnost uživatelského zadávání konstant ovlivňujících vyhledávání optimálních cest pro jízdu na kole v závislosti na kvalitě povrchu a dále je možné v rámci inteligentního vyhledávání uživatelsky volit průměrnou rychlost chůze nebo pohybu na invalidním vozíku a jízdy na kole.

Na stránce inteligentního vyhledávání *smart.htm* je v dolní části odkaz na nápovědu k inteligentnímu vyhledávání obsaženou v souboru *smarthelp.htm* s analogickou strukturou, která je obohacena o podrobné pokyny k vyhledávání s pokročilými funkcemi.

Rozšiřující parametry jsou stejně jako základní parametry na úrovni kódu HTML opět součástí jediného formuláře a uživatelsky zadané údaje se odesílají na vzdálený server metodou `post`.

Jak v případě základního vyhledávání, tak u inteligentního vyhledávání jsou datové kolekce obsahující uživatelsky zadané parametry směřovány na stránku *go.php*. Pro jednoznačné rozlišení původu dat je ve formuláři obsažen skrytý parametr *smart* určující formou číselného kódu *0* nebo *1* skutečnost, zda byly informace získány z základního vyhledávání na hlavní stránce nebo z inteligentního vyhledávání. Hlavním znakem skrytého parametru je skutečnost, že jeho hodnotu nelze uživatelsky měnit, neboť se ve vizuální podobě HTML stránky nijak neprojevuje.

Stránka PHP se obvykle skládá z části, která odpovídá kódu HTML a z části obsahující PHP kód, nazývaný skript. PHP je serverový skriptovací jazyk, jak bylo popsáno v části Softwarové prostředky pro realizaci vyhledávače. To znamená, že pokaždé, když uživatel požádá prostřednictvím prohlížeče o zobrazení stránky PHP, je vykonán skript obsažený v dané stránce.

Jazyk PHP umožňuje používání standardních programových konstrukcí stejně jako řady sofistikovaných funkcí. Výsledkem PHP skriptu je většinou nahrazení kódu skriptu HTML kódem, který je výsledkem skriptu.

V případě vyhledávače Kudy v Nymburce určené pro Internet je jedinou stránkou se skriptem právě soubor *go.php*. Kód na stránce je v úvodní části standardním HTML definujícím umístění loga a textu na stránce.

Ve střední části je umístěn vlastní PHP skript. V jeho úvodní části se nacházejí funkce používané v hlavním podprogramu. Hlavní podprogram obsahuje na začátku úvodní definice proměnných a dále přebírá z kolekce předané metodou `post` proměnné uživatelsky zadané pomocí HTML formuláře. Počet těchto proměnných záleží na tom, zda byla kolekce odeslána z formuláře základního vyhledávání nebo vyhledávání s pokročilými funkcemi. Vyhledávání ale v obou případech probíhá na stejném principu, takže zbývající proměnné jsou v případě základního vyhledávání doplněny implicitními hodnotami. V případě koeficientů pro objížďení úseků s nekvalitním povrchem se jedná o hodnoty *1* pro všechny materiály, takže při základním vyhledávání nejsou preferovány žádné povrchy.

Uživatelsky zadané řetězce odpovídající číselným hodnotám se převedou na čísla, přičemž je zohledněna možnost použití jak desetinných teček, tak desetinných čárek. Jsou kontrolovány číselné hodnoty odpovídající konstantám pro objíždění úseků s nekvalitním povrchem. Pokud nejsou všechny hodnoty v intervalu od 0,5 do 5, pak je uživatel na počátku výpisu nalezené cesty upozorněn na skutečnost, že zadané koeficienty neodpovídají realitě. Rovněž je kontrolována hodnota zadané průměrné rychlosti, která nesmí být nulová. Všechny číselné hodnoty musí být kladné, takže záporná znaménka jsou při načítání uživatelsky zadávaných čísel ignorována.

Ověřování platnosti zbývajících parametrů musí být již provedeno za použití dat, která jsou čerpána z databázového systému řízeného MySQL. Pro tento účel navazuje PHP skript s tímto systémem relací. V daném případě se databáze nachází na lokálním disku, je pojmenována nymburk a obsahuje příslušné tabulky, které byly popsány v části *Převod dat do databázové struktury*. Ověřuje se především platnost zadaných adres.

Jednak se ověřuje, zda byly zadány oba názvy ulice a zda bylo v obou adresách zadáno alespoň jedno z číselných označení budovy. Pokud dojde ke zjištění nekorektnosti parametru, je skript přerušeno a příslušná chyba je uživateli sdělena. Názvy ulic se posléze převedou na reprezentaci v řetězci bez české diakritiky a bez mezer a teček. K tomu slouží funkce *reduce* definovaná na počátku skriptu. Dále se kontroluje, zda jsou názvy ulic zadány správně a jednoznačně, což se zjišťuje na základě počtu výsledků prohledávání v bázi názvů ulic, respektive v databázové tabulce *streets*. Prohledává se přitom mezi atributy *name*, kde jsou názvy ulic stejně jako zadaný řetězec zbaveny české diakritiky, mezer a teček.

Výsledek prohledávání tabulky musí být právě jedna entita. Pokud jsou nalezeny dvě a více entit odpovídajících danému řetězci, znamená to, že zadaný název je nejednoznačný, na což je uživatel upozorněn a zpracování skriptu je zastaveno. Pokud není daná ulice v bázi dat, nedojde k nalezení žádného výsledku, což je opět chybový stav.

Při korektním zadání ulic v adrese jsou u získány primární klíče z tabulky *streets*. Jeden klíč odpovídá výchozí adrese a druhý reprezentuje ulici cílovou. Pomocí těchto klíčů a čísel popisných nebo orientačních lze pak vyhledat adresy v tabulce *addresses*. Pokud není příslušná adresa nalezena, znamená to, že je neplatná nebo není v databázi obsažena.

V případě korektního zadání obou adres je vytvořeno dvouprvkové pole záznamů $\$address[\$i]$, kde pro $\$i = 0$ jsou uloženy informace o výchozí adrese a pro $\$i = 1$ informace o adrese cílové. Záznam obsahuje položku *street* s klíčem k tabulce *streets*, položku *fullname* s úplným názvem ulice v kódování Windows 1250, dále číselné proměnné *regnum* a *ornum* s číslem popisným a číslem orientačním, dále jsou v každém záznamu tři čtyřprvková pole. Prvním z těchto polí je *point[\\$j]*, které uchovává klíče potenciálně prvních respektive posledních uzlových bodů optimální cesty. Dva body odpovídají přitom pohybu na jízdním kole a dva body chůzi. Přitom vždy jeden bod z příslušné dvojice odpovídá cestě vpravo a druhý cestě vlevo od výchozího bodu, respektive od bodu cílového. Pole *offset[\\$j]* nese informaci o délce od výchozího bodu k potenciálně prvnímu uzlu na cestě, respektive se jedná o délku od potenciálně posledního uzlového bodu cesty k cílovému bodu. Poslední pole je *direction[\\$j]*, které obsahuje velikosti směřníku vektoru v daném bodě v gonech. Na základě této informace se generuje informace o směru chůze v uzlových bodech.

