

**SZÉCHENYI ISTVÁN EGYETEM
MŰSZAKI TUDOMÁNYI KAR
JEDLIK ÁNYOS GÉPÉSZ, INFORMATIKAI ÉS VILLAMOSMÉRNÖKI
INTÉZET
MATEMATIKA ÉS SZÁMÍTÁSTUDOMÁNY TANSZÉK**



MŰSZAKI INFORMATIKA SZAK
Beágyazott programozás szakirány

DIPLOMAMUNKA

Helymeghatározás WiFi alapján egy digitális városban

Szabados Mátyás
mérnök-informatikus

2006

Feladat:

GPS-t helyettesítő, vezeték nélküli hálózatok hozzáférési pontjai alapján történő, pontos helymeghatározás, valamint egy erre épülő, kliens-szerver alkalmazás megtervezése és megvalósítása. A klienseket turistaként képzeljük el, akik a közelükben lévő helyi nevezetességekről kaphatnak információkat a szerver-alkalmazástól.

Részfeladatok:**a) A szakdolgozat elkészítéséhez szükséges résztvevőképességek:**

A GPS és WLAN helymeghatározásának megismerése. A helymeghatározáshoz szükséges háromszögelés elméleti háttérének bemutatása és az erre épülő algoritmikus megvalósítás lehetőségei. A kliens-szerver alkalmazás megtervezése és megvalósítása. Egyéb felhasználási lehetőségek.

b) A szakdolgozat főbb részei:

Elméleti áttekintés. A felhasználható Open Source C nyelvű forrásprogramjainak rövid bemutatása. A háromszögelési algoritmus megvalósítása. A feladatra megtervezett és elkészített programrendszer dokumentációja.

Belső konzulens: Pusztai Pál egyetemi adjunktus,

SZE MTK Matematika és Számítástudomány Tanszék

Külső konzulens: Szabó Péter szoftverfejlesztési igazgató,

Szintézis-Net Kft.

A diplomamunka benyújtásának határideje: 2006. december 15.

Győr, 2006. december 1.

.....

tanszékvezető

.....

szakfelelős

.....

szakirány-felelős

A diplomamunkát ellenőriztem: *beadható* – *nem adható be*

dátum

belső konzulens aláírása

A diplomamunkát ellenőriztem: *beadható* – *nem adható be*

dátum

külső konzulens aláírása

Bíráló adatai:
.....
.....

A diplomamunka minősítése

A bíráló javaslata:

éremjegy

dátum

bíráló aláírása

A Tanszék javaslata:

éremjegy

dátum

tanszékvezető aláírása

Az ÁVB határozata:

éremjegy

dátum

ÁVB elnök aláírása

SZABADOS MÁTYÁS
HELYMEGHATÁROZÁS WIFI ALAPJÁN EGY
DIGITÁLIS VÁROSBAN
ÖSSZEFOGLALÁS

A GPS-nek 1987-es megjelenése óta nem akadt ellenfele. A kezdetektől használják navigációra, térképek készítésére valamint a pontos idő meghatározására.

A technológia hátránya, hogy bár a nehezen kiépíthető infrastruktúrát már létrehozták, a felhasználó oldali hardver ára még mindig magas, illetve az optimális vételhez a mai zseb-eszközökhöz képest nagy antennára van szükség, így a mobiltelefon, smartphone, PDA gyártók többsége inkább a WiFi-t integrálja készülékeibe.

A GPS sűrűn beépített helyen, épületekben a leárnyékoltság miatt legtöbbször nem használható. Ezt a problémát próbálja orvosolni az egyre növekvő számban megjelenő WiFi hálózatok alternatív felhasználása segítségével - az ezen hálózatok hozzáférési pontjai által sugárzott információk alapján - ez az új technológia.

Segítségével nem csak GPS hardver nélkül határozhatjuk meg saját pozíciókat, hanem a WiFi-n keresztül csatlakozhatunk egy adathálózathoz, amelyről aztán hely-specifikus információkat kaphatunk.

A felhasználási lehetőségek száma határtalan, e technológia segítségével valóban digitálissá tehető egy város, integrálva mindezt a tömegközlekedésbe, a rendőrség, tűzoltóság, valamint a mentők kommunikációjába, illetve a városlakók mindennapjaiba.

MÁTYÁS SZABADOS
POSITIONING USING WIFI IN A DIGITAL
CITY
SUMMARY

Since the appearance of GPS in 1987, it has been unchallenged. From the beginning it has been used for navigation, mapping and determination of exact time.

However although the difficult to establish infrastructure already exist, the disadvantage of the technology is the relative high price of hardware for users, and the fact that a relatively large antenna is needed for optimal reception. This is the reason why most of the cellphone, smartphone and PDA manufacturers prefer to integrate WiFi into their products.

In most cases GPS does not work in urban areas or inside buildings due to shielding. This new technology tries to resolve this problem by using the information broadcasted by the access points of the growing number WiFi networks.

Using this technology, we will not only be able to determine our position, but by connecting to the WiFi network, access location specific informations.

Possibilities of usage have no end. With the help of this technology, a city can be made really digital, with integration into the public transport, emergency services and the everyday life of the citizens.

TARTALOMJEGYZÉK

1. BEVEZETÉS.....	3
2. A GPS ÉS A WLAN BEMUTATÁSA.....	4
2.1. A GPS.....	4
2.2. A WLAN.....	7
2.2.1. Általános bemutatás.....	7
2.2.2. A WLAN architektúrája.....	9
2.2.3. Biztonság.....	11
3. HELYMEGHATÁROZÁS.....	13
3.1. Háromszögelés.....	13
3.2. Gyakorlati alkalmazás földmérésnél.....	14
3.3. Gyakorlati alkalmazás WLAN esetén.....	15
4. HERECAST – AZ OPEN SOURCE MEGOLDÁS.....	18
4.1. Bemutatás.....	18
4.2. A HerecastLib fontosabb osztályai.....	19
4.2.1. AccessPoint osztály.....	19
4.2.2. AccessPointList osztály.....	19
4.2.3. WiFi osztály.....	19
5. A WPLACE SZOFTVER.....	21
5.1. Tervezés.....	21
5.1.1. Eszköz megválasztása.....	21
5.1.2. Fejlesztőrendszer megválasztása.....	22
5.2. Architektúra.....	23
5.2.1. Szoftver.....	23
5.2.2. Adatbázis.....	23
5.3. Kliens oldal.....	25
5.3.1. frmWplace osztály.....	25
5.3.2. frmAddAP osztály.....	27
5.3.3. frmAddPOI osztály.....	28
5.3.4. frmAboutYourLocation osztály.....	29
5.3.5. Osztálydiagramok.....	30
5.4. Szerver oldal.....	32
5.4.1. AP osztály.....	32
5.4.2. APlist osztály.....	32
5.4.3. Szerviz réteg.....	33
5.4.4. Adatbázis réteg.....	33
5.4.5. Osztálydiagramok.....	35
5.5. Felhasználói segédlet.....	36

TARTALOMJEGYZÉK

5.6. Továbbfejlesztési lehetőségek.....	38
6. ELÉRHETŐ KERESKEDELMİ TERMÉKEK.....	39
6.1. Ekahau.....	39
6.1.1. T301 - WiFi Location Tag.....	39
6.1.2. EPE – Ekahau Positioning Engine.....	39
6.1.3. ESS – Ekahau Site Survey.....	40
6.2. Skyhook.....	41
6.3. Navizon.....	41
7. KISLEXIKON.....	42
8. IRODALOMJEGYZÉK.....	44

1. BEVEZETÉS

Az emberek régóta vándorolnak, tesznek meg nagyobb távolságokat, előbb gyalog, majd lóháton, manapság pedig leginkább autóval. Először térképek és csillagállások alapján próbálták meghatározni helyüket, sebességüket, irányukat, többkevesebb sikerrel. A GPS megjelenése ennek a problémának tudható be, bár napjainkban többségében magánemberként használjuk, mint annyi minden, ez is a katonasághoz, a háborús technikai fejlesztésekhez köthető.

A navigáció elsősorban a valamilyen járművet vezetők tájékozódását segíti, bár nagy szerepe van a túrázóknál, kirándulóknál is. Ha az ember egy nagyobb városba látogat, ahol még soha nem járt előtte, nagy eséllyel eltéved. Amennyiben GPS segítségével tájékozódik, egy épületbe belépve valószínű elveszti a jelet, így egyből „vakká” válik. Tegyük fel, hogy ebben az épületben működnek vezeték nélküli hálózatok. Ezeknek a jelét emberünk tudja fogni, miért ne használná ezeket, hogy meghatározza helyét. Ehhez csupán arra van szükség, hogy legalább egy hozzáférési pont pontos földrajzi helyét tudja. Ez alapján meg tudja határozni, hogy körülbelül hol lehet, segítségül hívva a többi hozzáférési pont jelét, pedig teljesen pontos helymeghatározást is kaphat. Ettől még nem biztos, hogy tudni fogja, merre kell indulnia. A vezeték nélküli hálózatokra csatlakozva információt kaphat az épülettel, a közeli helyekkel kapcsolatban.

Napjainkban egy város tele van vezeték nélküli hálózatokkal, miért ne használhatnánk GPS helyett ezeket helymeghatározásra? A megfelelő szoftver rendszer létrehozásával a város számos elemét be tudjuk kapcsolni a rendszerbe, kihasználva a WLAN hatalmas előnyét, a kétirányú kommunikáció lehetőségét.

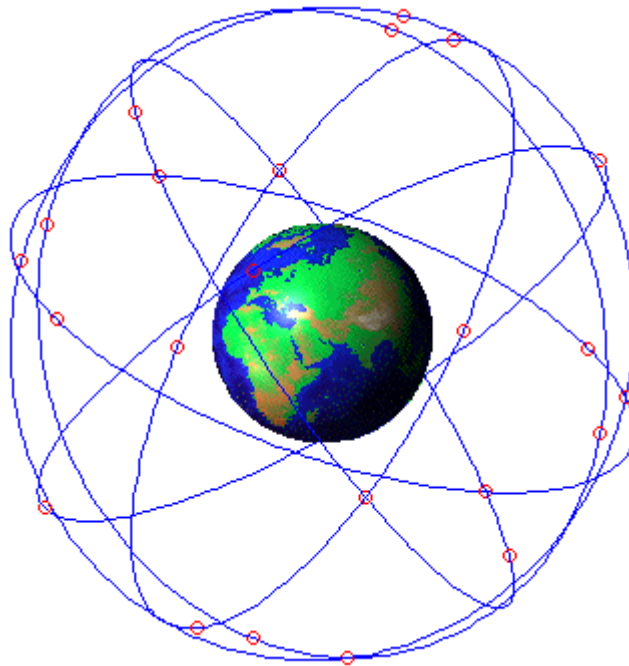
Képzeljük el, hogy moziba szeretnénk menni. A WiFi hálózatok és egy mobiltelefon, smartphone, vagy PDA segítségével megtudhatjuk a mozikban játszott filmek címét, valamint tartalmát. Az eszköz segítségével elnavigálhatunk a választott mozihoz és akár jegyet is válthatunk a nekünk tetsző előadásra, amennyiben a filmszínház erre lehetőséget biztosít.

