Liite 1: VOOKA-projektin ETL-dokumentaatio

Sisällys

1. Yleistä

ETL-prosessi toteutettiin Python-ohjelmointikieltä hyödyntäen ja kaikki kehitetyt ohjelmakoodit ovat MIT-lisenssin piirissä avoimesti saatavilla hankkeen GitHub-sivuilta. GitHubiin on listattu ETL-työkalun tekniset vaatimukset (requirements.txt), joista tärkeimpinä moduuleina toimivat GeoPandas sekä Shapely. Prosessin metodologia yhdistelee sekä algoritmisuutta että heuristista lähestymistapaa.

Ohjelmallinen työnkulku on kuvattu kokonaisuudessaan alla. Erilliset vaiheet taas on kuvattu tarkemmin omissa kappaleissaan. Yleiskuvauksena ETL-prosessi koostui kuudesta päävaiheesta (linkit GitHub-sivuille):

1. Tiedonkeruu ja esikäsittely
2. Tiedon yhdistely
3. Tiedon korjaukset
4. Tiedon vertailu
5. PDF-linkityskonversio
6. Kaavatietomallin implementointi

2. Tiedonkeruu ja esikäsittely

2.1 Rajapinnat

ETL-työkalussa tietojen kerääminen on mahdollista teknisesti kolmesta eri lähteestä:

1. OGC:n WFS-standardin mukaisista rajapinnoista (kuntien kaava-aineistot),
2. Esrin ArcGIS Feature Layereista (kuntien kaava-aineistot) sekä
3. OGC API Features -rajapinnoista (MML:n kiinteistötiedot)

Kaksi ensimmäiseksi mainittua palauttavat syötetyn URL:n mukaisen tiedon kokonaisuudessaan GeoPandas GeoDataframena. OGC API Features -skripti sen sijaan palauttaa API:n sisältöä GeoJSON-formaatissa tarkasteltavan maakunnan rajauksella (bounding box), joka on kovakoodattu toteutukseen. Jatkokehityksessä kovakoodattu maantieteellinen rajaus tulee korvata halutun alueen koordinaateilla. OGC API Features -skriptiin on myös lisätty apufunktioita, joiden avulla MML:n kiinteistötietojen GeoJSON-muotoinen dataskeema voidaan normalisoida tabulaariseksi tiedoksi Pandas DataFrameen.

ETL-työkalun hyödyntämisessä on myös hyvä huomioida, ettei vaihe 1.1 toimi kaikkiin WFS-rajapinnoista tehtäviin aineistoirrotuksiin. Tämä havaittiin Pohjois-Savon VOOKA-toteutuksen yhteydessä. Virhe johtuu luultavasti ETL-työkalussa käytetystä Geopandasin versiosta, joka ei pysty käsittelemään rajapinnassa olevia geometrioita oikein. WFS-rajapinnoista voi kuitenkin tehdä manuaalisen aineistoirrotuksen tarvittaessa. Versiopäivitys on kirjattu työkalun jatkokehitystarpeisiin.

2.2 Tietojen hankkiminen url-linkeistä

Pohjois-Savon toteutuksessa huomattiin, että useilla kunnilla on nettisivuilla url-linkkejä, jotka johtavat suoraan projekteissa tarvittaviin asiakirjoihin (kaavakartta, kaavamääräykset). Tätä varten luotiin ETL-vaihe, jonka avulla voitiin tehostaa dokumenttien lataamista. Lataamista varten tarvitaan csv-taulukko, joka sisältää url-osoitteet. Prosessissa url-osoitteiden tiedostot ladataan automaattisesti haluttuun kansioon. Mikäli tiedosto ei ole ladattavissa kyseisestä osoitteesta, ohjelma antaa virheilmoituksen käyttäjälle.

HUOM! Lataa tiedostoja vain sellaisista osoitteista, jotka ovat luotettavia. Kehitetty koodi suorittaa latauksen automaattisesti kaikille csv-tiedostossa oleville osoitteille. On käyttäjän vastuulla varmistaa se, että url-linkit ovat luotettavasta lähteestä.

2.3 Esikäsittely

Erään kunnan osalta WFS-rajapinta oli toteutettu KuntaGML-formaatissa, jonka sisäänluku ei onnistunut perinteisin menetelmin. Ongelma ratkaistiin erillisellä XML-parser-skriptillä, jossa kaavatiedot irrotettiin suoraan merkintäkielen rakenteesta.

Osa kunta-aineistoista saatiin rajapintojen sijaan erillistoimitettuina CAD-piirroksina, joissa ominaisuustiedot oli sidottu pistegeometrioihin varsinaisten kaavarajojen sijasta. Ominaisuustietojen yhdistäminen kaavarajoihin toteutettiin erillisellä skriptillä.

MML:n KTJ-aineistoissa kuntaliitosalueiden kaavoille on ilmoitettu vanha kuntakoodi. Nämä päivitettiin vastaamaan voimassa olevaa kuntakoodia kattavan vertailun mahdollistamiseksi kunta-aineistojen kanssa.

