

EÖTVÖS LORÁND TUDOMÁNYEGYETEM INFORMATIKAI KAR
TÉRKÉPTUDOMÁNYI ÉS GEOINFORMATIKAI TANSZÉK

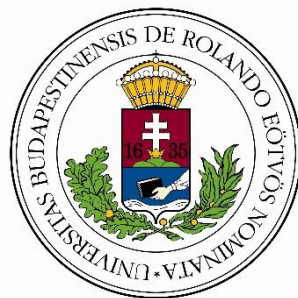
Térinformatikai háttér elkészítése a
„Dendrological Atlas” című botanikai
könyvsorozat areálgeográfiai adatbázisához

DIPLOMAMUNKA

KÉSZÍTETTE:
Csomós Boglárka
térképész mesterszakos hallgató

TÉMAVEZETŐ:
dr. Gede Mátyás
adjunktus
ELTE Térképtudományi és Geoinformatikai Tanszék

KÜLSŐ KONZULENS:
dr. Rácz István
Magyar Természettudományi Múzeum
Növénytár



Budapest, 2016

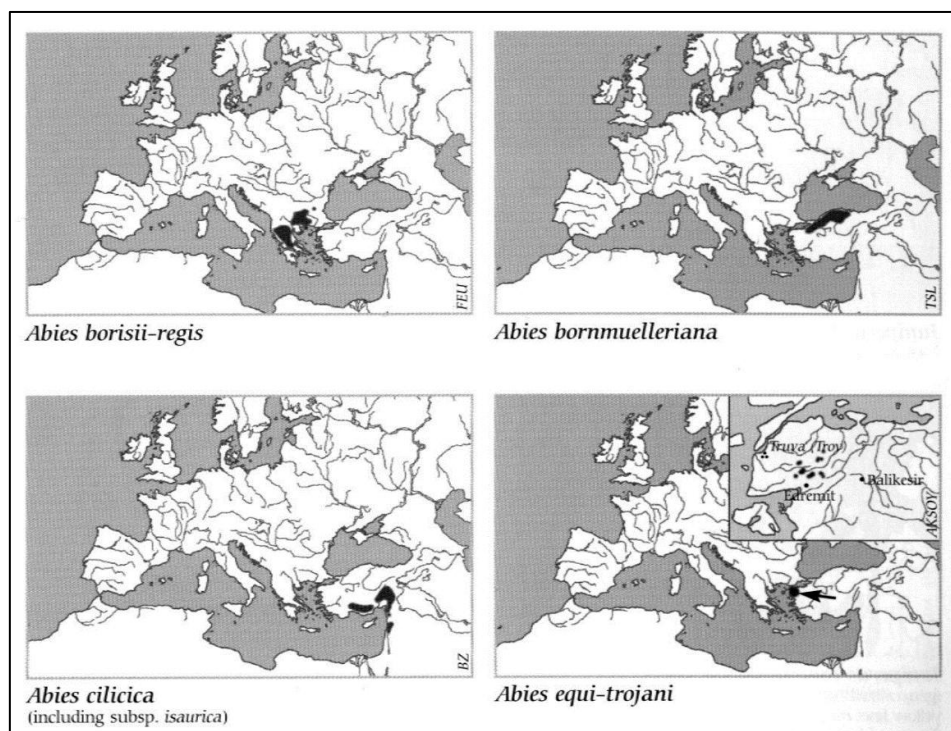
Tartalomjegyzék

1. Bevezetés	3
2. Az alapanyag feldolgozása.....	5
2.1 Georeferencia.....	5
2.2 Vektorizálás	6
2.3 Adatbázis felépítése	8
2.3.1 Rendszertani kategóriák	9
3. Megjelenítés eszközei	11
3.1 Egy weblap általános felépítése.....	11
3.2 OpenLayers.....	12
3.3 MapServer	14
3.4 MapBox	15
4. A weblap ismertetése	19
4.1 A weblap tartalmi elemei.....	20
4.2 Stílusbeállítások.....	21
4.3 Az OpenLayers eszközök	23
4.4 A mapfile tartalma	24
4.5 JavaScript kód.....	27
5. Összefoglalás	30
6. Köszönetnyilvánítás	32
7. Abstract	33
8. Irodalomjegyzék	35
9. Mellékletek	37

1. Bevezetés

A dolgozat alapjául a „Conifers Around the World” (Debreczy - Rácz, 2001) című kétkötetes botanikai mű szolgál. A könyvek célja a világ fenyőféléinek bemutatása leírásokkal, fényképekkel, botanikai illusztrációkkal, elterjedési térképekkel, és egyéb mellékletekkel. A könyvek nagymértékű terepi munkán alapszanak, aminek jelentős részét egy korábbi alkotás, a „Fenyők a Föld Körül” (Debreczy – Rácz, 2000) készítése során tették meg a szerzők. Az újabb könyvek gyakorlatilag a „Fenyők a Föld Körül” kibővített, angol nyelvű kiadása. A projekt célja egy gazdagon illusztrált és átfogó dendrológiai atlasz létrehozása (Debreczy - Rácz, 2001).

A könyvek célja a világ nagymértékű biológiai sokféleségének (biodiverzitásának) bemutatása oly módon, hogy az széles közönség számára is érthető legyen. Az elsődleges szempont a fajok természetes élőhelyén történő bemutatása. A mű két kötetben, földrajzi szemszögből felosztva jeleníti meg a fajokat. A fák leírásának jelentős eszközei az elterjedési térképek (1. ábra). Az areálgeográfiai térképek a fajok földrajzi elterjedését mutatják be, melynek alapadatként a szerzők terepi megfigyelései szolgálnak. A szerzők a jelenlegi térképek minőségének javítását, létrehozásuk leegyszerűsítését kívánják elérni a dolgozat eredményeként.



1. ábra: Példák a könyvben szereplő elterjedési térképekre.

Az elterjedési térképek képezik a dolgozat alapját, melyeket alapanyagként használok fel. A dolgozatom célja a térképek alapján létrehozott, megfelelő térinformatikai adatbázis szerkezetének megtervezése; az adatbázis feltöltése a különböző fajok elterjedési poligonjainak digitalizálásával, a megfelelő attribútum adatok megadásával; esztétikus háttértérkép szolgáltatása — mindezt egy interaktív webes felületen megjelenítve.

2. Az alapanyag feldolgozása

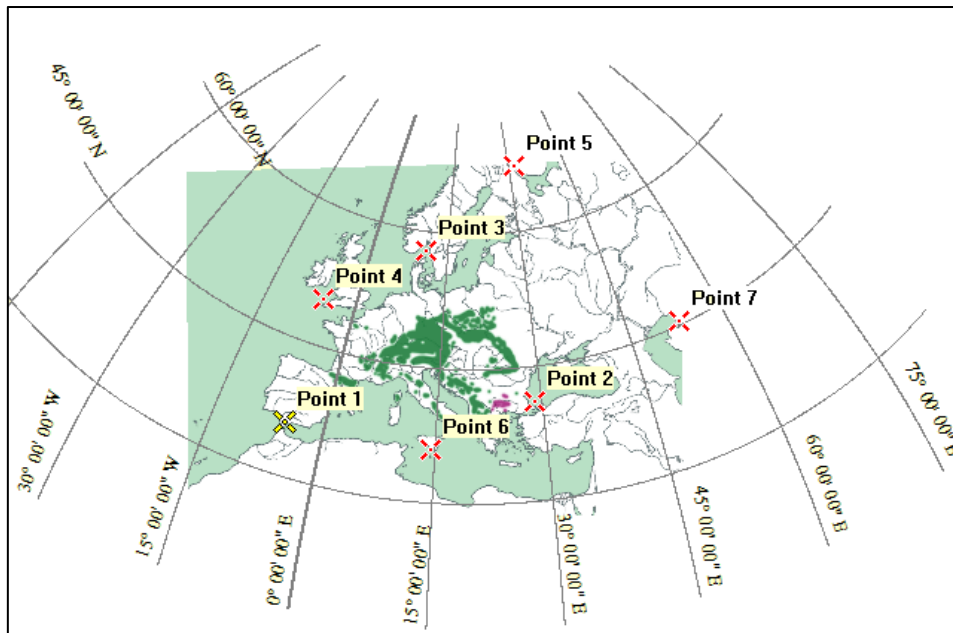
A könyvekben található elterjedési térképeket tekintem alapanyagnak. A térképek digitálisan rajzolt képek, raszteres állományok. A raszteres képek hézag és átfedés nélkül fedik le a pixelek, azaz képpontok által meghatározott területet (Elek, 2011). A kép részeihez nem lehetséges adatbázisba rendezett leíró adatokat társítani, továbbá a fajokhoz tartozó felületeket sem lehet hatékonyan elkülöníteni a háttértérképtől.

Ahhoz, hogy célnak megfelelően használjam ezeket a képeket, raszteresből vektoros képfórmátumot kell létrehoznom. A vektorizálás első lépése a térképek georeferálása, földrajzi térben való elhelyezése, a helyes vetület meghatározásával.

2.1 Georeferencia

Egy raszteres kép képpontjainak csak a kép síkkordináta-rendszerében vannak érvényes pixelkoordinátái. Georeferálás során ezekhez illesztőpontokat rendelünk, melyek célja az, hogy a képpontokhoz térképi koordinátákat csatoljanak a térkép vetületének koordinátarendszerében (Tímár, 2008). Az alaptérképeken alkalmazott ismeretlen vetület azonosítására a térkép fókálózat vonalainak, jellegének vizsgálata ad legmegbízhatóbb alapot (Érdi-Krausz, 1958). Jelen esetben, fókálózat hiányában pusztán a földrészek alakjából lehetséges megállapítani a vetületet. Mivel az Európát ábrázoló térképekkel kezdtem a vektorizálást, így egy azt ábrázoló WMS (Web Map Service) réteget betöltöttem Global Mapperben, majd jól meghatározható pontoknak megadtam a földrajzi koordinátáit (2. ábra). Ismételt próbálkozások után arra az eredményre jutottam, hogy a Lambert-féle területtartó kúpvetület egyedi attribútumokkal a leginkább megfelelő (kezdőmeridián 20° , kezdő szélességi kör 50°).

Global Mapperből georeferált raszteres képként exportáltam a térképeket. A munkát egy másik, rajzolásra és adatbázis kezelésre alkalmasabb geoinformatikai szoftverben, QGIS-ben folytattam. Ezzel a programmal kellett végrehajtanom a fák kiterjedési poligonjainak vektorizálását, illetve az adatbázis megalkotását.



2. ábra: Georeferencia Global Mapperben.

