

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

## KNIHA JÍZD PRO IPHONE A WINDOWS MOBILE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

FILIP HUBÍK

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

# KNIHA JÍZD PRO IPHONE A WINDOWS MOBILE

ELECTRONIC LOG BOOK FOR IPHONE AND WINDOWS MOBILE

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

FILIP HUBÍK

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. PAVEL BARTOŠ

BRNO 2010

## **Abstrakt**

Tato bakalářská práce se zabývá návrhem a implementací elektronické on-line evidence jízd pro mobilní přístroje Apple iPhone a přístroje podporující operační systém Windows Mobile. Zaměřuje se na systém GPS zařízení i globální systém navigace obecně. Je zde podrobně rozebrána problematika klientských aplikací obou platforem. Dále je navrhována a v rámci práce implementována serverová aplikace, pracující s oběma klientskými. Výsledkem práce systému mohou být například podklady pro účetnictví nebo vizualice cest firemními vozidly, jinak řečeno jednoduchá kniha jízd.

## **Abstract**

This bachelor's thesis designs and implements electronic on-line log book for mobile devices Apple iPhone and Windows Mobile operating systems, focusing on GPS and navigation systems. There is analyzed in detail the issue of applications both platforms. It is designed and implemented the work of a server application too, working with both clients. The result of this work system may be such as accounting or data for visualization journeys corporate vehicles - a simple electronic log book.

## **Klíčová slova**

GPS, Kniha jízd, Evidence jízd, Apple, iPhone, Windows Mobile, NAVSTAR, GLONASS, GALILEO

## **Keywords**

GPS, Log book, Apple, iPhone, Windows Mobile, NAVSTAR, GLONASS, GALILEO

## **Citace**

Filip Hubík: Kniha jízd pro iPhone a Windows mobile, bakalářská práce, Brno, FIT VUT v Brně, 2010

# Kniha jízd pro iPhone a Windows mobile

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana ing.Pavla Bartoše

.....

Filip Hubík  
18. května 2010

## Poděkování

Rád bych poděkoval vedoucímu této práce panu Ing. Pavlu Bartošovi za spolupráci a trpělivost.

© Filip Hubík, 2010.

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



# Obsah

<b>1</b>	<b>Úvod</b>	<b>4</b>
<b>2</b>	<b>Legislativní rámec a zaměření práce</b>	<b>5</b>
<b>3</b>	<b>Systém globální navigace</b>	<b>6</b>
3.1	Historie	6
3.2	Složení systému GPS	7
3.3	Základní princip	8
3.4	Nadmořská výška	10
3.5	Metody určování polohy GPS	10
3.6	Zpracování signálů	11
3.7	Navigační zpráva	11
3.8	Inicializace GPS přijmače	11
3.9	Omezení služby GPS a korekce	12
3.10	Komunikační protokol a přenos dat	14
3.11	Další využití GPS v praxi	15
<b>4</b>	<b>Mobilní zařízení Apple iPhone a Windows Mobile</b>	<b>16</b>
4.1	Úvod	16
4.2	Apple iPhone	16
4.3	Windows Mobile	17
<b>5</b>	<b>Klientská aplikace pro Windows Mobile 6.5</b>	<b>19</b>
5.1	Návrh aplikace	19
5.1.1	Uživatelské rozhraní	19
5.1.2	Komunikační rozhraní se zdrojem GPS dat	19
5.1.3	Internetové připojení	20
5.1.4	Komunikace se serverem	20
5.2	Vývojové prostředí	20
5.2.1	Visual Studio	21
5.2.2	Windows Mobile 6 SDK	21
5.2.3	Microsoft ActiveSync a Windows Mobile Device Center	21
5.2.4	Windows Mobile 6.5 Developer Tool Kit	22
5.2.5	Windows Mobile Emulator Images	22
5.2.6	Microsoft Virtual PC	22
5.2.7	Microsoft .NET Framework	22
5.3	Implementace	22
5.3.1	Implementační část pracující s GPS	23

5.4	Synchronizace a komunikace . . . . .	24
5.5	Distribuce aplikace . . . . .	24
5.6	Testování aplikace . . . . .	24
<b>6</b>	<b>Klientská aplikace pro Apple iPhone</b>	<b>26</b>
6.1	Návrh aplikace . . . . .	26
6.1.1	Uživatelské rozhraní . . . . .	26
6.1.2	Komunikační rozhraní se zdrojem GPS dat . . . . .	26
6.1.3	Internetové připojení . . . . .	26
6.1.4	Komunikace se serverem . . . . .	27
6.2	Vývojové prostředí . . . . .	27
6.2.1	Apple iPhone SDK . . . . .	27
6.3	Implementace . . . . .	28
6.3.1	GPS rozhraní . . . . .	29
6.3.2	Grafické rozhraní . . . . .	30
6.3.3	Synchronizace . . . . .	30
6.4	Distribuce a testování aplikace . . . . .	30
<b>7</b>	<b>Uchování a zpracování dat</b>	<b>33</b>
<b>8</b>	<b>Závěr</b>	<b>35</b>
<b>A</b>	<b>Vizualizovaná data</b>	<b>38</b>
<b>B</b>	<b>Obsah CD</b>	<b>39</b>

# Seznam obrázků

3.1	Princip určení polohy bodu v 1D prostoru . . . . .	9
3.2	Princip určení polohy v 2D prostoru . . . . .	9
3.3	Nevhodné a) a ideální b) rozpoložení družic pro měření pozice bodu . . . . .	13
3.4	Chyba měření v závislosti na vzájemné poloze družic . . . . .	14
3.5	Základní popis obecné zprávy protokolu NMEA . . . . .	15
4.1	Apple iPhone 3GS s technickou specifikací[2] . . . . .	17
4.2	HTC Touch2 s OS Windows Mobile 6.5 a technickou specifikací[1] . . . . .	18
5.1	Přehled základních paušálních tarifů největších českých operátorů <sup>1</sup> . . . . .	20
5.2	Diagram knihoven a tříd . . . . .	23
5.3	Finální klientská aplikace pro zařízení s OS Windows Mobile 6.5 . . . . .	25
6.1	Diagram tříd jádra aplikace pro iPhone . . . . .	28
6.2	Kompletní rozbor vývojářských programů pro Apple iPhone[4] <sup>4</sup> . . . . .	31
6.3	Výsledná aplikace, ukázka chodu programu a chyby v běhu . . . . .	32
6.4	Výsledná aplikace, ukázka nastavení aplikace a spouštění . . . . .	32
A.1	Příloha č.1 - Vizualizace dat pomocí služby GPS Visualizer[3] . . . . .	38

# Kapitola 1

## Úvod

Online elektronická kniha jízd je náhrada tradiční knihy jízd, která slouží pro zaznamenávání a uchovávání cest dopravními prostředky. Je často používána jak soukromníky, tak firmami pro monitorování pohybu svých zaměstnanců na cestách. Smyslem této bakalářské práce je vytvořit aplikace pro konkurenční mobilní zařízení Apple iPhone a Windows Mobile, které budou ve spolupráci s databázovým serverem jednoduchou elektronickou knihu jízd implementovat. Využití těchto zařízení pro tyto účely je více než vhodné, obě obsahují hardware pro získávání GPS souřadnic, navíc jejich cena na trhu neustále klesá. Výhoda oproti vestavěným zařízením, například v automobilech, spočívá v jejich snadné manipulaci a v možnosti zařízení využívat jinak po ukončení jeho použití jako zdroje dat pro knihu jízd. Na závěr práce budou dále rozebrány možnosti rozšíření této jednoduché elektronické knihy jízd, jako je vizualizace přijatých zaznamenaných dat.

## Kapitola 2

# Legislativní rámec a zaměření práce

Ze zákona č. 586/1992 Sb. o daních z příjmů musí podnikatel vést evidenci (knihu) služebních jízd nebo si může odepisovat paušální výdaj na dopravu[10] ve výši 5000Kč měsíčně jako náklad v případě používání vozidla výhradně na služební cesty, nebo 4000Kč v případě používaného vozidla i na soukromé cesty a nesmí takto využívat více než 3 vozidla. Zákonný pokyn D-300, odstavec 24 zní takto:

Pro účely uplatnění daňových výdajů na pohonné hmoty, pokud správce daně nestanoví jinak, vede poplatník evidenci jízd tak, aby takto vynaložené výdaje (náklady) mohl prokázat (§ 24 odst. 1 zákona). V evidenci jízd jsou uváděny minimálně tyto údaje: datum jízdy, cíl jízdy, účel jízdy, ujeté km. Dále poplatník vede údaje o typu vozidla, registrační značce (státní poznávací značce), stavu ujetých km k 1. lednu (případně k datu zahájení činnosti nebo zahájení používání vozidla) a k 31. prosinci kalendářního roku (případně k datu ukončení činnosti nebo ukončení používání vozidla).[11]

Psaní knihy jízd je pro řadu podnikatelů nepříjemná a zdržující záležitost. Také střední a velké firmy s více než třemi firemními auty nemohou využívat paušálu, tedy mají povinnost psát evidenci jízd pro všechny služební vozidla. Další cílovou skupinou této práce jsou podnikatelé, kteří chtějí znát přesně své náklady na provoz služebních vozidel. Pokud používají paušál, mohou tímto způsobem pohodlně zjistit, zda je pro ně vedení knihy jízd výhodnější než paušál. Pro všechny tyto případy je určen software navrhovaný v této práci.

Konečným produktem navrhovaného systému může být například sada vytištěných dokumentů o cestách vozidel v daném časovém období se všemi zmiňovanými parametry, které může sloužit jako evidence jízd předávaná finančnímu úřadu. Může být použita také k dohledání či zpětným kontrolám cest, například v papírové knize jízd. Jako rozšíření může sloužit i jako vizuální reprezentace dat o cestách formou map, nebo jako plugin k některému softwaru zabývajícím se účetnictvím.

## Kapitola 3

# System globální navigace

### 3.1 Historie

Pokud chceme mluvit o dnešním globálním družicovém systému, angl. Global Positioning System[8], dále jen "GPS", jako takovém, měli bychom si přiblížit jeho počátky a podněty jeho vzniku.

Ty sahají do začátku druhé poloviny 20.století, kdy se tehdejší SSSR podařilo vyslat družici Sputnik na oběžnou dráhu Země. Již tehdy zúčastnění vědci zjistili, že při pohybu družice Sputnik kolem Země klesá a roste síla jejího signálu v závislosti na poloze. Toto zjištění pomalu otevíralo tehdy netušené možnosti, ale také nové pole konkurenčního boje dvou světových velmocí v době Studené války.

V roce 1973 padlo v USA rozhodnutí o vybudování systému založenému na technologii GPS výhradně pro vojenské účely. Od roku 1978 bylo po řádném testování započato vysílání družic systému GPS na oběžnou dráhu. Celkem bylo USA vysláno 11 družic s kódovým označením NAVSTAR - Navigation System with Timing And Ranging[13]. V roce 1980 byly spuštěny atomové hodiny obsažené ve vypuštěných družicích. Tyto hodiny jsou nepřesnější dostupný přístroj pro měření času, jejich přesnost se pohybuje v řádech nanosekund. Díky nim je možné provádět vysoce přesné výpočty polohy v přijímačích. Od roku 1983 není již systém NAVSTAR doménou vojenskou, ale přechází i do komerčního sektoru. Do této doby sloužil hlavně pro monitorování území SSSR, či jejich zbraňového a jaderného arzenálu. Kapacita systému již dále nedostačovala a funkcionalita byla zatím značně omezená. V roce 1988 se začal počet družic navyšovat na 24. Tento systém byl plně uveden do provozu až v dubnu 1995 v kompletním počtu 24 družic a v této konfiguraci pracuje i dnes. Družice obíhají ve výšce 20 000km, což je téměř o 1000km výše než družice ruského GPS programu.

V průběhu vývoje systému GPS v USA docházelo také k problematickým událostem, které se často stávaly příčinou zpoždění či odkladů prací na této technologii. Jednalo se například o havárii raketoplánu Challenger nebo korejského letadla Airline flight 007 nad ruským územím. Tyto události byly přisuzovány chybám v systému GPS. V průběhu války v Perském zálivu 1990–1991 došlo k výraznému omezení této služby pro veřejnost, protože byla více využívána armádou.

