

OTDK-dolgozat

Kádár Iván

2010

**SZABAD SZOFTVEREKKEL KÉSZÍTETT, WEBES, SZABAD SZAVAS
TÉRKÉPKEZELŐ ÉS KERESŐ RENDSZER MEGVALÓSÍTÁSA**

**THE REALIZATION OF THE UTILIZATION OF OPEN SOURCE
SOFTWARE IN DESIGNING WEB BASED FREE TEXT MAP SEARCH
AND MAP MANIPULATION SYSTEMS**

Konzulensek:

Dr. Pődör Andrea

Rab András

Kézirat lezárva: 2010. november

Tartalomjegyzék

1 Bevezetés.....	3
2 A feladat bemutatása.....	4
3 A felhasznált nyílt forráskódú alkalmazások bemutatása.....	5
3.1 Felhasználási feltételek, GPL.....	5
3.2 Apache HTTP Server.....	5
3.3 PHP.....	6
3.4 PostgreSQL.....	6
3.5 MapServer.....	6
3.6 OpenLayers.....	7
3.7 XHTML, CSS.....	7
4 A feladat megvalósítása.....	8
4.1 Az adatbázis szerkezete.....	8
4.2 Az adminisztrátor felület tervezése, megvalósítása.....	9
4.2.1 A feltöltési folyamathoz kapcsolódó függvények.....	9
4.2.2 Rekordok kilistázása, törlése.....	12
4.2.3 Rekordok módosítása.....	13
4.2.4 A MapServer fájl és a felhasználói felület generálása.....	14
4.3 A felhasználói felület tervezése, megvalósítása.....	16
4.3.1 Térkép megjelenítése.....	16
4.3.2 Keresés az adatbázisban.....	17
5 Eredmények, a rendszer használatának bemutatása.....	19
5.1 Adminisztrációs felület.....	19
5.2 Felhasználói felület.....	21
6 A rendszer továbbfejlesztésének lehetőségei.....	25
6.1 Támogatott fájlformátumok bővítése.....	25
6.2 Felhasználói felület fejlesztése.....	25
7 Összegzés, tapasztalatok.....	26
8 Irodalomjegyzék.....	27

9 Mellékletek.....	28
9.1 A program térinformatikai szempontból fontos kódjai.....	28
9.1.1 A keresés megvalósítása (keres.inc).....	28
9.1.2 AutoCAD DXF fájlok megjelenítése (dxf_viewer.map).....	29
9.1.3 A térképek középpontjának és nevének megjelenítése (points.map).....	32
9.2 A mellékelt CD tartalma.....	34
9.2.1 A work könyvtár tartalma.....	34
9.2.2 A www könyvtár tartalma.....	35

1 Bevezetés

Dolgozatom célja egy térinformatikai rendszer megvalósítása volt, mely előre kartografált térképeket jelenít meg, és tesz kereshetővé egy webes felületen keresztül. Fontos szempont volt, hogy a munka folyamán csak nyílt-forráskódú alkalmazásokat használjak fel, így bemutatva, hogy van ingyenes, platformfüggetlen, mindenki számára elérhető alternatívája a tanulmányaink során megismert rendszereknek.

TDK dolgozatomban röviden bemutatom a felhasznált nyílt forráskódú alkalmazásokat, majd ismeretem a megvalósítás menetét a tervezéstől a programozás befejezéséig. Ezt követően bemutatom az elkészült alkalmazást és annak használatát. Végül ismeretem a fejlesztés közben szerzett tapasztalataimat.

2 A feladat bemutatása

A rendszer megvalósítására a felkérést a külsős konzulensemtől, Rab Andrástól kaptam. Ő, mint a Theodolight Kft. egyik tulajdonosa, és munkatársa kért fel, hogy készítsek egy web alapú térinformatikai rendszert, mellyel a honlapjukon be tudják mutatni az érdeklődőknek a cég által végzett földmérési munkákat. A látogatók számára elérhető megjelenítőn túl el kellett készítenem a háttérrendszert is, ahol el lehet végezni a térképek és a leíró adatok feltöltését, módosítását, törlését.

A feladat fő részét a PHP programozás tette ki. Elkészítettem az adminisztrációs felületet, mely az adatbázis-műveleteket végzi, tehát lehetővé teszi az AutoCAD DXF térkép-fájlok, és a hozzájuk kapcsolódó leíró adatok feltöltését, módosítását, törlését. Ezen kívül legenerálja a megjelenítő oldalt és komponenseit is, amit a honlap látogatói érnek el. A megjelenítő oldal a MapServer térkép-megjelenítő és az OpenLayers JavaScript függvénykönyvtár segítségével megjeleníti a térképeket, és kapcsolódva az adatbázishoz, kereshetővé is teszi azokat.

3 A felhasznált nyílt forráskódú alkalmazások bemutatása

3.1 Felhasználási feltételek, GPL

A GPL betűszó a GNU General Public License, avagy magyarul GNU Általános Nyilvános Licenc rövidítése. Megalkotója a Free Software Foundation, mely eredetileg a GNU Projekthez alkotta meg ezt a licencet. Legelterjedtebb változata az 1991-ben elkészült GPLv2.[1]

A licenc lényege, hogy a készített mű, legyen akár program, akár más alkotás, szabadon terjeszthető, módosítható, és akár pénzért is terjeszthető, azonban módosítás esetén ismét csak GPL alatt adható ki, terjeszthető az eredmény, tehát az eredményt ismét bárki felhasználhatja. Ebből következően egy nyereségorientált cég ingyen hozzájuthat a mások által fejlesztett programhoz, azonban ha saját fejlesztéseket végez rajta, és úgy értékesíti, akkor kötelezően neki is elérhetővé kell tenni a módosított forráskódot. [2]

Ez a megoldás egy mindenki számára nyertes helyzetet teremt: a cégeknek nem kell a semmiből újra elkészíteniük olyan alkalmazásokat, amik már léteznek, és a megvásárlásukra se kell költeniük, azonban az így megspórolt összeg egy részét az alkalmazás továbbfejlesztésére költhetik, ezzel hozzájárulva a további fejlődéshez, illetve remélheti, hogy az általa megvalósított funkciót a későbbiekben mások továbbfejlesztik, így ismét haszna származhat belőle.

A GPL rengeteg párhuzamos munkától, vagyis fölös költségtől és időpazarlástól kíméli meg a gazdaság szereplőit, ez által segítve a hatékonyabb működést.

Az általam felhasznált programok közül nem mindegyiket GPL licenccel adták ki, azonban mindegyik nyílt forráskódú, szabadon felhasználható alkalmazás.

3.2 Apache HTTP Server

Az Apache egy webkiszolgáló. Célja, hogy a weboldalakat elérhetővé tegye a látogatók számára, legyen szó akár a szerveren található statikus fájlról (html, kép, stb.), vagy egy másik program által előállított dinamikus tartalomról (PHP, Perl, Python, MapServer,

adatbáziszszerverek, stb.)

Az Apache nem GPL alatt kerül kiadásra, hanem a saját licence alatt, ami megengedi a zárt forráskódú terjesztést is. [3]

3.3 PHP

A PHP egy script nyelv, tehát a benne írt programokat nem szükséges külön lefordítani futtatás előtt, hanem futási időben fordul le. Tervezésének fő célja dinamikus weboldalak készítése volt, ennek megfelelően gyorsan és hatékonyan lehet vele html kódokat generálni, űrlapokat feldolgozni, és az elterjedt adatbáziszszerverekhez is biztosít elérést. [4]

A könnyű, és univerzális használhatóságának hátránya, hogy komoly biztonsági kockázatot jelent egy rosszul megírt program. Mivel a felhasználóval is kommunikál, és a szerver erőforrásaihoz is hozzáférése van, ezért nagyobb programok esetén alapos végiggondolást igényel, hogy hogyan lehet feldolgozni a kéréseket úgy, hogy közben ne tudjanak illetéktelenek hozzáférni a szerveren tárolt adatokhoz.

