



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

ZPRACOVÁNÍ VÝSTUPU ZE ZARÍZENÍ GPS FORMOU TRÉNINKOVÉHO DENÍKU

PROCESSING THE OUTPUT FROM THE GPS DEVICE IN THE FORM
OF TRAINING DIARY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JAN ZHÁŇAL

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JAN ROUPEC, Ph.D.

BRNO 2012

Zadání závěrečné práce

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2011/2012

ZADÁNÍ DIPLOMOVÉ PRÁCE

student: Bc. Jan Zháňal

který studuje v **magisterském navazujícím studijním programu**

obor: **Aplikovaná informatika a řízení (3902T001)**

Ředitel ústavu Vám v souladu se zákonem c.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Zpracování výstupu ze zařízení GPS formou tréninkového deníku

v anglickém jazyce:

Processing the Output from the GPS Device in the Form of Training Diary

Stručná charakteristika problematiky úkolu:

Data zaznamenaná GPS zařízením lze zpracovat různými způsoby. V rámci této práce bude na základě těchto dat vytvořen tréninkový deník vytrvalostního sportovce se zaměřením na orientační běh.

Cíle diplomové práce:

1. Popis GPS zařízení, jeho možností, rozhraní.
2. Analýza potřebných údajů pro tvorbu tréninkového deníku
3. Návrh datových struktur pro uložení dat získaných z GPS zařízení
4. Návrh výstupu aplikace
5. Tvorba aplikace ve vhodném programovacím jazyce

Seznam odborné literatury:

1. Konecký, J.: Měření vzdáleností a plochy pomocí GPS, VUT v Brně, diplomová práce, Brno, 2009.
2. Leick, A. GPS satellite surveying. 3rd ed.. Hoboken, Wiley and Sons, 435 s, 2004.
3. Xu, G. GPS: theory, algorithms and applications. Berlin; Heidelberg; New York, Springer, 315 s., 2003.
4. Neumann, G.: Trénink pod kontrolou: metody, kontrola a vyhodnocení vytrvalostního tréninku. Grada Publishing a.s., 2005.

Vedoucí bakalářské práce: Ing. Jan Roupec, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2011/2012.

V Brně, dne 27. 2. 2012

L. S.

Ing. Jan Roupec, Ph.D.
Ředitel ústavu

prof. RNDr. Miroslav Doupovec, CSc.
Děkan fakulty

Abstrakt

Diplomová práce se zabývá zpracováním výstupu ze záznamníku GPS pozice. Tento zaznamenaný soubor se poté stává hlavní součástí aplikace Tréninkový deník. V textu lze nalézt zhodnocení GPS jako celku, dále popis použitých přístrojů. Část se zabývá použitím moderního prostředí Google Maps API. Program je popsán v závěru práce a je přiložen na CD nosiči.

Abstract

The Diploma Thesis deals with the processing of the output from a GPS position logger. The recorded file will then become main part of application Training diary. There can be found evaluation of GPS as a system as well as a description of used instruments in the text. Part deals with the use of modern environment Google Maps API. The program is described at the end of work and is attached on a CD.

Klíčová slova

Tréninkový deník, GPS, Google Maps

Keywords

Training diary, GPS, Google Maps

Prohlášení o originalitě

Prohlašuji, že jsem práci vypracoval samostatně, dle rad a pokynů vedoucího práce, za použití uvedených literárních pramenů.

Bibliografická citace

ZHÁŇAL, J. Zpracování výstupu ze zařízení GPS formou tréninkového deníku. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2012. 59 s. Vedoucí diplomové práce Ing. Jan Roupec, Ph.D.

Poděkování

Tímto bych chtěl poděkovat vedoucímu práce panu Ing. Janu Roupcovi, Ph.D. za trpělivost, kterou se mnou měl.

Obsah

Obsah	11
1. Úvod	13
2. Popis zařízení pro určení polohy	15
2.1 Princip určování polohy pomocí družic	15
2.2 Historie[5]	16
2.3 Struktura GPS systému	17
2.4 Data přenášená mezi vysílačem a přijímačem [2].....	19
2.5 Přesnost GPS pozice [2].....	21
2.6 Přesnost měření [1]	21
2.7 Modernizace	23
2.8 Systém Galileo [9].....	24
2.9 Použité přijímače GPS signálu.....	26
2.10 Výstup GPS zařízení.....	29
3. Tréninkový deník.....	31
3.1 Orientační běh [16]	31
3.2 Trénink	31
3.3 Obsah tréninkového deníku.....	34
4. Návrh aplikace.....	37
4.1 Návrh datové struktury.....	37
4.2 Práce se soubory[18]	37
4.3 ADO. NET [18].....	38
4.4 Navržená databáze	41
4.5 Mapové podklady.....	42
4.6 Google maps API [19]	43
4.7 Zpracování XML souborů.....	46
4.8 Návrh výstupů aplikace.....	47
5. Aplikace – tréninkový deník	49
5.1 Vzorce a výpočty použité v programu	49
5.2 Výstup GPS zařízení	50
5.3 Zobrazení zaznamenaného souboru na mapě.....	52
5.4 Uživatelské rozhraní.....	54
5.5 Instalace aplikace	55
6. Závěr.....	57
Literatura	59

1. Úvod

Ve své diplomové práci jsem se zabýval problematikou GPS přístrojů, dat, která z nich lze získat a jejich následným zpracováním. Tyto vědomosti jsem poté uplatnil při tvorbě aplikace „Tréninkový deník orientačního běžce“.

K výběru tématu této diplomové práce mě vedl můj zájem o technologii GPS a její nyní již téměř běžné využití v mojí oblíbené aktivitě – orientačním běhu. Použití GPS je v současné době u nás již běžnou záležitostí pro většinu obyvatel. Většinou pro navigaci a občas i pro záznam absolvované trasy. V orientačním běhu se jako v jednom z prvních sportovních odvětví začaly tyto zařízení používat pro analýzu výkonu a možnosti jeho zlepšení. Proto jsem se rozhodl tento úkon svým kolegům touto aplikací zjednodušit.

Hlavními cíli této práce se tedy staly nastudování materiálů dostupných pro správné pochopení funkce GPS systému, zjištění potřeb analýzy tréninku vytrvalostního sportovce a orientačního běžce zvláště. Samostatnou mnohdy podceňovanou kapitolou je navržení aplikace, tak aby byla funkční a pro uživatele co možná nejjednodušší. Poslední částí je samostatná tvorba aplikace v jazycích C# a JavaScript.

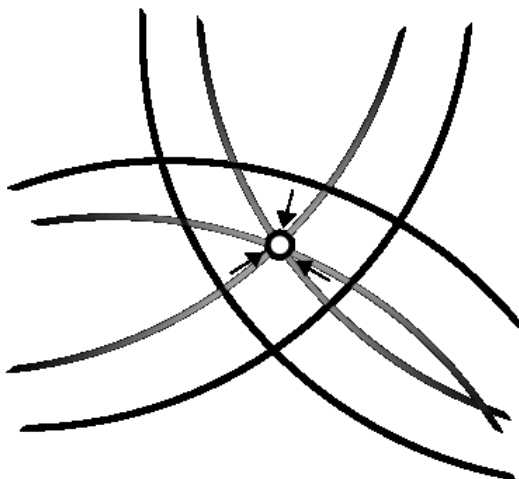
2. Popis zařízení pro určení polohy

2.1 Princip určování polohy pomocí družic

Dnes nejpoužívanějším fyzikálním principem pro určování polohy je využití rádiových vln. Tyto rádiové systémy jsou většinou tvořeny sítí vysílačů (vysílají navigační signály) a přijímačů, které na základě přijatých signálů vyhodnocují svoji polohu. V našem případě jsou radiomajáky představovány systémem družic na oběžné dráze Země.

Družicové systémy jsou globální – za pomoci konečného počtu družic zajistí pokrytí signálem pro celou Zemi. Hlavní výhodou je, že pomocí jednotného systému souřadnic můžeme určit polohu téměř kdekoli a kdykoli.

2.1.1 Princip určení polohy pomocí rádiových vln



Obr.: 1. Geometrické znázornění hledání polohy pomocí těžiště v průniku tří kružnic.
V reálném prostoru se spíše bude jednat o paraboly.

Existují 4 metody:

- úhломěrná – pomocí směrové antény se měří elevační úhly k několika družicím (případně k jedné s časovými odstupy)
- dopplerovská – využívá Dopplerova posuvu (změna frekvence signálu vyslaného pohybujícím se objektem), přijímač je schopen, na základě frekvence, časových značek a parametrů oběžné dráhy družice, vypočítat svoji polohu v prostoru (vzhledem k využití převážně v námořní dopravě, se většinou výška nepočítá)
- dálkoměrná - měření vzdálenosti mezi bodem, jehož poloha se určuje a radiomajáky. Většinou je měření uspořádáno tak, že se neurčuje přímo vzdálenost, nýbrž doba šíření signálu od radiomajáku k navigačnímu přijímači a z ní se teprve vypočítá vlastní vzdálenost. V případě globálních družicových navigačních systémů se vzdálenost mezi přijímačem a navigačními družicemi určuje tak, že přijímač určuje čas t_{di} , který potřebuje signál k tomu, aby dorazil z navigační družice, nacházející se v místě o

souřadnicích (x_i, y_i, z_i) , do místa měření o souřadnicích (X, Y, Z) rychlosti šíření radiových vln rovné rychlosti světla c . Teoreticky stačí, abychom provedli změření vzdáleností od tří navigačních družic, a můžeme polohu neznámého bodu určit vyřešením soustavy tří rovnic o třech neznámých.

$$t_{d1} \cdot c = \sqrt{(x_1 - X)^2 + (y_1 - Y)^2 + (z_1 - Z)^2}$$

$$t_{d2} \cdot c = \sqrt{(x_2 - X)^2 + (y_2 - Y)^2 + (z_2 - Z)^2}$$

$$t_{d3} \cdot c = \sqrt{(x_3 - X)^2 + (y_3 - Y)^2 + (z_3 - Z)^2}$$

Dálkoměrné signály jednotlivých družic je nezbytné od sebe odlišit. K tomu se používá několik způsobů (kmitočtové dělení, kódové dělení, časové dělení).

- fázová - pokud spočítáme počet vlnových délek radiové vlny, které se nacházejí mezi přijímačem a vysílačem v okamžiku měření, vynásobením vlnovou délkou přijímané radiové vlny zjistíme skutečnou vzdálenost. V praxi není problém určit desetinnou část vlny, ale s určením celočíselného počtu vln jsou problémy. [5]

2.2 Historie[5]

Satelitní navigační systémy a jejich vznik se datuje přibližně do 60. let 20. století. Největší zásluhy na vývoji této technologie mají především armádní zájmy. Prvním takovýmto systémem se stal americký TRANSIT používaný pro sledování polohy plavidel (v roce 1964 uvolněn pro civilní použití). S jeho pomocí bylo možné určit polohu s přesností desítek metrů (v počátcích stovek metrů). Tento systém využíval 3 aktivní – vysílající družice a 3 záložní družice. Provoz tohoto systému byl ukončen v roce 1996, družice však zůstaly na oběžné dráze.

Koncepce GPS byla navržena tak, aby uživatel kdekoli na Zemi mohl přijímat signál z dostatečného počtu vhodně rozmístěných družic. Byly provedeny podrobné studie rozebírající mnoho různých faktorů, jakými jsou například selhání družice, minimální úhel nad obzorem (pět stupňů), vyzařovací úhel antén ($\pm 14.3^\circ$) vychýlení ze stanovené dráhy a jiné. Na základě všech požadavků byl projekt GPS realizován pomocí 24 družic obíhajících v šesti kruhových drahách kolem Země. V každé oběžné dráze se nacházejí čtyři družice. Družice obíhají ve výšce 20 200km nad zemským povrchem.

Počátky vývoje GPS (Global Positioning Systém) se datují do roku 1973, kdy byla zahájena první fáze - vypuštění 4 pokusných družic a začátek vývoje uživatelských zařízení. Do roku 1979 bylo vypuštěno celkem 11 družic. V další fázi byla postavena pozemní řídicí střediska a počet družic se zvýšil na 24 – počáteční operační stav, kdy byly družicemi obsazeny všechny plánované pozice. Rok 1995 se stal významným mezníkem - došlo k oficiálnímu vyhlášení plné operační způsobilosti systému.

Technologie GPS byla na počátku využívána jen jako přesný vojenský lokalizační a navigační prostředek sledování pozic vojenských jednotek, zaměřování cílů, apod.), v 80. letech 20. století americká vláda rozhodla o jeho uvolnění i pro civilní účely. Poté došlo k mohutnému rozšíření technologie GPS do všech oblastí lidské činnosti.

Od 25. března 1990 byla do C/A[6] kódu radiového signálu zanášena umělá chyba. Toto opatření pod názvem Selective Availability (SA) mělo zabránit zneužití např. možnosti navádět balistické rakety, pomocí nepřesných efemerid a časových značek. SA způsobovalo chybu 45 m horizontálně (95% RMS).[6] Tuto chybu bylo možno výrazně potlačit diferenčním měřením nebo dlouhodobým statickým měřením.

Protože USA vyvinuly systém, jak lokálně rušit signál GPS, bylo SA 1. května 2000 **zrušeno** a přesnost kódového měření polohy se tak zvýšila na 10m a s touto přesností pracuje dodnes.

2.3 Struktura GPS systému

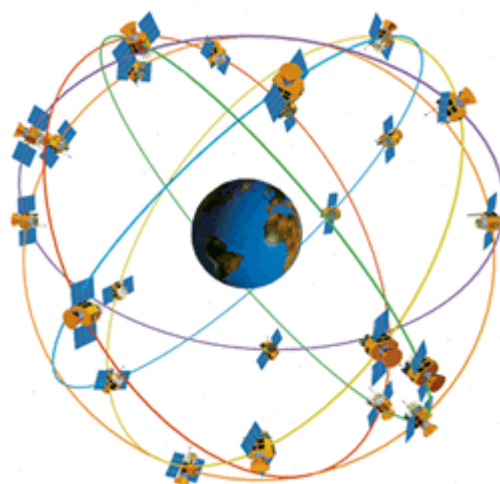
2.3.1 Kosmický segment

Byl projektován na 24 družic, ale nyní je využíván až na maximální počet – 32.