2. A GPS ÉS A WLAN BEMUTATÁSA

2.1. A GPS

A GPS-t az Amerikai Védelmi Minisztérium fejlesztette ki a 1970-es években, annak érdekében, hogy járművei mindig tudják pontos pozíciójukat, valamint a vezetés is tudjon helyzetükről. E rendszer segítségével sikerült megnyerniük a Perzsa öbölháborút 1991-ben, mivel így éjszaka tudtak vonulni a sivatagban anélkül, hogy vizuális tárgyak alapján kellett volna tájékozódniuk.

A rendszert összesen 24 darab, 20200km-es távolságban lévő pályákon keringő műhold alkotja, melyekből egy adott pozícióból legalább 6 mindig látszik. A műholdak folyamatosan küldik hely és időadataikat a földfelszínre, amik alapján aztán a kliens oldali hardver háromszögelés segítségével meghatározza a koordinátákat, illetve a tengerszint feletti magasságot.



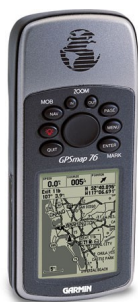
1. ábra : A 24 GPS műhold 6 pályán

Egészen 2000 májusáig a GPS jelek kódolva érkeztek a műholdakról, így az egyszerű felhasználó, csak kb. 100 méteres pontossággal tudta meghatározni helyét. A katonaságnak ugyanakkor a teljes dekódolásra alkalmas vevői voltak, amelyek jóval pontosabbak, kb. 20-30 méteres hibával működtek. A kódolás feloldásával bárki elérheti az eddig csak katonai célokra használt pontosságot, ugyanakkor a legtöbb gyártó így is csak kb. 17m-es pontosságot garantál. Hogy kiküszöböljék a GPS jelek különböző légköri, földi és hardver zavar okozta hibáit, bizonyos hibajavító megoldásokat fejlesztettek ki.

A legegyszerűbb a DGPS, ahol a hibák meghatározásában segít egy külön egység (beacon), mely a GPS jel hibáira vonatkozó információkat kapja egy földi állomásról, kiterjesztve ezzel a feldolgozható információk körét, növelve a pontosságot 3-5 méterre.

Ennél pontosabb a WAAS (WDGPS) (Európában EGNOS, Japánban MTSAT), ahol több földi helyről gyűjtenek információkat, majd ezeket elküldve egy geostacionárius műholdra, juttatják el a GPS vevőhöz. Ennek vételéhez nem szükséges külön egység, a vevő ezt a holdat is GPS műholdnak érzékeli, így egyik csatornáján tőle fogadja az adatokat, majd miután megállapítja hogy a többi műhold által sugárzott információk alapján melyik vonatkozik az ő aktuális pozíciójára, alkalmazza a hibajavításhoz. Ezzel a módszerrel már 1-3 méteres pontosságot lehet elérni.

A harmadik féle hibajavítás a LAAS, amely már elegendő pontosságot ad például egy repülőgép leszállásához. Ez a VHF frekvenciát használó technológia ugyancsak földi állomások korrekciós adataira támaszkodik, viszont itt az állomások elég kis területet fognak át. Ezt a módszert nem használják a civil életben.



2. ábra : Garmin GPSmap76



3. ábra : Garmin GPSmap276c

A Garmin cég a legismertebb gyártók egyike. Kínálatában megtalálható sok különböző célra készített GPS készülék: Túrázáshoz, hajózáshoz, autózáshoz, futáshoz, biciklizéshez. A vevőknek van színes vagy monokróm, térképes vagy térkép nélküli változatuk.

2.2. A WLAN

2.2.1. Általános bemutatás

Az első kvázi WLAN-t az 1970-es években hozták létre a Hawaii Egyetem kutatói, ez volt az ALOHA. Ez a hálózat egy dedikált központi gépet használt, hogy a többiek elérjék egymást. Az ALOHA, amellyel a vezetékes telefonvonalakat helyettesítették, még nagyban különbözött a mai WLAN-tól, mégis ez volt az első ilyen jellegű sikeres és valóban használható hálózat.

A mai WLAN szabványait a 1990-es években dolgozták ki, ezeket az IEEE 802.11 (WiFi) írja le. A szabványnak több változata van:

Szabvány	Frekvencia	Maximális sebesség	Tipikus sebesség	Kiadás éve	Hatótávolság
802.11	2,4 Ghz	2 Mbps	1 Mbps	1997	
802.11a	5 Ghz	54 Mbps	25 Mbps	1999	~50 m
802.11b	2,4 Ghz	11 Mbps	6,5 Mbps	1999	~100m
802.11g	2,4 Ghz	54 Mbps	25 Mbps	2003	~100m
802.11n	2,4 Ghz	540 Mbps	200 Mbps	2006	~50m

1. táblázat : A WLAN szabványai

Már az első szabványban is szerepelt a CSMA/CD csomag-ütközés védelmi protokoll használata és bár a csatorna megbízhatóbb lett, így odaveszett a valóban használható sebesség körülbelül fele.

A WLAN előnyei:

- *Kényelem:* Segítségével a nekünk tetsző helyről csatlakozhatunk a hálózatra, legyen ez egy kanapé van egy íróasztal.
- *Mobilitás:* A nyílt WLAN hálózatok segítségével nem csak otthonról vagy munkahelyünkről csatlakozhatunk az internethez. Napjainkban például sok kávézó is kínál vezeték nélküli internetet.
- *Produktivitás:* A felhasználó a WLAN-on keresztül bárhol elérheti saját céges vagy otthoni hálózatát, hozzáférhet például egy fontos árajánlat-hoz, amit e-mailen elküldhet egy partnerének.

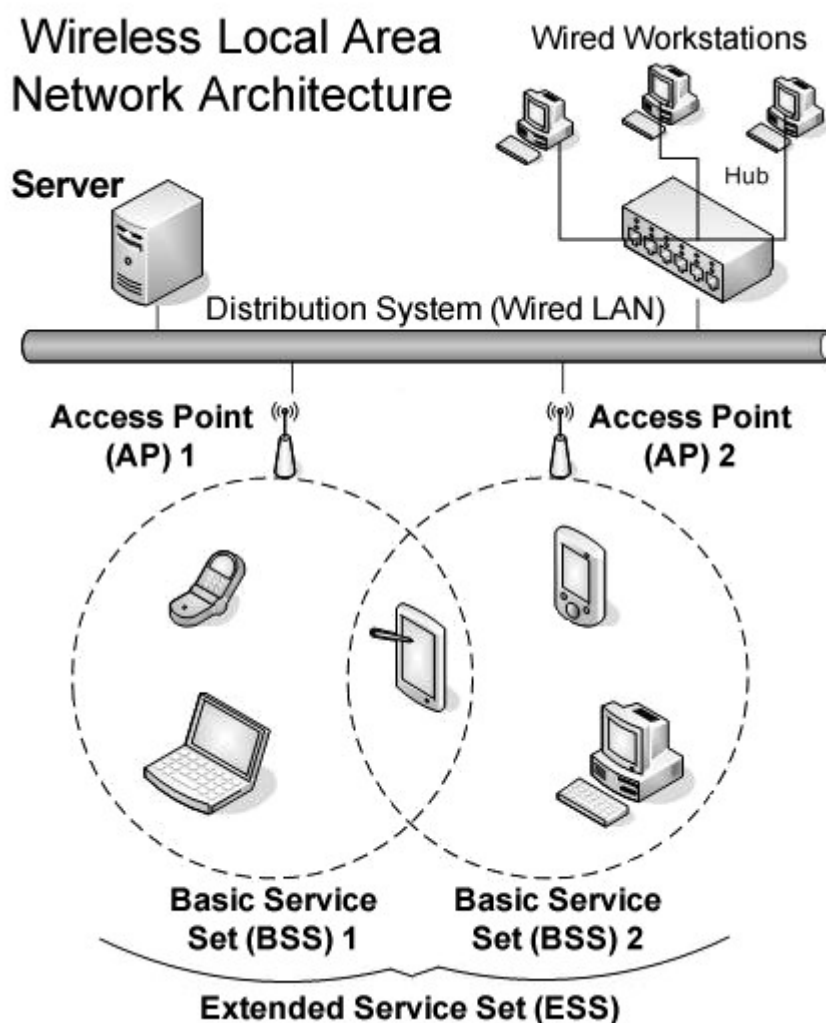
- *Telepítés:* Egy WLAN hálózat telepítéséhez nem sokkal többre van szükség egy Access Point-nál, míg a vezetékes hálózat esetén minden klienshez kábeleket kellett vezetni.
- *Bővíthetőség:* A vezeték nélküli hálózatok egyszerűen ki tudják szolgálni a megnövekedett számú felhasználókat, anélkül, hogy újabb berendezéseket kellene vásárolni.
- *Költséghatékonyság:* A vezeték nélküli eszközök ma már nem kerülnek sokkal többre vezetékes társainál, általuk időt és kábelezési költségeket takaríthatunk meg.

A WLAN hátrányai:

- *Biztonság:* A WLAN adók nem irányított sugárzást használnak, az információt minden irányba adják, hiszen nem tudják, hogy az általuk elérni próbált hardver merre található. Ezeket a sugárzott csomagokat a hálózat hatókörén belül bárki elkaphatja. Speciális irányított antennákkal a vételi körzet megnövelhető, így a csomagvadászok (*wardriver*) akár a hálózat hatókörénél tízszer távolabbról is elérhetik az információkat. Amennyiben ezek nincsenek titkosítva, úgy bárki bizalmas adatokhoz juthat.
- *Hatókör:* A WiFi hálózatokat is befolyásolják az épületek. Ez esetünkben kevésbé fontos, hiszen a minimális 1 Mbit-es sebességnél is pillanatok alatt átvihetők a kliensek által közölt helyre vonatkozó információk. A maximális sebesség eléréséhez tíz méteres távlatokban kell gondolkodni, így hamar észrevehetjük, hogy egy hálózat, nem elég egy nagyobb épület lefedéséhez. Ezt a problémát úgynevezett ismétlők beiktatásával szokták megoldani. Egy más célra szánt technológia a WiMax segítségével, akár több 100 kilométeres távolságokat is áthidalhatunk.
- *Megbízhatóság:* Mint minden rádiófrekvenciás technológia, a WiFi jelek is ki vannak téve a zavaró tényezők széles körének, csakúgy, mint a környezet hatásainak (visszaverődés). Ennél a típusú jelnél a bonyolult modulációs technológiák (PSK - fázis kulcs moduláció, QAM – kvadratúra amplitudó moduláció) ezt még rosszabbá teszik.

- *Sebesség*: A vezeték nélküli hálózatok sebessége (1-54 Mbps) jóval elmarad a vezetékesekéitől (100Mbps – 1Gbps). Bár vannak gyártó specifikus gyorsítások (D-link 108Mbps), a felhasználók többségének ez nem túl lényeges kérdés. Egy DSL előfizetés esetén a felhasználó és az internet közötti kapcsolat maximum 6 Mbps, így ez bőven elegendő. Ugyanakkor a helyi hálózatok történő nagyobb fájlok mozgatásához még mindig a vezetékes hálózat a jobb megoldás.

