Teendők listája

☐ Itt számít, hogy konkrétan Shapefile vagy csak annyi, hogy fájlba mentett?

– Számít, ugyanis shapefile esetén más a feature id-k kezelése 10

Eötvös Loránd Tudományegyetem

Informatikai Kar

Programozáselmélet és Szoftvertechnológiai

Tanszék

Téradatok verziókövetésének megvalósítása Quantum GIS-ben

Témavezető:

Cserép Máté

egyetemi tanársegéd

Szerző:

Dorogi Benjámin

programtervező informatikus BSc

Budapest, 2019

Tartalomjegyzék

1. fejezet

Bevezetés

Az informatikai projekteknek a munkafolyamatok követése, a különböző állapotok megtekintésének és visszaállításának lehetősége kulcsfontosságú részét képezi az agilis projektmenedzsment, valamint a kooperatív és elosztott munkavégzés elősegítése érdekében. Napjainkban a szöveges alapú munkák – például szoftverkód, szöveges dokumentumok, táblázatok – esetében már számtalan eszköz áll rendelkezésünkre ezen funkcionalitás eléréséhez. Attól függően, hogy milyen módon tárolják a kezelt állományokat, a verziókezelő rendszereket két részre oszthatjuk. A centralizált verziókezelők egy központi tárhelyen tárolják a fájlokat, és minden felhasználó ezt a központi verziót módosíthatja, ilyen elven működik például az SVN. Az elosztott revíziókezelők esetében minden felhasználó rendelkezik a követett projekt teljes másolatával, és szerkesztéskor ez módosul, ilyen rendszerre a legismertebb példa a Git. Az említett verziókezelők közös tulajdonsága, hogy mind úgynevezett állapot alapú revíziókezelést valósítanak meg, azaz minden egyes módosításkor az állományok nyers változásait, sorok törlését, hozzáadását tárolják, és ezek alapján állítanak elő újabb vagy régebbi revíziókat.

Az állapot alapú verziókezelés egyik hiányossága, hogy az adatokat binárisan tároló folyamatok esetében – amik jellemzően valamilyen grafikus információt tárolnak –, nagy tárigényű és jóval kevésbé hatékony. Ezen kívül problémát jelent az is, hogy az állapot alapú verziókezelés esetén a módosítások szemantikai információja elvész, nem követhető, hogy pontosan milyen transzformációk kerültek végrehajtásra, függetlenül a tárolás módjától. Az ilyen téradatok esetében a művelet alapú revíziókezelés sokkal célravezetőbb, mivel ebben az esetben a változások (vagy más szóval

delták) az adott objektumokon végzett műveletek, melyek újbóli végrehajtásával megkaphatjuk a frissített változatát a módosított adatnak, az invertált műveletek alkalmazásával pedig korábbi állapotokat érhetünk el. A téradatok hatékony verziókezelésére való igényre válaszként született az Eötvös Loránd Tudományegyetemen fejlesztett *AEGIS* geoinformatikai programcsomag [**giachetta2013aegis**, **aegis**] revíziókezelő modulja [**cserep2013versioning**, **cserep2015operation**], ami interfészt biztosít az összes szükséges folyamathoz, viszont ezen szoftvernek még nem született valós, ipari környezetben is felhasználható implementációja.

Szakdolgozatom motivációja az előbbi hiány megszüntetése, a széleskörűen elterjedt és használt nyílt forráskódú *QGIS* térinformatikai programhoz [**qgis**] írt verziókezelő beépülő modul megvalósításával, mely a *Q-Aegis* nevet kapta.

A modul feladata a *QGIS* alapvető geometriai műveleteinek naplózása, tetszőleges verzióállapotok betöltése az *AEGIS* könyvtár funkcióinak felhasználásával

2. fejezet

Felhasználói dokumentáció

2.1. A verziókezelésről röviden

Verziókezelés alatt azon tevékenységet értjük, amely során egy változó állapotú folyamat változásait tároljuk és kezeljük. Egy verziókezelt projekt esetén nem csak visszaállíthatóak korábbi állapotok és előállíthatóak újabbak, hanem követhető az állományok alakulása is, valamint leegyszerűsödik a munkamegosztással járó problémák mennyisége is.

2.2. A téradatok verziókezelésének problémája

Téradatokkal – vagy bármilyen kép alapú adatokkal – egyre több szakterület dolgozik, és általánosan ezek a folyamatok kifejezetten magas számú felhasználók által végzett módosítással járnak. Az információk tárolása vagy binárisan, vagy valamilyen összetett geodéziai adatbázis állománnyal történik, ezek verziókövetésére pedig nem született még általánosan alkalmazható megoldás. A QGIS egy széles körben elterjedt, főleg térképészeti adatok létrehozását, módosítását és tárolását támogató szoftver, ami még nem rendelkezik verziókezelést megvalósító kiegészítéssel, a *Q-Aegis* modul ezen hiány betöltésére született.

2.3. A modul működése

A *Q-Aegis* plugin az AEGIS nevű téradatkezelő keretrendszer felhasználásával biztosítja a *QGIS*-ben létrehozott projektek műveletalapú verziókezelését. A verziókezelő és a szerkesztőprogram közti technikai különbségeket a nyílt forráskódú *Python.Net* [**pythonnet**] modul használatával hidalja át. A program követi a rétegek és a rajtuk lévő objektumok (*feature*ök) változásait, a geometriák alakulását reverzibilis transzformációkká alakítja és ezeket tárolja. A korábbi vagy újabb verziók betöltésekor így nem szükséges forrásfájlokat cserélni, a rétegek új állapotait beállítani, mindez automatikusan megtörténik. Ezen kívül a megvalósítása miatt a *Q-Aegis* plugin lehetővé teszi, hogy a *QGIS*-ben végzett munka során csupán az eredetileg nem menthető memóriában tárolt rétegekkel dolgozzon a felhasználó.

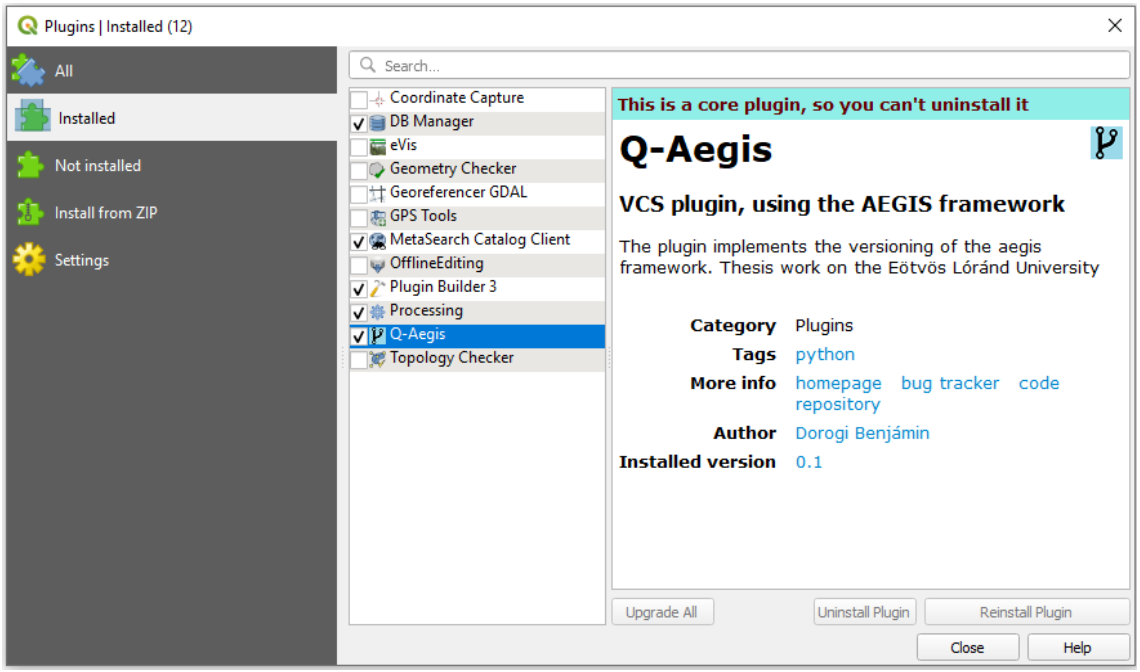
2.4. A beépülő modul telepítése

2.4.1. Rendszerkövetelmények

Mivel a program egy beépülő modul (*plugin*) a *QGIS* térinformatikai programhoz, ezért használatához szükséges legalább a szoftver jelenléte. A modul a *QGIS* 3.x-es verzióit támogatja, a tesztelés 3.6-os verzióval történt. Az *AEGIS* csomag használata miatt, amennyiben nincs még jelen a rendszerben, telepíteni kell a *Microsoft .NET* keretrendszer 4.0-s, vagy annál frissebb verzióját. Az előbbi dependencia miatt a program Linux és MacOS operációs rendszerek alatt a Mono platform telepítése is szükséges.

