

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

WEBOVÉ SLUŽBY PRO PODPORU GEOLOKACE V ROZSÁHLÝCH SÍTÍCH

DIPLOMOVÁ PRÁCE

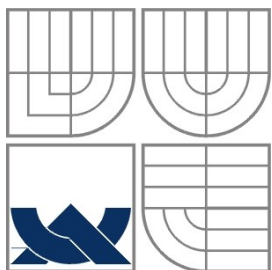
MASTER'S THESIS

AUTOR PRÁCE

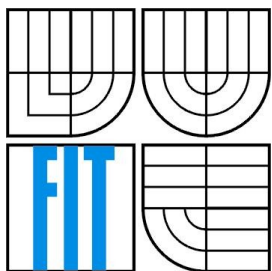
AUTHOR

BC. MICHAL IMLAUF

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

WEBOVÉ SLUŽBY PRO PODPORU GEOLOKACE V ROZSÁHLÝCH SÍTÍCH

WEB-SERVICES FOR GEO-LOCATION SUPPORT IN WIDE-AREA NETWORKS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

BC. MICHAL IMLAUF

VEDOUCÍ PRÁCE

SUPERVISOR

RNDr. MAREK RYCHLÝ, PH.D.

BRNO 2012

Abstrakt

Cílem této diplomové práce je popsat techniky geolokace v rozsáhlých bezdrátových sítích a způsoby vyjádření lokace jednotlivých uzlů. Praktická část popisuje implementaci webových služeb, jenž simulují pohyb uzlů v těchto bezdrátových sítích a implementaci webové služby, jenž bude agregovat souřadné systémy těchto bezdrátových sítí.

Abstract

The goal of this master's thesis is to describe geolocation techniques in wireless wide-area networks and the ways of expressing the individual node location. Practical part describes implementation of web services that simulate the node movement in these wireless networks and implementation of web service that is going to aggregate the coordinate systems of these wireless networks.

Klíčová slova

Geolokace, bezdrátová síť, rozsáhlé sítě, UTM, webová služba, WCF, MEF, REST, souřadné systémy

Keywords

Geolocation, wireless network, wide-area networks, UTM, web service, WCF, MEF, REST, coordinate systems

Citace

Michal Imlauf : Webové služby pro podporu geolokace v rozsáhlých sítích, diplomová práce, Brno, FIT VUT v Brně, 2012

Webové služby pro podporu geolokace v rozsáhlých sítích

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením RNDr. Marka Rychlého, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Michal Imlauf
22.května 2012

Poděkování

Tímto bych chtěl poděkovat vedoucímu této práce RNDr. Marku Rychlému Ph.D. za poskytnutou pomoc při řešení této diplomové práce.

©Michal Imlauf, 2012

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

Obsah

Obsah.....	1
1 Úvod.....	3
2 Geolokace v bezdrátových sítích.....	4
2.1 Rozsáhlé bezdrátové sítě.....	4
2.1.1 GPS.....	4
2.1.2 GSM.....	5
2.2 Senzorové bezdrátové sítě.....	6
2.2.1 Infračervené senzory.....	6
2.2.2 ZigBee	7
2.2.3 Bluetooth.....	8
2.2.4 Ultrazvukové senzory.....	8
3 Způsoby vyjádření polohy uzlů.....	9
3.1 Souřadné systémy.....	9
3.2 Vyjádření polohy uzlů na povrchu země.....	9
3.3 Vyjádření polohy pomocí systému šířka-délka.....	10
3.4 Univerzální transversální Mercatorův systém souřadnic.....	10
3.5 Souřadnicový systém.....	11
4 Webové služby.....	13
4.1 Webové služby SOAP.....	13
4.1.1 SOAP.....	14
4.1.2 WSDL.....	16
4.1.3 UDDI.....	16
4.2 Webové služby REST.....	17
4.3 REST versus SOAP.....	18
4.4 Formáty uložení dat ve webových službách.....	18
4.4.1 XML.....	18
4.4.2 JSON.....	19
5 Návrh a implementace webové služby simulující pohyb uzlů v bezdrátových sítích.....	20
5.1 Návrh rozhraní.....	20
5.2 Návrh vnitřní struktury.....	21
5.3 Alternativní senzorová síť.....	21
5.4 Implementace	22
6 Webová služby pro skládání souřadných systémů a převod na globální souřadnice.....	24
6.1 Data přístupná pomocí rozhraní.....	25

6.2 Návrh rozhraní.....	25
6.3 Návrh vnitřní funkcionality webové služby.....	26
6.3.1 Převod mezi dvěma dvourozměrnými souřadnými systémy.....	27
6.3.2 Převod mezi dvěma trojrozměrnými souřadnými systémy.....	27
6.3.3 Převod mezi globálními souřadnými systémy.....	27
6.4 Modulárnost.....	28
7 Implementace.....	29
7.1 Rozhraní webové služby.....	29
7.1.1 WCF.....	29
7.1.2 Implementace rozhraní.....	30
7.2 Konfigurace webové služby.....	32
7.3 Modulárnost webové služby.....	33
7.3.1 Rozhraní modulů.....	33
7.4 Modul pro získávání informací ze senzorových sítí.....	35
7.5 Modul pro agregaci senzorových sítí.....	36
7.5.1 Třída pro kombinaci dvou dvojrozměrných souřadnicových systémů.....	36
7.5.2 Třída pro kombinaci dvou trojrozměrných souřadnicových systémů.....	38
7.5.3 Třída pro doplnění globální pozice jednotlivých uzlů.....	39
8 Demonstrace funkcionality.....	40
8.1 Přehled senzorových sítí.....	40
8.1.1 Primární síť.....	40
8.1.2 Sekundární síť.....	41
8.2 Konfigurace služby a modulů.....	44
8.3 Agregace senzorových sítí.....	45
8.4 Práce s globálními souřadnicemi.....	48
9 Závěr.....	51
Literatura.....	52

1 Úvod

V současné době je pojem geolokace v bezdrátových sítích poměrně často diskutovaným problémem. Existuje snaha vytvořit co možná nejlevnější a nepřesnější způsoby geolokace, které by vyhovovaly všem možným prostředím a zároveň byly dostupné pro každého. Většinou se také nespokojíme pouze s jednou senzorovou sítí, ale používáme jich několik. Tato diplomová práce rozebírá různé bezdrátové sítě umožňující geolokaci.

Tato práce dále pojednává o způsobech lokalizace uzlů jak v zavedených bezdrátových sítích používaných pro jiné účely, jako je například síť GSM, tak i dedikované senzorové sítě instalované pro možnost geolokace v nich. Poté popíšeme jednotlivé možnosti vyjádření polohy uzlů od nejnižší úrovně až po globální vyjádření polohy. Závěrečným bodem teoretické části bude popsání funkcionality webových služeb včetně protokolů potřebných k jejich fungování.

V praktické části se zaměříme na návrh a implementaci webových služeb simulující pohyb v bezdrátových sítích, návrh a implementaci webové služby, jenž bude mít k dispozici informace o těchto sítích a dokáže agregovat souřadné systémy do jednoho výsledného nezávisle na typu použitého systému.

V další kapitole poté ukážeme příklady užití webové služby a spojení jednotlivých souřadných systémů do jednoho výsledného systému.

V poslední kapitole shrneme dosažené výsledky v této diplomové práci a navrhneme možná rozšíření.

2 Geolokace v bezdrátových sítích

Tato kapitola bude věnována pojmu geolokace, bezdrátovým sítím a možnostem geolokace v těchto sítích. Geolokace je pojmenování pro určení geografické lokace bodu na základě daných informací. Většinou můžeme tento pojem slyšet ve spojitosti s určováním polohy síťových zařízení na internetu. Existují různé možnosti geolokace, které zmíním dále. V této diplomové práci se budu věnovat geolokaci na základě dat získaných z agentů v bezdrátových sítích, které vysílají mimo jiné i informace, se kterými lze určit polohu zařízení v síti. S touto informací můžeme vyjádřit jeho geografickou polohu pomocí různých způsobů, například pomocí souřadnicového systému WGS-84.

2.1 Rozsáhlé bezdrátové sítě

V předchozím textu jsme se věnovali geolokaci bez specifitějšího zaměření. V následujících podkapitolách si přiblížíme bezdrátové sítě, ve kterých pro geolokaci využíváme již existující infrastrukturu tak, že je potřeba ke zprovoznění geolokace pouze přijímač schopný vypočítat lokaci, případně musí být schopný předat informace potřebné k vypočítání aktuální polohy dalšímu prvku, který již tuto polohu vypočítá.

Bezdrátová síť je taková síť, která ke komunikaci mezi dvěma stanicemi používá bezdrátový přenos. Nejčastěji se jedná o některý z druhů elektromagnetického záření, například rádiové vlny a paprsky světla používané v přenosu za pomoci laserového paprsku.

2.1.1 GPS

Informace v této kapitole jsou volně převzaty z [1] a [2].

Ke zjištění polohy pomocí GPS je potřeba signál alespoň 3 satelitů ze 30, jenž obíhají zemi ve výšce 20 km. Nejsou tedy geostacionární. K zajištění kvality signálu je ve většině případů potřebné prostředí bez překážek, které by způsobovaly oslabení či zastínění signálu. GPS je tedy spíše vhodné k určování polohy mimo budovy, kde je zaručena kvalita signálu.

Je možné s jistými úpravami upravit systém GPS i pro vnitřní použití. Například pomocí predikce pozice satelitů. Takovéto systémy mají přesnost v jednotkách desítek metrů, je tedy diskutabilní, zda je tento systém vůbec použitelný.

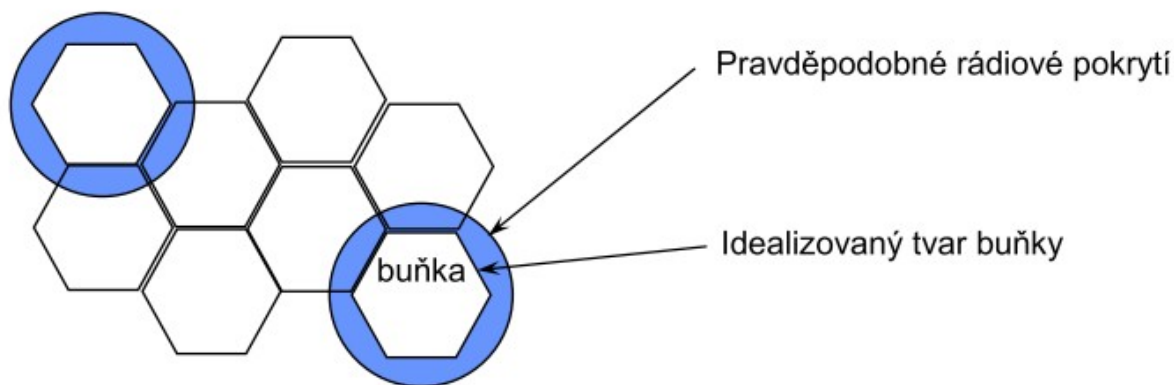
Zajímavější jsou úpravy, ve kterých jsou použity vysílače nazývané pseudolity, tedy pseudo-satelity, které generují signál podobný GPS signálu. V roce 1999 byl tento systém úspěšně navrhnout a otestován v národní univerzitě v Soulu, kde byl tento systém schopný dosáhnout přesnosti v řádu centimetrů.

Vypočtení polohy pouze ze signálu satelitů je poměrně náročné na výpočetní výkon, proto byly vyvinuty další metody pro zjednodušení výpočtu, a tudíž i zkrácení doby, ve které je možné zjistit první zaměření polohy přístroje, například Asisted-GPS, tedy systém využívající ke zrychlení výpočtu informace pro danou oblast, která je přístroji dostupná předem. Tyto informace obsahují údaje o satelitech, například jaké jsou použité frekvence a předpokládaná poloha na orbitu.

I tak je ale využití pro vnitřní prostory velmi omezené. Výpočet polohy je v porovnání s ostatními systémy pro lokalizaci uvnitř budov zdoluhavý a náročný na energii. Řešením těchto problémů by mohlo být uvedení do provozu evropského systému Galileo, který by měl poskytovat lepší penetraci signálu v budovách a přesnější zaměření polohy. Prozatím je systém určování lokace pomocí satelitů použitelný pouze v otevřených prostranstvích.

2.1.2 GSM

Informace v této kapitole jsou čerpány z [3]. Další, již zaběhnutý systém, který je možné využít pro geolokaci, je GSM. Jedná se o buňkovou síť poskytující celosvětové hlasové a datové služby. Klasické vysílače rádií a televizí se vyznačují tak vysokým vysílacím výkonem, aby pokryly co největší území. Na rozdíl od nich používají buňkové sítě jako je GSM většinou vysílače s mnohem menším výkonem, zato jsou rozmístěny mnohem blíže u sebe než klasické vysílače. Velikost buněk začíná od 100 m, a dosahuje až 35 km. Zde platí, že čím méně uživatelů na daném území bydlí, tím větší je velikost buňky.



Obrázek 1. Systém GSM buněk, převzato z [3]

Geolokace je v této síti možná díky tomu, že se rádiová pokrytí jednotlivých buněk překrývají. Jak je patrné z obrázku výše, je možné získat signál z několika buněk. Dle modelu Okamura-hata lze pomocí síly signálu spočítat vzdálenosti od vysílačů, pokud známe jejich vysílací výkon, umístění vysílače a frekvenci, na které vysílají. Tyto informace jsou volně dostupné, takže není problém tyto vzdálenosti skutečně vypočítat.

Jakmile známe vzdálenosti od vysílačů v okolí, dají se sestrojit kružnice, které mají střed ve vysílačích, poloměr je daný vzdáleností od přijímače. Pokud v této chvíli máme alespoň 3 kružnice, které se protínají, je možné z polohy těchto průsečíků sestrojit trojúhelník. Po protnutí dvou těžnic tohoto trojúhelníku získáme jeho střed, který by se měl nacházet v místě měření síly signálů.

2.2 Senzorové bezdrátové sítě

Ne vždy je možné využít stávající infrastrukturu ke geolokaci. V případech, kdy potřebujeme přesnější lokalizaci, než kterou jsme schopni vypočítat z GSM nebo GPS, jsou výhodnější senzorové sítě. Zároveň mají senzorové sítě tu výhodu, že jednotlivé uzly mají obvykle mnohem menší spotřebu energie. Dokáží tedy fungovat déle bez údržby. Tento předpoklad ovšem funguje pouze v případě, že nedojde k poruše na jednotlivých zařízeních.

2.2.1 Infračervené senzory

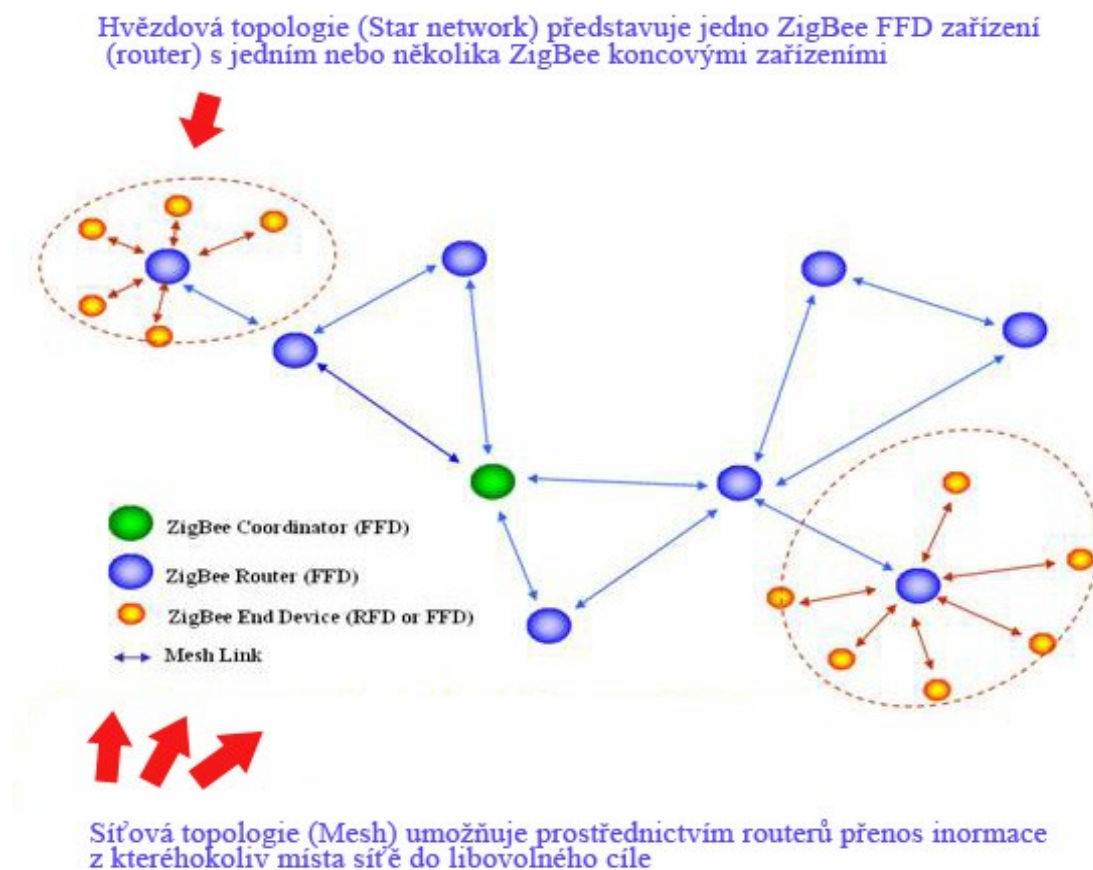
Informace z této kapitoly jsou volně převzaté z [4]. Infračervené záření je běžně využívané pro dálkové ovládání elektroniky a ovládacích prvků v domácnostech. Senzory pro geolokaci využívající infračervené spektrum používají přenosy modulovaného infračerveného světla na krátké vzdálenosti umožňující zaslání identifikační informace od mobilního vysílače ke statickému přijímači, jehož pozice je vysílači známa.

Pro lokaci mobilního vysílače infračerveného záření, je potřeba do každé lokace, ve které si přejeme detekovat přítomnost tohoto vysílače, potřeba umístit přijímač infračerveného záření. Zařízení periodicky vysílá svoje identifikační údaje a pozice mobilního zařízení je tedy známa. Tento princip lze využít i obráceně umístěním statických vysílačů, které budou vysílat svoji identifikaci. Poté může mobilní jednotka přijímat tyto identifikace a na jejich základě určit, na jakém místě se nachází. Přesnost v tomto systému je omezena pouze na určování polohy pomocí známé pozice pevných bodů. Dokážeme tedy určit, že mobilní zařízení se nachází v dané místnosti, ale nedokážeme již určit jeho přesnou polohu za předpokladu, že v místnosti je pouze jeden statický přijímač IR.