2.4 PDF-linkityskonversio

PDF-kaava-asiakirjojen linkittäminen yhtenäisessä muodossa kaavojen paikkatietomuotoiseen indeksiin on mahdollista, jos tiedetään, mihin kaavaindeksitunnukseen kukin asiakirja kytkeytyy. Tässä ETL-työkalussa esitetyt automatisoinnit vaativat validointia, eli linkittäjän tuottama aineisto tulee tarkistaa osaltaan manuaalisesti. Lisäksi automatisaation avulla ei voida kokonaan poistaa tarvetta manuaaliselle linkittämiselle.

Työssä PDF-linkitystä varten koostettiin linkitystaulu, jossa jokaisella rivillä oli tieto kaavan indeksitunnuksesta (KTJ, kunnan aineisto tai molemmat) sekä kaavan dokumenttityypistä (esim. kaavakartta). Linkitystaulun (taulukko 1) skeema oli seuraavanlainen:

Taulukko 1. PDF-linkitystaulun ominaisuustietokentät ja niiden selitteet.

Ominaisuustieto	Selite
Kuntanumero	Kunnan virallinen kuntakoodi.
Kunnan kaavatunnus	Kunnan paikkatietomuotoisen kaava-aineiston kaavatunnus.
KTJ-indeksitunnus	KTJ-aineiston indeksitunnus (kaavatunnus_1).
Original filename	Kunnalta saadun kaavaliitteen alkuperäinen tiedostonimi.
New filename	Sarake uudelle tiedostonimelle
Kaavalaji	Kaavalajikoodisto ylätasolla.
Manuaalisesti tarkistettu	Onko aineisto tarkistettu manuaalisesti (boolean)
Dokumentin tyyppi	Dokumenttityyppikoodisto, numeerinen
Match equivalency %	Automaattisen linkityksen vastaavuusprosentti
Huomioita	Kenttä havaintojen kirjaamista varten
Multipage	Onko asiakirjasssa useampi kuin yksi sivu (boolean)
Tila	Koodisto, joka kertoo, onko kaava-asiakirja validi.
Voimassa oleva	Onko kaava-asiakirja voimassa vai ei (boolean).
Geometry origin	Geometrian lähdeaineisto
Virhetyyppi	Kuvaus mahdollisesta havaitusta virheestä, numeerinen
Kuvaus	Kaavatietomallia varten geometry origin ja virhetyyppi -kentistä generoitu kuvaava teksti

Luokiteltujen ominaisuustietojen kuvaukset:

Kaavalaji-koodisto:

- ak
- rak
- yk

Dokumentin tyyppi -koodisto (sanallinen ja numeerinen):

- 1 = kaavakartta (sis. merkinnät ja määräykset)
- 2 = kaavakartta (ei sis. merkinnät ja määräykset)
- 3 = merkinnät ja määräykset (erillisenä)
- 6 = muu

Tila-koodisto:

- ok
- ei ok

Geometry origin-koodisto:

- kunta
- KTJ
- digitoitu VOOKA:ssa
- ei geometriaa

Virhetyyppi-koodisto:

- 0 = ei virhettä
- 1 = Virhe/puute asiakirjassa
- 2 = Virhe/puute rajauksessa
- 3 = Virhe/puute asiakirjassa sekä rajauksessa
- 4 = Muu virhe/puute
- 5 = Ei indeksitunnusta

2.4.1 Linkitystauluun tehdyt muutokset

Pohjois-Savon toteutuksessa lisättiin sekä poistettiin linkitystaulusta dataa. Projektissa kerättiin kaavakartat sekä määräykset, joten selostukset ja osallistumis- ja arviointisuunnitelmat jätettiin PDF-linkitystaulusta pois. Tauluun lisättiin match equivalency-prosentti. Se kertoo automaattisen linkityksen vastaavuudesta ja auttaa manuaalisessa tarkistuksessa. Jos prosentti jää pieneksi, on ehdottoman suositeltavaa tarkistaa, että vastaako asiakirja indeksiä. Lisäksi automatiikka saattaa generoida väärän kanavatunnuksen. Tämäkin voidaan päätellä matalasta vastaavuusprosentista. Automaattinen linkitys ei siis ole täysin luotettava ja vaatii jonkin verran manuaalista tarkistamista ja validointia.

Lisäksi tauluun lisättiin seuraavat uudet ominaisuustietokentät: virheluokitus, geometry origin sekä kuvaus. Virheluokitus-sarake kertoo, onko aineistossa (joko indeksi tai asiakirja) havaittu mahdollinen virhe, joka olisi syytä tarkistaa. Virheet luokiteltiin tyypeittäin asiakirjaa, rajausta, näitä molempia, muuta virhettä sekä indeksitunnuksen puuttumista koskeviin virheisiin. Projektissa virheiden kirjausta hyödynnettiin aineiston linkittämiseen ja validoimiseen. On kuitenkin huomattava, että kunnat ovat oman alueensa kaavoituksen asiantuntijoita, joten virheiden tarkistus käytännössä jää kuntien vastuulle.