2.2.2. A WLAN architektúrája



4. ábra : A WLAN hálózati architektúrája

- *Állomások (Stations)*: Minden egység, amely a vezeték nélküli hálózatra kapcsolódik, egy állomás. Ezeknek az egységeknek rendelkezniük kell egy vezeték nélküli hálózati kártyával (WNIC). Az állomások két félék lehetnek: Kliensek, vagy Access Point-ok.
 - *Access Point*: A hozzáférési pontok központi egységei egy vezeték nélküli hálózatnak, ők adják és veszik a rádiófrekvenciás jeleket, amelyek által az Állomások kommunikálnak.
 - *Kliensek*: A vezeték nélküli kliensek mobil egységek lehetnek, mint például egy laptop, PDA, IP telefon, vagy lehetnek helyhez kötöttek, például egy asztali számítógép WNIC-kel.
- *Basic Service Set (Alap összeállítás)*: Ebben az összeállításban minden állomás kommunikálhat a másikkal. Két fajtája lehet: Az Independent (független) BSS, illetve az Infrastructure (központosított) BSS. A központosított BSS-nek saját azonosítója van (BSSID), amely megegyezik annak az Access Point-nak a MAC címével, amelyik kiszolgálja a BSS-t.
 - *Independent (független) BSS*: Ezek, az úgynevezett ad-hoc hálózatok nem tartalmazzák Access Point-ot, így nem tudnak másik BSS-hez kapcsolódni. Esetünkben ezek az összeállítások érdektelenek, hiszen az Access Point kivételével a többi állomás általában mobil, így nem lehet felhasználni helymeghatározáshoz.
 - *Infrastructure (központosított) BSS*: Az ebben lévő állomások más BSS állomásaival is tudnak kommunikálni az Access Point-on keresztül.
- *Extended Service Set (Bővített összeállítás)*: Az ESS egy BSS-ekből álló összeállítás, melynek Access Point-jai egy elosztó hálózaton (Distribution system) kapcsolódnak egymáshoz. Minden ESS-nek saját SSID-ja van, amely egy maximum 32 bájt hosszú karakterlánc.
- *Distribution System (elosztó hálózat)*: Ez köti össze az ESS Access Point-jait. Az elosztó hálózatok általában vezetékes hálózatok.

2.2.3. Biztonság

Vezeték nélküli hálózatok esetén a csomagok kódolatlanul utaznak a levegőben, lehetővé téve bárki számára, hogy elkapja őket és információt nyerjen ki belőlük. Ezt a problémát legegyszerűbben kódolással lehet megoldani, mely többféle lehet:

- *WEP (Wired Equivalent Privacy, Vezetékessel Egyenértékű Biztonság)*: Sajnos nevével ellentétben, vagy éppen azért ez a titkosítási forma nem nyújt túl nagy védelmet. Az előre meghatározott 128 vagy 256 bites kulcsokkal titkosított adatátvitel, némi lehallgatás után a kulcsok visszafejthetősége miatt dekódolható.
- *WPA (WiFi Protected Access, WiFi Védett Hozzáférés)*: Ez a titkosítás a TKIP (Temporal Key Integrity Protocol) segítségével már sokkal biztonságosabb. A WPA-PSK (Pre-shared Key, Előre kiosztott kulcs) egy 8-63 karakterig terjedő jelszót használ.
- *WPA2*: Az IEEE 802.11i szabványában rögzített titkosítási forma. AES-CCMP algoritmust használ a csomagok titkosítására. Ez a ma elérhető legjobb WiFi titkosítás.

A biztonság tovább növelhető hibrid megoldásokkal, ezekben a fent említett titkosítási formák egyike mellett az egész kapcsolat egy VPN-en keresztül folyik, amely egy nyílt kulcsú titkosító algoritmikus alapján azt a kommunikációt is kódolja.



5. ábra : Biztonságos WLAN

A zöld vonalon megvalósított kommunikáció biztonságos, így az illegális behatoló nem jut be, hála a WPA vagy WEP titkosításnak, a MAC cím blokkolásának, a 802.11x szabvány szerinti autentikációnak, vagy SSID titokban tartásának.

3. HELYMEGHATÁROZÁS

3.1. Háromszögelés

A háromszögelést Tycho Brache fedezte fel 1589-ben, de nem ismerték fel annak jelentőségét. Az első triangulációt Snellius használta 1617 körül fokmérésre. Ettől kezdve minden fokmérés háromszögelésre támaszkodik.

A háromszögelés minden nagyobb területre kiterjedő mérnöki munka első és legfontosabb része. Az egyes pontoknak egymáshoz viszonyított helyzetét meghatározó legtokéletesebb módszert a háromszögelés adja. Azért nevezzük így, mert három pontnak egymáshoz viszonyított helyzetét azonnal ismerjük, mihelyt tudjuk, hogy mekkorák a szögei és oldalai annak a háromszögnek, amit a három pont meghatároz.

A háromszögelés nem is áll másból, mint ilyen háromszögek oldalhosszának és szögeinek meghatározásából. Akárhány pontot kell is felvenni, azokat, ha csak nem esnek egy egyenesbe, mindig össze lehet kötni háromszögek bizonyos rendszerével és ha a háromszögek alkotó részeit mind ismerjük, akkor a pontoknak egymáshoz viszonyított helyzete is mind ismert lesz.

Összefüggő háromszögekben minden oldal hosszát ismerni fogjuk, amint a szögeket és egy hosszát ismerünk. Így az oldalak meghatározása egyetlen hosszának közvetlen megmérésére vezetődik vissza, ez az első lépés, az úgynevezett bázismérés. Erre az egy hosszra azután, mint alapra, bázisra támaszkodik a többi oldal hosszának meghatározása. A második lépés a szögmérés. Miután szögeket sokkal kisebb hibával tudunk mérni, mint hosszakat, igyekezzünk mindent lehetőleg szögmérés által meghatározni, hogy így a háromszögeléssel a lehető legjobbat érjük el.

3.2. Gyakorlati alkalmazás földmérésnél

A háromszöghálózat fix pontjai lehetőleg úgy legyenek megválasztva, hogy egyenlő oldalú háromszögek képződjenek, mert igen hegyes szögekkel nem lehet pontosan dolgozni. Azonkívül minden pontból lehetőleg sokat lehessen látni, hogy sok szöget mérhessünk meg. A pontok egymástól 30-60 kilométernyire fekdhetnek. A megállapított és felveendő fixpontokat jól alapozott kőépítményekkel jelölik meg.

Az elsőrendű háromszögelésnél a legfinomabb, nagy *teodolit*okat lehet csak használni. Minden szöget sokszor (20-24-szer) megmérünk még pedig a *teodolit* elkerülhetetlen kis hibáinak kiküszöbölése miatt.

A háromszögelést használják magasságmérésekre is. Egy pontnak a másik feletti való magasságát kiszámítani igen egyszerű, ha ismerjük a két pont vízszintes távolságát és azt a szöget, amit a két pont összekötő vonala a vízszintessel bezár. Itt tehát a magassági szögek megmérése a legfontosabb. Magassági szögek mérésénél pedig mindig tekintetbe kell venni a levegő vízszintes rétegeinek különböző sűrűsége miatt keletkező sugártörést. Távol fekvő pontok magasságának kiszámításánál azonkívül még a föld görbületére is figyelni kell.



6. ábra : Magyarország határmenti háromszögelési pontjai

3.3. Gyakorlati alkalmazás WLAN esetén

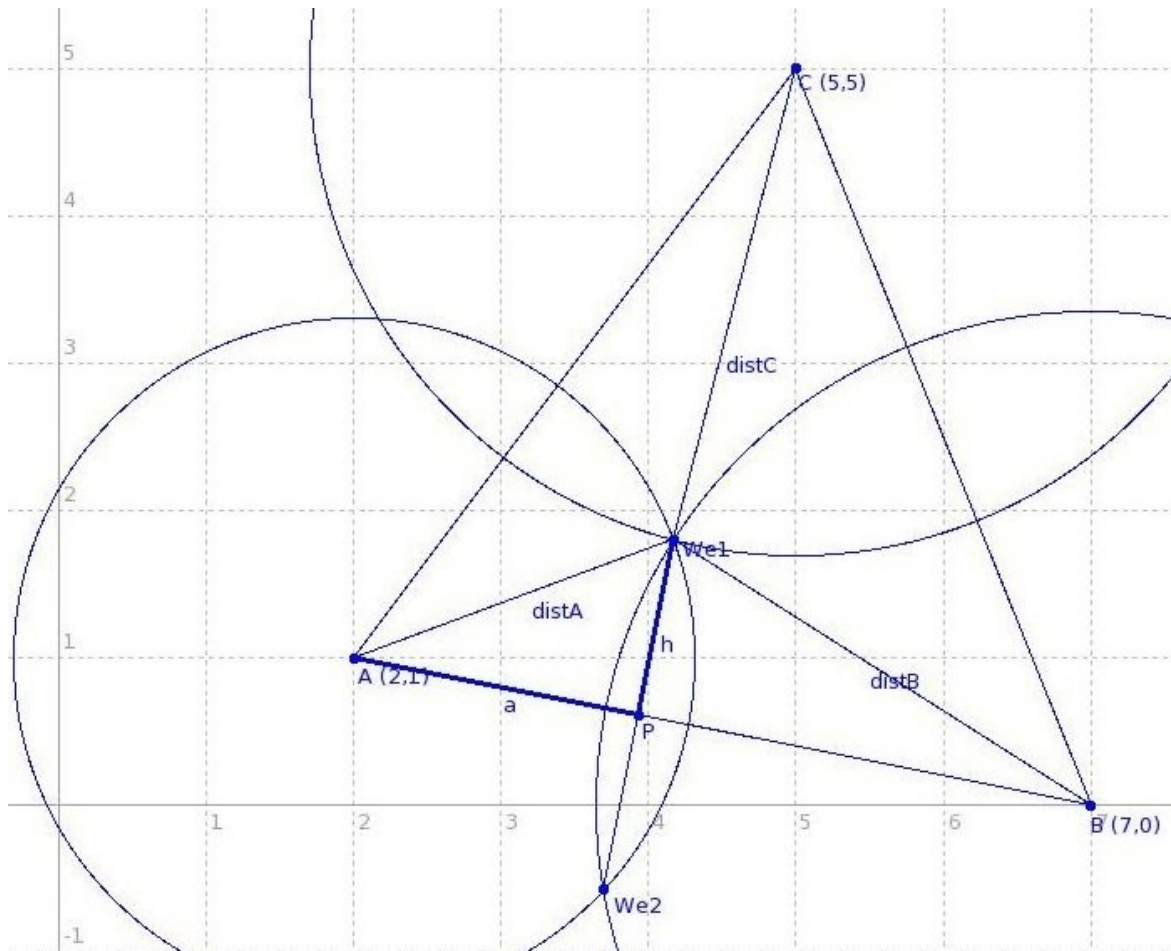
WLAN esetén a háromszögelés nem úgy működik, mint a földmérésnél. Mivel a kliens nem tud pontos távolságot mérni, szögekről pedig nem is beszélhetünk, más adatokra kell hagyatkoznunk. A hardver csupán az Access Point-ok BSSID-ját tudja kiolvasni, illetve a hozzájuk kapcsolódó jelerősséget tudja mérni.