2.4.2. Telepítés

A beépülő telepítése kifejezetten egyszerű, csak ki kell csomagolni a mellékelt tömörített állomány tartalmát a helyi QGIS verzió `python/plugins` könyvtárába, amely telepítési könyvtár a `apps/qgis/python/plugins` útvonalán érhető el. Ezután a QGIS-t elindítva a *Plugins / Manage and install plugins* menüponton keresztül megnyitott plugin kezelőben be kell kapcsolni a Q-Aegis plugint, hogy az ábrán látható állapotot kapjuk :



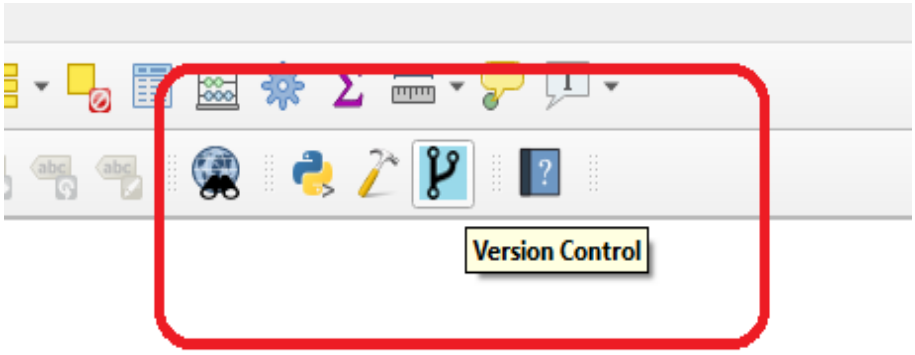
2.1. ábra. Az aktiválás utáni állapot

Ezután a verziókezelő modul használatra kész.

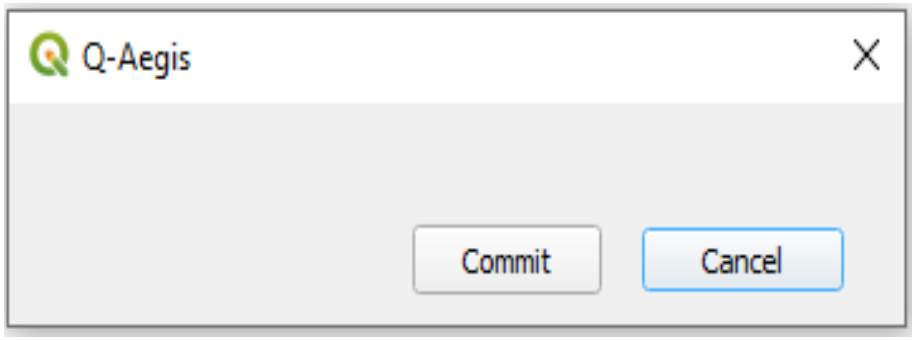
2.5. A verziókezelő használata

2.5.1. A kezelőfelület

A beépülő modul a *QGIS*-en belül létrehozott projektekkel van összekötve, így amíg nem nyitunk meg egy projektet, nem érjük el az új funkciókat. Projekt létrehozása vagy megnyitása után a modul kezelőfelülete a *QGIS* beépülő modulok eszköztárán található ikonra (2.2. ábra) kattintva nyitható meg, és a 2.3. ábrán látható módon néz ki alapértelmezett állapotban.



2.2. ábra. Az eszköztáron található ikon



2.3. ábra. A kezelőfelület alapállapota

Az ablak a *Cancel* bezárás gombbal zárható be, a többi gomb funkciójára, valamint a felület esetleges változásaira a használatot magyarázó részekben térünk ki.

2.5.2. Új követett projekt létrehozása

Új projekt létrehozása esetén, -a *QGIS* eredeti működésétől eltérően- a modul felajánlja a projekt mentését. Amennyiben mentésre kerül az új projekt, a verziókezelő funkciók elérhetővé válnak. Mivel a modul megfelelő működéséhez szükséges a projektfájl jelenléte, ezért a mentés hiányában nem jön létre verziózott projekt, amire figyelmeztet is a program. Ha már meglévő projektet szeretnénk verziókezelés alá vonni, akkor a projektfájl megnyitása után a kezelőfelületen található *Commit* beküldés gombra kattintva a projekt teljes állapota bekerül a rendszerbe és a további módosítások már követhetőek lesznek.

2.5.3. Módosítások mentése a verziókezelőbe

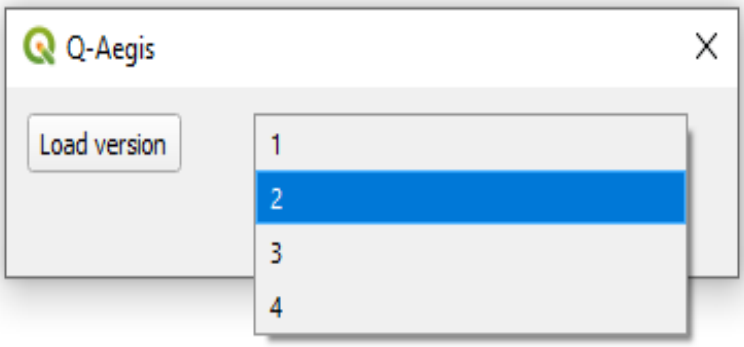
Miután módosításokat végeztünk a projekten, a *Commit* gomb újbóli megnyomásával az új állapot bekerül a revíziókezelő rendszerbe. A *QGIS* szerkesztési funkciói azok széles terjedelme miatt (egyelőre) nem teljes körűen támogatottak *Q-Aegis*ben, hanem a következő alapvető módosítások kezeltek:

- Új réteg hozzáadása
- Létező réteg eltávolítása
- Geometria hozzáadása réteghez
- Geometria eltávolítása rétegről
- Geometria módosítása:
  - Tetszőleges geometria részének vagy egészének eltolása
  - Vonal típusú geometriákba új pont felvétele, pontok eltávolítása vonalakból
  - Poligon típusú geometriákhoz "lyukak" hozzáadása vagy eltávolítása
  - Multigeometriákhoz részek hozzáadása vagy eltávolítása

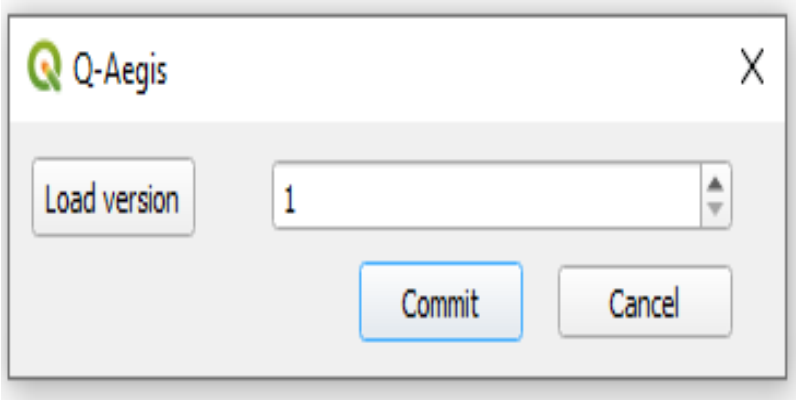
2.5.4. Tetszőleges verzió betöltése

Amennyiben már van a projekthez mentett verzió, a kezelőfelület némileg megváltozik, attól függően hogy mennyi elérhető verzió van jelen, megjelenik egy legördülő lista, vagy egy léptethető numerikus beviteli mező (2.4) (3.3) . Egy tetszőleges számú verziót kiválasztva, majd a *Load Version* gombra kattintva a projektbe betöltődik a kiválasztott verzió. Amennyiben a betöltés előtt a projektben vannak olyan módosítások, amelyek még nem kerültek be a rendszerbe, a program egy felugró ablakkal figyelmezteti a felhasználót, és felajánlja neki a lehetőséget, hogy commitolja a módosításait, mielőtt betöltené az új verziót.