SSSR mělo také vlastní paralelní program systému GPS, a to GLONASS (Globalnaja Navigacionnaja Sputnik Systema[8]). Vývoj tohoto systému schválila vláda SSSR v roce 1976[16]. V roce 1982 byly vypuštěny první družice URAGAN na oběžnou dráhu, vypouštění pokračovalo až do roku 1991 s počtem dvanácti provozních družic. Systém se stal funkčním ale až v roce 1995. Bylo jej ale potřeba udržovat (životnost družice je kolem 3 let). Vzhledem ke špatné ekonomické situaci počet provozuschopných družic klesal, až se

system stal nepoužitelným. V roce 2001 byl Ruskou vládou schválen plán na zprovoznění systému se všemi 24 družicemi potřebnými pro úplné pokrytí Země, který měl být dokončen v roce 2011. V roce 2006 a 2007 vynaložila Ruská vláda na tyto účely 561 milionů dolarů. V současné době je provozuschopných 10 družic.

GPS družice vysílané Ruskem prošli jistým vývojem hlavně v oblasti životnosti a odolnosti. První typ družice URAGAN měl životnost pouhé tři roky. Následovaly typy URAGAN-M s životností 7 let a URAGAN-K s životností až 12 let. Poslední uvedené měla vynést na oběžnou dráhu raketa Sojuz-U, čímž se měly snížit náklady na tuto operaci.

System GLONASS byl však původně také určen k vojenským účelům. V době Rusko-Čečenského konfliktu byl používán pro monitorování infrastruktury a pro mapování vojenských cílů. V současné době ve stavu desíti provozuschopných družic je pokrytí tohoto systému velmi malé, 33% v Rusku a 27.2% po světě, proto jeho využití v komerční sféře není rozšířené. Tyto družice obíhají ve výšce 19 100km nad Zemí[16].

Evropská unie však dnes buduje svůj vlastní alternativní program a jeví snahu stát se nezávislou na programech USA či Ruska. Tento systém oproti americkému systému NAVSTAR nevzniká jako armádní, je určen primárně pro komerční účely a nese název GALILEO. GALILEO je vyvíjeno prostřednictvím evropských států a spadá pod správu Evropského parlamentu a Evropské vesmírné agentury, která v této době sdružuje 18 členských států (1.1.2010) včetně České Republiky. Na návrhu tohoto systému se v roce 1999 podílely 4 evropské státy: Německo, Francie, Itálie a Spojené Království. Projekt od roku 2001 trápily problémy jak s financováním, tak se snahou USA tento projekt zrušit a udržet si monopol. I přes tyto snahy byly pro projekt do roku 2005 vyhrazeny prostředky ve výši 1,1 bilionu EUR a započalo vysílání družic na oběžnou dráhu. Do roku 2007 stál projekt ovšem 3 biliony EUR[14]. V roce 2007 byl Evropskou unií schválen plán na oficiální spuštění systému v roce 2013, tento termín byl ale odložen na rok 2014. V konečné fázi by měl být kosmický segment systému GALILEO tvořen 30 družicemi. Tyto družice se mají pohybovat ve výšce 23222 km s délkou životnosti větší než 12 let[14].

## 3.2 Složení systému GPS

Jako každý celek i GPS se skládá z celků menších. Nebudeme se již dále zabývat rozбором rozdílů této technologie mezi jednotlivými státy, ale popisem základních principů technologie jako takové. System GPS se dle[8] skládá ze 3 základních částí:

- Uživatelský segment
- Kosmický segment
- Řídící segment

Uživatelský segment je tvořen přijímači ve vlastnictví koncových uživatelů.

Kosmický segment je tvořen družicemi na oběžné dráze Země. Tyto družice v přesně daném rozpoložení obíhají kolem Země tak, aby pokryly co největší plochu. Je spočítáno, že pro úplné pokrytí zemského povrchu je potřeba, aby bylo ve stejném čase na oběžné dráze 24 družic. Tyto družice mají přesně dané dráhy, výšky, sklony a rozmístění. Každá družice má svůj vlastní kód, kterým se identifikuje.

Řídící segment je zpravidla reprezentován stanicemi na zemském povrchu. Tento segment zajišťuje komunikaci a synchronizaci uživatelského segmentu se segmentem kosmickým. Dále se stará o korekce drah družic a jejich dalších parametrů, důležitých pro správnou



funkčnost celého systému. Rovněž plní důležitou funkci korekce a synchronizace atomových hodin družic, údržby a řízení vůbec. Kompenzuje různé nežádoucí vlivy, se kterými družice přichází na oběžné dráze do styku. Jsou to například sluneční větry nebo slapové síly. Pokud je některá družice vyhodnocena jako nefunkční, je nutné ji nahradit družicí záložní. Rozlišujeme dva typy družic podle aktivování:

- Hlavní družice - Je aktivována od doby svého vypuštění na oběžnou dráhu
- Záložní družice - Je aktivována pouze pokud se některá hlavní družice porouchá a nahradí její místo

V řídicím segmentu rozlišujeme 3 typy stanic provádějící některé z výše uvedených úkonů[8]:

- **Monitorovací stanice** Jejich účelem je monitorování stavu družic a distribuce takto získaných dat k řídicím stanicím.
- **Hlavní řídicí stanice** Hlavní stanice zpracovává signály z ostatních stanic, provádí důležité korekce a dříve uvedené operace, tyto data posílá stanicím pro komunikaci s družicemi.
- **Stanice pro komunikaci s družicemi** Slouží pro ovládání a odesílání koncových upravených dat na družice.

### 3.3 Základní princip

Pro určení polohy mezi dvěma body na přímce je potřeba dvou zdrojů signálu. Tyto zdroje proti sobě vysílají signály a na přímce mezi nimi je umístěn pozorovatel. Pozorovatel se nachází na přímce procházející těmito zdroji. Aby pozorovatel mohl určit svou polohu vůči zdrojům musí vypočítat rozdíl časů příchodů signálů. Z tohoto rozdílu je snadné se dopočítat pomocí jednoduchého vzorce 3.1 a znalosti rychlosti světla, tj. přibližně 300000 km/s, vzdálenost zdroje signálu, jehož signál přišel k GPS zařízení později. Situace je znázorněna na obrázku 3.1. Tímto vlastně určujeme pozici bodu v jednorozměrné souřadnici, samozřejmě při znalosti délky dráhy mezi oběma zdroji. Informaci o délce dráhy v systému GPS nepřímo dostává přijímač od družic, se kterými právě komunikuje. Družice zasílají přijímači data o své pozici na oběžné dráze v předem daných synchronizovaných intervalech. Spolu s těmito daty je také zasílán aktuální atomový čas družice, od kterého pak GPS přijímač počítá potřebné rozdíly.

$$v = \frac{s}{t} \tag{3.1}$$

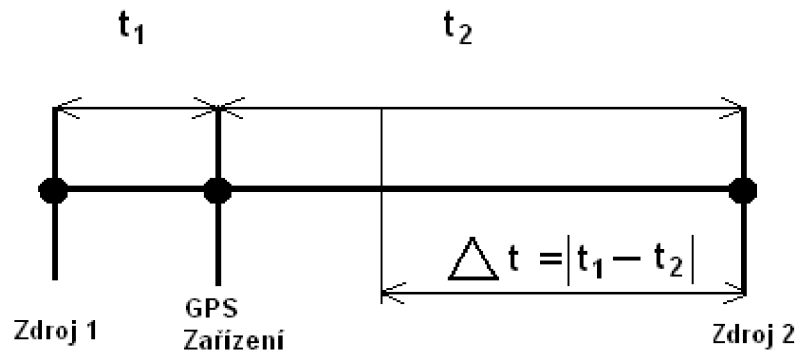
v = rychlost pohybu signálu (300000 km/s)

s = dráha uražená signálem za čas

t = čas, po který signál putuje k přijímači

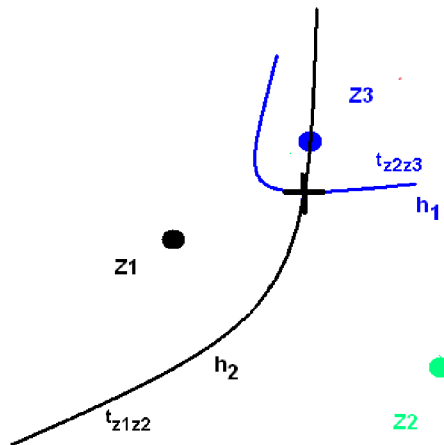
Hledaný reálný bod však neleží na přímce mezi dvěma vysílači a tudíž je potřeba nějakým způsobem určit jeho polohu v prostoru. V prostoru již může být rozdíl časů konstantní v různých bodech. Tyto body leží na hyperbole, která má konstantní vzdálenosti od obou ohnisek, které reprezentují oba zdroje. Pro určení přesného bodu na této hyperbole je potřeba zavést třetí zdroj signálu, který pozici hledaného bodu již řádně vymezí, viz obrázek





Obrázek 3.1: Princip určení polohy bodu v 1D prostoru

**3.2** . Pokud tedy máme tři rozdíly časů, můžeme určit tři hyperboly. Pro výpočet pozice hledaného bodu však stačí dva rozdíly časů. Na obrázku č.2 vidíme dvě hyperboly, které jsou funkcí rozdílu času příchodu signálu od jejich ohnisek (zdrojů). Třetí hyperbolu pro tento výpočet potřebovat nebudeme. Například mějme rozdíl časů mezi zdrojem Z1 a Z2 označen  $\Delta t_{Z1Z2}$ , platí pro  $t_{Z1} < t_{Z2}$ . Dále rozdíl  $\Delta t_{Z2Z3}$  mezi zdroji Z1 a Z3, platí pro  $t_{Z1} > t_{Z3}$ . Přijímač signálů z těchto zdrojů bude ležet na průsečíku hyperbol h1 a h2. Je možno dopočítat se správného výsledku i pomocí kombinace jiných dvou kombinací hyperbol. Hledaný bod však vždy leží na jejich průsečíku.



Obrázek 3.2: Princip určení polohy v 2D prostoru

Tímto jsme získali bod v dvojrozměrném prostoru. GPS ale pracuje s trojrozměrným prostorem. Třetím rozměrem je nadmořská výška bodu, která je nutná pro správné zjištění polohy. Proto je potřeba zachytit signál z dalšího, čtvrtého zdroje, který tuto veličinu určí. Bez informace ze všech 4 zdrojů není možné určit s maximální přesností, kterou technologie GPS umožňuje, polohu bodu, pouze jeho dvojrozměrné parametry, tzn. souřadnice x a y.

Čtvrtý zdroj rovněž slouží jako synchronizace času.

Výše uvedená metoda výpočtu souřadnic se jmenuje Time Difference of Arrival a na jejím principu funguje systém GPS.

### 3.4 Nadmořská výška

Nadmořská výška naměřená systémem GPS je vztažena k matematické reprezentaci Země, neboli k ploše tzv. elipsoidu, přepočtená ze souřadnic WGS-84[8]. Tato nadmořská výška ale není přesná. Na její nepřesnosti se podílejí faktory jako zakřivení Země, příliv, odliv a hladina oceánů. V případě elipsoidu se jedná hlavně o faktor zkreslení zeměpisnými parametry Země. V praxi se ale nadmořská výška vztahuje k geoidu, který je pro účely měření přesnější. Tento geoid se odchyluje od referenčního elipsoidu a je potřeba určit rozdíl výšky obou těchto těles. Toto lze docílit korekcí pomocí známé značky nadmořské výšky nějakého místa. Uvedený rozdíl se v České republice pohybuje v rozmezí 42,5 až 47 m[8]. Přesnost určování výšky u kapesních přijímačů se pohybuje v rozmezí 15 až 40 metrů[8].

### 3.5 Metody určování polohy GPS

Systémy GPS jsou konstruovány jako systémy pasivní, tj. takové, které nevysílají žádná data do svého okolí, ale slouží pouze a výhradně jako přijímače. Toto provedení je hlavně pro vojenské účely, kde by vysílání signálů mohlo riskovat prozrazení. Určování pozice na zemském povrchu se provádí pomocí techniky protínání, což znamená, že přijímač určuje svou polohu podle signálů přijatých od dostupných družic a z nich provede výpočet polohy. Pro vzdálenosti mezi přijímačem a družicí se používají následující měření:

- Kódová měření
- Fázová měření
- Měření využívající Dopplerova jevu

Kódová měření využívají speciální druhy kódů, tzv. dálkoměrné kódy[8]. Tyto kódy jsou vysílány družicí a obsahují časovou značku generovanou pomocí vnitřních atomových hodin družice. Kód obsahující tuto značku je přijat přijímačem a z rozdílu hodnoty časové značky, tj. doby odeslání signálu, a času přijetí přijímačem určí požadovanou vzdálenost. Toto je prováděno pro každou dostupnou družici pro zpřesnění určení polohy místa.