Klíčové části družic GPS (NAVSTAR) jsou: [7]

- 3 až 4 velmi přesné (10^{-13} s) atomové hodiny s rubidiovým oscilátorem
- 12 antén pro vysílání rádiových kódů v pásmu L (2000-1000 MHz)
- antény pro komunikaci s pozemními kontrolními stanicemi v pásmu S (2204,4 MHz)
- antény pro vzájemnou komunikaci družic v pásmu UHF
- solární panely a baterie jako zdroj energie

V České republice je nejčtenější viditelnost 8 družic, minimum pak 6, maximum 12 družic.[8]



Obrázek 2
Schéma drah jednotlivých družic[11]



Obrázek 3 Družice systému NAVSTAR - GPS, 3. generace[11]

2.3.2 Řídící a kontrolní segment[1]

- velitelství, na letecké základně Los Angeles v Kalifornii v USA.
- řídicí středisko, na Schrieverově letecké základně USAF v Colorado Springs,
- 3 povelové stanice, které jsou umístěny na základnách USAF: Kwajalein, Diego Garcia, Ascension Island.
- 18 monitorovacích stanic

Řídící a kontrolní segment monitoruje kosmický segment, zasílá povely družicím, provádí jejich manévry a údržbu atomových hodin. Výsledek jejich monitoringu je zveřejňován v navigační zprávě každé družice a jeho platnost je řádově několik hodin. Pokud by došlo ke zničení pozemních vojenských stanic řídicího a kontrolního segmentu, přechází družice do režimu *AUTONAV* (Autonomous Navigation Mode), ve kterém jsou schopny dále pracovat až 6 měsíců. V tomto režimu spolu družice komunikují a porovnávají vzájemně mezi sebou své efemeridy a stav palubních hodin. Výsledky poskytují uživatelskému segmentu v navigační zprávě. Tento režim však nikdy nenastal, nejsou ani známy výsledky jeho případných testů.

2.3.3 Uživatelský segment[1]

Uživatelé pomocí GPS přijímače přijímají signály z jednotlivých družic, které jsou v danou chvíli nad obzorem. Na základě přijatých dat (časových značek z jednotlivých družic a znalosti jejich polohy) a předem definovaných parametrů přijímač vypočítá polohu antény, nadmořskou výšku a zobrazí přesné datum a čas. Komunikace probíhá pouze od družic k uživateli, GPS přijímač je tedy pasivní.

Rozdělení přijímačů podle přijímaných pásem:

- jednofrekvenční
- dvoufrekvenční
- vícefrekvenční (připravují se pro pásmo L5)

Rozdělení přijímačů podle kanálů:

- jednokanálové (používané v raných fázích projektu GPS)

- vícekanálové

Rozdělení přijímačů podle principu výpočtů:

- kódová
- fázová a kódová

Běžně dostupné přijímače k amatérskému (tj. negeodetickému a nevojenskému) využití se vyrábí jako jednofrekvenční, vícekanálové a kódové. Jednoduchý přijímač signálu GPS se skládá z:

- antény
- předzesilovače
- procesoru
- časové základny (často křemíkový krystal o přesnosti $<10^{-6}$ s)
- komunikačního rozhraní

2.4 Data přenášená mezi vysílačem a přijímačem [2]

2.4.1 Rádiové signály z družic

Družice systému GPS vysílají radiové signály, které umožňují uživatelům určovat svou polohu a čas. Původní plán výstavby systému GPS počítal se dvěma různými kódy:

- **C/A kód** (Coarse / Acquisition code), který je veřejně dostupný
- **P kód** (Precision code), který je přístupný jen autorizovaným uživatelům

a pro modernizaci na GPS III se připravují:

- **C kód** (Civilian code), který bude veřejně dostupný
- **M kód** (Military code), který bude přístupný jen autorizovaným uživatelům

2.4.2 C/A kód

Každá družice generuje navigační zprávu s frekvencí 50 Hz na základě periodicky přijímaných dat z řídicího segmentu a binárně ji sčítá (operace modulo 2) s dálkoměrným kódem C/A, který má frekvenci 1,023 MHz. Vzniká tak signál s rozprostřeným spektrem. Ten je poté modulován na nosnou frekvenci L1 1575,42 MHz, čímž vznikne výsledný signál vhodný pro přenos radiovou cestou, který je pomocí pole spirálových antén s pravotočivou polarizací vysílán směrem k Zemi. Každá družice má vlastní unikátní kód C/A, což umožňuje rozlišit jednotlivé družicové konstelace. Jedná se tedy o kódové dělení CDM (Code Division Multiplex).

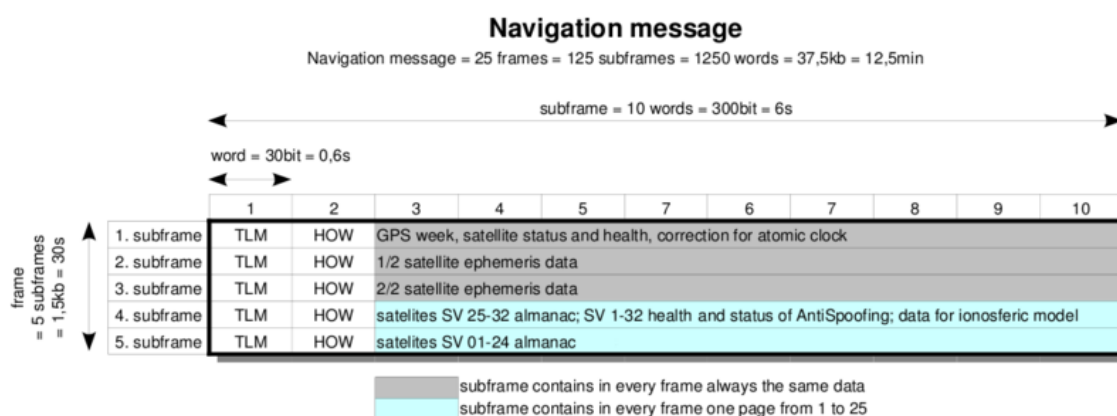
Obdobně družice sčítají navigační zprávu i s druhým dálkoměrným kódem označovaným jako P kód, který má frekvenci desetkrát vyšší než C/A kód, tedy 10,23 MHz. Ten moduluje nejenom nosnou frekvenci L1 jako C/A kód, ale také nosnou frekvenci L2 1227,6 MHz.

2.4.3 Navigační zpráva

Navigační zpráva jsou vlastně užitečná data vysílaná družicí potřebná pro stanovení přesného času a polohy uživatele. Zpráva však neobsahuje přímo polohu družice, nýbrž parametry její dráhy a další nezbytné informace potřebné pro výpočty polohy, rychlosti a času, které se provádí v přijímači.

Protože navigační zpráva je generována s frekvencí pouze 50 Hz (bitový tok 50 bit/s), je přenášena spolu s dálkoměrným kódem pomocí techniky rozprostřeného spektra, tzn., že jeden bit navigační zprávy je přenášen pomocí mnoha bitů rozprostírací sekvence – tento poměr se nazývá rozprostírací faktor SF (Spreading Factor).

Pokud do vztahu dosadíme frekvence dálkoměrných kódů a navigační zprávy, vypočteme rozprostírací faktory $SFC/A=20460$ a $SFP=204600$, tzn., že jeden bit navigační zprávy je přenášen 20460 bity C/A kódu, resp. 204600 bity P kódu. Dekódování zprávy v přijímači je prováděno opačným způsobem.



Obrázek 4 Navigační zpráva [3]

Navigační zpráva, která je zobrazena na obr. 4, se skládá z 25 rámců a její přenos trvá 12,5 minuty. Každý rámeček trvající 30 sekund obsahuje 1500 bitů rozdělených do 5 šestisekundových podrámčů po 300 bitech. Každý podrámec se dále dělí na deset slov.

Slovo trvá 0,6 sekundy a skládá se ze 30 bitů, z nichž 6 slouží pro zabezpečení přenosu pomocí Hammingova kódu. Ten umožňuje detekci 3 chyb a opravu jedné. Význam prvních dvou slov je u všech podrámčů stejný. Slovo TLM (Telemetry Word) se stará o synchronizaci podrámce a také nese telemetrickou informaci určenou pro řídicí segment. Slovo HOW (HandOver Word) obsahuje informaci o pořadí podrámce od začátku týdne, o čísle podrámce v rámci a o tom, zdali je zapnut režim A-S. Pokud ano, pak je místo kódu P použit kód Y.

Zbývá slova 1. podrámce obsahují detailní informace o stavu družice, údaje o týdnu GPS a další informace potřebné pro korekci času. Obsahem 2. a 3. podrámce jsou efemeridy družice, pomocí kterých lze určit polohu družice v daném okamžiku.

5. podrámec a část 4. jsou vyhrazeny pro almanach. Almanach nese méně přesné informace o poloze a stavu ostatních družic – to umožňuje při příjmu signálu alespoň jedné družice snáze vyhledávat ostatní. 4. podrámec obsahuje také údaje, pomocí nichž lze potlačit zpoždění signálu způsobené ionosférou.

2.5 Přesnost GPS pozice [2]

Přesnost GPS pozice závisí nejen na schopnostech měření doby letu signálů, ale také na množství viditelných satelitů a jejich geometrické konfiguraci. Představíte-li si 2D případ, kdy by výsledná pozice byla průnikem tří kružnic (známe jejich středy — pozice satelitů, a relativní rozdíly poloměrů). Naše měření jsou však nepřesná a tak místo kružnic počítáme průsečíky tří *mezikruží*. Výsledný průnik má pak nenulový obsah a jeho velikost záleží jak na chybě jednotlivých měření, tak na vzájemné poloze satelitů.

Chceme-li tuto geometrickou nepřesnost nějak uchopit, tak k popisu použijeme místo kružnic první členy Taylorovy řady v okolí bodu blízkého očekávaného průsečíku R_0 . Průsečík *mezikruží* tak nahradíme průsečíkem *pásů*. Matematicky pak můžeme úlohu popsat pomocí matic, kde nás, za předpokladu nekorelovaných chyb měření, zajímá pouze matice C , jejíž řádky jsou normované vektory (satelit, R_0). Výsledná kovarianční matice $P = \sigma^2 * (C^T * C)^{-1}$.

Posledním krokem je určení chyby v místní souřadné soustavě (zeměpisná délka, šířka a výška). To lze provést transformací souřadnic vektorů pro matici C , a pak členy na diagonále (rozptyly chyb času a polohových souřadnic σ_{Sever} , $\sigma_{\text{Východ}}$, $\sigma_{\text{Výška}}$) odpovídají PDOP (odmocnina se součtu čtverců σ_{Sever} , $\sigma_{\text{Východ}}$ a $\sigma_{\text{Výška}}$), HDOP (odmocnina ze součtu čtverců σ_{Sever} a $\sigma_{\text{Východ}}$) a VDOP ($\sigma_{\text{Výška}}$).

2.6 Přesnost měření [1]

Přesnost výpočtu polohy přijímače podléhá vlivům, které vnáší do výpočtu chyby (tyto chyby jsou náhodné veličiny). Velikost chyby popisujeme statistickým parametrem efektivní hodnota chyby (RMS), což je odmocnina z průměru kvadrátu chyby:

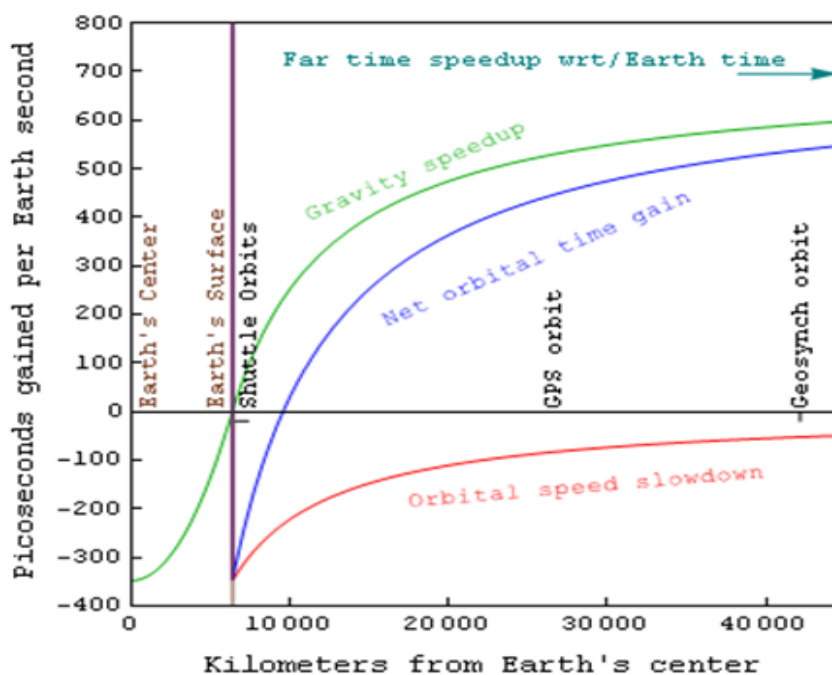
$$RMS = \sqrt{\sum_{i=1}^n \frac{chyba^2}{n}} = E(X^2)$$

Příčina	Velikost RMS
Efemeridy družic	± 2,1m
Družicové hodiny	± 2,1m
Ionosférická refrakce	± 4,0m
Troposférické refrakce	± 0,7m
Vícecestné šíření signálu	± 1,4m
Přijímač	± 0,5m

2.6.1 Efemeridy

Jsou to předpovídané polohy družic na oběžných drahách. Jsou dobře matematicky popsitelné. Přesto se vlivem kolísání tíhových sil Země, Slunce a Měsíce a sluneční jaderné aktivity jejich dráha mírně mění.

2.6.2 Družicové hodiny a relativistické efekty[10]



Obrázek 5 Graf vlivu relativistických efektů na čas v blízkosti gravitačního pole Země [10]

Podle Einsteinova principu relativity lze pro družice NAVSTAR GPS na orbitu, vztažené k Zemi jako inerciálnímu referenčnímu systému, očekávat efekty ovlivňující palubní hodiny.