V dalším kroku se již připravují výchozí podmínky pro Dijkstrův algoritmus. Pro účely tohoto algoritmu je ve skriptu pole záznamů *point[\\$i]*. Záznam nese položky *edge*, *price*, *distance*, *next* a *done*. Počet záznamů v tomto poli musí odpovídat počtu uzlových bodů v modelu, tedy velikosti největšího číselného klíče z tabulky *points*. Daná struktura je ve výsledku připravena uchovávat data popisující strom optimálních cest, tedy analogii stromu nejkratších cest grafem, jak byl popsán v části *Dijkstrův algoritmus*. Každý prvek pole přitom odpovídá jednomu uzlu stromu. Položka *edge* přitom uchovává klíč hrany, která je ve stromu

nejblíže k danému uzlu a klíč druhého uzlu incidujícího s touto hranou je uchován v položce záznamu *next*.

Položka *distance* nese informaci o vzdálenosti uzlu od kořene stromu v centimetrech. Položka *price* odpovídá ocenění vzdálenosti daného uzlu od kořene stromu s ohledem na povrch, jakým je opatřena cesta modelovaná danou hranou. Závislost ceny hrany na její délce je přitom lineární. Pokud je koeficient povrchu například 2 a délka hrany je 1000 cm, pak cena takové hrany je 2000 cm. V případě vyhledávání cesty pro chodce je ovšem cena hrany vždy shodná s délkou v modelu.

Pomocí položky *done* se implementuje množina *D* popsaná v části *Dijkstrův algoritmus*. Pokud je příznak *done* nenulový, pak je bod zahrnut v množině *D* a vzdálenost uzlu je již konečná, tedy minimální možná. V opačném případě je hodnota *done* rovna nule.

Aby byl usnadněn následný výpis pokynů popisujících optimální cestu, postupuje se při hledání cesty způsobem, kdy kořenem výsledného stromu je cílový bod a tento strom tedy znázorňuje nejkratší cesty ze všech uzlů, které mají vzdálenost do cílového uzlu stejnou nebo menší než je optimální cesta mezi zadaným výchozím a cílovým bodem.

V rámci přípravy výchozích podmínek pro vykování Dijkstrova algoritmu nad vektorovým topologickým modelem uloženým v databázi je tedy určena vzdálenost z potenciálně posledních uzlů cesty do cílového bodu. Tyto vzdálenosti jsou přitom určeny z hodnot *offset[\$j]* v prvku *\$address[1]* podle toho, zda se hledá optimální cesta pro cyklisty nebo pro chodce. V souladu s definicí Dijkstrova algoritmu je nutné zajistit, aby množina *D* byla na počátku prázdná a ohodnocení položek *price* všech vrcholů bylo provedeno hodnotou nekonečno. Tato operace se provádí již na počátku běhu skriptu při inicializaci proměnných, kdy jsou všechny proměnné *\$point[\$i]@done* nastaveny na hodnotu nula. Hodnoty *\$point[\$i]@price* jsou nastaveny na hodnotu *\$infinity*, což je hodnota 1000000000 cm, která je dostatečně velká, aby zajistila stejnou funkci jako hodnota ∞ v definici Dijkstrova algoritmu.

Vlastní Dijkstrův algoritmus je realizován pomocí do while cyklu, který se opakuje dokud existuje v modelu takový uzel, který má hodnotu *done* rovnu nule a přitom existuje alespoň jedna cesta mezi takovým bodem a cílovým bodem. Většinou je však cyklus zastaven alternativní podmínkou, kterou je zařazení obou potenciálních prvních uzlů cesty do množiny *D*, které se po implementační stránce děje přiřazením hodnot *done* rovných jedné.

Implementace algoritmu je přitom připravena i na častou situaci, kdy potenciální první uzel následující za výchozím bodem je pouze jeden. Taková situace nastává v případě, kdy se hledá cesta pro jízdu na kole a výchozí adresa je v ulici s jednosměrným provozem.

Uvnitř cyklu se přistupuje do tabulky *edges* databáze *nymburk* a vyhledávají se vhodné hrany. Kritériem pro výběr je typ hledané cesty, tedy zda jedná o cestu pro chodce či cyklisty, a směr pohybu po hraně, který musí odpovídat směru hledané cesty. Všechny hrany ve výsledném stromu musí být určeny pro pohyb směrem ke kořenu stromu. Při vyhledávání cesty pro osoby na invalidním vozíku jsou vyloučeny hrany modelující schodiště.

Algoritmus vyhledává cestu s nejmenší cenou, což je vlastně modifikovaná informace o délce hrany. V případě cesty pro chodce je cena hrany totožná s délkou hrany. U cesty pro cyklisty je odvozována jednak v závislosti na kvalitě povrchu vynásobením délky koeficientem pro příslušný povrch a dále je zohledňována hodnota penalizace.

Penalizace je dána u hran modelujících přejezd křižovatky s hlavní silnicí. Systém penalizace byl vysvětlen v části *Realizace digitálního vektorového modelu sítě komunikací*. Pokud má penalizace hodnotu 1 až 3, pak je k ceně hrany přičtena dráha, kterou cyklista urazí při rovnoměrném přímočarém pohybu za dobu odpovídající předpokládanému čekání na křižovatce. Pro penalizaci 1 je tato doba empiricky stanovena na 2 s. Pro penalizaci 2 je to 6 s. Penalizace 3 odpovídá zpoždění 18 s. Tyto hodnoty byly odvozeny z pozorování, které je popsáno v části *Realizace digitálního vektorového modelu sítě komunikací*. Pro výpočet délky

dráhy je používána průměrná rychlost jízdy, ať už se jedná o implicitní hodnotu 15 km/h v případě základního vyhledávání nebo o uživatelsky zadanou hodnotu. Délka dráhy v centimetrech je připočtena k ceně hrany.

Při běhu Dijkstrova algoritmu dochází k rozsáhlému vyhledávání informací o jednotlivých hranách, které je časově náročné. Aby byla práce s databází urychlena, je vhodné provést indexaci tabulky *edges* podle cizích klíčů uzlových bodů. Indexace je specifický proces umožněný systémem řízení báze dat vedoucí k vybudování zvláštní datové struktury urychlující vyhledávání podle atributu, nad kterým byla tato struktura vytvořena.

Po vykonání algoritmu je vyhodnocena délka, respektive cena, úplné cesty, která musí zahrnovat i cenu části hrany z výchozího bodu do prvního uzlu modelu.

Výsledná cesta je následně vypsána formou textu, kdy je uživateli sdělena úplná výchozí i cílová adresa, délka cesty, odhad doby nutné k jejímu překonání a pokyny pro pohyb k cíli. Příklad výpisu optimální cesty je uveden v následujícím odstavci

Optimální cesta pro chodce z adresy Komenského 518/36 do adresy Zbožská 497/13 je dlouhá asi 301 metrů. Při průměrné rychlosti 5 kilometrů za hodinu potrvá asi 4 minuty. Vede následujícím způsobem:

Pokud se nacházíte na ulici zády k výchozí budově, vyjděte vpravo po chodníku ulicí Komenského 55 metrů,
dále jděte rovně přes silnici vedoucí ulicí Jičínská 7 metrů,
zabočte vlevo a jděte po chodníku ulicí Jičínská 224 metrů,
dále jděte rovně přes silnici vedoucí ulicí Zbožská 12 metrů,
zabočte vpravo a jděte po chodníku ulicí Zbožská 3 metry.
Cílová budova se nachází vlevo.