A jelerősséget egy úgynevezett RSSI-ben kapjuk meg, ami egy 0-255-ig terjedő, 1 bájtos érték, ez hardverenként mást jelent. A gyártók általában nem használják ki mind a 256 értéket, ezért létezik egy RSSI_Max érték, amely a maximális jelerősségnek megfelelő RSSI értéket jelenti. Amennyiben rendelkezünk a hardver RSSI referenciatáblájával, ezeket az értékeket át tudjuk váltani mW vagy dBm értékekre. A kézi arányosításon alapuló átváltás azért nem lehetséges, mert ezek az értékek nem lineárisak. Sajnos ugyan így nem lineáris a jelerősség illetve a távolság összefüggése.

Egy Access Point adatai alapján csak egy kört tudunk meghatározni, ezen belül helyezkedünk el valahol.

Két Access Point esetén már meg tudnánk határozni két pontos koordinátát, amelyek közül az egyik helyes, azonban nem tudjuk megállapítani, hogy melyik az, így inkább a két Access Point között elhelyezkedő szakaszra tesszük pozíciónkat, a tőlük mért távolság segítségével pontosítva.

Amennyiben rendelkezünk legalább 3 Access Point adataival, amelyeknek ismerjük pontos helyét, úgy már nagy valószínűséggel meg tudjuk állapítani saját helyünket.



7. ábra : Háromszögelési példa

Az A , B , C pontok jelölik az Access Point-okat. A $We1$ és $We2$ pontok jelölik a lehetséges tartózkodási helyünket. A P egy köztes pont, mely a We koordinátáinak kiszámításában segít. A $distA$, $distB$, $distC$ Az általunk az Access Point-októl mért távolság.

Először az A és B Access Point koordinátái alapján ki kell számolnunk a távolságukat:

$$ABdistance = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

Ez után ki kell számolnunk az A Access Point és a P pont közötti távolságot:

$$a = \frac{(distA^2 - distB^2 + ABdistance^2)}{(2 * ABdistance)}$$

Szükségünk lesz még a P pont illetve a We pontok közti távolságra, a h -ra:

$$h = \sqrt{(distA^2 - a^2)}$$

Ezután már ki tudjuk számolni a We pontok koordinátáit:

$$x_{we} = x_P \pm \frac{h * (y_B - y_A)}{ABdistance} \quad y_{we} = y_P \pm \frac{h * (x_B - x_A)}{ABdistance}$$

Az így kapott 2 pont közül a C pont és a $distC$ segítségével döntjük el, hogy melyik vonatkozik ránk:

$$distC1 = \left| \left(\sqrt{(x_C - x_{we1})^2 + (y_C - y_{we1})^2} \right) - distC \right|$$

$$distC2 = \left| \left(\sqrt{(x_C - x_{we2})^2 + (y_C - y_{we2})^2} \right) - distC \right|$$

$distC1$ és $distC2$ kettő közül az vonatkozik ránk, amelyik kisebb, így már meg is határoztuk saját koordinátáinkat.

4. HERECAST – AZ OPEN SOURCE MEGOLDÁS

4.1. Bemutató

A Herecast helyfüggő szolgáltatásokat biztosít WiFi eléréssel rendelkező Windows CE, vagy Windows XP-t futtató PDA-k, illetve számítógépek számára. A rendszer tanult információk alapján megpróbálja megmondani, hogy a felhasználó körülbelül hol tartózkodik, és ehhez a helyhez tartozó szolgáltatásokat felajánlja.

A szoftver az ismert vezeték nélküli hálózat alapján mondja meg a tartózkodási helyet, nem koordináták segítségével, hanem sokkal barátságosabb szöveg-alapú kifejezésekkel.

A felhasználó a hálózatokhoz különböző információkat társíthat, ezek:

ország, megye, város, városrész (körzet) épület, cím, emelet, hely neve.

Ezeket az információkat később, amennyiben ehhez jogosultsággal rendelkezik, feltöltheti egy szerverre, majd innen bárki elérheti azt.

Kísérleti szolgáltatásként kifejlesztettek egy helyfüggő chat-et (HereSay), illetve egy barát keresőt (Friend Finder). A helyfüggő chat segítségével az abban a hálózatban tartózkodók cseveghetnek a szerveren keresztül, a barát kereső segítségével pedig a szerverről lekérdezhetők, hogy az adott nevű barátot (klienst) melyik hálózaton látták utoljára.

A szoftver az ingyenes Microsoft Embedded Visual Studio 4.0-ban készült, C++ nyelven, szabadon felhasználható. Rendelkezik egy HerecastLib nevű könyvtárral, amely a WLAN hardver és hálózat kezelő osztályokat tartalmazza. A könyvtárakat nem volt lehetőségem beépíteni a programrendszerben, mivel a Microsoft Visual Studio nem támogatja a beágyazott operációs rendszerű eszközök C++ nyelven történő programozását.

4.2. A HerecastLib fontosabb osztályai

4.2.1. *AccessPoint* osztály

Ez az osztály reprezentál egy Access Point-ot.

```
AccessPoint(UCHAR acMacAddr[6], INT iRssi, DWORD dTime);
UCHAR   _acMacAddr[6];    // BSSID (NDIS formátumban)
INT     _iRssiAvg;        // átlagos RSSI (e korábbi RSSI-ek alapján)
DWORD   _dLastSeen;       // utolsó találkozás ideje
```

4.2.2. *AccessPointList* osztály

Ez az osztály reprezentálja az Access Point-ok listáját.

```
AccessPointList(int size);
AccessPoint**  aplist;           // rekurzív lista AccessPoint* ele-
                                // mekből

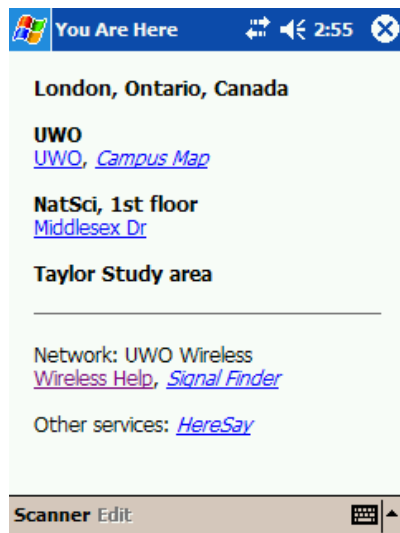
int            iMaxListSize;     // maximális elemszám
int            iCount;           // aktuális elemek száma
int            iMaxAge;          // maximális idő, amíg érvényes egy
                                // AccessPoint
```

4.2.3. *WiFi* osztály

Ez az osztály valósítja meg a vezeték nélküli hardver, valamint a működésére vonatkozó paraméterek lekérdezését az NDIS-en keresztül.

- `adatperAutoselectEnabled()`
Visszaadja, hogy a program automatikusan választja-e ki a használni kívánt WiFi adaptert.
- `bssidListScan()`
Meghívja az `OID_802_11_BSSID_LIST_SCAN`-t, ami egy listát ad az Access Point-okról.
- `bssidToManufacturer(UCHAR *MacAddress)`
A paraméterként kapott MAC címből próbálja meghatározni az eszköz gyártóját. Az egyes gyártók bizonyos tartományból adhatnak MAC címet eszközeiknek.
- `enumAllAdapters(bssidCallback callback, void *callbackData)`
Visszaadja az eszközben található vezetékes és vezeték nélküli eszközök listáját.
- `enumBssidList(bssidCallback callback, void *callbackData)`
Visszaadja azoknak a BSSID-knek a listáját, amelyek az adapter által ismertek.
- `frequencyToChannel(ULONG buflen, TCHAR *buf)`
A kapott frekvenciát a WiFi szabványban meghatározott csatorna számra konvertálja.
- `getAdapter()`
Visszaadja az aktuálisan használt adaptert.
- `getBssid(UCHAR MacAddress[])`
Visszaadja az aktuális társított BSSID-t.

- `getFirstWifiAdapter(ULONG buflen, TCHAR *buf)`
Visszaadja az első olyan adaptert, ami a `getSignalStrength()` hívásra nem hibával válaszol.
- `getRssiMax()`
Az RSSI maximális értékét adja vissza.
- `getRssiMin()`
Az RSSI minimális értékét adja vissza.
- `getSignalStrength()`
Az aktuális adapter által vett jel erősségét adja vissza dBm-ben.
- `getSignalStrengthFromAdapter(TCHAR *wName)`
A paraméterben kapott adapter által vett jel erősségét adja vissza dBm-ben.
- `loadAdapter()`
Meghatározza a használandó adaptert.
- `openNdisDevice(LPCTSTR lpDeviceName)`
Megnyitja a paraméterben kapott nevű NDIS eszközt.
- `recalibrateSignalMeter()`
A jelerősség mérő minimum és maximum értékét az alapértelmezésre állítja, így azok automatikusan újralibrálódnak.
- `rssiToPercentage(LONG dRssi)`
Az RSSImin és RSSImax értékek alapján százalékos formában adja vissza a jelerősséget.
- `setAdapter(TCHAR *ptcNewAdapter)`
Kiválasztja a használni kívánt adaptert.
- `setManufacturersFile(wstring ws)`
Beállítja azt a fájlt, amely a gyártó, MAC tartomány hozzárendeléseket tartalmazza.
- **WiFi()** - Konstruktor
Megnyitja az NDIS eszközt és kiolvassa a beállításokat a regisztrációs adatbázisból a későbbi függvényhívásoknak.



8. ábra : A Herecast által meghatározott pozíció

5. A WPLACE SZOFTVER

5.1. Tervezés

5.1.1. Eszköz megválasztása

A kliens oldalon használni kívánt eszköznél a következő dolgokat kellett mérlegelni:

- WiFi hardver
- Hordozhatóság
- Jó programozhatóság
- Kompatibilitás

A napjainkban használatban lévő PDA-k (Personal Data Assistant), illetve sokkal inkább a PocketPC-k 95%-án Microsoft Windows CE található. A most eladott gépek közül majdnem mindegyik alapfelszereltségként tartalmaz WiFi hardvert, ha nem, az operációs rendszer szoftveresen támogatja a Compact Flash illetve SDIO csatolófelületeken keresztül történő bővítést. Ugyancsak ennek az operációs rendszernek egy módosított változatával üzemelnek az úgynevezett okostelefonok, smartphone-ok, amelyek a WiFi-s adatátvitelen kívül GPRS, EDGE, vagy akár HSDPA adatátvitelt is támogatnak.

Az operációs rendszernek több változata van, a legfrissebbek: PocketPC 2000, PocketPC 2002, PocketPC 2003, PocketPC 2003 SE, Windows Mobile 5.0, illetve ezek SmartPhone változata, amely képes ugyanazoknak a bináris fájloknak a futtatására, mint alap változata.

A kód, módosítása nélkül, vagy apróbb módosításokkal újrarendíthető bármelyik változatra, illetve bármelyik processzor-architektúrára, amennyiben erre szükség van, bár a legnépszerűbb, 2002-2003-2003SE programjai módosítás nélkül futnak az operációs rendszer változatain.

Az összes ilyen eszköz akkora méretű, hogy az ember zsebre rakhatja, így bárhol, bármikor lehetősége van azt használni.