Fázová měření jsou založena na měření pomocí zjišťování parametrů nosných vln při komunikaci družice s přijímačem. Počítá se počet celých vln při komunikaci při znalosti délky jedné vlny. Zbývající necelou část vlny lze pomocí aktuálního stavu její amplitudy dopočítat. Stejně jako u předchozího typu měření i zde roste přesnost měření s počtem signálů od různých dostupných družic. Tato měření jsou výpočetně náročnější, hlavně z důvodů složitějších výpočtů hodnot signálů, ale přesnější než kódová měření.

Měření využívající Dopplerova jevu využívají uvedený fyzikální jev. Dopplerův jev popisuje změnu frekvence a vlnové délky přijímaného oproti vysílanému signálu, pokud se přijímač nebo vysílač pohybuje[8]. Využívá například v silničních radarech, kde vyhodnocuje rychlost auta přibližujícího se nebo vzdalujícího se od vysílače signálu. Tento jev byl objeven v roce 1842 rakouským fyzikem Johannem Christianem Dopplerem. Obecná rovnice jevu[6]:

$$f' = f \frac{v \pm vd}{v \pm vz} \quad (3.2)$$

kde znaménka závisí na pohybu vysílače či přijímače.

V běžně používaných přijímačích pro domácí použití se využívá metody kódových měření. V pokročilejších přístrojích, jako jsou nástroje pro přesné geodetické měření, se používá měření fázového. Třetí způsob měření se nevyužívá přímo pro určování polohy, ale pro určování vzdálenosti mezi dvěma body, což je jedna ze nejdůležitějších dílčích součástí měření.

Pro spolehlivé zjištění souřadnic je však potřeba mít na paměti, že GPS přijímač musí přijímat data od minimálně 4 družic. 3 družice jsou využity pro potřeby výpočtů přesné polohy, poslední je využita pro potřeby časové synchronizace a určení nadmořské výšky. Při nižším počtu než 4 dostupných družic mohou hlásit přijímače chybové hlášení či signalizovat nedostupnost služby.

### 3.6 Zpracování signálů

Existují dva druhy signálů vysílaných družicemi k přijímačím stanicím. Jedná se o signál L1 na nosné frekvenci 1575,42 MHz a signál L2 na nosné frekvenci 1227,60 Mhz. Každý z těchto signálů je kombinací nosné vlny, dálkoměrného kódu a navigační zprávy[8] a je kódován. První frekvence je využívána veřejnými GPS přijímači, avšak frekvence L2 je zakódovaná neveřejným kódem a využívají ji pouze speciální přijímače.

První frekvence může být taktéž omezena nebo dokonce zablokována provozovatelem systému GPS, tj. Ministerstvem obrany USA. Může se tak stát v rámci zvýšené potřeby využití a vytížení tohoto systému, např. při vojenském konfliktu. Systém tehdy přejde do stavu tzv. "Selective availability mode", kdy se stane pro osobní GPS přijímače nepoužitelným, využívat jej budou moci jen vojenské přijímače vybavené příslušným dekodérem. Spolu se signálem je přenášena i tzv. navigační zpráva.

### 3.7 Navigační zpráva

Navigační zpráva je zasílána družicí k přijímači. Pomocí ní přijímač zjišťuje polohu družice. Aby mohla družice tuto navigační zprávu - informaci o své poloze - odeslat, musí nejprve tuto polohu vypočítat z jí dostupných údajů, které jsou závislé na trajektorii družice a dalších parametrech. Zpráva obsahuje nejen tyto informace, ale další doplňkové informace o svém stavu a stavu komponent, jako je například signalizace nefunkčnosti nebo poškození. Vlastní GPS přijímač provádí rozbor navigační zprávy a provádí korekce, které opravují možné chyby či odchylky při průchodu ionosférou.

Souhrnným údajům o parametrech družice, např. poloze, obsaženým v navigační zprávě se říká efemerid. Tyto údaje popisují predikované dráhy družic. Dráhy družic lze predikovat výpočtem, vycházejícím z jejich dobře předvídatelného pohybu po oběžné dráze.

### 3.8 Inicializace GPS přijímače

Po zapnutí GPS přijímače dojde nejprve k inicializaci přístroje. Ten si vyžádá z družice tzv. almanach, což je soubor informací o ostatních družicích. Almanach je 37500 bitů velký, a jeho přenos by trval rychlostí 30bps asi 12,5 minuty. To je často velmi nepříjemné při zapínání přístroje, proto je tato doba zkrácena, a to tak, že je přijata pouze ta část almanachu, která obsahuje informace o družicích v dosahu. Rychlost příjmu této zprávy je také ovlivněna okolím. Je doporučeno provést příjem almanachu na vysoce položených místech,

nikoliv ve městě, kde může být signál nedostatečně silný nebo nekvalitní v důsledku omezení výškovými budovami.

### 3.9 Omezení služby GPS a korekce

Schopnost příjmu GPS signálu přijímačem závisí na mnoha faktorech a není samozřejmostí, že bude signál dostupný po celém povrchu Země, v budovách apod. Z nejdůležitějších faktorů ovlivňujících kvalitu služby je potřeba uvést tyto:

- Počet dostupných družic
- Úroveň signálu družic
- Typ metody měření
- Možnost odražení signálu

V budovách je potřeba mít na paměti, že zdi budovy jsou pravděpodobně železobetonové, což může značně omezovat signál a elektromagnetické vlnění vůbec. Pokud bude nedostatečný, nemusí dojít k přesnému získání souřadnic. Nejlépe je tedy získávat GPS souřadnice na místech vysoko položených či volných prostranstvích. Pro největší přesnost je důležité přijímat signál z co největšího počtu družic, minimum je však 4[8]. Při příjmu z tří družic nedojde k ověření nadmořské výšky. Vzhledem k tomu, že určení GPS pozice probíhá v trojrozměrném prostoru, je tato informace potřeba.

Pro jednoznačné zjištění kvality přijatých GPS dat se využívá výpočtu odchyly přesnosti DOP (Dilution of Precision). Tento výpočet z relativní polohy družice vzhledem k ostatním dokáže určit přesnost a kvalitu dat. Čím nižší hodnota, tím větší kvalita dat, avšak běžná hodnota se pohybuje okolo čísla 5. Nejnižší hodnota je 1. Pokud je vyžadována největší přesnost přijímače při nějakém měření osobním GPS přijímačem, je možné vystihnout moment nejvhodnějšího relativního uspořádání dostupných družic a provést dané měření v tomto momentu. Pro tento účel lze využít specializovaný software. Odchyly DOP se skládá z několika dílčích faktorů[12].

- Horizontální odchylyka
- Vertikální odchylyka
- Časová odchylyka
- Geometrická odchylyka
- Relativní odchylyka

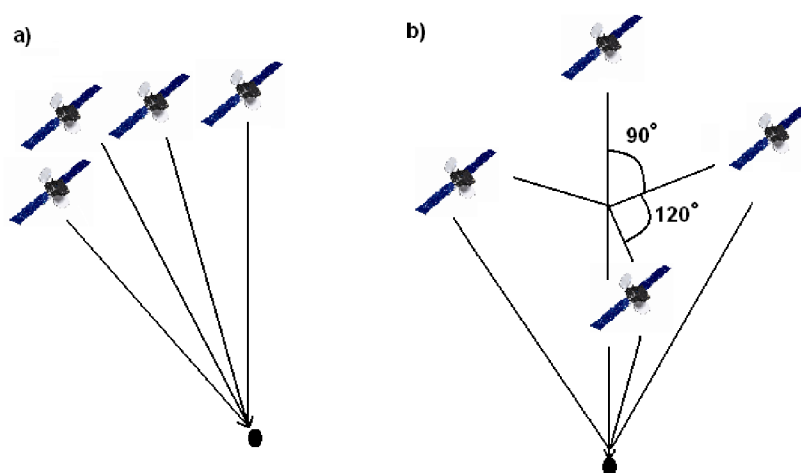
Kombinaci horizontální a vertikální DOP se říká PDOP - poziční odchylyka (position dilution of precision). Časová odchylyka TDOP určuje míru nepřesnosti časového systému. Geometrická odchylyka je kombinací všech dříve uvedených. Výpočet relativní odchylyky RDOP zahrnuje počet GPS přijímačů ku počtu družic, které je dokáže obsluhovat, dále konfiguraci geometrie a míru zorného pole družic. Ideální rozpoložení družic s nejmenším DOP je takové, kdy je jedna družice přímo nad měřeným bodem a ostatní 120 stupňů od sebe v rámci horizontálního roviny. Faktor DOP lze přibližně aproximovat upravením vzorce pro výpočet nepřesnosti měření v závislosti na DOP z [12] na vzorec 3.4:

$$\sigma = DOP * \sigma_0 \quad (3.3)$$

$$DOP = \frac{\sigma}{\sigma_0} \quad (3.4)$$

Kde DOP = odchylka přesnosti  
 $\sigma$  = indikátor nepřesnosti pozice  
 $\sigma_0$  = indikátor nepřesnosti měření

Rozdíl v poziční odchylce pro krajní případy je znázorněn na obrázku 3.3. Obrázek a) reprezentuje nevhodné umístění družic pro měření polohy, obrázek b) však nejvhodnější umístění. Proč je rozpoložení družic b) přesnější než a) je znázorněno na obrázku 3.4.



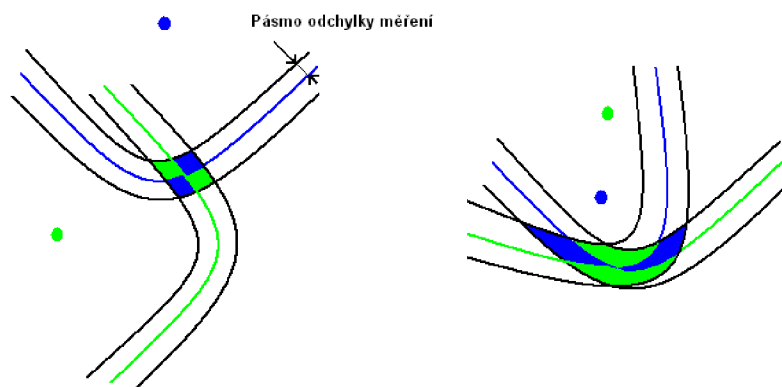
Obrázek 3.3: Nevhodné a) a ideální b) rozpoložení družic pro měření pozice bodu

Vždy při měření jako takovém vzniká určitá chyba měření. Stejně tak je to i u měření polohy pomocí GPS. Tuto chybu je potřeba určitým způsobem minimalizovat. Z obrázku je patrné, že chyba, která vzniká při každém zjišťování pozice je tím větší, čím jsou zdroje blíže sobě. Hyperboly reprezentující funkce konstantního rozdílu časů příchodu (kap 3.3) svírají menší úhel na obrázku 3.4 b). Černými čarami je na obrázku označena pomyslná mez chyby měření. V tomto obrázku pokrývá možnost chyby měření, reprezentovaná průnikem obou hyperbol, větší plochu než je tomu v obrázku 3.4 a). Proto je měření v takovém rozpoložení zdrojů (GPS družic) mnohem přesnější.

Je potřeba také brát v potaz možnosti odražených signálů. Některé stavby dokáží odrazet signál a tím mást přijímače, protože odrazem se časová značka signálu posouvá a tím mění rozdíl časů - vytváří se odchylka v jednom směru.

Při přenosu signálu z družice často dochází i k jeho zkreslení ve vrstvách atmosféry. Nejvíce signál zkresluje tzv. Ionosféra, vrstva od 50 km n.m. do výšky 1000 km nad mořem. V této vrstvě dochází k jevu ionizace a nachází se zde zvýšená elektronová hustota, která je problematická pro přenos všech elektromagnetických signálů. Signály procházející touto částí zemské atmosféry jsou zkreslovány nezávisle na frekvenci. Dochází k jejich zpoždění a v konečném důsledku prodloužení doby cesty k přijímači a dráhy signálu.





Obrázek 3.4: Chyba měření v závislosti na vzájemné poloze družic

Přenos troposférou je k elektromagnetickému vlnění méně razantní. Pro signály s frekvencí menší než 30 GHz dochází pouze k zanedbatelnému zkreslení. Tato vrstva se nachází mezi povrchem Země a pásem 50 km nadmořské výšky a z fyzikálního hlediska se skládá z neutrálních částic.