Hlavním představitelem tohoto způsobu geolokace je systém active badge, který se skládá ze sítě infračervených statických přijímačů a z mobilních vysílačů. I když jsou jednotlivé složky vysílače a přijímače poměrně levné, je cena instalace celého systému navýšena kvůli údržbě. Baterie vysílačů vydrží v průměru 1 rok a je tedy nutná jejich údržba. Další nevýhodou je to, že tento systém nelze využívat na přímém slunci, nebo v místnostech s vysokou teplotou.

2.2.2 ZigBee

ZigBee je radiový standard na frekvencích 2,4 GHz, 915 MHz a 868 MHz, který se zaměřuje na komunikaci s co největší úsporou energie. Tento způsob komunikace je tedy ideální pro zařízení, u nichž požadujeme dlouhou dobu provozu bez nutnosti časté údržby, případně přímého napájení.



Obrázek 2. Topologie ZigBee, převzato z [5]

Lokalizace v ZigBee je vždy založena na určení vzdáleností nebo úhlů pohyblivého uzlu vzhledem k pevným uzlům.

Vzdálenost lze určit z:

1. Úrovně přijatého signálu (RSSI) - tento způsob je poměrně nepřesný a vyžaduje složitou kalibraci.
2. Časového zpoždění signálu - při použití radiové vlny je minimální měřitelná vzdálenost 150 m. Tuto metodu je nutné použít v kombinaci s ultrazvukovými senzory, kde je minimální měřitelná vzdálenost v řádu centimetrů.
3. Přepínáním kanálů - měření časového zpoždění signálu je převedeno na měření změny fáze. Mezi uzly sítě musí být minimálně vzdálenost 3,75 m, maximálně pak 60 m.

Je možné použít kombinaci výše uvedených metod pro určování vzdáleností jednotlivých uzlů.

Pro určování úhlů se použije interferometr, kde je nutné použít dva přijímače na mobilním zařízení. Měření časového zpoždění signálu je zde realizováno měřením rozdílu mezi fázemi signálů na výstupu přijímačů. Informace lokalizaci v bezdrátové síti ZigBee jsem převzal z [6]

2.2.3 Bluetooth

Informace v této kapitole jsem čerpal z [7] a [8]. Bluetooth je radiový standard, který popisuje komunikaci na frekvenci 2,4 GHz pro vzdálenosti do sta metrů. Je rozdělený na 3 třídy podle maximálního povoleného vysílacího výkonu. Vzhledem k masovému rozšíření bluetooth do mobilních zařízení, jako jsou mobilní telefony, notebooky a tablety, je možné tato zařízení používat jako mobilní jednotky, u nichž budeme určovat pozici pomocí infrastruktury.

Pro samotnou geolokaci můžeme využít dvou metod. Buďto budeme jako v předchozích případech využívat geolokaci na základě rozpoznání vysílání známých statických bodů, nebo můžeme použít triangulaci, která bude poskytovat mnohem vyšší přesnost.

Pro triangulaci potřebujeme mít okolo sebe minimálně 3 body a na základě komunikace s nimi získáme hodnotu RSSI (sílu signálu každého jednotlivého bodu). Z této síly signálu jsme schopni vypočítat vzdálenost od těchto bodů a podobně jako u GMS sítě vypočítáme aktuální pozici. Jelikož bluetooth v mobilních zařízeních většinou poskytuje pouze dosah do 3 metrů, není vždy možné určit pozici zařízení na základě informací od statických bodů v geolokační síti. V této chvíli lze díky povaze standardu získat informace od okolních mobilních jednotek a obdržet tak přibližnou pozici na základě informace poskytnuté druhou mobilní jednotkou.

2.2.4 Ultrazvukové senzory

Ultrazvukové senzory jsou používány zejména pro jejich vynikající přesnost v určování polohy zařízení. Ze všech způsobů zmíněných v této práci je metoda geolokace pomocí ultrazvukových senzorů nejpřesnější.

Ultrazvukové čidlo vysílá periodicky vysokofrekvenční impuls, který se šíří prostorem rychlostí zvuku. Pokud tento impuls narazí na nějaký předmět, odrazí se od něj a vrací se zpět k čidlu. Z časového intervalu mezi vysláním impulsu a návratem ozvěny odvodí čidlo vzdálenost k předmětu.

Dva nejznámější systémy využití ultrazvukových senzorů pro geolokaci uvnitř budov se nazývají MIT cricket a Active bat. Oba tyto systémy ale vyžadují infrastrukturu vysílačů a přijímačů k tomu, aby mohly efektivně vypočítat pozici mobilního zařízení, stejně jako v případě infračervených senzorů. Informace v této kapitole jsem čerpal z [9].

3 Způsoby vyjádření polohy uzlů

Vyjádřit polohu jednotlivých uzlů lze několika způsoby. V této práci je pro nás ale relevantní vyjádření polohy uzlů na povrchu naší planety, případně vyjádření polohy v prostoru bez dalšího upřesnění jejich polohy vzhledem k zemskému povrchu.

3.1 Souřadné systémy

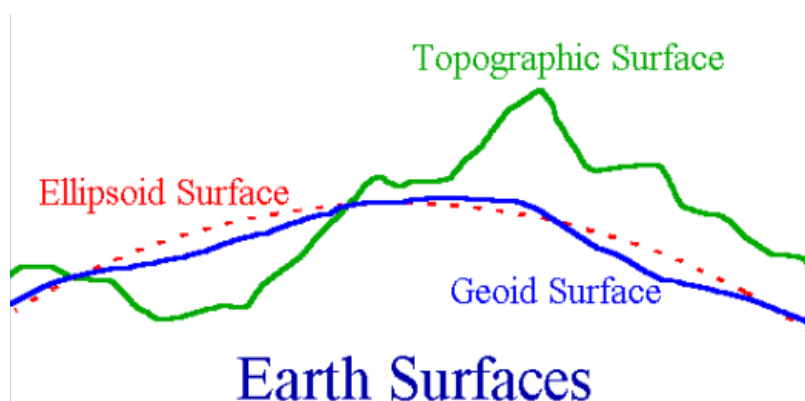
Před tím, než začneme určovat polohu bodu, je potřeba definovat souřadný systém, který by tuto polohu jednoznačně definoval.

Souřadný systém musí splňovat tři požadavky [10]:

1. Definice polohy je jednoznačná - objekty se stejnou polohou jsou identické a objekty s různou polohou nejsou identické.
2. Definice polohy je kvantifikovatelná - jsou zde definovány jednotky, které budou porovnatelné a tudíž změřitelné.
3. Definice metriky pro měření vzdáleností takové, aby bylo možné vypočítat vzdálenosti mezi jednotlivými body.

3.2 Vyjádření polohy uzlů na povrchu země

Informace v této kapitole jsou čerpané z [10]. Povrch naší planety je poměrně zvrásněný a členitý, z tohoto důvodu je nemožné popsat tvar pomocí koule nebo jednoduchého elipsoidu. Takovýto způsob by byl poměrně nepřesný. Bylo ale potřeba popsat tvar naší planety pomocí tělesa. Takto vznikl počáteční aproximací povrchu země původní model zvaný geoid.



Obrázek 3. Geoid, převzato z [10]

V obrázku uvedeném výše můžeme vidět zelenou barvou znázorněný skutečný povrch země, modrou povrch geoidu a červenou povrch elipsoidu. I tak ale geoid není optimální pro zobrazení povrchu země, jelikož ho nelze jednoduše matematicky popsat, jak je vidět z předchozího obrázku. Z tohoto důvodu byl zaveden další model geoidu a tím je náhradní elipsoid. Elipsoid může být buďto globální jako například WGS-84, nebo lokální jako například Besselův, který je určený pro mapování území Evropy, především České Republiky.

3.3 Vyjádření polohy pomocí systému šířka-délka

Tento systém zahrnuje dva koordináty pro určení pozice na povrchu země (zeměpisnou šířku a délku). Pojem zeměpisná šířka definuje úhel, který svírá rovník s přímkou, která prochází středem země a zároveň daným bodem. Počátkem, tedy nulovou hodnotou, by byl bod, který leží na rovníku. Tento systém rozděluje planetu na jižní a severní polokouli.

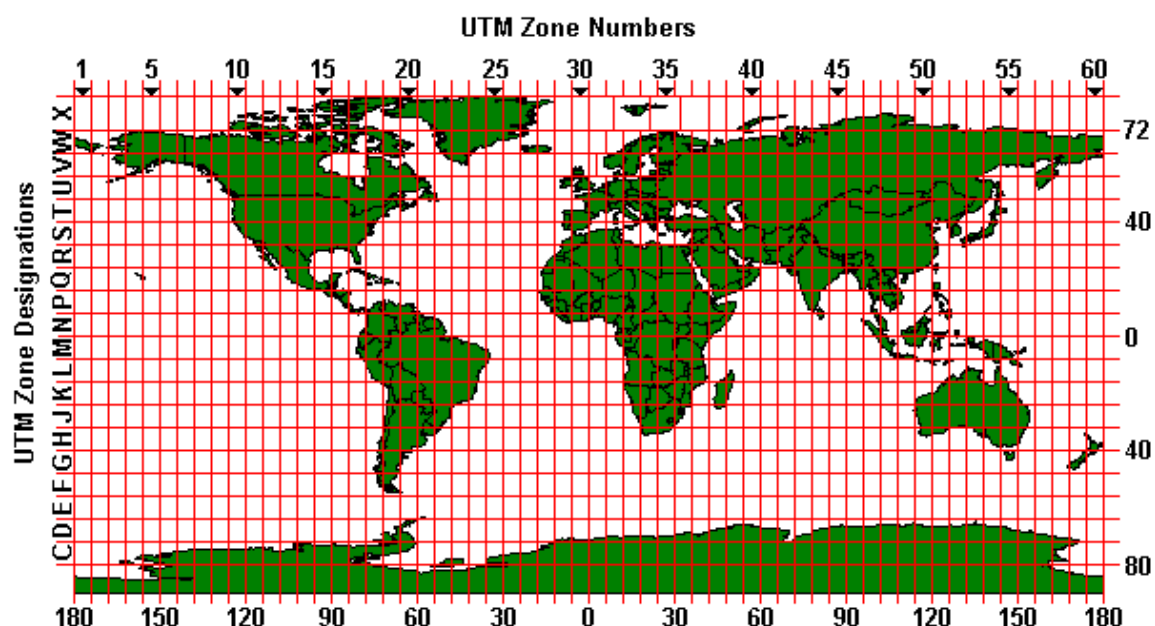
Zeměpisná délka je určena stejným způsobem jako šířka. V případě zeměpisné délky je výchozí přímka nazývaná nultý poledník. Prochází hvězdárnou Greenwich nedaleko Londýna. Tento poledník rozděluje planetu na západní a východní polokouli [10].

Vyjádření polohy nabývá hodnot severní/jižní šířky: Stupně [0-89], minuty [0-59], sekundy [0-59] a východní/západní délky: Stupně [0-179], minuty [0-59], sekundy [0-59] délky. Poloha daného bodu by mohla vypadat například následovně 49° 12' 50" severní šířky 16° 37' 35" východní délky.

3.4 Univerzální transverzální Mercatorův systém souřadnic

UTM je kartografické zobrazení pro souřadnicový systém WGS 84. Jedná se o způsob určování polohy, ve kterém zemský povrch rozdělíme na síť zón zobrazených pomocí Mercatova zobrazení. Jelikož se jedná o zobrazení elipsoidu WGS 84 do roviny, lze v těchto zónách měřit vzdálenost mezi dvěma body pomocí Pythagorovy věty. Toto platí pouze za předpokladu, že oba tyto body leží v jedné zóně.

WGS 84 je vojenský souřadnicový systém používaný státy NATO. Referenční plochou je elipsoid WGS 84 (World Geodetic System). Systém má počátek v hmotném středu Země (s přesností cca 2 m) – jedná se o geocentrický systém. Osa Z je totožná s osou rotace Země v roce 1984. Osy X a Y leží v rovině rovníku. Počátek a orientace jeho os X, Y, Z jsou realizovány pomocí 12 pozemských stanic se známými přesnými souřadnicemi, které nepřetržitě monitorují dráhy družic systému GPS-NAVSTAR. [11]



Obrázek 4. Zóny UTM, převzato z [12]

Zobrazení UTM je příčné konformní válcové Mercatorovo zobrazení poledníkových pásů. Každý takový pás má vlastní souřadnicovou soustavu, osa N je vložena do obrazu osového poledníku, osa E se vkládá do obrazu rovníku. Souřadnicový systém UTM se využívá k určení souřadnic mezi 80° jižní šířky a 84° severní šířky. Pro oblasti mimo tyto hodnoty se využívá projekce (UPS).

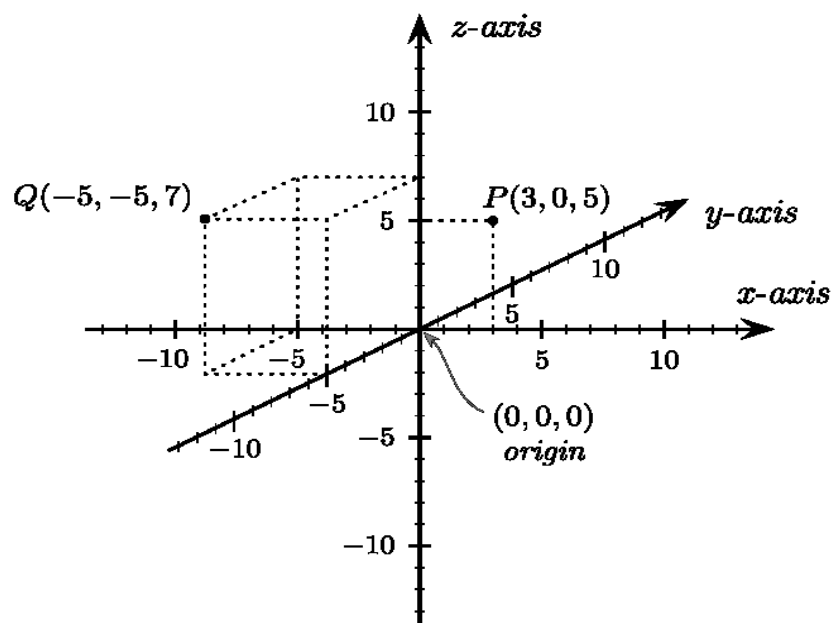
3.5 Souřadnicový systém

V případě, že bychom chtěli vyjádřit polohu jednotlivých bodů přesnějším způsobem, bude nutné využít jiné metody zobrazení bodů. Jednou z možností je souřadnicový systém.

V této práci budeme využívat souřadnicové systémy popisující dvoudimenzionální a trojdimenzionální prostory. V souřadnicovém systému pro trojdimenzionální prostory můžeme použít kartézskou soustavu souřadnic, kde budeme vyjadřovat polohu daných bodů vzhledem k referenčnímu bodu.

Kartézská soustava souřadnic se skládá ze tří os, které jsou na sebe navzájem kolmé. Bod je zde vyjádřen jako trojice (x, y, z) , kde jednotlivé znaky značí polohu na ose x , y , z . Počátek soustavy je umístěn v bodě $(0, 0, 0)$.

Pomocí této soustavy dosáhneme při vyjadřování polohy jednotlivých bodů mnohem větší přesnosti než v případě souřadnicového systému UTM.



Obrázek 5. Kartézská soustava souřadnic, převzato z [13]

4 Webové služby

Tato kapitola a podkapitoly 4.1, 4.2 a 4.4.1 jsou převzaty z mé bakalářské práce.[14]

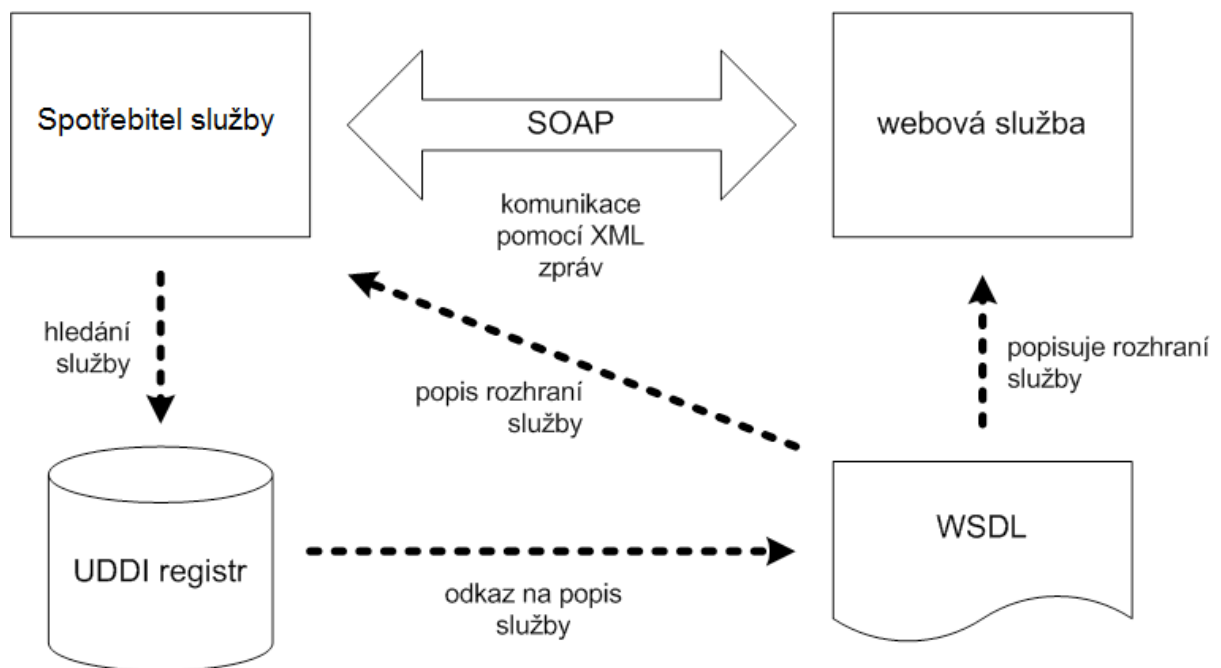
Webová služba funguje na principu spotřebitel-poskytovatel. Hlavní výhodou webových služeb je fakt, že konzument i poskytovatel služby mohou běžet na různých platformách a jazycích, ale i tak bude komunikace mezi nimi možná.

Webová služba je softwarový systém navržený pro podporu interakce dvou počítačů. Její rozhraní je popsáno ve strojově zpracovatelném formátu (konkrétně WSDL). Ostatní systémy pracují s webovou službou způsobem popsaným v její definici, tedy pomocí zpráv SOAP, typicky přenášených pomocí protokolu HTTP v XML formátu ve spojení s ostatními webovými standardy [15].

To, že spolu konzument a poskytovatel mohou bezproblémově komunikovat, je způsobené faktem, že zprávy se předávají pomocí známých a často používaných protokolů, tedy SMTP a HTTP. Jelikož jsou to protokoly používané pro e-mail a pro přístup k webovým stránkám, nebývají jejich porty zablokovány firewallem a tudíž mohou bez problému fungovat.