Uživatel se dozví, jakým směrem se má vydat a sérii dalších pokynů nesoucích informace o názvech ulic, kterými se má pohybovat, o délce jednotlivých úseků. Také je upozorňován na nutnost zabočit. Systém pokynů je syntetizován na základě atributů databázové tabulky *edges*, přičemž pořadí jednotlivých hran je dáno výsledkem Dijkstrova algoritmu minimální cesty.

Implementace vyhledávání prostřednictvím technologie WAP je koncipována obdobným způsobem jako vyhledávač pro Internet. Pracuje nad stejnými daty a jeho funkčnost je obsahově shodná se základním vyhledáváním ve verzi pro klasické internetové vyhledávače. Inteligentní vyhledávání není implementováno, neboť v rámci technologie WAP nelze realizovat rozsáhlé formuláře tak jako je to možné v HTML stránkách. Zásadním limitujícím aspektem je zde maximální rozsah WML stránky omezený velikostí 1492 B.

Prostředí je vytvořeno ve standardu WAP 1.1, které patří mezi starší specifikace, čímž je zaručena funkčnost v širokém spektru mobilních zařízení. Všechny texty v rozhraní jsou bez české diakritiky, tedy bez háčků a bez čárek. WAP sice umožňuje vkládání českých znaků pomocí zvláštních kódů, takzvaných entit, ale podle [12] některé mikroprohližeče nezobrazují tyto znaky korektně.

Rozhraní WAP vyhledávače je tvořeno čtyřmi stránkami ve formátu WML a čtyřmi stránkami ve formátu PHP. Soubor *index.wml* obsahuje úvodní stránku, která je zachycena na obrázku 10.5.

Úvodní stránka obsahuje hypertextové odkazy na WML stránky s nápovědou a s informacemi o vyhledávači. Stránka s informacemi o vyhledávači obsahuje základní popis aplikace a účel jejího vzniku a je obsažena v souboru *about.wml*.

Stránka s nápovědou je rozdělena na dvě části obsažené v souborech *help1.wml* a *help2.wml* tak, aby byla zachována maximální přípustná velikost souborů. Tyto stránky jsou vzájemně propojeny pomocí hypertextových odkazů.

Především však jsou na úvodní stránce s hlavní nabídkou odkazy na formuláře vyhledávání. Zde dochází k volbě, pro jaký typ pohybu má být hledaná optimální cesta určena.

To je dáno tím, že daná specifikace WAP neumožňuje vkládat přepínací tlačítka, kterými byla tato volba řešena v HTML verzi. Podle typu pohybu je mikroprohlížeč přesměrován na jeden ze tří souborů *search1.php*, *search2.php* nebo *search3.php*.

Soubor *search1.php* je určen pro vyhledávání cesty pro chodce. Obsahuje dvě trojice polí určených pro zadání názvu ulice, čísla popisného a čísla orientačního jak výchozí, tak cílové adresy. Vyhledávání se spustí pomocí hypertextového odkazu *Vyhledat cestu*, přičemž jsou zároveň metodou get odeslány uživatelem zadané údaje ke zpracování v PHP skriptu stránky *go.php*. Mezi zasílané informace patří i parametr *who* určující typ pohybu. Stránky v souborech *search2.php* a *search3.php* jsou formálně totožné se *search1.php*, jedinou odlišností je hodnota parametru *who* v odesílané datové kolekci.

Na stránkách s formulářem je také možnost zvolit hypertextový odkaz *Vyhledat s GPS*, kde je v kolekci parametr *gps* upraven tak, že při výpisu navigačních pokynů dojde ke zobrazení zeměpisných souřadnic na elipsoidu WGS 84.

Kolekce má až na příznak *smart* a metodu předání shodnou strukturu s kolekcí odesílanou v internetové verzi vyhledávače. Díky tomu je soubor *go.php* obsahující jádro vyhledávače do značné míry shodný se stejně pojmenovaným souborem pro HTML verzi vyhledávače.

Odlišnosti jsou patrné v odstranění součástí skriptu, které implementovaly pokročilé funkce inteligentního vyhledávání, ale zejména v části PHP skriptu implementující syntézu navigačních pokynů z atributových informací o jednotlivých hranách. Je totiž nutné realizovat rozdělení výpisu na větší počet stránek, neboť soubory navigačních informací obvykle značně převyšují maximálně přípustnou velikost WML stránek.

Toto rozdělení je provedeno způsobem, v rámci kterého jsou informace zobrazovány až do naplnění mezní kapacity 1492 B. Přitom se musí zohledňovat i data, která jsou na každé stránce totožná, jako jsou hlavičky souboru. Pokud je kapacita naplněna a navigační zpráva není úplná, dojde k vygenerování hypertextového odkazu směřujícího na zprávu s následujícími pokyny. Pokud se nejedná o první soubor s navigačními pokyny, je vygenerován též odkaz na zprávu s předcházejícími pokyny. Na každé stránce je též odkaz na hlavní nabídku.