### 3.10 Komunikační protokol a přenos dat

Pro komunikaci mezi přijímačem a dalšími zařízeními se v případě potřeby využívá protokolu NMEA (National Marine Electronics Association). Jedná se o textový formát přenosu dat. Komunikace probíhá jednosměrně a sériově v podobě vět, které obsahují informace o poloze, dále např. almanach či jiné důležité údaje. Výrobci přijímačů často ale protokol obměňují či do něj přidávají vlastní úpravy, což často končí problémy s kompatibilitou.

Většina dnešních přijímačů je kompatibilní se standardem NMEA 0183, který podrobně dále rozebereme. Tento standard využívá specifikace EIA-422-B, známé jako rozhraní RS-422, které lze pro většinu operací považovat za kompatibilní s rozhraním RS-232. Přenos dat je prováděn výhradně ve větách v kódování ASCII, probíhá rychlostí 4800 Bdps po 8 datových bitech, bez parity s jedním stop-bitem. Každá věta začíná znakem dolaru '\$' a končí znakem známým jako CR (carriage return) následovaným znakem LF (line feed). Data jsou oddělená znakem čárka. Někdy bývá přidáván i kód kontrolující validitu dat.[7] Věta začíná posloupností pěti znaků, z nichž první dva jsou volitelné a identifikují id zařízení. Dále následuje vlastní identifikace typu věty. Příklad obecného formátu NMEA protokolu je na obrázku 3.5. Z obrázku je patrné, že nejdůležitější část komunikace mezi GPS přijímačem a připojeným zařízením je na pozicích 14-23 a 24-34, a to aktuální zeměpisná šířka a délka.

Byte	Data	
1	\$	
2	M	device
3	P	address
4	K = kilometres N = nautical miles U = microseconds	cross track   error   units
5 - 8	0 - 9 or .	cross track error value
9	L or R	cross track error position
10	T or M	True or Magnetic bearing
11 - 13	0 - 9	bearing to next waypoint
14 - 23	12D34'56"N or 12D34.56"N	present latitude
24 - 34	123D45'56"W or 123D45.67"W	present longitude
35	non-ASCII status byte bit 0 = 1 for manual cycle lock 1 = 1 low SNR 2 = 1 cycle jump 3 = 1 blink 4 = 1 arrival alarm 5 = 1 discontinuity of TDs 6 = 1 always	
36	"NUL" character (hex 80)(reserved status byte)	
37	"ETX" character (hex 83)	
	Any unavailable data is filled with "NUL" bytes.	

Obrázek 3.5: Základní popis obecné zprávy protokolu NMEA

### 3.11 Další využití GPS v praxi

Nejčastější využití systému GPS:

1. geodézie
2. navigace
3. geocaching
4. různé druhy her s GPS atd.

## Kapitola 4

# Mobilní zařízení Apple iPhone a Windows Mobile

### 4.1 Úvod

V dnešní době jsme svědky trendu stále se zlepšujících technologií výroby elektronických součástí, přístrojů a hlavně počítačů. Podmnožina funkcionality počítače je často implementována prostřednictvím mobilních či vestavěných zařízení nesrovnatelně menší velikosti, často mobilních telefonů. Klade se důraz na výkon, který je sice řádově nižší, avšak pro účely použití těchto výrobků dostačující, ale na velikost, cenu, konkurenceschopnost a míru funkcionality zařízení. Nejžádanějšími funkcemi mobilních zařízení jsou kromě základních schopností telefonovat, posílat zprávy, fotografovat či nahrávat zvukovou stopu také schopnost připojovat se k bezdrátové síti Wi-Fi, schopnost prohlížení internetových stránek přímo na zařízení a také z pohledu této práce nejdůležitější vlastnost, a to schopnost pracovat jako GPS přijímač. Všechny tyto body mobilní zařízení Windows Mobile a Apple iPhone splňují.

### 4.2 Apple iPhone

Apple iPhone je mobilní zařízení implementující veškeré uvedené funkce. Vyrábí ho firma Apple se sídlem v Cupertino v Kalifornii, USA. V USA se začal prodávat 29.června 2007, tehdy pod názvem iPhone 2G. Cena za jeho nejlevnější variantu s 8 GB úložným prostorem byla 399 USD. Stal se hitem tentýž rok kvůli novinkám jako dotykový displej či Wi-Fi připojení. Pro připojení k internetu je možno využít služeb GPRS/EDGE. Nové aplikace se instalují prostřednictvím tzv. Apple Store - elektronického systému pro distribuci iPhone aplikací a další výrobky firmy Apple. Instalování aplikací třetích stran jinak než prostřednictvím Apple Store je zakázáno v licenčních podmínkách společnosti. iPhone existuje ve třech generacích, nejnovější (k 15.5.2010) je iPhone 3GS, pro představu je uveden i s technickými parametry na obrázku 4.1. Druhá generace, generace 3G se liší od 2G přínosem nového druhu internetového připojení. Generace 3GS se však od 3G liší ve více parametrech. Zvláště to je nový procesor AAR 11 600 MHz, který nahrazuje starý 412 MHz ARM 11 dále pak například podpora OpenGL 2.0 nebo 32 GB úložného prostoru. Cena iPhone 3G 16GB se u nás k 12.5.2010 pohybuje okolo 14 000Kč[zdroj [www.vodafone.cz](http://www.vodafone.cz)].

Operační systém iPhone se nazývá iPhone OS a je odvozen od operačního systému MAC OS X. iPhone OS je založen na unixovém jádru a využívají jej další produkty jako Apple iPad a iPod Touch.





Obrázek 4.1: Apple iPhone 3GS s technickou specifikací[2]

### 4.3 Windows Mobile

Zařízení Windows Mobile device (tím je myšleno Pocket PC, Smartphone, HTC nebo jakékoliv zařízení podporující OS Windows Mobile) je konkurenčním produktem firmy Microsoft s hlavním sídlem na Microsoft Way 1, Redmond WA, USA. Toto zařízení využívá operačního systému Windows Mobile, odvozeného od OS Windows CE, který je určen pro vestavěná zařízení. Disponuje stejně jako iPhone veškerými funkcemi uvedenými v 4.1. Název Windows Mobile dostalo zařízení až v roce 2003, kdy se jako pokračovatel tehdejšího Microsoft Pocket PC přejmenovalo na Windows Mobile (WM) 2003. Pocket PC bylo vydáno v roce 2000, umělo pracovat na architekturách SH-3, MIPS a ARM[15], obsahovalo hardware pro vyměnitelné karty. Následovalo Pocket PC 2002. Po WM 2003 přišlo WM 2003 SE, 5.0, 6.0, 6.5 s drobnými úpravami oproti předešlým verzím.

V roce 2010 se objevil na trhu pokračovatel této série - Windows Mobile 7, přejmenován na Windows Phone. Tento výrobek obsahuje nový, zpětně nekompatibilní operační systém Windows Phone 7 series, nebude mít schopnost kopírovat a vkládat text mezi aplikacemi stejně jako iPhone a bude výrazně omezovat multitasking. Tímto přístupem jde o napodobení obchodní strategie Applu. V dalších kapitolách se budu zabývat a navrhovat software pro operační systém Windows Mobile 6.5. Tento umožňuje oproti iPhone OS instalace aplikací třetích stran, oproti Windows Phone má dostupnější dokumentaci a je nejrozšířenější. Ukázka zařízení HTC Touch2 podporující systém Windows Mobile je na obrázku 4.2. Jeho současná cena (12.5.2010) se v ČR pohybuje okolo 6000Kč[zdroj www.sunnysoft.cz].

Tolik k porovnání obou zařízení a nyní můžeme přistoupit k návrhu aplikací. Bude se jednat o dvě klientské aplikace pro každé zařízení komunikující s jedinou aplikací serverovou.



Model	HTC Touch2 (T3333)
Processor	Qualcomm® MSM7225™, 528 MHz
Operating System	Windows Mobile® 6.5 Professional
Memory	ROM: 512 MB RAM: 256 MB
Dimensions (LxWxT)	104 X 55 X 12.9 mm (4.1 X 2.16 X 0.51 inches)
Weight	110 grams (4 ounces) with battery
Display	2.8-Inch TFT-LCD touch-sensitive screen with QVGA resolution
GPS	Internal GPS antenna
Connectivity	Bluetooth® 2.1 with Enhanced Data Rate and A2DP for wireless Wi-Fi®: IEEE 802.11 b/g 3.5 mm audio jack HTC ExtUSB™ (11-pin mini-USB 2.0)
Camera	3.2 megapixel color camera with fixed focus
Battery	Rechargeable Lithium-ion battery Capacity: 1100 mAh Talk time: <ul style="list-style-type: none"> <li>■ Up to 370 minutes for WCDMA</li> <li>■ Up to 440 minutes for GSM</li> </ul> Standby time: <ul style="list-style-type: none"> <li>■ Up to 500 hours for WCDMA</li> <li>■ Up to 370 hours for GSM</li> </ul>

Obrázek 4.2: HTC Touch2 s OS Windows Mobile 6.5 a technickou specifikací<sup>[1]</sup>

## Kapitola 5

# Klientská aplikace pro Windows Mobile 6.5

### 5.1 Návrh aplikace

Aplikace získává ze svého GPS rozhraní informace o poloze ve formě zeměpisné šířky a délky. Tyto údaje pomocí internetového připojení a zvoleného druhu komunikace odesílá na server, kde se data ukládá, popř. Jako rozšíření reprezentuje vizuálně. Klientská aplikace pro Windows Mobile má být spustitelná na verzi OS Windows Mobile 6.5 a obsahovat následující součásti:

- Uživatelské rozhraní
- Komunikační rozhraní se zdrojem GPS dat
- Internetové připojení
- Vhodný internetový komunikační interface klient/server

#### 5.1.1 Uživatelské rozhraní

Aplikace by má být spustitelná na všech zařízeních podporující OS WM 6.5, zahrnujících od typů Pocket PC až po Smartphone, tudíž by měla mít i téměř totožné grafické rozhraní. Očekává se, že tento předpoklad zajišťuje software development kit (SDK, viz implementace kap. 5.3) zmíněného operačního systému, pomocí něhož aplikaci programujeme. V grafickém rozhraní aplikace zjišťuje uživatelské vstupy, jako je adresa serveru, port či interval odesílání GPS informací v sekundách, prostřednictvím komponent k tomuto účelu určených. Ovládání grafického rozhraní však závisí na typu hardware. Pokud se bude jednat o hardware typu Pocket PC, ovládá se podobně jako PDA, pokud ovšem je hardware typu Smartphone, musí se mezi prvky uživatelského rozhraní přepínat pomocí klávesnice.

#### 5.1.2 Komunikační rozhraní se zdrojem GPS dat

Pro získávání souřadnic je na Windows Mobile potřeba zpřístupnit funkce pro práci s tímto rozhraním. Tohoto je docíleno připojením dynamické knihovny typu ".dll" ke zdrojovému kódu, která vše potřebné zajišťuje.

### 5.1.3 Internetové připojení

Internetové připojení pro mobilní telefony nabízí celá řada společností. Ceny se pohybují řádově od stovek Kč za nejlevnější alternativy, za náročnější internetová připojení se ceny pohybují okolo tisíce Kč měsíčně. Pro naše účely odesílání GPS souřadnic (několik kilobytů textových dat) postačují nejlevnější alternativy, využijeme technologii GPRS nebo EDGE. Přehled viz tabulka 5.1.

	O2	T-Mobile	Vodafone
Název tarifu	O2 Internet v mobilu	T-Mobile Surf+	Vodafone Internet v mobilu
Cena za měsíc(s DPH)	150,00 Kč	118,80 Kč	177,00 Kč
Limit přenesených dat(FUP)	150MB/měsíc	50 MB/měsíc	100 MB/měsíc
Rychlost připojení	až 3,6Mbit/s	až 236,8kb/s	neuveďeno
Podporované technologie	GPRS, EDGE, HSDPA, UMTS	EDGE/GPRS	EDGE/GPRS

Obrázek 5.1: Přehled základních paušálních tarifů největších českých operátorů<sup>1</sup>

### 5.1.4 Komunikace se serverem

Pro komunikace se serverem využijeme spojovaného připojení TCP. Komunikace na nejnižších úrovních bude na straně klienta plně probíhat v režii operačního systému. TCP klientovi se bude předávat adresa serveru a port, na které má data odeslat. Pokud se vyskytne jakákoliv chyba v TCP komunikaci, bude vrácena chyba a textová reprezentace její příčiny, která se zobrazí viditelným způsobem v GUI. Při výskytu chyby se odesílání dat na server přeruší a aplikace bude čekat na další reakci od uživatele. Reakce uživatele může být buď ukončení programu nebo opakovaný pokus o navázání spojení.