4.1 Webové služby SOAP

I když se jedná o klient-server síťovou aplikaci je v některých případech pro komunikaci mezi klientem a serverem ještě potřeba třetí strana. Tou je registr webových služeb neboli UDDI. Tímto registrem se budu podrobněji zabývat v kapitole 4.1.3. Celý proces funguje tak, že poskytovatel služby uloží do registru WSDL popis své webové služby. Spotřebitel si následně z registrů může tento popis stáhnout a na základě tohoto popisu vytvořit aplikaci, která bude s danou webovou službou komunikovat pomocí SOAP zpráv zapsaných ve formátu XML. Celý postup je názorně zobrazen na následujícím obrázku.



Obrázek 6. Popis architektury webových služeb SOAP , převzato z [16].

4.1.1 SOAP

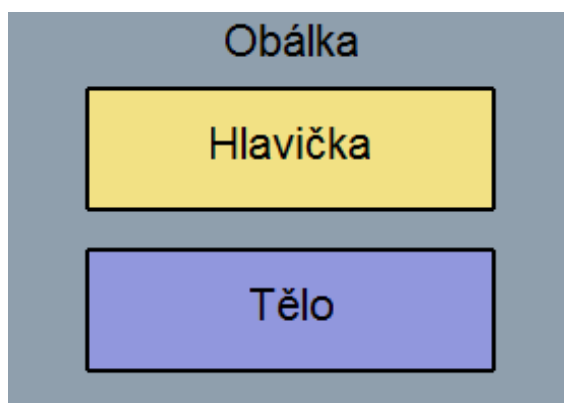
V této kapitole budeme vycházet z [17], kde se nachází přehled tohoto protokolu. SOAP je zkratka pro simple object access protocol. SOAP je komunikační protokol, který popisuje syntaxi zpráv používaných ve webových službách založených na tomto protokolu.

Co je SOAP:

- SOAP znamená jednoduchý protokol objektového přístupu
- SOAP je komunikační protokol
- SOAP byl vytvořen pro komunikaci mezi aplikacemi
- SOAP je formát pro odesílání zpráv
- SOAP je navržen pro komunikaci po internetu
- SOAP je nezávislý na platformě
- SOAP je nezávislý na jazyku
- SOAP je založen na XML
- SOAP je jednoduchý a rozšiřitelný
- SOAP bude vyvíjen jako standard W3C

SOAP umožňuje komunikaci i v prostředích s různými operačními systémy a jazyky, jelikož používá pro zápis XML a pro přenos protokol HTTP, jenž je široce rozšířený. Základ funguje na konceptu jednosměrné zprávy, v praxi se ale využívá pro složitější modely komunikace.

Struktura SOAP zprávy



Obrázek 7. Struktura SOAP zprávy, převzato z [18].

- **Obálka (Envelope)** - Kořenový element XML dokumentu, který ji definuje jako SOAP zprávu. Do obálky je možno přidat jak deklaraci namespace, tak i ostatní atributy.
- **Hlavička (Header)** - Pokud je ve zprávě přítomna, musí být umístěna ihned za obálkou. Většinou jsou v ní umístěny informace o tom, jak by měla být SOAP zpráva zpracována. Tato část zprávy není povinná.
- **Tělo (Body)** - Pokud zpráva neobsahuje hlavičku, tělo následuje okamžitě za obálkou. Obsahuje zprávu, kterou chceme předat.

4.1.2 WSDL

Informace v této kapitole jsou volně převzaty z [19]. WSDL čili Web Service Description Language je možné volně přeložit jako jazyk popisující webové služby. Dokument WSDL je textový soubor zapsaný ve formátu XML obsahující umístění a popis funkcí dané webové služby, například vstupy a výstupy či názvy proměnných.

WSDL používá 4 hlavní elementy XML k popisu webové služby. Tyto elementy jsou uzavřeny do lomených závorek < > a jsou vypsány v následujícím odstavci.

- **types** - Definice datových typů, které jsou použity v dané webové službě.
- **messages** - Popisuje data, která jsou vyměňována mezi službou a uživatelem. Mohou se skládat z více částí.
- **binding** - Specifikuje, jak budou operace definované v "<portType>" přenášeny. Je možné specifikovat více možností přenosu pro jeden druh operace.
- **portType, Interfaces** - Je výčet operací prováděných službou. Definuje webovou službu, operace, které je možné provádět, a zprávy zasílané v rámci těchto operací.

Ve webových službách SOAP jsou podporovány 4 druhy zpráv :

1. Jednosměrné - klient zašle zprávu webové službě
2. Požadavek-odpověď - zaslání zprávy webové službě a čekání na odpověď
3. Vyžádání odpovědi - webová služba zašle žádost a čeká na odpověď od klienta
4. Upozornění - webová služba zašle zprávu

4.1.3 UDDI

Co je to UDDI (Universal Description, Discovery and Integration)

- UDDI znamená univerzální popis, objevení a integraci
- UDDI je místo pro uložení informací o webových službách
- UDDI je místo, kde jsou popsána rozhraní jednotlivých webových služeb
- UDDI komunikuje pomocí protokolu SOAP
- UDDI je postavena na platformě Microsoft .NET

UDDI je tedy místo, které slouží firmám i jednotlivcům pro registraci jejich webových služeb, případně pro vyhledávání webových služeb vhodných pro jejich záměry. Před UDDI zde nebyl žádný sjednocený systém pro tento typ služeb. V současné době se tohoto projektu účastí i giganti softwarové a hardwarové výroby. Jedni z nich jsou Dell, Fujitsu, HP, Hitachi, IBM, Intel, Microsoft, Oracle, SAP a Sun [20].

4.2 Webové služby REST

Informace v této kapitole jsem čerpal z [21]. REST (Representational State Transfer) je další způsob implementace webových služeb. Na rozdíl od SOAP je REST orientován datově nikoliv procedurálně.

V praxi to znamená, že REST přistupuje přímo ke zdrojům bez nutnosti volat danou funkci, která by datovou reprezentaci tohoto zdroje vrátila jako návratovou hodnotu. Zdrojem v tomto kontextu mohou být jak data, tak i stav aplikace, jenž lze popsat konkrétními daty. Všechny zdroje mají vlastní identifikátory URI. URI jsme v protokolu SOAP používali k identifikaci funkcí, zde je používáme k identifikaci jednotlivých zdrojů.

Jednou z klíčových charakteristik RESTu je to, že využívá metody HTTP pro stejné účely, pro které byly určeny. V SOAP se většina funkcí volá pomocí metody POST nebo v některých případech GET. Nezáleží, jakou operaci budeme zrovna provádět. REST ale následuje původní význam metod HTTP a využívá metody následovně:

- K získání zdroje ze serveru - metoda GET
- K vytvoření zdroje na serveru - metoda POST
- Změna, nebo úprava stavu nad zdrojem - metoda PUT
- Smazání zdroje - metoda DELETE

Znamená to tedy, že každý zdroj má svoji vlastní URI. S pomocí výše uvedených metod ji můžeme získat, vytvořit, upravit nebo smazat. Zkratkou pro tento jednotný přístup je CRUD (Create, Read, Update, Delete).

Jelikož přístup k datům RESTové webové služby nevyžaduje složitou implementaci na straně klienta, je hojně využívána v mobilních technologiích. Implementace webové služby se sice může zdát složitější, tento fakt je ovšem vyvážen jednoduchostí implementace pro konzumenta služby, jenž nevyžaduje žádné speciální rozhraní jako v případě webových služeb SOAP.

4.3 REST versus SOAP

Jak už bylo zmíněno v předchozí kapitole, hlavní rozdíl mezi těmito přístupy v implementaci webových služeb je to, že v SOAP voláme funkce implementované na webovém serveru, kdežto v REST přistupujeme rovnou ke zdrojům uložených na serveru. Je to způsob komunikace pomocí HTTP tak, jak byl původně zamýšlen.

Výhody REST:

- Výsledky jsou čitelné
- Snadněji implementovatelné, tudíž nejsou potřeba sady nástrojů jako u SOAP
- Nezávislost na platformě a jazyku
- Umožňuje dočasné ukládání dat
- Bezstavová komunikace

Výhody SOAP:

- Zavedený systém - jsou dostupné sady nástrojů pro implementaci složitějších služeb
- Existence WSDL - snadnější implementace klientů konzumujících webovou službu, odpadá problém vyšší složitosti SOAP.

4.4 Formáty uložení dat ve webových službách

Webové služby typu SOAP používají pro uložení dat pouze jeden formát, a to formát XML. Webové služby založené na principu REST mají v tomto ohledu větší variabilitu a uložení dat umožňují jak ve formátu XML, tak i ve formátu JSON.

4.4.1 XML

V této kapitole se budeme věnovat XML a budeme čerpat ze zdroje [22]. XML je zkratka pro Extensible markup language (rozšiřitelný značkovací jazyk). Patří mezi jazyky popisné, což znamená, že jeho syntaxe obsahuje popis toho, co znamenají jednotlivé data v dokumentu. Tento jazyk je primárně určený pro uchování a výměnu dat. Hlavní výhodou jazyka XML je jeho otevřenost a snadná zpracovatelnost, kdokoli má přístup k jeho specifikaci a je možné ho bez problémů strojově zpracovat nebo otevřít v textovém editoru a jednotlivé prvky upravit.

Dokument v XML je vždy textový s použitím znaků unicode. Další důležitý znak dokumentu je to, že má stromovou strukturu právě s jedním kořenem. Všechny prvky v dokumentu XML jsou definovány pomocí XML tagů, které jsou párové a case - sensitive. Párový v tomto případě znamená,

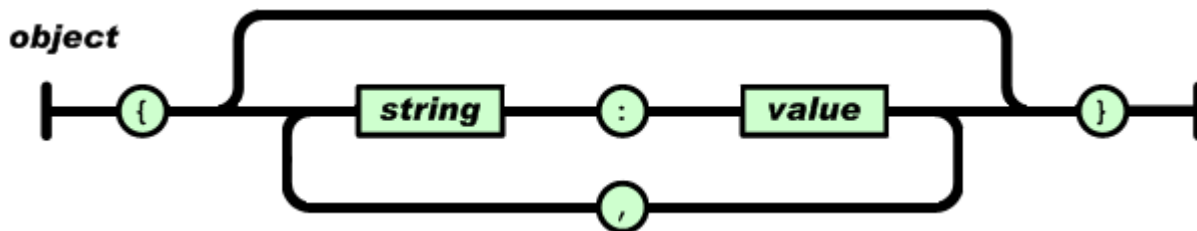
že je zde tag, který prvek uvozuje, a tag, který ho ukončuje. Tagem začíná a končí každý jednotlivý prvek dokumentu. Následně si ukážeme strukturu XML dokumentu [22].

```
<kořen>
  <potomek>
    <<Zde je místo pro informace>>
  </potomek>
</kořen>
```

4.4.2 JSON

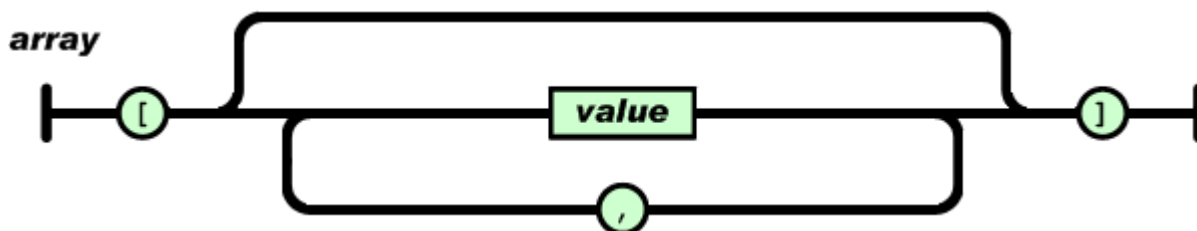
Informace o datovém formátu JSON jsem čerpal z [23]. JSON je jednoduchý formát uložení dat určených k transportu. Je založen na podmnožině programovacího jazyka JavaScript. Je to textový formát kompletně nezávislý na programovacích jazycích, ale používá pravidla, jenž jsou typická pro jazyky typu C a jazyky jemu podobné. Syntaxe JSONu je založena na dvou strukturách.

- **Kolekce párů název - hodnota.** V programovacích jazycích se takovému páru říká objekt, záznam, struktura atd. Tato kolekce začíná znakem "{", následuje pár název - hodnota, kde jsou tyto dvě informace od sebe odděleny znakem ":", kolekce může obsahovat více těchto párů, každý takový pár je oddělen čárkou. Kolekce je ukončena znakem "}". Pro znázornění je k dispozici diagram tvorby kolekce.



Obrázek 8. Diagram tvorby kolekce, převzato z [23]

- **Řazený seznam hodnot** - V programovacích jazycích se takovýto seznam nazývá list, seznam, pole či vektor. Tento seznam začíná znakem "[", následuje sled hodnot, které jsou odděleny čárkou, tento seznam je ukončen znakem "]". Pro znázornění je k dispozici diagram tvorby seznamu.



Obrázek 9. Diagram tvorby seznamu, převzato z [23]

5 Návrh a implementace webové služby simulující pohyb uzlů v bezdrátových sítích

Jelikož nemáme k dispozici senzorovou síť a k ní náležející mobilní jednotku vysílající informace potřebné ke geolokaci, je zapotřebí vytvořit softwarové řešení, které bude simulovat jednotlivé senzorové sítě, umožňovat přístup k informacím o poloze jednotlivých uzlů, upravovat tyto informace, případně přidávat a odebírat uzly z dané senzorové sítě. Tyto senzorové sítě musí být konfigurovatelné, aby bylo umožněno případné testování výsledné aplikace.

5.1 Návrh rozhraní

Rozhraní těchto webových služeb musí umožňovat kompletní přístup k jednotlivým uzlům, tak i k jejich celkové kolekci. Tímto přístupem se rozumí jak čtení dat o jejich poloze, tak i úprava těchto informací. Z těchto důvodů se mi jeví jako ideální navrhnout tuto webovou službu podle standardů REST, jelikož byl určený právě pro takovouto manipulaci s daty. Rozhraní této webové služby tedy bude tvořeno jednotlivými URI označující jednotlivé uzly nebo URI označující celou kolekci uzlů. Na základě toho, jakou HTTP metodu použijeme, bude služba reagovat různými odpověďmi přesně dle principů architektury REST.

Při použití metody GET na URI uzlu vrátí webová služba informace o daném uzlu ve formátu XML. Jelikož všechny potřebné informace pro identifikaci daného uzlu předáme již v URI adrese, nebude potřeba předávat webové službě žádné další informace. Pokud použijeme metodu GET na adresu webové služby, bude služba odpovídat vrácením celé kolekce uzlů. Tato kolekce bude opět ve formátu XML.

Další metodou v protokolu HTTP je metoda POST. Při použití na URI webové služby tato metoda umožní vytvořit nový uzel. Vstupní formát dat bude v XML, tato data budou popisovat všechny informace potřebné k vytvoření nového uzlu, jeho identifikační číslo, polohu v souřadném systému, identifikaci sítě, do které patří, a v určitých případech i umístění v jiné síti.

V takovýchto případech bude potřebná identifikace této sítě. Tato metoda bude vracet pouze identifikační číslo vytvořeného uzlu. Uzel bude dále umožňovat přidání do datové části uzlu i globální polohu spolu s typem těchto dat. V počátku budeme počítat s polohou v systému UTM a v systému šířka/délka.

Pro editaci existujících uzlů se v architektuře webových služeb REST používá HTTP metoda PUT. Pomocí této metody budeme upravovat stávající uzel, formát dat bude stejný jako výstupní formát při metodě POST, dále bude možné upravovat celý uzel a všechna data popsaná v odstavci výše. Odpovědí na tuto metodu bude pouze kladný či záporný výsledek operace.

Poslední metodou v tomto návrhu bude metoda DELETE. Tato metoda je definována pro mazání zdrojů. I zde bude mít stejný význam. Bude sloužit pro mazání jednotlivých uzlů na základě zadané URI.

5.2 Návrh vnitřní struktury

Po návrhu rozhraní přichází na řadu návrh vnitřní struktury aplikace, která bude dané data spravovat. Jelikož bude nutné přistupovat jak k informacím o jednotlivých uzlech, tak i celé kolekci, je příhodné využít seznam, který bude umožňovat vyhledání daného uzlu za pomoci jeho identifikačního čísla. Dále by tento seznam měl mít možnost přidávat, odstraňovat a modifikovat jednotlivé uzly. Jednotlivé přístupové body rozhraní budou sloužit právě pro operace s daným seznamem uzlů. Kromě tohoto seznamu je zapotřebí možnost konfigurace senzorové sítě představované touto službou. Jelikož se jedná o webovou službu, nejlepším způsobem bude konfigurační soubor pro každou instanci této webové služby.

5.3 Alternativní senzorová síť

Jelikož je důležité otestovat webovou službu, jenž je primárním cílem této diplomové práce, při různých nastaveních, je potřeba vytvořit senzorovou síť, která bude poskytovat odlišnou funkcionalitu. Tato alternativní senzorová síť bude umožňovat vytvoření uzlů, jejichž poloha nebude určena pouze v jedné síti, ale i jejich polohou v jiné síti.

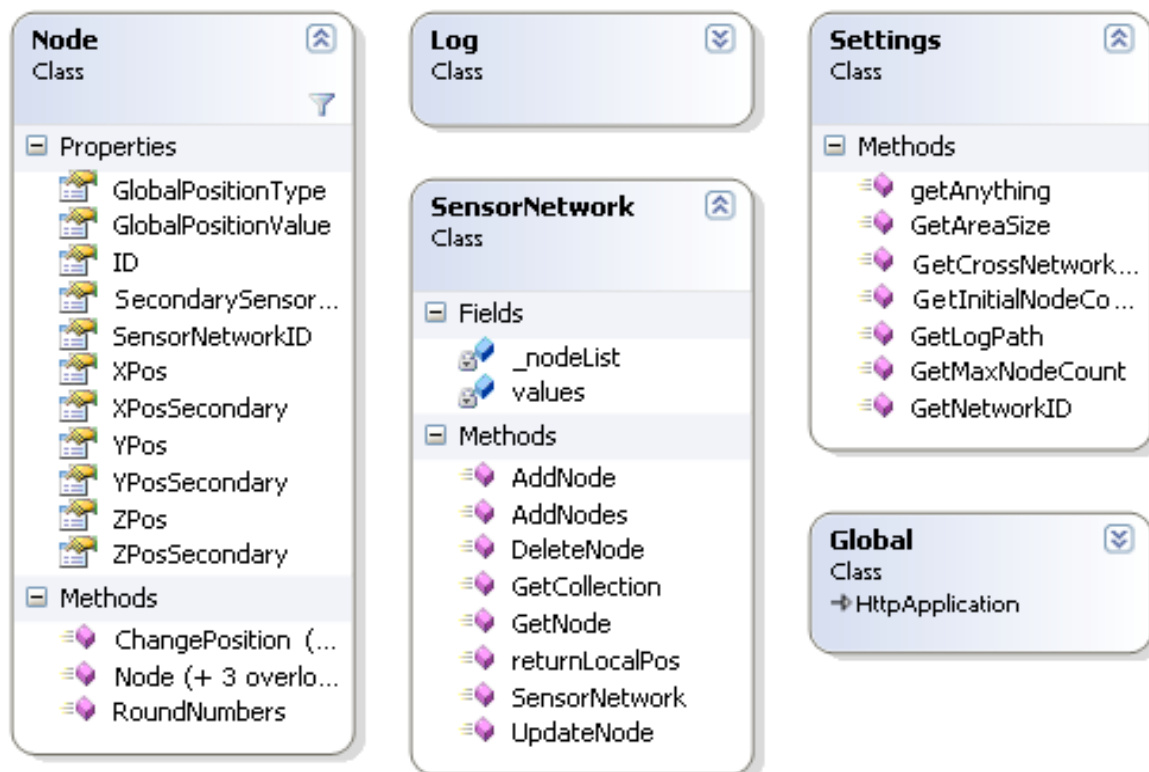
Polohy v jednotlivých sítích je možné generovat náhodně v případě, že budeme vyjadřovat polohu pouze v jedné síti. Pokud ale budeme chtít vyjádřit polohu uzlů ve dvou sítích, musíme polohu v druhé síti vypočítat. Pokud bychom použili náhodná čísla, nebylo by možné určit parametry natočení a posunu těchto sítí.