2.2 Vektorizálás

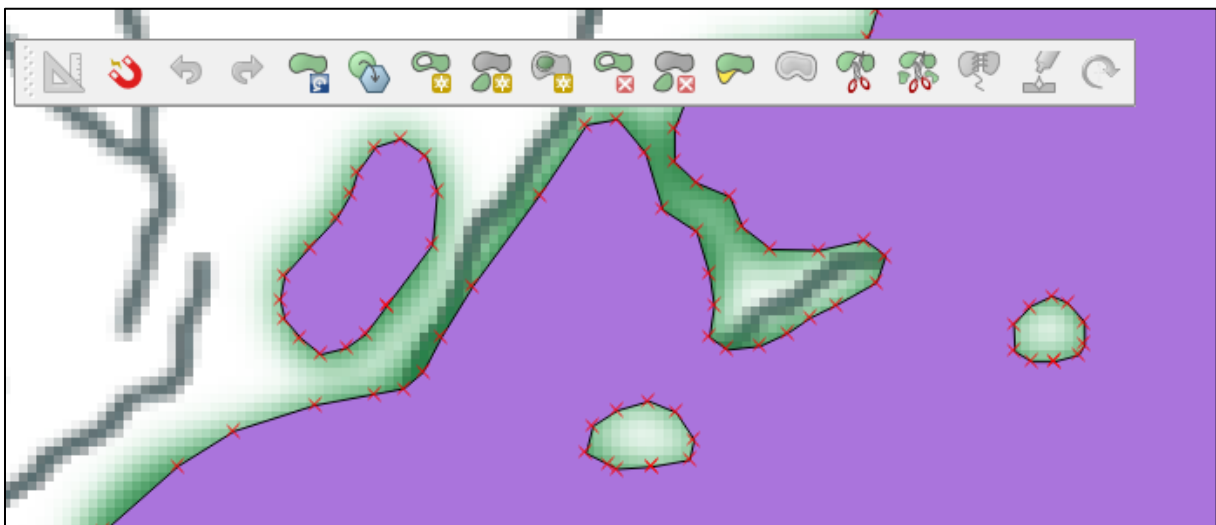
Az alapanyagot raszteres képek alkotják, így az alkalmazható adatfeldolgozási módszerek jelentős mértékben korlátozottak, csak olyan eszközöket lenne lehetséges használni, amelyek a célnak nem felelnek meg, hiszen nem tartozik hozzájuk adatbázis, illetve a képpontokból álló megjelenítés sem lenne megfelelő interaktív térképes alkalmazáson keresztüli bemutatáshoz. Mivel arra van szükségünk, hogy relációs adatbáziskezelőkkel foglalkozzunk a fajok bemutató képekkel, ezért vektorizálni kell a fajterjedési poligonjait. A vektoros adatmodell előnye az is, hogy a különböző grafikus elemekhez (pontok, vonalak, poligonok) tartoznak leíró adatok is (Elek, 2011). Ennek tárolására kiválóan alkalmas az úgynevezett shape fájl.

A shape fájl az ESRI (Environmental Systems Research Institute) által kifejlesztett geoinformatikai fájlformátum. Lényegében nem egy, hanem minimum három különböző kiterjesztésű fájlból áll, melyek külön tárolják a geometriát és a leíró adatokat, viszont nem képes tárolni a különböző stílusbeállításokat. (ESRI, 1998). Habár az ESRI fejlesztette ki, a jelenleg használatban lévő legtöbb GIS szoftver is tudja kezelni.

A munka során kézi vektorizálást alkalmaztam, ami azt jelenti, hogy a poligonokat egyesével átrajzoltam a raszteres képek alapján. A Global Mapperből exportált térképeket QGIS-ben raszteres réteggként megnyitottam; ezek a képek biztosítanak alaptérképet a

rajzoláshoz. Létrehoztam egy új shape fájl réteget, aminek típusa poligon réteg (így csak poligonokat tartalmazhat), illetve tartalmaz egy, a fafajok rendszertani osztályozását tartalmazó adatbázist. A réteg létrehozása után a rajzi eszközökkel megrajzoltam a felületeket.

Többször előfordult, hogy bizonyos fajoknál az elterjedési felületet nem egy poligonnal rajzolták meg, vagy lyuk található a felületen. Ehhez igazodva használnom kellett a haladó rajzi eszközöket is, amik lehetőséget nyújtanak ezen problémák kiküszöbölésére (3. ábra).

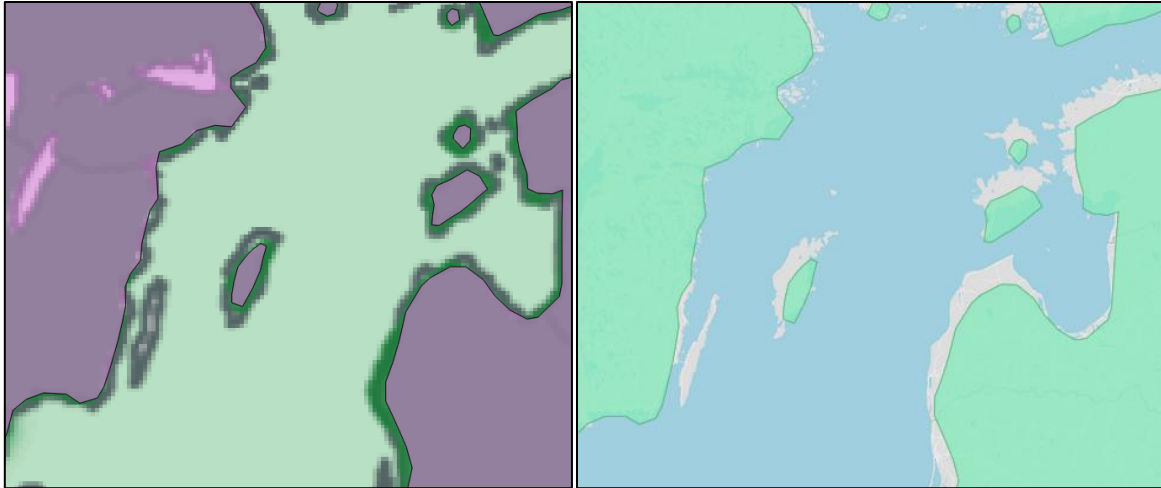


3. ábra: Haladó rajzi eszközök; példa lyukas felületre és több elemből álló felületre, azaz multipolygonra.

Multipolygonok (több, különálló felületből összeadódó területesség) létrehozásának alternatív módszere az, hogy a „geoprocessing” eszközök között található összevonás eszköz alkalmazásával valamelyik oszlop azonos értékei alapján a program létrehoz egy vagy több új felületet.

Egy másik felmerült probléma az, hogy az alaptérkép pontatlansága miatt néhány felület más háttértérkép alkalmazása esetében lelóg a vízfelületre (4. ábra).

E probléma kiküszöbölésére a legkézenfekvőbb megoldás, ha egyszerűen egy ablakban (a későbbiekben ismertetett weboldalon) nézve a felületeket egy másik ablakban a térinformatikai szoftver kezelőfelületén keresztül szerkesztem a poligonokat. Sajnos ezzel a módszerrel nem tökéletes a rajzolat, de pontosabb megoldás hiányában, és azt figyelembe véve, hogy az eredeti térképeken is csak vázlatos a felületek kiterjedése, ez egy praktikus és gyors megoldásnak bizonyult.

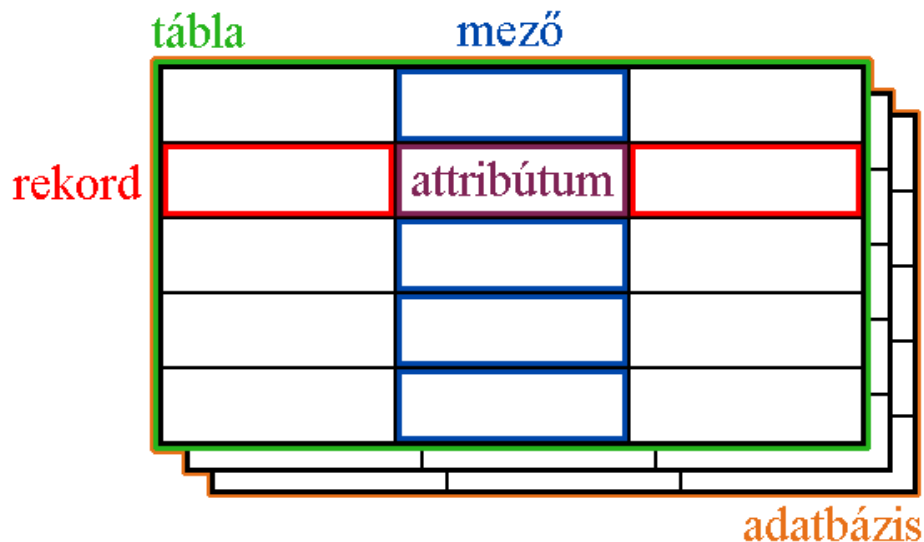


4. ábra: A háttérkép eltéréséből adódó különbség. Bal oldali a könyvekben, jobb oldali a létrehozott weboldalon található térképről.

2.3 Adatbázis felépítése

Az adatbázis lényegében adatok rendezett gyűjteménye. Egy adatbázis táblákból épül fel: léteznek egytáblás adatbázisok, illetve több táblából álló, úgynevezett relációs adatbázisok is. Egy tábla sorait rekordoknak, oszlopait mezőknek nevezzük, ezek keresztmetszetét pedig attribútumoknak (5. ábra). Térinformatikában előforduló megnevezés még tábla esetében réteg, rekordnál entitás-előfordulás, mezőnél attribútum osztály. Minden táblában létre kell hozni egy elsődleges kulcsot, ami egyértelműen azonosít minden sort (Elek, 2011).

A fenyőfákat tartalmazó adatbázisban mindenképp szükség van egy adattáblára, amely tárolja a fajok nevét, rendszertani besorolását, ezáltal későbbiekben kereshetővé válnak. Elsődleges kulcsként az `id` mező áll az adatbázisban. Megadtam mezőként a fajokhoz tartozó törzset, osztályt, rendet, családot, nemzetséget, fajt. Egyes fák esetében továbbá meg kell adni egy újabb mezőt, mely tárolja a különböző alfajokat, változatokat, vagy formákat.



5. ábra: Az adatbázisok általános felépítése.

2.3.1 Rendszertani kategóriák

Az adatbázis kialakításának bővebb megértéséhez érdemes biológiai kitekintést végeznünk a fajok rendszertani beosztásával kapcsolatban.

A bioszféra sok diszkrét, egymástól elkülönülő életformából épül fel. Tudósok régóta próbálják elevezni, osztályozni, illetve csoportokba sorolni a növényeket, állatokat, gombákat és egyéb élőlényeket. A történelem során kifejlesztettek több szabványosított rendszert, melyek célja, hogy a fajok nevei különböző országbeli emberek számára is közérthetőek legyenek. A latin nevek és latin nyelvű rövid leírások rendszerének kialakítója Carl Linnaeus, svéd botanikus, aki létrehozta a biológiában mind a mai napig használatos kettős nevezéktan rendszerét (melyben egy faj neve első tagként tartalmazza a nemzetséget nagybetűvel, és a fajnevet, kisbetűvel). Latin nyelvű tudományos nevek már korábban léteztek, de a *Species Plantarum* (Linnaeus, 1753) című műnek köszönhető széleskörű elterjedésük. A növényeknek egyedi, egyszerű nevük lett. Bevezetésre került egyfajta hierarchikus rendszer is. Az osztályozás tudományával a taxonómia foglalkozik, osztályozás alapegysége a taxon, azaz rendszertani kategória. Az osztályozás legtöbbször méréseken alapszik, de előfordulhat, hogy szubjektív megközelítéssel történik (Debreczy - Rác, 2001).