- pohybová rychlost družice: hodiny se zpomalují o $-5 \times 10^{-9}\%$ vůči pozemským
- rozdílné gravitační potenciály: ve značné vzdálenosti nad Zemí, protože intenzita gravitačního pole klesá s druhou mocninou vzdálenosti (pro orbit družic 16×): hodiny se zrychlují $+50 \times 10^{-9}\%$

Výsledkem je rozdíl $+45,5 \times 10^{-9}\%$ oproti pozemským hodinám. Řešení tohoto efektu je rozdílné hardwarové nastavení základní frekvence oproti zemským hodinám.

2.6.3 Ionosférická a troposférická refrakce

Radiový signál vysílaný z družice se ve vzdálenosti 500-20200km od povrchu šíří vakuem. Ionosféra, která se nachází v rozmezí 50-500km nad povrchem země, obsahuje množství volných elektronů a iontů, které způsobují refrakci (lom) rádiového signálu, tedy jeho delší dráhu a zpoždění. Stav ionosféry ovlivňuje aktuální i cyklická (11 let) aktivita slunce, globální meteorologické vlivy, roční období, fáze dne.

Obdobný vliv má troposféra, která se nachází od 0-15km. Její stav ovlivňují především lokální meteorologické vlivy jako je teplota, tlak, vlhkost. Její možnost predikce je pro globální systém minimální a eliminuje se diferenčními systémy nebo lokálními modely.

2.6.4 Vícecestné šíření signálu a přijímač

Pokud je anténa přijímače částečně zastíněna, nebo jsou v blízkosti odrazivé materiály, existuje možnost, že přijímá také signály odražené a tedy opožděné. Velikost této chyby je závislá na vlastnostech okolí a míře zastínění.

Konstrukce přijímače dříve výrazně ovlivňovala měření kvůli malému počtu kanálů, snížené přesnosti u 8bitových procesorů a malé citlivosti na vstupu. Dnes má vliv především metodika výpočtu, kdy jsou do algoritmu vnášeny předpoklady podle způsobu využití přijímače (doprava, letectví, turistika) a vlastnosti antény, její konstrukce a umístění.

2.6.5 Geometrické rozmístění družic[11]

Chybu měření výrazně ovlivňuje rozmístění družic na hemisféře a obecně se nazývá **DOP** (Dilution of Precision, rozptyl přesnosti). Souhrnný GDOP z intervalu 1-50 nabývá v našich zeměpisných šířkách a nadmořských výškách hodnot 1-4.

2.6.6 Selektivní dostupnost (Selective Availability)

Jak jsem se již zmínil od 25. března 1990, byla do C/A kódu radiového signálu zanášena umělá chyba. Toto opatření pod názvem **Selective Availability** (SA) mělo zabránit zneužití např. možnosti navádět balistické rakety, pomocí nepřesných efemerid a časových značek. SA způsobovalo chybu cca 45 m horizontálně. SA bylo zrušeno 1. května 2000. [6]

Přijímač vybírá čtveřici družic s nejlepším rozmístěním. Proto na volném prostranství bývá určení polohy přesnější, než když je výhled na oblohu částečně zakryt. Při výpočtu pozice zjišťujeme tři souřadnice a rozmístění družic způsobuje, že pro každou souřadnici vychází jiná odchylka. Obecně platí, že stanovení výšky je méně přesné než určování zeměpisných souřadnic.

Z dat získaných v průběhu tvorby této práce jsem zjistil (nicméně neověřil výpočtem, neboť to nebylo podstatou této diplomové práce), že všechny použité přístroje počítají na otevřeném prostoru svoji polohu velmi přesně, ale čím je terén členitější (větší kopce), tím nepřesnější jsou zaznamenané souřadnice. Podobné je to např.: v uzavřených uličkách historických center měst, či velmi vysokých budov.

Na samostatnou kapitolu by mohl vyjít pohyb přijímače v lese. Zásadní roli tu totiž stejně jako ve městě hraje odrazení signálu – v nízkém (do výšky ramen) hustém lese, je pozice většinou počítána naprosto správně i přesto, že člověk by zde s určením své polohy mohl mít velký problém. Naopak ve vysokém i řidším listnatém lese bývá odchylka často několik metrů, což je způsobeno právě odražením signálu od listů. Jehličnatý les je pro příjem signálu příznivější (jehličí má menší plochu pro odrazení signálu), takže i výsledky jsou řádově o polovinu lepší.

K této úvaze mě vedly hlavně zkušenosti z orientačního běhu – sportu, kde se využívá přesných map. Při jejich použití lze srovnat pozice člověka v terénu a následným vyčtením GPS přístroje lze zjistit rozdíl.

2.7 Modernizace

Technologie GPS prochází neustálým vývojem z důvodů zvyšujících se požadavků na navigaci a přesnost určení polohy vojenským i civilním sektorem. Další pohnutkou modernizace systému GPS je zcela zřejmě i opětovný pokus o širší uplatnění navigačního systému GLONASS, ale hlavně současná výstavba navigačního systému

Galileo, který bude poskytovat mnohem vyšší přesnost určení polohy než současný systém GPS.

Z hlediska civilního uživatele bude velkým přínosem pro zvýšení přesnosti určení polohy brzké zavedení již výše zmíněného nového civilního C (Civil) kódu, který bude modulovat frekvenci L2 a umožní tak civilním uživatelům měření na dvou frekvencích, což významně eliminuje vliv ionosférické a troposférické refrakce.

Ve stejné době bude zaveden také nový vojenský M (Military) kód s mnohem silnějším šifrováním, který by měl zamezit možnému podvržení nepřítelem. M kód bude vysílán na frekvenci L1 i L2. Zároveň dojde ke zdokonalení technik umožňující odepření přístupu na určitém území například při válečném konfliktu. O něco později je naplánováno vytvoření zcela nové frekvence L5 s kmitočtem 1176,45 MHz, která bude přizpůsobena především potřebám letecké navigace, zejména přistávání letadel.

Zavedení nových frekvencí L1 M, L2 M a L2 C umožní použít modernizované družice bloku IIR-M. Zároveň budou tyto družice disponovat zvýšeným vysílacím výkonem. Jejich koncepce vycházející z bloku IIR však neumožňuje zavést vysílání na frekvenci L5. To umožní až družice bloku IIF, které jsou v současné době vyvíjeny firmou Boeing. Jejich velkou předností bude také zavedení digitální atomových hodin s cesiovým standardem. Tyto procesorem řízené hodiny mohou kompenzovat měnící se podmínky okolního prostředí, provádět diagnostické testy, upravit své vnitřní parametry a zajistit tak co nejpřesnější chod.

To je jedinečná výhoda oproti klasickým atomovým hodinám. Ty po nastavení při výrobě již nelze upravovat a vlivem stárnutí klesá jejich frekvenční stabilita.[11]

2.8 Systém Galileo [9]

Navigační systém Galileo je plánovaný evropský autonomní globální družicový polohový systém (GNSS), který by měl být nezávislou obdobou amerického systému Navstar GPS a ruského systému GLONASS. Jeho výstavbu zajišťují státy Evropské unie prostřednictvím Evropské kosmické agentury (ESA) a dalších institucí.

Projekt byl pojmenován podle toskánského vědce Galilea Galileiho, který se mimo jiné zajímal i o problémy námořní navigace. Administrativní sídlo tohoto systému se od letošního roku nachází v Praze a jeho spuštění je (k 9. 9. 2012) naplánováno na rok 2014.

Základní výhodou tohoto řešení proti GPS a GLONASS je nezávislost na armádním sektoru. Tím je na rozdíl od těchto systémů garantováno, že pokud bude vypnut nebo omezen, jako se to u všech systémů může stát v případě, že se jeho provozovatelé budou cítit vlastním produktem ohroženi, bude o tom uživatel pomocí samotného systému informován.

Druhou podstatnou výhodou je možnost přivolání pomoci. Trosečník vybavený satelitním přijímačem Galileo s příslušnou funkcí bude schopen vyslat nouzový signál, který kosmický systém zachytí a předá speciálnímu středisku pro záchranné operace. To pak zařídí záchrannou akci a zároveň vyšle k nešťastníkovi do jeho přístroje zprávu, že už se o něm ví.

První dvě družice systému Galileo byly vypuštěny 21. října 2011, další 2 poté v první polovině roku 2012.

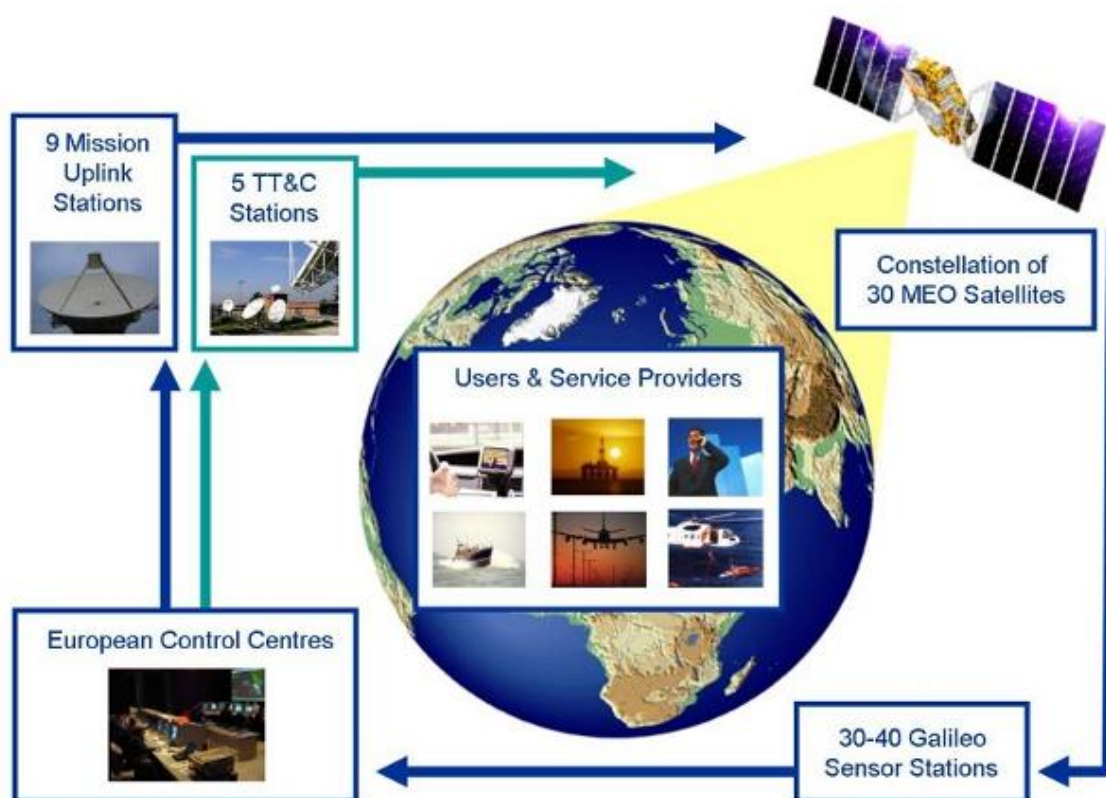


2.8.1 Technické informace [24]

Systém má být tvořen 30 operačními družicemi (27+3), obíhajícími ve výšce přibližně 23 tisíc kilometrů nad povrchem Země po drahách se sklonem 56° k zemskému rovníku ve třech rovinách, vzájemně vůči sobě posunutých o 120° . Každá dráha bude mít 9 pozic pro družice a 1 pozici jako zálohu, aby systém mohl být při selhání družice rychle doplněn na plný počet. Každá družice bude obsahovat atomové hodiny více typů, takže by se vůči předchozím systémům měly změnit i odchylky v čase.

Dále bude na Zemi zprovozněna síť stanic a čidel pro dosažení větší přesnosti (zvláště u komerčních a vyšších služeb systému).

Střední počáteční operační schopnosti systému (zahájení provozu) bude dosaženo ve chvíli, kdy na oběžné dráze bude 18 funkčních satelitů (4 z toho jsou tzv. In-Orbit Validation, které již jsou ve vesmíru a jak název napovídá, mají za úkol hlavně otestování všech systémů).



Obrázek 6 Architektura systému Galileo [24]

2.8.2 Služby systému:

- **Open Service (OS)** bude volně dostupná. Její signály budou využívat 2 pásma: 1164–1214 MHz a 1563–1591 MHz. Přijímače budou mít horizontální přesnost lepší než 4 m a vertikální lepší než 8 m (nebo horizontálně pod 15 m a vertikálně pod 35 m při použití jednoho pásma). Protože bylo dosaženo dohody o kompatibilitě s americkým systémem, budoucí přijímače navíc budou zároveň využívat i GPS.

- **Commercial Service (CS)**, která bude šifrovaná, zpoplatněna a má poskytnout přesnost lepší než OS.
- **Safety of Life Service (SOL)**, která bude šifrovaná s důrazem na integritu a bezpečnost, pro nasazení např. v řízení letového provozu.
- **Search and Rescue (SAR)**, služba nouzové lokalizace v rámci celosvětové družicové záchranné služby COSPAS/SARSAT s možností oboustranné komunikace.
- **Public Regulated Service (PRS)**, která bude šifrovaná, s kontrolovaným přístupem a dlouhodobou podporou, určená pro armády a bezpečnostní složky států.

2.9 Použité přijímače GPS signálu

Pro tuto práci jsem využíval několika přijímačů. Většinou se jednalo o sportovně založené modely, jejichž hlavní funkcí je zaznamenání trasy (tzv. logování), z nichž první hodinky vlastním, ostatní přístroje mi byly zapůjčeny. Popíšu zde základní vlastnosti těchto modelů.