Ha korábbi állapot van betöltve, és módosításokat végzünk rajta, akkor ha menteni próbálunk, a rendszer hibaüzenettel jelzi, hogy előbb a legfrissebb verzióra kell állni és utána lehet módosítani.



2.4. ábra. A verzióválasztó lista



2.5. ábra. A verzióválasztó beviteli mező

Megjegyzés. *A* QGIS *több opcióval is rendelkezik arra, hogy milyen módon tároljuk a rétegeinket. A* Q-Aegis *csak a vektoros rétegtípusokon végzett műveleteket támogatja. Szerkesztés közben nem számít, hogy memóriában tárolt ideiglenes réteggel, vagy* Shapefile*-ban tárolt réteggel dolgozunk , viszont verzió betöltésekor az összes réteg törlődik, és a betöltendő verziónak megfelelő rétegek jönnek létre, amik mind memóriában tároltak.*

Itt számít, hogy konkrétan Shapefile vagy csak annyi, hogy fájlba mentett? – Számít, ugyanis shapefile esetén más a feature id-k kezelése

2.5.5. Aktuális verzió száma

Mivel ha nem a legutóbbi verzión állunk, nem engedélyezett a módosítások mentése, ezért szükséges, hogy tudjuk követni épp melyik verzión állunk. Ezt teszi le-

hetővé a *QGIS* ablak bal alsó sarkában található verziószám címke, ami az alábbi módon néz ki.

2.6. ábra. A verziószámot jelző panel

2.5.6. Verziókezelt projekt betöltése

A *Q-Aegis* jelenleg még nem ad lehetőséget arra, hogy létező tárolóból hozzunk létre egy új QGIS projektet, de az alábbi lépésekkel ez megvalósítható:

1. Hozzunk létre egy új projektet, mentsük el, így létrehozva egy új, üres tárolót
2. Másoljuk le a projekt fájlt tartalmazó könyvtárban található, `projektnév_qaegis_repository` mappát
3. Szúrjuk be a lemásolt mappát az új projektfájl mellé, és módosítsuk a nevét az új projektnek megfelelőre
4. Az új projektet megnyitva rendelkezésre áll a projekt másolata

2.6. A modul eltávolítása

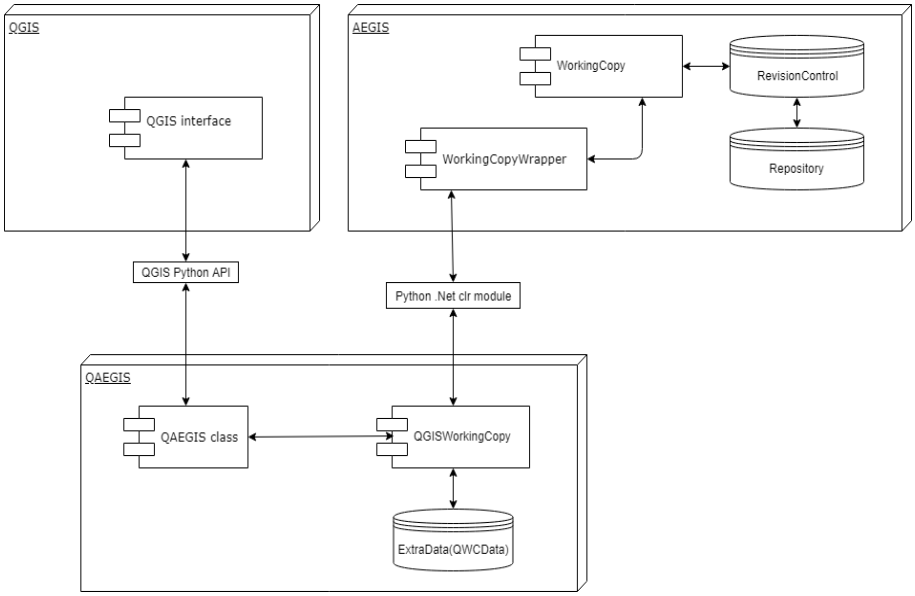
A *Q-Aegis* modul kikapcsolható a *QGIS* beépülő modulokat kezelő felületén, ugyanúgy ahogy hozzáadásra került. A teljes eltávolításhoz töröljük ki a `qaegis` mappát a `plugins` könyvtárból.

3. fejezet

Fejlesztői dokumentáció

3.1. A feladat specifikálása

A beépülő modul célja a téradatok verziókezelésének megvalósítása a QGIS-en belül az AEGIS keretrendszer segítségével. Ez szétválasztható egy *back-end* részre, ami a verziókezelésért valamint az adatok és transzformációk tárolásáért felel, és egy *front-end* részre, ami egyrészt grafikus felhasználói felületet biztosít, valamint felel az adatreprezentációs és architekturális különbségek áthidalásáért.

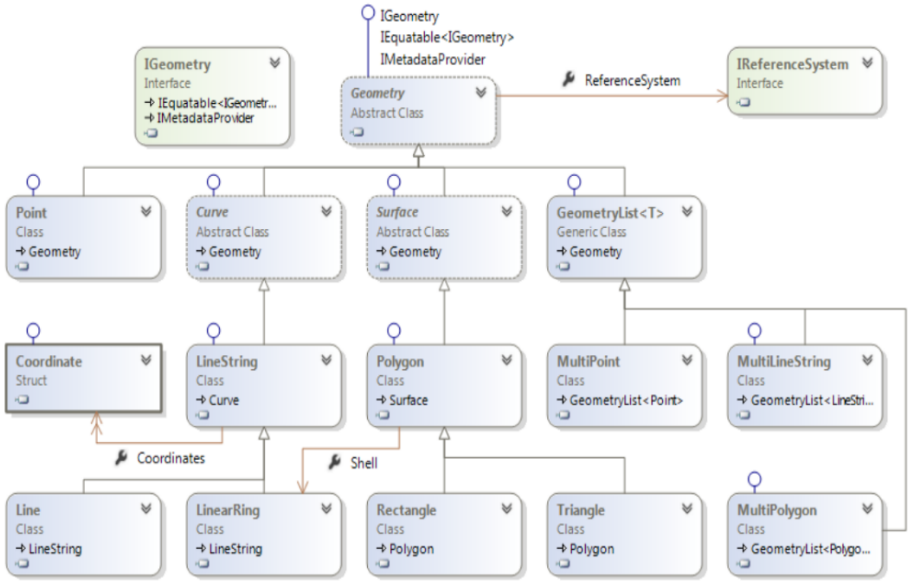


3.1. ábra. A végleges program komponensdiagramja

3.1.1. A geometriaadatok kezelése

A geometriák tárolása, betöltése, valamint eltávolításuk kezelése mind az AEGIS létező funkciói voltak a projekt kezdeténél, így a tényleges feladat, a transzformációk definiálása, és létrehozása. Az átalakítások tervezése közben az a fő szempont, hogy a lehető legáltalánosabbak legyenek, és minden lehetséges módosítás leírható legyen véges alkalmazásukkal. Ezt a törekvést segíti az, hogy az AEGIS geometriatípusai megfelelnek a *Simple Feature Access* (röviden *SFA*) standardnak [**sfa**], ezáltal egy adott hierarchiával rendelkeznek: A pontok különálló típust képeznek, A vonalak koordináták sorozatai, amik hasonlóan viselkednek a pontokhoz, a sokszögek pedig gyűrűkkel vannak leírva, amik a vonalak speciális változatai, így az "alacsonyabb szintű" geometriák transzformációi felhasználhatóak a bonyolultabb térobjektumok átalakításainak leírásához is.