A „Conifers Around the World” által leírt hagyományos rendszertani kategóriák megnevezését az alábbi táblázat mutatja be:

Latinul	Magyarul	Megjegyzés
regnum	ország	
divisio/phylum	törzs	divisio növényeknél és gombáknál, phylum állatoknál
classis	osztály	
ordo	rend	
familia	család	
genus	nemzetség/nem	nemzetség növényeknél és gombáknál, nem állatoknál
sectio	fajcsoport	csak növényeknél létező kategória
species	faj	
subspecies	alfaj	
varietas	változat	
forma	alak	

1. táblázat: A hagyományos rendszertani kategóriák bemutatása.

A faj alatti kategóriák megkülönböztetése koránt sem egyértelmű. Az alfaj a növénytanban hagyományosan a földrajzilag és részben morfológiailag elkülönült fák csoportját jelöli a fajon belül. A változatot általában ezzel megegyező jelentéstartalommal használják, de jelölhet egy vagy több alaktani tulajdonság alapján elválasztott csoportot (pl. levél hossza, színe alapján). Az alak kategóriát akkor alkalmazzák, ha csak egy morfológiai különbség fedezhető fel, és ez a faj földrajzi kiterjedésén belül elszórtan jelenik meg. Ezen kívül abban az esetben használható, ha kertészeti- vagy mezőgazdasági felhasználásra van kiválogatva egy típus, akár vad élőhelyén található, akár termesztett növényről van szó, illetve természetes benépesítés során találkozunk velük (Debreczy - Rác, 2001).

A hagyományos rendszertani kategóriáktól eltér a modern kladsztika, melyet a 20. században fejlesztett ki Willi Henning német biológus. A módszer filogenetikán, azaz fejlődéstörténeten alapszik. A kladsztika leszármazási viszonyokat kíván felderíteni és fagráfokként ábrázolni, illetve a legfiatalabb közös ősök felkutatása is célja (Podani, 2014).

3. Megjelenítés eszközei

Az egész mű végeredménye egy honlap, ami bemutatja a vektorizált fafajokat. Fontos szempont a fajok kereshetősége. A végcél az, hogy a megjelenített kép exportálható legyen ahhoz, hogy könyvi illusztrációként szerepelhessen. Egy, az OpenLayers eszközeit felhasználó weblapot készítettem. A QGIS-ben megalkotott shape fájlok megjelenítéséhez MapSertvert alkalmaztam.

3.1 Egy weblap általános felépítése

Egy weblap létrehozásakor több különböző technológiát használata lehetséges (W3C, 2016). Különböző standardok léteznek webes alkalmazások építésére, tervezésére és futtatására, mint például HTML, CSS, SVG, JavaScript, különböző API-k (alkalmazásprogramozási felületek), stb. Ezek kliens oldalon, azaz a felhasználó böngészőjében futnak le; ettől elkülönítjük a szerver oldalon futó szkript nyelveket, amelyek alkalmazásával bonyolultabb alkalmazásokat lehet kiépíteni; dolgozatban nem használtam ezt a technológiát.

A dolgozat során alkalmazott fájlformátumok és programozási nyelvek, melyek segítségével létrehozható egy működő, térképes alkalmazást tartalmazó weboldal (Hazard, 2011):

- HTML, mely meghatározza egy weboldal struktúráját és tartalmát,
- CSS, mely segítségével irányítjuk a weblap megjelenítését,
- JavaScript, mely az oldalak logikáját, a felhasználó interakcióit írja le.

A HTML egy olyan szabvány, amely szerkezetet ad a megjelenítendő tartalmi elemek sokaságának, azt szabályozza, hogy a különböző weblapokat miként kell megírni ahhoz, hogy a számítógépek megértsék és a böngészők megfelelő módon jelenítsék meg azokat (DeBolt, 2005). Pusztán HTML használatakor a honlap tartalmát stílusbeállítások nélkül tölti be a böngésző, külalak formázásához szükséges CSS stíluslapok írása is, ami vizuális megjelenési tulajdonságokkal látja el a leírókódon belüli elemeket (6. ábra). A két technológia együttes használata minden weblap magját képezi.



6. ábra Ugyanaz a tartalom CSS nélkül és CSS-sel.

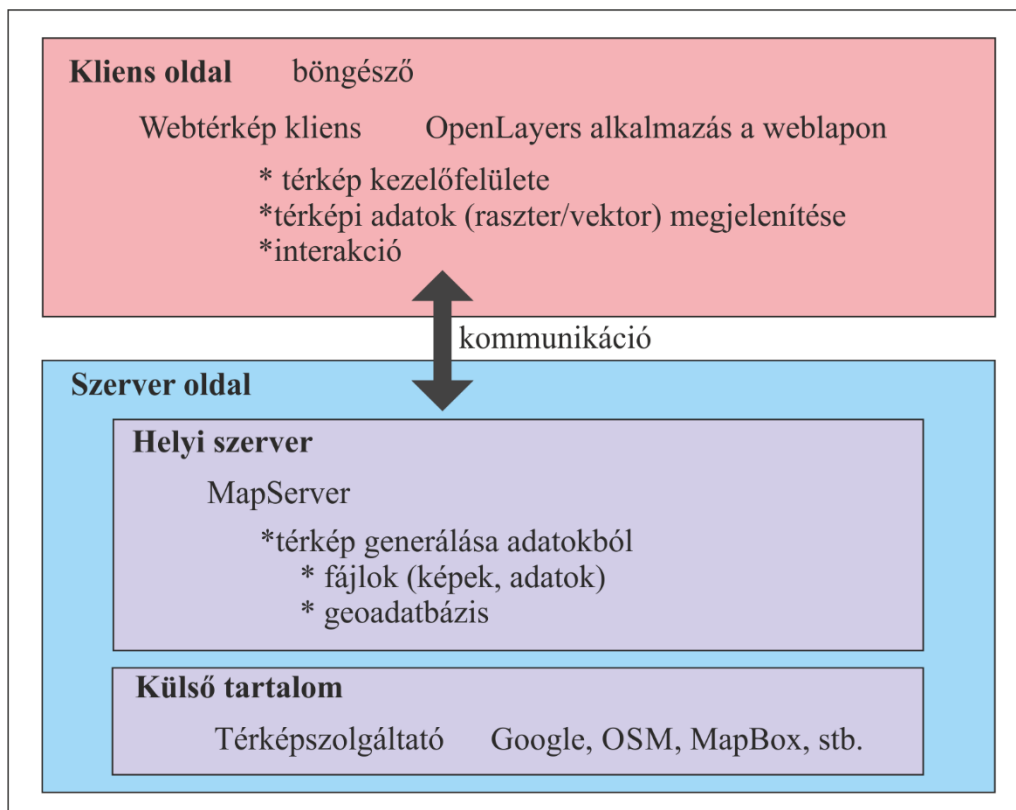
JavaScript alkalmazásával érhetjük el, hogy egy statikus képet mutató weblap dinamikussá váljon, válaszoljon felhasználói bemenetre, esetleg mást jelenítsen meg napszaktól, vagy egyéb változótól függően (Brown, 2016). E nyelven íródott kód végrehajtódik vagy a böngészőbe való betöltéskor, vagy valamilyen eseményre válaszként; a változások a weblap újra betöltése nélkül mennek végbe.

3.2 OpenLayers

Az OpenLayers egy nyílt forráskódú, kliens oldali JavaScript könyvtár, melynek legfontosabb szerepe térképek megjelenítése és kezelése weblapokon (Hazzard, 2011). Központi eleme az interaktív térkép. A térkép az információmegjelenítés eszköze, mely végtelen számú raszteres és vektoros réteget is tartalmazhat. Ezen kívül a térképen szerepelhetnek úgynevezett kontrol eszközök, amikkel interakcióba léphetünk vele (Perez, 2012).

A könyvtár azt jelenti, hogy egy előre megírt eszköztár függvényeit felhasználva hozzuk létre a honlapot, nem szükséges mindent magunknak megírni. A kliens oldalúság azt jelenti, hogy nincs szükség szerver oldali szoftverre, a munkát a felhasználó böngészője végzi.

Az elsődleges művelet, amit a kliens oldal végrehajt, az kérés egy térkép szerver/szolgáltató felé azzal kapcsolatban, hogy mit jelenítsen meg, mi az, amit a felhasználó látni szeretne (7. ábra). Több különböző térkép szolgáltató létezik, amik biztosítják az adatokat, ilyenek pl. Google Maps, OpenStreet Map, WMS. Az OpenLayers ezeket használja fel megjelenítés során (Hazard, 2011).

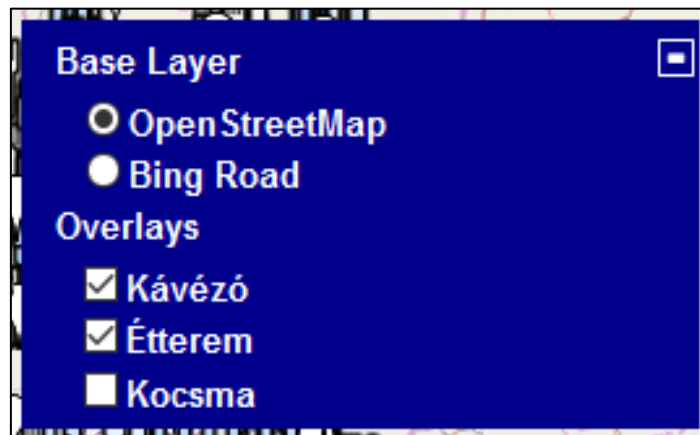


7. ábra: Szerver-kliens modell a dolgozat példáján bemutatva (Gede, 2016).

Ahhoz, hogy megfelelően funkcionáló webtérképet állítsak elő, mindenképp szükség van egy háttértérképet tartalmazó alaprétegre. Ez lehet a már korábban említett térképszervereken keresztül szolgáltatott térkép. Nem csak egy alapréteg lehet az alkalmazásban, de csak egy lehet aktív (8. ábra). A másik alapvető rétegtípus az úgynevezett fedőrétegek, amik mindig az alaprétegen felül mutatkoznak. A fedőrétegeket szabadon lehet ki-bekapcsolni, egyszerre több is lehet aktív, a programkódtól függ, hogy mi a sorrendjük (Hazard, 2011).

A dolgozathoz készített honlapon a fafajok poligonjait nem lehetséges vektoros fedőréteggként használni, mivel az OpenLayers a dolgozat készítésekor nem képes shape fájlokat vektoros képként kezelni. Emiatt szükségem van egy olyan megoldásra, amivel a fájlokat alternatív módon hozzá tudom adni a térképes alkalmazáshoz. E problémakor

megoldására megfelelően alkalmas az úgynevezett MapServer, mely szolgáltatást a következő alfejezetben ismertetek.



8. ábra: Rétegkezelő ablak, alapréteg (Base Layer), fedőrétegek (Overlays).

3.3 MapServer

A MapServer egy nyílt forráskódú programcsomag, melyet térinformatikai, illetve webes térképészeti felhasználásra fejlesztettek ki. Térképet hoz létre térbeli információkból, melyeken megjelenítésen kívül lekérdezéseket hajthatunk végre. Bemeneti adatként szerepelhet raszteres és vektoros adat, köztük az OpenLayers által fel nem használható shape fájlok is. Kimeneti formátum lehet szintén raszteres és vektoros kép, pl. png, svg, kml, stb. Az alkalmazásához egy map kiterjesztésű szöveges fájlt, úgynevezett mapfile-t hozunk létre; ez írja le, hogy a különböző adatforrások tartalmát hogyan ábrázolja a generált térképen az OpenLayers, illetve lekérdezéseket írhatunk a fájlba, ezzel meghatározva, hogy különböző rétegek milyen adatokat mutassanak. A rétegekhez különböző megjelenítési stílust állíthatunk be, ami szintén függhet lekérdezésektől. A rétegek tartalmazzák a bemeneti fájlt. Lehetőség van adatmező alapján osztályozásra a rétegeken belül, ekkor rétegekhez hasonlóan stílusbeállításokkal különböztetjük meg az osztályokat. Osztályok megjelenítése lehet méretarányfüggő (Kropla, 2005).