Pro výpočet umístění bude nejprve potřeba nastavit parametry. Následně použijeme matici pro natočení bodu ve 2D prostoru. Matici budeme popisovat později. Po rotaci bodu na základě zadaného úhlu přičteme k oběma bodům posuv v jejich osách.

5.4 Implementace

Implementaci této webové služby jsem se rozhodl realizovat ve vývojovém prostředí Visual Studio 2010 s využitím jazyka C# a .Net Framework verze 4. Důležitou součástí této implementace je i WCF (Windows Communication Foundation). Je to framework určený pro tvorbu aplikací zaměřených na služby, například pro webové služby typu REST či SOAP.

V následující části kapitoly si přiblížíme třídy implementované v této webové službě. Tento diagram tříd byl vytvořen ve Visual Studio 2010.



Obrázek 10. Diagram tříd webové služby, zdroj vlastní

- **SensorNetwork**

Tato třída je jádrem webové služby. Obsahuje list objektů "Node", které jsou přístupné přes REST rozhraní, které je implementováno za pomoci WCF. Konstruktor třídy obsahuje inicializaci, jenž vytvoří zadaný počet objektů "Node", které vloží do listu zmíněného výše. Objekty "Node" se vytváří podle nastavení obsaženého v souboru web.config, který obsahuje nastavení potřebná pro chod služby a zároveň i parametry simulované senzorové sítě.

Tato třída využívá třídy, Node, Log, Settings.

- Node

Tato třída představuje uzel senzorové sítě. Jednotlivé proměnné, které jsou obsažené v této třídě, musejí být typu public, jelikož k nim budeme přistupovat i z jiných tříd. Zároveň musejí obsahovat atribut DataMember, aby je bylo možno serializovat a deserializovat. Toto chování je potřebné, jelikož tyto objekty budou zasilány jako odpověď na HTTP dotazy GET na webovou službu a zároveň přijímány jako argument při úpravě či vytvoření nového uzlu s metodami PUT či POST.

Tato třída využívá třídy Log, Settings.

- Global

Třída zapouzdřující celou implementaci této webové služby. Inicializuje třídu sensor network a spustí ji tak, aby mohla přijímat HTTP požadavky.

- Settings

Tato třída má na starosti načítání informací ze souboru web.config. S její pomocí dokážeme konfigurovat jednotlivé parametry senzorové sítě.

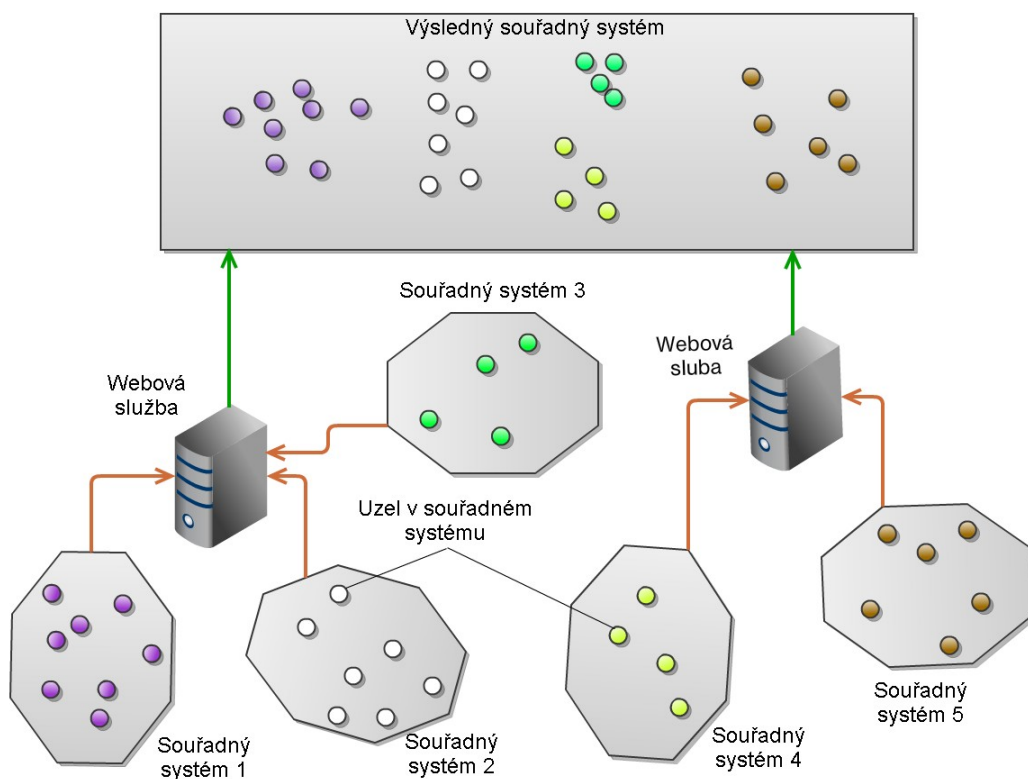
- Log

Jelikož webová služba nemá žádnou konzoli, kam by se dala vypisovat chybová hlášení, je nutné řešit tento problém jinak. Třída Log umožňuje vytvářet záznamy o nestandardním chování webové služby. Cesta k souboru log je uložena v souboru web.config.

6 Webová služby pro skládání souřadných systémů a převod na globální souřadnice

Hlavním cílem této implementace je agregace několika souřadných systémů do jednoho výsledného s možností přiřadit každému jednotlivému uzlu ve výsledném systému globální pozici. Agregace je zde míněna jako postupný převod souřadnic v jednotlivých souřadných systémech na vyšší souřadný systém. Výsledný systém tedy nebude obsahovat pouze jednu webovou službu, ale stromovou hierarchii webových služeb umožňující skládání lokálních souřadnicových systémů tak, abychom na vrcholu stromu měli všechny body těchto lokálních souřadnicových systémů zároveň umístěny do jednoho společného systému, a zároveň tyto uzly obsahovaly i informaci o globálním vyjádření jejich polohy.

Je tedy potřeba implementovat postup, který umožní vypočítat vztahy mezi jednotlivými souřadnými systémy, aby bylo možné doplnit ke každému uzlu libovolný souřadný systém a zároveň i jeho pozici v globálních souřadnicích.



Obrázek 11. Hierarchie webových služeb, zdroj vlastní

6.1 Data přístupná pomocí rozhraní

Než začneme s návrhem rozhraní, musíme specifikovat, k jakým datům budeme přistupovat. Nejmenší datovou jednotkou zde bude jednotlivý uzel senzorové sítě, jenž bude obsahovat všechny informace jak o své poloze, tak i o způsobu vyjádření polohy. Například u globální polohy je potřeba zapsat, jaký systém je použit pro zápis, aby bylo možné s těmito daty pracovat. Dále by tento uzel měl obsahovat informace o pozici v souřadném systému, do kterého patří, a zároveň by v této datové struktuře mělo být místo pro souřadnice jiného souřadného systému, jelikož budeme chtít ponechat originální souřadnice, ale zároveň i přidávat umístění daného uzlu v jiném souřadném systému. Jelikož budeme ukládat umístění uzlu ve více souřadných systémech, bude potřeba rozlišit, ve kterých bychom proto měli ukládat kromě identifikace uzlu i identifikaci systému, v jejichž lokalitě se uzel nachází.

Všechny uzly budou uloženy v seznamu, který bude umožňovat vyhledávání jednotlivých uzlů podle zadaných kritérií. Tento požadavek je důležitý, jelikož potřebujeme mít kompletní a jednoduchý přístup k vybraným uzlům

6.2 Návrh rozhraní

V této kapitole se budeme věnovat návrhu rozhraní webové služby. Rozhraní je vlastně způsob, jakým bude služba obsluhovat příchozí požadavky na úpravu či přístup k datům. Obsluha požadavků bude probíhat na portu 80 (port asociovaným s protokolem HTTP). Veškerá komunikace se službou bude tedy probíhat za pomoci tohoto protokolu. Podobu jednotlivých požadavků a odpovědí rozebereme v následující kapitole, ovšem datová část jednotlivých požadavků bude ve tvaru XML či JSON. Služba bude navržena dle standardu REST, bude bezstavová a zároveň zde budeme definovat jednotný přístup pro získání a manipulaci s daty v podobě operací CRUD, tzn. create, read, update, delete (volně přeloženo jako vytvoření, čtení, úprava a smazání dat). V následujících odstavcích se budeme věnovat právě návrhu jednotlivých operací.

Jak již bylo zmíněno výše, hlavním prvkem rozhraní bude implementace operací create, read, update, delete. Všechny tyto operace budou prováděny jak na jednotlivých uzlech, tak v případě čtení a smazání i na celých seznamech. Operace create bude přístupná pomocí metody HTTP POST, kde předáme webové službě uzel zapsaný v syntaxi XML. Odpovědí na tuto metodu bude pouze odpověď v podobě identifikace vytvořeného uzlu. Podobným způsobem budeme přistupovat k operaci update, zde bude použita HTTP metoda PUT určená k editaci zdroje. V tomto případě bude opět předán v XML syntaxi celý uzel, a pokud v seznamu uzlů nalezneme uzel se stejnou identifikací, upravíme jeho hodnoty v závislosti na dodaných datech. V případě, že daný uzel nenalezneme, budeme muset

v implementaci rozhodnout, zda data o uzlu zahodit, případně vytvořit uzel na základě dodaných informací. Chování operace delete je zřejmé, po zadání příslušné identifikace smažeme odpovídající uzel.

Předchozí operace mají poměrně jednoduchou úlohu, není tedy důvod je v návrhu jakkoliv komplikovat. V případě operace read toto ovšem neplatí. Je nutné zajistit, aby data, která budou k dispozici, byla aktualizovaná v rozumném intervalu. Je zde prostor pro 2 řešení.

V prvním případě by mezi webovou službou a senzorovými sítěmi fungovala aplikace, která by se starala o pravidelné aktualizace polohy jednotlivých uzlů, případně jejich přidávání a odebrání. Tento pohled by byl efektivní z pohledu jednoduchosti výsledné služby. Služba by se v tomto případě starala pouze o samotný převod mezi souřadnými systémy a obsluhovala příchozí požadavky na data. Na druhou stranu by se mohlo stát, že by aplikace obstarávající aktualizaci dat havarovala, případně by se dostala do stavu, ve kterém by nemohla dále aktualizovat data. V tom případě by bylo možné, že by chvíli trvalo, než by si fakt, že data nejsou aktualizovaná, někdo všiml.

V druhém případě by si potřebná data aktualizovala služba samotná buďto periodicky na základě zadaného časového intervalu, nebo v okamžiku, kdy by služba zpracovávala požadavek na data o konkrétním uzlu. Výhodou tohoto přístupu by bylo, že bychom měli veškerá nastavení a funkcionalitu na jednom místě. Nevýhodou by mohla být zbytečná komplikovanost webové služby.

Důležitou funkcionalitou bude možnost aktualizovat pouze vybranou podmnožinu uzlů a ne celou kolekci uzlů v dané senzorové síti. Toto chování je důležité z důvodu úspory energie. V některých systémech je velice důležité co nejvíce omezit zbytečné dotazy na konkrétní uzly. Například systém ZigBee je navrhnut tak, že jednotlivé uzly komunikují s koordinátorem sítě v daných časových intervalech z důvodu efektivního využití energie. Bylo by tedy nelogické ve webové službě umožnit pouze aktualizace celé kolekce, obzvláště pokud bude uživatel požadovat pouze informace o jednom konkrétním uzlu.

6.3 Návrh vnitřní funkcionality webové služby

Jelikož tato webová služba není určena pouze pro předávání informací ze senzorových sítí k uživatelům, je potřeba navrhnout vnitřní funkcionalitu, která bude realizovat vzájemné převody mezi souřadnými systémy, ve kterých jsou umístěny uzly jednotlivých souřadných sítí. V počátku bude třeba implementovat tři hlavní komponenty převodu, které zmíníme v následujících kapitolách. Tyto komponenty budou do webové služby dodány v podobě třídni knihovny, neboli modulu implementující požadovanou funkcionalitu.

6.3.1 Převod mezi dvěma dvourozměrnými souřadnými systémy

Nejužívanější částí této webové služby bude bezesporu agregování souřadnicových systémů, ve kterých je poloha jednotlivých uzlů vyjádřena pomocí souřadnic typu x, y . V implementaci budeme předpokládat, že souřadný systém je pravoúhlý a jednotlivé souřadnice jsou vyjádřeny v metrech. Bude nutné implementovat dva případy převodu souřadného systému. V jednom případě budeme mít k dispozici polohu uzlu v jednom souřadném systému (posuv x a y souřadnic) a natočení vůči druhému souřadnému systému. Na základě těchto údajů vypočítáme polohu každého uzlu v druhém souřadném systému.

V případě, že nebudeme mít k dispozici údaje o vzájemné poloze dvou souřadných systémů, bude potřeba tyto informace dopočítat. K tomu bude zapotřebí minimálně 3 uzlů, které budou mít svoji polohu vyjádřenou v obou souřadných systémech, aby se daly potřebné údaje dopočítat, a pak následně ke každému uzlu, který bude mít svoji pozici uvedenou pouze v jednom souřadném systému, doplnit i souřadnice druhého souřadného systému.

6.3.2 Převod mezi dvěma trojrozměrnými souřadnými systémy

V případě souřadných systémů, ve kterých jsou polohy objektů vyjádřeny třemi souřadnicemi, je situace komplikovanější. Posuv v rámci tří souřadnic by problém nebyl, je zde ovšem možnost natáčet souřadný systém podél tří os. Z tohoto důvodu bude obtížné pomocí běžných metod vypočítat vzájemnou polohu těchto dvou souřadných systémů. V těchto případech budeme implementovat pouze první případ, tedy že informace o posuvu a rotaci máme k dispozici. V implementaci samozřejmě uvedeme i možnost převodu mezi dvou a trojrozměrnými souřadnými systémy.

6.3.3 Převod mezi globálními souřadnými systémy

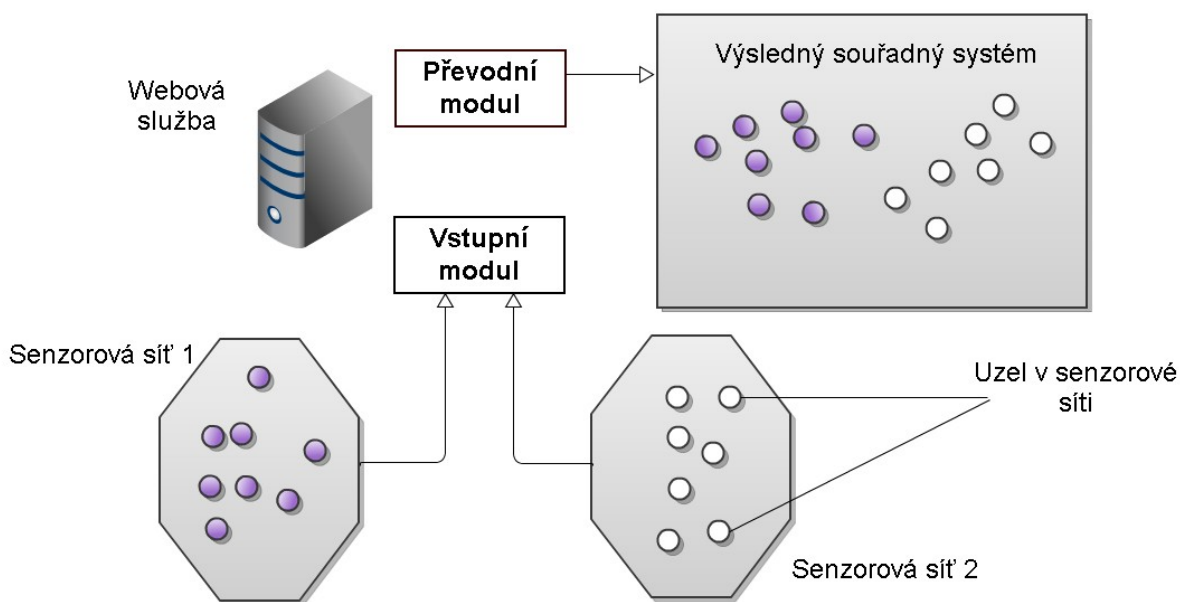
Samozřejmou součástí této webové služby bude převod mezi dvěma nejpoužívanějšími globálními souřadnými systémy, jedná se o souřadné systémy WSG-84 a UTM. Jakmile budou všechny uzly jednotlivých senzorových sítí agregovány do jednoho souřadného systému, bude umožněno přiřazení globální pozice, jejíž podobu budeme vyjadřovat dle nastavení ke každému uzlu. Toto ovšem bude možné pouze v případě, že budeme mít k dispozici alespoň 3 uzly s vyjádřenou polohou v jednom z výše uvedených globálních souřadnicových systémů. Pro přesnější vyjádření globální polohy jednotlivých uzlů by ovšem bylo vhodnější, aby uzlů s globální polohou bylo uvedeno více. Zamezí se tak nepřesnosti při přiřazování jednotlivých uzlů do globálního pozičního systému.

6.4 Modulárnost

Nelze předpokládat, že by každá síť přidělující polohu objektům patřící do dané sítě používala stejný systém vyjádření polohy. Také nelze předpokládat, že rozhraní či formát dat bude stejný. Z důvodů této obrovské variability v pozičních systémech je potřeba, aby tato služba umožňovala také značnou variabilitu. Toho je možné dosáhnout implementováním nejvíce užívaných formátů komunikace a uložení dat a budeme spoléhat na to, že tento přístup bude dostačující. Služba ale bude značně komplikovaná a bude konzumovat množství systémových prostředků. Druhou možností řešení tohoto problému je navrhnout webovou službu tak, aby podporovala modularitu, a tím by bylo možné do ní potřebné metody implementovat dodatečně na základě potřeby. Toto řešení může být obtížnější pro implementaci, ale ve výsledku bude umožňovat mnohem větší variabilitu při agregaci souřadných systémů jednotlivých sítí.