2.9.1 Garmin 110 Forerunner [21]



Rozměry zařízení	4.5 cm x 6.9 cm x 1.4 cm
Displej - průměr	2.5 cm
Displej – rozlišení	52 x 30 pix.
Hmotnost	52 g
Vysoce citlivý přijímač GPS	
Výstup pro PC	*.fit soubor ¹

Určení: běžecké hodinky s přijímačem GPS signálu pro záznam a vyhodnocení trasy

Obrázek 7 Hodinky s GPS přijímačem - 110 Forerunner

Hodinky se od řady ostatních sporttesterů liší velmi jednoduchým a přehledným menu s nastavením funkcí. Forerunner 110 je vhodný jako sporttester, který jednoduše změří a na displeji ukáže tempo/rychlost (průměrnou), zdolanou vzdálenost, čas a srdeční tep. Všechny měřené údaje, včetně nadmořské výšky, záznamu proběhnuté trasy či aktuální rychlosti pro každý bod záznamu, ukládá forerunner do své paměti.

¹ Typ souboru, jehož specifikace je ve vlastnictví společnosti Garmin International Inc. Problematice výstupu bude věnována kapitola v dalším průběhu práce

2.9.2 Garmin Forerunner 610



Rozměry	4.5 cm x 6.9 cm x 1.4 cm
Displej - průměr	2.5 cm
Displej – rozlišení	52 x 30 pix.
Hmotnost	52 g
Vysoce citlivý přijímač GPS	
Výstup pro PC	*.gpx soubor

Obrázek 8 Forerunner 610

Vyšší model než první zmíněný Forerunner 110, nicméně veškeré odlišnosti se týkají pouze firmware a ovládání hodinek. GPS čip je v obou případech stejný. Zajímavější jsou hlavně v tom, že z nich lze ihned po (bezdrátovém) připojení k PC získat .gpx soubor, se kterým pracuje aplikace „Tréninkový deník“

2.9.3 Holux GPSport 260



GPS čip	MTK 3329
Displej	128 × 128 bodů
Elektromagnetický kompas	
Barometrický výškoměr	
Krokoměr	
Nemá záznam TF	
Rozhraní: USB	
Výstup: *.gpx soubor	
Určení: Turistický cyklopočítač bez záznamu TF	
=> mírně nevhodný pro trénink.	

Obrázek 9 cyklopočítač s GPS Holux 260

Tento GPS záznamník se ze všech zkoušených modulů vyznačoval nejdelší výdrží na baterie (až 25h), ale jeho gps čip není z nejlepších, takže v těžkém terénu (strmé svahy, listnatý les) dokonce občas ztrácel signál a svou velikostí se také pro běhání příliš nehodil.

2.9.4 HOLUX M-241 Wireless GPS Logger



GPS přijímač: MediaTek MT3318, 32 kanálů
Rozhraní: USB / Bluetooth
Výstup: gpx soubor
Nemá záznam TF

Obrázek 10 Holux GPS Logger

Jednoduchý záznamník polohy. Jeho velkou výhodou je možnost nastavení intervalu zápisu souřadnic, takže při orientačním běhu dosahuje při zapisování každou sekundu velmi pěkného záznamu tratě a hodně vypovídá o chybách závodníka, které při delším intervalu ani nemusí být vidět. Rozměry 30x10x10mm jsou výborné.

2.9.5 GPS logger GT-750



Rozměry: 74 x 44 x 22 mm
Rozhraní: USB, Bluetooth
Paměť: 256 000 bodů tratě

Obrázek 11 GT-750

Bezdrátový GPS logger s přenosovou technologií Bluetooth®. Velmi nevhodný pro trénink orientačního běžece, velmi malá paměť vedla výrobce k velmi dlouhému intervalu mezi zápisem jednotlivých bodů.

Relativně větší rozměry jsou též nevýhodné, jelikož se nevejde do kapsy běžeckých kalhot.

2.10 Výstup GPS zařízení

GPS záznamová zařízení zatím nemají povinně standardní souborový výstup, takže některé firmy používají své vlastní souborové typy (např.: firma Garmin, ale jen u některých zařízení). Nicméně většina těchto typů je nějakým způsobem převeditelná do tzv. GPX souboru. Což je ve skutečnosti jednoduchý XML soubor a jako takový je velmi dobře počítačově zpracovatelný.

Vzhledem k tomu, že jsem převážně pracoval s hodinkami Garmin Forerunner 110, které mají výstupní soubor ve formátu *. FIT, pokusil jsem se získat od této firmy specifikaci tohoto souboru, nicméně mi firmou Garmin International Inc. bylo sděleno, že soubor podléhá firemnímu tajemství a není možno jej šířit. Z tohoto důvodu vzala za své možnost programově přistupovat přímo k připojenému zařízení a vstup do programu probíhá pomocí vložení GPX souboru (u hodinek Forerunner 110 to znamená vyexportovat GPX soubor pomocí připojeného programu).

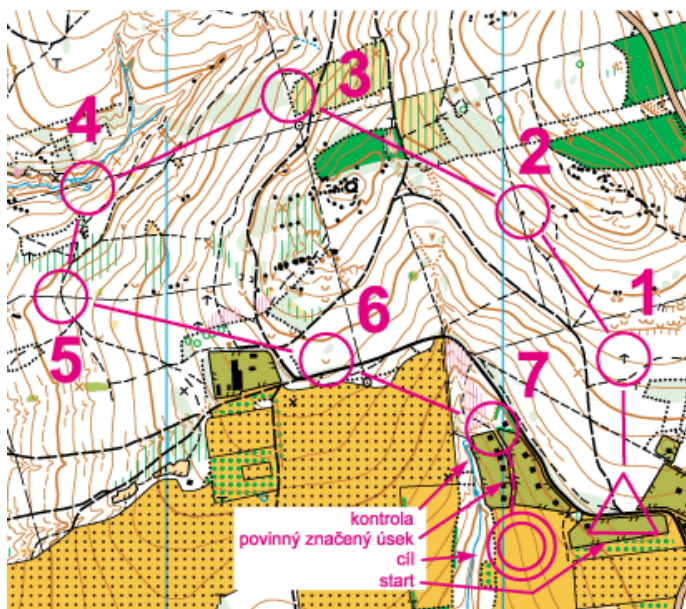
3. Tréninkový deník

Nejdůležitější, při tvorbě deníku, je potřeba vědět pro koho je určen a kdo jej bude analyzovat. Je určen pro evidenci – záznam tréninkového a závodního zatížení sportovce. Poskytuje nám podklady pro vyhodnocení a kontrolu tréninkového procesu, řízení tréninku a plánování.[17]

V zásadě může takovýto deník plnit několik úloh

- Motivační – porovnání dvou sportovců z hlediska „měl bych něco dělat, protože on má odtrénováno víc“
- Analytický – pro trenéra, který svěřenci chystá trénink, je deník naprosto nezbytnou pomůckou, bez něj by nemohl určit, slabiny, silné stránky a už vůbec by nemohl plánovat rozvoj svého svěřence.
- Motivační 2 – porovnání vůči tréninkovému plánu: „v plánu mám napsáno 200km na cyklus, ale zatím mám jen 150, musím se snažit“
- Analytický 2 – pro vylepšení načasování formy do dalších let
- Předcházení zranění, přetrénování

3.1 Orientační běh [16]



Obrázek 12 Mapa pro OB

Orientální běh (zkratka OB) je moderní sportovní odvětví vytrvalostního charakteru, při němž je nutno se správně a rychle orientovat v neznámém terénu. Při závodě se hledají kontrolní stanoviště (kontroly) ve stanoveném pořadí a v nejkratším možném čase. Cestu mezi kontrolami si každý volí podle vlastní úvahy za pomoci mapy, buzoly a stručného popisu kontrol. O úspěchu tedy rozhoduje správná orientace a rychlý běh.[12]

Již z této jednoduché definice lze určit pár základních údajů pro tvorbu tréninkového

deníku: Budeme se zabývat vytrvalostním sportovcem, který navíc potřebuje analyzovat i běh podle mapy.

3.2 Trénink

Vzhledem k tomu, že výstupem této diplomové práce má být aplikace tréninkový deník, rád bych uvedl, že nebude určena rekreačnímu sportovci, protože jednotlivé souhrny a výsledky začínají poskytovat relevantní informace až ve chvíli, kdy mluvíme o systematicky² trénujícím jedinci.

² Za systematický trénink se považuje dodržování předem připraveného plánu tréninků, který obsahuje nejméně 4 tréninkové jednotky týdně

3.2.1 Disciplíny orientačního běhu

Rozlišujeme 3 základní disciplíny:

- Sprint – podle názvu by se mohlo zdát, že jde o běh na velmi krátkou vzdálenost, ale i přesto se jedná o vytrvalostní disciplínu, kdy vítěz ve sprintu běží závod 12-15 minut
- Middle – trať na 35 minut, která by měla být velmi mapově obtížná, uspěje jen ten s precizní prací s mapou
- Long – královská disciplína, úspěch zaručuje jen výborná fyzická připravenost – čas vítěze je okolo 90 minut

Dále se ještě běhají závody štafet, jejichž charakter je ale velmi blízký závodů „middle“ a drobné rozdíly jsou jen v tom, že štafety začínají hromadným startem a závodníci nemají stejnou trať.

Ačkoli se na mezinárodním poli začínají objevovat závodníci se specializací na jednotlivé disciplíny, pro úspěch v republikovém měřítku je v zásadě rozhodující fyzická připravenost podpořená kvalitní prací s mapou. Tzn.: pokud je závodník výborně fyzicky připraven a práce s mapou je jeho slabší stránkou může za přispění štěstí uspět. Pokud ale závodník má práci s mapou výbornou a slabší kondici, bude pravděpodobně podávat kvalitní a vyrovnané výkony, ale bez výraznějšího úspěchu.



Obrázek 13 Orientační běh - závod štafet

3.2.2 Fáze tréninku [4]

Systematický trénink orientačního běžce, ale i dalších vytrvalostních sportovců bývá rozdělen do několika částí:

- přípravná
- předzávodní
- závodní
- odpočinková

Každá z nich se poté dělí do cyklů a ty potom na jednotlivé týdny. Jeden cyklus má 4 týdny. Jeden tréninkový rok potom 13 cyklů. Každý týden v roce je jiný a to, že má 7 dní neznámá, že bude mít i 7 fází. Pro analýzy tréninku je důležité znát mnoho dílčích údajů.

V **přípravném období** se sportovec zaměřuje na budování kapacity oběhového systému. Ze začátku si zvyká a postupem času se dostává k dlouhým tréninkovým fázím, kterých ale nebývá více než 10 týdně. Součástí tohoto období je i potřeba značné fyzické regenerace, která potom má velký vliv na další tréninky a je proto vedle samotného trénování druhou nejsledovanější částí. Celé toto období by mělo být

charakterizováno pestrostí výběru tréninkových prostředků. Mimo běhu je to jízda na kole, lyžích, posilování, bruslení, plavání, spinning, chůze po horách apod.

Předzávodní období se vyznačuje zkracováním jednotlivých tréninků a zvyšováním intenzity i četnosti fází. Hlavním tréninkovým prostředkem se stává běh. Také je třeba začít trénovat mapovou techniku, tzn.: jeden z pomalejších tréninků se odehrává s mapou v ruce, vzhledem k délce dnů v zimě bývá tato fáze často noční.

Závodní období – jsou dvě první na jaře (květen – červen) druhé na podzim (září – říjen), mezidobí je vyplněno zkrácenou verzí odpočinek – příprava – předzávod. Snaha o udržení maximální výkonnosti, Minimum rušivých vlivů (např. cyklistika zpomaluje či odkládá nástup běžecké formy)

Odpočinek: 1-2 podzimní měsíce (ale také cca 8 - 14 dní po jarním závodním období), kdy se tělo zotaví z náročného roku. Neznamená to ale, že člověk jen sedí doma, naopak, měl by mít co nejpestřejší pohyb, zapojit pokud možno, co nejvíce svalových partií, aby napravil tréninkem vzniklé nerovnoměrnosti.

3.2.3 Tréninkový cyklus

Pro orientační běžce, ale většinou i ostatní vytrvalostní sportovce to znamená jeden celý rok.

	A	B	C	D	Σ km	Σ hod.	posilování	mapa
1	27	29	35	37	128	11	1,5	1
2	36	40	53	36	165	15	2,5	1
3	53	31	47	53	184	16	2,5	1
4	52	55	68	37	212	19	3	1
5	51	57	40	45	193	17	3	6
6	31	42	53	39	165	15	2,5	9
7	40	45	36	36	157	14	2	9
8	36	36	36	39	147	13	1,5	9
9	28	38	47	33	146	13	1	9
10	28	33	43	52	156	14	1,5	10
11	48	48	44	26	166	15	1,5	8
12	41	47	35	41	164	15	1	8
13	42	16	16	16	90	8	0	2
Σ					2073	185	23,5	74

Obrázek 14 Ukázka tréninkového cyklu juniorské reprezentantky v orientačním běhu. Čísla 1-13 označují mezocykly. Písmena A, B, C, D jsou poté mikrocykly. Hodnoty v tabulce říkají, kolik by měl člověk v daném období odtrénovat km a hodin.

3.2.4 Tréninkové mezocykly

Jsou to části ročního tréninku, z nichž každá je jiná a řídí se tím, ve kterém období a jeho části se sportovec nachází. Možná délka tréninkového mezocyklu je 2 – 4 – 6 týdnů. Jsou to nejdůležitější části při zpětné analýze výkonů v průběhu celého roku.

	I. týden	II. týden	III. týden	IV. týden
Pondělí	Výklus 40'	A-tunel	V 40'	VT Velikonoce
Úterý	tělocvična	S 55' (cesty) tělocvična	R 7 x 5'	volno nebo výklus
Středa	Volně 40'	T 7 x 6'	V 80' (cesty)	mapa Bílý potok
Čtvrtek	aerobic	aerobic	VT Velikonoce	dráha (úseky + posilování)
Pátek	Střídačka 35' (R2', S3')	volno		V 40'
Sobota	vytrvalost 100' (velké cesty)	Cena sporka		M Moravy - klasika
Neděle	T 35' (cesty, pěšiny)	Cena sporka		M Moravy - štafety

Obrázek 15 Ukázka tréninkového mezocyklu (4 týdny) s jednotlivými mikrocykly (každý týden), takto vypadá plán závodníka mužské elity začínajícího trénovat po zranění.