Geometry origin -sarake kertoo, mistä indeksiaineiston geometria on peräisin. Tämä on oleellista kunnille, jotta he voivat tarkastaa oman aineistonsa laadun. Kuvaus-sarake sisältää geometrian lähteen sekä

virheluokituksen samassa sarakkeessa tekstimuotoisena. Kuvaus-tietokenttä löytyy sekä GeoPackage-muotoisesta paikkatietoaineistosta että lopullisesta kaavatietomallin mukaisesta JSON-tiedostosta, johon on koottu lisäksi kaavaindeksiä vastaava KTJ-tunnus.

2.5 Kaava-asiakirjojen hakemistorakenne

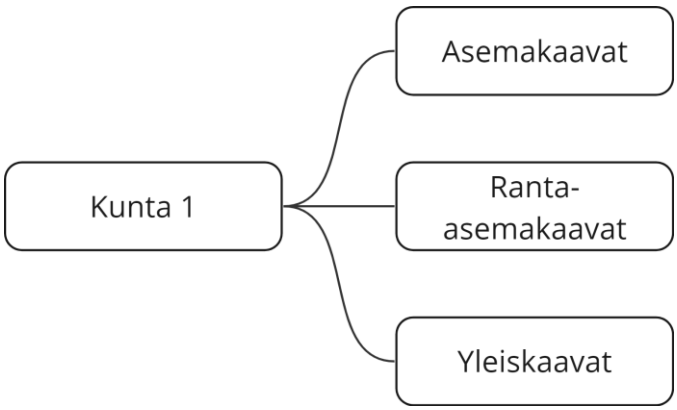
Kunnilta saadut kaava-asiakirjat tallennettiin vakioituun hakemistorakenteeseen resurssienhallintaan. Hakemistorakenteen runko oli muotoa:

```
documents
|
└──kuntakoodi
|   |
|   |
|   └──ak
|   |   asiakirja.pdf
|   |   asiakirja2.pdf
|   |   ...
|   └──rak
|   |   asiakirja.pdf
|   |   asiakirja2.pdf
|   |   ...
|   └──yk
|     asiakirja.pdf
|     asiakirja2.pdf
|     ...
 ...
```

ETL-työkalu hyödyntää kyseistä hakemistorakennetta tiedostojen uudelleen nimeämisessä!

2.5.1 Automaattinen kansiolajittelu

Pohjois-Savon toteutuksessa kehitettiin automaattinen kansiolajittelu niin asiakirja- kuin indeksiaineistolle. Projektissa kunnat latasivat aineistoa Sharepoint-kansioon, jonka sisällä aineisto saattoi olla useissa eri alikansioissa. Tällöin oikeiden tietojen etsiminen rakenteesta on haastavaa ja aikaa vievää. Ratkaisuksi kehitettiin automaattinen kansiolajittelija. Vähimmäisvaatimus koodin toimivuudelle on se, että kansiot on lajiteltu ennen koodin ajamista kuntakohtaisesti, minkä lisäksi kuntakohtaisen kansion sisällä alikansioiden tulee olla lajiteltu kaavalajikohtaisesti (kuva 1). Tämän jälkeen koodi valikoi halutut tiedostomuodot ja vie ne kaavalajikohtaisen kansion alle.



Kuva 1. Automaattisen kansiorakenteen lajittelijan vähimmäisvaatimus alkutilan kansiorakenteelle.

Koodi tunnistaa kaava-asiakirjoiksi PDF-tiedostot. Mikäli toimitetut kaava-asiakirjat ovat jotain muuta tiedostomuotoa, niin ohjelma ei käsittele niitä. Koodi tunnistaa indekseiksi SHP-, GPKG-, DWG- ja DXF-aineistot. Mikäli indeksiaineisto on jotain muuta kuin edellä mainittuja muotoja, niin ohjelma ei käsittele niitä.

Käyttäjän tulee lopuksi tarkistaa manuaalisesti, että lajitellut kohteet ovat päätyneet oikeaan kaavalajikansioon. Ohjelma ei myöskään osaa tunnistaa, onko tiedosto VOOKA-aineiston kannalta oleellinen, ts. kaavakartta tai määräys, vai jokin muu projektin kannalta ei-oleellinen tiedosto, esimerkiksi kaavaselostus. Käyttäjän pitää automaattisen lajittelun jälkeen tarkistaa lajittelun tulokset ja mahdollisesti poistaa turhat tiedostot.