3.4 PostgreSQL

A PostgreSQL, avagy régi nevén Postgres egy relációs adatbázis-kezelő rendszer. Bár nem olyan elterjedt mint a MySQL, azonban amellett, hogy annak legtöbb funkcióját tartalmazza, további adattípusokat is támogat, mint például a geometriai adattípusok (pont, vonal, poligon vetülettel együtt), vagy a weben használt speciális adatok (MAC-cím, IPv4, IPv6 címek).

További előnye, hogy a geometriát képes indexelni a gyorsabb elérés érdekében, amit én ugyan nem használtam ki, azonban egy nagy térinformatikai adatbázis esetén elengedhetetlen funkció. [5]

3.5 MapServer

A MapServer a rendszer legfontosabb térinformatikai alkalmazása. Ez a program felel a térképek megjelenítéséért. Megadva neki a bemeneti térképállományokat, és definiálva a kimeneti formátumot, létrehozza a raszter képet, amit aztán az OpenLayers segítségével

meg lehet jeleníteni a webböngészőben. Képes kezelni az elterjedt raszter és vektorábrákat, illetve adatbázisból is képes megjeleníteni adatokat, mint azt majd az alkalmazásomban is bemutatom. [6]

3.6 OpenLayers

Az OpenLayers egy JavaScript függvénykönyvtár. Segítségével felhasználói oldalon történik meg a térkép lekérése, megjelenítése a böngészőben.

Hátránya, hogy feltételezi a JavaScript motor meglétét, és megfelelő működését a felhasználó böngészőjében. Ezáltal a rendszer leggyengébb pontjának tekinthető. Mivel a felhasználók különböző cégek különböző böngészőit használják, nem lehetünk benne biztosak, hogy a használt JavaScript motor megbirkózik-e a kóddal.

Főleg az Internet Explorer különböző verzióira jellemző, hogy figyelmen kívül hagyva a szabványosított megoldásokat, sokszor hibásan értelmezik a feldolgozandó kódot. Ebből adódóan a térkép helyett csak hibakódokat, vagy még azokat se jelenítenek meg.

3.7 XHTML, CSS

A HTML a HyperText Markup Language (hiperszöveges jelölőnyelv) rövidítése. Az XHTML ennek egy szigorúbb, XML (Extensible Markup Language) alapú variánsa.

A HTML egy leírónyelv, mely megadja a böngészőnek a weboldal logikai felépítését. Bár lehetőség van benne a megjelenés definiálására is, ám ezt célszerűbb a CSS-re (Cascading Style Sheets) bízni, mely szintén egy leíró nyelv, és célja a grafikus megjelenítés testreszabása.

Bár ezen leírások értelmezése is a felhasználó oldalán található böngésző feladata, az itt vétett feldolgozási hibák jellemzően csak megjelenítési problémákat eredményeznek, szemben a JavaScripttel, ami az oldal működését alapvetően befolyásolja.

4 A feladat megvalósítása

4.1 Az adatbázis szerkezete

A rendszer adatbázisa egyetlen táblából áll, mely a térképek leíróadatait tartalmazza. Maguk a térképek egy-egy előre kartografált AutoCAD dxf állományban kerülnek lementésre a szerver egy kijelölt könyvtárába.

Az adattábla tartalma

- **ID**: Automatikusan sorszámozott record azonosító.
- **fajl**: Szövegmező, mely a térkép-fájl nevét tartalmazza.
- **nev**: Szövegmező, mely a térkép megjelenített nevét tartalmazza.
- **date**: dátummező a felmérés időpontjának tárolására
- **munka**: a felméréshez szükséges munkamennyiség 1-5-ig tartó skálán pontozva
- **sikrajz**: igaz/hamis; megadja, hogy van-e síkrajzi tartalma a felmérésnek.
- **foldhiv**: igaz/hamis; kapcsolódott-e földhivatali munka a felméréshez?
- **kulcs_szavak**: Szövegmező; a szabadszavas kereső a név mellett még ezt a mezőt használja fel kereséskor.
- **megrendelo**: Szövegmező, a megrendelő cég, vagy személy neve.
- **megrend_tip**: Szövegmező, a megrendelő típusát adja meg. Például magánszemély, cég, állami megrendelés.
- **EOVY**: Valós szám, a térkép középpontjának EOY Y koordinátája.
- **EOVX**: Valós szám, a térkép középpontjának EOY X koordinátája.
- **WGS84lon**: Valós szám, a térkép középpontjának WGS84-beli hosszúsága.
- **WGS84lat**: Valós szám, a térkép középpontjának WGS84-beli szélessége.
- **wgs84**: Geometriai mező, a térkép középpontjának WGS84 koordinátája.

Új record felvételekor kötelező kitölteni a *nev*, az *EOVY*, az *EOVX* mezőt, és síkrajz esetén meg kell adni a térkép-fájlt is.

A rendszer a megadott EOY koordinátákat a `cs2cs` programmal – mely a `PROJ.4` – `Cartographic Projections Library` részét képezi – konvertálja WGS84 koordinátákká. A koordináták a fent ismertetett módon redundánsan tárolódnak. Erre azért

van szükség, mert különben minden egyes oldallekérdezéskor, futási időben kéne elvégezni a konverziót, ami jelentősen lassítaná a látogatók kiszolgálását.

4.2 Az adminisztrátor felület tervezése, megvalósítása

Az adminisztrációs felületnek az alapvető adatbázis-kezelési funkciókat kell ellátnia. Így meg kellett valósítanom az adatok feltöltésének, módosításának, és eltávolításának lehetőségét.

Az oldal az `admin.php` címen érhető el az arra jogosultaknak. A jogosultság ellenőrzését az Apache HTTP Server végzi a könyvtárban található `.htaccess` fájl alapján.

A programkód jobb átláthatósága érdekében a különböző funkciókat külön fájlba tároltam el, melyeket az `admin.php` tölt be.

Az első funkció az *Új térkép felvétele az adatbázisba*. Itt a tömör kinézet érdekében az oldal adminisztrátora csak egy gombot talál, melyre rányomva megjelenik a térkép adatait kérő űrlap. A feltöltéshez szükséges minden programkód az `upload.inc` fájlban található meg.

4.2.1 A feltöltési folyamathoz kapcsolódó függvények

4.2.1.1 new_rec_form()

Megjeleníti az eredetileg elrejtett feltöltési űrlapot, mely kitöltés után visszaküldi az adatokat a szervernek.

4.2.1.2 add_record()

Az űrlap által visszaküldött adatok átvételét végzi el.

Funkciója négy részre bontható:

1. Ellenőrzi azon adatokat, melyek hibája esetén nem folytatható a feldolgozás:
 - ki lett töltve a név mező
 - érvényes EOY koordináta lett megadva
 - síkrajz esetén fel lett töltve térkép-fájl is

- a térképájl mérete nem haladja meg a szerver által megengedett maximális méretet
2. ha történt fájlfeltöltés, meghívja a `try_save_file($file_array)` függvényt, és ellenőrzi annak visszatérési értékét, hogy hiba esetén ne folytatódjon az adatok felvitele
 3. meghívja a `convert_record($record)` függvényt, mely a megadott adatok átalakításáért felel, hogy fel lehessen venni az adatbázisba
 4. végezetül meghívja a `save_record($record)` függvényt, mely az adatbázisba írja a már megfelelő formátumúvá alakított adatokat

4.2.1.3 `try_save_file($file_array)`

Ez a függvény végzi el a fájl mentését, ami egy összetett feladat. A PHP a fájlt egy ideglenes néven a szerver tmp könyvtárába menti, és a `$_FILES` szuperglobális tömbbe elmenti a fájl fő adatait.