3.2.5 Tréninkový mikrocyklus

Má rozhodující úlohu při realizaci tréninkového procesu, ale už není tak důležitá pro pozdější analýzu výkonu.

3.2.6 Tréninková jednotka

Jedna fáze tréninku (za jeden den je možné absolvovat více tréninkových či závodních jednotek).

3.2.7 Analýza tréninku

Pro samotnou aplikaci „Tréninkový deník“ je jednotka důležitá z toho hlediska, že sportovec zapisuje, jakou konal aktivitu, jak dlouho a jak intenzivně. Při použití pro nalezení optimální výkonnosti v jednotlivých částech roku jsou důležité mezocykly. Pro řešení vrcholové formy je třeba využívat mikrocyklů. Což ale využívají převážně vrcholoví sportovci při přípravě na MS, OH, atd.

3.3 Obsah tréninkového deníku

Tréninkový deník by měl obsahovat kromě historie tréninkových jednotek i několik dalších důležitých údajů:

- Cíle závodníka
- Váhu průběžně v každém týdnu
- Informace o rozcvičení před každou jednotkou (stačí popis)
- Popis tréninku, délku (minutově, kilometrově), pokud je k dispozici, informace o tepové frekvenci
- Regeneraci

- Informace o tom „jak to šlo“
- Rozbor absolvovaných závodů včetně chyb, ztráty na vítěze
- Volitelně doporučuji i poznámku o počasí a stavu podkladu (např.: bahno, sníh,...)
- Atletické ale i jiné osobní rekordy

4. Návrh aplikace

Samotná aplikace má několik částí, jimž se budu věnovat postupně. Jako první vznikl program pro osobní počítače, který je ale nyní převáděn na serverovou variantu s přístupem přes webové rozhraní.

4.1 Návrh datové struktury

Vzhledem k předpokládanému využití vytvořeného programu, byla jako nejvhodnější možnost ukládání dat zvolena relační databáze. Očekávaný počet uživatelů se blíží 100 a u každého z nich je pravděpodobný počet cca 400 záznamů tréninků. Což znamená každý rok přibližně 40000 záznamů.

Samotný návrh byl ztížen tím, že v počátcích projektu bylo počítáno s použitím databáze MySQL, ale v průběhu přešel majitel serveru na databázový systém PostgreSQL. Řešením bylo ošetření programové vrstvy tak, aby mohla pracovat s oběma databázovými servery, pro což byla navržena třída databazova.cs. Samotný přístup do databáze je realizován pomocí platformy ADO .NET.

4.2 Práce se soubory[18]

Protože se v průběhu vývoje aplikace vyskytla potřeba kam mimo aplikaci uložit výchozí nastavení, studoval jsem problematiku práce se soubory.

V prostředí MS. NET frameworku jsou všechny vstupně/výstupní operace realizovány pomocí takzvaných datových proudů, neboli streamů. Stream je abstrakce určité sekvence bytů z nějakého zařízení, souboru, paměti nebo soketu TCP/IP. V rámci .NET je takovýto datový proud představován abstraktní třídou Stream. Tato třída je базovou třídou pro všechny datové proudy pracující s konkrétními objekty. Třídy představující konkrétní datové proudy implementují její abstraktní členy potřebné pro uskutečnění požadovaného přenosu dat. To znamená velkou výhodu v podobě zobecnění přístupu k různým objektům, na kterých chceme provádět I/O operace, protože použití tříd je díky společnému rozhraní pořád stejné a tím pádem vůbec není potřeba se zabývat implementačními detaily souvisejícími s přístupem ke konkrétnímu druhu objektu. Všechny třídy související s problematikou vstupně/výstupních operací patří do jmenného prostoru System.IO základní knihovny tříd .NET frameworku. Na jednotlivých implementacích datových proudů se nacházejí implementace metod Read a Write, které použijeme ke čtení bytů respektive k jejich zápisu. Oběma metodám je ve formě parametru předán odkaz na pole bytů, představující buffer (vyrovnávací paměť). V případě metody Write je obsah tohoto bufferu zapsán do datového proudu a v případě metody Read je tomu naopak a data z proudu jsou do bufferu ukládána.

Některé datové proudy podporují náhodný přístup. Podpora náhodného přístupu u proudu znamená, že v dané sekvenci bytů se můžeme nastavit na libovolnou pozici a ne jen číst sekvenci od začátku. To jestli konkrétní proud, který chceme použít pro přístup k objektu, náhodný přístup podporuje, zjistíme přečtením instanční vlastnosti CanSeek. Pokud proud tuto vlastnost podporuje, můžeme požadovanou pozici nastavit pomocí vlastnosti Position.

V případě přístupu k souborům použijeme třídu FileStream.

Kromě implementací tříd TextReader a TextWriter se v základní knihovně tříd .NET frameworku nachází ještě druhá dvojice implementací, kterou je implementace pro ty nejjednodušší druhy úložišť, kterými jsou instance třídy String nebo

StringBuilder. Tyto třídy nesou jméno StringReader a StringWriter. Pro tu první, tedy pro třídu StringReader, je objekt, ze kterého jsou čtena data pomocí rozhraní, které je definováno třídou TextReader, představován instancí třídy String.

Práce s objekty typu StringReader je pro mnoho úloh stejná jako při použití třídy StreamReader a to díky použití rozhraní definované třídou TextReader

4.3 ADO.NET [18]

Tato komponenta je následníkem, technologie společnosti Microsoft s názvem ADO.

Jednou z důležitých vlastností komponenty ADO .NET je, že umožňuje standardizovaným způsobem přistupovat k datům v různých datových zdrojích. Jinak také můžeme říct, že ADO .NET je soubor typů (tříd, rozhraní, výčetů...), které můžeme použít pro přístup k datům v různých datových zdrojích z aplikací pro .NET framework. Typy spojené s komponentou ADO .NET se nacházejí ve jmenných prostorech rozšiřující jmenný prostor System.Data (System.Data.Common, System.Data.SqlClient, System.Data.Odbc, ...).

Možnost přístupu k datům v různých datových zdrojích je v technologii ADO .NET zajištěna tak, že jsou předepsána určitá rozhraní (IDbConnection, IDbCommand, IDataReader, IDataAdapter ...).

Tato rozhraní předepisují funkčnost, kterou musí splňovat jednotliví zprostředkovatelé přístupu ke konkrétnímu datovému zdroji, pro který je zprostředkovatel implementován.

Tito zprostředkovatelé jsou oficiálně nazýváni ADO .NET data providers. Použitím přístupu pomocí rozhraní je dosaženo toho, že je možné psát takový kód, který bude fungovat pro jakýkoli datový zdroj. To znamená, že pokud se změní datový zdroj naší aplikace (např. databáze MS SQL na databázi Oracle) a naše aplikace je napsána s použitím přístupu přes zmíněná obecná rozhraní ADO .NET, chod aplikace zůstane stejný.

Základní knihovna tříd .NET frameworku obsahuje několik základních ADO .NET data providerů, které obsahují třídy implementující výše uvedená rozhraní a to pro následující konkrétní datové zdroje:

- **Databáze Microsoft SQL Server** – jmenný prostor System.Data.SqlClient
- **ODBC datové zdroje** – jmenný prostor System.Data.Odbc
- **OLE DB datové zdroje** – jmenný prostor System.Data.OleDb
- **Databáze Oracle** – jmenný prostor System.Data.Oracle

Další datové zdroje je možno přidat (např.: Mysql,...),

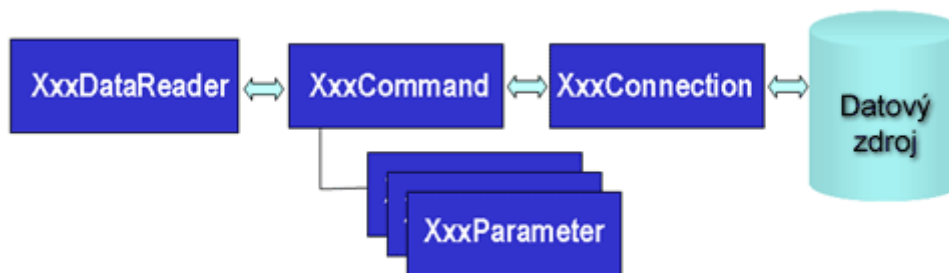
4.3.1 Připojené a odpojené aplikace

Pro tvorbu aplikace komunikující s nějakým datovým zdrojem, lze použít dva scénáře pro přístup k datům. Scénář, kdy je aplikace takzvaně připojena k datovému zdroji (on-line), nebo naopak je od datového zdroje většinu času odpojena (off-line). Každý z těchto scénářů má své výhody a nevýhody, proto se hodí v různých typech situací. Komponenta ADO .NET je navržena tak, aby bylo možné ji použít v obou těchto scénářích.

- **Připojená aplikace**

Tato aplikace má vždy při manipulaci se souvisejícím datovým zdrojem aktivní připojení k tomuto zdroji. To je vhodné v případech, kdy je třeba přistupovat k datům, která jsou často měněna, a v aplikaci je třeba vždy aktuální verze těchto dat. Na druhou stranu to může představovat nevýhodu v podobě zvýšené komunikace mezi aplikační vrstvou a vrstvou datovou.

V případě ADO .NET jsou tyto scénáře řešeny pomocí kombinace jednotlivých implementací rozhraní `IDbConnection`, `IDbCommand` a `IDataReader` pro konkrétní datový zdroj.



Obrázek 16 ADO .NET – připojená aplikace

- **Odpojená aplikace**

Novějším způsobem je použití odpojeného datového zdroje, díky čemuž se aplikace tohoto způsobu užívající, nazývají odpojenými aplikacemi. V tomto případě jsou data získána z datového zdroje do aplikace a poté je spojení s datovým zdrojem ukončeno. Od chvíle odpojení od datového zdroje aplikace manipuluje s obrazem získaných dat, který je uložen v paměti.

Po dokončení potřebných modifikací dat na úrovni aplikační logiky je opět vytvořeno aktivní spojení k datovému zdroji, zjištěny rozdíly mezi daty z aplikace a daty v datovém zdroji a po té jsou tyto rozdíly promítnuty do příslušného datového zdroje. Toto řešení je vhodné pro aplikace, u kterých chceme redukovat komunikaci mezi jednotlivými vrstvami aplikace (aplikační a datovou).

V případě velkého množství uživatelů ale může docházet ke změnovým konfliktům.

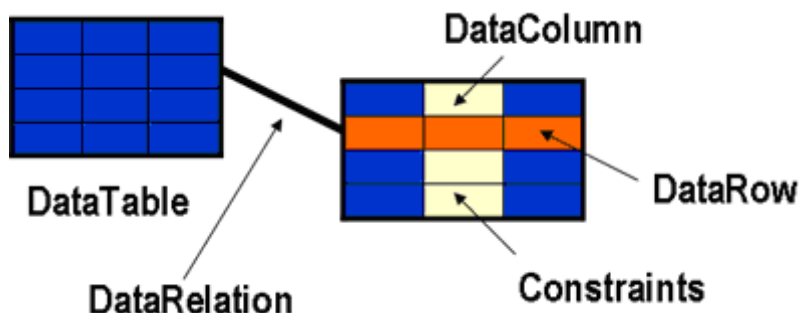
Technologie ADO .NET má pro toto řešení podporu v podobě třídy **DataSet**, která představuje obraz získaných dat v paměti. Instance této třídy je naplněna daty pomocí konkrétní implementace rozhraní `IDataAdapter`, který využívá implementace rozhraní `IDbConnection` a `IDbCommand`



Obrázek 17 DataSet

DataTable[22]

Základním typem pro použití odpojených datových aplikací je DataSet. K tomu, aby mohl DataSet v paměti reprezentovat datový zdroj používá jeden nebo více objektů typu DataTable. Objekty typu DataTable slouží k reprezentaci jedné tabulky dat v paměti.



Obrázek 18 Schéma objektu DataTable

Jak ukazuje obrázek výše, instance třídy DataColumn reprezentují jeden sloupec v instanci třídy DataTable. To znamená, že kolekce těchto sloupců, kterou je typ DataColumnCollection, svázaná s konkrétním objektem DataTable vlastně určuje schéma tabulky, kterou objekt DataTable reprezentuje.

K přidání řádku do tabulky je k dispozici třída DataRow, jejíž instance představují jednotlivé záznamy v objektu DataTable. Stejně jako v případě objektů DataColumn má objekt DataTable interní kolekci k uchovávání instancí třídy DataRow. Tato kolekce je přístupná skrze vlastnost **Rows** objektu DataTable a je typu DataRowCollection..

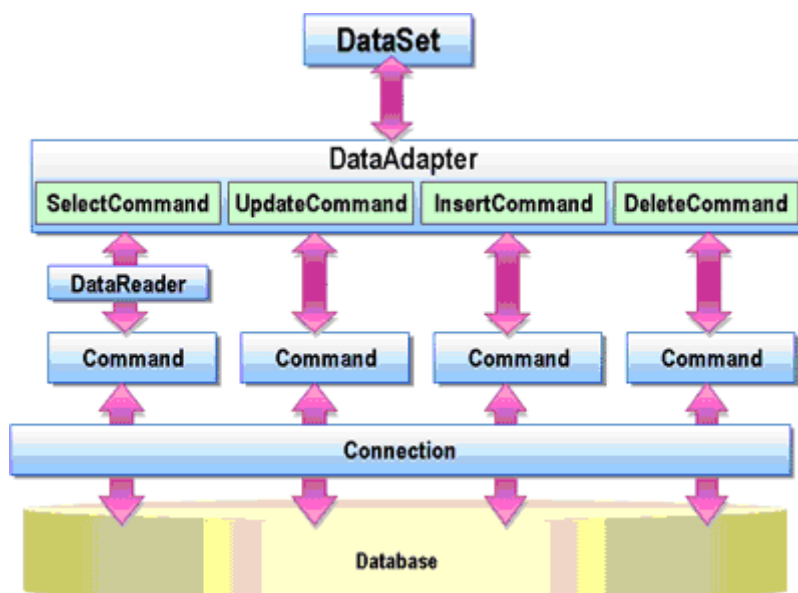
Práce s objekty typu DataRow je rozdílná od práce s objekty typu DataColumn, protože v případě objektu DataRow nevytváříme nové instance přímo použitím konstruktoru tohoto typu, ale odpovědnost za vytvoření nové instance nese objekt DataTable, do něhož chceme záznam přidat pomocí metody NewRow.