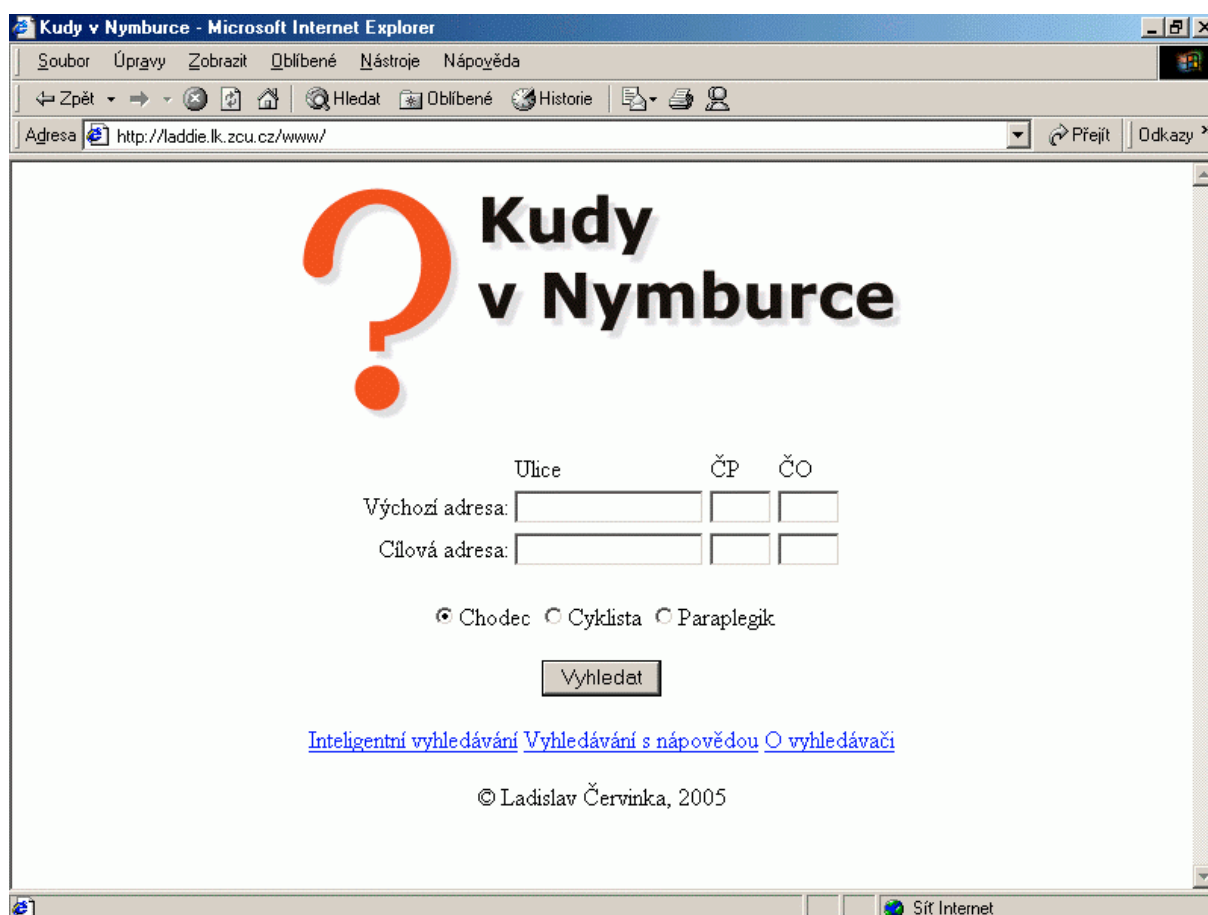
Soubor *go.php* je upraven tak, že generovaný text je bez české diakritiky. K tomu se využívají též jiné atributové informace z tabulky *streets*. Místo položky *fullname* obsahující úplný název ulic v kódování Windows 1250 se používají atributy *redname*, kde jsou tyto názvy zbaveny háčeků a čárek.

10 Uživatelská příručka

Vyhledávač optimální cesty městem Nymburk je online služba přístupná prostřednictvím Internetu a také pomocí technologie WAP. Obě verze jsou založeny na stejném základě.

Přístup ke službě prostřednictvím Internetu je možný s použitím počítače připojeného k síti Internet, na kterém je nainstalován libovolný prohlížeč podporující stránky ve formátu HTML 4.01. Takovými prohlížeči jsou například vyšší verze prohlížeče Internet Explorer společnosti Microsoft, prohlížeče Mozilla Firefox a další. Vyhledávač je nezávislý na druhu operačního systému

Po zadání adresy vyhledávače optimální cesty do prohlížeče se zobrazí úvodní stránka prohlížeče. Vzhled této stránky v prohlížeči Microsoft Internet Explorer verze 5.5 je na obrázku 10.1. V jiných prohlížečích a při odlišném nastavení obrazovky počítače může být vizuální vzhled prohlížeče odlišný.



Obrázek 10.1

Pod logem vyhledávače se nachází šest polí určených k zadávání textu. V prvním řádku jsou tři pole určená k zadávání údajů o výchozí adrese.

První pole je určeno pro název ulice nebo náměstí, kde se nachází budova, odkud bude vyhledávána optimální cesta. Lze použít úplný název ulice jako například Maršála Koněva, lze použít tvar bez české diakritiky, bez mezer a s libovolnou kombinací malých a velkých písmen jako například marsalaKONEVA. Lze ovšem také používat pouze části názvů ulic. Přitom je ale nutné, aby taková část názvu ulice byla jednoznačnou charakteristikou skutečného názvu ulice. Jako analogii předcházejících příkladů lze tedy použít název konev.

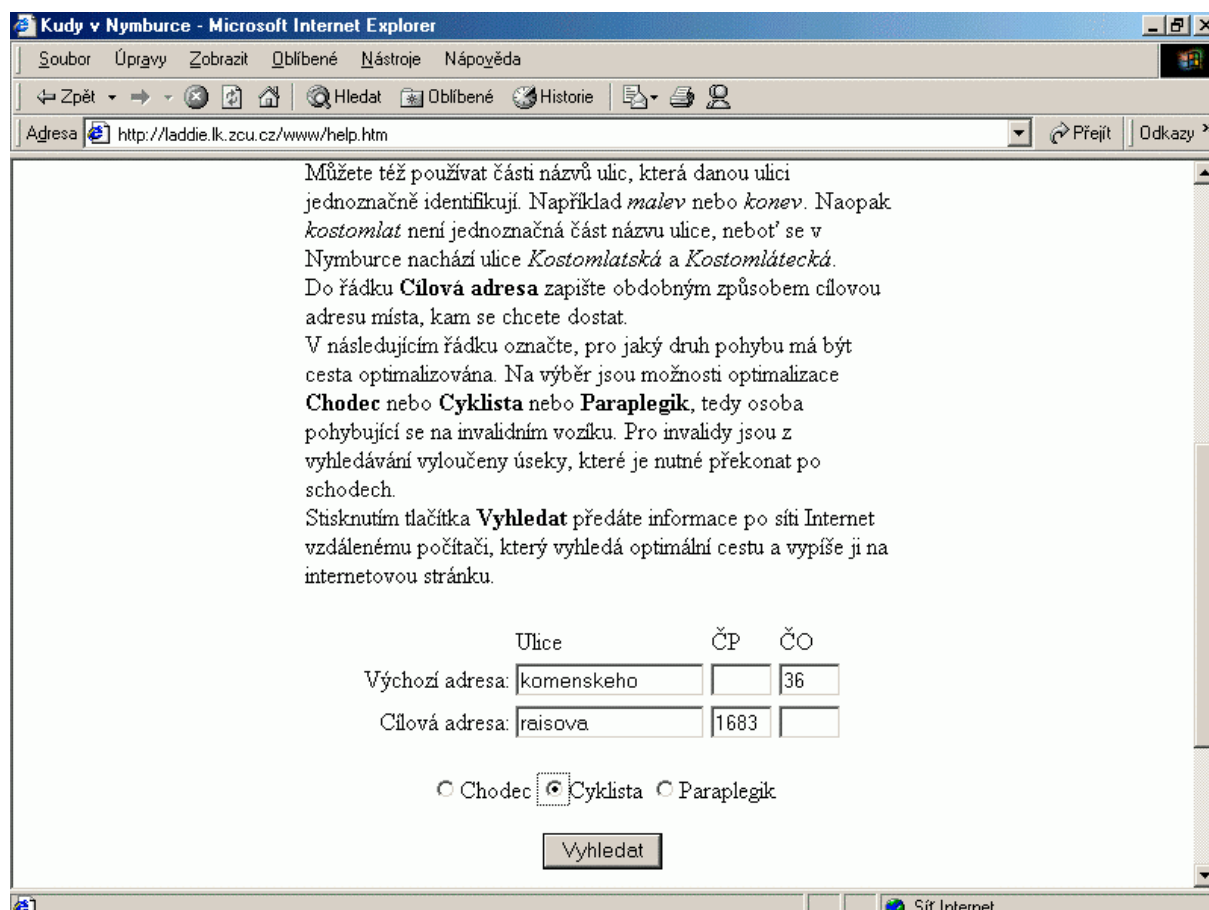
Naopak například část názvu kostomlat neurčuje ulici jednoznačně, neboť v Nymburce se nachází ulice Kostomlatská a Kostomlátecká.