Megjegyzés. *A* Simple Feature Access *geometriák tárolására és elérésére kialakított két részből álló szabvány. Az első rész (SFA-CA azaz* Common Architecture*) egy modellt definiál a két dimenziós geometriáknak pontok közti lineáris interpolációval. Ez a rész definiálja WKT és WKB szabványos reprezentációkat is. A második rész a szabvány SQL alapú implementációját írja le. Az SFA standard az OGC (*Open Geospatial Consortium*) és az ISO (*International Organization for Standardization*) szervezetek által elfogadott.*



3.2. ábra. A *Simple Feature Access* objektum-relációs modellje

Az alakzatok lehetséges változásainak megfelelő rekreálásához a következő transzformációkat szükséges definiálni:

- Pont eltolása
- Vonal részleges vagy teljes eltolása
- Pont felvétele a vonalba
- Pont eltávolítása a vonalból
- Sokszög héjának módosítása
- Sokszögben lévő lyuk módosítása
- Lyuk felvétele sokszögbe
- Lyuk eltávolítása sokszögből

Ezeken kívül az összes többi alakzatot egybefogó, úgynevezett multigeometriák változásainak leírásához szükségesek a következő műveletek:

- Multigeometria egy vagy több részének módosítása a megfelelő egyedi geometria transzformációkkal
- Geometria hozzáadása multigeometriához
- Geometria eltávolítása multigeometriából

Ezen transzformációk segítségével bármilyen alapvető geometriaváltozást le lehet írni.

A transzformációk generálása komplex feladat, melynek során elemeznünk kell a kiindulási geometriát, és a végeredményt, ez alapján pedig létrehozni azt a transzformációgyűjteményt amit végrehajtva az eredetin, megkapjuk az újat. Vonalláncokra és poligonokra a következő algoritmussal adható meg:

1. Tekintsük az eredeti és a módosult geometria komponenseinek számát (ld. 3.12. szakasz)
2. A két geometria megegyező hosszú részein -ami megegyezik a komponensszámuk minimumával- hozzuk létre azokat a transzformációkat, amik az eredeti részeit az új részeknek megfelelő részekre alakítják

3. Amennyiben az új geometria hosszabb, a régiben még nem létező részeket adjuk hozzá

4. Amennyiben a régi geometria hosszabb, az újban már nem létező részeket töröljük ki

Mivel pont geometriáknál csak az eltolás művelete létezik, nem szükséges részekre bontani a transzformációgenerálást, a koordináták különbségeiből kiszámolható az összes szükséges paraméter.

Ezen részek implementálásával a geometriák alakulásának követése megvalósítható.

3.1.2. A QGIS reprezentáció kezelése

A fő különbség az AEGIS és a QGIS geometriákat kezelő része között az, hogy az előbbi csak a téradatot kezeli, az utóbbi viszont ezeket még rétegekre osztva jeleníti meg. Ezzel önmagában nem lenne probléma, viszont a QGIS nem rendel egyedi azonosítót a geometriákhoz, így előfordulhat, hogy két különböző rétegen azonos geometria szerepel, és módosításukkor nem tudjuk megállapítani, hogy az AEGIS-ben tárolt adatok közül melyik módosult. Ez alapján a beépülő modulon belül megvalósítandó plusz funkciók a következőek:

- A QGIS objektumok (feature-ök) és az AEGIS geometriák kapcsolatának nyilvántartása
- A QGIS műveleteinek megfeleltetése az AEGIS műveleteknek
- Az AEGIS-ben tárolt állapotok megjelenítése
- A verziókezelés lépéseihez grafikus kezelőfelület biztosítása

3.2. Geometriai adatok változásának kezelése

3.2.1. Az AEGIS WorkingCopy

Az AEGIS verziókezelő moduljának külső interfésze az úgynevezett `Working Copy`, avagy munkapéldány. Ezen kereszült a következő funkciók érhetőek el:

1. Teljes revíziókezelő, és hozzá tartozó tároló létrehozása vagy betöltése, attól függően, hogy van-e már jelen egy
2. Geometriák hozzáadása az aktuális verzióhoz
3. Geometriák eltávolítása az aktuális verzióból
4. Geometria módosítása az aktuális verzióban, az IGeometryTransformation interfészt megvalósító transzformációpéldánnyal
5. Aktuális verzió mentése a verziókezelőbe
6. Tetszőleges verzió betöltése és megjelenítése

A WorkingCopy két hiányossággal rendelkezik a beépülő modulban való használathoz. Transzformációk előállítását nem támogatja, valamint az aktuális állapotát kifelé csak geometriák halmazaként biztosítja, ami a QGIS réteges megjelenítésében problémákat okozhat.

3.2.2. Az AEGIS WorkingCopyWrapper

A munkapéldány említett hiányosságainak pótlására jött létre a `WorkingCopyWrapper` osztály, amely a `WorkingCopy`-hoz képest az alábbi többletfunkciókkal bír:

1. Eredeti és módosult geometria alapján transzformációk generálása, majd ezzel a workingcopy módosító függvényének használata
2. Egy egyedi azonosítókkal ellátott geometrialista biztosítása a lokális állapotról

A szükséges transzformációkat és generálásukat az AEGIS Processing könyvtárában kellett implementálni.

3.2.3. A transzformációk

A geometriák átalakító műveletek kialakítása az alábbi elven alapul:

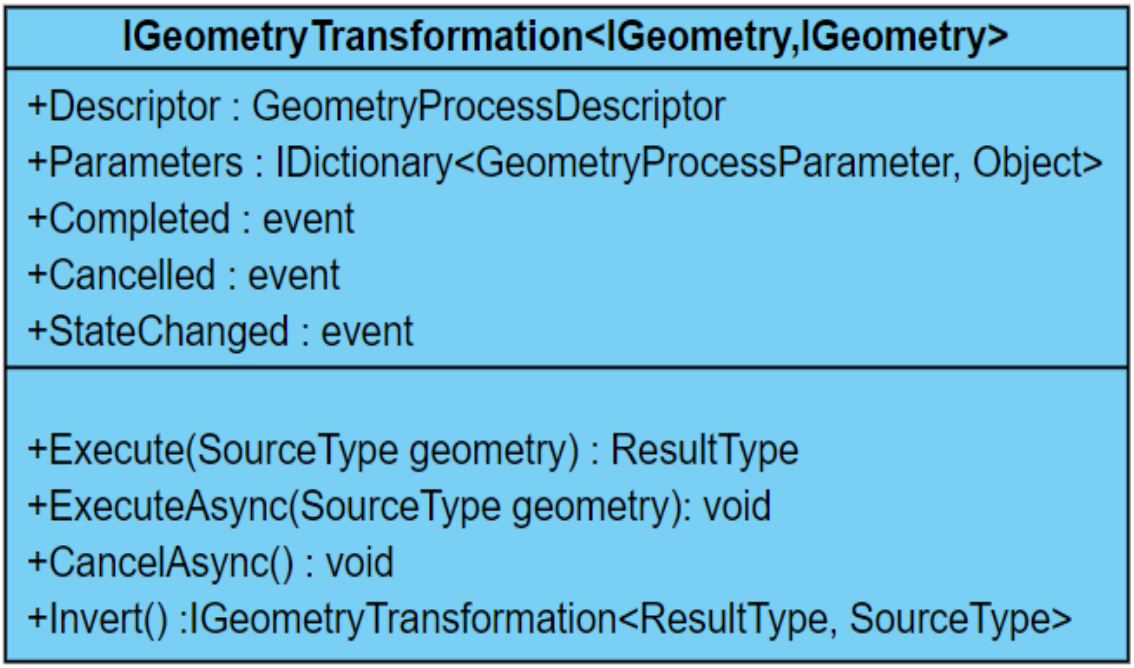
- A pontok transzformációja leírható egy egyszerű vektor hozzáadással
- Az összes többi geometria kisebb részekből áll (ld. 3.12. szakasz)