## 5.2 Vývojové prostředí

Vývoj aplikací pro OS Windows Mobile probíhá výhradně v operačních systémech Microsoft Windows verzí 2000 SP4 a vyšší. V Linuxu nebo MAC OS X není možné vyvíjet tyto aplikace, protože pro ně neexistují alternativy vývojových nástrojů. Samotný operační systém však nestačí, je potřeba doinstalovat další součásti potřebné pro vývoj software. Při vývoji aplikace byly použity tyto:

1. Microsoft Visual studio 2008
2. Windows Mobile 6 SDK + Window Mobile Device Emulator
3. Microsoft ActiveSync nebo Windows Mobile Device Center
4. Windows Mobile 6.5 Developer Tool Kit (nepovinné)
5. Windows Mobile 6.1.4 Emulator Images (nepovinné)
6. Microsoft Virtual PC 7 (nepovinné)
7. Microsoft .NET Framework 2 SP1 a .NET Compact Framework 3.5

<sup>1</sup>Tarifů pro připojení k internetu platné k 13.5.2010

### 5.2.1 Visual Studio

Visual studio je rozsáhlé vývojové prostředí firmy Microsoft nad operačním systémem Microsoft Windows. Pro účely vývoje Windows Mobile aplikací je potřeba používat alespoň verze Visual Studio 2005. Firma Microsoft jej definuje takto:

Visual Studio 2008 Professional Edition (US) je ucelená sada nástrojů urychlující proces přeměny programátorské vize ve skutečnost. Byla navržena pro podporu projektů vyvíjených pro web (včetně ASP, .NET a AJAX), Windows Vista, Windows Server 2008, systém Microsoft Office 2007, SQL Server 2008 a zařízení Windows Mobile. Oceníte zejména programování a ladění vícevrstevných serverových aplikací zcela uvnitř vývojového prostředí Visual Studia včetně technologie "Unit testing", plnou podporu všech typů služeb Click Once, tvorby MSI instalačních balíčků, nejvyšší plnohodnotnou verzi SQL Serveru v podobě produktu SQL Server Developer Edition a vytváření tiskových sestav.[5]

Je možné zde pracovat v několika programovacích jazycích jako je C++, Visual Basic či C# + .NET. Pro tvorbu aplikace však bude použit poslední zmíněný.

Visual Studio zajišťuje spolupráci součástí 2-5. Po naprogramování kódu ve zvoleném jazyce je možnost kompilace tohoto kódu přímo ve vývojovém prostředí. Vývojové prostředí dále spolupracuje s případně doinstalovaným Windows Mobile 6.5 SDK (viz bod 2.) a s Windows Mobile 6.1.4 Emulator Images (viz bod 3.). V prostředí je možno zvolit si z řady různých projektů od projektů pro mobilní zařízení určitého typu až po vytváření balíčků pro distribuci naprogramované aplikace.

### 5.2.2 Windows Mobile 6 SDK

Jedná se o oficiální vývojový kit spolupracující s Visual Studiem a obsahující veškeré oficiální zdrojové kódy, dokumentace, hlavičkové soubory, knihovny a základní obrazy emulátorů potřebné pro tvorbu Windows Mobile aplikací. Je dostupný ve dvou verzích z nichž každá je určena pro specifický typ hardware, viz následující rozdělení:

- Windows Mobile 6 Standard SDK - je určen pro vývoj aplikací na Smartphone
- Windows Mobile 6 Professional SDK - určen pro vývoj aplikací pro Pocket PC a Pocket PC Phone Edition

I když tato aplikace bude vyvíjena v šabloně projektu pro Smartphone, jádro bude stejné pro všechny tři typy zařízení, protože využívat budeme Windows Mobile 6 Professional SDK. Rozdíl je pouze zjednodušeně řečeno pouze v hardwarovém ovládní GUI. SDK obsahuje i další obrazy virtuálních zařízení pro emulátor.

Visual Studio také obsahuje důležitý emulátor virtuálních zařízení (**Windows Mobile Device Emulator**). Tato součást je velmi důležitá pro testování aplikace jak při vývoji tak po dokončení. Díky ní je možné testovat bez nutnosti mít vlastní přístroj, pro který danou aplikaci vytváříme. Emulátor načte všechny nainstalované obrazy zařízení v systému podporující operační systém Windows Mobile. Je možné si vybrat z načtených obrazů ten který potřebujeme. Výběr pokrývá značnou část hardwaru.

### 5.2.3 Microsoft ActiveSync a Windows Mobile Device Center

Microsoft ActiveSync je služba synchronizující obsah mobilního telefonu s OS Windows Mobile s daty v počítači. Zajišťuje i přenos dat jak ze zařízení, tak do zařízení. Microsoft ActiveSync je dostupný pouze pro verze OS Windows XP, pro vyšší nikoliv. Ve vyšších

verzích OS Microsoft Windows (v současnosti<sup>2</sup> Windows Vista a Windows 7) je služba ActiveSync dostupná pod názvem Windows Mobile Device Center. Od služby ActiveSync je liší prakticky jen ve vzhledu uživatelského rozhraní.

#### 5.2.4 Windows Mobile 6.5 Developer Tool Kit

Jedná se opět o vývojový balík obsahující hlavičkové soubory, knihovny, dokumentaci a ukázkové zdrojové kódy k programování aplikací však již pro OS Windows Mobile 6.5. Pro správný běh uvedeného software musí být korektně nainstalován Windows Mobile 6 SDK. Obsahuje i další obrazy pro emulátor virtuálních zařízení.

#### 5.2.5 Windows Mobile Emulator Images

Tento doplňkový balíček přidává do vývojového prostředí další lokalizované obrazy pro virtuální emulátor zařízení.

#### 5.2.6 Microsoft Virtual PC

Software MS Virtual PC slouží k vytváření virtuálních prostředí v rámci systému Microsoft Windows. Tyto prostředí lze využít jako testovací například pro betaverze, starší programy nebo testování kompatibility. Samotný software ale není potřeba, důležitý je jeho ovladač pro spojení připojení k internetu počítače a komunikačního rozhraní zařízení v emulátoru, který po instalaci do systému přidá. Prostřednictvím tohoto ovladače se docílí připojení emulovaného zařízení do internetu. V takto připraveném vývojovém prostředí se mohou spouštět a testovat aplikace využívající například třídy pro práci s TCP a UDP a internetové protokoly vůbec.

#### 5.2.7 Microsoft .NET Framework

Microsoft .NET Framework (čti dot-net) je soubor knihoven a služeb pro operační systémy Microsoft Windows, zajišťující určitou vrstvu abstrakce nad operačním systémem pro vývojáře, zpřístupňuje jasně a srozumitelně operace nad jeho základními součástmi a obsahuje předkompilované kódy pro nejpoužívanější nízkourovňové konstrukce. Obsahuje také virtuálně běžící službu interpretující tyto příkazy. Spektrum použití je široké, přes různé programovací techniky, druhy aplikací a zařízení. Ve směru vývoje aplikace, která je předmětem této kapitoly, je tento software potřebný, kromě nutnosti jej instalovat pro správný běh některých dříve uvedených aplikací, k vkládání prvků GUI. Microsoft .NET Compact Framework zajišťuje obdobné funkce, ale pro mobilní zařízení.

### 5.3 Implementace

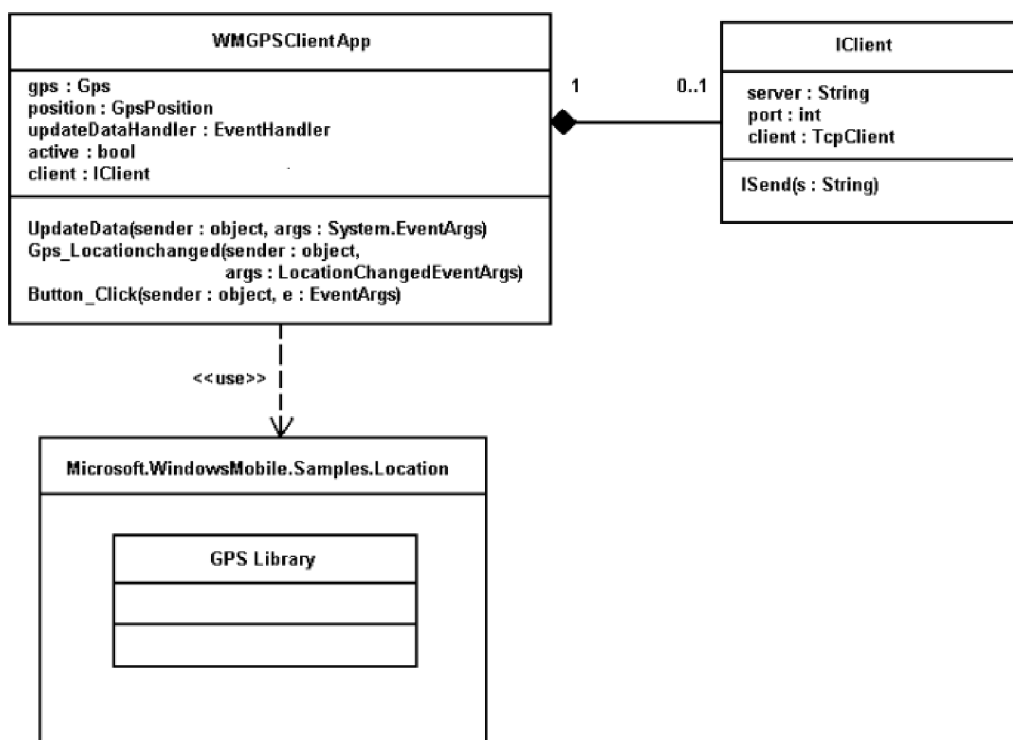
Pro implementaci popsaného softwaru bylo použito vývojového prostředí uvedeného v 5.2 a jako programovací jazyk byl zvolen C# (čti sí šárp) s knihovnou .NET Framework 2.5. Pro připojení k internetu bylo zvoleno připojení od O2 (O2 internet v mobilu viz 5.1). Implementace proběhla v operačním systému Microsoft Windows XP se Service Pack 2. Diagram objektové reprezentace vidíme na obrázku 5.2.

Důležitým prvkem implementace je část pracující s hardwarem přístroje pro práci s GPS.

---

<sup>2</sup>Platné ke dni 13.5.2010





Obrázek 5.2: Diagram knihoven a tříd

### 5.3.1 Implementační část pracující s GPS

K práci s GPS hardware bylo využito dynamicky linkované knihovny (DLL) "Microsoft.WindowsMobile.Samples.Location.dll", která zajišťuje přítomnost důležitých tříd. Obsahuje dva důležité prvky, a to

1. třídu *Microsoft.WindowsMobile.Samples.Location.Gps()*
2. nastavení obsluhy události  
*Microsoft.WindowsMobile.Samples.Location.LocationChangedEventHandler (EventHandler e)*

Třída obsahuje metody *Open()* a *Close()*, které uvádějí do chodu GPS rozhraní nebo jej vypínají. Důležitějším prvkem je ale nastavení obsluhy události pomocí syntaxe "+=", která provádí registraci delegáta zmíněné události.

#### Úryvek kódu 5.1: Ukázka registrace delegace a obsluhy události

```
private Gps gps;
...
this.gps.LocationChanged += new
Microsoft.WindowsMobile.Samples.Location.
LocationChangedEventHandler(this.gps_Locationchanged);
...
void gps_Locationchanged(object sender, LocationChangedEventArgs args){
... //Získání GPS dat z argumentu funkce invokované událostí
}
```

Po vzniku události (úryvek 5.1) *Microsoft.WindowsMobile.Samples.Location.LocationChanged* v systému dojde k delegování této události na atribut (objekt *gps*) třídy *WMGPSClientApp* viz 5.2 a vyvolání metody pro obsluhu *gps\_Locationchanged()*. V takto vyvolané metodě pak lze získat zeměpisnou šířku a délku ve formě GPS souřadnic a s těmito údaji dále pracovat. Data jsou přijímána asynchronně k běhu programu po aktivaci objektu třídy *Microsoft.WindowsMobile.Samples.Location.Gps()* metodou *Open()*. Ukončení práce objektu potažmo i GPS rozhraní se provádí metodou *Close()*.