Z důvodu větší variability se autor práce rozhodl pro druhý přístup, navrhnout webovou službu tak, aby podporovala modularitu. Jelikož je návrh rozhraní pevně daný, budeme se zabývat pouze modularitou ve vnitřním fungování aplikace. Toto ovšem neznamená, že nebude umožněno pomocí modulu importovat nebo exportovat data o uzlech jinou cestou než přes implementované rozhraní.

Každý modul bude mít přístup k datové struktuře obsahující uzly jednotlivých sítí, budeme je rozlišovat na moduly, jenž komunikují s okolním světem, například budou moct importovat a exportovat uzly jednotlivých sítí, a na ty, jejichž úlohou bude agregovat souřadnicové systémy jednotlivých sítí, případně doplňovat k jednotlivým uzlům globální polohu.



Obrázek 12. Ukázka modularity, zdroj vlastní

7 Implementace

V této kapitole budeme rozebírat jednotlivé prvky implementace, popíšeme technologie potřebné k dílčím částem implementace, technologie použité pro vytvoření rozhraní webové služby a technologie, kterými zajistíme její modulárnost. Nakonec popíšeme funkcionalitu jednotlivých tříd nacházejících se v této implementaci.

Veškerá implementace byla realizována v jazyku C# za použití vývojového prostředí Visual Studio 2010 s využitím .Net Frameworku 4. Jelikož považujeme tyto 3 nástroje za všeobecně známé, nebudeme se jimi již dále zabývat. Místo toho se zaměříme na specifické technologie použité k vývoji této webové služby.

7.1 Rozhraní webové služby

Tato webová služba komunikuje na portu 80, tzn. na portu určeném pro protokol HTTP. Na rozdíl od služeb typu SOAP, je zde komunikace realizována pomocí HTTP požadavků. Jelikož používáme vyšší programovací jazyk, kde můžeme využít již vyvinuté nástroje, je zbytečné se zabývat detaily HTTP požadavků a odpovědí. Budeme se zabývat pouze daty přenášenými za pomoci těchto metod. Proto využijeme nástroj pro usnadnění implementace webových služeb WCF.

7.1.1 WCF

Tato kapitola je volně převzatá z [24]. WCF je ve softwarová struktura neboli framework využívaná při implementování aplikací zaměřených na poskytování služeb (SOA). Nabízí asynchronní zaslání dat z jednoho koncového bodu aplikace do koncového bodu jiné aplikace. Tyto aplikace mohou být jak webové služby, tak i procesy, jenž tyto webové služby konzumují. Cílem WCF je usnadnit komunikaci mezi těmito stranami, a tím umožnit jednotlivým stranám se zaměřit na datovou část komunikace místo na způsob přenosu těchto dat.

WCF komunikuje pomocí zpráv. Zpráva znamená skupinu dat obsahující záhlaví a tělo zprávy. Příkladem může být například HTTP požadavek. Tento framework rozlišuje klienta a službu, kde klient je aplikace začínající komunikaci a čekající na odpověď. Služba je aplikace, jenž disponuje několika koncovými body a čeká na příchozí požadavky, na které bude odpovídat přes dané koncové body. V tomto frameworku existují tři modely zaslání zpráv.

- Simplex - jednosměrné zaslání zprávy
- Duplex - asynchronní komunikace mezi klientem a službou
- Request - reply - synchronní komunikace mezi klientem a službou

Jak jsem již zmínil výše, komunikace probíhá přes koncové body (endpoint). Každý koncový bod se skládá ze tří parametrů.

- Adresy - Specifikuje adresu, kam zasíláme zprávy
- Vazby - Specifikuje komunikační mechanismus - protokol ve kterém mají být zprávy zasílány
- Kontraktu - Specifikuje sadu zpráv, jenž mohou být pomocí koncového bodu přijímány a odesílány.

Nastavení koncového bodu by mohlo vypadat následovně.

```
<endpoint address="http://localhost:80/library" binding="basicHttpBinding"
          contract="Client.Library.ILibrary" />
```

Příklad 1. Nastavení endpointu

7.1.2 Implementace rozhraní

Webová služba je již připravena na přijímání HTTP požadavků na portu 80, nyní je potřeba implementovat jednotlivé CRUD operace za pomoci funkcí `webget` a `webinvoke` implementující reakci webové služby na HTTP metody GET, POST, PUT a DELETE. Nyní si ukážeme nastavení daných funkcí při implementaci jednotlivých operací CRUD.

Operace Read

V kontextu této webové služby jde o čtení uzlů, případně celého seznamu uzlů. Webová služba nabízí výstup této operace ve dvou možných syntaxích (JSON, XML). Nastavení metody `webget` pro načtení celého seznamu uzlu, jenž vrátíme v podobě XML, by bylo následující.

```
[WebGet(UriTemplate = "xml/")]
public List<Node> GetCollection()
```

Příklad 2. Nastavení WebGet

Je zde vidět, že do funkce vracející daný seznam nebudeme předávat žádné parametry. To, že se data mají vrátit v podobě XML, je dosaženo pomocí tvaru URI. Všechny GET operace, které provedeme s URI začínající `xml/`, budou vracet výsledky v XML syntaxi. Adresa takového HTTP dotazu by mohla být například následující.

```
http://localhost/Compute/Combine/xml
```

Příklad 3. Adresa rozhraní webové služby

Pokud bychom zvolili URI `/json`, webová služba by vracela všechna data v syntaxi JSON.

V případě, že bychom chtěli načíst pouze jeden uzel, je potřeba specifikovat jeho identifikaci jak v rámci jeho sítě, tak i identifikace jeho původní sítě. V tomto případě by mohla nastavení metody WebGet vypadat následovně.

```
[WebGet(UriTemplate = "json/{ID};{NetworkID}")]  
public Node GetJsonNode(string ID, string NetworkID)
```

Příklad 4. Nastavení WebGet

V tomto případě vidíme, že jsme v části webové služby, která bude vracet data v JSON podobě. Do funkce GetJsonNode v tomto případě předáváme pomocí funkce webget proměnné ID a NetworkID, které předáme v podobě stringu. Tato funkce bude vracet jeden uzel.

Operace Create

Vytvoření nového uzlu, budeme realizovat pomocí funkce webinvoke, již dodáme příznak POST. Jednotlivá vstupní data budou oddělena středníkem.

```
[WebInvoke(UriTemplate = "{id};{networkID};{xPos};{yPos};{zPos}", Method =  
"POST")]  
public string SimpleCreate(string id, string networkID, string xPos, string  
yPos, string zPos)
```

Příklad 5. Nastavení WebInvoke u metody POST

V tomto případě vytvoříme jednoduchý uzel, jenž bude mít souřadnice pouze v jedné souřadné síti a bude bez vyjádření globální polohy.

Operace Update

Editace existujícího uzlu je realizována podobně jako operace create. Funkci webinvoke dodáme příznak PUT. I zde budou jednotlivá data oddělena středníkem.

```
[WebInvoke(UriTemplate = "{id};{networkID};{xPos};{yPos};{zPos}", Method =  
"PUT")]  
public string SimpleUpdate(string id, string networkID, string xPos, string  
yPos, string zPos)
```

Příklad 6. Nastavení WebInvoke u metody PUT

Pokud by se daný uzel nepodařilo najít v seznamu uzlů, data o editaci se zahodí. Pro vytvoření nových uzlů lze použít metodu Create.

Operace Delete

Poslední operace rozhraní maže vybraný uzel. Opět je realizována stejnou funkcí jako metoda create a update, tentokrát ovšem dodáme příznak DELETE.

```
[WebInvoke(UriTemplate = "{id};{networkID}", Method = "DELETE")]  
public string Delete(string id, string newtorkID)
```

Příklad 7. Nastavení WebInvoke u metody DELETE

Po zadání této metody se služba pokusí najít a smazat uzel se zadanými údaji.

7.2 Konfigurace webové služby

Jelikož webová služba nemá klasické rozhraní pro ovládání aplikace, kde bychom ji mohli nastavit, je potřeba službu konfigurovat jiným způsobem. Webové aplikace většinou mívají konfigurační soubor, jenž obsahuje veškerá nastavení pro danou aplikaci. Zde je veškerá konfigurace webové služby umístěna v souboru web.config, kde budou umístěny všechny informace potřebné jak pro agregování souřadných systémů, tak i pro chod služby. Je to například cesta k modulům, cesta pro soubor, do kterého zapisujeme nestandardní události, adresa rozhraní jednotlivých webových služeb a další. Syntax jedné položky nastavení by pak vypadal následovně.

```
<add key="LogPath" value="C:/inetpub/wwwroot/Compute/bin/App_Data" />
```

Příklad 8. Položka nastavení v souboru web.config

K získání informací ze souboru web.config, slouží třída Settings, jenž obsahuje metody pro získání jednotlivých dat v již potřebném datovém formátu. Byla vytvořena i univerzální metoda pro získání libovolného nastavení na základě zadaného klíče. V příkladu výše by se jednalo o "LogPath" a bude nám vrácena hodnota value. Tímto způsobem lze získat libovolnou položku z konfiguračního souboru. Třída Settings má ještě jednu důležitou vlastnost, umožňuje měnit nastavení uvedené v konfiguračním souboru. Toto je obzvláště užitečné, pokud některá nastavení budeme dopočítávat až v průběhu funkce webové služby, nebo kdybychom tato nastavení potřebovali v jiné části modulu a zároveň by je již nebylo možné dopočítat. Tento způsob konfigurace nám umožňuje libovolně manipulovat s nastavením webové služby.

7.3 Modulárnost webové služby

Jak jsme již zmínili v návrhu, hlavní předností této webové služby bude její schopnost přizpůsobovat se jednotlivým parametrům různých senzorových sítí, jelikož bude umožněno přidávat do webové služby moduly, jenž budou rozšiřovat její funkcionalitu. Tímto způsobem bude umožněno agregovat prakticky libovolné souřadnicové systémy, nebudeme omezeni pouze původní implementací, ale budeme moci kdykoliv připravit nový modul a chybějící funkcionalitu doplnit. Implementaci této funkcionality budeme realizovat za pomoci MEF. Co je to MEF rozebereme v dalším odstavci.

Informace o MEF jsem volně čerpal z [25]. MEF je framework, jenž slouží k vytvoření jednoduchých rozšiřitelných aplikací. Umožňuje použít moduly i bez nutnosti jejich konfigurace v původní aplikaci. Lze tedy přidávat funkce do již existujících aplikací bez nutnosti jakkoliv upravovat kód původní aplikace. Při patřičném nastavení není problém přidat modul s extra funkcionalitou i do běžící aplikace.

Zjednodušeně můžeme implementaci s pomocí MEF popsat následovně. V aplikaci, kterou budeme chtít rozšířit o případné moduly, připravíme rozhraní třídy, jejíž metody budou mít název a vstupní a výstupní hodnoty. Tyto funkce budeme volat při použití modulů.

Jakmile máme připravené rozhraní, je potřeba připravit postup pro nalezení a inicializaci jednotlivých modulů. Tento postup spočívá v projití zadané složky, nahrání všech souborů s koncovkou .dll a přidání do seznamu těch, které obsahují požadované rozhraní. K dispozici je identifikace jednotlivých modulů, záleží pouze na implementaci, jak je budeme využívat. Postup pro nalezení a inicializaci jednotlivých modulů lze libovolně opakovat.

Nyní můžeme vytvořit modul. Vytvoříme třídu s daným rozhraním a metodami uvedenými výše a přidělíme jí vlastní identifikátor. Tyto metody mohou mít samozřejmě jinou implementaci než metody v původní aplikaci, je ovšem důležité, aby měly stejný název, vstupní a výstupní hodnoty. Tuto třídu následně zkompilujeme jako třídní knihovnu. Následně tento zkompilovaný soubor umístíme do složky, kterou jsme vyčlenili pro moduly, a daný modul je možné použít.

7.3.1 Rozhraní modulů

Jelikož cílem modulárnosti této webové služby je umožnit různým modulům manipulovat s vnitřním seznamem uzlů webové služby, budeme proto odkaz na tyto moduly předávat pomocí tohoto rozhraní. V implementaci jsou definovány dva typy rozhraní, jeden pro získávání kolekce uzlů a druhý pro editaci kolekce uzlů. Jejich podoba je následující.

```
public interface IGetNodes
{
    List<List<Node>> GetNodes(List<List<Node>> nodeListCollection);
}
```

Příklad 9. Definice rozhraní pro získání hodnoty kolekce uzlů

```
public interface IModifyNodes
{
    List<List<Node>> ModifyNodes(List<List<Node>> nodeListCollection);
}
```

Příklad 10. Definice rozhraní pro editaci hodnoty kolekce uzlů

Jak již bylo uvedeno v návrhu, nedělitelnou součástí této webové služby musí být možnost aktualizovat hodnotu určené podmnožiny kolekce uzlů. Zde toto chování budeme umožňovat rozhraní pro získání dat o uzlu a rozhraní pro modifikaci dat uzlu.

```
public interface IGetNode
{
    Node GetNode(Node tempNode, int ID, int SensorNetworkID);
}
```

Příklad 11. Definice rozhraní pro získání hodnoty uzlu

```
public interface IModifyNode
{
    Node ModifyNode(Node tempNode, int ID, int SensorNetworkID);
}
```

Příklad 12. Definice rozhraní pro editaci hodnoty uzlu

Je důležité mít i možnost pojmenovat jednotlivé třídy, jenž budou implementovat operace dané tímto rozhraním. Proto musíme definovat rozhraní pro získání názvu metody.

```
public interface IGetMethodName
{
    char MethodName { get; }
}
```

Příklad 13. Definice rozhraní pro získání názvu metody

Tato možnost získání názvu metody je důležitá z důvodu možnosti ovládání jednotlivých modulů. Pokud by moduly neměly identifikaci, neměli bychom způsob, jak použít modul, který je momentálně potřeba. Tímto máme připravené rozhraní, které budeme používat k implementaci modulů pro tuto webovou službu. Každý takovýto modul budeme do služby importovat pomocí tříd DirectoryCatalog, AggregateCatalog a CompositionContainer. Podrobný postup je rozebrán ve zdrojovém kódu, zde pouze poznamenejme, že třída DirectoryCatalog slouží k objevení a importu

modulů a že třída AggregateCatalog a CompositionContainer slouží k zakomponování objevených modulů do již běžícího procesu.

7.4 Modul pro získávání informací ze senzorových sítí

I když má webová služba rozhraní umožňující přidávání uzlů, v případě agregace těchto senzorových sítí je žádoucí kontaktovat všechna rozhraní senzorových sítí a získat informace o jednotlivých uzlech. Adresy jednotlivých rozhraní v tomto případě budou v konfiguračním souboru webové služby. Rozhraní těchto senzorových sítí jsou implementována podobným způsobem jako rozhraní této webové služby, k potřebným informacím o uzlech budeme tedy přistupovat pomocí protokolu HTTP.

Jednotlivé webové služby představující senzorovou síť jsou implementovány pomocí standardu REST. Modul tedy sestaví HTTP GET požadavek na adresu rozhraní webové služby, jenž má na starosti vrácení celé kolekce uzlů. Následně odešle požadavek a čeká na odpověď. Při přijetí bude datová část odpovědi webové služby na požadavek GET vypadat následovně. Pro účely zkrácení textu zde ukazujeme pouze hodnotu jednoho uzlu, v odpovědi jsou ovšem všechny uzly v dané senzorové síti.

```
<ArrayOfNode      xmlns="http://schemas.datacontract.org/2004/07/SensorNetwork"
  xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <Node>
    <ID>10</ID>
    <SensorNetworkID>1</SensorNetworkID>
    <SecondarySensorNetworkID>0</SecondarySensorNetworkID>
    <XPos>0</XPos>
    <XPosSecondary>0</XPosSecondary>
    <YPos>43</YPos>
    <YPosSecondary>0</YPosSecondary>
    <ZPos>0</ZPos>
    <ZPosSecondary>0</ZPosSecondary>
    <GlobalPositionType>LonLat</GlobalPositionType>
    <GlobalPositionValue>49.2325058N;16.5708228E</GlobalPositionValue>
  </Node>
</ArrayOfNode>
```

Příklad 14. Datová část odpovědi na dotaz GET v XML syntaxi

V tomto pluginu tedy implementujeme parsování XML struktur uzlu. Pokaždé, když daný uzel načteme, je potřeba vytvořit objekt uzlu, který přidáme do seznamu objektů představující uzly, jenž nám byl předán z hlavní aplikace. Tímto způsobem získáme všechny uzly ze zadaných senzorových sítí.

7.5 Modul pro agregaci senzorových sítí

Metody tohoto modulu použijeme poté, co jsme načetli nebo aktualizovali informace o uzlech jednotlivých sítí. Tento modul má za úkol agregovat uzly jednotlivých sítí do jedné výsledné sítě. Pro potřeby této implementace předpokládáme, že souřadnice uzlů budou v pravoúhlé soustavě souřadnic, kde jednotkou bude jeden metr. Využíváme zde 3 různé třídy, kde se každá věnuje jinému problému. První kombinuje dvě sítě, jejichž uzly mají svoji polohu vyjádřenou ve dvojrozměrném souřadném systému. Druhá třída umožňuje kombinovat dvě sítě, kde mají uzly polohu vyjádřenou ve trojrozměrném prostoru. Poslední třída slouží k doplnění informací o globální poloze všech uzlů za předpokladu, že je příslušné doplnění možné. Tato možnost závisí na tom, jestli máme minimálně u 3 uzlů jejich globální polohu. Bez těchto uzlů není možné přiřadit uzlům jakékoliv sítě jejich globální polohu.

Tyto třídy využíváme tak, že vybereme jeden souřadný systém senzorové sítě, jenž bude primární. Všechny uzly ostatních sítí dostanou jako svoji sekundární polohu umístění v rámci této primární sítě. Postupně procházíme všechny sítě a za pomoci výše uvedených tříd doplňujeme informace ke každému jejich uzlu. Dále si ukážeme, jak jsou dané třídy implementovány.

7.5.1 Třída pro kombinaci dvou dvojrozměrných souřadnicových systémů

Funkcionalita této třídy bude klíčová pro celý modul. Budeme zde doplňovat informace o pozici uzlu v síti, kterou určíme jako primární. Před tím než doplníme k uzlům potřebné informace, je potřeba získat vztah mezi primární souřadnou sítí a sítí, jejíž uzly budeme chtít upravit. Bez této informace je nemožné umístit uzel do jiné sítě než do jeho vlastní. Do této informace patří posuv jednoho souřadného systému oproti druhému v rovině x, y a vzájemné natočení obou souřadných systémů.