A MapServer által szolgáltatott térképek megjelenítésére lehetőség van interaktív kliens-oldali megjelenítő használatára, amire példa az OpenLayers. Erre a legjobb megoldás az, ha WMS (Web Map Service) protokollt használjuk a MapServerrel való kommunikációhoz (Gede, 2016).

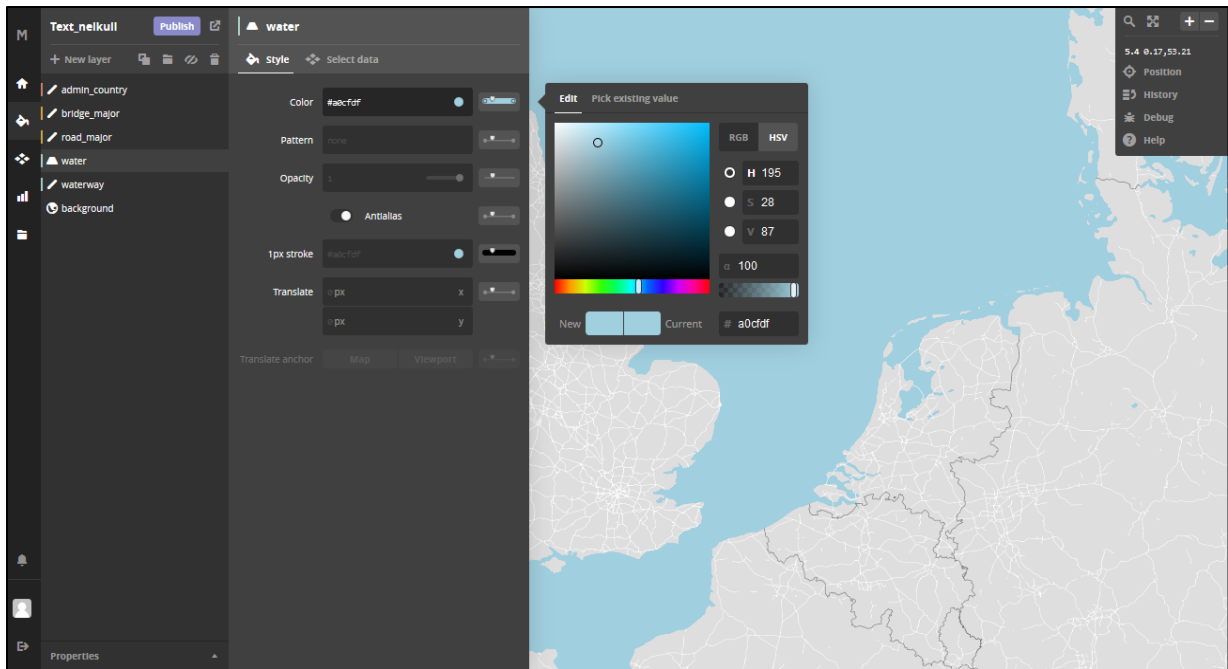
A dolgozat szempontjából a szolgáltatás lényege az, hogy a shape fájlokat oly módon jelenítse meg az OpenLayers alapú térképalkalmazás, hogy az tartalmazzon előre meghatározott stílusbeállításokat, és ezen felül a fájlhoz tartozó adatbázist is tárolja. Mivel interakciókat hajjt végre a felhasználó a réteggel, ezért nem lehetséges pusztán raszteres képként ábrázolni, és erre kínál ideális megoldást a MapServer. A kiindulási shape fájlt alakítja át WMS réteggként megjeleníthető képpé.

3.4 MapBox

A különböző térképszolgáltatók általában csupán néhány különböző térképtípust kínálnak fel felhasználásra, emellett sokszor nincs lehetőség a térképi tartalom befolyásolására. Erre kínál megoldást a MapBox nevű térképszolgáltató. Egyedi, személyre szabott térképeket tudunk létrehozni, majd implementálni a webes térképalkalmazásunkba (MapBox, 2016).

Ezt felhasználva létrehoztam egy, a dendrológiai atlaszokban lévő térképek megjelenítéséhez igazodó háttértérképet. Erre a MapBox hivatalos honlapján található, ingyenes regisztráció után bárki számára elérhető Studio menün keresztül van lehetőség. A Studio lényegében egy térképszerkesztő applikáció. Több különböző szerkesztési opciót kínál fel, jelenleg elérhetőek többek között feltöltött vektoros adatok alapján térképcsempék generálása, vagy a dolgozat során alkalmazott egyedi stílusok alkotása.

A térképstílus szerkesztő program használatakor első lépésként ki kell választani egy kiindulási állapotot, ami lehet egy előre létrehozott alaptérkép, vagy egy teljesen üres munkafelületet. Az előre definiált stílusok közt van úthálózatot bemutató vektortérkép, névrajz nélküli műholdfelvétel, domborzatrajz. A leginkább megfelelő kiválasztása után átirányít egy kezelőfelületre, ahol a térképrétegek megjelenítési beállításai szabályozhatóak (9. ábra).



9. ábra: MapBox térképszerkesztő kezelőfelülete.

A MapBox adatbázisából leválogatható, hogy milyen adatok jelenjenek meg, és ezekhez stílusbeállítások társíthatóak. A stílusbeállítások sokrétűségét, mely egy valódi, asztali geoinformatikai program széleskörű lehetőségeit eleveníti fel, a 10. és 11. ábra segítségével mutatom be.



10. ábra: Az egyik MapBox által felkínált alaptérkép.

Mint ahogy az ábrákon látható, a beállítási lehetőségek valóban sok mindenre kiterjedőek. A 11. ábrával szemléltetem, hogy változtatható a névrajzi elemek színe, betűtípusa, elhelyezése és az adatbázisból kiválasztható több nyelven elérhető felirat. Jelenleg magyar nyelvű nem elérhető, viszont feltölthető egyedi adatsor, amivel lehetséges létrehozni ezt is. Minden megjelenített elem színe és átlátszósága módosítható.



11. ábra: A MapBox Studio alkalmazásával létrehozott egyedi térkép.

A dolgozathoz tartozó térkép szerkesztése folyamán, az atlaszokban fellelhető térképek stílusához igazodó térkép létrehozásában megjelenítési szempontok a következők (12. ábra):

- szárazföld szürke kitöltéssel,
- vízfelület kék kitöltéssel,
- országhatár sötétszürke vonalrajzzal,
- bizonyos méretarány alatt fontosabb utak és hidak fehér vonalrajzzal,
- felirat ne szerepeljen.

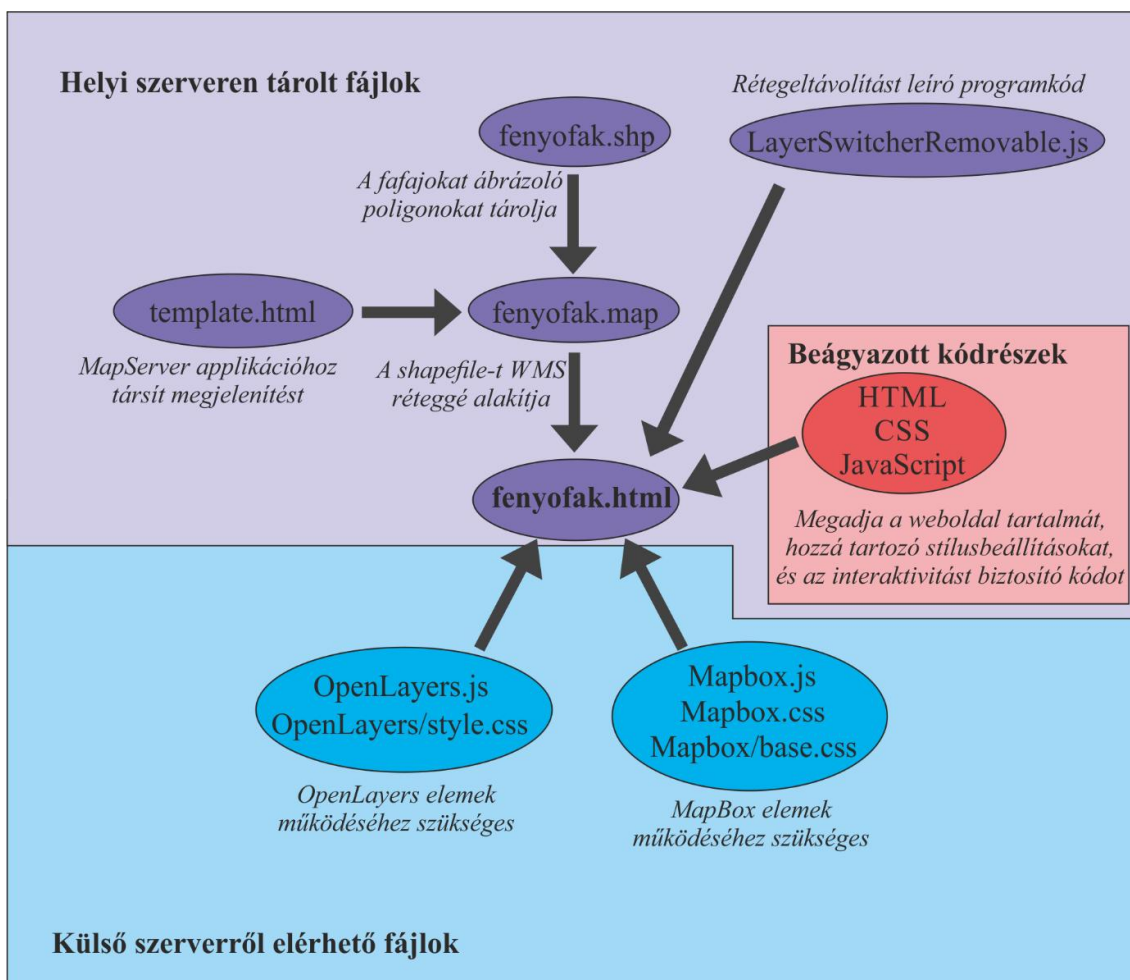


12. ábra: A MapBox-szal megalkotott háttérkép.

Mentés után később is bármikor szabadon szerkeszthető. A térképeket egyedi azonosítójukkal felhasználhatjuk OpenLayers rétegeként, mind alapréteggént, mind fedőréteggént. A dolgozat folyamán készített honlapon alapréteggént szerepel a fent bemutatott háttérkép (12. ábra). Mivel a térképek a MapBox szerverén tárolódnak, ezért ha későbbiekben módosítanám a térkép stílusbeállításait, a honlapon az automatikusan frissülne, és az újabb változatot töltené be.