## 5.4 Synchronizace a komunikace

Údaje o zeměpisné šířce a délce jsou synchronizovány s časovým intervalem zadaným uživatelem. Synchronizace probíhá tak, že data se zpracovávají nezávisle na příchodu událostí asynchronního rozhraní GPS, a to pouze v určitém časovém intervalu. Pokud není povolena aktualizace údajů (tzn. neuběhl časový interval), a přesto je v GPS rozhraní generována událost s novými dostupnými daty, budou tyto data zahozeny. Zpracovaná data se odesílají na server jednoduchou spojovanou komunikací pomocí protokolu TCP. Komunikace je plně v režii operačního systému a dostupná je pouze na nejvyšší úrovni (zadání portu, adresy serveru a obsluha chyb).

## 5.5 Distribuce aplikace

Výslednou aplikaci lze připravit k distribuci a testování přímo ve Visual Studiu. Do aktuálního projektu lze přidat další podprojekt, který zapouzdří veškeré zdrojové kódy projektu, nastavení a další zdroje do přenositelného formátu CAB. Takto vytvořený CAB soubor lze standardním způsobem přenést do fyzického zařízení Windows Mobile a na místě také aplikaci nainstalovat.

## 5.6 Testování aplikace

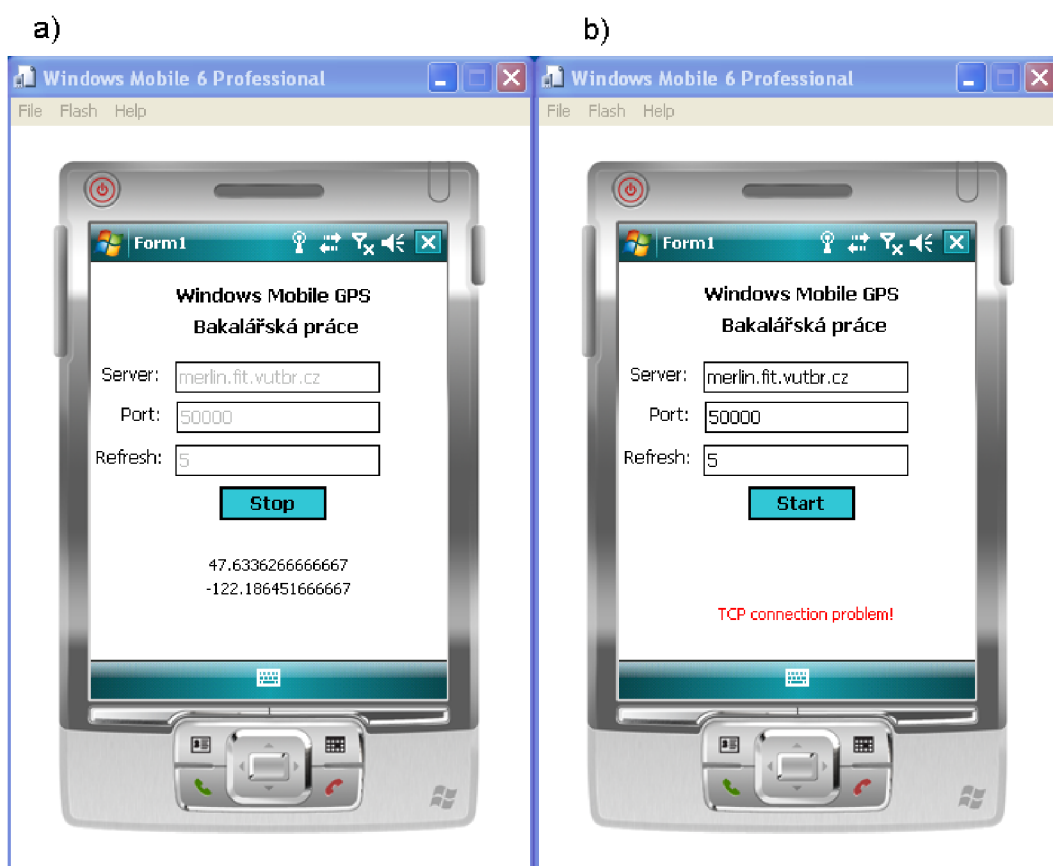
Pro testování aplikace lze rovněž využít emulátoru nebo je možné testovat ve fyzickém zařízení. Tato aplikace však byla testována v emulátoru, kde není přímo fyzicky dostupné GPS hardware, které je jádrem programu. Funkce GPS hardware se dají simulovat nainstalováním a spuštěním oficiální aplikace **FakeGPS** na obraze zařízení připojeném v emulátoru.

Tato aplikace zastupuje rozhraní GPS a přebírá jeho funkce a obsluhy. Je součástí Windows Mobile 6 SDK, umístěna je ve složce "..\Windows Mobile 6 SDK/Tools/GPS/". Po jejím spuštění je nutno vybrat zdrojový textový soubor se statickými uloženými daty



ve formátu NMEA (popsán v kapitole 3.10), který se využívá ke komunikaci GPS hardware s telefonem. Po uzavření této aplikace zůstává běžet na pozadí a generuje GPS data, jako by se jednalo o reálné hodnoty na fyzickém zařízení.

Další důležitou součástí testování je přístup buď fyzického zařízení nebo emulátoru do internetu. U fyzického zařízení je potřeba vybrat poskytovatele mobilního internetového připojení a tuto službu u něj aktivovat (možnost porovnat například v 5.1). V emulátoru je možné využít připojení k internetu počítače. Pro tento způsob musí být nainstalován Microsoft Virtual PC (kap. 5.2.6), který obsahuje síťový ovladač Virtual Network Driver. Po nainstalování ovladače se zpřístupní možnosti připojení k internetu ve Windows Device Emulator v nabídce File - Configure - Network. Zde je možné vybrat, které připojení k internetu chceme využít. Tímto způsobem nastavený emulátor připojí aktuálně spuštěný obraz do internetu a je možno otestovat aplikaci v celé míře funkcionality. Výsledná aplikace byla spuštěna na emulátoru typu Windows Mobile 6 Professional Emulator a je na obrázku 5.3.



Obrázek 5.3: Finální klientská aplikace pro zařízení s OS Windows Mobile 6.5

Obrázek 5.3 a) znázorňuje korektně běžící aplikaci, tzn. úspěšně se získají data z GPS rozhraní a ty jsou odeslána komunikačním rozhraním na server v zadaných intervalech. Po bezchybném odeslání se vypíše získaná zeměpisná šířka a délka.

Naopak obrázek 5.3 b) znázorňuje GUI aplikace při výskytu chyby v komunikaci se serverem. Chybová hláška je vypsána v poloze těsně pod místy pro výpis GPS souřadnic, aplikace pozastaví svou činnost a vyčkává na příkaz uživatele, jak také bylo zadáno v návrhu software.

## Kapitola 6

# Klientská aplikace pro Apple iPhone

### 6.1 Návrh aplikace

Aplikace pro Apple iPhone musí splňovat totožnou funkcionalitu jako aplikace pro Windows Mobile. Ze svého GPS rozhraní získává informace o poloze ve formě zeměpisné šířky a délky. Tyto pomocí internetového připojení odesílá na server, kde se data ukládají, popř. nějakým způsobem reprezentují. Podobně jak aplikace pro Windows Mobile se skládá z následujících součástí:

- Uživatelské rozhraní
- Komunikační rozhraní se zdrojem GPS dat
- Internetové připojení
- Vhodný internetový komunikační interface klient/server

#### 6.1.1 Uživatelské rozhraní

Aplikace je spustitelná na zařízení Apple iPhone 2G s verzí firmware 2.2.1. Uživatelské rozhraní zajišťuje a zpracovává uživatelské vstupy a výstupy programu. Uživatel vkládá pomocí textových polí a softwarové klávesnice informace o cílovém serveru, portu a časovém intervalu v sekundách, s jakým se data mají posílat.

#### 6.1.2 Komunikační rozhraní se zdrojem GPS dat

Pro získávání souřadnic je možno na iPhone využít oficiálního API pro práci s GPS, které je součástí iPhone SDK.

#### 6.1.3 Internetové připojení

Ačkoliv řada operátorů nabízí přímo balíčky internetových připojení pro iPhone<sup>3</sup>, pro účely této aplikace postačují základní alternativy viz tabulka 5.1. Pro připojení bude využita technologie GPRS nebo EDGE.

---

<sup>3</sup>k 15.3.2010 například O2

### 6.1.4 Komunikace se serverem

Pro komunikaci se serverem využijeme opět spojovaného připojení TCP. TCP klientovi se předává adresa serveru a port, na které mají data odeslat. Pokud se vyskytne jakákoliv chyba v TCP komunikaci, je vrácena chyba a textová reprezentace její příčiny, která se zobrazí viditelně v GUI. Při výskytu chyby se odesílání dat na server nepřerušuje, aplikace čeká na další reakci od uživatele. Reakce uživatele může být buď ukončení programu nebo opakovaný pokus o navázání spojení.

## 6.2 Vývojové prostředí

Pro vývoj aplikací pro Apple iPhone je potřeba výhradně systému řady MAC OS od firmy Apple, nejlépe MAC OS X verzí 10.0 a vyšší a počítač Apple Macintosh, na kterém tento systém běží. Jediný další software, který je potřeba, je Apple iPhone SDK.

### 6.2.1 Apple iPhone SDK

Jedná se o nejdůležitější vývojový software potřebný k programování iPhone aplikací. Programovací jazyk prostředí je **objective-C**, méně i jazyk C a C++. Zmíněný iPhone SDK obsahuje frameworky, hlavičkové soubory, ukázky kódu, dokumentaci, simulátor iPhone zařízení a další zdroje pro vývoj iPhone aplikací. Z nejdůležitějších je třeba uvést tyto tři:

- Apple Xcode
- Cocoa API (Application Programming Interface)
- iPhone Simulator

#### Apple Xcode

Jedná se o vývojové prostředí v operačních systémech MAC OS (X) běžících na počítačích Macintosh pro tvorbu aplikací různých platform. V tomto prostředí probíhá veškerý vývoj jak pro MAC OS (X) tak pro iPhone, iPod, iPad či další výrobky firmy Apple. Nejčastěji používaný jazyk je zde Objective-C, což je objektová nástavba jazyka C s prvky jazyka Smalltalk, ze kterého přejímá hlavně mechanismus zasílání zpráv objektům. Jazyk C samotný lze pro některé konstrukce využívat také. Součástí Xcode je také Interface Builder, tj. vývojový nástroj pro tvorbu uživatelských rozhraní.

#### Cocoa API

Cocoa API je soubor frameworků, utilit, zdrojových kódů, dokumentace atd. Je jádrem nejen iPhone SDK, ale i dalších SDK jako je MAX OS X SDK. Cocoa frameworky zpřístupňují na nejvyšší úrovni širokou funkcionalitu operačního systému iPhone OS.