Existují dvě metody pro získání těchto informací. První metodou by bylo ruční měření těchto parametrů. Zřejmou výhodou této metody je to, že již nejsou potřeba žádné další výpočty kromě samotného výpočtu umístění uzlu. Druhou metodou je postup, ve kterém budeme mít minimálně 3 uzly se sdílenou polohou. Na základě těchto uzlů vypočítáme parametry potřebné pro doplnění informací k jednotlivým uzlům. Čím více uzlů použijeme pro tuto metodu, tím budou získané parametry přesnější. Nyní k samotnému výpočtu parametrů.

Předpokládejme, že máme k dispozici alespoň 3 uzly, jež mají pozici v obou souřadných systémech. Pro dvourozměrné systémy jsou k výpočtu potřebné pouze 2 body, ale je lepší zpřesnit výpočet za pomoci více bodů. Nejprve si ukážeme výpočet natočení. Máme uzly 1 a 2. Jejich souřadnice budou X_{A1} , Y_{A1} a u druhého uzlu X_{A2} , Y_{A2} (souřadnice v primárním systému) a X_{B1} , Y_{B1} a u druhého uzlu X_{B2} , Y_{B2} (souřadnice uzlů v sekundárním systému). Nejprve vypočteme úhel natočení mezi osou X a směrnicí tvořenou body X_{A1} , Y_{A1} a X_{B1} , Y_{B1} . Úhel mezi osou X a směrnicí tvořenou body v primární síti označíme jako α . Úhel mezi osou X a směrnicí tvořenou body v sekundární síti označíme jako β . Ukážeme si výpočet úhlu, pro sekundární systém výpočet probíhá stejně. Samotný výpočet by vypadal takto.

$$\text{Vypočteme úhel v radiánech } \alpha_{\text{Radians}} = \text{Atan}\left(\frac{(X_{A1} - X_{A2})}{(Y_{A1} - Y_{A2})}\right) \quad (1)$$

$$\text{Převědeme radiány na úhel. } \alpha = \alpha_{\text{Radians}} * (180 / \Pi) \quad (2)$$

Poslední úpravou je zjištění, v jaké orientaci byla směrnice.

Pokud bude X_{A1} větší než X_{A2} a zároveň Y_{A1} je větší než Y_{A2} přičteme k α 90° .

Pokud bude X_{A1} větší než X_{A2} a zároveň Y_{A1} je větší než Y_{A2} přičteme k α 180° .

Pokud bude X_{A2} větší než X_{A1} a zároveň Y_{A1} je větší než Y_{A2} přičteme k α 270° .

Ve výpočtu β bychom postupovali stejným způsobem. Nyní máme k dispozici dva úhly. Úhel natočení nyní již zjistíme jednoduše. Je to absolutní hodnota rozdílu mezi úhly α a β . V postupu uvedeném výše jsme počítali pouze se čtyřmi body. V implementaci je umožněno počítat s libovolným počtem bodů, jelikož je žádoucí tento výpočet upřesnit. Na jeho základě budeme pracovat se všemi ostatními uzly sekundárního souřadného systému.

Nyní, když jsme vypočítali úhel posuvu, můžeme spočítat i posuv v osách X a Y. Ty spočítáme pomocí uzlu, jenž má body jak v primární, tak i sekundární síti. Použijeme polohu ze sekundární sítě. Jeho X a Y pozici natočíme pomocí matice pro natočení bodu v dvojrozměrném prostoru, kterou lze popsat následovně.

$$\begin{vmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{vmatrix}$$

Obrázek 13. Matice rotace, převzato z [26]

Touto maticí vynásobíme body X_{B1} , Y_{B1} , dostaneme dva rotované body X_{r1} , Y_{r1} . Od souřadnice bodu v primární síti X_{A1} odečteme X_{r1} a od souřadnice bodu v primární síti Y_{B1} odečteme Y_{r1} . Výsledky těchto operací jsou tedy X a Y posuv primárního souřadného systému vůči sekundárnímu souřadnému systému.

V momentě kdy máme spočítané parametry natočení a posunutí, je již snadné dopočítat polohu uzlu v primární síti na základě, jeho polohy v sekundárním uzlu. Využijeme této matice rotace.

$$\begin{vmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{vmatrix}$$

Obrázek 14. Zpětná matice rotace, převzato z [26]

Každé souřadnice X a Y v sekundární síti vynásobíme touto rotační maticí a přičteme k nim parametry posuvu. Výsledkem budou souřadnice v primární síti. Tímto tedy doplníme ke všem uzlům jejich polohu v primární síti.

Výše uvedený způsob platí pro agregaci dvou souřadných systémů, přesněji řečeno pro umístění uzlů jednoho souřadného systému do druhého souřadného systému. Tímto způsobem budeme postupovat pro všechny souřadné sítě, jejichž uzly chceme mít umístěné v primárním souřadném systému. Informace o rotacích v dvourozměrném systému souřadnic byly čerpány z [26].

7.5.2 Třída pro kombinaci dvou trojrozměrných souřadnicových systémů

Jelikož ne všechny souřadné systémy jsou dvourozměrné, bylo potřeba vytvořit i třídu zabývající se agregací systémů, jejichž objekty mají polohu definovanou pomocí tří proměnných. V tomto případě budeme při přiřazování primární souřadné sítě vycházet pouze z parametrů posuvu a rotace, které byly naměřeny ručně. Výpočet těchto parametrů ze společných bodů je mnohem složitější než v předchozím případě a ani se nám tento postup nepovedlo realizovat. Je problém zpětně vypočítat, které osy byly pootočený, jelikož pootočení jedné osy ovlivní polohu dvou dalších.

Jelikož se jedná o trojrozměrný prostor, budou místo jednoho natočení osy potřeba tři. Označíme je jako α β γ . Nyní budeme provádět 3 rotace podle jednotlivých os. Tyto rotace provedeme tak, že souřadnice ve tvaru X, Y, Z budeme násobit následujícími rotačními maticemi.

Rotace kolem osy X.

$$\begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{vmatrix}$$

Obrázek 15. Rotační matice pro rotaci kolem osy X, převzato z [27]

Rotace kolem osy Y.

$$\begin{vmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{vmatrix}$$

Obrázek 16. Rotační matice pro rotaci kolem osy Y, převzato z [27]

Rotace kolem osy Z.

$$\begin{vmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

Obrázek 17. Rotační matice pro rotaci kolem osy, převzato z [27]

Po rotaci podél všech os přičteme k souřadnicím uzlu jednotlivé posuvy v rámci daných os a tím vypočítáme souřadnice daného uzlu v primární síti. Opět přiřadíme všem uzlům jejich souřadnice v primární síti a s dalšími sítěmi postupujeme stejně. Informace o rotacích v 3D prostoru byly čerpány z [27].

7.5.3 Třída pro doplnění globální pozice jednotlivých uzlů

Tuto třídu budeme využívat v momentě, kdy všechny uzly sekundárních souřadných systémů budou mít přiřazenou polohu v primárním systému. K doplnění polohy bude zapotřebí minimálně tří uzlů, jenž budou mít vyjádřenou polohu v systému UTM nebo v systému šířka/délka. Tyto dva formáty budeme převádět mezi sebou za pomoci třídy GeoUTMConverter, jenž pochází z [28]. S uzly v primárním souřadném systému a možností převodu mezi globálními souřadnými systémy již není problém dopočítat globální polohu pro zbylé uzly.

Budeme postupovat podobně jako v případě agregace dvou dvourozměrných souřadných systémů. Globální souřadnice převedeme do systému UTM, pokud v něm už nejsou. Systém UTM se vyznačuje tím, že je to pravoúhlý souřadný systém, jehož jednotkou je metr. Budeme tedy systém UTM považovat za primární souřadný systém a současný primární systém, jako sekundární. Podle postupu uvedeného v kapitole 7.5.1 vypočítáme parametry posuvu a na základě těchto parametrů doplníme ke každému uzlu jeho primární souřadnice (souřadnice v systému UTM). Pokud bude webová služba nastavena tak, aby výstupní souřadnice byly ve formátu délka/šířka, převedeme je z formátu UTM pomocí výše uvedené knihovny.

8 Demontrace funkcionality

V této kapitole budeme prezentovat způsob užití a funkcionalitu webové služby. Webová služba byla primárně navržena a implementována tak, aby bylo možné ji uzpůsobit různým sítím a postupům jejich agregací. V této kapitole představím jeden takovýto postup. Budeme agregovat čtyři různé senzorové sítě, které budou polohu uzlů v nich položených vyjadřovat za pomoci pravoúhlého dvojrozměrného systému souřadnic. Tyto souřadnice budeme označovat jako x a y . Každá taková síť bude mít svoji identifikaci a identifikace bude také přiřazena jednotlivým uzlům. Každý uzel bude z počátku obsahovat pouze souřadnice označující jeho umístění v senzorové síti, do které patří. V průběhu agregace ovšem webová služba přiřadí jednotlivým uzlům i souřadnice v primární síti a následně i globální pozici. V následujících kapitolách se budeme věnovat jednotlivých souřadným sítím, jenž budeme používat pro demonstraci funkcionality webové služby pro podporu geolokace.

8.1 Přehled senzorových sítí

V této kapitole se budeme zabývat přehledem parametrů jednotlivých senzorových sítí, počátečním umístění jejich uzlů a vizualizací senzorové sítě se svými uzly. Také graficky znázorníme umístění uzlů v senzorové síti. Pro toto grafické znázornění jsem byla implementována aplikace která se spojí s webovou službou a získá kolekci uzlů, tuto kolekci poté zobrazí společně s hranicemi dané senzorové sítě.

8.1.1 Primární síť

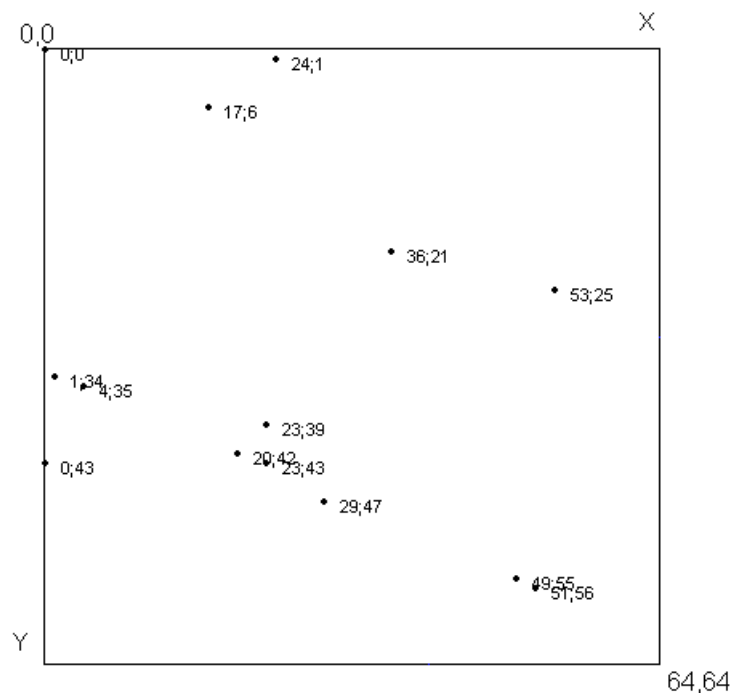
Primární síť je síť, na které bude založena tato demonstrace. Polohy uzlů z ostatních sítí budeme převádět právě do jejího souřadného systému. Tato síť bude obsahovat 10 uzlů s pozicí ve svém souřadném systému a navíc 4 uzly, ke kterým bude přiřazena globální poloha. Každá síť bude navíc obsahovat 4 uzly určující hranice souřadného systému, ty zde však přímo uvádět nebudeme, budou pouze viditelné jako body mezi přímkami tvořící hranici souřadného systému. Rozměry sítě budou 64 x 64 metrů.

V následující tabulce je uveden seznam obyčejných uzlů, uvedeny jsou hodnoty identifikace a poloha v X , Y souřadnicích.

1	2	3	4	5	6	7	8	9
16,83	51,48	53,46	19,8	3,96	28,71	40,59	0,99	35,64
5,94	56,43	24,75	41,58	34,65	46,53	37,62	33,66	20,79

Tabulka 1. Uzly primární sítě

Následující obrázek představuje grafické znázornění umístění uzlů v senzorové síti. Pro větší přehlednost jsou čísla zaokrouhlená na celé jednotky. Jednotkou v souřadném systému je metr.



Obrázek 18. Primární senzorová síť, zdroj vlastní

8.1.2 Sekundární síť

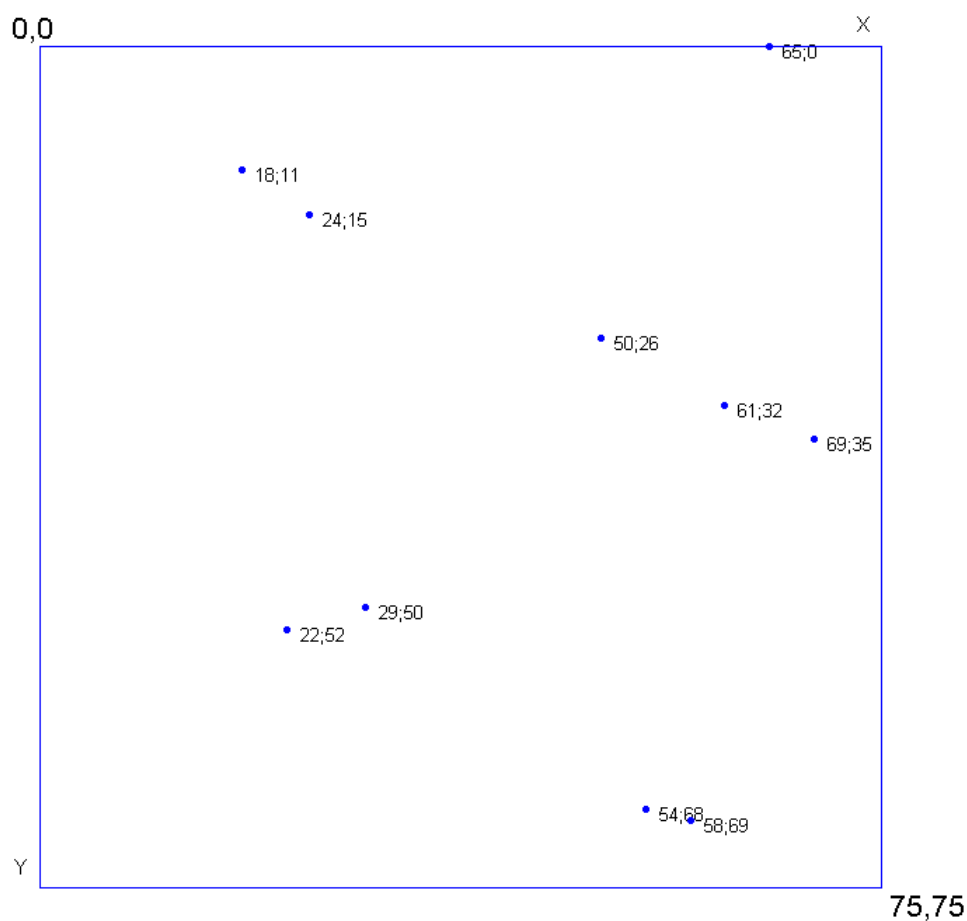
Tyto senzorové sítě budou obsahovat uzly, jejichž polohu se budeme snažit vyjádřit v primární síti. Budou celkově 3 a každá bude mít jiný vztah k primární síti. Žádné uzly těchto sítí nebudou mít k uzlům přiřazenou globální polohu. Tuto polohu přiřadíme až po agregaci těchto sítí s primární sítí a budeme tedy schopni vyjádřit polohu jednotlivých uzlů v primární síti.

První síť bude mít rozměry 75 x 75 metrů a bude obsahovat 10 uzlů. Tato síť bude vzhledem k primární síti pouze posunutá v ose X a ose Y.

Opět si tuto síť popíšeme tabulkou uzlů a vizualizací. Každá síť bude zobrazena jinou barvou. Jelikož budeme síť skládat, je potřeba, abychom je mohli od sebe ve výsledném obrázku odlišit.

ID	0	1	2	3	4	5	6	7	8	9
X	28,71	69,3	50,49	17,82	58,41	61,38	54,45	21,78	23,76	65,34
Y	50,49	34,65	25,74	10,89	69,3	31,68	68,31	52,47	14,85	0

Tabulka 2. Uzly první sekundární senzorové sítě

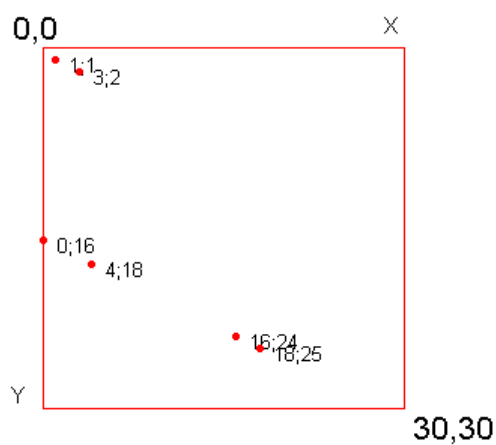


Obrázek 19. První sekundární senzorová síť, zdroj vlastní

Druhá síť bude mít rozměry 30 x 30 metrů a bude obsahovat 5 uzlů. Tato síť již bude vůči primární síti pootočená o zadaný úhel.

ID	0	1	2	3	4	5
X	2,97	0	0,99	17,82	3,96	15,84
Y	1,98	15,84	0,99	24,75	17,82	23,76

Tabulka 3. Uzly druhé sekundární senzorové sítě



Obrázek 20. Druhá sekundární senzorová síť, zdroj vlastní

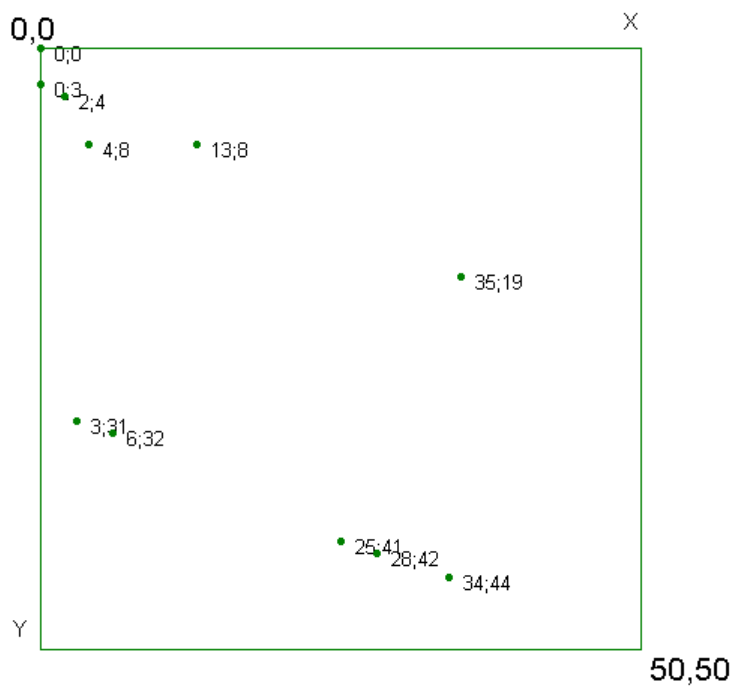
Poslední síť bude mít rozměry 50 x 50 metrů, a kromě klasických 8 uzlů bude obsahovat i 3 uzly, jenž budou mít sekundární polohu vyjádřenou v primární síti. Tyto uzly nebudou vyjádřeny v obrázku, jelikož nejsou uvnitř této sítě.