4. A weblap ismertetése

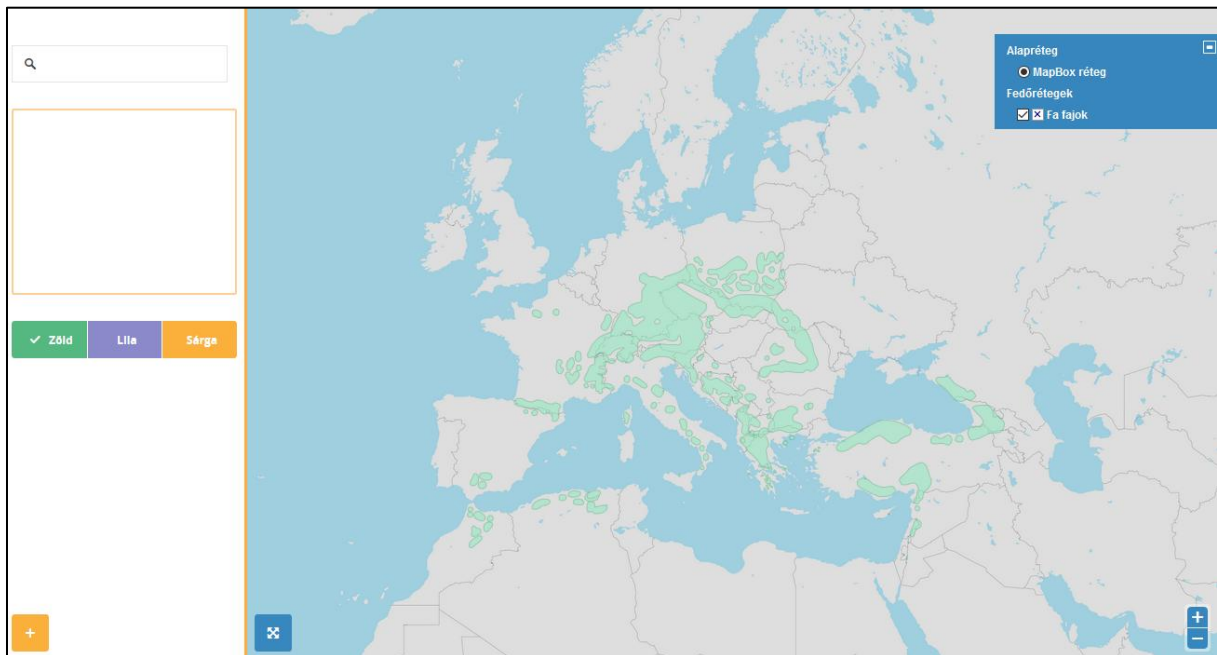
A dolgozat során készített honlap fő funkciója a dendrológiai atlaszokban fellelhető elterjedési térképek digitális reprezentációja. A webhely legfontosabb eleme az OpenLayers segítségével létrehozott interaktív térkép, hiszen ez a térképes adatok vizualizációjának eszköze. Fontossága miatt a térkép az oldalt csaknem teljes egészében kitölti, pusztán egy menüsor szerepel rajta kívül. A térkép beállításai között szerepel az, hogy a vetülete az úgynevezett Web Mercator (Battersby et al., 2014), avagy gömbi Mercator legyen, ami megegyezik a különböző webtérképek által használt vetülettel, illetve ezen kívül az, hogy mivel a dolgozat az európai fafajokra koncentrál, betöltés során a térkép középpontja úgy van meghatározva, hogy Európa területét mutassa. Ezt később a programkódban lehetséges módosítani. Több fájl csatlakozik egymáshoz, amelyekhez külső, interneten elérhető könyvtárak is tartoznak (13. ábra).



12. ábra: A fájlok közti összefüggés ismertetése.

4.1 A weblap tartalmi elemei

A weboldal központi és legfontosabb eleme az OpenLayers alkalmazásával létrehozott térkép, mely bemutatja a fafajok elterjedési poligonjait. A térképes elem területén található további OpenLayers kiegészítő elemek közül pár, mint például a rétegkezelő eszköz. A térkép mellett van egy, a felhasználói interakciókat vezérlő menü (14. ábra).



14. ábra: A weblap kinézete (2016.05.25.).

A menü olyan HTML elemeket tartalmaz, amelyekhez tartozik egy-egy JavaScript függvény, ami meghatározza a felhasználói interaktivitást. Mindez kiterjed többek között egy keresődobozra (15. ábra), mely lényegében egy űrlap elem. Ez lehetőséget nyújt a fafajokra való keresésre.



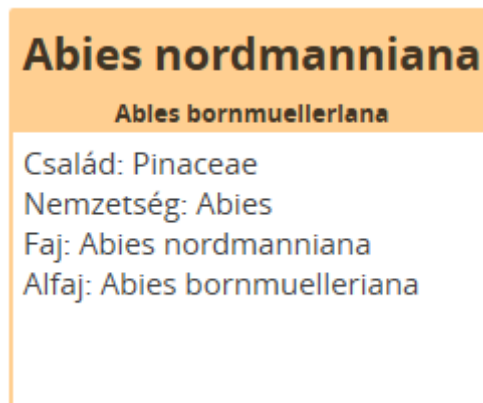
15. ábra: A keresődoboz.

Szerepel egy olyan kezelőfelület, amely segítségével lehetséges változtatni a térképen megjelenített poligonok színén. Három előre definiált szín érhető el, zöld, lila és sárga. Ezt lényegében három választási lehetőséget felkínáló rádiógomb segítségével hoztam létre, melynek nem látszanak a „kör részei” (16. ábra), így a megjelenített gombokra kattintva lehet választani közülük.



16. ábra: A poligonok színét módosító gombok, jobb oldalon a rádiógombokhoz tartozó körök láthatóak.

Készítettem egy gombot, ami segítségével a térképes alkalmazás tartalmát teljes képernyőre lehet küldeni. Van egy gomb, amivel a megjelenített poligonokat új fedőréteggént el lehet tárolni, így egyszerre több keresési- és színbeállítással mutatott terület is látható a térképen. Ezekon kívül még van egy szövegdoboz, ami a felületekhez tartozó attribútum adatokat írja ki, egyik poligonra való kattintás után (17. ábra).



17. ábra: A fajokhoz tartozó leírás.

4.2 Stílusbeállítások

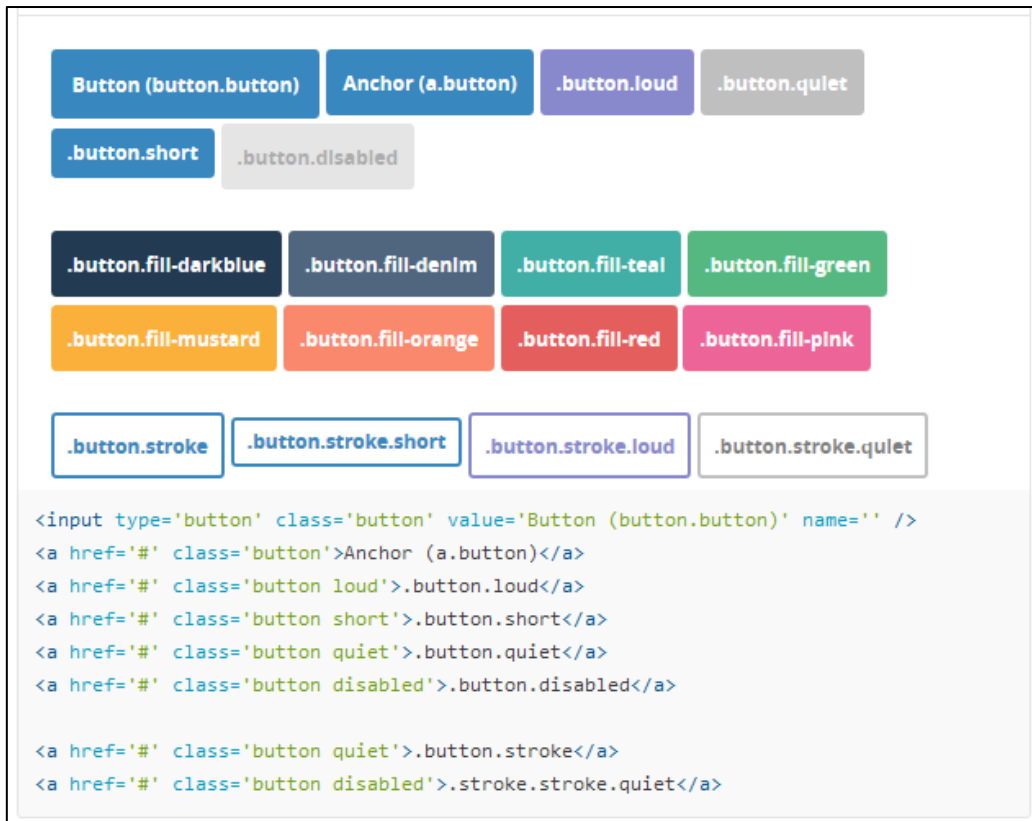
A HTML elemekhez stílusbeállításokat rendelni CSS szabályok írásával lehetséges. Mivel egy db weboldal tartalmazza a térképes alkalmazást, egyszerűség kedvéért a HTML elemeket leíró oldalon írtam meg a stílus szabályokat, `<style>` tagok közé illesztve. Ha több oldallal dolgoztam volna, akkor célszerűbb lett volna külön CSS dokumentumot írni és

azt csatolni a weboldalhoz, hiszen ebben az esetben, ha módosítanám a stílust, akkor az összes oldalon egy időben változna meg (DeBolt, 2005).

A menüsor elemeinek stílusbeállításai a MapBox síluslap alapján készítettem el. Bizonyos elemek esetében elegendő előre megírt MapBox osztályt hozzáadni az elem deklarációsakor, és így hozzárendeli a megfelelő stílust. A stílusok nevei és megjelenítése a szolgáltató hivatalos honlapján érhetőek el (18. ábra). Előre megadott stílusok közül válogathatunk jelenleg az alábbi elemek esetében:

- koordinátaháló,
- szöveg blokkok,
- űrlapok,
- táblázatok,
- gombok,
- színek,
- ikonok,
- kisegítő elemek,
- mobilra való optimalizáció eszközei.

Az OpenLayers eszközök stílusát is módosítottam, alapbeállítástól eltérő színbeállítás formájában. Ezt úgy tudtam megtenni, hogy a kiválasztott elem osztályához írtam új CSS szabályt, mivel a generált `id`-je változhat (Hazzard, 2011). A célból, hogy az `id`-hez tartozó szabályok ne írják felül a beállításaimat, hozzátettem az `!important` CSS utasítást, hisz az osztályhoz tartozó szabályok alacsonyabb rendűnek számítanak, mint az `id` (DeBolt, 2005).



18. ábra: Előre megírt MapBox stílusszabályok gombokhoz osztályokkal, vizualizáció és meghívás módja.

Ezekon kívül CSS stílusszabályokkal állítottam be azt, hogy teljes képernyős módban az OpenLayers térképfelületen megjelenített irányító eszközei ne legyenek láthatóak. Más módon, szkriptekkel is meg lehetett volna oldani ezt a problémát, de ezzel a módszerrel igen egyszerűen megoldható a probléma, csak pár sor szükséges hozzá a kódban.