2.6 Kaavatunnuksella nimettyjen asiakirjojen linkitys kaavatunnukselliseen kunnan indeksiaineistoon

Projektin yhtenä päätavoitteista oli linkittää kaava-asiakirjat niitä vastaaviin kaavaindekseihin. Aikaisemmin Etelä-Savon pilottiprojektissa linkittäminen tehtiin täysin manuaalisesti. Pilotin aikana tunnistettiin linkittämisen tehostamistarve. Pohjois-Savon toteutuksessa ETL-prosessin alkuvaiheeseen luotiin vaihe 1.4.6, jonka avulla kaavatunnuksella nimettyjä asiakirjoja voidaan linkittää kunnan kaavaindeksiaineistoon, mikäli indeksiaineisto sisältää kaavatunnukset.

Ohjelma vertaa tiedostonimiä kunnan kaavatunnuksiin indeksiaineistossa ja linkittää ne toisiinsa, mikäli vastaavuus on tarpeeksi hyvä. Samalla ohjelma tuottaa vastaavuusprosentin.

Tiedot koostetaan csv-taulukkoon PDF-linkityskonversion linkitystaulun vaatimaan muotoon. Koska ohjelma ei tuota kuin tiettyjä arvoja, jää osa tietokentistä tyhjiksi. Ohjelmassa vastaavuusprosentin minimiarvo on määritelty melko alhaiseksi (35%), jotta linkitys olisi mahdollisimman tehokas virheistä huolimatta. Tämän vuoksi käyttäjän tulee tarkistaa linkitystaulusta ne rivit, joissa vastaavuusprosentti on pieni.

Tähän vaiheeseen kehitettiin myös dokumentin tyypin määrittelijä, joka päättelee tiedoston nimestä sen tyypin. Tämän dokumentin julkaisuhetkellä (2/2024) työkalu palauttaa dokumentin tyypiksi määräyksen (3) mikäli tiedoston nimi sisältää määräykseen viittaavaa sanastoa. Jos kyseistä sanastoa ei löydy, niin työkalu antaa dokumentin tyypin oletusarvoksi kaavakartan sekä määräykset (1). Sanasto, jonka avulla koodi määrittelee dokumentin tyypin, löytyy täältä: document_type.

HUOM! Tiedoston tyypiksi tulee määräys, jos kyseinen sana esiintyy tiedostonimessä. Tämä aiheuttaa taulukon tuloksiin virhettä ja vaatii dokumentin tyypin tarkistamisen manuaalisesti jossain tapauksissa. Kuitenkin tämä automatisointi helpottaa aineiston käsittelyä, mutta vaatii käyttäjältä tarkkuutta. Tämän vaiheen jatkokehittämistarpeita on kuvattu luvussa 8.

Automatisaation avulla siis voidaan helpottaa linkittämistä, muttei täysin poistaa tarvetta manuaaliselle tarkastelulle.

ETL-prosessiin kehitettiin myös oma vaiheensa sellaisille tapauksille, jossa tiedostonimessä ei ole kaavatunnusta ja kaavatunnuksellinen indeksiaineisto puuttuu. Tällöin voidaan ajaa vaihe 1.4.7. Ohjelma luo linkitystauluun tarvittavat tiedot, mutta ei linkitä asiakirja-aineistoa indeksiaineistoon.

3. Datan yhdistely ja korjaukset

3.1 Yhdistetty kaavadata

Kunnilta saadut kaava-aineistot vaihtelivat sisällöiltään valtavasti. Stabiilin vertailun ja validoinnin mahdollistamiseksi ETL-työkalussa luodaan sekä KTJ- että kunta-aineistoille yhdistetty kaava-aineisto yhtenäisellä skeemalla (taulukko 2). Tiedot tallennetaan omina tasoinaan yhteisen master-geopackagen alle. Tasot ovat seuraavat:

- Asemakaavat_kunta
- Asemakaavat_ktj
- Yleiskaavat_kunta
- Yleiskaavat_ktj
- Kuntien_rekisterinpitoalueet_ktj (tarvittaessa)

Asemakaava-tasot kattavat sekä asema- että ranta-asemakaavat.

Kaavalajit on ilmoitettu KTJ-aineistossa valmiina Kiinteistötietojärjestelmän kiinteistörekisterin koodistojen mukaisesti. Ne muunnetaan ETL-työkalussa vastaamaan Rakennetun ympäristön tietojärjestelmän koodeja[1]. Kuntien aineistoissa kaavalajeja ei ole ilmoitettu erillisellä ominaisuustietokentällä. ETL-päättelee nämä aineiston kaavaselitteistä. Mikäli selitteitä ei ole aineistoissa kerrottu, kaavalajeiksi ilmoitetaan oletuksena yleiskaavoilla 23 = osayleiskaava, ranta-asemakaavoilla 33 = ranta-asemakaava ja asemakaavoilla 31 = asemakaava.

[1] https://koodistot.suomi.fi/codescheme;registryCode=rytj;schemeCode=RY_Kaavalaji

Taulukko 2. Yhdistetyn kaavadatan ominaisuustiedot ja niiden selitteet. Rasti kuvaa kunta- ja KTJ-aineiston osalta sitä, onko ominaisuustieto mukana aineiston skeemassa.