- Egy geometria módosítása leírható a részeire vonatkozó kisebb transzformációkkal

Ez alapján három típusú transzformáció alakult ki: *i*) a rész módosítása, ami a módosítandó részek indexeit és a részeken elvégzendő résztranszformációit tartalmazza; *ii*) a rész hozzáadása amely az új rész geometriáját és a beszúrás indexét tartalmazza; *iii*) valamint a rész eltávolítása, amely az eltávolítandó rész indexét tárolja, valamint az eltávolított rész geometriáját, erre az invertálhatóság megvalósításához van szükség.

3.2.4. Az IGeometryTransformation interfész

Ahogy korábban is említésre került, az AEGIS a transzformációkat az IGeometryTransformation implementációjaként várja.



3.3. ábra. A geometriatranszformációs interfész osztálydiagramja

Ezen interfész legfontosabb metódusai:

1. Execute(IGeometry geometry) : a transzformáció végrehajtása az IGeometry interfésznek megfelelő téradaton, a módosított alakzat visszaadása. A forrásgeometriának meg kell egyeznie a transzformáció bemeneti típusával.

2. Invert(): A transzformáció inverzének előállítása, mellyel helyreállítható az eredeti geometria. Hozzáadás típusú művelet inverze ugyanazon az indexen történő eltávolítás és fordítva, részmódosítás esetén a résztranszformációk inverzeinek végrehajtása ugyanazokon a részeken.

Az összes módosítást leíró osztály rendelkezik ezen metódusok egyedi implementációival, melyek megfelelnek a saját geometriatípusaiknak.

3.2.5. A transzformációk generálása

: Az eredeti és módosított geometriákból transzformációk generálását az AEGIS Processing modulban található TransformationFactory statikus osztályok végzik. A transzformációk létrehozása a specifikációban leírt algoritmus alapján működik, ez az alábbi pseudokóddal szemléltethető:

```
set unmodifiedCount to number of parts in unmodified geometry
set modifiedCount to number of parts in modified geometry
set result to new empty transformation list
for index := 0 to minimum(unmodifiedCount, modifiedCount) do
  create transformation from parts on the index and add to result
for index := unmodifiedCount to modifiedCount do
  create addition for new parts and add to result
for index := modifiedCount to unmodifiedCount do
  create deletion for removed parts and add to result
return the result
```

3.1. forráskód. A transzformációgenerátorok általános algoritmusa

Mint látható, a generáló függvény először létrehoz annyi résztranszformációt, amekkora részen még egyezik a két geometria hossza, majd a még nem, vagy már nem jelen lévő részgeometriák létrehozásának illetve eltávolításának műveleteit állítja elő. Az összes transzformációgenerátor hasonló módon működik.

3.3. C# folyamatok használata Python kódban, a Python.NET modul

Ahogy a specifikáció elején említésre került, a szoftver amihez a verziókezelő beépülő modul és a verziókezelést biztosító könyvtár két teljesen különböző technológiát használ. Mivel a Python programozási nyelv széleskörűen elterjedt különböző extra funkciók hozzáadására más környezetekhez, ezért több opció volt az eltérő megoldások összekötésére. Az egyik az IronPython nevű kiegészítő modul a Pythonhoz, viszon ez csak a nyelv 2.7-es verzióját támogatja, aminek hamarosan megszűnik a támogatottsága, így más utat kellett keresni. Az előbbihez hasonlóan nyílt forráskódú Python.NET modul azonos funkcionalitással rendelkezik, és a legújabb verziókkal is kompatibilis, így kézenfekvő megoldásnak bizonyult. A Python.NET projekt integrálása a QGIS pluginba egyedül azért okozott nehézséget, mivel a nyelvi összeköttetést biztosító szoftver fordításához számos dependenciának jelen kell lennie a rendszerben, így végül már fordított bináris formájában került a beépülő modulba. Ez sajnos kizárja annak lehetőségét, hogy a könyvtár esetleges frissülésével a plugin is dinamikusan megkapja a legújabb állományokat, viszont a program telepítését meglehetősen megkönnyíti.

3.3.1. A Python .NET használata

A Python.NET az úgynevezett *Common Language Runtime* (*CLR*) modult biztosítja a Python programozási nyelvhez. Amennyiben a fordított források a Python belső elérési útvonalain szerepelnek, importálható a CLR modul. Ezután a szintén classpathhoz adott dinamikusan linkelhető szoftverkönyvtárba (DLL) fordított C# könyvtárakat is hozzáadhatjuk a Python kódhoz. A tényleges implementációban a beágyazás az alábbi módon néz ki :

```
1 pluginpath = "{}/../python/plugins/qaegis".format(QgsApplication.
     pluginPath())
2 sys.path.insert(0,pluginpath+"/pythonnet")
3 sys.path.insert(0,pluginpath+"/aegis")
4
5 import clr
6 clr.AddReference('AEGIS.Core')
```

```
clr.AddReference('AEGIS.IO')
clr.AddReference('AEGIS.Versioning')
clr.AddReference('AEGIS.Processing')
```

3.2. forráskód. A CLR modul bekötése, és a dll-ek importálása a Q-aegis kódjában

Az importálás után pedig például ilyen módon hozhatunk létre egy String kulcsokkal egész számokat tároló C# szótárat:

```
dictionary =  Dictionary[String,int]()
```

3.3. forráskód. C# objektum használata python kódban

3.4. Adatátvitel Python és C# közt

A geometriák ugyan az AEGIS és a QGIS rendszerekben is szabványosan vannak kezelve, viszont a megvalósításaik különbözőek. Az ebből adódó problémák elkerülése érdekében a geometriák a legtöbb helyen a WKT reprezentációjukban, szövegesen kerülnek átadásra. Ezen kívül az AEGISben a geometriákhoz rendelt egyedi azonosító is használva van néhány helyen az objektumok kiválasztására. Erre azért van szükség, mert az AEGIS a geometriák közti ellenőrzésekor sokkal pontosabb adatokkal dolgozik mint a QGIS és ezek az eltérések sokszor nem várt viselkedéseket eredményeztek, például geometriák módosításakor a verziókezelő modul ellenőrzi, hogy van-e a módosított geometriával egyező a tárolóban, és ha nem talál ilyet, akkor szimplán felveszi újként, ami a pontatlanságok miatt lehetetlenné tette a módosítások hatékony követését, ezért módosításkor az eredeti állapotot közvetlenül kulcs alapján szedi ki a rendszer a tárolóból, ahelyett hogy a QGIS reprezentációval egyező geometriát keresne.