DataAdapter[23]

Datový adaptér je implementací rozhraní IDataAdapter. Zařizuje obousměrnou synchronizaci dat na základě svých čtyř příkazů SelectCommand, InsertCommand, UpdateCommand a DeleteCommand. První zmíněný příkaz slouží k definici dotazu, který zajistí naplnění DataSetu daty obsaženými ve vrácené výsledkové sadě. Zbývající tři slouží k synchronizaci druhým směrem, tedy pro přidávání, úpravy a mazání záznamů z datového zdroje na základě dat v DataSetu.

Pro naplnění DataSetu daty je na datovém adaptéru metoda Fill, jíž se předá objekt DataSet, který je třeba naplnit. Tato metoda pro naplnění použije výsledkovou sadu získanou použitím příkazu SelectCommand. Pro synchronizaci směrem k datovému zdroji (přidání, úpravy, mazání záznamů) se používá metoda Update, která zařídí použití příslušných příkazů (InsertCommand, UpdateCommand, DeleteCommand) pro úpravu dat v datovém zdroji.

Datový adaptér při vykonávání své metody Update kontroluje vlastnost RowState jednotlivých řádků (DataRow) v objektu DataTable. Podle hodnoty této vlastnosti provede příslušný příkaz. Takže v případě hodnoty Added provede adaptér pro daný řádek příkaz InsertCommand, pro hodnotu Modified provede příkaz UpdateCommand a pro hodnotu Deleted provede příkaz DeleteCommand.



Obrázek 19 Schéma připojení k databázi a zvýrazněná úloha adaptéru

Definice příkazu `SelectCommand` jednoduše vrátí výsledkovou sadu s kompletním obsahem databázové tabulky. O něco složitější je však definice příkazů ostatních, kde je zapotřebí použít příkazů parametrických.

4.4 Navržená databáze



Obrázek 20 Schéma databáze

Databáze obsahuje 6 tabulek:

- Aktivita (záznam jednotlivých aktivit a jejich detailů, hlavní tabulka databáze)
- Uživatel (tabulka nesoucí záznamy o jednotlivých uživateli)

- Soubory (seznam souborů, které se vážou k jednotlivým aktivitám – mapy, *.GPX)
- Druh_aktiv (pro popsání o jaký druh tréninkové aktivity se jedná)
- Intenzita (seznam intenzit pro jednotlivé aktivity)
- Klub (evidence klubů, používajících tento systém)

4.5 Mapové podklady

Pro zobrazení dat získaných z GPS přístrojů bylo nutno zvolit vhodný mapový podklad. Zde je přehled dostupných řešení vhodných pro použití v našem okolí. A na závěr je uvedeno zhodnocení mého výběru.

4.5.1 Mapy API, Seznam.cz

Mapy poskytované společností Seznam.cz obsahují kvalitní letecký snímek a výbornou mapu ČR, což se ale nedá říci o zbytku Evropy, neobsahují databázi map světa mimo Evropy.

Jejich použití pro tréninkový deník je proto spíše nevhodné (orientační běžec musí velmi často vyjíždět na soustředění a závody nejen po Evropě).

Výhodou tohoto řešení by mohla být oblíbenost u české veřejnosti.

4.5.2 Atlas AMapy, Atlas.cz

Služba společnosti Centrum.cz je na pokraji zastavení (k 1. 2. 2012), nepříliš známé prostředí, které obsahuje relativně kvalitní mapy a snímky ČR, ale slabší mapy EU a dalších zemí. API je velmi podobné Google Maps.

Ve skutečnosti jsem o tomto řešení ani neuvažoval a je uvedeno pouze pro kompletnost práce.

4.5.3 OpenStreetMap Api

V podstatě jediná alternativa k mapovým službám společnosti Google. Mapové podklady jsou vytvářeny uživatelskou komunitou. Přesto, nebo možná proto, je jejich úroveň relativně vysoká. Služba poskytuje podobné možnosti jako ostatní, ale API je odlišné.

Ostatní služby mají JS API, zde je RESTful API, které poskytuje přístup k RAW mapovým datům. Pro použití s typickými vlastnostmi JS API lze využít dalších open source projektů – OpenLayers nebo Mapstraction, které mají knihovny pro zobrazení mapových podkladů z libovolných zdrojů.

4.5.4 Zhodnocení

Po zvážení všech možností jsem se jednoznačně přiklonil k tomu, vytvořit aplikaci s podkladem Google maps.

Protože využití tohoto projektu pravděpodobně nedosáhne zpoplatněných 10 000 zobrazení za den a tudíž bude zdarma. Zároveň tyto podklady obsahují kvalitní letecké snímky i mapu „teréní“. Api a jeho použití by u všech možností znamenalo učit se něco nového a náročnost tohoto úkonu se jen těžko předem posuzuje.

4.6 Google maps API [19]

Společnost Google mimo jiné nabízí mapovou službu - Google Maps (maps.google.com) umožňuje zobrazit mapu a nad ní mnoho dalších možností. Pro uživatele s sebou také nese možnost využít programové rozhraní (API), díky kterému je možno používat mapové služby i ve vlastních aplikacích.

4.6.1 Základní funkce

Aplikace Google Maps zobrazuje svět v několika různých režimech:

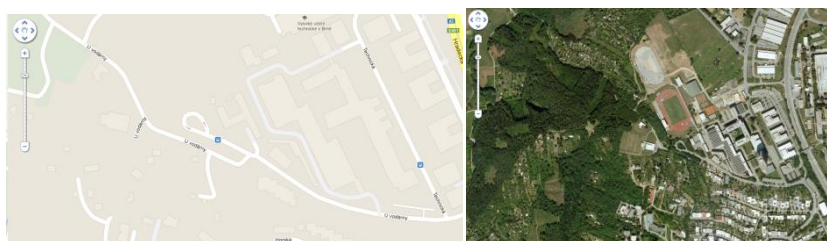
- mapovým podkladem,
- satelitním snímkem
- vizualizací terénu.

Po mapě se lze volně pohybovat a mapu zobrazovat s různou mírou zvětšení, přičemž limitem přiblížení je kvalita dat (mapových informací či satelitních snímků) v dané oblasti. Kvalita se v jednotlivých oblastech výrazně liší, hustě osídlené bývají pokryty lépe. V České republice je v obcích dobře zachycena většina ulic včetně obrysů budov.

Satelitní (přesněji letecké) snímky mají v českých městech rozlišení 20cm na jeden obrazový pixel, čili pohledem na mapu lze rozeznat auto na silnici. Snímky jsou aktualizovány průběžně, řádově jednou za rok, opět různorodě dle oblasti.

Vyhledávání adresy objekt na ulici přesně lokalizuje, syntaxe pro zadání adresy je dosti volná. Další funkcí je vyhledání trasy mezi dvěma objekty, autem, pěšky nebo veřejnou dopravou, a to spolu s odhadem času a vzdálenosti.

Mapa je zobrazena jako sada dlaždic, dopředu známých pro každou úroveň přiblížení.



Obrázek 21 Google maps - klasická mapa a letecký snímek

4.6.2 Možnosti využití

Jednoduché statické mapy

Tou nejjednodušší možností, jak využít API, je vložit do aplikace jednoduchou statickou mapu, tedy obrázek. K tomu slouží Static Maps API. Nabízí vcelku pokročilé možnosti. Do mapy je možno vkládat markery, či vlastní ikony, kreslit lomené čáry a oblasti, určit si typ obrázku. Využití nalezneme například v desktopové aplikaci, kde potřebujeme mapu pouze zobrazit. Tato služba je omezena na 1000 dotazů jednoho uživatele.

Google maps API v2

Jedná se o JavaScriptové API, které poskytuje možnost využívat všech možností klasických Google Maps, jako je například posouvání mapy v okně, přibližování, přepínání vrstev atd.

Před samotným použitím Google Maps API je třeba mít API klíč, pomocí kterého společnost Google monitoruje provoz a v případě velkého zatížení (10 000 přístupů/den) zpoplatňuje tuto službu. Tento klíč lze zdarma získat na code.google.com/apis/maps/signup.html.

- podmínkou je mít tzv. Google account. Klíč je vytvořen po odsouhlasení podmínek použití a zadání domény, na které poběží stránka s mapou.
- projekt s mapou musí být přístupný všem návštěvníkům (nesmí jít tedy o stránku, pro jejíž prohlížení je nutná registrace/přihlášení apod.)
- Google Maps Enterprise (placená verze) – pro projekty s velkým množstvím přístupů, či pro projekty určené omezenému počtu uživatelů

Google Maps API používá javascript. Před vložením mapy do projektu, je potřeba ji načíst (zde je nutné vložit API klíč a určit lokalizaci):

```
<script type="text/javascript" src="http://www.google.com/jsapi?key=API_KLIC&hl=cz"></script>
```

Při přípravě projektu pro vícejazyčné požití velmi pomáhá stránka: <https://spreadsheets.google.com/pub?key=p9pdwsai2hDMsLkXsoM05KQ&gid=1>, kde lze nalézt, kterými lokalizacemi API disponuje.

V průběhu tvorby projektu byla vydána nová stabilní verze API v3, která bude požitá pro webovou formu aplikace v blízké budoucnosti.

MVC model

Celé API je postaveno na MVC(Návrhový vzor - Model-View-Controller) modelu, díky kterému lze přistupovat k jednotlivým proměnným objektů. Veškeré vlastnosti objektů se nastavují pomocí tzv. setterů (např. `setCenter(latlng)`) a jejich hodnoty se získávají pomocí getterů (např. `getCenter()`). Různé stavy objektů jsou uloženy v jejich proměnných (přístup pomocí `get...`). Změny těchto stavů jsou řízeny pomocí událostního modelu.

Událostní model

- slouží k oznamování změn stavů objektů a k jejich řízení
- každá změna stavu vyvolá určitou událost, kterou je možné odchytit
- Odchycen se děje pomocí tzv. handlerů, které lze připojit k jednotlivým objektům.
- Existuje mnoho typů událostí, dají se rozdělit na uživatelem vyvolané (kliknutí myši), či vyvolané změnou stavu jiného objektu API (změna zoomu)

Možnosti Google Maps API:

- kontrola kompatibility prohlížeče - `google.maps.BrowserIsCompatible()`
- střed mapy - objekt `LatLng` pro určení, kde má vykreslená mapa střed. Tento objekt používá i mnoho jiných metod jako parametr. Zeměpisná šířka a zeměpisná délka jsou reálná čísla.

- Typ mapy
 - G_NORMAL_MAP – klasická mapa
 - G_SATELLITE_MAP – satelitní mapa
 - G_HYBRID_MAP – kombinace klasické a satelitní mapy
 - G_PHYSICAL_MAP – mapa terénu
- Informace - jednoduché okénko s informací.

Vrstvy

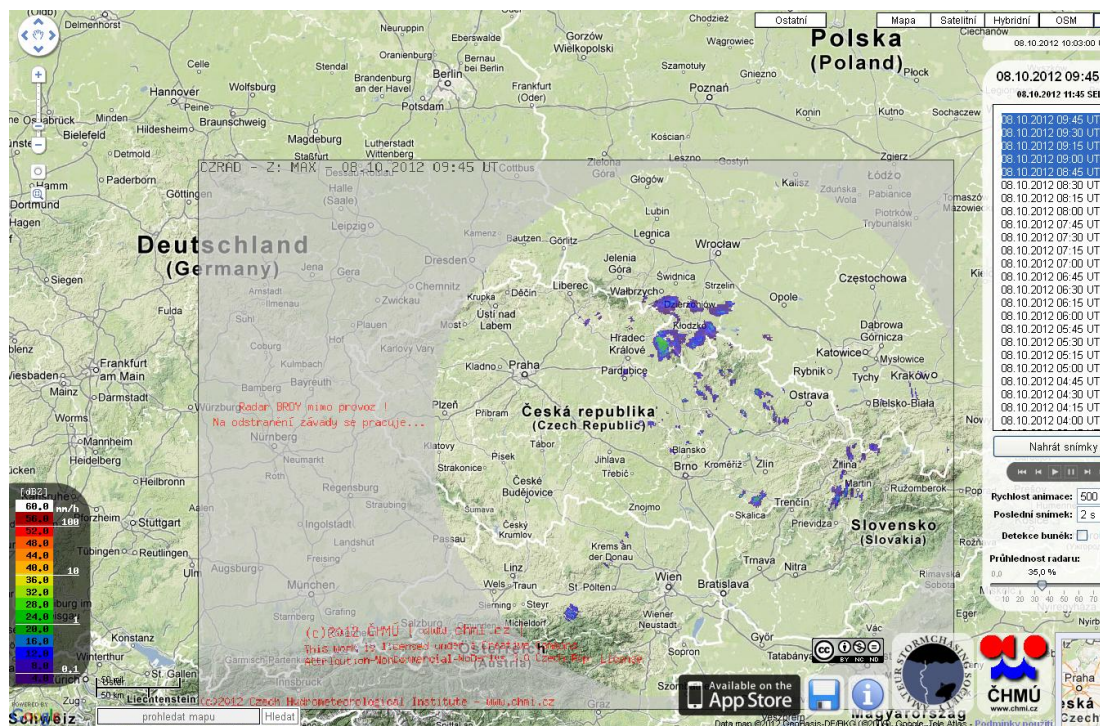
Do map lze vykreslovat mnoho prvků v různých vrstvách. Nicméně objekty, které nad mapu vykreslíme, poté nehlídá API, ale je třeba se o ně starat samostatně.