A függvény először ellenőrzi, hogy a feltöltött fájl formátuma megfelelő-e. Mivel az AutoCAD DXF fájlformátumának nincsenek azonosítókarakterei, mint a bináris fájlformátumoknak, ezért csak a fájl kiterjesztése ellenőrizhető PHP-ből.

Második lépésben konvertálja a fájl nevét, hogy csak az angol ábécében szereplő kis- és nagybetűk, számok, és elfogadható írásjelek legyenek benne (pont, alsó-vonal, mínusz).

Végül ellenőrzi, hogy létezik-e az adott nevű fájl a térképek könyvtárában, és ha nem, akkor elmenti a fájlt.

4.2.1.4 `convert_record($record)`

Az űrlap kitöltéséből származó nyers adatokat konvertálja, hogy azok felvehetőek legyenek az adatbázisba.

- A `nev` mezőt szöveggé konvertálja, hogy nehogy számként próbálja átadni az adatbázisnak, ha esetleg csak számok kerültek a mezőbe.
- Az `EOVY` és `EOVX` mezőkben a vesszőket lecseréli pontra, és az eredményt lebegőpontos számmá konvertálja.
- Ellenőrzi, hogy érvényes-e a dátum, és ha igen, akkor `éééé-hh-nn` formátumban,

szöveggént tárolja. Érvénytelen dátum esetén 2008-01-01-et tárol el, és figyelmezteti a feltöltőt.

- Ellenőrzi, hogy a munkamennyiség 1 és 5 közt van-e, és egész számként tárolja. Hibás érték esetén 3-at ad meg, és figyelmezteti a felhasználót a változtatásra.
- Megvizsgálja a `sikrajz` és a `foldhiv` mezők értékét. Ha értékük "on", tehát be volt pipálva a mező, akkor értéküket "true"-ra, különben "false"-ra állítja.
- A `kulcs_szavak`, `megrendelo`, `megrend_tip` mezők tartalmát szintén szöveggé konvertálja.
- Ha van feltöltött fájl, annak konvertált nevét szintén felveszi az adatbázisba mentendő adatok közé, ha nincs, akkor üres szövegmezőt illeszt a helyére.
- Meghívja az `eov2wgs` (`$eovy`, `$eovx`) függvényt, mely a koordinátakonverziót végzi.
- A megkapott WGS84 koordinátákat egy szövegmezőbe is lementi, melyet a PostgreSQL geometriai adatként tud értelmezni:

```
"ST_GeomFromText ('POINT(" . $conv_record['WGS84lon'] .  
" " . $conv_record['WGS84lat'] . ")', 4326)"
```

A szöveg végén található 4326 szám a WGS84 rendszer EPSG¹ kódja.
- Végül az összeállított adattömböt visszaküldi az őt meghívó függvénynek.

4.2.1.5 eov2wgs(\$eovy,\$eovx)

A megkapott EOVS koordináta-párt egy külső program segítségével konvertálja WGS84 hosszúsági és szélességi koordinátává.

Bár a konverziót elvileg el tudná végezni a PostgreSQL is, ám a beépített paraméterkészlet 100 m-es nagyságrendű hibákat eredményez az ország egész területén.

Ennek kiküszöbölésére használtam fel a PROJ. 4 projekciós függvénykönyvtár `cs2cs` programját. A programnak átadtam az EOVS koordinátapárt, és dr. Siki Zoltán QGIS jegyzetéből [7] vett paraméterkészletet:

```
+proj=somerc +lat_0=47.14439372222222  
+lon_0=19.048571777777778 +x_0=650000 +y_0=200000  
+ellps=WGS84 +datum=WGS84 +towgs84=+57.01,-69.97,-9.29
```

¹ EPSG: *European Petroleum Survey Group* által összeállított *Geodetic Parameter Dataset* része.

```
+units=m +no_defs +k_0=0.99993
```

Látható, hogy a transzformáció úgy tekint az EOVS koordinátákra, mintha azok WGS84 ellipszoidról lennének hengerre vetítve a Swiss Oblique Mercator Projection alapján (+proj=somerc), mely a magyar vetületi rendszerhez hasonlóan kettős, ferde tengelyű vetítés. Ezen kívül meg van adva a vetület ellipszoidi középpontja, a koordináták eltolási értékei, és a méretarány-tényező értéke is. Azonban ezekkel a paraméterekkel önmagukban nagyon rossz transzformációt kapnánk, ezért van szükség a +towgs84 paraméterre, mely egy három, vagy hét paraméteres transzformációt ad meg. Három paraméter esetén térbeli eltolásokat jelent méter egységben, míg hét paraméter esetén térbeli hasonlósági transzformációt végez.

Az így kapott koordináták méter alatti transzformációs hibával rendelkeznek, ami az adott felhasználásnál megfelelőnek tekinthető.

A `cs2cs` program visszatérési értéke tartalmaz egy ellipszoid feletti magasságot is, ezt azonban a függvény már nem küldi vissza, csak az ellipszoidi hosszúságot és szélességet.

4.2.1.6 save_record(\$record)

Ez a függvény kapcsolódik a PostgreSQL-hez, és kiírja a rekordot az adattáblába.

4.2.2 Rekordok kilistázása, törlése

4.2.2.1 list_records()

Ez a függvény kapcsolódik az adatbázishoz, lekérdezi az összes rekordot, majd az ID és a `nev` mező alapján a HTML formátumnak megfelelő listát készít. Ebből a listából választhatja ki a felhasználó a módosítani, vagy törölni kívánt térképeket.

4.2.2.2 del_record(\$id)

A paraméterként kapott rekordazonosító alapján lekérdezi a rekordhoz tartozó fájlnevet, és ha az nem üres, akkor meghívja a `del_map($fname)` függvényt. Ezt követően törli magát a rekordot is. Végezetül pedig lefuttatja a `generate_viewer.inc` fájlban található programkódot, mely frissíti a felhasználói felületet, hogy annak listájában se jelenjen meg a törölt rekord.

4.2.2.3 del_map(\$fname)

Törli a megadott fájlt, majd lefuttatja a `generate_map.inc` fájlban található programkódot, mely a MapServer adatfájlját frissíti. Erre azért van szükség, mert különben a MapServer megpróbálná megnyitni a már törölt fájlt, ami azt eredményezi, hogy leáll a feldolgozásban, és semmit sem jelenít meg a felhasználói felületen. A másik lehetőség a map fájl dinamikus generálása lenne minden lekérdezéskor, ám ez jelentősen lassítaná a felhasználók kiszolgálását.

4.2.3 Rekordok módosítása

A rekordok módosításához használt függvények az `edit.inc` fájlban találhatóak.

4.2.3.1 show_modrec(\$id)

Lekérdezi az adatbázisból a kiválasztott rekordot, majd egy űrlapban megjeleníti az adminisztrátornak.

4.2.3.2 edit_record()

A függvény először kapcsolódik az adatbázishoz, és lekérdezi a módosítani kívánt rekordot. Erre azért van szükség, mert ha új fájl került feltöltésre, akkor a régit törölni kell, ha viszont megmaradt a régi, akkor annak elérését kell lementeni a módosított rekordba is.

Ezt követi az alapvető adatok ellenőrzése, melyet a feltöltésnél már ismertettem, majd a térkép fájl frissítése (régitörlése, új mentése) a már ismertetett `del_map($fname)` és `try_save_file($file_array)` függvények meghívásával.

Következő feladat a módosított adatok konvertálása az adatbázisnak megfelelő formátumúvá a fentebb már leírt `convert_record($record)` függvény segítségével.

Végül pedig lefuttatja az `update_record($record)` függvényt.

4.2.3.3 update_record(\$record)

Ez a függvény kapcsolódik az adatbázishoz, majd elvégzi a rekord frissítését a

paraméterként kapott tömbben található adatok alapján.