Ominaisuustieto	Selite	Kunta-aineisto	KTJ-aineisto
FID	Rivin yksilöivä tunniste	X	X
originalref	Lähtöaineiston alkuperäinen koordinaattijärjestelmä epsg-koodina	X	X
vanhakuntakoodi	Kuntaliitosalueiden vanha kuntakoodi		X
kuntakoodi	Tilastokeskuksen mukainen voimassa oleva kuntakoodi	X	X
kuntanimi	Kunnan nimi	X	X
kaavatunnus	Kunnan antama kaavatunnus	X	
kaavatunnus_1	KTJ-aineiston kaavatunnuksen alkuosa		X
kaavatunnus_2	KTJ-aineiston kaavatunnuksen loppuosa		X
kaavaselite	Kaavan tai kaavaindeksin selite/nimi	X	X
kaavalaji	Rakennetun ympäristön tietojärjestelmän koodistojen mukainen kaavalajikoodi	X	X
hyvaksymispvm	Kaavan hyväksymispäivämäärä	X	X
vahvistamispvm	Kaavan vahvistamispäivämäärä	X	X
voimaantulopvm	Kaavan voimaantulopäivämäärä	X	X
kohderekisteriyksikot	Lista kaavaindeksiin liittyvistä kiinteistötunnuksista	X	X
kaavakartta_ja_maaraykset	Kaavakartan ja määräyksen PDF-asiakirjan nimi tai hyperlinkki	X	X
kaavakartta	Kaavakartan PDF-asiakirjan nimi tai hyperlinkki	X	X

maaraykset	Kaavamääräysten PDF-asiakirjan nimi tai hyperlinkki	X	X
Kuvaus	Kuvaus geometrian lähteestä sekä mahdollisesta virhetyypistä	X	X

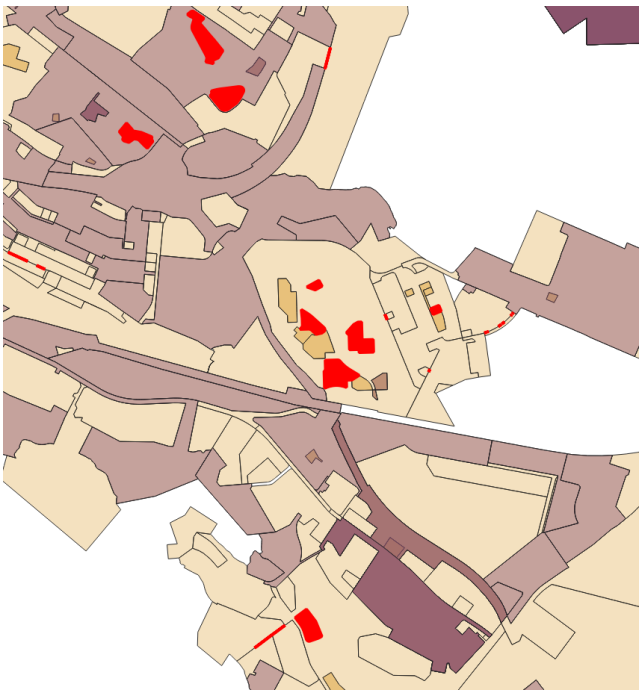
Yhdistetty kaavadata on keskeisessä osassa kaikissa ETL-työkalun vaiheissa, sillä koko toimintalogiikka perustuu muodostetun master-datasetin skeemaan.

3.2 Geometriakorjaukset

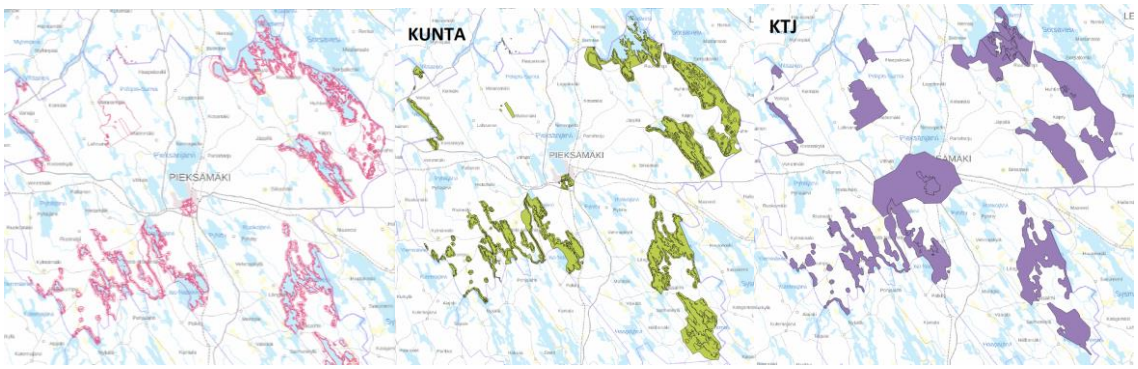
Geometriakorjauksissa ETL-työkalu nojaa vahvasti Pythonin Shapely-kirjastoon. Jokaiselle kaavaindeksille ajetaan explain validity -funktio, joka palauttaa tekstikenttänä tiedon, mikäli geometria itsessään on epävalidi. Esimerkiksi, jos kaavaindeksi risteää itseään tai sisältää ns. sliver-polygonin, funktio palauttaa tekstikentän “Ring Self-intersection” sekä ongelmallisen sijainnin koordinaatit. Mikäli kaavaindeksin geometria on virheellinen, se korjataan Shapelyn make valid -funktiolla. Joissakin tapauksissa em. Shapely-funktio ei pure geometria-ongelmaan (erityisesti CAD-aineistot), jolloin ongelmatapausten geometriat validoitiin QGIS-paikkatieto-ohjelmalla.