3.5. A QgisWorkingCopy

3.5.1. A hiányzó információk biztosítása

Az AEGISben létrehozott kibővített munkapéldány önmagában még nem elég ahhoz hogy a QGIS projektek verziókezelésére alkalmas legyen, mivel nem képes a rétegek adatainak és változásainak követésére, valamint nincsen megoldása arra

se, hogy a különböző featureökhöz rendelje a benne tárolt geometriákat. Ezeket a funkciókat a QgisWorkingCopy Python osztály biztosítja az úgynevezett QWCData állomány segítségével. Ebben egy extraData névre keresztelt objektum található amely az alábbi formátumot követi: Az egész objektum egy Python szótár, amelyben a kulcsok a verziószámok, a hozzájuk tartozó objektumok pedig a projekt adott verzióhoz tartozó állapotát írják le. Ebben az állapotleíróban a keyDict a featureök "Qgis id (ld. 3.12. szakasz)"-jához rendelve tárolja az Aegisben tárolt geometriák kulcsait, a layers pedig a rétegek betöltéséhez szükséges adatokat tartalmazza: az azonosítót, a geometriatípust és a referenciarendszert.

3.5.2. A QGIS munkapéldány funkciói

Extra adatok nyilvántartása

a QgisWorkingCopy tükrözi az AEGIS munkapéldány wrapper funkcionalitását, kiegészítve azokat az extra adatok karbantartásával. Példányosításakor ellenőrzi, hogy létezik-e a verziókezelő állományainak tárolására mappa, ha nem létrehozza, majd példányosítja a wrappert, szimpla fájl alapú tárolóval és kétirányú deltákat használó revízió kontrollerrel. Ezek után betölti az azonos mappában tárolt QWCData fájlból a szükséges további adatokat. A geometriák hozzáadása, módosítása és törlése egyszerűen a WorkingCopyWrapper azonos metódusait hívja meg, azonban a commit függvény hívásakor el kell tárolnia az extra adatokat. Ezen információk a Q-Aegis futása közben gyűlnek össze mely folyamat később lesz részletezve. A legtöbb adat kezelése nyílvánvaló, viszont a feature id-k karbantartása, valamint a verziók betöltése érdemel némi külön figyelmet.

A feature id probléma

A QGIS a rétegeken lévő objektumok azonosítóit a réteg tárolásától függően különböző módon kezeli (amire a dokumentációja nem sok említést tesz). Amennyiben memóriában tárolt a réteg , a *feature*-ök indexelése 1-től kezdődik, és a műveletektől függetlenül nem változik. *Shapefileban* tárolt adatok esetén viszont 0-tól indul az indexelés, és objektum eltávolítása esetén az összes olyan azonosító, ami nagyobb volt nála, csökken eggyel, hogy egy folyamatos azonosító állapot maradjon. Ezért

amikor *feature* eltávolításokat kezelünk, és a réteg tárolója shapefile volt, minden törlés után az összes eltárolt nagyobb azonosítót dekrementálni kell eggyel.

Verzió betöltése, rétegek kezelése

Egy kiválasztott verzió betöltése a munkapéldányon keresztül történik, először a *workingcopy*-t a megjeleníteni kívánt verzióra állítjuk, majd ebből lekérjük az aktuális geometria állapotot, amit a geometriák egyedi azonosítóiból és magukból a geometriákból álló párok formájában kapunk meg, C# szótár formájában. Ezt az adott verzióhoz tartozó extra adat objektum segítségével feldolgozzuk olyan módon, hogy rendelkezésünkre álljon az összes megjeleníteni kívánt réteg létrehozásához szükséges adat, valamint összepárosítjuk a `keyDict` segítségével a geometriákat a layereken kirajzolandó featureökkel. Ezt követően a rétegadatok alapján létrehozzuk a valódi QGIS rétegeket, feltöltjük őket az adott verzióban lévő állapotukkal, majd ezeket jelenítjük meg a QGIS grafikus felületén.

3.6. A QAEGIS osztály

A tényleges beépülő modul maga a QAEGIS Python osztály, melynek váza a QGIS-hez írt plugin builder beépülővel lett generálva. Ez biztosítja, hogy az alap szerkezete megfelelő legyen az alapprogramba való integráláshoz, és minden szükséges metaadattal rendelkezzen. Ebben az osztályban férünk hozzá a QGIS interfészéhez, amin belül az aktuális projekt példánnyal dolgozik a modul. Mivel a QGIS grafikus felülete a Qt szoftverkönyvtáron alapul, ezért az események kezelésére az úgynevezett *szignálokat* használjuk fel. A szignálokat a felhasználó tevékenységei váltják ki, és egy adott információhalmazt adnak, amire rákötethetünk saját implementálású feldolgozófüggvényeket. A szignálok és kezelésük áttekintésével könnyen megérthető a beépülő modul működése.

3.6.1. Szignálok és feldolgozásuk

A QGIS interfész projekt megnyitási szignál

Ezen jelzést használja a modul a munkapéldány megnyitására, amely folyamat vagy létrehoz egy új üres revíziókezelőt, vagy ha létezik a megnyitott projekthez egy,

betölti azt. Ezen szignál kezelése során jönnek létre azok a listák is amik követik az aktuális szerkestésekben a rétegek és featureök létrehozását és törlését.

A projekt példány réteg szignáljai

Az aktuálisan megnyitott projekt réteg létrehozásakor vagy törlésekor egy jelzés formájában ad információkat a végzett műveletről. Ezeket kezelve kerülnek a bufferlistákba az elvégzett műveletek. A bufferekre azért van szükség, mert a projekt mentéséig nem tekinthető véglegesnek a réteg jelenléte vagy eltávolítása, ezért amíg nem mentett állapotban vannak, nem kerülnek be a revíziókezelésbe. A réteg törléséhez tartozó szignál kezelésekor nem csak a rétegadatok eltávolítása kerül a bufferbe, hanem az összes rajta lévő objektum törlése is, így ha a layer már nincs jelen, a tartalma se marad felesleges adatként a rendszerben.

A projekt mentésének szignálja

A projekt mentésekor tekinthető ténylegesen elvégzettnek egy layer törlése vagy felvétele, ezért ezen jelzés kiváltásakor kerülnek ezek a módosítások a munkapéldányba. Új rétegek esetén a rajtuk végzett geometriaműveletek mentésig nem relevánsak -hiszen ha elvetnénk a mentésüket, az összes rajtuk lévő objektum is törlésre kerülne-, így mentés után kerül be minden feature is új geometriaként az AEGIS rendszerbe.

Réteg mentésekor kiváltott jelzések

Fontos megjegyezni, hogy a rétegek rendelkeznek az egyes műveletek végrehajtásakor megjelenő szignálokkal is, viszont ezek nem biztos hogy mentésre is kerülnek, így hasonlóan a rétegműveletek bufferéléséhez, ezek is egy köztes állapotban tárolódnak a véglegesítésükig. Szerencsére ezt a funkciót maga a QGIS biztosítja, így a függőben lévő műveletek tárolását nem szükséges újraimplementálni.A rétegen végzett műveletek mentésekor három különböző jelzés kerül feldolgozásra.

- **Hozzáadott featureök mentése**: Ezen szignál szolgáltatja az adatokat az újonnan létrehozott geometriákról, így ezen keresztül kerülnek be az új téradatok a verziókezelésbe. Érdemes kiemelni itt az alábbi kódrészletet :

```
1  geomToAdd = feature.geometry()
2  for layer in self.iface.mapCanvas().layers():
3    if layer.id() == layerId:
4      layerType = layer.wkbType()
5  if layerType in range(4,7) :
6    geomToAdd.convertToMultiType()
```

3.4. forráskód. A multigeometria réteg probléma kezelése

Erre azért van szükség, mert amennyiben a réteg multigeometriákkal dolgozik, új geometria létrehozásakor -valamilyen ismeretlen okból- még szimpla téradatként kezeli azokat, viszont bármilyen módosítást végzünk rajtuk, átalakítja az egyszerű alakzatokat egyelemű többszörös változataikká. Ez komoly problémákat tud okozni a módosítások kezelésénél, hiszen követetlenül változik meg a geometria típusa. Ezért ellenőrzi a program, hogy a réteg geometriatípusa többszörös-e, amit a 4,5,6 kódokkal jelöl, és ha igen, akkor még az új adat bekerülése előtt konvertálja azt multigeometria formába.