4.2.4 A MapServer fájl és a felhasználói felület generálása

Mint az fentebb látható volt, a konzisztencia megőrzése érdekében törléskor automatikusan frissülnek a megjelenítésért felelős fájlok, azonban a feltöltést követően ez nem történik meg, csak ha az adminisztrátor külön rányom a „térkép frissítése” gombra.

4.2.4.1 generate_map.inc

Az AutoCAD DXF fájlok megjelenítését a MapServer a `dxf_viewer.map` fájl alapján végzi. Ennek a fájlnak a tartalmát generálja le a `generate_map.inc` fájlban található programkód.

A `map` fájl a MapServer alapvető működését adja meg: a benne található adatokból tudja a program, hogy milyen típusú adatokat milyen formátumban kell megjelenítenie.

Mivel a rendszer egyelőre csak DXF fájlok megjelenítését támogatja, ezért a kód viszonylag egyszerű `map` fájlt készít.

A `map` fájlban van egy állandó bevezető része, ahol meg kell adni a megjelenítési mértékegységet (méter), a térképátlak elérési útját a szerveren, a weben megjelenő képek formátumát (png) és méretét (500x500 pixel), a kimeneti vetületi rendszert (epsg:4326), illetve a felhasználható betűtípusok elérési útját arra az esetre, ha szövegeket is szeretnénk megjeleníteni az oldalon. Erre majd a felhasználói felület tárgyalásánál lesz példa.

A `map` fájl másik része a feltöltött fájlokhoz megfelelően változó `LAYER` definíciók. A MapServer-t eredetileg az ESRI `shapefile` formátumának megfelelő megjelenítésre alakították ki, ahol egy-egy fájl valóban egy réteget tartalmaz, illetve a fájlok csak az adatokat tartalmazzák, formázást nem. Ennek megfelelően a `LAYER` részlegben lehetőség van méretarány függő formázási beállítások megadására is.

A mi esetünkben azonban egy-egy `dxf` fájl több réteget is tartalmaz, és a formázás is magában a fájlban található meg. Ebből adódóan minden fájlhoz elegendő lenne egy-egy `LAYER` definíciót létrehozni, azonban az adatok értelmezéséért felelős GDAL/OGR függvénykönyvtár egymástól függetlenül dolgozza fel a rajzi és a szöveges objektumokat. Ezért fájlanként két `LAYER` definíció található, melyek nevükben, és típusukban (line /

annotation) különböznek. Ezen kívül meg kell adni minden esetben az adatok geometriai értelmezéséhez szükséges projekciót, aminek adatai nem, csak formátuma változott a fentebb már ismertetthez képest:

```
PROJECTION
    "proj=somerc"
    "lat_0=47.14439372222222"
    "lon_0=19.04857177777778"
    "x_0=650000"
    "y_0=200000"
    "ellps=WGS84"
    "datum=WGS84"
    "towgs84=+57.01,-69.97,-9.29"
    "units=m"
    "no_defs"
    "k_0=0.99993"
END
```

4.2.4.2 generate_viewer.inc

Az ebben a fájlban található programkód a `viewer.php.inc` fájlt generálja le a munkakönyvtárba, amit aztán a felhasználók által elért `viewer.php` tölt be. Erre a megoldásra ezért van szükség, mert a PHP-nak biztonsági okokból nincs joga írni a webről közvetlenül elérhető tartalmaz, így kivédve, hogy egy behatoló kihasználva egy esetleges programozási hibát, létrehozson tudni a szerveren olyan ártalmas kódot, akit aztán közvetlenül le is tud futtatni, így hozzáférve bármilyen, a gépen található tartalomhoz.

A programkód a későbbiekben a felhasználói felületnél ismertetett statikus tartalom túl egy változó részt ír bele a fájlba. Ez annak a listának a tartalma, melyből a felhasználó ki tudja választani, hogy melyik térképre szeretne ugrani. Ehhez a program csatlakozik az adatbázishoz, lekérdezi a rekordokat, majd létrehozza a HTML kódba írható szöveget, mely tartalmazza a WGS84 koordinátákat, és megjeleníti az adott rekordba írt nevet.

4.3 A felhasználói felület tervezése, megvalósítása

A felhasználói oldalnál fontos szempont volt, hogy nem egy önálló oldalt kellett alkotni, hanem egy honlapra beszúrható elemet. Ennek megfelelően az oldalnak kicsinek, és mindenféle formázástól mentesnek kellett lennie, hogy akár a honlap kinézetének változása esetén is elegendő legyen a CSS stílusfájlt átírni, és ne kelljen a térképnezegető kódjába belenyúlni. Másik szempont az átláthatóság, egyértelmű kezelhetőség volt, hogy bármely látogató rögtön átlássa, és tudja használni az oldalt.

Az oldalnak két alapvető funkciója van: a megjelenítés, és a keresés.

4.3.1 Térkép megjelenítése

Az oldal fő tartalmát a MapServerre hivatkozó JavaScript kód alkotja. Az HTML kód fejrészében található meg a külső JavaScript hivatkozások az OpenLayers és a Google Maps függvénykönyvtárára. Míg az első a kezelőfelület és a térképek megjelenítéséért, addig a második a háttértérképek elérhetőségéért felel. Ezeken kívül található a fejrészben még egy `init()` függvény, melyet a böngésző az oldal betöltésekor futtat le.

Az `init()` függvény adja meg az OpenLayers függvénykönyvtárának, hogy mi, és milyen formában jelenjen meg az oldalon:

- beállítja a szélső koordinátákat, ameddig a térképet el lehet húzni: a bal felső és a jobb alsó sarok WGS84 koordinátáit adja meg, melyet Magyarország körül definiáltam (15.825806,45.6504; 23.120728,48.739769)
- megadja, hogy milyen eszközök jelenjenek meg a térképen:
 - navigációs nyilak
 - rétegválasztó
 - léptékvonal
 - méretarány
 - link az adott nézetre
 - nagyító csúszka
- definiálja a rétegeket:

- a `dxf` fájlokat megjelenítő `.map` fájl
 - Google utca térkép
 - Google műhold térkép
 - előző kettőből összeálló hibrid térkép
 - Google fizikai földfelszín térkép
 - a `dxf` fájlok középpontjait, és a hozzájuk tartozó neveket megjelenítő `.map` fájl
- megadja a térkép középpontját, mely alapértelmezés szerint a Budakeszin végzett telekalakításra áll be, ám ha az oldal linkjében más van megadva, akkor a kapott koordinátára ugrik.

A Google rétegek szolgáltatják a háttérrel. Ezek közül egyszerre csak egy lehet kiválasztva. Ezeken kívül két MapServer réteget tartalmaz az oldal, melyek tetszés szerint ki/be kapcsolhatóak. Az egyik a már bemutatott AutoCAD DXF fájlokat megjelenítő `.map` fájlt takarja, míg a másik a `points.map` fájlra hivatkozik.

A `points.map` az PostgreSQL adatbázishoz kapcsolódik, és onnan veszi az adatokat. A `wgs84` nevű geometriai típusú mezőből veszi a térképekhez tartozó középpontokat, és a `nev` mezőből a neveket. Ezen adatok alapján minden térképhez generál egy pontot, és alá írja a megadott nevet.

A térkép ablak mellett található a feltöltött térképek listáját tartalmazó böngészőablak, melyben ha rákattint a felhasználó egy térkép névre, akkor a térképablakban az adott felmérés jelenik meg. A lista fölött található a kereső doboz, melynek működését az alábbiakban ismertetem.

4.3.2 Keresés az adatbázisban

A keresést a `keres.inc` fájlban található programkód végzi. A felhasználó az oldal meglátogatásakor a térkép melletti listában az összes feltöltött térképet látja. A keresési funkció használatával ezek közt tud szűrést végezni. A cél egy szabadszavas kereső megalkotása volt, ebből következően mind szavakra, mind szótöredékekre lehet keresni az adatbázisban, és a kereső nem kis-nagybetű érzékeny.