Následující pole je určeno pro číslo popisné nebo číslo evidenční a konečně poslední pole je určeno pro číslo orientační. Čísla popisná jsou přitom v Nymburce obvykle bílá na červeném podkladu, orientační čísla jsou obvykle bílá na modrém podkladu. Jedno z těchto čísel musí být vždy vyplněno, jinak je adresa neúplná a výchozí bod cesty tak nelze jednoznačně lokalizovat. Druhý řádek je určen pro analogické údaje o cílové adrese.

Dále je třeba zvolit, pro jaký druh pohybu má být cesta určena. Na výběr jsou varianty cesty pro chodce, pro cyklistu nebo pro osobu na invalidním vozíku. Různé varianty jsou dány přirozeně tím, že některé komunikace, jako cesty parkem nebo některé lávky nejsou určeny pro cyklisty a naopak chodci jsou při přecházení komunikací vázáni povinnostmi použít značený přechod pro chodce. Cesty pro osoby na invalidním vozíku jsou vyhledávány tak, aby neobsahovaly schodiště.

V dolní části stránky je umístěn odkaz *Vyhledávání s nápovědou* směřující na obdobnou stránku doplněnou podrobnou nápovědou, která provádí uživatele všemi kroky při zadávání údajů do vyhledávače.

Příklad správně vyplněné stránky s nápovědou připravené k vyhledávání optimální cesty z adresy Komenského 36, kde 36 je orientační číslo, do adresy Raisova 1683, kde 1683 je číslo popisné, je na obrázku 10.2. Vyhledávání se spouští stisknutím tlačítka Vyhledat.



Kudy v Nymburce - Microsoft Internet Explorer

Soubor Úpravy Zobrazit Oblíbené Nástroje Nápověda

← Zpět → Hledat Oblíbené Historie

Adresa <http://laddie.lk.zcu.cz/www/help.htm> Přejít Odkazy >>

Můžete též používat části názvů ulic, která danou ulici jednoznačně identifikují. Například *malev* nebo *konev*. Naopak *kostomlat* není jednoznačná část názvu ulice, neboť se v Nymburce nachází ulice *Kostomlatská* a *Kostomlátecká*.
Do řádku **Cílová adresa** zapíše obdobným způsobem cílovou adresu místa, kam se chcete dostat.
V následujícím řádku označte, pro jaký druh pohybu má být cesta optimalizována. Na výběr jsou možnosti optimalizace **Chodec** nebo **Cyklista** nebo **Paraplegik**, tedy osoba pohybující se na invalidním vozíku. Pro invalidy jsou z vyhledávání vyloučeny úseky, které je nutné překonat po schodech.
Stisknutím tlačítka **Vyhledat** předáte informace po síti Internet vzdálenému počítači, který vyhledá optimální cestu a vypíše ji na internetovou stránku.

	Ulice	ČP	ČO
Výchozí adresa:	<input type="text" value="komenskeho"/>	<input type="text"/>	<input type="text" value="36"/>
Cílová adresa:	<input type="text" value="raisova"/>	<input type="text" value="1683"/>	<input type="text"/>

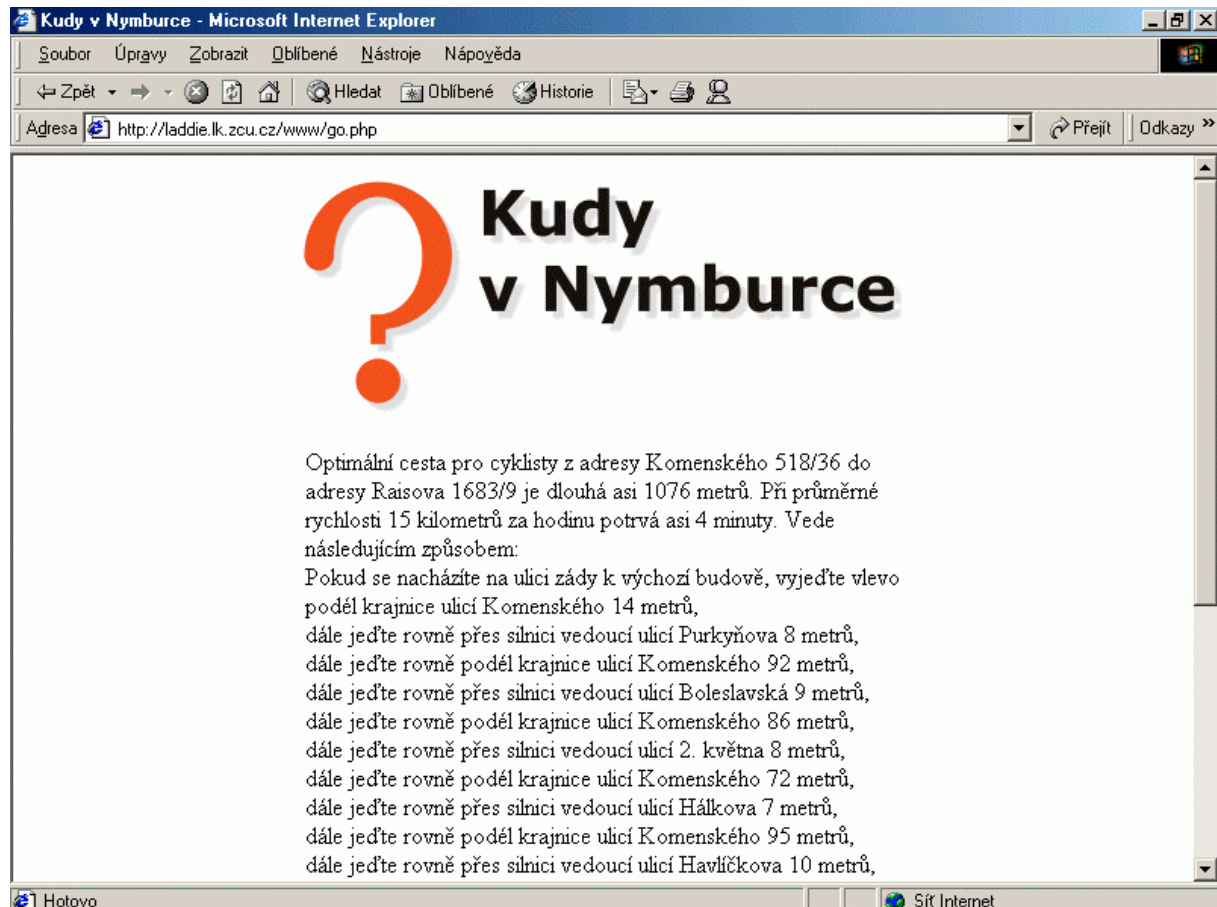
Chodec Cyklista Paraplegik

Síť Internet

Obrázek 10.2

Rychlost vyhledávání závisí na výpočetní kapacitě a zatížení serveru, na kterém je vyhledávač umístěn. Po nalezení cesty je internetový prohlížeč přesměrován na stránku obsahující informace popisující danou optimální cestu. V úvodu je zobrazena úplná adresa jak

výchozího tak cílového bodu a celková délka trasy. Rovněž je uvedena orientační doba pohybu vypočítaná z průměrné rychlosti. Pro chodce a osoby na invalidním vozíku je tato průměrná rychlost stanovena hodnotou 5 km/h, pro cyklisty je to 15 km/h. Následně jsou zobrazeny pokyny popisující vlastní pohyb z výchozí adresy do cílové adresy. Příklad vyhledané optimální cesty je na obrázku 10.3.