4.3 Az OpenLayers eszközök

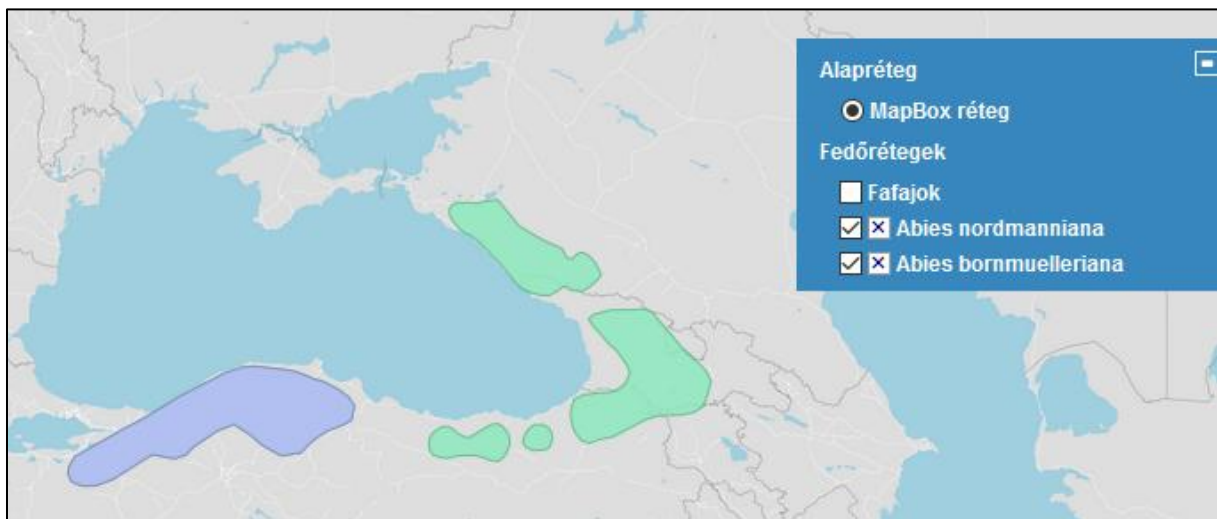
A térképen szükség van legalább egy alaprétegre, ami jelen esetben a már korábban ismertetett MapBox felhasználásával megalkotott háttértérkép. Importálásához meg kellett adnom a MapBox által kiosztott egyedi elérési azonosítót, illetve a térképstílushoz tartozó azonosító kódot is.

Fedőréteggént szerepel egy réteg, ami tartalmazza az összes, geoinformatikai szoftverrel megrajzolt fafaj elterjedési poligonjait. Mint már korábban volt róla szó, mivel ezek shape fájlként lettek elmentve, az OpenLayers jelenlegi változata nem képes behívni ezt a fájl típust megjeleníthető térképréteggént, ezért megoldásként a MapServer alkalmazásával

létrehoztam egy WMS réteget, amit már kezelni tud. Megadtam a létrehozott mapfile elérési útvonalát.

A weblapon az OpenLayers irányító eszközei közül a rétegkezelő eszközt használtam fel, aminek több szerepe is van, mint pusztán rétegek listázása, és láthatóságuk állítása. Az extra funkciók JavaScript függvényekként lettek megírva (19. ábra), és azt határozzák meg, hogy egy réteg törölhető legyen hozzáadás után. A feliratok neveit megváltoztattam „Alapréteg”-re és „Fedőrétegek”-re.

Szerepel még egy olyan eszköz is, amivel a WMS rétegből ki lehet nyerni az attribútum adatokat, és azt valamilyen módon kezelni. A dolgozat esetében úgy használtam fel, hogy kattintásra megjelenjenek egy szövegdobozban.



19. ábra: Több különböző fajhoz és alfajhoz tartozó terület egy időben való megjelenítése a weboldalhoz írt elemek és függvények segítségével. A rétegkezelő felületen az X doboz a dolgozat során lett létrehozva (lásd: 4.5 alfejezet).

4.4 A mapfile tartalma

A .map kiterjesztésű fájlban (ami lényegében egy egyszerű szöveges leírás) kell megadni, hogy a shape fájlt milyen szabályokat követve alakítsa át a MapServer a dolgozat céljának megfelelően. Kötelezően meg kell adni bizonyos szabályokat a kimeneti fájljal kapcsolatban, amit a szerver küld vissza, mint például a kép formátuma, ami jelen esetben png kiterjesztésű képfájl. Ez azért fontos, mert ez a fájl típus átlátszóságot is tárol, így nem fed el a kiterjedési poligonokon kívüli területet egy fehér folt (Zentai, 2000). Megadtam továbbá magának a képnek a nevét, méretét pixelben, a térkép földrajzi határait, legnagyobb

lehetséges képméretet, vetületet, shape fájl elérési útvonalát. Mivel WMS réteggént használtam fel a kimeneti képet, ezért hozzá tartozó kötelező paramétereket is megadtam. Ezt az alábbi kódrész írja le:

```
NAME "Fák"
  SIZE 1000 800
  IMAGETYPE png
  EXTENT -20 30 90 80
  MAXSIZE 4096
  SHAPEPATH "shp"
  PROJECTION
    "init=epsg:4326" #wgs84 lon-lat
  END
  WEB
  METADATA
    "wms_enable_request" "*"
    "wms_srs" "EPSG:23700 EPSG:4326 EPSG:3857 EPSG:900913"
    "wms_feature_info_mime_type" "text/html"
  END
END
```

Létrehoztam egy réteget, ami tartalmazza a fafajokat, alapadatként megadtam az ezeket tartalmazó shape fájlt, vetületként ugyanazt, amit korábban georeferálás során:

```
LAYER
  NAME "Fafajok"
  TYPE polygon
  DATA "fenyofak.shp"
  STATUS DEFAULT
  PROJECTION
    "proj=laea"
    "ellps=GRS80"
    "datum=WGS84"
    "lat_0=50"
    "lon_0=20"
  END
```

A honlapon szereplő keresési- és színmódosítási lehetőség alapjait ebben a fájlban kell kiépíteni, hiszen e műveletek során a WMS rétegen szereplő adatok megjelenítését változtatjuk. Megadtam, hogy milyen karaktereket tartalmazhat a keresés, és beállítottam, hogy az adatbázis faj és alfaj oszlopának attribútumai kereshetőek legyenek, amihez a FILTERITEM és FILTER függvények szükségesek, segítségével megadható, hogy valamilyen kritérium szerint szűrje a megjelenített adatokat a rétegen. A kritériumot a felhasználó adja meg a keresőmezőn keresztül. Alapértelmezett keresési beállítás az, hogy a teljes adathalmaz jelenjen meg a WMS rétegen. Ugyanígy meg kell adni a poligonok színbeállításait leíró kódot is. Megadtam egy alapértelmezett színbeállítást, és több további

színt is definiáltam, amik közül a honlapon lehet választani; az alapértelmezett szín a zöld. Ezekon kívül szerepel egy parancs, amivel csatoltam egy úgynevezett template (sablon) fájlt. Ez egy HTML kiterjesztésű fájl, aminek a lényege, hogy külalakot adjon egy MapServer applikáció felületéhez, jelen esetben a WMS információlekérés felületéhez (Kropla, 2005). A keresést és színmódosítást megalapozó kódrészlet lentebb olvasható. Ez a részlet a LAYER kezdetű sortól megismétlődik annyi eltéréssel, hogy a FILTERITEM függvényben faj helyett alfaj a megadott paraméter, és így két külön keresés is lebonyolítható, két külön rétegen kezeli a fajhoz és alfajhoz tartozó elemeket.

```
VALIDATION
    "SEARCH" "[0-9|a-z|A-Z|\.|\\-|\\+]"
    "COLOUR" "[0-9|a-z|A-Z|\.|\\-|\\+]"
END
FILTERITEM faj
FILTER /%SEARCH%/
METADATA
    'default_SEARCH' '.'
    'default_SZIN' 'green'
    "wms_enable_request" "*"
    "wms_include_items" "all"
END
TEMPLATE "template.html"
CLASS
    EXPRESSION ('%COLOUR%'='yellow')
    NAME "yellow"
    STYLE
        OUTLINECOLOR 251 176 59
        COLOR 241 240 117
    END
END #class
CLASS
    EXPRESSION ('%COLOUR%'='purple')
    NAME "purple"
    STYLE
        OUTLINECOLOR 138 138 203
        COLOR 183 183 253
    END
END #class
CLASS
    NAME "default"
    STYLE
        OUTLINECOLOR 86 184 129
        COLOR 136 234 189
    END
END #class
```

A mapfile-nak azon a szerveren kell lennie, amin maga a weboldal is van. Ez azért szükséges, mert a weboldalba beépített OpenLayers alkalmazás küld kérést a MapServer felé, leírva a mapfile-ban, hogy milyen utasítások alapján alakítsa át a shape fájlt WMS rétegen keresztül küldött képpé.

4.5 JavaScript kód

Egy honlap interaktivitása JavaScript kóddal növelhető. Különböző függvényeket írtam a célból, hogy ne csak egy statikus weblapot kapjak a fajok poligonjaival végeredményként, hanem egy interaktív, valóságban is jól használható, modern változatot. Az így létrehozott függvényeket ebben a fejezetben ismertetem.

- `cloneMe` függvény: Keresés és színmódosítás után csak a felületek egy része, valamelyik választható színnel látható. A célból, hogy ezt valami módon eltárolja a weblap, és így egyszerre többféle leszűkítés szerint jelenjenek meg adatok, írtam meg ezt a függvényt. Lényegében az történik meghívásakor, hogy a WMS réteget új fedőréteggént raktározza el az OpenLayers alapú térképalkalmazás. A rétegnek nevet is kell adni egy felugró ablakban, és ezután megjelenik a rétegkezelő irányítóelemben. A függvény lefutásához a hozzá tartozó, pluszjelet mutató gomb megnyomása szükséges. Az ide tartozó kód a következő:

```
function cloneMe() {
    var clone=wms.clone();
    var cloneName=prompt("Rétegnév:", "Fafajok");
    clone.removable=true;
    map.addLayer(clone);
    clone.setName(cloneName); }
```

A `clone()` funkció egy beépített OpenLayers függvény, a `removable` pedig a fejezet végén ismertetett, különálló JavaScript fájlban definiált tulajdonság.