Usein alkuperäinen geometria tulee räjäyttää useaksi kohteeksi, jotta geometriasta saadaan kelpo. Jos räjäytyksessä tulee luoda useampi kohde samalla geometriatyypillä, palautetaan moniosainen geometria (MultiPolygon). Jos korjauksessa tulee luoda kohteita eri geometriatyypeillä, palautetaan GeometryCollection. Kaavojen tapauksessa vain aluegeometriat ovat sallittuja, joten ETL-työkalu käsittelee mahdollisesti esiintyvät GeometryCollectionit erikseen ja palauttaa ne alueiksi.

Eräässä kunnassa CAD-aineiston muunnos paikkatiedoksi tuotti kaavaindekseille päällekkäisiä “haamugeometrioita”. ETL-työkalu suodatti nämä päällekkäisyydet pois ominaisuustietovertailuun pohjautuen.

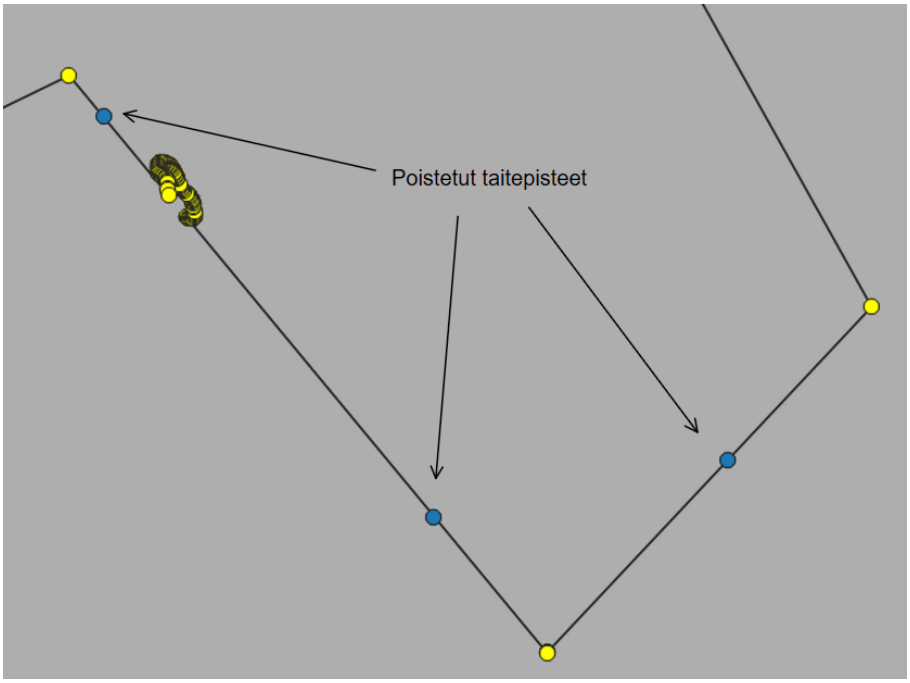


Kuva 2. Kuntien omat kaavaindeksigeometriat voivat olla haastavia. Tässä esimerkissä kunnan asemakaava- aineistossa oli 852 polygonia, joista löytyi 352 geometriavirhettä.



Kuva 3. Viivamaisia rajauksia on hankala automaattisesti muuntaa polygoneiksi, sillä tulkintaa joudutaan tekemään rajausten oikeellisuudesta. Vasemmalla alkuperäinen aineisto. Keskellä polygoneiksi muuntaminen ja oikealla KTJ:stä löytyvät rajaukset.

Pohjois-Savon toteutuksessa ETL-työkaluun kehitettiin vaihe, jolla voi poistaa ylimääräiset taitepisteet suoran viivasegmentin päätepisteiden välistä. Jotkin järjestelmät eivät nimittäin vastaanota dataa, joissa reunasegmenteiltään (edge) yhtyvillä vierekkäisillä geometrioilla suoralla viivalla sijaitsevat (periaatteessa turhat) taitepisteet poikkeavat toisistaan. Aineistolle ajetaan ETL-työkalussa removeUnnecessaryVertices-funktio, joka poistaa nämä ylimääräiset taitepisteet säilyttäen aineiston topologisen eheyden (kuva 4).