Obrázek 22 Vrstva - kontrolní prvky – měřítko, navigační posuvník

Vrstvami mohou být:

- Kontrolní prvky
- Markery – označení místa v mapě pomocí ikony (i vlastní ikony)
- Informační okna – pro zobrazení dalších informací, obsah okna je html stránkou
- Lomené čáry – pole objektů LatLng (nese informaci o jednom bodu a jeho souřadnicích), tato vrstva je v podstatě nejdůležitější součástí této části projektu a hlavní výhodou Google Maps oproti jiným mapovým řešením
- Polygonální oblasti – způsobem vytváření vrstvy se podobají lomeným čarám, jsou to uzavřené oblasti vyplněné barvou



Obrázek 23 Jeden z nejužitečnějších příkladů použití vlastní vrstvy v českém internetu – jedná se o aplikaci na zobrazení radarových dat ČHMÚ. Aplikace obsahuje ovládací prvky API, i prvky pro nastavení možností zobrazení dat. [20]

- Vlastní vrstvy – umožňují vykreslit různé obrázky nebo i jiné mapové podklady. Jednodušší variantou je obrázek přes celou mapu – na obrázku výše je to barevné měřítko množství srážek. Samotné zobrazení srážek na je ale provedeno pomocí vlastních mapových „dlaždic“

4.7 Zpracování XML souborů

Jak bude později rozebráno, je soubor *.GPX, který vytváří většina GPS záznamníků, vlastně XML souborem, proto se nyní budu věnovat možnostem jejich zpracování.

eXtensible Markup Language (rozšířitelný značkovací jazyk) je značkovací jazyk, který je podmnožinou historicky známého jazyku SGML (Standard Generalized Markup Language). Jazyk XML mimo jiné odstraňuje přílišnou složitost SGML.

XML je jazyk, který používá tagy, atributy atd., je založen na tvorbě vlastních tagů a je využíván hlavně pro přenos informací, které mohou být různými způsoby zobrazeny nebo dále zpracovány. Tyto jsou v případě XML uchovávány v hierarchické struktuře.

Všechny třídy, pro programování s XML v .NET jsou začleněny do jmenného prostoru System.Xml. V zásadě jsou při zpracovávání XML v .NET k dispozici dva odlišné způsoby zpracování. Prvním z nich je zpracování pomocí DOM. Pod touto zkratkou se ukrývá Document Object Model. Druhým poté jednosměrné čtení, které přináší oproti mechanismu DOM o mnoho menší paměťové nároky, ale nelze využít výhod hierarchie XML dokumentu.

• Jednosměrné čtení XML dokumentu

K jednosměrnému čtení od začátku XML dokumentu až po jeho konec jsou v .NET frameworku k dispozici třídy, XMLTextReader, XMLNodeReader a

XMLValidatingReader. XMLReader. Tato třída dokáže procházet XML dokumentem, který je k parsování (zpracování XML) dodán ve formě streamu.

- **Document Object Model**

Při druhém přístupu ke zpracování XML v .NET frameworku, využijeme hierarchické struktury XML. V případě použití tohoto přístupu vznikne v paměti počítače reprezentace struktury daného XML dokumentu, na rozdíl od přístupu dříve zmiňovaného, který byl založen na necachovaných streamech. To s sebou nese možnost, jak dokument číst nebo s ním manipulovat o mnoho pohodlnějším způsobem, než jedním směrem. Je nám tedy umožněno libovolně přistupovat k libovolné části XML dokumentu.

V případě použití DOMu je každá složka XML dokumentu představována instancí určité třídy. Základem těchto tříd je abstraktní třída **XMLNode**, která představuje jejich společného předka. Tak například instance třídy **XMLDocument** reprezentují vlastní XML dokument, **XMLElement** představuje element, **XMLAttribute** reprezentuje jeden atribut elementu, **XMLDocumentType** deklaraci dokumentu atd.

Třída XMLDocument obsahuje důležité metody, jejichž užitím je nám umožněno vytvořit požadovanou reprezentaci hierarchické struktury XML dokumentu v paměti a to jak např. ze souboru, tak z jednoduchého řetězce. Samozřejmě je také možné pomocí instancí této třídy vykonstruovanou hierarchii z paměti někde uložit (nejčastěji do souboru).

4.8 Návrh výstupů aplikace

Pro potřeby orientačního běžce a jeho osobního trenéra existuje mnoho důležitých faktorů, které ovlivňují jeho výkonnost. Tyto faktory se dají velmi často dobře zkoumat. Rozbor vytrvalostního sportu a OB speciálně již byl proveden výše, proto bych se nyní zaměřil na samotnou aplikaci. Vzhledem k tomu, že tento sport je ve své podstatě rozdělen na dvě disciplíny, byla takto koncipována i aplikace.

4.8.1 Výstupy běžecké části

Pro potřebu analýzy běžecké výkonnosti jsou hlavními výstupy grafy a tabulky. Ke každému tréninku či závodu:

- Tabulka času stráveného v jednotlivých tepových zónách
- Zobrazení délky tréninku včetně rozdělení na jednotlivá kola
- Zobrazení tréninku na mapě (v případě použití GPS)
- Graf průběhu tepové frekvence v závislosti na čase
- Graf průběhu rychlosti, nebo tempa na čase

Ke každému týdnu:

- Součty kilometrů v jednotlivých intenzitách
- Graf rozložení intenzit v jednotlivých dnech a týdnech

Ke každému cyklu

- Součty kilometrů v jednotlivých intenzitách
- Grafy rozložení běhů v intenzitách

- Součty a grafy pro porovnání cyklů navzájem

4.8.2 Část pro analýzu mapových faktorů

Analyzovat mapové schopnosti je náročnější, proto je výstup těchto činností pouze obecný

- Součty běhu s mapou (rychle, pomalu)
- Mapy tréninků a závodů, včetně zakreslených postupů
 - Pro tyto účely zavedena funkce nahrání dvou souborů typu obrázků, jelikož pro samostatný rozbor mapových tréninků a závodů existuje téměř dokonalá aplikace QuickRoute, kterou si v této práci pro její dokonalost a perfektní zpracování nedovolím nahradit.
- Zobrazení výstup GPS nad mapou

4.8.3 Další činnosti

Většina sportovců se nejen v přípravném období věnuje mnoha sportovním aktivitám, pro posouzení jejich vlivu (kladného i záporného) si je evidujeme. Zde je výčet těch nejdůležitějších:

- Běh na lyžích
- Jízda na kole
- Posilování
- Speciální běžecká síla
- Aerobní aktivity, hry
- Dlouhá chůze (nejlépe po horách)

Zobrazovat se budou pouze v součtech a grafech po celém cyklu a samozřejmě v celkové analýze.

5. Aplikace – tréninkový deník

5.1 Vzorce a výpočty použité v programu

Jak již bylo uvedeno, GPS zařízení ukládá svoji pozici do souboru ve sférických souřadnicích. Pro výpočet vzdálenosti dvou bodů, jejichž souřadnice známe, lze použít několik metod, které se značně liší svojí složitostí a tím i výslednou přesností. Podrobně uvedu pouze v programu užitý vzorec.

5.1.1 Haversinův vzorec

Pomocí tohoto vzorce počítáme vzdálenost dvou bodů jako vzdálenost dvou bodů po kulové ploše, neboli délku ortodomy mezi dvěma body. Tato metoda využívá tzv. referenční koule Země, což je koule, která nejlépe odpovídá tvaru Země. Výpočet za pomoci tohoto vzorce je pro naše využití plně dostačující a jeho přesnost se od náročnějších metod využívajících referenční elipsoid (Vincentův vzorec), či další Zemi podobnější tělesa liší řádově v 3 mm. Což je dáno tím, že největší počítaná vzdálenost je přibližně 50 m.

Postup pro výpočet je dle [1] následující:

$$R = 6372,795$$

$$\Delta lat = lat_2 - lat_1$$

$$\Delta long = long_2 - long_1$$

$$a = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos lat_1 \cdot \cos lat_2 \cdot \sin^2\left(\frac{\Delta long}{2}\right)$$

$$c = 2 \cdot \arctg\left(\frac{\sqrt{a}}{\sqrt{1-a}}\right)$$

$$d = R \cdot c$$

Kde R je poloměr zeměkoule v kilometrech, lat , $long$ jsou souřadnice bodů. Výsledná vzdálenost je potom označena písmenem d . Proměnná c se používá pro přehlednost vzorce.

Další použitelnou metodou je Vincentův vzorec, který na rozdíl od Haversinova nepočítá s kulovým modelem Země, ale s modelem elipsoidickým. Tím lze dosáhnout výpočtu vzdálenosti s přesností až půl milimetru na velkých vzdálenostech. S tím je ovšem spojena mnohem větší složitost tohoto výpočtu. Pakliže vezmeme v úvahu již uvedou nepřesnost GPS zařízení, bylo by jeho použití zbytečným plýtváním výpočetní kapacity.

5.1.2 Celková vzdálenost

Je jednoduchým součtem vzdáleností mezi všemi body. Případně mezi všemi body v daném kole.

$$s = \sum_{i=0}^n d_i$$

5.1.3 Výpočet rychlosti

Zajímavým údajem pro vyhodnocení závodu, či tréninku je rychlost. Vzhledem k rychlostem běhu je ale pro tento program mnohem zajímavější její převrácená hodnota uváděná v min/km. První počítanou rychlostí je tedy rychlost průměrná v a průměrné tempo p .

$$\bar{v} = \frac{s}{t_{celk}} = \frac{\sum_{i=0}^n d_i}{t_{celk}}, \quad \bar{v} = km/h$$

$$\bar{p} = \frac{1}{\bar{v}} \cdot 60 \text{ min}, \quad \bar{p} = \text{min}/km$$

Pro zobrazení v grafu je třeba zjišťovat i aktuální rychlost, tempo:

$$v = \frac{ds}{dt} = \frac{\Delta s}{\Delta t} = \frac{d_i}{t_i}$$

$$p = \frac{1}{v} \cdot 60 \text{ min}, \quad p = \text{min}/km$$

Průměrná tepová frekvence

$$\overline{TF} = \frac{\sum_{i=0}^n TF_i}{n}$$

5.2 Výstup GPS zařízení

GPS záznamová zařízení zatím nemají povinně standardní souborový výstup, takže některé firmy používají své vlastní souborové typy (např.: firma Garmin, ale jen u některých zařízení). Nicméně většina těchto typů je nějakým způsobem převeditelná do tzv. GPX souboru. Což je ve skutečnosti XML soubor a jako takový je velmi dobře počítačově zpracovatelný.

Při psaní aplikace bylo mým cílem zjednodušit uživateli práci a zkrátit co nejvíce dobu, po kterou do aplikace ukládá data. U některých přístrojů bylo jednoduché přistoupit k datům a zpracovat je, ale bohužel nestejná struktura souborů u jednotlivých výrobců nebo dokonce jiný formát souborů (Garmin Forerunner 110, které mají výstupní soubor ve formátu FIT) mi v tomto zabránili. Dokonce když jsem se pokusil získat specifikaci tohoto souboru, bylo mi oficiální cestou sděleno, že specifikace podléhá firemnímu tajemství a nelze ji volně šířit.

Proto vstup dat do aplikace probíhá pomocí otevření .GPX souboru (u všech zařízení lze tento soubor získat pomocí programů, které jsou k zařízení dodány), který se může nacházet kdekoliv v počítači.

Záznam se skládá z jednotlivých bodů. Každý bod je poté zaznamenán v jednom elementu xml souboru (popis dále).

5.2.1 Zpracování .gpx

Příklad struktury jednoho elementu souboru:

```
<?xml version="1.0" encoding="utf-8"?>
<gpx xmlns: možné atributy >
  <trk>
    <name>pojmenování - často čas</name>
    <trkseg>
      <trkpt lat="49.1741303" lon="16.5621675">
        <ele>336.4000000</ele>
        <time>2012-02-03T11:55:24Z</time>
        <extensions>
          <tp1:TrackPointExtension>
            <tp1:hr>112</tp1:hr>
          </tp1:TrackPointExtension>
        </extensions>
      </trkpt>
    </trkseg>
  </trk>
</gpx>
```

Popis jednotlivých uzlů:

- trkseg – odděluje jednotlivá kola
- trkpt – jeden bod tratě, v attributech lat a lon obsahuje souřadnice daného bodu
- ele – nadmořská výška daného bodu
- extensions – uzel který obsahuje rozšíření, z nichž asi nejzajímavější může být tepová frekvence. Tento uzel není povinný.

Pro tuto činnost byly navrženy třídy:

- otevreniGPX.cs
 - vstupem je řetězec cesty k souboru
 - výstupem oddělené souřadnice, čas, tep a údaje o výšce
 - vzhledem k tomu, že se jedná o xml soubor je použita metoda třídy XmlTextReader
- Cas.cs
 - výpočet času
 - výstupem jednotlivých metod jsou: čas mezi 2 body, celkový čas
- Vzdalenost.cs
 - vzdálenost mezi 2 body
 - celková vzdálenost
- Vyska.cs
 - celková nastoupaná (lze požit i pro jedno kolo), mezi 2 body

- Nastaveni.cs a Vyjimky.cs
 - slouží pro počáteční nastavení programu
 - ošetřuje jednotlivé vyjimky
- Tempo_R.cs
 - výpočet tempa
 - jednotlivé metody počítají: průměrné tempo mezi 2 body, celkové průměrné tempo
 - využívá vypočtených vzdáleností a časů

5.3 Zobrazení zaznamenaného souboru na mapě

Tento úkon je proveden pomocí Google maps API. Vzhledem k tomu, že nejjednodušší implementace úkonů nad těmito mapami je v javascriptu, byl tento programovací jazyk použit i pro tuto aplikaci.

Probíhá v několika fázích:

- převedení .gpx souboru do xml, který je podporován Google Maps API
- odeslání daného souboru pod jménem uživatele na server, kde je umístěna js aplikace
- zobrazení dané stránky ve webovém prohlížeči

5.3.1 Použité metody Google Maps API

- setCenter(LatLng); – nastaví střed zobrazené mapy na zadané koordináty (parametrem je objektLatLng)
- panTo(LatLng); – změni střed mapy na zadané koordináty (skokem nebo pohybem)
- getBounds(); – vrací hranice aktuálně zobrazené části mapy
- getZoom(); – metoda vrací aktuální hodnotu přiblížení (rozsah: 0 (minimum, tj. svět) až 19 (maximum))

Ovládací prvky

Stejně jako na maps.google.com, byly přidány ovládací prvky.