- `searchMe` függvény, `empty` függvény: A honlapon van keresési funkció, amit egy keresődobozal lehet aktiválni. A keresendő szöveg beírása után Enter megnyomására, vagy a nagyító ikonra való kattintásra új WMS kérést hajt végre a program úgy, hogy a szöveget leszűkítési paraméterként használja fel. A keresés alapjai a `map` fájlban vannak megírva. Ehhez tartozik még egy függvény, ami azt határozza meg, hogy az oldal betöltésekor üres legyen a keresődoboz, azaz ne legyen értéke, és ekkor az alapértelmezett keresési eredmény, azaz az összes megjeleníthető poligon látszik a WMS rétegen. A keresést irányító kód az alábbi, ahol a `search_input` azonosítójú elem a keresődoboz, és a `mergeNewParams` egy előre definiált OpenLayers függvény:

```
function searchMe() {
    var searchVal=document.getElementById("search_input").value;
    wms.mergeNewParams({search:searchVal}); }
```

- `recolourPoly` függvény és a hozzá tartozó rádiógombok, `checkedStyle` függvény: Mivel az eredeti atlaszokban több színnel (zöld és lila) jelennek meg a fafajok felületei, ezért a honlapnak is igazodnia kell ehhez. A színek módosításához a WMS réteget is változtatni kell, minden alkalommal új lekérést kell végrehajtani, paraméterek módosítása szükséges. Ez azt is jelenti, hogy a függvény alapjainak a map fájlban kell lennie, és már ebben szükséges definiálni a színekhez tartozó stílusleírásokat, a színek RGB kódját. A definiált színekhez tartozó nevet kell meghívni, és új paraméterként megadni. A nevek meghívását viszont a felhasználó végzi, hiszen neki kell eldöntenie, hogy mit szeretne megjeleníteni, milyen színnel kívánja látni a felületeket. Létrehoztam egy megoldást, amivel könnyedén és egyértelműen el lehet érni ezt. A legszebb és legpraktikusabb módszernek azt gondoltam, ha színes gombok közül választhat a felhasználó, és a kiválasztott gomb megnyomása után a betöltött felületek színe változik. Úgynevezett rádiógombokat hoztam létre, amikkel végrehajtható a felvázolt interakció, amelyeknél több lehetőség közül mindig egy a kiválasztott. A rádiógombok értékét hívja meg a függvény, ami a gombokra való kattintás során hajtódik végre.

```
function recolourPoly() {
    if (document.getElementById("green").checked==true) {
        var colour=document.getElementById("green").value;
        wms.mergeNewParams({colour:colour});
    }
    if (document.getElementById("purple").checked==true) {
        var colour=document.getElementById("purple").value;
        wms.mergeNewParams({colour:colour});
    }
    if (document.getElementById("yellow").checked==true) {
        var colour=document.getElementById("yellow").value;
        wms.mergeNewParams({colour:colour}); } }
```

Ennél a függvénynél a `wms.mergeNewParams({colour:colour});` sor első `colour` tagja, illetve a kereséshez tartozó függvény `search` tagja megegyezik a map fájlban található paraméterekkel.

Ehhez a részhez tartozik még egy függvény, ami a stílust módosítja. A kiválasztott színhez tartozó gomb mindig kék színnel jelent meg, és ennek felülírására készítettem el egy függvényt, ami `id` alapján megváltoztatja a címke háttérszínét úgy, hogy az ne térjen el az alapértelmezett megjelenéstől:

```
function checkedStyle() {
    var labels=document.getElementsByTagName("label");
    for (var i=0; i<labels.length; i++) {
        if (labels[i].htmlFor != "") {
            var elem=document.getElementById(labels[i].htmlFor);
            if (elem)
                elem.label=labels[i]; } }
document.getElementById("green").label.style.background="#56B881";
document.getElementById("purple").label.style.background="#8A8ACB";
document.getElementById("yellow").label.style.background="#FBB03B";}
```

- `fullScreenMode`: A weblapon gombnyomásra lehetőség van teljes képernyős módra állítani a térkép felületét. Jelenleg a teljes képernyős módra váltáshoz szükséges API kísérleti módban érhető el, és a közeljövőben jelentősen megváltozhat a hozzá tartozó függvények szintaktikája (Mozilla Developer Network, 2016). Ha a megfelelő gombra kattint a felhasználó, akkor aktiválódik a teljes képernyős mód, amiből kiléphet `Esc` megnyomásával.

- `LayerSwitcherRemovable.js`: Egy különálló JavaScript fájl, amely lényegében egy kiegészítés az `OpenLayers`-höz. A rétegváltoztató eszköz egy olyan egyedi változatát definiálja, amely a `removable` paramétert tartalmazó rétegek esetében hozzáad egy törlés gombot a réteg neve elé. Az első fedőréteg nem tartalmazza ezt, mivel azon a rétegen folyik a keresés és színmódosítás, de minden hozzáadott réteg törölhető. A törlés gomb megjelenítését is ez a fájl írja le.

5. Összefoglalás

Végeredményként megalkottam a weboldalt, amit terveztem a dolgozat írásának kezdetén, mely alapja egy modern, interaktív, térképes alkalmazás. Alapanyagként a „Conifers Around the World” című atlaszokban található térképeket használtam fel. A raszteres térképeket georeferáltam, a helyes vetületre – földrajzi fókuszálzat hiányában – Európa partvonalából következtettem (e kontinens fafajaira fókuszáltam a dolgozat során). A georeferált térképeket QGIS-ben vektorizáltam, ami azért fontos lépés része a munkának, mert így lehetséges adatbázist társítani a képekhez, és ez szükséges a célból, hogy a weboldalon a keresést lehessen végrehajtani a felületek között. Ezen kívül a vektoros kép jobban használható az olyan típusú webes alkalmazásokon, mint amit létrehoztam. Az adatbázis a növényekhez tartozó rendszertani osztályokat tartalmazza. Ez a térinformatikai háttere a weboldalnak.

A honlap legfontosabb eleme egy OpenLayers alapú térképes alkalmazás. Ezzel egy interaktív felületen lehetséges a felületek ábrázolása valamilyen háttértérképen. A QGIS-ből kinyert shape fájlokat az OpenLayers nem képes önmagában kezelni, ezért alkalmaztam a MapServer által kínált technológiát. Ez lényegében létrehoz egy képfájlt, amit WMS rétegen keresztül lehetséges felhasználni. A háttértérkép igazodik az atlaszokban fellelhetőhöz, és ezt a MapBox nevű szolgáltató segítségével alkottam meg. Egy teljesen egyedi stílusbeállításokkal rendelkező térképet hoztam létre, amely funkciójában hasonlít a különböző térképszolgáltatók által kínáltakkal (például a Google Maps által nyújtott háttértérképek, amelyek alkalmazásakor szerzői jogvédelemmel kapcsolatos problémákba ütközhetünk, ezt küszöböli ki a szabad felhasználású MapBox).

A weboldal interaktivitásának növelésére JavaScript függvényeket írtam, melyeket hozzacsatoltam HTML elemekhez, a mapfile-hoz, vagy az OpenLayers alkalmazáshoz. Függvényeket hoztam létre a WMS rétegen való kereséshez; a megjelenített poligonok színének változtatásához; a térképfelület teljes képernyős vizualizációjához; a felületek új réteggként való hozzáadásához az OpenLayers rétegkezelőhöz; rétegek törléséhez. Ezekkel a függvényekkel lehetséges egyszerre egy-két fajt és alfajt láttatni különböző színekkel, mint ahogy az a könyvekben is megtalálható.

Az európai fafajok jelentős részét már most bemutatja, amin végre lehet hajtani különböző lekérési funkciókat. Jelenlegi állapota mutatja, hogy nagy vonalakban hogy is nézhet ki, hogyan működhet egy, a célnak megfelelő webhely, melyet a munkámra alapozva

tovább lehet bővíteni a célból, hogy létrejőjön egy ideális, a dendrológiai atlaszt teljes mértékben kiegészítő webes alkalmazás. Amint lehetőség nyílik újabb OpenLayers programcsomag használatára (minimum 3.1), akkor egyszerűen, egy hivatalos irányító eszköz implementálásával lehetőség nyílik képernyőmentést készíteni a megjelenített térképi tartalomról, és ezt illusztrációként lehet alkalmazni a későbbiekben létrehozott nyomtatott irodalom részeként. Ez a projekt egyik hosszú távú célja, amihez megtettem a szükséges kezdő lépéseket. Ezen kívül felmerült a lehetőség arra, hogy további webfejlesztői technikák bevonásával (pl.: szerverrel való kommunikáció) további bővítések is hozzáadhatóak, mint például az, hogy egyedi színbeállításokat adhasson meg a felhasználó, a jövőben ezt is lehetséges lenne kialakítani a program részeként.

6. Köszönetnyilvánítás

Hálával tartozom mindenkinek, aki valamilyen formában hozzá járult ahhoz, hogy az elmúlt évek kemény munkája után eljutottam a mesterszakos diploma befejezéséhez. Külön kiemelném témavezetőmet, dr. Gede Mátyást, aki lelkesedéssel követte és segítette a dolgozat létrejöttét; Mészáros Jánost, akitől bármikor segítséget kérhettem munkám során; szaktársaim közül Tóth Erikát, akivel a közös dolgozással töltött idő gyorsabban telt; illetve Taracsák Zoltánt, aki rengeteg hasznos tanáccsal látott el.

7. Abstract

The primary goal of this thesis was to create a webpage that implements the newest web mapping techniques, utilising maps delivered by geographical information systems. The work was based on the book titled “Conifers Around the World”, specifically their maps representing the spread and natural habitat of conifers.

During the course of the work I have used several technologies such as georeferencing the original raster maps in order to vectorise them later. The maps lacked coordinate grid system and I had to determine the correct projection by making observations on the coastline of Europe (as I have focused on the conifers of this continent). After I gained the georeferenced maps I created vectorised versions of them in QGIS. It was a crucial step in the thesis, as it was required to make a database containing attributes that can be used as a basis for searching in the webpage. Besides, vector images are more suitable for an online application that would be the outcome of the development process. The database ultimately contains the modern botanic taxonomical classification of plants.

After acquiring the necessary images and attribute data in a so called shape file a website had to be created that has the ability to present the data. OpenLayers was applied for building a web map application that can be implemented in the website. The application exhibits the polygons of the conifers on a base map. OpenLayers cannot use shape file format as of now, so a workaround solution with MapServer was employed, which reformats these files to an image file and hosts them on a WMS server. As for the base map, a map akin to the ones in the atlases were produced with the services offered by MapBox, which allows users to create an individually styled map and then use that as a base layer in OpenLayers (as opposed to utilising a readily available map by mapservers such as Google Maps, in which case one can raise copyright issues).

In order to elevate the interactivity of the website I wrote several JavaScript functions tied with HTML elements, with the mapfile, and with OpenLayers. There is a function for making searches on the WMS layer, and only exhibit certain polygons; for choosing the colour of the polygons; for adding layers to the OpenLayers layer switcher; for deleting layers; for entering full screen mode with only the surface of the map application. These functions provide a possibility to show only a few species and subspecies with different colour settings, similarly to the original maps.

In conclusion I have created the website I have planned to; I have made the essential steps for a modern, realistically working website that can serve as a basis for a GIS

application accompanying the books. In the future, when newer technologies are readily available to work with, many extensions can be made prolong the usefulness of the site. I had ideas for example letting the user define desired colours for the polygons. With the newer OpenLayers versions (minimum 3.1) a prewritten function can be added to the application that can save good quality print screens of the map area with a simple user input, and these pictures could be utilised as illustration in further literature.

8. Irodalomjegyzék

Battersby, S., Finn, M., Usery, L., Yamamoto, K., (2014): Implications of Web Mercator and Its Use in Online Mapping. *Cartographica* 49:2, pp. 85-101.

Brown, E., (2016): *Learning JavaScript*. O'Reilly Media, Sebastopol, Kalifornia, Amerikai Egyesült Államok. ISBN: 978 1 491 91491 5, p. 364.

DeBolt, V., (2005): *HTML és CSS, Webszerkesztés stílusosan*. Kiskapu, Budapest. ISBN: 963 9301 96 5, p. 456.

Debreczy, Zs., Rácz, I., (2001): *Conifers Around the World*. Vol. 1-2. DendroPress, Budapest. p.1089.

Debreczy, Zs., Rácz, I., (2000): *Fenyők a Föld körül*. Dendrológiai Alapítvány, Budapest. ISBN: 9630058987, p. 552.