5.1.2. Fejlesztőrendszer megválasztása

A Microsoft nem csak külön SDK-t adott ki a Windows CE operációs rendszerhez, hanem integrálta a programfejlesztési lehetőséget a Visual Studio .NET fejlesztőkörnyezetbe. Az IDE-ben Smart Application Projectek segítségével hozhatunk létre PocketPC-n futó alkalmazásokat, amelyeket C# illetve Visual Basic nyelven kódolhatunk. A Wplace szoftver megírására a C#-ot választottam, mivel úgy gondolom ez az a nyelv, amely elég fejlett és letisztult ahhoz, hogy komolyabb alkalmazást lehessen írni vele. A Visual Studio-ban lehetőség van az alkalmazás emulátorban történő futtatására, vagy akár közvetlenül az eszközön is futtathatjuk fejlesztés alatt lévő programunkat.

A szerver oldal fejlesztésére szintén kézenfekvő megoldást nyújt a Visual Studio, hiszen C# nyelven lehetőség van webszolgáltatások fejlesztésére ASP.NET keretrendszerben. Így egyszerűbben meg lehet valósítani a kapcsolatot a kliens és szerver oldal között.



9. ábra : HP iPAQ1940 PocketPC,
Microsoft Windows PocketPC
2003

5.2. Architektúra

5.2.1. Szoftver

Az alkalmazás objektum orientált gondolkodás mód szerint készült, a C#-ban minden lehetőség adott volt ennek megvalósítására.

A programrendszer 3 rétegből (tier-ből) áll:

- Kliens oldali réteg
- Szerver oldali réteg
- Adatbázis réteg

A kliens oldalon történik mindenféle felhasználói beavatkozás, innen kapja a szerver oldal a helymeghatározáshoz szükséges adatokat, vagy éppen az új hálózatok helyét, vagy az új POI-k (Point Of Interest) adatait.

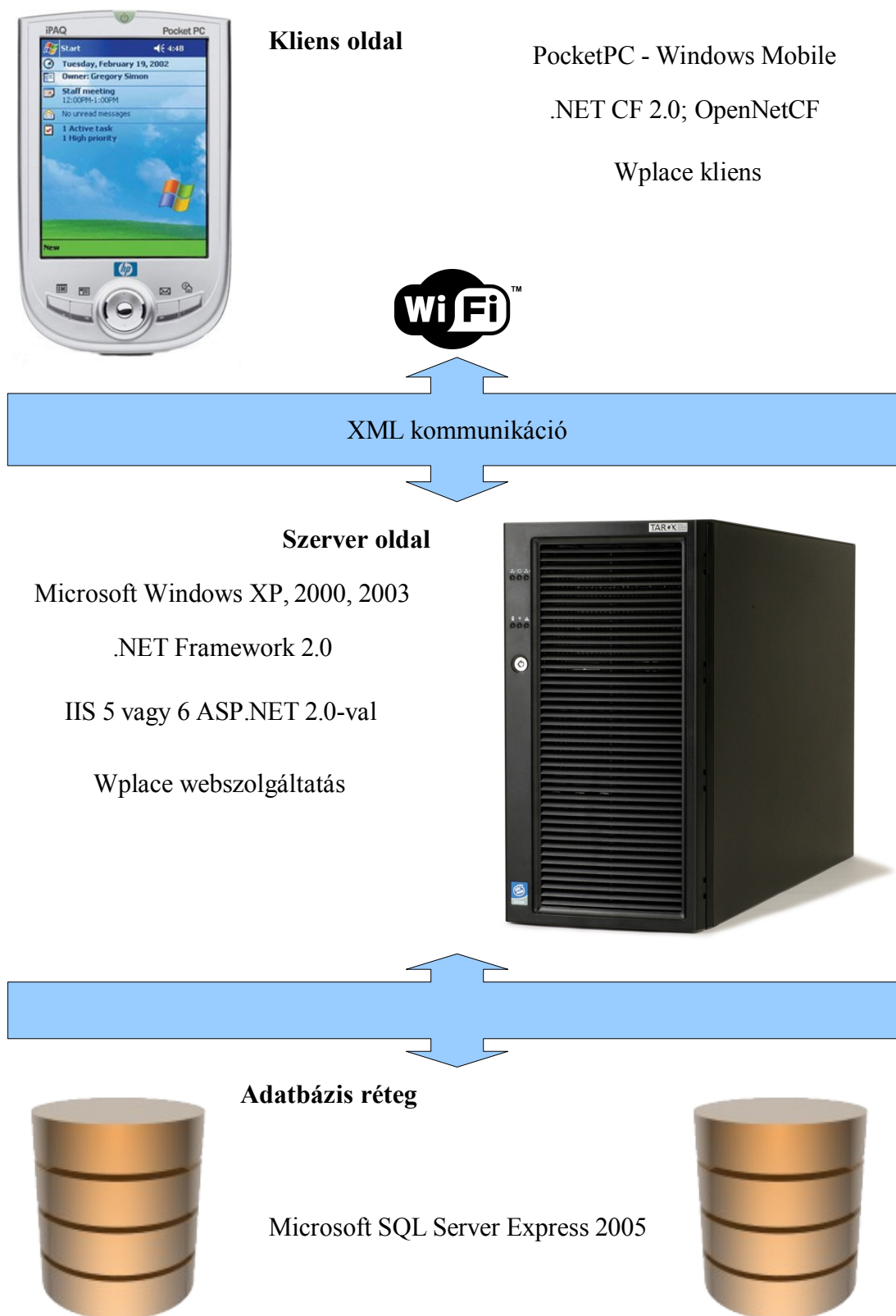
A szerver a klienssel XML-ben kommunikál webszolgáltatásokon keresztül, ez implicit módon történik. Ez az oldal gondoskodik az adatbázis rétegtől lekérdezett adatok alapján történő számolásról, majd a kliens pontos információkkal való ellátásáról.

Az adatbázis réteg kommunikál az SQL adatbázissal, kezelve az APs és a POIs táblát.

5.2.2. Adatbázis

APs TÁBLA	Ez a tábla tartalmazza az Access Point-ok adatait.		
apName	nvarchar	30 hosszú	Az AP nevét tartalmazza
apBSSID	nvarchar	15 hosszú	Az AP BSSID-ját tartalmazza
apX	double		Az AP X koordinátája
apY	double		Az AP Y koordinátája
apDesc	nvarchar	255 hosszú	Az AP leírását tartalmazza.

POIs TÁBLA	Ez a tábla tartalmazza a POI-k adatait.		
poiNama	nvarchar	30 hosszú	A POI nevét tartalmazza
poiX	double		A POI X koordinátája
poiY	double		A POI Y koordinátája
poiURL	nvarchar	100 hosszú	A POI-hoz tartozó URL.
poiDesc	nvarchar	255 hosszú	A POI leírását tartalmazza.



10. ábra : A Wplace 3 rétegű (3-tier) architektúrája

5.3. Kliens oldal

5.3.1. *frmWplace* osztály

Ez az osztály tartalmazza az alkalmazás főképernyőjét. A program elindulása-kor ezt példányosítja a *Program* osztály *Main* metódusa.

Az osztály adattagjai:

```
private AdapterCollection AcAdapters = Networking.GetAdapters();  
private wrfWPlaceService.Service ws = new wrfWPlaceService.Service();
```

Az osztály metódusai:

```
public frmWPlace()
```

A form konstruktora, inicializálja a vizuális elemeket.

```
private void Form1_Load(object sender, EventArgs e)
```

A form betöltéskor hívódik meg, inicializálja a `ListView` elemet, majd meghívja a `FindNetworks` metódust, amely megkeresi az elérhető WiFi hálózatokat.

Paraméterként kapja a küldő objektumot, illetve az esemény argumentumait.

```
private void InitListView()
```

Inicializálja a `ListView` elemet, létrehozza az oszlopait, beállítja azok fejlécét.

```
private void FindNetworks(object sender, EventArgs e)
```

Megvizsgálja az eszközben található hálózati adaptereket, majd kiválasztja az első vezeték nélkülit. Ezután megkeresi az elérhető WiFi hálózatokat, azokat betölti a `ListView` elembe.

Paraméterként kapja a küldő objektumot, illetve az esemény argumentumait.

```
private void btnSend_Click(object sender, EventArgs e)
```

A küldés gomb megnyomásakor hívódik meg. A `ListView` első 3 elemét elküldi a webszerviznek, majd a visszatért értékekkel meghívja a `FindPOI` webszervizt, ennek segítségével megkeresi a legközelebbi POI-t és beállítja azt legutóbb látottnak. Ezek után meghívja a `ShowLocation` metódust.

```
private void ShowLocation(double[] Coordinates, string Name)
```

Példányosítja és előtérbe helyezi a `frmAboutYourLocation` form-ot.

Paraméterként kapja az aktuális pozíció koordinátáit, valamint a legközelebbi POI nevét.

```
private void AddAPmenu_Click(object sender, EventArgs e)
```

Az Access Point hozzáadása menüpont kiválasztásával hívódik meg, példányosítja az `frmAddAP` form-ot, majd előtérbe helyezi. Amennyiben volt Access Point kijelölve a `ListView` elemben, úgy átadja annak adatait.

Paraméterként kapja a küldő objektumot, illetve az esemény argumentumait.

```
private void button1_Click(object sender, EventArgs e)
```

Az Access Point adatai gomb megnyomására hívódik meg. A `getAPposition` webszerviztől elkéri az aktuálisan kijelölt Access Point adatait.

Paraméterként kapja a küldő objektumot, illetve az esemény argumentumait.

```
private void APlistMenu_Click(object sender, EventArgs e)
```

Az Access Point-ok listája menüpontot kiválasztva hívódik meg. A `getAllAPdata` webszerviztől elkéri az összes adatbázisban lévő Access Point adatát. A megjelenítés nincs implementálva.

Paraméterként kapja a küldő objektumot, illetve az esemény argumentumait.

```
private void APdeleteMenu_Click(object sender, EventArgs e)
```

Az Access Point törlése menüpontot kiválasztva hívódik meg. Törli a `ListView` elemben kijelölt Access Point-ot a `deleteAP` webszerviz segítségével.

Paraméterként kapja a küldő objektumot, illetve az esemény argumentumait.

```
private void AddPOImenu_Click(object sender, EventArgs e)
```

A POI hozzáadása menüpontot kiválasztva hívódik meg. Példányosítja az `frmAddPOI` form-ot.

Paraméterként kapja a küldő objektumot, illetve az esemény argumentumait.

```
private void POIlistMenu_Click(object sender, EventArgs e)
```

A POI-k listája menüpontot kiválasztva hívódik meg. Ez a funkció nincs implementálva.

Paraméterként kapja a küldő objektumot, illetve az esemény argumentumait.

5.3.2. *frmAddAP* osztály

Ez az osztály tartalmazza azt a képernyőt, amelyen új Access Point-okat adhatunk hozzá az adatbázishoz.

Az osztály adattagjai:

```
private wrfWPlaceService.Service ws = new wrfWPlaceService.Service();  
private string APname;  
private string BSSID;
```

Az osztály metódusai:

```
public frmAddAP(string[] data)
```

A form konstruktora, inicializálja a vizuális elemeket, valamint beállítja a belső változókat.

Paraméterként kapja a felveendő Access Point nevét és BSSID-ját.

```
private void Form2_Activated(object sender, EventArgs e)
```

A form aktiválásakor hívódik meg, beállítja a konstruktor által kapott adatokat a megfelelő `TextField` `Text` tulajdonságába.