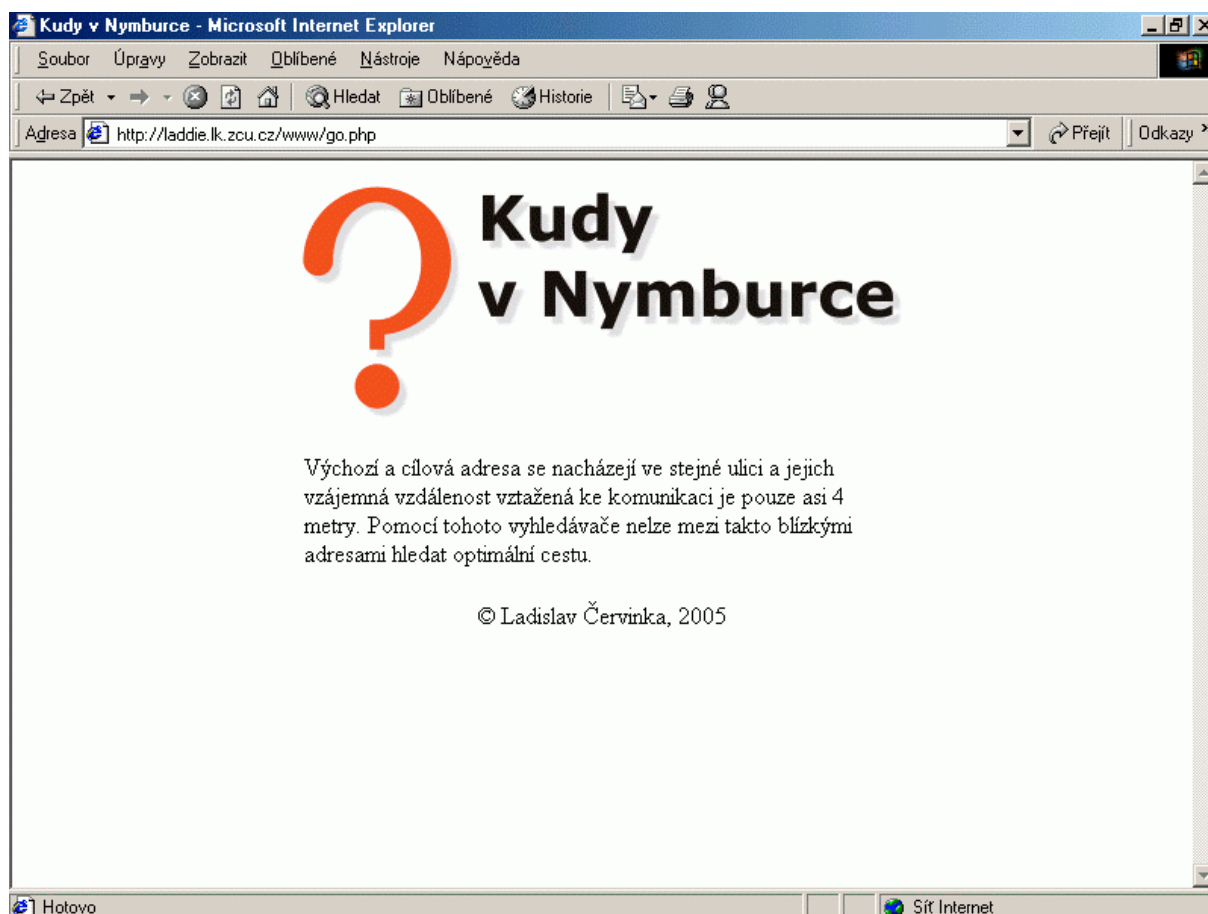
Obrázek 10.3

Pokud byly vstupní informace vyplněny nekorektně, k vyhledání cesty nedojde. Uživatel je v tomto případě upozorněn na vzniklé problémy. Mezi tyto problémy může patřit chyba v názvu ulice nebo nejednoznačnost v tomto údaji, dále může být nekorektní číslo popisné nebo číslo orientační.

Rovněž nelze vyhledávat optimální cestu mezi budovami, které se nacházejí ve stejném uličním bloku. Příklad takového stavu je zachycen na obrázku 10.4.

Na úvodní stránce internetového vyhledávače jsou v její dolní části uvedeny vedle odkazu *Vyhledávání s nápovědou* odkazy na další stránky. Odkaz *O vyhledávači* přesměruje prohlížeč na stránku zobrazující základní informace o vyhledávači a jeho účelu. Odkaz *Inteligentní vyhledávání* směřuje na obdobnou stránku poskytující rozšiřující funkce při vyhledávání.

První z rozšiřujících funkcí je možnost vypisovat optimální cestu souběžně pomocí polohy lomových bodů v souřadnicovém systému WGS 84. Tato funkce je nazvána *Vypisovat souřadnice GPS*. Souřadnice jsou vypisovány v pořadí zeměpisná šířka a zeměpisná délka. Hodnoty jsou zaokrouhlovány na setiny vteřiny.



Obrázek 10.4

Dále je možné uživatelsky definovat parametry zajišťující vedení optimální cesty jiným způsobem, pokud nejkratší cesta využívá komunikace opatřené nevhodným povrchem pro jízdu na jízdním kole. Takovým povrchem je například dláždění z kamenných kostek.

Příslušné koeficienty udávají kolikrát delší může být objížďka vyhýbající se úseku s nevhodným povrchem.

Pokud je například dán výchozí a cílový bod tak, že nejkratší cesta mezi nimi měří právě 1 km. Pokud celá cesta vede po vozovce s kamenným dlážděním a koeficient příslušející tomuto povrchu je 2, pak bude vyhledávač hledat nejkratší objížďku, která nebude delší než dvojnásobek délky úseku s nevhodným povrchem, to je v tomto případě 2 km. Pokud taková objížďka neexistuje, vypíše se původní cesta.

Na stránce inteligentního vyhledávání je možnost zadat konstanty pro povrch s kamenným dlážděním, s betonovými dlaždicemi, se silničními panely a pro nebezpečný povrch. Též je zde konstanta pro povrch ze dřeva. V Nymburce je takovým povrchem opatřena lávka přes Labe podél železničního mostu.