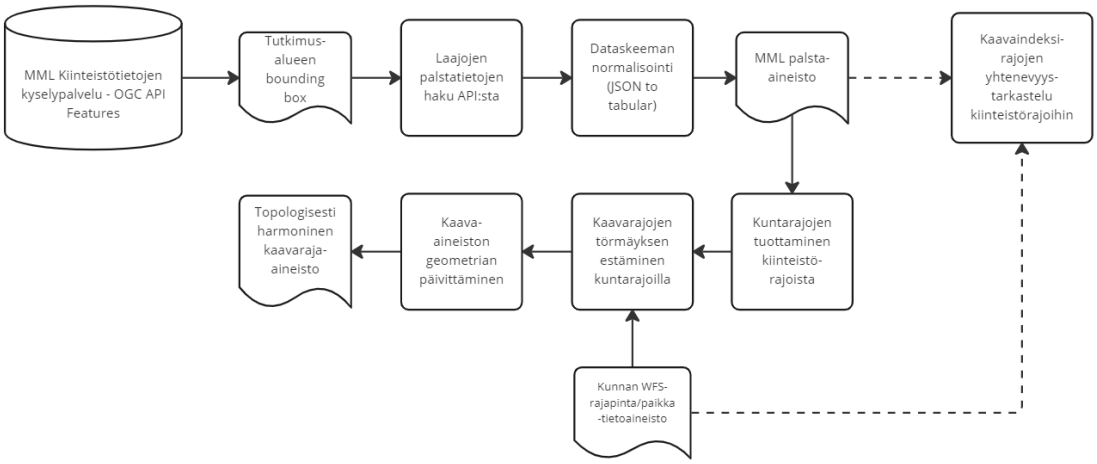
Kuva 4. Esimerkki korjatusta aineistosta. Kuvassa päällekkäin alkuperäinen sekä korjattu aineisto esitettynä Select by location -valinnalla, jossa valittu kaikki päällekkäin osuvat pisteet (keltaisella). Sinisellä on kuvattu poistetut ylimääräiset taitepisteet. Topologinen eheys säilyy myös CAD-aineistoista peräisin olevilla kaarimuodoilla, jotka koostuvat GeoPackage-vektoriformaatissa lyhyistä ja suorista viivoista.

3.3 Kohderekisteritunnusten linkittäminen

Kiinteistötunnuksien sijaintitiedot ovat saatavilla MML:n kiinteistötietojen OGC API Features -rajapinnasta omana pistetasonaan. ETL-muunnin linkittää jokaiselle kaavaindeksille tiedon siihen liittyvistä kiinteistötunnuksista, mikäli yksittäisen kiinteistötunnuksen sijaintipiste on kaavaindeksin ulkorajojen sisällä. Tämä tarkoittaa sitä, että monilla kaavoilla on useita kiinteistötunnuksia, jolloin nämä on listattu peräkkäin kaavan ominaisuustietoihin.

3.4 Yleiskaavojen kuntarajatörmäysten korjaus kiinteistörajoilla

ETL-prosessin yleistä-osuudessa on mainittu, että prosessin metodologia yhdistelee sekä algoritmisuutta että heuristista lähestymistapaa. Yleiskaavojen kuntarajatörmäysten korjaus kiinteistörajoilla edustaa jälkimmäistä - toteutus perustuu päättelyyn sekä datan arviointiin. Kuva 5 esittää prosessin vaiheet.



Kuva 5. Yleiskaavojen kuntarajatörmäyksen korjausprosessi. Katkoviivalla erotettu osuus on kuvattu tarkemmin osana geometris-topologista vertailua.

Yleiskaavarajojen ETL-muuntimen toimintalogiikka on seuraavanlainen:

1. Kunnalle muodostetaan kuntarajamaski MML:n palstatiedoista saatavista kiinteistörajoista. Käyttäjä voi halutessaan päättää, otetaanko eksklaavit huomioon vai ei (kiinteistörajat, jotka sijaitsevat varsinaisen kuntarajauksen ulkopuolella).
2. Iteroidaan kaavaindeksit yksittäin läpi ja tarkistetaan, onko kaavaindeksi muodostetun kuntarajamaskin sisällä. Jos indeksi on kokonaan maskin sisällä, ei tehdä mitään (ei ole tarvetta rajamuutoksille).
3. Jos kaavaindeksi ei ole täysin maskin sisällä, se leikkaa kiinteistörajaa osin tai on täysin maskin ulkopuolella (jälkimmäistä tapausta ei esiintynyt Etelä-Savon kunnilla). Tällöin on tarve rajan oikaisemiselle.
4. Lasketaan geometria niille osuuksille, joissa kaavaindeksi risteää kuntarajamaskin kanssa (=intersection).
5. Lasketaan geometria niille osuuksille, joissa kuntarajamaski eroaa kaavaindeksistä (=difference).
6. Filtteröidään difference-geometrioita. Hylätään ne diffrence-geometrian osat, jotka ovat vertailussa olevan kaavaindeksin ulkorajojen sisällä. Lisäksi hylätään ne difference-geometriaosuudet, joissa pinta-ala on poikkeavan suuri. Jokaisella kunnalla pinta-alan suuruudelle on olemassa toleranssi-raja-arvo, jotka on listattu ETL-skriptin docstringiin sekä Jupyter Notebookiin.
7. Lopuksi lasketaan muodostetun intersect-geometrian sekä filtteröidyn difference-geometrian liitos (=union), josta muodostuu käsiteltävän kaavaindeksin uusi geometria.