Paraméterként kapja a küldő objektumot, illetve az esemény argumentumait.

```
private void btnOK_Click(object sender, EventArgs e)
```

Az OK gomb megnyomásakor hívódik meg, a `storeAP` webszerviz segítségével eltárolja az adatbázisba az Access Point adatait, majd felszámolja a form-ot.

Paraméterként kapja a küldő objektumot, illetve az esemény argumentumait.

```
private void btnCancel_Click(object sender, EventArgs e)
```

Az Mégse gomb megnyomásakor hívódik meg, felszámolja form-ot.

Paraméterként kapja a küldő objektumot, illetve az esemény argumentumait.

5.3.3. *frmAddPOI* osztály

Ez az osztály tartalmazza azt a képernyőt, amelyen új POI-kat adhatunk hozzá az adatbázishoz.

Az osztály adattagjai:

```
private wrfWPlaceService.Service ws = new wrfWPlaceService.Service();
private double POIx;
private double POIy;
```

Az osztály metódusai:

```
public frmAddPOI(double poix, double poiY)
```

A form konstruktora, inicializálja a vizuális elemeket, valamint beállítja a belső változókat.

Paraméterként kapja a felveendő POI koordinátáit.

```
private void frmAddPOI_Activated(object sender, EventArgs e)
```

A form aktiválásakor hívódik meg, beállítja a konstruktor által kapott adatokat a megfelelő `TextField` `Text` tulajdonságába.

Paraméterként kapja a küldő objektumot, illetve az esemény argumentumait.

```
private void btnOK_Click(object sender, EventArgs e)
```

Az OK gomb megnyomásakor hívódik meg, a `storePOI` webszerviz segítségével eltárolja az adatbázisba a POI adatait, majd felszámolja a form-ot.

Paraméterként kapja a küldő objektumot, illetve az esemény argumentumait.

```
private void btnCancel_Click(object sender, EventArgs e)
```

A Mégse gomb megnyomásakor hívódik meg, felszámolja form-ot.

Paraméterként kapja a küldő objektumot, illetve az esemény argumentumait.

5.3.4. *frmAboutYourLocation* osztály

Ez az osztály tartalmazza azt a képernyőt, amelyen új POI-kat adhatunk hozzá az adatbázishoz.

Az osztály adattagjai:

```
private wrfWPlaceService.Service ws = new wrfWPlaceService.Service();
private double X;
private double Y;
private static double OneMeterCoords = 0.0000136;
private string[] POI1 = new string[3];
private string[] POI2 = new string[3];
private string[] POI3 = new string[3];
```

Az osztály metódusai:

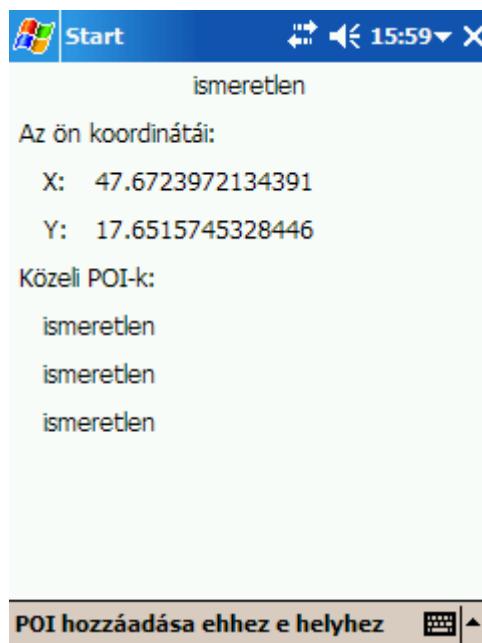
```
public frmAboutYourLocation(double [] Coordinates, string Name)
```

A form konstruktora, inicializálja a vizuális elemeket, valamint beállítja a belső változókat. A FindPOI webszerviz többszöri meghívásával megkeresi a távolabbi POI-kat.

Paraméterként kapja a felveendő POI koordinátáit.

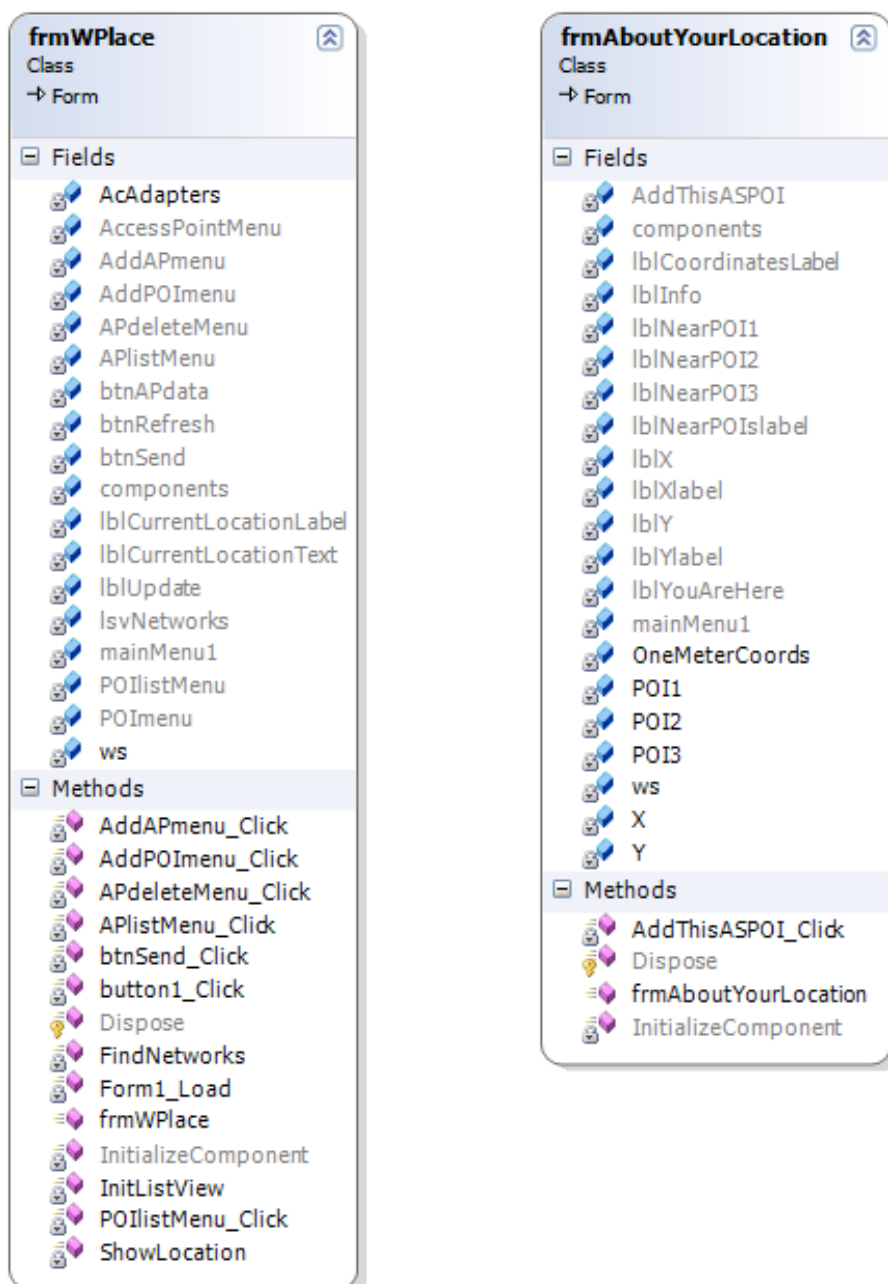
```
private void AddThisASPOI_Click(object sender, EventArgs e)
```

A POI hozzáadása ehhez a helyhez menüpont kiválasztásával hívódik meg. Példányosítja az *frmAddPOI* form-ot, átadja az aktuális hely koordinátáit.



11. ábra : az *frmAboutYourLocation* form

5.3.5. Osztálydiagramok



frmAddAP
Class
→ Form

Fields

- APname
- BSSID
- btnCancel
- btnOK
- components
- lblAPname
- lblBSSID
- lblCoordinates
- lblDesc
- lblX
- lblY
- mainMenu1
- txtapName
- txtAPx
- txtAPy
- txtBSSID
- txtDesc
- ws

Methods

- btnCancel_Click
- btnOK_Click
- Dispose
- Form2_Activated
- frmAddAP
- InitializeComponent

frmAddPOI
Class
→ Form

Fields

- btnCancel
- btnOK
- components
- lblCoordinates
- lblDesc
- lblPOIname
- lblURL
- lblX
- lblY
- mainMenu1
- POIx
- POIy
- txtDesc
- txtPOIname
- txtURL
- txtX
- txtY
- ws

Methods

- btnCancel_Click
- btnOK_Click
- Dispose
- frmAddPOI
- frmAddPOI_Activated
- InitializeComponent

Program
Static Class

Methods

- Main

5.4. Szerver oldal

5.4.1. *AP* osztály

Ez az osztály egy Access Point-ot reprezentál.

Az osztály adattagjai:

```
public string name;  
public string bssid;  
public double x;  
public double y;  
public string description;
```

Az osztály metódusai:

```
public AP(string _name, string _bssid, double _x, double _y,  
           string _description)
```

Az osztály konstruktora, beállítja az osztály attribútumait.

5.4.2. *APlist* osztály

Ez az osztály maximum 256 AP adatait képes eltárolni.

Az osztály adattagjai:

```
private int count;  
private AP[] elements = new AP[256];
```

Az osztály metódusai:

```
APlist()
```

Létrehozza az osztályt és beállítja a `count` elemszámlálót nullára.

```
public void Add(AP ap)
```

A paraméterben kapott AP-t hozzáadja a listához.

```
public void Clear()
```

Kitörli a lista összes elemét.

```
public AP GetElementAt(int i)
```

Visszaadja a lista *i*. elemét.

5.4.3. Szerviz réteg

```
[WebMethod]
```

```
public double[] WhereAmI(string[] apBSSIDs, double[] apSSs)
```

Ez a szerviz a paraméterben kapott adatok alapján meghívja a `triangulate` privát metódust, amely a `getAPposition` által az adatbázisból kinyert Access Point koordináták alapján kiszámolja a kliens koordinátáit..

Paraméterként kapja az első három látható Access Point BSSID-ját és a hozzájuk kapcsolódó jelerősséget.

```
private double dBm2meter(double dBm)
```

A kapott jelerősséget közelítő algoritmus segítségével átszámítja távolsággá.

```
private double[] triangulate(double[, ] Coordinates)
```

A kapott koordináták és jelerősségek alapján kiszámítja a kliens koordinátáit, aszerint, hogy egy, kettő, vagy három Access Point-ot lát.

```
private string toDot(double number)
```

A kapott dupla pontosságú számot tizedesponttal elválasztott string-ként adja vissza.