Tato funkce je určena pouze pro určování optimální cesty pro cyklisty. Pro sestavování optimální cesty chodců a paraplegiků nemá význam.

Dále lze nastavit průměrnou rychlost chůze a jízdy na kole. Pro osoby na invalidním vozíku se používá stejná hodnota rychlosti jako pro chůzi. Tato rychlost se následně používá při výpočtu předpokládané doby pohybu po optimální cestě.

Odkaz *Inteligentní vyhledávání s nápovědou* směřuje na stránku obsahující nápovědu k rozšiřujícím funkcím. Odkaz *Základní vyhledávání* směřuje na úvodní stránku.

Vyhledávač optimální cesty městem Nymburk založený na technologii WAP je postaven na stejných základech jako vyhledávač pro přístup přes Internet. Nabízí však omezené spektrum funkcí.

Pro přístup k tomuto vyhledávači je nutné mobilní zařízení podporující protokol WAP nebo počítač připojený k síti Internet, na kterém je nainstalován prohlížeč WAP stránek. Takovým prohlížečem je například Klondike WAP Browser 1.5.

Technologie WAP je primárně určena pro používání v mobilních zařízeních, jakými jsou především mobilní telefony GSM. Většina současně rozšířených modelů přitom WAP umožňuje. Před prvním použitím WAP v mobilním telefonu je zpravidla nutná konfigurace, která závisí na druhu přístroje a mobilním operátorovi.

WAP stránky se v mobilních telefonech zobrazují pomocí takzvaných mikroprohlížečů. Po otevření tohoto programu je třeba zadat adresu vyhledávače. První stránka obsahuje hlavní nabídku pěti odkazů na další stránky. První tři odkazy směřují na vyhledávání optimální cesty podle typu pohybu, pro který má být optimální cesta určena. Cesta může být optimalizována pro chodce, pro cyklisty a pro osoby na invalidním vozíku. Dále je na úvodní odkaz na stránku se stručnou nápovědou a na stránku se základními informacemi o vyhledávači a účelu jeho vzniku. Podoba hlavní nabídky zobrazená v simulátoru displeje mobilního telefonu je na obrázku 10.5. Vizualní vzhled stránky se může lišit v závislosti na typu mobilního telefonu.



Obrázek 10.5

Všechny stránky obsahují texty bez české diakritiky. Ačkoliv je v současnosti korektní zobrazování Češtiny v rámci technologie WAP možné, některé typy mobilních zařízení háčky a čárky nepodporují.

Po zvolení jednoho z prvních tří odkazů je prohlížeč přesměrován na stránku, jejíž vzhled je totožný, ať se jedná o vyhledávání optimální cesty pro chodce, cyklisty nebo paraplegiky. Stránka obsahuje celkem šest polí pro zadávání údajů o výchozí a cílové adrese, mezi kterými má být vyhledána optimální cesta. Úvodní tři pole jsou určena pro údaje o výchozí adrese. Postupně je třeba zadat název výchozí ulice nebo náměstí, číslo popisné a číslo orientační. Stejně jsou uspořádána i pole pro cílovou adresu. Pro zadávání adresy platí stejné podmínky jako u internetového vyhledávače. Horní část této stránky je na obrázku 10.6.

Vyhledávání se spouští pomocí dvou odkazů v dolní části stránky. Odkaz Vyhledat cestu s GPS spustí vyhledávání, jehož výsledkem budou navigační pokyny včetně výpisu polohy lomových bodů cesty v zeměpisných souřadnicích na elipsoidu WGS 84. Odkaz Vyhledat cestu vypíše příslušné informace o optimální cestě bez souřadnic. Poslední odkaz směřuje zpět na úvodní stránku s hlavní nabídkou.

Kudy v Nymburce

Vychozi adresa:
Ulice

Cislo popisne

Cislo orientacni

Cilova adresa:

Obrázek 10.6

Technologie WAP umožňuje zobrazování stránek nepřesahujících velikost přibližně 1200 znaků. Z tohoto důvodu je třeba v naprosté většině případů navigační instrukce rozdělit do několika stránek. Pro pohyb mezi jednotlivými stránkami slouží odkazy *Dalsi pokyny* a *Predchazejici pokyny*. Spodní část stránky instrukcí s příslušnými odkazy je na obrázku 10.7.

ulici Haikova 7 m,
dale jdete rovne po chodniku
ulici Komenskeho 95 m,
dale jdete rovne pres
krizovatku se silnici vedouci
ulici Havlickova 10 m,
dale jdete rovne po chodniku
ulici Komenskeho

[Dalsi pokyny](#)
[Predchazejici pokyny](#)

[Hlavni nabídka](#)

Obrázek 10.7

Informace o optimální cestě zobrazované v mobilním zařízení mají stejnou strukturu jako informace vyhledané pomocí Internetu. Pouze je vynechán odhad doby pohybu.

11 Závěr

Tato diplomová práce ukázala možnost jak vybudovat s minimálními softwarovými prostředky užitečnou a snadno přístupnou GIS aplikaci, která může potenciálně sloužit široké veřejnosti. Při tvorbě vyhledávače bylo mým prvořadým cílem vytvořit robustní aplikaci, která optimálně kombinuje jednoduchost a uživatelskou přívětivost a na druhé straně komplexní řešení dané problematiky.

Sofistikovaným základem aplikace je digitální model komunikační sítě města, který je v předkládané podobě vyhotoven přibližně na polovině rozlohy města Nymburk. Enormní časová náročnost vektorizace komunikací mi bohužel nedovolila dokončit celý model. Přesto si trůfám říci, že za modelem stojí spousta dobře odvedené práce. V předkládané podobě je tvořen více než 1700 vektorovými řetězci, přičemž každý z nich je opatřen sedmi atributy.

Určitým selháním je skutečnost, že se mi nepodařilo včas získat kompletní soubor adresních bodů Nymburka. Vzhledem k tomu vyhledávání funguje pouze mezi testovací sadou sta adres. Kdybych měl daná data k dispozici, mohl by vyhledávač fungovat mezi všemi adresami nacházejícími se na území Nymburka pokrytém digitálním modelem komunikací.

Pokud se zamyslím nad možnostmi rozšíření funkcí vyhledávače, rozhodně se na prvním místě nabízí výzva realizovat zobrazování optimální cesty v mapě a to alespoň ve verzi vyhledávače určené pro Internet.

Výzvou je též možnost zakomponovat do vyhledávače možnost používat městskou hromadnou dopravu.

Určitou slabinou, která si též žádá pozornost, je též velmi zjednodušené modelování vlivu dopravy na pohyb cyklistů. Kvalita výběru optimální cesty by rozhodně vzrostla, pokud by byla hustota provozu a s tím spojené doby při čekání při přejezdu křižovatek s hlavní silnicí modelovány v závislosti na době, kdy má být cesta optimální použita.

V modelu nebyl též vůbec uvažován vliv tvaru reliéfu. Zanedbání tohoto fenoménu bylo možné pouze vzhledem ke skutečnosti, že se Nymburk nachází v rovině. Obecně však toto zjednodušení aplikovat nelze.