- GLargeMapControl – panel pro ovládání mapy (posun a úroveň přiblížení)
- GScaleControl – měřítko mapy
- GOverviewMapControl – minimapa v rohu, (nakonec nevyužita)
- GMapTypeControl – tlačítka umožňující přepínání mezi typy mapy
- Nastavení pozice ovládacího prvku (G_ANCHOR_TOP_RIGHT, ...)

Lomené čáry

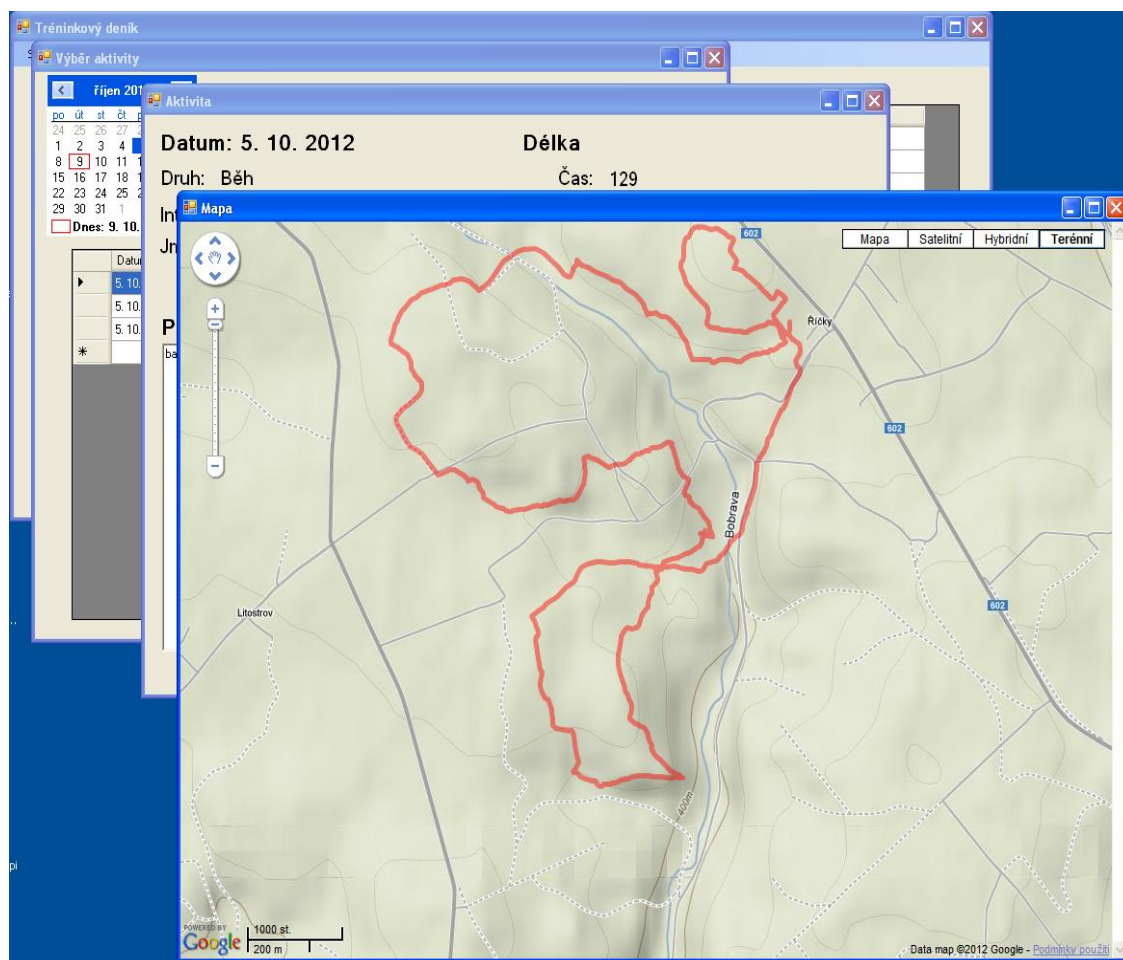
- GPolyline – lomená čára pro zobrazení tratě nad mapou, tato funkce má 3 vstupní hodnoty: pole objektů LatLng – všechny použité souřadnicové body, barva čáry, šířka čáry.

5.3.2 Popis programu

Obsluhu této části zajišťuje jedna třída `mapa.cs` a hlavně na externím serveru (pro aplikaci pro PC) běžící javascriptový program, jehož účelem je načtení a zobrazení mapy a tratě nad ní.

Třída `mapa.cs` – obsahuje formulář, který je vyplněn komponentou webový prohlížeč – jedná se o instanci Microsoft Internet Explorer[13].

Dále se její metody starají o již uvedené převedení souboru, přenos a následné zobrazení mapy v prohlížeči. Vzhledem k tomu, že mým cílem nebylo poskytnout uživateli další plnohodnotnou alternativu webových prohlížečů, je tento uzamčen pouze pro zobrazení stránky s mapou a tratí, kterou může prohlížet.



Obrázek 24 Zobrazení absolvované tratě nad mapou

Soubory `index.html` a `gpx.js` zajišťují:

- zobrazení mapy – zadání jedinečného klíče pro Google Maps[15]
- zobrazení trasy
- nastavení barvy, šířky
- zobrazení označení začátků jednotlivých kol
- pozici mapy přibližně uprostřed nad zobrazenou tratí (výpočet průměrné polohy bodů)

- přiblížení mapy tak, aby byla vidět celá trasa (nejvzdálenější body na trase od každé z os a jejich vzdálenost od okraje mapy)
- nastavení nejmenší vzdálenosti delta, dvou bodů, pomocí kterých vykresluji křivku (např.: pro pohyb na kole nemusím mít takovou přesnost jako pro orientační běh)

Při tvorbě .js souboru jsem se inspiroval zdrojem [14]. Výsledek můžete vidět na předcházejícím obrázku. Samozřejmostí je poté možnost zobrazení ortofota – leteckého snímku (v mapách používán nesprávně termín satelitní snímek) a hybridní mapy, což je část mapy nad leteckým snímkem.

5.4 Uživatelské rozhraní

Uživatelské rozhraní se skládá se z formulářů typu Windows.Form. Mojí snahou bylo navrhnout ho tak, aby bylo co nejjednodušší a pokud možno intuitivní na ovládání. Samozřejmostí poté je ošetření proti možným chybám ze strany uživatele, ale i hostitelského počítače.

Skládá se z těchto formulářů:

- Úvodní obrazovka
Obsah menu
 - Aktivita
 - upravit (nabídne dialog, kde lze vybírat z dnešních aktivit)
 - nová - dialog nová aktivita
 - Součty - formulář, kde lze zobrazit součty
 - Připojení - nastavení connectionstringu k databázi

Na formuláři jsou poté umístěny komponenty:

- kalendář pro výběr data, ve kterém chce uživatel editovat aktivity (po označení data přejde na dialog výběru z aktivit v daném dni)
 - zobrazený výpis všech aktivit za poslední rok, pro možnost rychlého procházení
- Výběr aktivity
 - zobrazení aktivit pro určené datum, možnosti editace, vytvoření nové, smazání
 - Aktivita
 - zobrazení aktivity včetně možnosti zobrazit mapy a uložené postupy
 - Mapa – internetový prohlížeč se připojí k serveru, na který program předtím odeslal záznam z GPS
 - Postupy – zobrazí přiložený obrázek
 - Nová aktivita
 - vytvoření nové aktivity, včetně zpracování dat z gps záznamníku
 - možnost prohlížet zaznamenaná data na mapě
 - Připojení - pozor, při nastavení špatného connectionstringu nebude program funkční
 - Součty - formulář, kde lze analyzovat měsíční, týdenní výkony a zkoumat objem tréninku v různých intenzitách, či při různých sportech

- Upravit aktivitu - stejný formulář jako pro novou jen při zvolení úpravy dané aktivity je tato načtena z databáze a zobrazena pro úpravy

5.5 Instalace aplikace

Aplikace je zabalena v samorozbalovacím archívu, k jejímu otevření stačí pouze spustit soubor `Treninkovy_denik.exe`.

Pro správnou funkci této aplikace je ji spouštět pod MS Windows XP a novější, je třeba mít nainstalovaný server MSSQL R2 2008 a v něm spuštěný přiložený soubor databáze `Treninkac.dbf`. Po spuštění je třeba zapsat správně řetězec pro připojení k databázi. Existují pro to dvě možnosti:

- ručně v konfiguračním souboru `konf.txt`, který se nachází ve složce programu.
- V případě, že toto uživatel neudělá, a program spustí, pokusí se aplikace připojit k defaultní databázi. Pakliže se jí to nepovede, bude uživatel dialogovým oknem vyzván k zadání správného řetězce k připojení. Poté po restartu aplikace již bude funkční

Pozor v případě že tento řetězec nebude zadán dobře, aplikace se spustí v omezeném režimu zpracování GPS souboru.

6. Závěr

Cílem této diplomové práce bylo vytvořit aplikaci „Tréninkový deník“, kde bude využito zpracování dat získaných za pomoci GPS záznamového přístroje.

V teoretické části jsem rozebral principy fungování GPS, chyby, které se při používání těchto přístrojů mohou vyskytnout, a použité přístroje. Dále jsem se věnoval potřebám orientačního běžce. Tato část byla důležitá pro analýzu výstupů aplikace.

Praktická část se věnuje samotnému programu. Podařilo se mi zpracovat data z několika různých GPS přístrojů. Tato samostatná kapitola mě donutila nastudovat samostatně programovací jazyk Javascript, který byl nutnou podmínkou pro možnost využití tzv.: GoogleMaps API. Tato část projektu byla velmi zajímavá, naučila mě mnoho věcí a věřím tomu, že mnohé budu schopen v budoucnosti využít.

Dále jsem se snažil o zapracování těchto dat do celkové struktury deníku sportovce.

Myslím si, že největší přínos této práce byl jinde než v pouhém psaní aplikace. Po rychlém promyšlení a pokusu rozvrhnout práci jsem zjistil, že se jedná o vcelku rozsáhlý projekt, na kterém by pravděpodobně v komerčním prostředí pracovalo více lidí než jeden. Proto jsem se po počátečním nezdařeném pokusu napsat vše bez rozmyslu, dostal k teoretickému návrhu projektu od prostudování dostupné literatury, přes studium potřeb uživatele, návrh databáze až ke konečnému psaní jednotlivých oddělených částí aplikace. Na závěr zbylo spojení těchto částí, kde se opět projevil nedostatek dílů ze začátku vývoje, jež bylo třeba odstranit.

V nejbližších asi 3 měsících plánuji převod aplikace do webové podoby a její spuštění na serveru společnosti, která má partnerský vztah k mému oddílu orientačního běhu.

Literatura

- [1] Konecký, J.: Měření vzdáleností a plochy pomocí GPS, VUT v Brně, diplomová práce, Brno, 2009.
- [2] Leick, A. GPS satellite surveying. 3rd ed.. Hoboken, Wiley and Sons, 435 s, 2004.
- [3] Xu, G. GPS: theory, algorithms and applications. Berlin; Heidelberg; New York, Springer, 315 s., 2003.
- [4] Neumann, G.: Trénink pod kontrolou: metody, kontrola a vyhodnocení vytrvalostního tréninku. Grada Publishing a.s., 2005.
- [5] Rapant, P.: Družicové polohové systémy. VŠB-TU Ostrava, 2002. 200 str. ISBN 80-248-0124-8
- [6] How good is GPS? Chris Rizos, SNAP-UNSW. 1999
- [7] NavtechGPS: GNSS Facts, <http://www.navtechgps.com/extra/GNSSfacts.asp>
- [8] Analýza četnosti výskytu družic GPS a GLONASS Trimble's Planning Software 17. duben 2008, <http://gis.templ.net/gps/Prague-20081704/>
- [9] Informace o evropském navigačním systému na stránkách ESA, <http://www.esa.int/esaNA/galileo.html>, 10. 9. 2012
- [10] http://www.aero.org/education/primers/gps/images/24-const_001.gif
- [11] <http://www.spacedepartment.cz/3-sekce/gnss-systemy/gnss-mimo-evropu/americky-navstar-gps/>, citace: 10. 9. 2012
- [12] Co je orientační běh, <http://www.orientacnibeh.cz/csob/cojeob.php>
- [13] Microsoft Visual C# Help
- [14] GPX Viewer – internetový článek, <http://notions.okuda.ca/geotagging/projects-im-working-on/gpx-viewer>, 31. července 2007, citace 14. srpna 2012
- [15] Náповěda a fóra Google Maps API, <https://developers.google.com/maps/>
- [16] Lenhart, Z.: Taktika a technika orientačního běhu, 69 s, 2009
- [17] Jameson, A.: Stavba sportovního tréninku, 250 s, 1988
- [18] Puš, P.: Poznáváme C# a Microsoft .NET – 54. díl – ADO. NET, 23. 12. 2005, <http://www.zive.cz/clanky/poznavame-c-a-microsoft-net--54-dil--ado-net/sc-3-a-128344/default.aspx>, citace: 10. 9. 2012
- [19] Google Maps API, 2. 9. 2008, <http://clanky.gug.cz/2008/04/google-maps-api.html>
- [20] ČHMÚ, 3. 9. 2012, <http://radar.bourky.cz>,
- [21] GARMIN International Inc., webové stránky výrobce GPS zařízení: <http://www.garmin.cz/produkty/sport/forerunner-110.html>.
- [22] Puš, P.: Poznáváme C# a Microsoft .NET – 61. díl – ADO .NET – DataTable, 10. 2. 2006, <http://www.zive.cz/clanky/poznavame-c-a-microsoft-net-61-dil--ado-net--datatable/sc-3-a-129087/default.aspx>, citace: 10. 9. 2012
- [23] Puš, P.: 64. díl – ADO .NET – Datový adaptér, 3. 3. 2006, <http://www.zive.cz/clanky/poznavame-c-a-microsoft-net-64-dil--ado-net--datovy-adapter/sc-3-a-129467/default.aspx>, citace: 10. 9. 2012
- [24] Stránky Evropské komise – Galileo program, http://ec.europa.eu/enterprise/policies/satnav/galileo/programme/index_en.htm, 8. 9. 2012