5.4.4. Adatbázis réteg

```
[WebMethod]
```

```
public string storeAP(string APname, string APBSSID, double APX, double APY, string APdesc)
```

Ez a szerviz a paraméterben kapott adatok alapján létrehoz egy bejegyzést az adatbázis APs táblájában.

```
[WebMethod]
```

```
public string storePOI(string POIname, double POIX, double POIY, string POIURL, string POIdesc)
```

Ez a szerviz a paraméterben kapott adatok alapján létrehoz egy bejegyzést az adatbázis POIs táblájában.

```
[WebMethod]
```

```
public string deleteAP(string APBSSID)
```

Ez a szerviz a paraméterben kapott BSSID-val rendelkező bejegyzést törli az adatbázis APs táblájából.

```
[WebMethod]
```

```
public string deletePOI(int id)
```

Ez a szerviz a paraméterben kapott POIid-val rendelkező bejegyzést törli az adatbázis POIs táblájából.

```
[WebMethod]
```

```
public double[] getAPposition(string BSSID)
```

Ez a szerviz a paraméterben kapott BSSID-val rendelkező bejegyzés koordinátáit adja vissza az adatbázis APs táblájából.

```
[WebMethod]
```

```
public string[] getAPdata(string BSSID)
```

Ez a szerviz a paraméterben kapott BSSID-val rendelkező bejegyzés egyéb adatait adja vissza az adatbázis APs táblájából.

```
[WebMethod]
```

```
public string[] FindPOI(double[] Coordinates, double Accuracy)
```

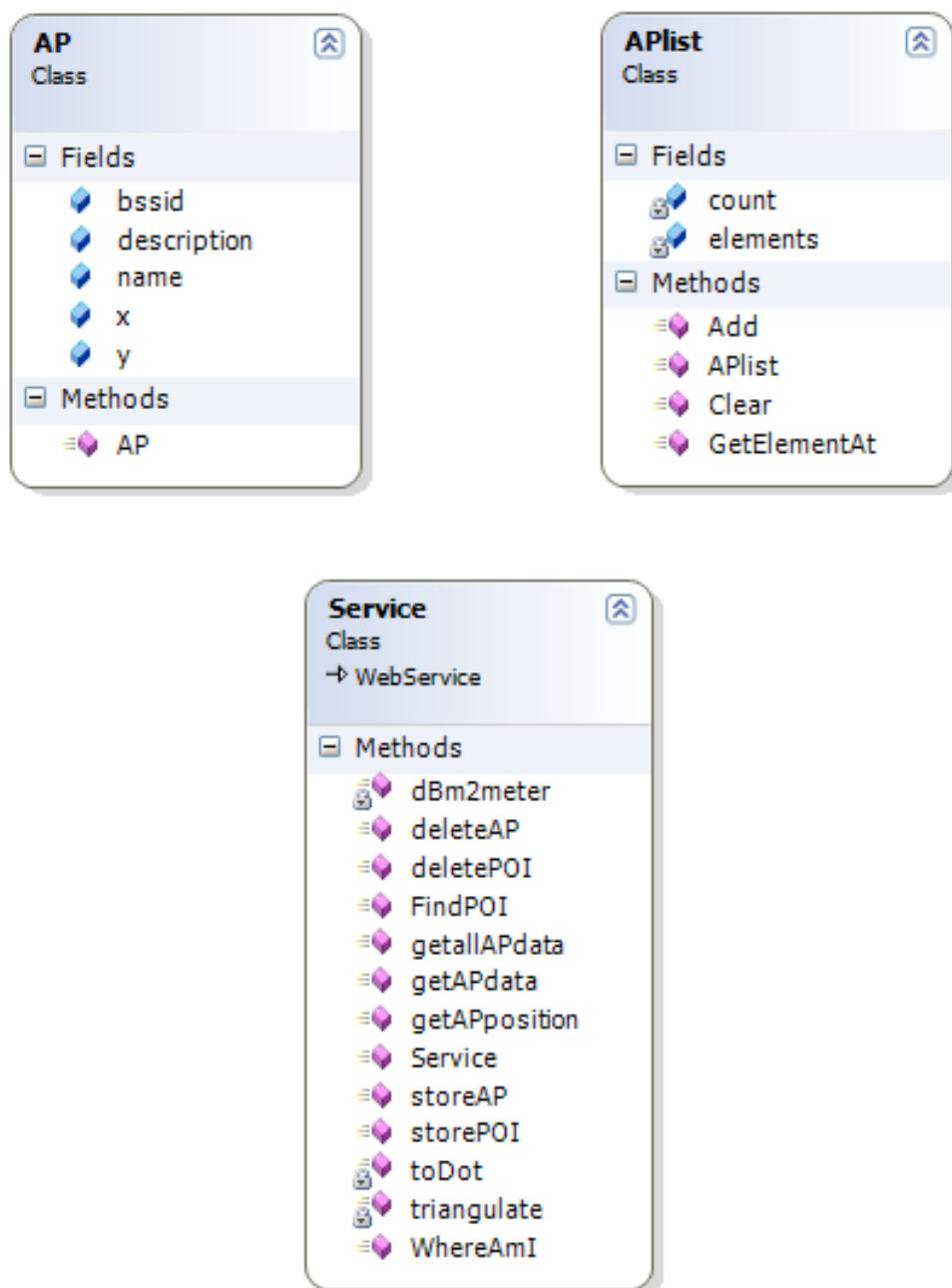
Ez a szerviz a paraméterben kapott koordinátákhoz az ugyancsak paraméterben kapott távolságon belül lévő első POI adatait adja vissza.

```
[WebMethod]
```

```
public APlist getAllAPdata()
```

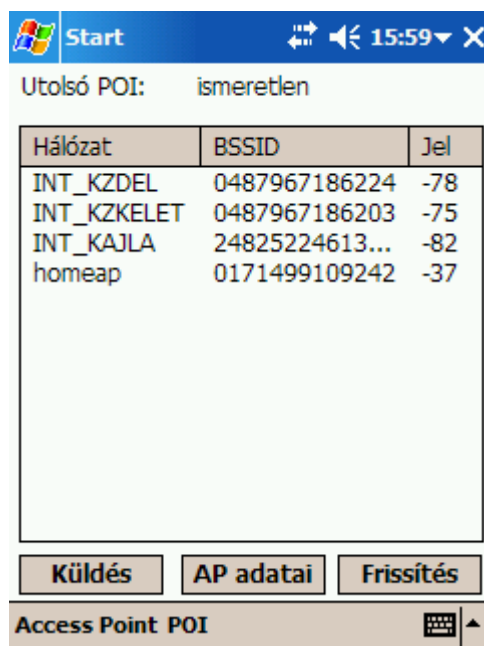
Ez a szerviz a adatbázis APs táblájában lévő összes bejegyzést adja vissza.

5.4.5. Osztálydiagramok



5.5. Felhasználói segédlet

A program kezelése nem igényel különösebb hozzáértést. Az indulási képernyőn láthatjuk az éppen észlelt vezeték nélküli hálózatokat, valamint azok jelerős-ségét. Ezt a *Frissítés* gombbal tudjuk aktualizálni, illetve a *Küldés* gombbal elküldeni a szervernek.



12. ábra : A Wplace indulási képernyője

Az *AP adatai* gombbal a kijelölt Access Point-okról kérhetünk információt, amennyiben az az adatbázisban szerepel.

Az adatok elküldése után egy információs képernyő jelenik meg, amely közli az aktuális koordinátáinkat, illetve, hogy milyen POI-k vannak a közelben. Itt lehetőségünk nyílik az aktuális pozíciót a POI-k közé felvenni.

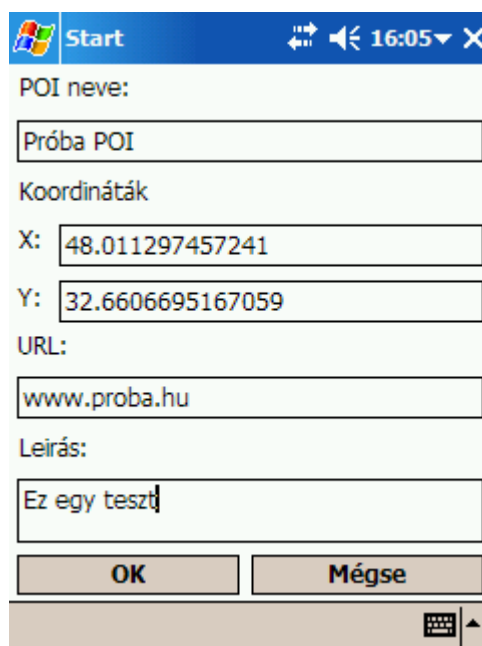
Az indulási képernyőről, a menüből válogatva lehetőségünk van még Access Point hozzáadására illetve törlésére is. Hozzáadásnál, amennyiben van kijelölve Access Point, a neve, illetve a BSSID-ja automatikusan átmásolódik a hozzáadás lapra.

Törlésnél a kijelölt Access Point törlődik az adatbázisból.



13. ábra : Access Point felvétele

A 13. ábrán egy Access Point hozzáadása látható, a név és a BSSID még ki van töltve.



14. ábra : POI felvétele

A 14. ábrán egy POI hozzáadása látható, lehetőségünk van későbbi felhasználás céljából egy URL-t hozzáadni a bejegyzéshez.

5.6. Továbbfejlesztési lehetőségek

E program csak demonstrálni hivatott a WiFi alapján történő helymeghatározás működését. A felhasznált OpenNetCF-ből adódóan pontossága elmarad az elérhetőtől, mivel a WiFi hardver vételének jel / zaj aránya alapján lehetne pontosabb számításokat végezni, ám az OpenNetCF segítségével csak a korábban említett RSSI-t illetve, az ahhoz tartozó dBm-et lehet kiolvasni.

A program képes valós földrajzi koordinátákkal dolgozni, ellenben nincsen megjelölve, a kelet/nyugat a hosszúságnál, valamint az észak/dél a szélességnél.

A Windows Mobile lehetőségeit kihasználva, akár egy háromszögletű térképet is ki lehetne rajzolni a kijelzőre, így vizualizálva a látott Access Point-ok helyét, illetve azokhoz képest a sajátunkat.

Mivel a kliens szabványos webszerviz hívásokat használ, így akár lehetséges a szerver oldal teljes cseréje. Elképzelhető például egy java-ban írt szerviz, amely egy mySQL adatbázissal kommunikál. Ezzel a szerver oldal máris platformfüggetlen lett.

A kliens oldalon szóba jöhet még a Symbian operációs rendszer, amely hasonlóan jó programozhatósággal, ám nem ilyen komoly IDE támogatottsággal rendelkezik. Másik nagy hátránya, hogy az egyes verziók általában nem kompatibilisek egymással.

A szoftver kliens része némi átalakítás után Win32 architektúrán is futtatható, így egy laptopot is használhatunk helymeghatározásra.

6. ELÉRHETŐ KERESKEDELMI TERMÉKEK

6.1. Ekahau

6.1.1. T301 - WiFi Location Tag

A T301-es tag (jel, jelölés, jelölő) segítségével embereket vagy szállítmányokat lehet nyomon követni beltéren és kültéren egyaránt.

Ez egy harmadik generációs aktív tag, amely beépített energiaforrása révén jeleket sugároz, amelyek aztán az Ekahau RTLS (Real Time Location System, Valós Idejű Helymeghatározó rendszer) segítségével, 802.11b/g infrastruktúrán keresztül pontosan nyomon követhetők. Ez különösen fontos olyan helyeken, mint egy kórház, vagy gyár, ahol emberek, vagy nagy értékű csomagok útját kell követni, ezáltal növelve a biztonságot, vagy a gyártósor hatékonyságát.