Zanedbaným problémem je v neposlední řadě přívětivost systému na straně operátora udržujícího aktuálnost digitálního modelu. Přitom, pokud by byl tento vyhledávač nasazen v reálném provozu, rozhodně by bylo třeba tuto problematiku dořešit. V moderních městech se reálná síť komunikací mění s netušenou dynamikou a to jak v negativním smyslu vlivem nejrůznějších dopravních uzávěr ale i v pozitivním smyslu výstavbou nových koridorů pro cyklisty i chodce. Na tyto skutečnosti pak musí reagovat i model, který musí být průběžně aktualizován. Předložený systém lze však aktualizovat pouze přepracováním modelu, přičemž tato operace vyžaduje od operátora netriviální znalosti a postupy. Zároveň se jedná o práci, která je velmi citlivá na kvalitu provedení. Pokud je jen jeden vektor v modelu opatřen vadnými atributy, může tato skutečnost negativně ovlivnit vyhledávání optimální cesty v širokém okolí, v určitých případech může dojít k paralyzování celého systému.

Příloha

Nedílnou součástí této diplomové práce je datový kompaktní disk vytvořený v souladu s normou ISO9660 / Joliet obsahující soubory, které vznikly v rámci praktické realizace vyhledávače nebo jiným způsobem souvisí s touto diplomovou prací.

V kořenovém adresáři disku jsou obsaženy soubory:

Addresses.csv – soubor ve formátu CSV s daty databázové tabulky addresses

Addresses.txt – soubor vzorových adres

AddressSource.csv – soubor ve formátu CSV s informacemi o adresních bodech

DP.doc – soubor ve formátu Microsoft Word 97 obsahující text této diplomové práce

DP.pdf – soubor ve formátu Adobe Portable Document Format 1.2

Edges.csv – soubor ve formátu CSV s databázovou tabulkou edges

Nymburk.sql – soubor obsahující databázovou formu vektorového topologického modelu komunikační sítě Nymburka prostřednictvím posloupnosti SQL příkazů

Plain.sql – soubor obsahující posloupnost SQL příkazů pro tvorbu prázdných tabulek databáze nymburk

Points.csv – soubor ve formátu CSV s daty databázové tabulky points

Streets.csv – soubor ve formátu CSV s daty databázové tabulky streets

StreetSource.csv – soubor ve formátu CSV s názvy ulic a pojmenováními hran

TopoSource.vtx – soubor s vektorovou kresbou vektorového modelu komunikací v

Nymburce

VTX2CSV.c – zdrojový kód programu VTX2CSV.exe v jazyce ANSI C

VTX2CSV.exe – spustitelný soubor provádějící převod VTX souboru s vektorovým modelem komunikační sítě města Nymburk do tabulek ve formátu CSV

V adresáři WWW jsou obsaženy následující soubory uživatelského rozhraní vyhledávače ve verzi pro Internet:

about.htm – stránka se základními informacemi o vyhledávači

go.php – soubor s jádrem vyhledávače

help.htm – stránka s nápovědou k základnímu vyhledávání

index.htm – úvodní stránka vyhledávače

kudy.gif – rastrový soubor s logem vyhledávače

smart.htm – stránka s inteligentním vyhledáváním

smarthelp.htm – stránka s nápovědou k inteligentnímu vyhledávání

V adresáři WAP jsou obsaženy následující soubory uživatelského rozhraní vyhledávače ve verzi pro WAP:

about.wml – stránka se základními informacemi o vyhledávači

go.php – soubor s jádrem vyhledávače

help1.wml – první stránka s nápovědou k vyhledávání

help2.wml – první stránka s nápovědou k vyhledávání

index.wml – úvodní stránka vyhledávače

search1.php – stránka vyhledávání optimální cesty pro chodce

search2.php – stránka vyhledávání optimální cesty pro cyklisty

search3.php – stránka vyhledávání optimální cesty pro paraplegiky

Seznam použitých pramenů

- [1] Buchar Petr. Matematická kartografie 10. 2. vydání. Praha: Vydavatelství ČVUT, 2002. 203 s. ISBN 8001025349
- [2] Cimbálník Miloš – Mervart Miloš. Vyšší geodézie 1: Geometrická. 2. vydání. Praha: Vydavatelství ČVUT, 2002. 184 s. ISBN 8001025276
- [3] Demel Jiří. Grafy a jejich aplikace. 1. vydání. Praha: Academia, 2002. 257 s. ISBN 8020009906
- [4] Hába Milan. WAP: Tvorba stránek pro mobilní zařízení. 1. vydání. Praha: Computer Press, 2002. 181 s. ISBN 8072268147
- [5] Kolář Jan: Geografické informační systémy 10. 2. vydání. Praha: Vydavatelství ČVUT, 2003. 161 s. ISBN 8001026876
- [6] Lerdorf Rasmus – Tatroe Kevin. Programming PHP. Sebastopol: O'Reilly and Associates, 2002. 507 s. ISBN 1565926102
- [7] Mach Jakub. PHP pro úplné začátečníky. 2. vydání. Brno: Computer Press, 2003. 168 s. ISBN 8072268341
- [8] Moravec Dalibor. Kartografické a geoinformační modelování. 1. vydání. Praha: Karolinum, 2001. 242 s. ISBN 8024603381
- [9] Williams Hugh – Lane David. PHP a MySQL: Vytváříme webové databázové aplikace. 1. vydání. Praha: Computer Press. 530 s. ISBN 8072267604
- [10] Jedlička Karel. Úvod do geografických informačních systémů [online]. [cit. 2005-04-10] URL: <http://www.gis.zcu.cz/studium/ugi/elearning/index1.htm>
- [11] Kocan Marek. Databázový svět: S licencemi opatrně [online]. [cit. 2005-04-11] URL: <http://www.dbsvet.cz/view.php?cislocianku=2002061205>
- [12] Kosek Jiří. WAP [online]. [cit. 2005-03-08] URL: <http://www.kosek.cz/clanky/wapkurz/ar03s22.html>
- [13] Roleček Luděk: Jak pomocí formuláře zpracovat data v ASP [online]. [cit. 2005-05-12] URL: <http://interval.cz/clanek.asp?article=521>
- [14] Central European Data Agency: Adresní body ČR [online]. [cit. 2005-05-02] URL: <http://www.ceda.cz/article.asp?nDepartmentID=242&nArticleID=298&nLanguageID=1>
- [15] ČÚZK: Databáze trigonometrických a zhušťovacích bodů [online]. [cit. 2005-03-02] URL: <http://dataz.cuzk.cz/prehledky.php?a=3&z=15&t=24>
- [16] ČÚZK: Digitální ortofoto České republiky [online]. [cit. 2005-04-18] URL: http://www.cuzk.cz/Dokument.aspx?TYPPRAC=CUZK&PRARESKOD=0&MENUID=10222&AKCE=DOC:30-ZU_ORTOFOTO
- [17] Garmin: What is WAAS? [online]. [cit. 2005-03-02] URL: <http://www.garmin.com/aboutGPS/waas.html>