ID	0	1	2	3	4	5	6	7
X	33,66	34,65	12,87	0	27,72	5,94	24,75	2,97
Y	43,56	18,81	7,92	2,97	41,58	31,68	40,59	30,69

Tabulka 4. Obyčejné uzly třetí sekundární senzorové sítě

ID	8	9	10
X	0	2	4
Y	0	4	8
Xsec	15	13	11
Ysec	-30	-34	-38

Tabulka 5. Uzly třetí sekundární senzorové sítě s polohou v jiné síti



Obrázek 21. Třetí sekundární senzorová síť, zdroj vlastní

8.2 Konfigurace služby a modulů

Konfigurace webové služby má v tomto případě dva rozměry. V prvním rozměru budeme v konfiguračních souborech senzorových sítí určovat jejich parametry. V případě služby pro agregaci, budeme určovat parametry natočení a posuvu jednotlivých sítí mezi sebou. Nastavení každé webové služby se upravovalo v soubor web.config. Tato nastavení jsou v příloze 2.

Druhým rozměrem je implementace modulů pro tuto agregaci. Hlavní funkcionalitu vstupního modulu můžeme popsat takto

```
string[] networkPaths = settings.ReturnSensorNetworks();
foreach (var uri in networkPaths)
{
    List<Node> tempCollection = HttpGetNodeCollection(uri);
    AddNodes(ref nodeListCollection, tempCollection);
}
return nodeListCollection;
```

Příklad 15. Kód vstupního modulu

Do pole řetězců "networkPaths" předáme všechny adresy rozhraní senzorových sítí, které jsou v konfiguračním souboru. Následně všechny tyto adresy kontaktujeme ve funkci HTTPGetNodeCollection, která všechny získané uzly uloží do seznamu uzlů a tento seznam vrátí. Ve funkci AddNodes již pouze přesuneme do nodeListCollection, který vrátíme hlavní aplikaci.

Modul pro úpravu hodnot uzlů využívá následující postup. Uzly roztřídíme podle příslušnosti k jednotlivým sítím. Poté využijeme třídu TwoDimensionCombine. V následujících dvou případech dopočítá k uzlům daných sítí jejich polohu v primárním souřadném systému. Hodnoty posuvu a natočení získá modul z nastavení webové služby. Pro každé použití této třídy jsou hodnoty posuvu a natočení jiné. Pro přehlednost zde uvádíme pouze použití třídy. Parametry představují seznam uzlů, posun v souřadnici X, posun v souřadnici Y, úhel natočení, a primární síť.

```
var Combine = new TwoDimensionCombine(nodeList1, xShift, yShift, angle, 1);
Combine.ClearCombine(nodeList2, xShift, yShift, angle, 1);
```

Příklad 16. Kód modulu pro editaci uzlů

V případě čtvrté sítě nemáme k dispozici parametry posuvu a natočení. Budeme tedy třídu používat jinak. Do třídy předáme seznam uzlů obou sítí, seznam uzlů jenž mají souřadnice ve více sítích, ID primární sítě a ID sekundární sítě.

```
var mixedCombine = new TwoDimensionCombine(nodeList3, nodeList4, 1,4);
```

Příklad 17. Kód modulu pro editaci uzlů

V případě doplnění globální pozice ke všem uzlům není třeba nic konfigurovat, pouze výsledný seznam uzlů je třeba vložit do třídy GlobalPosUpdate spolu se seznamem uzlů s globální polohou.

8.3 Agregace senzorových sítí

Poté co jsme si ukázali detaily jednotlivých sítí, začneme s popisem agregace. Agregovat sítě budeme dvojím způsobem, buďto budeme novou polohu uzlu vypočítávat ze zadaných parametrů, které byly naměřeny, nebo budeme dané parametry vypočítávat z uzlů, jenž mají polohu definovanou v obou sítích. V případě druhé a třetí sítě budeme využívat naměřených parametrů, v případě čtvrté sítě budeme parametry dopočítávat z pozice uzlů. Pro každou síť si ukážeme příklad výpočtu alternativní polohy u jednoho uzlu.

Začneme s přepočtem polohy uzlů první sekundární senzorové sítě do souřadného systému primární sítě. Parametry jsou následující:

- Posuv v souřadnici X - 40 metrů.
- Posuv v souřadnici Y - 30 metrů
- Vzájemné natočení sítí - 0 stupňů.
- Uzel s polohou - X: 28,71 Y: 50,49.

Polohu uzlu v primární souřadné síti v tomto případě dostaneme pouhým přičtením hodnoty posuvu k jednotlivým hodnotám souřadnice. Tabulka uzlů s přepočítanými hodnotami vypadá následovně.

ID	0	1	2	3	4	5	6	7	8	9
X	28,71	69,3	50,49	17,82	58,41	61,38	54,45	21,78	23,76	65,34
Y	50,49	34,65	25,74	10,89	69,3	31,68	68,31	52,47	14,85	0
Xsec	68,71	109,3	90,49	57,82	98,41	101,38	94,45	61,78	63,76	105,34
Ysec	80,49	64,65	55,74	40,89	99,3	61,68	98,31	82,47	44,85	30

Tabulka 6. Uzly první sekundární senzorové sítě

V případě sekundární druhé sítě jsou parametry vzájemné polohy sítí následující:

- Posuv v souřadnici X - -30 metrů.
- Posuv v souřadnici Y - 50 metrů
- Vzájemné natočení sítí $\alpha = 75$ stupňů.
- Uzel s polohou - X: 2,97 Y: 1,98.

Výpočet nové polohy. Používáme matici otočení pro reverzní otočení, snažíme se tedy dostat originální souřadnice v primárním systému, které bychom použili, abychom po otočení o zadaný úhel dostali výchozí souřadnice.

$$X = \cos(\alpha) * X - \sin(\alpha) * Y = -31,14$$

$$Y = \sin(\alpha) * X + \cos(\alpha) * Y = 53,38$$

K těmto novým souřadnicím opět přičteme posun v osách a získáme polohu uzlu v souřadném systému primární sítě. X: -31,14 Y: 53,38. Jako v předchozím případě přikládáme tabulku uzlů třetí souřadné sítě.

ID	0	1	2	3	4	5
X	2,97	0	0,99	17,82	3,96	15,84
Y	1,98	15,84	0,99	24,75	17,82	23,76
Xsec	-31,14	-45,3	-30,7	-49,29	-46,19	-48,85
Ysec	53,38	54,1	51,21	73,62	58,44	71,45

Tabulka 7. Uzly třetí sekundární sensorové sítě

V případě poslední sekundární sítě již nebudeme mít k dispozici parametry natočení a posuvu, ale budeme mít k dispozici uzly, jenž mají polohu vyjádřenou v obou sítích, a bude tedy možné toto dopočítat. Jedná se o tyto uzly.

ID	8	9	10
X	0	2	4
Y	0	4	8
Xsec	-15	-17	-19
Ysec	30	34	38

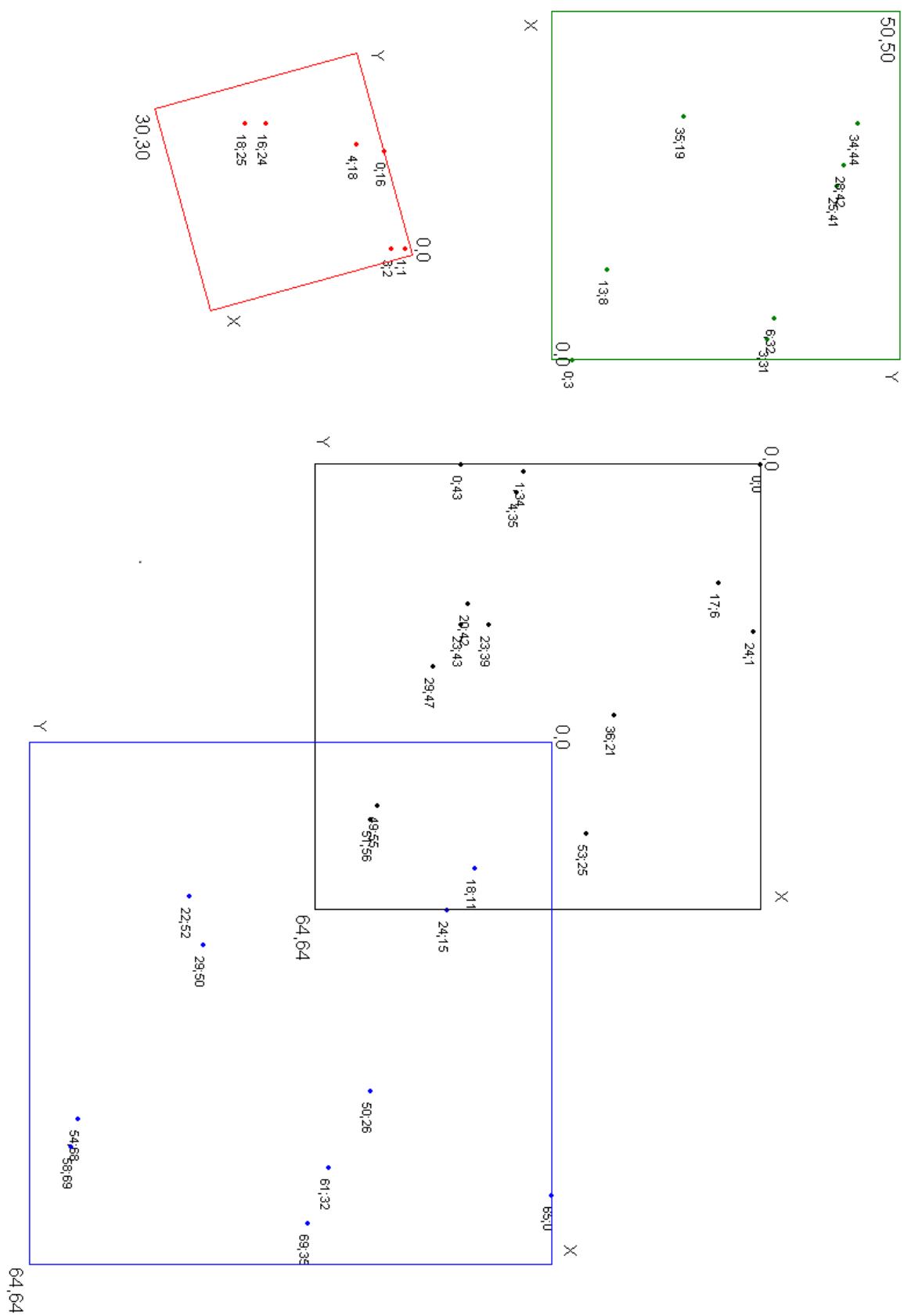
Tabulka 8. Uzly třetí sekundární sensorové sítě

Nejprve je potřeba vypočítat úhel α (úhel natočení mezi souřadnými systémy). Ten vypočítáme tak, že vypočítáme nejprve úhel, jenž svírá směrnice tvořená dvěma body v primárním souřadném systému s osou X. Poté spočítáme úhel, jež svírá směrnice tvořená dvěma body v sekundárním souřadném systému s osou X. Pro primární systém se tedy jedná o body [0,0] a [2,4], pro sekundární systém se jedná o body [-15,30] a [-17,34]. Výpočet těchto úhlů je popsán v kapitole 7.5.1. V primárním systému vychází úhel 63,435 a v sekundárním systému 243,435. Pokud od sebe tyto dva úhly odečteme, zjistíme, že souřadné systémy jsou otočené o 180 stupňů. Nyní opět zpětně natočíme souřadnice sekundárního systému a tyto souřadnice odečteme od souřadnic primárního systému. Získáme tak posun v souřadnicích. Nyní, když máme parametry, stačí pouze dopočítat primární souřadnice k ostatním uzlům jako v předchozích případech. Výsledná tabulka má následující hodnoty.

ID	0	1	2	3	4	5	6	7
X	33,66	34,65	12,87	0	27,72	5,94	24,75	2,97
Y	43,56	18,81	7,92	2,97	41,58	31,68	40,59	30,69
Xsec	-48,66	-49,65	-27,87	-15	-42,72	-20,94	-39,75	-17,97
Ysec	-13,56	11,19	22,08	27,03	-11,58	-1,68	-10,59	-0,69

Tabulka 9. Uzly třetí sekundární sensorové sítě

Nyní máme tedy u všech uzlů sekundárních sensorových sítí dopočítanou jejich polohu v síti primární, celkový výsledek by tedy po vykreslení vypadal následovně.



Obrázek 22. Výsledek agregace senzorových sítí, zdroj vlastní

8.4 Práce s globálními souřadnicemi

V této kapitole si ukážeme, jak modul webové služby pro agregaci sítí funguje pro globální souřadné systémy. V této kapitole budeme prezentovat, jakým způsobem budeme vypočítávat globální polohu uzlů, i když žádný uzel v jejich souřadném systému nemá globální polohu definovanou. Využijeme toho, že víme, jaké jsou vztahy mezi dvěma sítěmi a dopočítáme globální polohu uzlů druhé sítě právě na základě zadaných parametrů. Spojíme dvě souřadné sítě, z nichž pouze jedna z nich bude mít uzly umístěné v globálním souřadném systému. Souřadný systém, ve kterém budou umístěny uzly, bude délka/šířka, výsledný globální poziční systém, do kterého budeme převádět uzly druhé sítě, bude UTM. Tabulka uzlů prvního souřadného systému vypadá následovně.

ID	0	1	2	3
X	0	23	0	24
Y	43	43	0	1
Délka	49.2325058N	49.2326097N	49.2321608N	49.2322633N
Šířka	16.5708228E	16.5711231E	16.5710706E	16.5713653E

Tabulka 10. Uzly první senzorové sítě

Tabulka uzlů druhé souřadné sítě by vypadala následovně.

ID	0	1	2	3
X	11,88	27,72	14,85	1,98
Y	6,93	0,99	24,75	17,82

Tabulka 11. Uzly druhé senzorové sítě

Parametry posuvu a natočení mezi sítěmi.

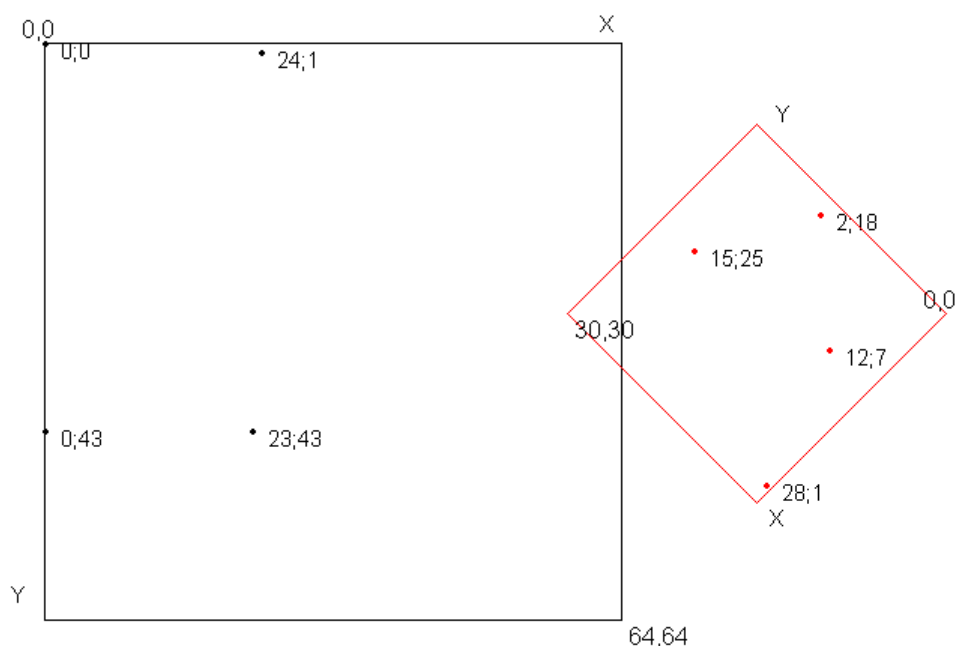
- Posuv v souřadnici X - 100 metrů.
- Posuv v souřadnici Y - 30 metrů
- Vzájemné natočení sítí $\alpha = 135$ stupňů.

Nyní opět dopočítáme polohu uzlů druhé sítě v síti první. Tento výpočet bude probíhat stejně jako v předchozích případech agregace sítí, není tedy již nutné ho tu znovu opakovat. Výsledná tabulka uzlů druhé sítě by poté vypadala následovně.

ID	0	1	2	3
X	11,88	27,72	14,85	1,98
Y	6,93	0,99	24,75	17,82
Xsec	86,7	79,7	72	86
Ysec	33,5	48,9	23	18,8

Tabulka 12. Uzly druhé senzorové sítě po výpočtu

Výsledný obrázek souřadných sítí.