A T301 tulajdonságai:

- Alacsony energiafelhasználás, akár 5 éves akkumulátor élettartam.
- Intelligens mozgásérzékelő, csak akkor sugároz, amikor mozgásban van.
- Nem szükséges új infrastruktúra, a meglévő WiFi hálózaton keresztül üzemel.
- Távolról konfigurálható, monitorozható, frissíthető WiFi-n keresztül.
- 2 programozható gomb, akár riasztás küldésére is alkalmas.
- Optikai elmozdítás védelem.
- Időjárásálló.

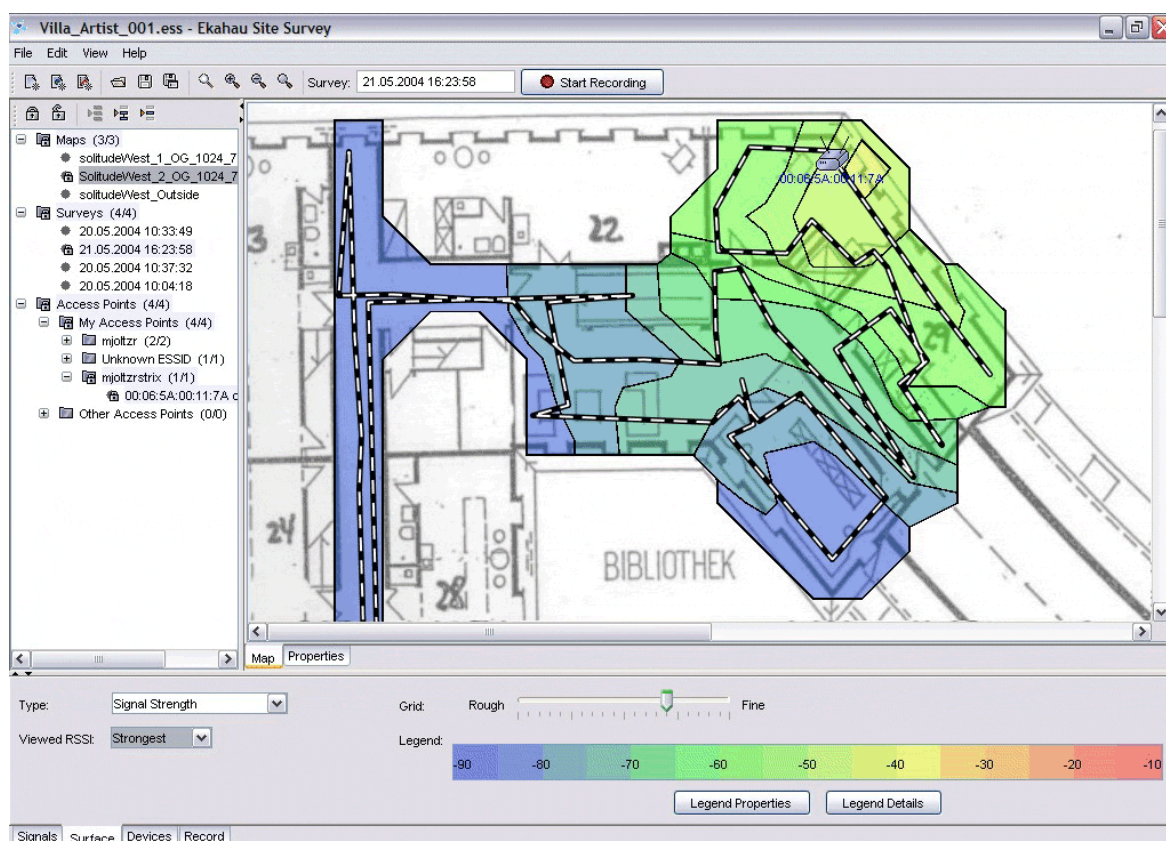
6.1.2. EPE – Ekahau Positioning Engine

Az EPE egy szoftveres valós idejű helymeghatározó. Amellett, hogy kiegészíti a hardveres követést, teljesen jövőbiztos. Az EPE képes megmondani a WiFi tagok, és bármilyen mobil egység pozícióját, 1 méter átlagos pontosságot garantálva.

6.1.3. ESS – Ekahau Site Survey

Az ESS segítségével lehetőség nyílik WiFi hálózatok lefedettség tervezésére, biztosítva ezzel az egységek számára az optimális hálózat elérést, illetve a helymeghatározás pontosságát.

A programba egyszerűen tölthetők be térképek, majd ezekre Access Point-okat helyezve különböző tesztek futtathatók, meghatározva a maximális átviteli sebességet, a zaj-jel arányt, valamint a hálózati átfedéseket.



15. ábra : Ekahau Site Survey 2.0

A 11. ábrán egy térkép látható, egy Access Point-tal, valamint az az útvonal, amelyet a feltérképező személy járt be. A különböző színek azt jelzik, hogy az egyes helyeken milyen a vételi szint.

6.2. Skyhook

A Skyhook nem kicsiben gondolkodik, hozzáférést és szoftvereket kínál az általa az Egyesült Államokban feltérképezett Access Point-ok listájához, amelyek segítségével aztán helyzet információt tud adni az WiFi egységnek, vagy a WiFi egységről.

Az általa felépített adatbázis segítségével a kormány által támogatott E911 (Enhanced 911) programnak is komoly segítségére lehet. Az E911-et használó segélyhívó készülék GSM, vagy akár WLAN-on VoIP-n keresztül kommunikál, így lehetősége van közölni a központtal, hogy mi a legközelebbi Access Point BSSID-ja, ezután a hely könnyen kikereshető a Skyhook adatbázisából és a segítség jó helyre érkezik.

A fejlesztők fontosnak tartják azt is, hogy míg a GPS esetén van bemelegedési idő, ami perceket is igénybe vehet, WiFi esetén BSSID-k beolvasása akár egyetlen másodperc alatt is megtörténhet.

6.3. Navizon

A Navizon kétféle terméket forgalmaz. Az egyik a Navizon GPS ingyenesen elérhető bárki számára. E szoftver működéséhez mindenképpen GPS szükséges, ennek segítségével be tudja mérni a látható WiFi hálózatok illetve mobiltelefon cellák pontos koordinátáit, amelyeket majd elküld egy központi szervernek. Bizonyos számú cella illetve WiFi után a cég 20 dolláros jutalmat fizet. Ezzel a megoldással viszonylag egyszerűen széles körből tudnak információkat gyűjteni.

A másik termék a Navizon Premium, amely 20 dolláron áron vásárolható meg. Ennek segítségével a Navizon GPS-t használók által összegyűjtött információkat (WLAN, illetve GSM) használhatjuk fel. A szoftver soros porton GPS-t emulál, így tetszés szerinti GPS-re épülő navigációs megoldást használhatunk segítségével.

7. KISLEXIKON

- **GPS:** Global Positioning System. Globális Helymeghatározó Rendszer. Műholdak alapján történő helymeghatározást lehetővé tevő rendszer.
- **WiFi:** Wireless-Fidelity. Vezeték nélküli-”valószerűség”.Az IEEE 802.11-es szabványában megfogalmazott specifikációk alapján meghatározza a WLAN technológia működését.
- **WLAN:** Wireless Local Area Network. Vezeték nélküli Helyi Hálózat. Segítségével vezeték nélkül kapcsolhatunk össze számítógépeket.
- **DGPS:** Differential GPS. Differenciált GPS. GPS hibajavító eljárás, pontossága: 3-5 m.
- **WAAS:** Wide Area Augmentation System. Széles körű Kiterjesztő Rendszer. GPS hibajavító eljárás, pontossága: 1-3 m.
- **LAAS:** Local Area Augmentation System. Helyi Kiterjesztő Rendszer. GPS hibajavító eljárás, pontossága: 1-0,01m.
- **Open Source:** Nyílt forráskóddal rendelkező szoftver. Általában felhasználható és megváltoztatható a forrás megjelölésével.
- **VHF:** (Very High Frequency) 30 és 300 Mhz közötti frekvencia. Például földfelszíni analóg TV sugárzásra használják.
- **CSMA/CD:** (Carrier Sense Multiple Access / Collision Detection) Ütközés figyelési protokoll, az IEEE 802.3 írja le.
- **ALOHA:** Az első kvázi WLAN Hawaii szigetén.
- **Access Point:** A vezeték nélküli hálózat egy hozzáférési pontja.
- **DSL:** (Digital Subscriber Line) Manapság elterjedt széles sávú digitális internet-hozzáférés.
- **(W)NIC:** ((Wireless) Network Interface Card) Ezekkel a kártyákkal lehet hozzáférni egy hálózathoz.
- **(B)SSID:** Lásd 10. oldal.
- **WEP, WPA, WPA2:** Lásd 11. oldal.

- **VPN:** (Virtual Private Network) Virtuális hálózat, melyen az adatforgalom titkosított.
- **MAC:** (Medium Access Control) A hálózat adatkapcsolati rétegében helyezkedik el. Minden a hálózatra kapcsolódó gépnek saját MAC címe van.
- **Teodolit:** Vízszintes és magassági szögek mérésére alkalmas műszer.
- **RSSI:** (Received Signal Strength Indication) Hardverhez kapcsolható, az általa vett jel erősségét fejezi ki.
- **Herecast:** Open Source, C++ nyelven írt egyszerű WLAN-os helymeghatározó szoftver.
- **NDIS:** (Network Driver Interface Specification) Az ebben leírtak alapján lehet elérni egy hálózati interfészt.
- **Wplace:** Saját fejlesztésű, a WiFi-s helymeghatározást bemutató szoftverrendszer.
- **GPRS:** (General Packet Radio Services) Csomagkapcsolt adatátvitel GSM hálózatokon. Körülbelül 80 kbit/s letöltési sebességgel.
- **EDGE (EGPRS):** Enhanced GPRS. Fejlett GPRS, körülbelül 236,8 kbit/s letöltési sebességgel.
- **HSDPA:** (High-Speed Downlink Packet Access) körülbelül 3,6 Mbit/s letöltési sebesség.
- **SDIO:** Secure Digital kártya méretű fejlett eszközök használatát lehetővé tevő protokoll.
- **SDK:** (Software Development Kit) Szoftverfejlesztő készlet.
- **IDE:** (Integrated Development Environment) Integrált fejlesztőkörnyezet.
- **POI:** (Point Of Interest) Érdekes hely.
- **OpenNetCF:** Ingyenes Compact Framework Visual Studio .NET-hez.
- **dBm:** Itt: a vevő érzékenységet jellemzi, negatív érték. Decibel-miliwatt.
- **VoIP:** (Voice Over IP) IP protokollon történő hangátvitel.

8. IRODALOMJEGYZÉK

- [1] wikipedia.org
- [2] geography.about.com/od/geographictechnology/a/gps.htm
- [3] www.gpsinformation.org/dale/dgps.htm
- [4] en.wikipedia.org/wiki/Wireless_LAN
- [5] Pallasz Nagylexikon
- [6] local.wasp.uwa.edu.au/~pbourke/geometry/2circle
- [7] ekahau.com
- [8] www.skyhookwireless.com
- [9] www.navizon.com