• **Eltávolított featureök mentése**: A geometriák eltávolításakor problémás, hogy az újonnan létrehozott adatokat mentésükig ideiglenes azonosítóval kezeli a QGIS, és mentéskor ezeket az ideiglenes változatokat törli, és felveszi véglekesként. Mivel az ideiglenes verzióik nem szerepelnek a verziókövetésben, így ha az eltávolításukat próbálnánk feldolgozni, hibára futnánk. Ezt kiküszöbölendő kihasználjuk az ideiglenes featureök egy tulajdonságát. Amíg a már véglegesített objektumok tárolás módjától függően 0-tól vagy 1-től kezdve kapnak azonosítót, ideiglenes verzióik a negatív irányba kapnak ID-t, ezért ha egy feature id-ja 0 vagy 1 alatti, ideiglenesnek tekinthető. Ennek felhasználásával megfelelő módon eltávolíthatóak a ténylegesen a felhasználó által törölt téradatok.

• **Módosítótt geometriák mentése**: A módosított geometriák kezelése elég egyértelmű mivel a valódi változáskezelést az AEGIS kiegészítései biztosítják, az egyedüli plusz ellenőrzés ami belekerült azt vizsgálja, hogy a módosított geometria mentetlen rétegen foglal-e helyet, és ha igen, nem tekintjük módosításnak, hiszen még nem került be a rendszerbe.

A projekt bezárásának szignálja

A projekt bezárásakor fontos, hogy az összes jelzésről leiratkozzon a program, ugyanis ennek hiányában a QGIS implementációjából kifolyólag egy másik projekt megnyitásakor a különböző szignálok többszörösen kerülnének kezelésre, ami nem kívánt működéshez vezet.

3.6.2. A kezelőfelület és eseményei

A kezelőfelület alapja a plugin generátor által létrehozott Qt ablak, amely Qt designer segítségével lett végleges formájára alakítva. A felület tervezésekor a fő szempont az volt hogy annyira egyszerű és intuitív legyen amennyire az csak lehetséges, így végül mindössze két elemmel vezérelhetővé vált a teljes verziókezelés

1. **A commit gomb**: Az aktuális verzió mentésére szolgáló gomb kettős funkciót tölt be. Az általánosabb, hogy ha már kezelt projekttel dolgozuk, megnyomásakor változtatásaink véglegesítésre kerülnek a revíziókezelő modulban, és létrejön egy újabb verzió. Másik funckióját abban az esetben látja el, ha egy létező, de még nem verziókezelt projekt van megnyitva. Ekkor megnyomásával a projekt teljes szerkezete, azaz rétegei és azok tartalma bekerülnek a friss munkapéldányba mint kiindulási verzió.

2. **A verzióválasztó lista vagy beviteli mező, és a hozzá tartozó betöltés gomb**: A verzióválasztó vezérlőelem nem jelenik meg a kezelőfelületen addig, amíg a munkapéldány nem rendelkezikbetölthető verzióval. Amennyiben már használható, a kívánt verziószámot kiválasztva és a betöltőgombra kattintva elindul a kiválasztott revízió megjelenítése. Mivel a verziómegjelenítés implementációjából fakadóan elveti az összes aktuális módosítást amit esetlegesen még nem mentett a felhasználó, ezért amennyiben vannak ilyen függőben lévő műveletek, a program felajánlja azok mentését egy új verzióba a betöltés megkezdése előtt.

   Megjegyzés. *A verzióválasztó elem attól függően, hogy mennyi verzió érhető el, két alakot vehet fel : kevesebb verzió esetén egy legördülő menü, bizonyos darabszám fölött pedig egy léptethető szám beviteli mező. Lista esetén egyértelműen nem választható ki hibás adat, léptetőmező esetén pedig megfelelő minimum*

és maximum értékek megkötésével biztosított, hogy nem ad meg a felhasználó nem létező verziószámot.

3.7. A program fájlszerkezete

3.7.1. A nyers programkód

A létrehozott programkód szerkezete szétválasztható a C#-ban valamint a Pythonban írt részekre.
A C# nyelven implementált kód az alábbi struktúrát követi:

1. Az AEGIS egyes funckinalitásai a megfelelő könyvtárakban található állományokban vannak
2. A geometriatranszformációk, valamint a transzformációgenerátorok osztályai az AEGIS Processing könyvtárának `QgisTransformation` alkönyvtárában kerültek mentésre
3. A munkapéldány wrapper osztálya az AEGIS Versioning `WorkingCopy` könyvtárában lett létrehozva, az általa használt eredeti `WorkingCopy` mellett

A Python kód a qaegis mappában található, az alábbi szerkezetel:

1. A plugin magját adó `QAEGIS` osztály definícióját, valamint a `QgisWorkingCopy` osztályt a qaegis.py állomány tartalmazza
2. A szükséges konverziók implementációi a converters.py fájlban találhatóak
3. A kezelőfelület leírása a `qaegis_dialog.ui` állományba, a működtetéséhez szükséges kód az azonos nevű de Python kiterjesztésű fájlba került

3.7.2. A működéshez szükséges binárisok

A beépülő modul működéséhez szükséges az AEGIS csomag dllekké fordított verziója a plugin könyvtáron belüli aegis mappában kapott helyet, a lefordított Python.NET CLR modul pedig az azonos szülőmappájú `pythonnet` könyvtárban található.

3.8. A működésben lévő beépülő modul könyvtárszerkezete

A használatban lévő plugin a QGIS Python beépülőket tartalmazó mappájában található, ebben a könyvtárban van minden szükséges állomány. A tényleges verziókezelést biztosító tárolók a kezelt projektet tartalmazó mappában kerülnek létrehozásra, egy `projektnév_qaegis_repository` nevű mappában, ezen belül pedig a kezelt projekt nevével megegyező mappákban találhatóak a szükséges fájlok. Ezen belül a changesets mappa verziószámokkal elnevezett forrásaiban találhatóak a revíziókhoz tartozó változáslisták, a snapshots mappa tartalmazza az esetlegesen mentett verzióállapotokat, a revisioncontrol fájl pedig az összes többi használt adatnak ad helyet. A QWCData állomány tartalma az összes AEGIS-en kívüli követendő adat szerializált formája.

3.9. A program fordítása

A program fordítása három lépésből áll: Az AEGIS fordítása, a Python .NET fordítása, végül pedig a plugin deploy.

3.9.1. Az AEGIS modulok fordítása

Az AEGIS-ben található források fordítása értelemszerű C# fordítás. A fordítás után keletkezett dll állományokat át kell másolni a plugin munkakönyvtárának aegis almappájába

3.9.2. A pythonnet fordítási környezet előkészítése

A QGIS architektúrája miatt -ami saját belső verziót használ a python nyelvből- a *Python .Net* clr modul fordításához több előzetes lépést is végre kell hajtani:

1. A QGIS-t az OSGeo4W telepítőn keresztül kell installálni, a setuptools, pyrcc5 csomagokkal együtt

2. Az OSGeo4W shellt megnyitva futtatni kell a bin könyvtárban található *py3_ env.bat* valamint *qt5_ env.bat* fájlokat

3. ezután a setuptools által biztosított easy_install parancs segítségével telepítsük a pip-et

4. a pip segítségével telepítsük a wheel modult.

Ezek után lépjünk be a pythonnet könyvtárba, és fordítsuk le a következő paranccsal : `python setup.py build_ext -inplace`

Megjegyzés. *A pythonnet fordításához a felsorolt lépéseken kívül szükséges még a .Net keretrendszer 3.5-ös verziója, ami már nem alapértelmezetten része a Windows operációs rendszernek, valamint szükséges a Windows SDK vagy egy Visual Studio jelenléte is a számítógépen. Mindezeken felül a pythonnet fordítása során használt* UnmanagedExports *modul miatt a rendszer nyelve angol kell hogy legyen, ez azt jelentheti, hogy az operációs rendszert újra kell telepíteni angol nyelven.*

3.9.3. A plugin fordítása (*deploy*)

Ha lefordítottuk az AEGIS és Python .Net modulokat, akkor a `pb_tool deploy` paranccsal fordíthatjuk le a python kódból valamint ui fájlokból származó forrásokat. Ez a parancs be is másolja a QGIS plugin könyvtárába a beépülő modult.