A keresés egyetlen függvényből áll, mely akkor fut le, ha a felhasználó beírt valamit a kereső mezőbe, majd rányomott a Keresés gombra.

A függvény először egy tisztítást hajt végre a keresett kifejezésen, mely során csak a

magyar ábécé betűit és a számokat hagyja meg, minden mást szóközre cserél. Második lépésként szétvágja a szöveget, és az első tíz szót egy tömbbe teszi. A korlátozásra azért volt szükség, hogy a keresés gyorsan lefusson, illetve ennél több szóra nem is érdemes egyszerre keresni.

Ezt követően a program minden szóra elvégzi a keresést az adatbázison a `kulcs_szavak` és a `nev` mezőkön, rögzítve, hogy mely rekordra hány találat érkezett. A keresés után összeszámolja a találatokat, és csökkenő sorba rendezi az eredményt. Végül lekérdezi a rekordokat, és sorrendnek megfelelően generál egy HTML listát a WGS84 koordinátákból és a nevekből, mely megjelenik a felhasználó számára, mint eredmény.

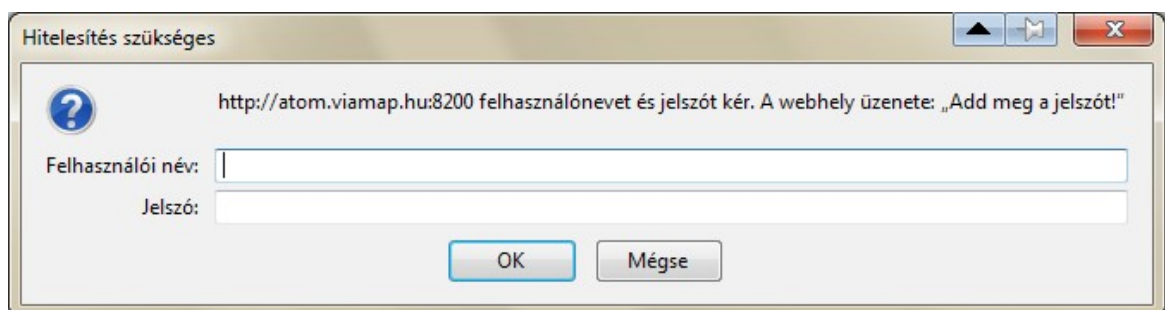
Így a felhasználó egy olyan listát kap, melyben a legrelevánsabb találat van legfelül. Amennyiben nincs találat, illetve a felhasználó üres keresést hajt végre, úgy az eredeti állapotra áll vissza a lista tartalma, illetve kiír a program egy értesítést a keresődoboz alá, melyben tájékoztatja a felhasználót, hogy a keresés eredménytelen volt.

5 Eredmények, a rendszer használatának bemutatása

A rendszer alapjai megvalósultak, és készek a honlapba illesztésre. Rengeteg további fejlesztési lehetőség van, ám ezeket majd a következő fejezetben fogom ismertetni.

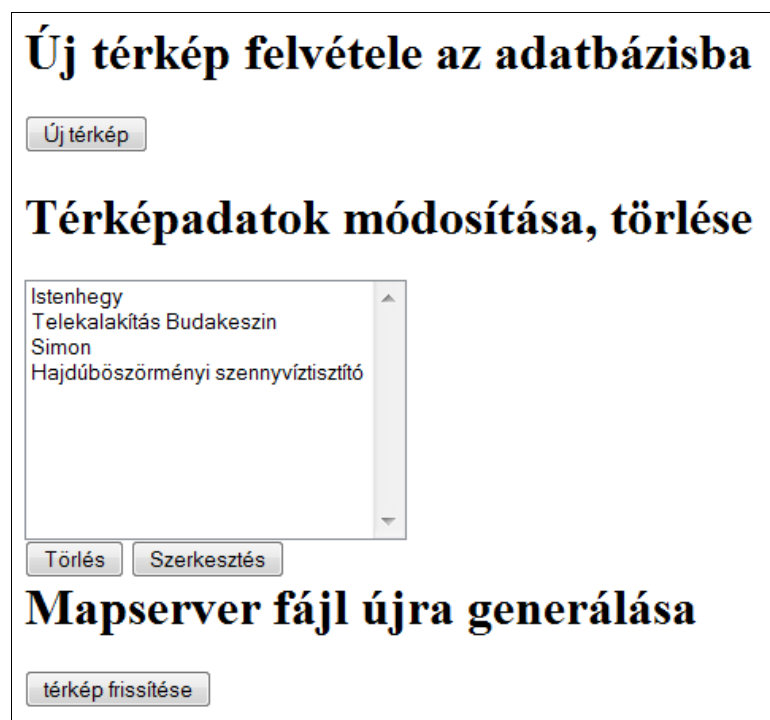
5.1 Adminisztrációs felület

Az adminisztrációs felületen keresztül az adminisztrátor kezelni tudja az adatokat. Az admin.php megnyitásakor megjelenik egy ablak, melyben meg kell adni a felhasználónevet és a jelszót:



5.1. ábra: hitelesítés

Helyes jelszó megadása esetén megjelenik az adminisztrációs felület:



5.2. ábra: adminisztrációs felület

Az adminisztrációs felületen lehetőség van új térképet felvenni az adatbázisba, illetve a már fent lévőket szerkeszteni, vagy törölni.

Az Új térkép gombra kattintva megjelenik a beviteli űrlap:



The image shows a web form titled "Új térkép felvétele az adatbázisba". The form contains the following fields and controls:

- Megjelenített név: [text input]
- Felmérés időpontja: [date input]
- Munka mennyisége: [dropdown menu]
- Síkrajz?
- Földhivatal?
- Kulcsszavak: [text input]
- Megrendelő neve: [text input]
- Megrendelő típusa: [text input]
- EOV Y: [text input]
- EOV X: [text input]
- térkép fájl: [file upload field] with a "Tallózás..." button
- A "Feltöltés!" button is located at the bottom left of the form.

5.3. ábra: Új térkép űrlap

Az adatok kitöltése után elvégezhető a feltöltés. Kötelezően kitöltendő adatok a megjelenített név és az EOV koordinátapár, illetve síkrajz esetén maga a térkép fájl. Ezek hiányában az rekord nem kerül tárolásra. A többi mezőt nem kötelező kitölteni, azonban a munkamennyiség és a dátum érvénytelen adat esetén kap egy alapértelmezett értéket, melyek utólag lehet módosítani a szerkesztés menüpontban.

A Térképadatok módosítása, törlése részben egy lista található a már feltöltött adatrekordokról. Értelem szerűen a Törlés gombbal lehet törölni, míg a Szerkesztés gombbal szerkeszteni a kiválasztott rekordot.

A szerkesztést kiválasztva megjelenik egy, a feltöltéshez hasonló űrlap, melyben láthatóak a régi adatok.

Térképadatok módosítása, törlése

Istenhegy
Telekalakítás Budakeszin
próba
Simon
Hajdúböszörményi szennyvíztisztító

Törlés
Szerkesztés

Megjelenített név:

Felmérés időpontja:

Munka mennyisége: ▼

Sikrajz?

Földhivatal?

Kulcsszavak:

Megrendelő neve:

Megrendelő típusa:

EOV Y:

EOV X:

jelenlegi térkép fájl:

térkép frissítése: Tallózás...

Módosítás!