#### iPhone Simulator

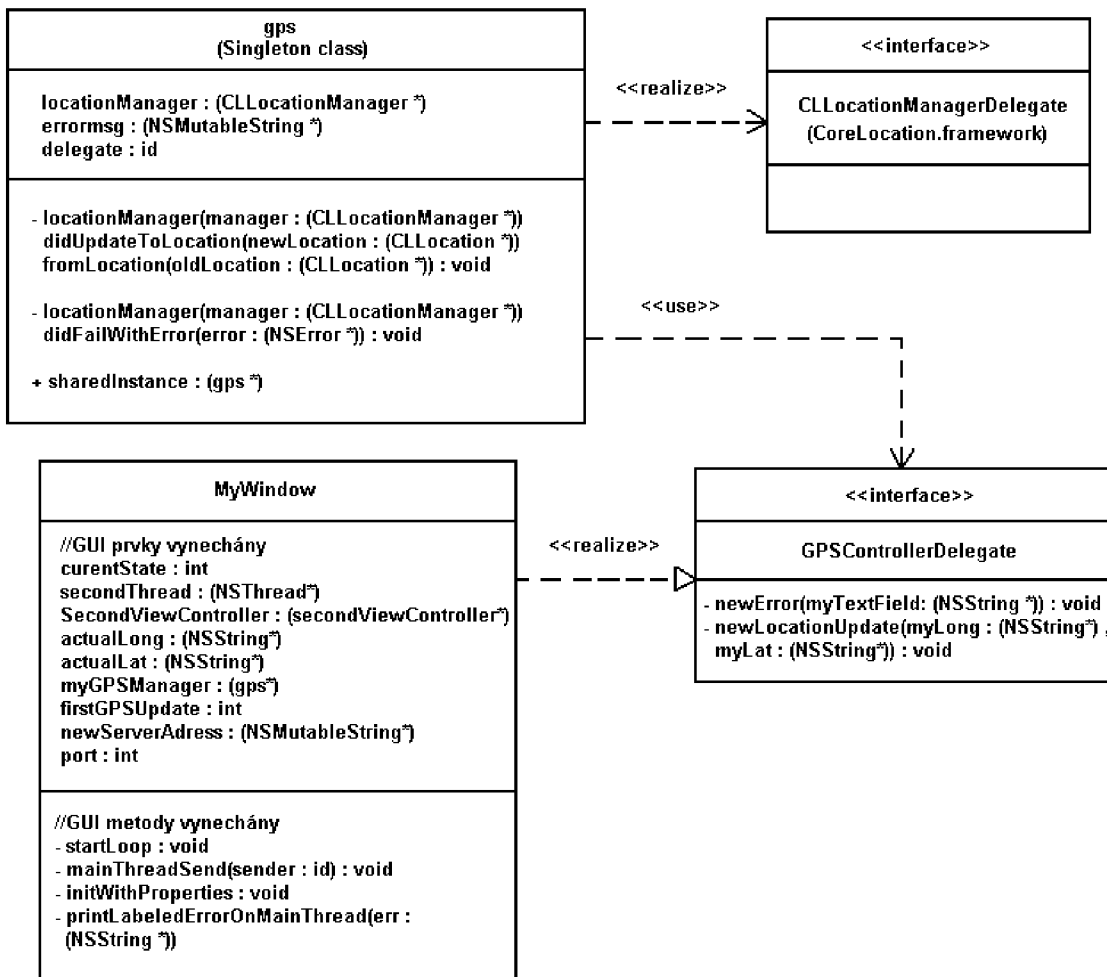
Pro testování a vývoj aplikací je možno použít simulátor iPhone Simulator z SDK. Jedná se o jedinou možnou alternativu testování programu pokud nevlastníme přímo fyzicky zařízení iPhone. Rozdíl simulátoru oproti fyzickému zařízení je hlavně v cílové architektuře iPhone simulátoru, a tou jsou procesory Intel. Fyzické zařízení ovšem pracuje s procesorem ARM (kap. 4.2). Tento problém může být příčinou nekompatibilit. Poslední verze iPhone

SDK zahrnující iPhone Simulator vždy obsahuje i všechny verze minulých jader - je možno simulovat i jiné verze firmware. Následující iPhone firmware jsou dostupný v iPhone SDK 3.1: 2.0, 2.1, 2.2, 2.2.1 (firmware implementované aplikace), 3.0, 3.1.

Pro každý z těchto firmware je možno vyvíjet a testovat aplikace, je potřeba však mít na paměti, že v každé verzi firmware nemusí být a nejsou zpětně kompatibilní API - jsou často pozměňovány. Při přechodu na jiný firmware je potřeba zkontrolovat, zda jsou všechny funkce a třídy použité v souladu s firmwarem starým, dostupné i ve firmwaru novém. Simulator zvládá prakticky všechny funkce fyzického iPhone, kromě ovládání více prsty a akcelerometru.

### 6.3 Implementace

Pro programování aplikace byl použit Apple iPhone SDK 3.1 a verze firmwaru iPhone OS 2.2.1 (kódové označení 5H11) na operačním systému MAC OS X 10.5 Leopard. Implementace proběhla v programovacím jazyce Objective-C.



Obrázek 6.1: Diagram tříd jádra aplikace pro iPhone

### 6.3.1 GPS rozhraní

Základem aplikace je opět práce s GPS rozhraním přístroje. Diagram jádra programu je na obrázku 6.1. Základem získávání GPS souřadnic je rozhraní *CLLocationManagerDelegate* frameworku *CoreLocation.framework* obsaženého v iPhone SDK 3.1. Třída *gps* toto rozhraní implementuje a tímto získává přístup k metodám, jež vyvolávají změny GPS polohy přístroje. Třída je vytvořena podle návrhového vzoru Singleton, tudíž pro práci s GPS rozhraním může být v rámci této aplikace pouze jedna instance této třídy. Dále obsahuje atribut *delegate*, který ukazuje na objekt, na který jsou delegována GPS data při změně. Tento objekt musí samozřejmě implementovat rozhraní *GPSControllerDelegate*, které deklaruje metody pro práci s takto přijatými daty. Takovým objektem je instance třídy *MyWindow*.

Úryvek kódu 6.1: Ukázka delegace GPS dat na sdílenou instanci

```
// Metoda vyvolaná událostí změny souřadnic je součástí
// rozhraní CLLocationManagerDelegate
- (void)locationManager:(CLLocationManager *)manager
  didUpdateToLocation:(CLLocation *)newLocation
  fromLocation:(CLLocation *)oldLocation{
    ...
    // V parametru newLocation se nachází nové souřadnice
    ...
    // Tyto souřadnice upravíme do požadovaného formátu
    // a zašleme delegátovi
    ...
    [self.delegate locationManagerUpdate:longt:lat];
}
```

V úryvku kódu 6.1 vidíme, jak v objektu třídy *gps* implementující rozhraní *CLLocationManagerDelegate* dochází k delegaci dat, produkovaných změnou GPS souřadnic, na sdílenou instanci. V této sdílené, v případě popisované aplikace v instanci třídy *MyWindow*, instanci dojde k vyvolání metod rozhraní *GPSControllerDelegate*, a to buď metody *newLocationUpdate(...)* v případě korektního a bezchybného chování GPS systému, nebo *newError(...)* v případě chyby v rozhraní. Ukázka obsluhy při bezchybné práci aplikace je patrná z úryvku kódu 6.2.

Úryvek kódu 6.2: Ukázka obsluhy změny GPS souřadnic ve sdíleném delegátu

```
- (void)newLocationUpdate:(NSString *)myLong : (NSString *)myLat{
    actualLong = [myLong copy];
    actualLat = [myLat copy];
    myGPSLabel.text = [actualLat stringByAppendingString:actualLong];
}
```

Kód uvedený v 6.2 je součástí třídy *MyWindow*, která zajišťuje práci s grafickým uživatelským rozhraním. Důvodem je možnost okamžitě vypisovat data na výstup. Přijaté souřadnice jsou zpracovány a zobrazeny prostřednictvím komponenty *UILabel* s názvem *myGPSLabel* viditelně uživateli.

### 6.3.2 Grafické rozhraní

Grafické rozhraní je stavěno na podmnožině technologie Apple Aqua GUI, která je původně určená pro MAC OS X. Vstupy a výstupy jsou vyvedeny přesně podle návrhu aplikace.

### 6.3.3 Synchronizace

Stejně jako u Windows Mobile se aktualizují GPS data asynchronně k běhu programu. Odesílání dat na server však probíhá synchronně v daném časovém intervalu. Synchronizace probíhá implicitně tak, že pokud jsou nová data k dispozici, aktualizuje se vnitřní proměnná příslušného objektu. Pokud vyprší daný časový interval, je tato informace, ač z již aktualizované nebo neaktualizované, vnitřní proměnné zaslána na server.

## 6.4 Distribuce a testování aplikace

Aplikaci je možno testovat dvěma způsoby:

1. pomocí iPhone Simulatoru
2. testováním na fyzickém zařízení

### 1.iPhone Simulator

Toto je nejjednodušší cesta pro otestování aplikace. iPhone Simulator je integrován přímo ve vývojovém prostředí Xcode. Spouští se po kliknutí na tlačítko "Build and Go". Celá aplikace se zkompiluje a vytvoří se binární a podpůrné soubory pro spuštění v simulátoru, tzn. na architekturu procesoru Intel. Simulátor zpřístupňuje pro spuštěný program automaticky prvky prostředí operačního systému MAC OS X, jako například připojení k internetu, což je pro tuto aplikaci zásadní. Není potřeba dále nic nastavovat.

### 2.Fyzické zařízení

Nyní se dostáváme ke komplikované části vývoje iPhone aplikací, a to testování na fyzickém přístroji iPhone. Toto testování není přímo možné bez vývojářské licence, registrace a zaplacení poplatku. Je možné využít některý z dvou hlavních vývojářských programů:

- Standard program
- Enterprise program

**Standard program** Licence Standard programu stojí 99 dolarů ročně. Vytvořenou aplikaci je možno v rámci Standard programu distribuovat přes Apple App Store (Apple si účtuje 30% z ceny transakce), který je dostupný po internetu z každého iPhone přístroje, tudíž každý uživatel si danou aplikaci může stáhnout. Toto je nevýhodné, když nechceme, aby si všichni mohli ihned naši novou aplikaci stáhnout a používat. Aplikace vkládaná na App Store musí nejprve projít schvalovacím procesem, který zjišťuje různé nekorektní techniky v rozporu s Apple iPhone SDK licencí, jako jsou třeba využití privátních frameworků. Vložení takové aplikace na App Store je potom zamítnuto.

Druhá možnost je ve formě tzv. **Ad-Hoc distribuce**. Tato je určena k uzavřené distribuci aplikace nejvíce 100 lidem (100 kopií programu). Po jejím zakoupení je vývojáři zpřístupněna možnost zaslat program kompilovaný speciálně pro určité ID fyzického zařízení majiteli tohoto zařízení přes e-mail. Postup celého procesu je přibližně následující:

1. Vývojář si vyžádá ID Vašeho iPhone
2. Vývojář kompiluje program přímo pro toto ID
3. Aplikace je odeslána uživateli spolu se speciálním souborem .mobileprovision
4. Uživatel aplikaci spolu s .mobileprovision souborem přetáhne do okna iTunes
5. Aplikace se objeví v iTunes a po synchronizaci i na přístroji

**Enterprise program** Tento distribuční program je určen pro firmy s vyšším počtem zaměstnanců než 500. Vytvořenou aplikaci je možno distribuovat například uvnitř firmy. Licence stojí 299 dolarů ročně. Distribuce přes App Store není možná. Dále je potřeba aby firma vlastnila své takzvané Data Universal Number. Toto je identifikační číslo firem využívané pouze v USA. V ČR tedy není možné tento program využívat. Kompletní výpis dostupných developerských programů je na obrázku 6.2.

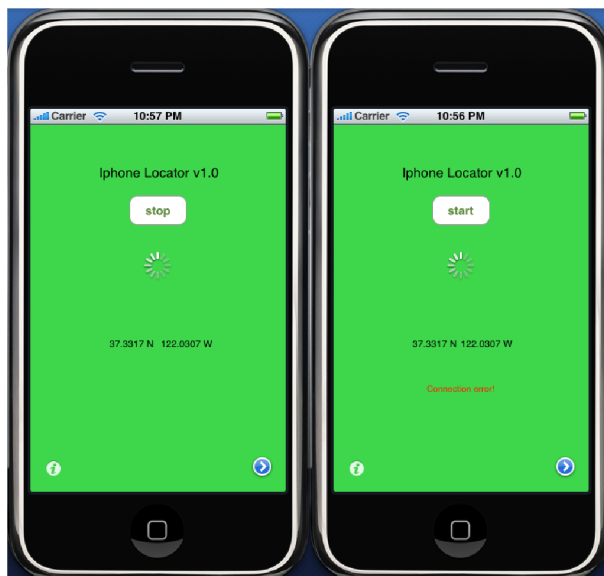
	iPhone Developer Program - Individual <a href="#">Learn more</a>	iPhone Developer Program - Company <a href="#">Learn more</a>	iPhone Developer Enterprise Program <a href="#">Learn more</a>	iPhone Developer University Program <a href="#">Learn more</a>	Registered as an Apple Developer <a href="#">Learn more</a>
Dev Center Resources	✓	✓	✓	✓	✓
iPhone SDK	✓	✓	✓	✓	✓
Select Pre-Release Software & Tools	✓	✓	✓		
Ability to Create Development Team		✓	✓	✓	
Apple Developer Forums	✓	✓	✓	✓	
Technical Support Incidents	2 per membership year	2 per membership year	2 per membership year		
Test on iPad, iPhone, and iPod touch	✓	✓	✓	✓	
Ad Hoc Distribution	✓	✓	✓		
In-House Distribution			✓		
App Store Distribution	✓	✓			
Price	\$99/year	\$99/year	\$299/year	Free	Free

Obrázek 6.2: Kompletní rozbor vývojářských programů pro Apple iPhone[4]<sup>4</sup>

**Výsledná aplikace** Náhled výsledné aplikace je na obrázku 6.3. Obrázek vlevo znázorňuje správnou činnost aplikace - odeslání dat na server. Obrázek vpravo však výskyt chyby a výpis jejího popisu. Na obrázku 6.4 jsou další doplňkové části aplikace, a to obrazovka nastavení projektu "Preferences" (vlevo) a ukázka nainstalované aplikace připravené ke spuštění (vpravo).