Obrázek 23. Souřadné sítě 1 a 2, zdroj vlastní

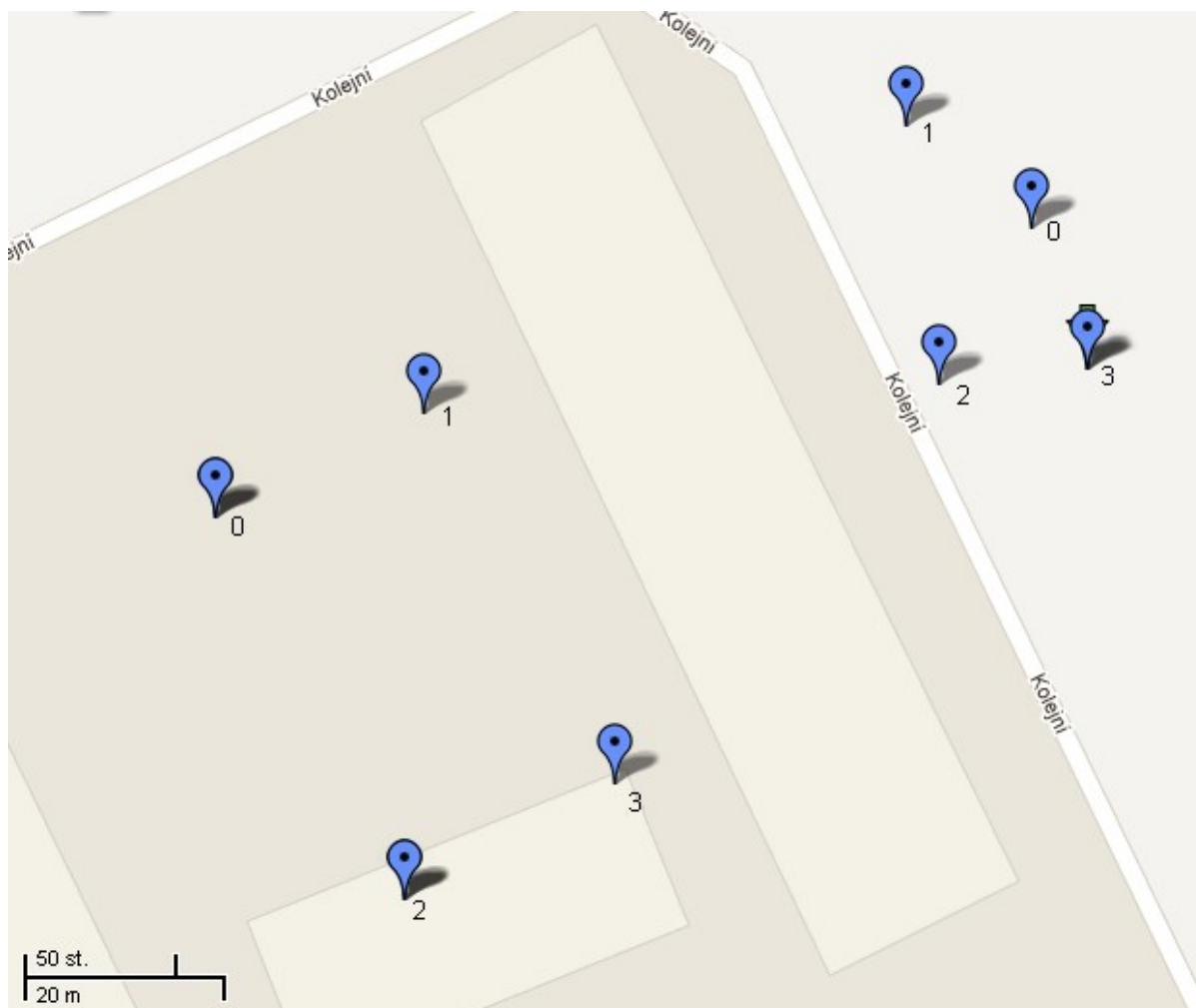
Nyní k samotnému propočtu globálních poloh druhé sítě. Nejprve převedeme globální polohu typu šířka/délka jednotlivých uzlů na typ UTM. Jak jsme již zmiňovali, UTM je pravoúhlý souřadný systém, jehož jednotkou je metr. V tomto se neliší od souřadných systémů našich sensorových sítí. Pro výpočet budeme tedy předpokládat, že systém UTM je primární systém a systém první sensorové sítě je sekundární. Vypočteme tedy parametry posuvu a natočení stejným způsobem, jaký jsme použili i pro předchozí výpočty v případě, kdy neznáme parametry, ale máme pouze uzly s polohou vyjádřenou v obou souřadných systémech.

Výsledná tabulka druhé souřadné sítě s doplněnými globálními souřadnicemi.

ID	0	1	2	3
X	11,88	27,72	14,85	1,98
Y	6,93	0,99	24,75	17,82
Xsec	86,7	79,7	72	86
Xsec	33,5	48,9	23	18,8
Zóna	33	33	33	33
UTM X	614437,22	614424,01	614428,86	614443,24
UTM Y	5454521,35	5454531,91	5454505,33	5454507,92
Hemisféra	Severní	Severní	Severní	Severní

Tabulka 13. Výsledné hodnoty uzlů druhé sensorové sítě

Tato tabulka ovšem není příliš vypovídající o správnosti převodu a přiřazení, které jsme tu provedli. Proto jsem všechny body umístil do mapy pro názornou ukázkou. Body první sensorové sítě jsou vlevo, body druhé sensorové sítě jsou vpravo. Oproti obrázkům, které jsme vykreslovali v předchozích případech, je osa Y orientována opačným způsobem.



Obrázek 24. Uzly senzorových sítí zaznamenané v mapě. Převzato z [29]

Tento obrázek slouží spíš k potvrzení, že jsme schopni body umístit do globálních souřadnic tak, aby jejich umístění odpovídalo umístění v lokálním souřadném systému. Pro ověření reálné přesnosti této metody by bylo potřeba využít skutečné uzly senzorové sítě, u nichž by bylo možné měřit jejich globální i lokální polohu zároveň a poté tato čísla porovnávat s vypočítanými hodnotami.

9 Závěr

V této práci jsme popsali možnosti geolokace v bezdrátových sítích. Ty jsme poté rozdělili na základě toho, zda můžeme použít již existující infrastrukturu a pouze uvést do provozu jednotky, které by dokázaly vysílat informace o své poloze. Druhou možností je instalovat kompletní infrastrukturu potřebnou pro geolokaci, jak je tomu v případě senzorových sítí. Tyto sítě je samozřejmě možné využívat i pro jiné účely než je pouze geolokace. Dalším bodem teoretické části diplomové práce byl popis různých způsobů vyjádření pozice uzlu, kde je možné použít jak lokální popis pozice uzlu, tak i možnosti vyjádření jeho globální pozice.

Posledním bodem teoretické části této práce bylo shrnutí informací o webových službách, způsob, jakým fungují, jaké protokoly jsou využívány pro jejich fungování a formát dat použitý při komunikaci.

V praktické části jsme navrhli a implementovali webové služby simulující pohyb uzlů v síti, jenž byly použity jako základ pro další praktickou část. Tím byl návrh a implementace webové služby umožňující agregaci těchto sítí. Webová služba byla navržena tak, aby podporovala modularitu a tudíž by umožňovala rozšiřovat její funkcionalitu. S ohledem na množství různých provedení senzorových sítí je takovýto přístup nutný. Webová služba byla navržena i s ohledem na senzorové sítě, jenž mají za cíl co nejnižší energetickou náročnost. Implementovali jsme tedy i možnost aktualizovat pouze podmnožinu uzlů v kolekci, při aktualizaci polohy uzlu tedy není nutné požadovat po senzorové síti aktuální polohy všech uzlů.

V poslední části jsme demonstrovali funkcionalitu modulu pro agregaci senzorových sítí včetně doplňování globální polohy k jednotlivým uzlům. Při vizualizaci agregace jsme implementovali vlastní způsob zobrazení senzorových sítí pro prezentaci výsledků práce. V druhé části demonstrace, tedy v práci s globálními souřadnými systémy, jsme vycházeli pouze z ručního měření v mapách, a proto výsledky jsou spíše orientační. Pro skutečnou demonstraci funkcionality webové služby v této oblasti by bylo potřeba tuto službu použít pro agregaci reálných senzorových sítí.

Jak jsme již naznačili v předchozím odstavci, službu by bylo dále možné rozšířit o moduly, které by dokázaly agregovat skutečné senzorové sítě různých druhů, a tím by tato webová služba získala na použitelnosti. V současné době lze na tuto webovou službu pohlížet jako na výchozí bod, ze kterého by se dala agregace různých senzorových sítí snadno implementovat. Dále by bylo možné přidat větší variabilitu při definici polohy uzlu. V současné době tato webová služba podporuje určení polohy v dvojrozměrném a trojrozměrném souřadném systému v případě lokální polohy. Globální pozici lze definovat buďto v systému zeměpisná šířka/délka nebo v systému UTM.

Literatura

- [1] Federal Aviation Administration. What is GPS? [online]. 2010 [cit. 2011-11-17]. Dostupný z WWW: <http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/faq/gps/index.cfm#1s>
- [2] Krzysztof W. Kolodziej, Johan Hjelm. Local Positioning Systems. Boca Raton: Taylor&Francis Group, 2006. ISBN 978-0-8493-3349-1
- [3] Petr Hanáček. Přednáškové materiály k předmětu BMS - BMSx04 GSM: FIT VUTBR.
- [4] Krzysztof W. Kolodziej, Johan Hjelm, ref. 2 s. 110
- [5] Jan Uhlíř. Přednáškové materiály k předmětu. Speciální číslicové systémy - ZigBee FELD VUTBR [cit.2011-12-20] Dostupný z WWW: <<http://noel.feld.cvut.cz/vyu/scs/prezentace2006/ZigBee/index.html>>
- [6] Capalini Richard. Lokalizace uzlů v senzorové síti ZigBee, diplomová práce, Brno, FIT VUT v Brně, 2010
- [7] Krzysztof W. Kolodziej, Johan Hjelm, ref. 2 s. 113
- [8] The Bluetooth Special Interest Group, Bluetooth basics [online]. 2011 [cit. 2011-12-13] Dostupný z WWW: <<http://www.bluetooth.com/Pages/Basics.aspx>>
- [9] Krzysztof W. Kolodziej, Johan Hjelm, ref 2 s. 112
- [10] Martin Hrubý. Studijní opora Geografické informační systémy: FIT VUT v Brně, 2008
- [11] Václav Čada, Přednáškové texty z Geodézie - Souřadnicové systémy. Západočeská Univerzita. [online]. 2011 [cit. 2012-01-02] Dostupný z WWW: <<http://gis.zcu.cz/studium/gen1/html/ch02s03.html>>
- [12] Bob Mesibov. UTM system. [online] Dostupná z WWW: Dostupný z WWW: <<http://www.utas.edu.au/spatial/locations/spautm.html>>
- [13] Wordpress. Cartesian Co-ordinate System. [online] Dostupný z WWW: <<http://sr789.wordpress.com/category/cartesian-co-ordinates/>>
- [14] Michal Imlauf. E-mailová brána informačního systému jako webová služba, bakalářská práce, Brno, FIT VUT v Brně, 2010
- [15] W3C. Web Services Architecture Part 1.4 What is a Web service? [online]. 2002-2004 [cit. 2012-01-03]. Dostupný z WWW: <<http://www.w3.org/TR/ws-arch/#whatis>>
- [16] KOSEK, Jiří. Inteligentní podpora navigace na WWW s využitím XML : Využití webových služeb a protokolu SOAP při komunikaci [online]. Vysoká škola ekonomická v Praze, 2002. 72 s. Diplomová práce. Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky. [cit. 2012-01-02] Dostupné z WWW: <<http://www.kosek.cz/diplomka/html/websluzby.html>>
- [17] W3school. SOAP Tutorial [online]. 1999-2010 [cit. 2012-01-03]. Dostupný z WWW: <<http://www.w3schools.com/soap/default.asp>>

- [18] W3school. WSDL Tutorial [online]. 1999-2010 [cit. 2012-01-03]. Dostupný z WWW: <<http://www.w3schools.com/wsdl/default.asp>>
- [19] W3school. WSDL Tutorial, WSDL and UDDI [online]. 1999-2010 [cit. 2012-01-02]. Dostupný z WWW: <http://www.w3schools.com/WSDL/wsdl_uddi.asp>
- [20] W3school. XML Tutorial [online]. 1999-2010 [cit. 2012-01-02]. Dostupný z WWW: <<http://www.w3schools.com/xml/default.asp>>
- [21] Alex Rodriguez. RESTful Web services: The basics [online]. 2008 [cit. 2012-01-07]. Dostupný z WWW: <<http://www.ibm.com/developerworks/webservices/library/ws-restful/>>
- [22] W3school. XML Tutorial [online]. 1999-2010 [cit. 2012-01-07]. Dostupný z WWW: <<http://www.w3schools.com/xml/default.asp>>
- [23] JSON[online]. 2006 [cit. 2012-01-07]. Dostupný z www: <http://www.json.org/index.html>
- [24] Marián Košťál. WCF - Základné pojmy [online]. 2007 [cit. 2012-5-13]. Dostupný z WWW: <<http://www.vyvojar.cz/Articles/454-wcf-zakladne-pojmy.aspx/>>
- [25] Microsoft. Managed Extensibility Framework Overview [online]. 2010 [cit. 2012-5-14]. Dostupný z WWW: <<http://msdn.microsoft.com/en-us/library/dd460648.aspx/>>
- [26] Nikos Drakos. Computer Graphics - Rotate [online]. 1994 [cit. 2012-5-15]. Dostupný z WWW: <<http://cs.fit.edu/~wds/classes/cse5255/thesis/rot/rot.html>>
- [27] Petr Minařík. Teorie 3D prostoru [online]. 2004 [cit. 2012-5-15]. Dostupný z WWW: <<http://programovani.net-mag.cz/?action=art&num=446/>>
- [28] Henning Mosand Stephansen. C# Geographic/UTM Coordinate Converter Class [online]. 2011 [cit. 2012-5-15]. Dostupný z WWW: <<http://www.m0sand.com/henningms/p=305>>
- [29] Google. Mapy Google.[online]. 2012[cit. 2012-5-20]. Dostupný z WWW: <<http://maps.google.cz/>>

Příloha 1

Adresářová struktura CD

- /src/ - Zdrojové kódy diplomové práce
- /bin/ - Přeložené binárky webových služeb
- /doc/ - Text této práce ve formátech .odt a .pdf
- /doxygen/ - Dokumentace generovaná v doxygenu

Příloha 2

Nastavení webových služeb

Toto je nastavení první senzorové sítě. Důležité jsou body:

- NetworkAreaSize označující velikost oblasti ve které se budou uzly pohybovat.
- NetworkMaxNodeCount - maximální počet uzlů v síti
- InitialNodeCount - výchozí počet uzlů
- NetworkID - ID sítě
- LogPath - cesta k souboru log
- GlobalPosNodeCount - počet uzlů v globální pozici
- Global a Local position - tyto hodnoty budou přiřazeny jednotlivým uzlům s globální pozicí
- Global Position type - typ globální informace

```
<appSettings>
  <add key="NetworkAreaSize" value="64" />
  <add key="NetworkMaxNodeCount" value="64" />
  <add key="InitialNodeCount" value="0" />
  <add key="NetworkID" value="1" />
  <add key="LogPath" value="C:/inetpub/wwwroot/1/bin/App_Data/Log" />
  <add key="GlobalPosNodeCount" value="4" />
  <add key="Global Position - 0" value="49.2325058N;16.5708228E" />
  <add key="Global Position - 1" value="49.2326097N;16.5711231E" />
  <add key="Global Position - 2" value="49.2321608N;16.5710706E" />
  <add key="Global Position - 3" value="49.2322633N;16.5713653E" />
  <add key="Local Position - 0" value="0;43" />
  <add key="Local Position - 1" value="23;43" />
  <add key="Local Position - 2" value="0;0" />
  <add key="Local Position - 3" value="24;1" />
  <add key="Global Position type" value="LonLat" />
</appSettings>
```

Příklad 18. Nastavení senzorové sítě 1

Nastavení druhé a třetí sensorové sítě.

```
<appSettings>
  <add key="NetworkAreaSize" value="75" />
  <add key="NetworkMaxNodeCount" value="64" />
  <add key="InitialNodeCount" value="10" />
  <add key="NetworkID" value="2" />
  <add key="LogPath" value="C:/inetpub/wwwroot/2/bin/App_Data/Log" />
</appSettings>
```

Příklad 19. Nastavení sensorové sítě 2

```
<appSettings>
  <add key="NetworkAreaSize" value="30" />
  <add key="NetworkMaxNodeCount" value="64" />
  <add key="InitialNodeCount" value="4" />
  <add key="NetworkID" value="3" />
  <add key="LogPath" value="C:/inetpub/wwwroot/1/bin/App_Data/Log" />
</appSettings>
```

Příklad 20. Nastavení sensorové sítě 3

Nastavení čtvrté sensorové sítě je trochu jiné než v předchozích případech, jelikož tato síť obsahuje uzly, jenž mají polohu vyjádřenou ve dvou souřadných systémech.

Obsahuje navíc body:

- SecondaryNetworkID - ID sekundární sítě
- CrossNetworkNodes - počet uzlů jejichž poloha bude vyjádřena v obou sítích

```
<appSettings>
  <add key="NetworkAreaSize" value="50" />
  <add key="NetworkMaxNodeCount" value="64" />
  <add key="InitialNodeCount" value="8" />
  <add key="NetworkID" value="4" />
  <add key="LogPath" value="C:/inetpub/wwwroot/4/bin/App_Data/Log" />
  <add key="CrossNetworkNodes" value="3" />
  <add key="SecondaryNetworkID" value="1" />
</appSettings>
```

Příklad 21. Nastavení sensorové sítě 4

Nakonec nastavení samotné webové služby pro agregaci.

Důležitými body jsou:

- PluginPath - cesta k modulům
- SensorNetworks - adresy rozhraní webových služeb
- XShift - posuv mezi první a druhou sítí, v tomto duchu jsou nastavena i ostatní posuvy
- GlobalPosSystem - systém, ve kterém bude zapsána výsledná globální poloha uzlů
- MainNetworkId - Sít', jenž bude považována za primární

```
<appSettings>
```

```
<add key="LogPath" value="C:/inetpub/wwwroot/Compute/bin/App_Data" />
```

```
<add key="Plugin Path" value="C:\inetpub\wwwroot\compute\bin\plugins" />
```

```
<add key="SensorNetworks" value=
```

```
"http://localhost/1/SensorNetwork/;http://localhost/2/SensorNetwork/;
```

```
http://localhost/3/SensorNetwork/;http://localhost/4/SensorNetwork/" />
```

```
<add key="MethodUsed" value="2D" />
```

```
<add key="XShift 1-2" value="4" />
```

```
<add key="YShift 1-2" value="3" />
```

```
<add key="ShiftAngle 1-2" value="0" />
```

```
<add key="XShift 1-3" value="15" />
```

```
<add key="YShift 1-3" value="10" />
```

```
<add key="ShiftAngle 1-3" value="60" />
```

```
<add key="GlobalPosSystem" value="UTM" />
```

```
<add key="MainNetworkId" value="1" />
```

```
</appSettings>
```

Příklad 22. Nastavení webové služby pro agregaci

Příloha 3

Instalace služby

K bezproblémovému chodu webové služby, je zapotřebí IIS 6.1 s instalovaným .Net frameworkem 4.0. Tato instalace vytvoří na hlavním disku adresář inetpub. V tomto adresáři je složka wwwroot, do které zkopírujeme soubory ze složky bin nacházející se na CD. Je nutné přidat práva pro čtení, zápis a spouštění pro uživatele IIS_IUSRS.

Nyní v IIS manageru je potřeba nastavit složky jednotlivých webových služeb, jako aplikace. Toto se provede pravým kliknutím na složku a volbou convert to application. Všechny aplikace patří do Application poolu ASP.NET v4.0 Integrated Pipeline. Pokud se tento aplikační pool v nabídce nenachází. Je potřeba spustit v konzoli následující dávku.

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis.exe -ir
```

Příklad 23. Dávkový příkaz

Po spuštění všech webových služeb by měl systém fungovat. Funkcionalitu je možné zkoušet pomocí přiložené aplikace Visualize, jenž umožňuje jednoduché zobrazení uzlů.

Adresy rozhraní pro jednotlivé služby jsou následující.

```
http://localhost/Compute/Combine/xml/
```

```
http://localhost/1/SensorNetwork
```

```
http://localhost/2/SensorNetwork
```

```
http://localhost/3/SensorNetwork
```

```
http://localhost/4/SensorNetwork
```

Příklad 24. Adresy rozhraní webových služeb