5.4. ábra: Módosítás űrlap

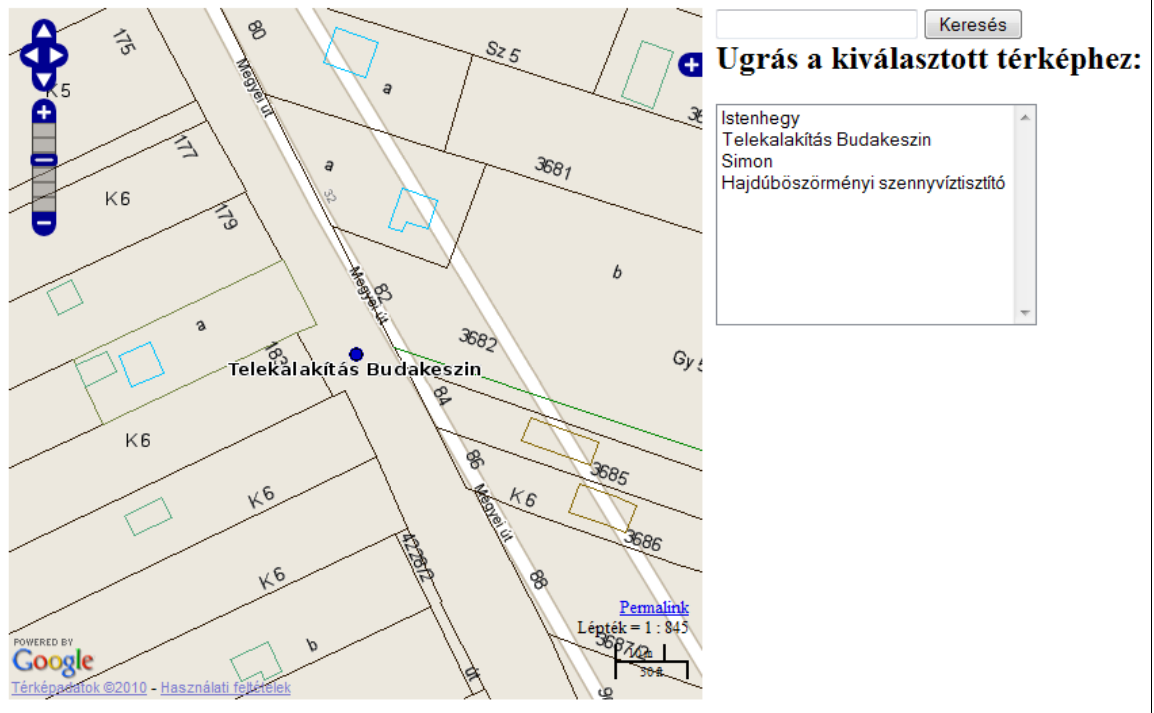
Változtatás után a módosítás gombbal lehet elmenteni az új adatokat. Látható, hogy szerkesztéskor nem csak az adatok, de a térkép fájl is frissíthető. Amennyiben az adminisztrátor új fájlt ad meg, a régi törlésre kerül, azonban, ha nem ad meg új fájlt, úgy a régi marad a rekordhoz mentve.

Az adminisztrációs felület utolsó része a frissítés, mely legenerálja az új .map fájlt, és frissíti a felhasználói felület térképlistáját. Törléskor ugyan automatikusan frissülnek ezek a fájlok, azonban feltöltés, módosítás után le kell futtatni a frissítést, hogy azok meg is jelenjenek a térképen.

5.2 Felhasználói felület

A felhasználói felületen alapértelmezés szerint az egyik térkép jelenik meg, mellette a térképek listája, és a kereső mező.

Térkép megjelenítő



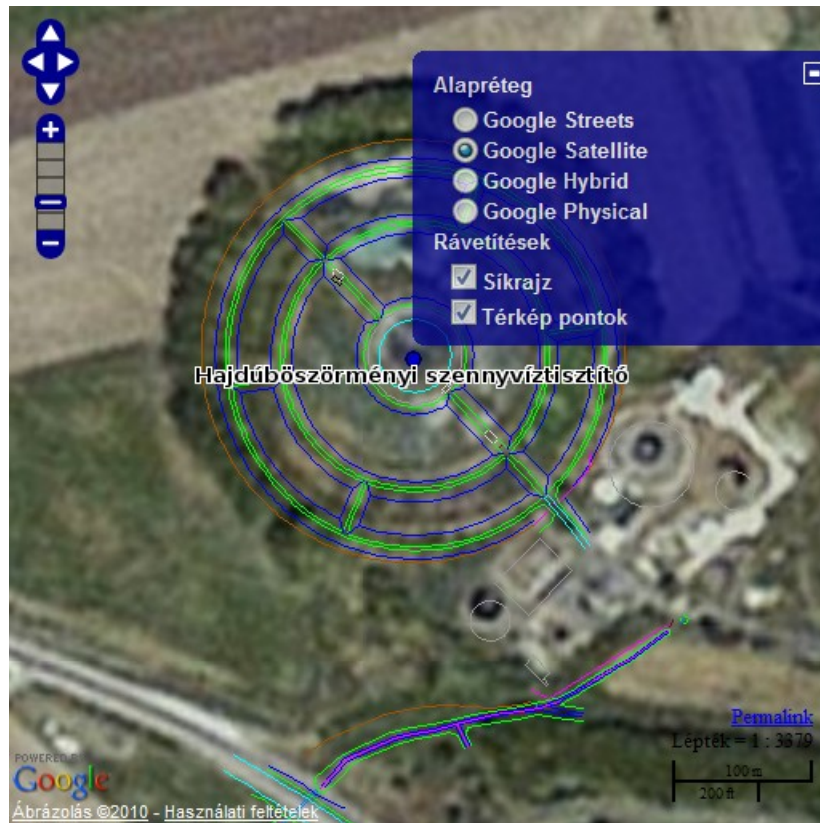
5.5. ábra: Felhasználói felület

A térkép bal felső sarkában jelennek meg a vezérlőgombok, melyekkel kis lépésekkel mozgatható a térkép. Alatta a nagyítósáv, mellyel a térkép méretarányát lehet változtatni. Természetesen a térképet lehet mozgatni egérrel is: vonszolással mozgatható, míg görgővel nagyítható és kicsinyíthető a térkép.

A jobb felső sarokban található a rétegválasztó menü, melyet a plusz gombra kattintva lehet megnyitni. Itt lehetőség van kiválasztani, hogy melyik alapréteget, és mely rávetítéseket kívánjuk megjeleníteni. Az alapréteg adja meg, hogy milyen háttéren szeretnénk megjeleníteni a térképeket:

- Google utcatérkép
- Google műholdtérkép
- előző kettő hibridje
- Google fizikai földfelszín térképe

A rávetítések a MapServer által generált képek. Itt ki-be lehet kapcsolni a síkrajzok, illetve a térképek közepét jelző pontok megjelenítését. A térképpontok alatt megjelenő nevek szintén a pontok réteghez tartoznak.



5.6. ábra: rétegválasztó

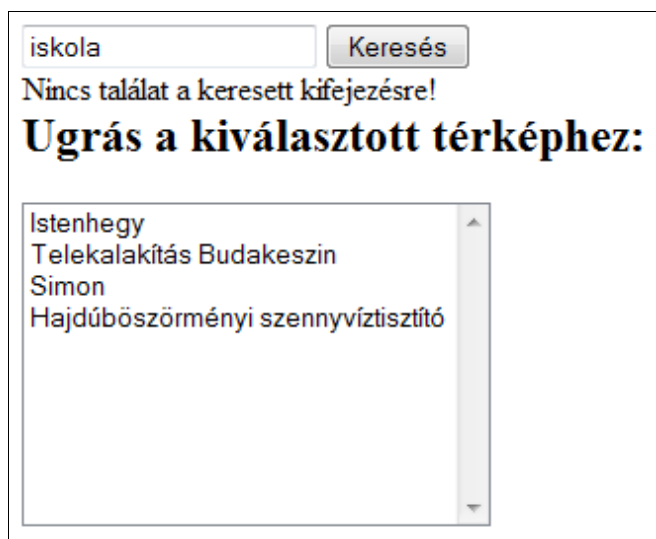
A jobb alsó sarokban a méretarányon és a léptéken kívül egy permalink is, mellyel az adott nézetről lehet linket készíteni. Ezt aztán elmenthetjük, vagy elküldhetjük másoknak, akik a linkre kattintva ugyanazon a nagyítási szinten, ugyanabban a pozícióban, és ugyanazokkal a rétegekkel nézhetik meg a térképet, mint amiben mi elmentettük.