3.10. Tesztelés

A tesztelés hagyományos feketedoboz típusú tesztelésként történt, az alábbi tesztelési terv mentén:

Végrehajtott művelet	*Várt viselkedés*
Megnyitott projekt nélkül a verziókezelő ablak megnyitása	Az ablak nem nyílik meg, a felhasználó kap egy üzenetet, hogy projekten kívül nem használhatóak a verziókezelési funkciók
Új projekt létrehozása mentés nélkül	Hibaüzenet jelenik meg, ami jelzi a felhasználónak, hogy projektfájl nélkül nem használható a verziókezelés

Végrehajtott művelet	*Várt viselkedés*
Új projekt létrehozása mentéssel	A projekt mentésre kerül, elérhető a verziókezelő ablak, a verzióválasztó és a betöltőgomb nélkül
A commit gomb megnyomása a legutóbbi verzión álló verziókezeléssel	Az előző verzió óta végrehajtott változtatások mentésre kerülnek, bezárul a verziókezelő ablak, a felhasználó kap egy üzenetet, hogy a mentés sikeres.
A commit gomb megnyomása régebbi verzión álló verziókezeléssel	A változtatások mentése nem történik meg, a felhasználót hibaüzenet értesíti arról, hogy csak a legutóbbi verzión menthetőek módosítások.
Nem verziókezelt projekt megnyitása, majd a commit gomb megnyomása	A projekt megnyitásakor létrejön egy üres tároló, a commit gomb megnyomása után az összes réteg és tartalmuk bekerül a verziókezelésbe
Verziókezelt projekt megnyitása	A projekt betöltése után a verziókezelő a legutóbbi állapotra áll, elérhető a korábbi verziók betöltése, valamint a módosítások mentése.
Korábbi verzió betöltése, mentetlen módosítások nélkül	A régebbi állapotnak megfelelően létrejönnek a rétegek és a rajtuk lévő geometriák, a verziókezelő ablak bezárul.
Korábbi verzió betöltése, mentetlen módosításokkal	A program megnyit egy párbeszédablakot, amelyben figyelmezteti a felhasználót, hogy a verzió betöltésével a még nem commitolt módosításai elvesznek, és felajánlja ezek mentését. Ezután a verzió betöltése megtörténik.
Bármilyen aktuális verziószám változásával járó művelet	Az aktuális verziót feltűntető mező a megváltozottra frissül

Végrehajtott művelet	*Várt viselkedés*
Réteg hozzáadása a projekthez, majd a változtatások mentése	A réteg hozzáadása mentésre kerül, a létrehozás verziójától kezdve betöltésnél megjelenik
Réteg eltávolítása a projektből, majd a változtatások mentése	A réteg törlése mentésre kerül, a törlés verziójától kezdve betöltésnél nem jelenik meg
Geometriák módosítása, a réteg mentése, majd a változtatások commitolása	A geometriák módosítása mentésre kerül, minden verzióban az adott verziónak megfelelő állapotban kerülnek betöltésre
Geometriák módosítása, a réteg mentése nélkül, majd a változtatások commitolása	A módosítások nem kerülnek mentésre (mivel nem véglegesek)

3.1. táblázat. A tesztelési terv

3.11. Továbbfejlesztési lehetőségek

3.11.1. Az AEGIS továbbfejlesztési lehetőségei

Az AEGIS szerkezetéből adódóan számos lehetőséget biztosít további funkciók létrehozására:

1. A geometriákhoz tartozó egyéb metaadatok módosítására létre lehetne még hozni új transzformációs osztályokat, ezáltal ezek is kezelhetőek lennének

2. A módosítások kezelését tovább lehetne fejleszteni olyan adatok hozzáadásával, mint a szerkesző felhasználó azonosítója, ezzel alkalmasabbá téve a verziózást több ember által végzett projektek esetén is.

3. Esetlegesen a rétegrendszer implementálható lenne az AEGIS-en belül, mivel a legtöbb vektorgrafikus szoftver hasonló elven működik, így nem az alkalmazás részben kéne ezeket az információkat kezelni.

3.11.2. A Python modul továbbfejlesztése

1. A QAEGIS és QgisWorkingCopy osztályok erőteljesen az AEGIS keretrendszerre támaszkodnak, ezért annak bármilyen kiegészítése implementálható lenne ezen beépülő modulban való alkalmazásra is.

2. A kezelőfelület bővíthető lenne egy, a verziókezelők exportálására és importálására alkalmas funkcióval, így nem kéne az állományok másolását alkalmazni projekt betöltésére.

3. Mivel a QGIS a geometriák megjelenítésének vizuális részleteit, például a színüket, a rétegekhez kötve szabályozza, a rétegek stílusának követésével pontosabb állapotbetöltésre lenne lehetőség

3.12. Fogalomtár

Feature : A QGIS geometriai egysége, a layeren tárolt geometria és az összes hozzá tartozó egyéb adat.

Geometriatípus : A szabványos geometrialeírás típusainak egyike, lehetséges értékei Point, MultiPoint, LineString, MultiLineString,Polygon,MultiPolygon

Geometria komponenseinek száma : A geometria közvetlen részgeometriáinak száma. Vonal esetén a pontjainak száma, sokszög esetén 1 (azaz a héj számossága) + a hozzáadott lyukak száma stb.

Geometria részgeometriái : Pont geometriának nincs részgeometriája. Vonal geometria részei a pontok amelyek összekötésével megkapjuk a vonalat. Sokszög esetén az alakzat külső héja, valamint a síkidomon lévő lyukak, ezek vonalakkal írhatóak le. Multigeometriák részgeometriái az egyes különálló részeik.

Layer :A QGIS megjelenítési rétege, kötött geometriatípussal, projekten belül tetszőleges darabszámú létezhet

MultiGeometria : Olyan geometria, amely több különálló geometriát tartalmaz, de egy egységként van kezelve

Project : A QGIS munkapéldánya, rétegeket lehet hozzáadni, és azon geometriákkal lehet műveleteket végezni

WKB,WKT : Well Known Binary, Well Known Text, geometriák leírására használt szabványos bináris és szövegformátum

Working Copy : A munkapéldány a verziókezelésen belül, egy adott verzió állapotát tartalmazza, a legfrissebb verzión szerkeszthető

Qgis id : A featureök egyedi beazonosítására használt karakterlánc, a réteg azonosítójából és a feature id-jából áll elő

Szótár adatszerkezet : Kulcs-érték párokat tartalmazó lista. C#-ban kötött a kulcsok és értékek típusa, Pythonban akár elementként eltérhet.

4. fejezet

Összegzés

A fejlesztési folyamat végére sikeresen kialakult egy olyan verziókezelő bővítmény a *QGIS*-hez, amely a szerkesztési funkciók java részét támogatja, a műveletek követése hatékonyan megtörténik. A feldolgozási és tárolási folyamatokat jól sikerült elválasztani a megjelenítéstől. Az *AEGIS* csomag új funkcióinak implementálása nem csak lehetővé tette a beépülő modul létrejöttét, de jó kiindulási pontként is szolgálhat továbfejlesztési projektekhez. A *Python .Net* modul kifogástalanul használhatónak bizonyult, valamint a *QGIS* python interfészének, valamint belső adatreprezentációjának egyedisége se bizonyult túlzottan problémásnak.

Meglátásom szerint a kitűzött célokat sikerült elérni, és bár egyértelműen továbbfejleszthető, a végleges szoftver egy jól használható program lett, ami megfelel az összes vele szemben támasztott elvárásnak.