<sup>4</sup>Vývojářské programy platné k 15.5.2010





Obrázek 6.3: Výsledná aplikace, ukázka chodu programu a chyby v běhu



Obrázek 6.4: Výsledná aplikace, ukázka nastavení aplikace a spouštění



## Kapitola 7

# Uchování a zpracování dat

Na serveru se zpracovávají data z klientských aplikací, která jsou přijímána v jednoduchém textovém formátu obsahující zeměpisnou šířku a délku aktuálního umístění v čase. Pro potřeby elektronické knihy jízd je však nutno znát reálnou vzdálenost mezi jednotlivými dvěma body určenými zeměpisnou šířkou a délkou, přijatými z klientských aplikací. Ze znalosti vzdálenosti vždy mezi jednotlivými dvěma body součtem dostaneme celou uraženou vzdálenost např. při služební cestě. S touto informací a informací o aktuální ceně pohonných hmot již nic nebrání výpočtu nákladů na pohonné hmoty vozidla.

Výpočet vzdálenosti mezi dvěma body, kterou vozidlo urazilo, se z přijatých GPS souřadnic provádí pomocí Haversinova vzorce 7.1 - 7.5[9]. Rovnice uvažuje střední rádius Země 6,371km a úhly v radiánech. Výsledná vzdálenost je označena  $d$ .

$$\Delta lat = lat_2 - lat_1 \quad (7.1)$$

$$\Delta long = long_2 - long_1 \quad (7.2)$$

$$a = \sin^2(\Delta lat/2) + \cos(lat_1) \cdot \cos(lat_2) \cdot \sin^2(\Delta long/2) \quad (7.3)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \quad (7.4)$$

$$d = R \cdot c \quad (7.5)$$

Tato rovnice vypočítá poměrně přesně vzdálenost mezi souřadnicemi na zemském povrchu v 2D zakřiveném prostoru. Rovnice uvažuje Zemi jako sférické těleso, což není přesné, protože Země se blíží tělesu zvanému elipsoid. Další nepřesnost rovnice spočívá v neuvažování změn nadmořské výšky bodu. Například při cestě skrz několik pohoří nebo cestě s velkou změnou nadmořské výšky může docházet ke zvětšeným nepřesnostem. K všednímu využití je však tento druh výpočtu vzdálenosti dvou bodů ideální.

Přijatá data se dále ukládají do souboru typu GPX (přípona .gpx), který se používá pro archivaci geografických dat. Tento formát umí načíst řada internetových aplikací, převádějících takto načtené data přes některou databázi map do vizuální podoby. Z této je možné si celou trasu prohlédnout, například produktem GPS Visualizer[3]. Tento formát je také možno přes zmíněné webové rozhraní převést do formátu KML (.kml). Trasa cesty uložená ve formátu KML je zobrazitelná například v Google Maps nebo v programu Google Earth. V neposlední řadě je také možné vytvořit si vlastní aplikaci, například využívající Google API, která získaná data bude vizualizovat prostřednictvím některých veřejných map. Ukázka vizualizace přijatých dat prostřednictvím rozhraní webového rozhraní GPS Visualizer[3] je v příloze práce A.1.

Do dalšího souboru typu TXT se ukládají vypočtené vzdálenosti a časy cesty. Tuto informaci nelze ukládat do souboru typu GPX, protože by byla porušena validita a struktura formátu.

Celkový proces tvorby evidence jízd pomocí elektronické knihy jízd popisované v této práci a implementované prostřednictvím této práce je následující:

1. Spuštění implementovaného serveru na určitém počítači
2. Nastavení a spuštění jedné z klientských aplikací na fyzickém zařízení
3. Zahájení služební cesty ve vozidle
4. Server zpracovává data o pozici vozidla v průběhu cesty
5. Ukončení aplikace a odpojení od serveru
6. Server uloží informace o cestě a její přesná data do formátů GPX a TXT
7. Řidič nebo osoba pověřená si s odstupem času například prohlédne trasy jízd, vytiskne detaily jízd, vypočítá náklady na pohonné hmoty atd.<sup>5</sup>

---

<sup>5</sup>Samozřejmě musí být nějakým způsobem uvedeny všechny náležitosti pro psaní evidence jízd dle zákonného pokynu D-300, odstavce 24

# Kapitola 8

## Závěr

Výsledkem této bakalářské práce je navržení a implementace jednoduchého systému pracujícího jako online elektronická kniha jízd. Práce je více zaměřena na detailní návrh a implementaci dvou nezávislých klientských aplikací pro zařízení iPhone a libovolných zařízení s operačním systémem Windows Mobile. Součástí je i serverová aplikace pro zpracování a reprezentaci dat. Výstup systému ve formě databáze je možné automatizovat k využití v praxi přidáním určitých implementačních detailů podle zákonného pokynu D-300, odstavce 24, části o náležitostech evidence jízd (kap. 2). Funkce systému spočívá v odesílání dat o pozici vozidla pomocí GPS hardwaru zařízení iPhone nebo Windows Mobile umístěného ve vozidle na server, kde se data zpracovávají do určitého formátu dále použitelného např. pro vizualizaci nebo archivaci. Systém je vhodný pro podnikatele například jako náhrada papírové knihy jízd nebo firmy nevyužívající měsíčního zákonného paušálu na dopravu. Další využití by aplikace našla například jako plugin k software se zaměřením na účetnictví, kde by usnadňovala výpočet výdajů za používání služebního vozidla.

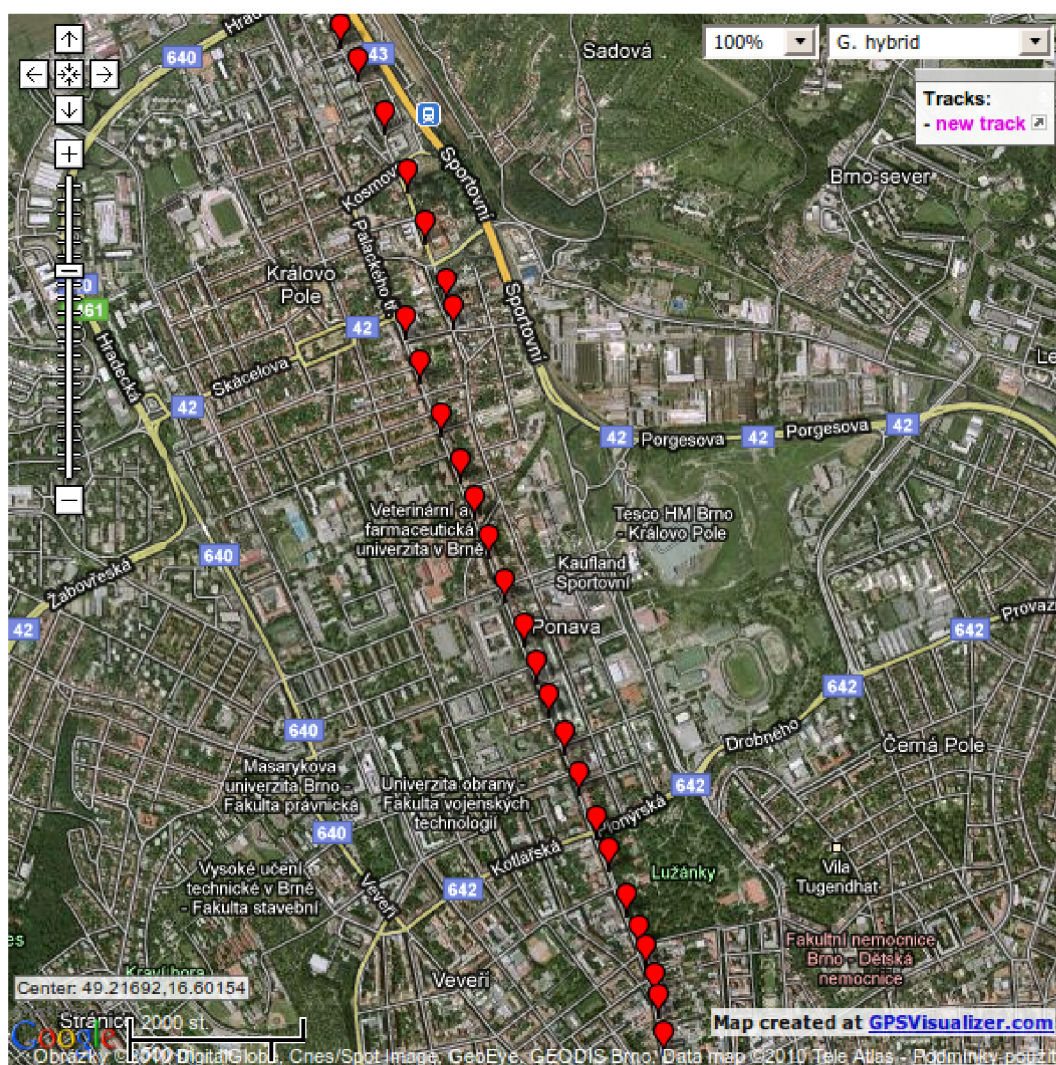
# Literatura

- [1] HTC Touch2 Specification [online].  
<http://www.htc.com/europe/product/touch2/specification.html>,  
[cit. 2010-05-13].
- [2] iPhone History - iPhone 3GS [online].  
<http://www.iphonehistory.com/iphone-3gs/>, [cit. 2010-05-13].
- [3] GPS Visualizer utility [online]. <http://www.gpsvisualizer.com>, [cit. 2010-05-14].
- [4] iPhone Developer Program [online]. <http://developer.apple.com>, [cit. 2010-05-14].
- [5] Visual Studio 2008 Professional [online].  
<http://www.microsoft.com/cze/msdn/produkty/vstudio/professional/default.aspx>,  
[cit. 2010-05-14].
- [6] David Halliday, J. W., Robert Resnick: *FYZIKA Vysokoškolská učebnice obecné fyziky*. Vysoké učení technické v Brně, nakladatelství VUTIUM, 2000,  
ISBN 80-214-1869-9.
- [7] Hirschelmann, K. H.: NMEA 0183 [online]. <http://www.kh-gps.de/nmea-faq.htm>,  
2007-08-07 [cit. 2010-05-13].
- [8] Hojgr, R.; Stankovič, J.: *GPS praktická uživatelská příručka*. Computer Press, 2007,  
ISBN 80-251-1734-0.
- [9] Ltd, M. T.: Movable Type Scripts [online].  
<http://www.movable-type.co.uk/scripts/latlong.html>, [cit. 2010-05-14].
- [10] Ministerstvo financí ČR: Zákon č. 586/1992 Sb., o daních z příjmů [online].  
<http://business.center.cz/business/pravo/zakony/dprij/cast3.aspx>,  
1992-11-20 [cit. 2010-05-14].
- [11] Ministerstvo financí ČR: Pokyn č. D-300 - uplatňování zákona o daních z příjmů  
[online]. <http://business.center.cz/business/finance/dane/d-300.aspx>,  
2006-11-12 [cit. 2010-05-14].
- [12] Oliveira Camargo, Paulo de: Quality of TEC Estimated with Mod Ion Using GPS and  
GLONASS Data. *Mathematical Problems in Engineering*, Article ID 794578, 2009.
- [13] Snively, J.: GPS History - How it all started [online].  
<http://www.maps-gps-info.com/gps-history.html>, 2009-03-12 [cit. 2010-05-12].

- [14] Woreck, D.: The Galileo Project: European plans for global navigation supremacy. <http://www.wsws.org/articles/2007/jul2007/gali-j07.shtml>, 2007-7-7 [cit. 2010-05-14].
- [15] Zunk, A.: A brief history of Windows Mobile [online]. <http://www.notebooks.com/2010/04/12/a-brief-history-of-windows-mobile/>, 2010-04-12 [cit. 2010-05-13].
- [16] Šunkevič, M.: Ruský globální družicový navigační systém GLONASS [online]. <http://www.czechspace.cz/cs/galileo/aktuality-GPS-Glonass/GLONASS>, 2007-01-08 [cit. 2010-05-12].

## Příloha A

# Vizualizovaná data



Obrázek A.1: Příloha č.1 - Vizualizace dat pomocí služby GPS Visualizer[3]

## Příloha B

### Obsah CD

1. Zdrojové kódy klientské aplikace pro Apple iPhone (jazyk Objective-C)
2. Zdrojové kódy klientské aplikace využívající OS Windows Mobile (jazyk C#)
3. Zdrojové kódy serverové aplikace (jazyk C++)
4. Zdrojové kódy bakalářské práce v jazyce Latex