A térkép mellett található keresőmező segítségével szűrni lehet a térképek listáját. Beírva egy kifejezést, az arra leginkább illeszkedő találat kerül az első helyre, és az egyre kevésbé releváns találatok egyre lejjebb. Azon térképek, melyekre nincs találat, nem jelennek meg a listában.



5.7. ábra: Sikeres keresés

Ha egy találat sincs, akkor a felhasználó értesítést kap, és a lista tartalma visszaáll az alapértelmezett állapotra.



5.8. ábra: Sikertelen keresés

6 A rendszer továbbfejlesztésének lehetőségei

6.1 Támogatott fájlformátumok bővítése

A rendszer jelenleg csak `dxf` állományok megjelenítésére képes, azonban a `MapServer` önmagában is képes megjeleníteni az `ESRI Shape` fájlokat, illetve `PostgreSQL` vagy `MySQL` adatbázisból adatokat, a `GDAL/OGR` függvénykönyvtárak pedig 51 vektor és 115 raszter állomány típust támogatnak, melyek várhatóan tovább fognak bővülni, ahogy a fejlesztésük halad előre. [8], [9]

A fájlformátumok támogatásával kapcsolatban két fejlesztési irány lehetséges. Egyrészt meg lehet valósítani a megjelenítésüket külön-külön, ami azonban egy bonyolult folyamat lenne, és minden hozzáadott formátum tovább bonyolítaná a programkódot, nehezítve a használatot, átláthatóságot, fejlesztést. A másik lehetőség egy univerzális megjelenítő megvalósítása, mely az `OGR` közvetlen meghívásával próbálná eldönteni, hogy egy-egy fájl megjeleníthető-e, és ha igen, akkor milyen paraméterkészlet használatával.

6.2 Felhasználói felület fejlesztése

A tervek közt szerepel, hogy az adott térkép adatai megjelenjenek egy szövegbuborékban, amikor a felhasználó a térkép középpontját jelző pontra kattint. Ezzel együtt egy látványosabb kezelőfelületet is kaphatna a felhasználói felület. Ezen változtatásokhoz a `GeoExt` JavaScript függvénykönyvtár használata lenne célravezető, mellyel azonban még csak ismerkedem, így a fejlesztés a dolgozat beadásig nem tudtam megvalósítani

7 Összegzés, tapasztalatok

Megismerkedve a nyílt forráskódú alkalmazások használatával, szembesülnöm kellett azzal, hogy egy komoly, jól kiforrott program készítéséhez továbbra is rengeteg pénz és idő kell. Az elterjedt, általános célú alkalmazások, mint az `Apache Server`, vagy a `PHP`, probléma nélkül, gördülékenyen működnek. Ha fejlesztés közben a programozóban valami kérdés merül fel, akkor arra gyorsan talál választ az interneten, mivel mások is napi rendszerességgel használják ezeket az alkalmazásokat, és tapasztalataikat már megosztották egymással fórumokon és egyéb oldalakon, illetve tapasztalataikkal hozzájárultak a fejlesztéshez is. Természetesen az angol nyelvtudás elengedhetetlen a válaszok felkutatásához, de ez manapság a legtöbb területre, és főleg az informatikára igaz.

Ellenben a jóval kisebb közösség által használt térinformatikai megoldások, mint a `MapServer` és az `OpenLayers` sokszor küzd problémákkal. Bár a `MapServer` története egészen a kilencvenes évek elejéig nyúlik vissza, sokáig csak egy igen szűk réteg használta. A webes, közösségi térképezés a `Google Maps` megjelenése után, 2006-tól indult meg jelentősen. Ekkor alapították az `Open Source Geospatial Foundation`-t, mely felvette projektjei közé előbb az `OpenLayerst` (2007 nov.), majd a `MapServer`-t (2008 dec.) is. [10] Az elmúlt pár évben sokat fejlődtek ezek az alkalmazások, ám még messze vannak a kényelmes fejlesztést biztosító megoldásoktól. Erre példa, hogy az EOV vetületű térképeket a beépített paraméterkészlet 100m-es hibával transzformálta, ezért meg kellett találnom a megfelelő paramétereket.

Azonban kisebb-nagyobb nehézségek árán a problémák megoldhatóak, illetve nem garantált, hogy ugyanezek a problémák nem jelentkeznek a drága szoftvermegoldásoknál is.

8 Irodalomjegyzék

- 1: Wikipedia, GNU General Public License, 2010, <http://en.wikipedia.org/wiki/GPL>**
- 2: GNU Project, GNU General Public License, 2007,
<http://www.gnu.org/licenses/gpl.html>**
- 3: Apache HTTP Server Project, About the Apache HTTP Server Project, 2010,
http://httpd.apache.org/ABOUT_APACHE.html**
- 4: The PHP Group, PHP: General Information, 2010,
<http://php.net/manual/en/faq.general.php>**
- 5: Wikipedia, PostgreSQL, 2010, <http://en.wikipedia.org/wiki/PostgreSQL>**
- 6: Siki Zoltán, Geodéziai és térinformatikai szabadszoftverek, 2010,
<http://www.agt.bme.hu/gis/gis.html>**
- 7: Siki Zoltán, Vetületek és vetületi átszámítások a QGIS-ban (1.5 verzió), 2010,
<http://www.agt.bme.hu/gis/qgis/proj.pdf>**
- 8: Open Source Geospatial Foundation, OGR Vector Formats, 2010,
http://www.gdal.org/ogr/ogr_formats.html**
- 9: Open Source Geospatial Foundation, GDAL Raster Formats, 2010,
http://www.gdal.org/formats_list.html**
- 10: Open Source Geospatial Foundation, MapServer History, 2010,
<http://trac.osgeo.org/mapserver/wiki/MapServerHistory>**

9 Mellékletek

9.1 A program térinformatikai szempontból fontos kódjai

9.1.1 A keresés megvalósítása (keres.inc)

```
02: function keres () {
03:     $str = preg_replace ('/^[[:alnum:]]áéúóóúúóúóíÁÉŰÚÓÓÜÖÍ/' , " ",
$_REQUEST['keres']);
04:     $words = array_slice ( preg_split ("/[\s]/" , $str , null ,
PREG_SPLIT_NO_EMPTY) , 0 , 10 );
05:
06:     if ( count ($words) == 0 ) {
07:         return 0;
08:     }
09:
10:     $dbconn = pg_pconnect (DB_STR) or die ("Nem lehet csatlakozni
az adatbázishoz: " . pg_last_error ());
11:     $result = pg_prepare ($dbconn , "key_word_query" , 'SELECT "ID"
FROM szakdolg_terkep WHERE kulcs_szavak ILIKE $1 OR nev ILIKE $1;');
12:     $ids = NULL;
13:     foreach ($words as $word) {
14:         $word = "%" . $word . "%";
15:         $result = pg_execute ($dbconn , "key_word_query" ,
array ("$word"));
16:         while ( $record = pg_fetch_array ($result) ) {
17:             $ids[] = $record[0];
18:         }
19:
20:     }
21:     pg_query ('DEALLOCATE "key_word_query"');
22:
23:     if ( $ids == NULL ) {
24:         return 0;
25:     }
26:
27:     $idlist = NULL;
28:     foreach ($ids as $id) {
```

```

29:             $idlist[$id]++;
30:         }
31:         arsort($idlist);
32:         $ker_eredmeny = NULL;
33:
34:         $dbconn = pg_pconnect(DB_STR) or die("Nem lehet csatlakozni
az adatbázishoz: " . pg_last_error());
35:         foreach ($idlist as $id => $count){
36:             $query = "SELECT * FROM szakdolg_terkep
WHERE \"ID\" = $id;";
37:             $result = pg_query($dbconn, $query);
38:             $record = pg_fetch_array($result);
39:             $ker_eredmeny .="\t\t\t<option value=" .
$record['WGS84lon'] ."," . $record['WGS84lat'] . ">" . $record['nev'] .
"</option>\n";
40:         }
41:         return $ker_eredmeny;
42: }