Elek I., (2011): *Térképészet és geoinformatika (II. kötet, Geoinformatika)*. ELTE Eötvös Kiadó, Budapest. p. 183.

Érdi-Krausz Gy., (1958): *Vetületanalízis. Térképtudományi Tanulmányok (Studia Cartologica)*. Honvédelmi Minisztérium Térképészeti Intézet, Budapest, pp. 194-270.

ESRI Shapefile Technical Description (1998).

Gede M., (2016): *A MapServer használata. Oktatási segédanyag*.
<http://mercator.elte.hu/~saman/hu/okt/mapsver/>. Utolsó elérés: 2016.04.21.

Hazard, E., (2011): *OpenLayers 2.10 Beginner's Guide*. Packt Publishing, Birmingham, Egyesült Királyság. ISBN 978-1-849514-12-5. 353 pp.

Linnaeus, C., (1753): *Species Plantarum*. Stockholm, 1200 p.

MapBox (2016): <https://www.mapbox.com/>. Utolsó elérés: 2016.04.21.

Mozilla Developer Network (2016): Using fullscreen mode. https://developer.mozilla.org/en-US/docs/Web/API/Fullscreen_API. Utolsó elérés: 2016.05.16.

Kropła, B., (2005): Beginning Mapserver: Open Source GIS Development. Apress, New York City, Amerikai Egyesült Államok. ISBN 1-59059-490-8. p. 448.

Perez, A.S., (2012): OpenLayers Cookbook. Packt Publishing, Birmingham, Egyesült Királyság. ISBN 978-1-84951-784-3. p. 300.

Podani J., (2014): A szárazföldi növények evolúciója és rendszertana: Vezérfonal egy nem is olya könnyű tárgy taulásához. ELTE Eötvös Kiadó, Budapest.

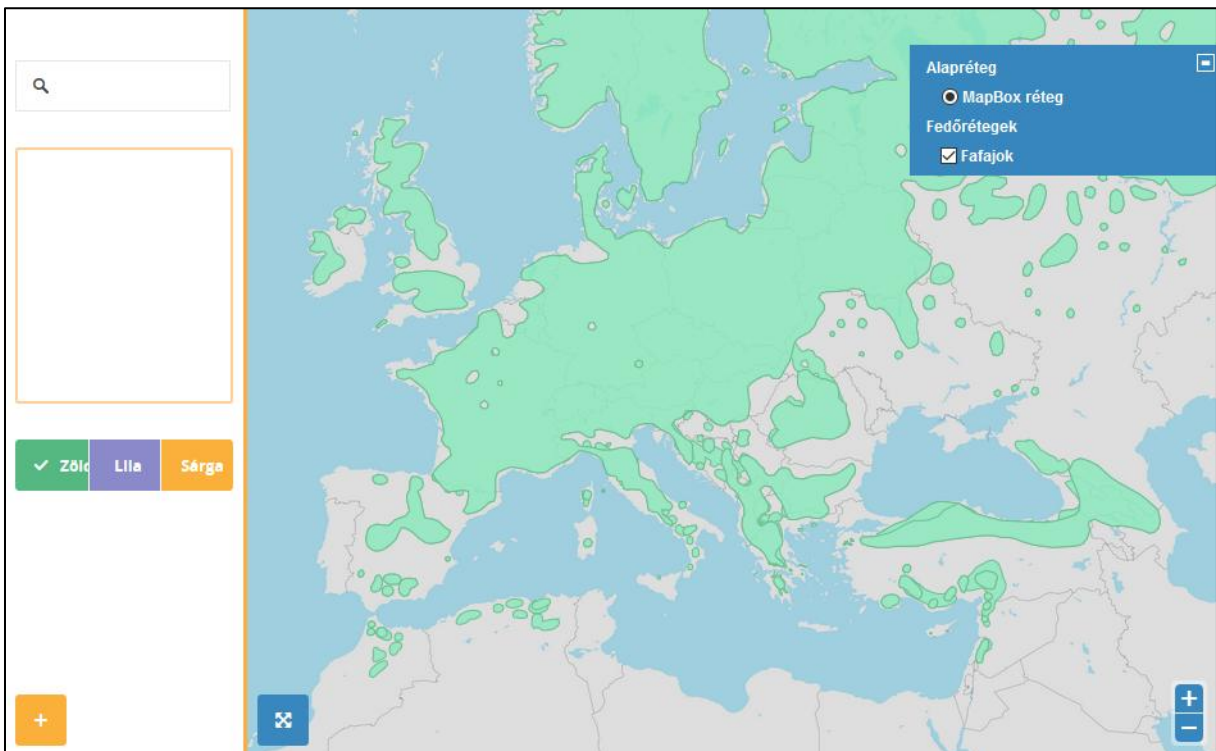
Tímár G., (2008): Georeferencia: Térképi vetületek és geodéziai dátumok szabatos használata a térinformatikában. Elektronikus jegyzet. <http://sas2.elte.hu/tg/georeferencia.htm> Utolsó elérés: 2016.05.18.

Zentai L., (2000): Számítógépes térképészet. Egyetemi tankönyv. ELTE Eötvös Kiadó, Budapest. ISBN 963 463 317 X. p. 245.

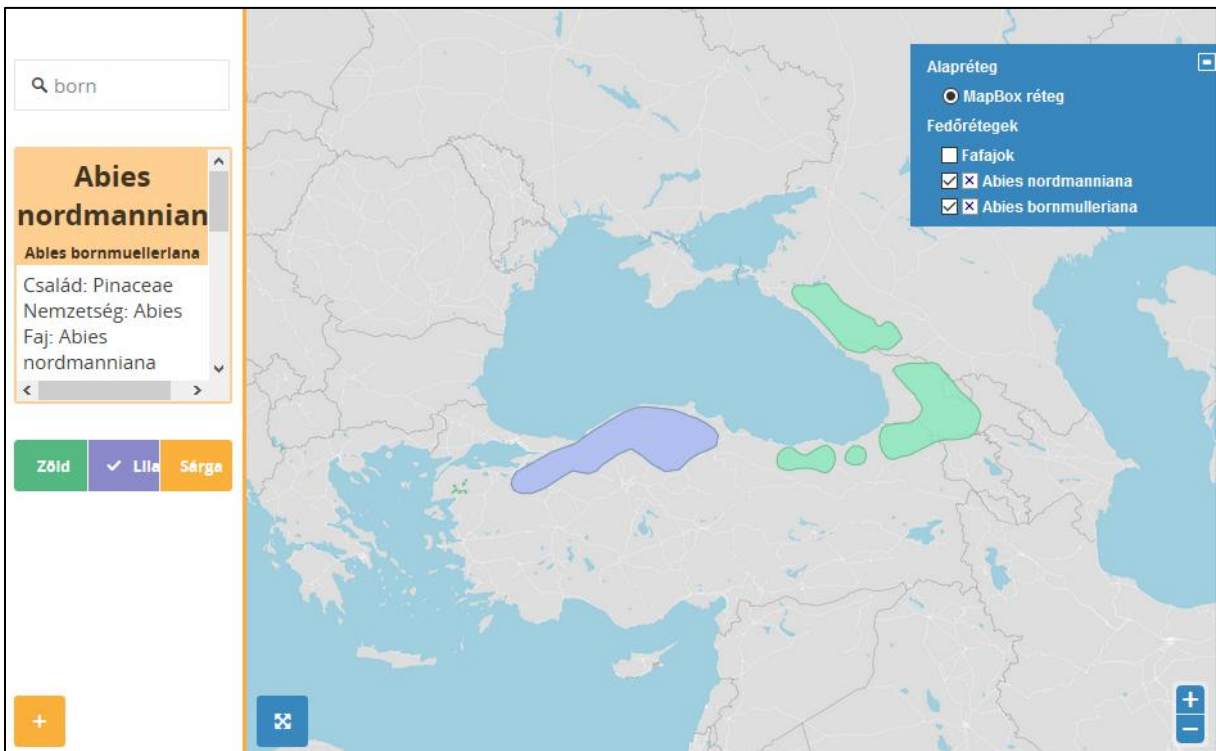
W3C (World Wide Web Consortium) (2016): Standards. <https://www.w3.org/standards/> Utolsó elérés: 2016.05.19.

9. Mellékletek

1. Képernyőmentések a weboldalról:



20. ábra: A weboldal a dolgozat befejezése előtti állapotában.



21. ábra: Keresés, színmódosítás, több réteg egy időben való tárolása, adatbázis információk kiírásának demonstrálása.

Attribute table - fenyofak :: Features total: 19, filtered: 19, selected: 0

	id	torzs	osztaly	rend	csalad	nemzetseg	faj	alfaj
4	4	Pinophyta	Pinopsida	Pinales	Pinaceae	Abies	Abies ciliica	NULL
5	7	Pinophyta	Pinopsida	Pinales	Pinaceae	Abies	Abies cephalonica	NULL
6	6	Pinophyta	Pinopsida	Pinales	Pinaceae	Abies	Abies numidica	NULL
7	8	Pinophyta	Pinopsida	Pinales	Pinaceae	Abies	Abies nordmanni...	Abies equi-trojani
8	9	Pinophyta	Pinopsida	Pinales	Pinaceae	Abies	Abies nebrodensis	NULL
9	10	Pinophyta	Pinopsida	Pinales	Pinaceae	Abies	Abies pinsapo	var. pinsapo
10	11	Pinophyta	Pinopsida	Pinales	Pinaceae	Abies	Abies pinsapo	var. marocana
11	12	Pinophyta	Pinopsida	Pinales	Pinaceae	Cedrus	Cedrus atlantica	NULL
12	13	Pinophyta	Pinopsida	Pinales	Pinaceae	Cedrus	Cedrus brevifolia	NULL
13	14	Pinophyta	Pinopsida	Pinales	Pinaceae	Cedrus	Cedrus libani	supsp. libani
14	15	Pinophyta	Pinopsida	Pinales	Pinaceae	Cedrus	Cedrus libani	supsc. stenocoma
15	16	Pinophyta	Pinopsida	Pinales	Cupressaceae	Cupressus	Cupressus dupre...	NULL
16	17	Pinophyta	Pinopsida	Pinales	Cupressaceae	Juniperus	Juniperus brevifolia	NULL
17	18	Pinophyta	Pinopsida	Pinales	Cupressaceae	Juniperus	Juniperus cedrus	NULL
18	19	Pinophyta	Pinopsida	Pinales	Cupressaceae	Juniperus	Juniperus commu...	var. communis

Minden elem

22. ábra: A fajokhoz tartozó adatbázisból részlet.

2. A CD lemez tartalma:

- A dolgozat elektronikus formában
- A weboldal forráskódja
 - fenyofak.html
 - fenyofak.map
 - template.html
 - LayerSwitcherRemovable.js
 - A fenyőfák felületeit és az adatbázist tároló shape fájl

NYILATKOZAT

Alulírott, **Csomós Boglárka (NEPUTN kód: A8L0LH)** nyilatkozom, hogy jelen dolgozatom teljes egészében saját, önálló szellemi termékem. A dolgozatot sem részben, sem egészében semmilyen más felsőfokú oktatási vagy egyéb intézménybe nem nyújtottam be.

A témavezető által benyújtásra elfogadott diplomamunka PDF formátumban való elektronikus publikálásához a tanszéki honlapon **hozzájárulok**.

Budapest,

a hallgató aláírása