```

9.1.2 AutoCAD DXF fájlok megjelenítése (dxf_viewer.map)

```

001: MAP
002:   NAME dxf_maps
003:   SIZE 500 500
004:   UNITS meters
005:   SHAPEPATH "/home/mapadmin/kadari/dxf/"
006:   FONTSET "/home/mapadmin/kadari/fontset/fontset.txt"
007:
008:   PROJECTION
009:     "init=epsg:4326"
010:   END
011:
012:   IMAGECOLOR 192 192 192
013:   IMAGEQUALITY 95
014:   IMAGETYPE png
015:   OUTPUTFORMAT
016:     NAME png
017:     DRIVER "GD/PNG"
018:     MIMETYPE "image/png"

```

```

019:     EXTENSION "png"
020: END
021:
022: WEB
023:     TEMPLATE "/home/mapadmin/www/kadari/./test2/first.html"
024:     IMAGEPATH "/home/mapadmin/www/kadari/./tmp/"
025:     IMAGEURL  "/tmp/"
026: END
027:
028: LAYER
029:     NAME "bk3682-3685telekalak2andras_v"
030:     TYPE LINE
031:         CONNECTIONTYPE OGR
032:         CONNECTION "bk3682-3685telekalak2andras_v.dxf"
033:         STATUS default
034:         STYLEITEM "AUTO"
035:         PROJECTION
036:             "proj=somerc"
037:             "lat_0=47.14439372222222"
038:             "lon_0=19.048571777777778"
039:             "x_0=650000"
040:             "y_0=200000"
041:             "ellps=WGS84"
042:             "datum=WGS84"
043:             "towgs84=+57.01,-69.97,-9.29"
044:             "units=m"
045:             "no_defs"
046:             "k_0=0.99993"
047:         END
048:
049:     CLASS
050:         NAME 'map_style'
051:         STYLE
052:             SYMBOL 0
053:             SIZE 5
054:             OUTLINECOLOR 144 50 207
055:             COLOR 0 0 0
056:         END
057:     END

```



```

058: END
059:
060: LAYER
061:     NAME "bk3682-3685telekalak2andras_v_anno"
062:     TYPE annotation
063:     CONNECTIONTYPE OGR
064:     CONNECTION "bk3682-3685telekalak2andras_v.dxf"
065:     STATUS default
066:     STYLEITEM "AUTO"
067:     PROJECTION
068:         "proj=somerc"
069:         "lat_0=47.14439372222222"
070:         "lon_0=19.04857177777778"
071:         "x_0=650000"
072:         "y_0=200000"
073:         "ellps=WGS84"
074:         "datum=WGS84"
075:         "towgs84=+57.01,-69.97,-9.29"
076:         "units=m"
077:         "no_defs"
078:         "k_0=0.99993"
079:     END
080:
081:     CLASS
082:         NAME 'map_style'
083:         STYLE
084:             SYMBOL 0
085:             SIZE 5
086:             OUTLINECOLOR 144 50 207
087:             COLOR 0 0 0
088:         END
089:     END
090: END
091:

```

(a további rétegek az előzőhöz hasonlóan...)

```

283:
284: END

```

9.1.3 A térképek középpontjának és nevének megjelenítése (points.map)

```
01: MAP
02:   NAME postgre
03:   SIZE 500 500
04:   UNITS meters
05:   FONTSET "/home/mapadmin/kadari/fontset/fontset.txt"
06:
07:   PROJECTION
08:     "init=epsg:4326"
09:   END
10:
11:   IMAGECOLOR 192 192 192
12:   IMAGEQUALITY 95
13:   IMAGETYPE png
14:   OUTPUTFORMAT
15:     NAME png
16:     DRIVER "GD/PNG"
17:     MIMETYPE "image/png"
18:     EXTENSION "png"
19:   END
20:
21:   WEB
22:     TEMPLATE "/home/mapadmin/www/test2/first.html"
23:     IMAGEPATH "/home/mapadmin/www/tmp/"
24:     IMAGEURL "/tmp/"
25:     METADATA
26:       "wfs_title"           "Térkép adatok"  ## REQUIRED
27:       "wfs_onlineresource"
28:       "http://atom.viamap.hu:8200/cgi-bin/mapserv?
29:       map=/home/mapadmin/kadari/points.map"  ## Recommended
30:       "wfs_srs"           "EPSG:4326"  ## Recommended
31:       "wfs_abstract"      "A térképek középpontjai, és
32:       leíró adatai."  ## Recommended
33:     END
34:   END
35:   SYMBOL
```

```

34:          NAME 'point'
35:          TYPE ELLIPSE
36:          FILLED TRUE
37:          POINTS 2.83 2.83 END
38:      END
39:
40:  LAYER
41:      NAME "adatreteg"
42:      STATUS ON
43:      TYPE POINT
44:      CONNECTIONTYPE POSTGIS
45:      CONNECTION "host=***** dbname=***** user=*****
password=*****"
46:      DATA "wgs84 from szakdolg_terkep using srid=4326"
47:      DUMP TRUE
48:      METADATA
49:          "wfs_title"          "Adatréteg"
## REQUIRED
50:          "gml_featureid"      "ID" ## REQUIRED
51:          "wfs_srs"            "EPSG:4326" ## REQUIRED
52:          "gml_include_items"  "all" ## Optional
(serves all attributes for layer)
53:      END
54:
55:  CLASS
56:      NAME 'point_style'
57:      TEXT ([nev])
58:      STYLE
59:          SYMBOL "point"
60:          SIZE 10
61:          COLOR 0 0 200
62:          OUTLINECOLOR 0 0 0
63:          ANTIALIAS TRUE
64:          WIDTH 3
65:      END
66:      LABEL
67:          PARTIALS FALSE
68:          TYPE TRUETYPE
69:          FONT DejaVuSans-Bold

```

```
70:                SIZE 10
71:                COLOR 0 0 0
72:                OUTLINECOLOR 255 255 255
73:                OUTLINEWIDTH 2
74:                ANTIALIAS TRUE
75:                POSITION CC
76:                OFFSET 0 10
77:                END
78:    END
79: END
80:
81: END
```

9.2 A mellékelt CD tartalma

database.sql: Az adatbázis szerkezetét tartalmazó SQL fájl.

TDK dolgozat-kadari.pdf: A TDK dolgozat PDF formátumban.

TDK rezume-kadari.pdf: A rezümé PDF formátumban.

9.2.1 A work könyvtár tartalma

Ezek a fájlok nem érhetőek el közvetlenül az internetről.

.htpasses: Az adminisztrációs felület felhasználó neveit és jelszavait tartalmazó fájl.

programfájlok:

- **dx_viewer.map**
- **edit.inc**
- **generate_map.inc**
- **generate_viewer.inc**
- **globals.inc**
- **keres.inc**
- **points.map**
- **upload.inc**
- **viewer.php.inc**

9.2.2 A www könyvtár tartalma

Ennek a könyvtárnak a tartalma érhető el az internetről.

.htaccess: A mappa tartalmához való hozzáférést szabályozó fájl.

admin.php: Az adminisztrációs felület fájlja.

globals_path.inc: A globális állandókat betöltő fájl.

viewer.php: A felhasználói felületet betöltő fájl.