3.5 Ominaisuustietokorjaukset

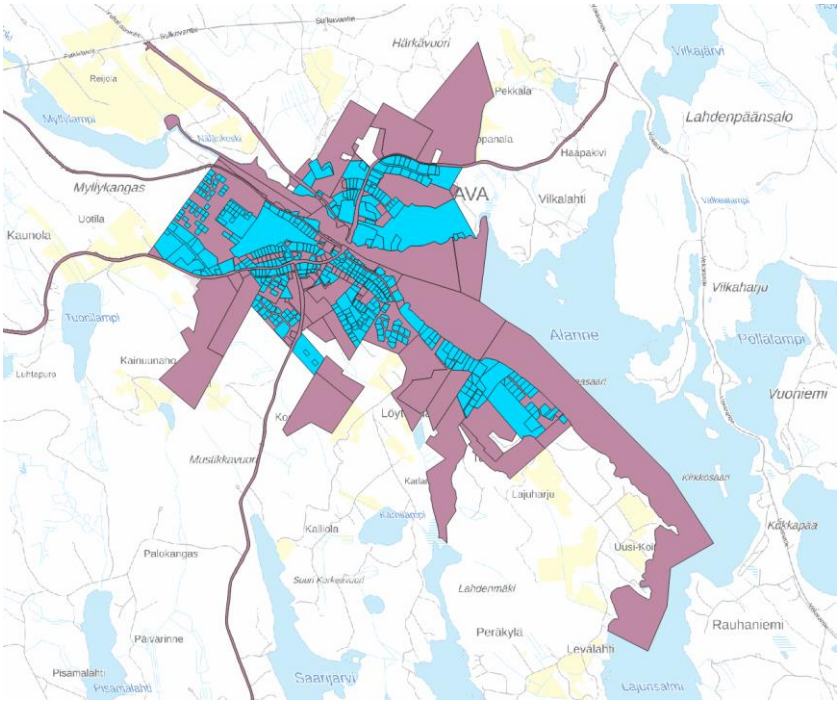
Päivämäärät (esim. kaavan hyväksymispäivämäärä, vahvistamispäivämäärä) oli ilmoitettu kunnilta saaduissa kaava-aineistoissa hyvin vaihtelevissa muodoissa sekä osin puutteellisina. Eri kunnilla oli käytössä päivämääräformaatteina mm. DD.MM.YYYY, YYYYMMDD sekä YYYY. Lisäksi tieto oli usein ilmoitettu päivämäärä-tietotyypin sijasta tekstikenttänä.

Osana ETL-prosessia muodostettiin funktio, joka kääntää tekstimuodossa olevan tabulaarisen tiedon date-tietotyypiksi muotoon YYYY-MM-DD. Mikäli lähtötietona on ainoastaan vuosi, päivämääräksi annettiin kyseisen vuoden tammikuun ensimmäinen päivä.

4. Geometris-topologinen vertailu

4.1 Kiinteistöpalstojen ja kaavaindeksien vertailu

Kiinteistöpalstojen ja kaavaindeksien vertailussa tuodaan esiin, miten hyvin kiinteistörajat yhtyvät kaavaindeksien rajoihin. Topologinen yhtenevyys ilmaistaan prosenttiarvona, joka lasketaan yksittäisen kiinteistöpalstan ja kaavaindeksin risteyskohdan (intersection) pinta-alan suhteena kiinteistöpalstan pinta-alaan (kuva 6).



Kuva 6. Esimerkki palstojen ja kaavaindeksien vertailusta. Sinisellä valittuna palstat, jotka vastaavat kaavaindeksirajoja vähintään 90 % tarkkuudella. Punertavalla palstat, joiden vastaavuus kaavaindeksirajoihin on alle 90 %.

4.2 KTJ- ja kunta-aineistojen vertailu

Geometris-topologinen vertailu tehdään ETL-työkalussa OGC:n DE-9IM -standardin mukaisesti (Dimensionally Extended 9-Intersection Model), jonka avulla tunnistetaan yhtenevät, päällekkäiset sekä toisistaan eriävät geometriat. Tämän lisäksi ETL-muunnin vertailee kaavarajojen muotoa ja pinta-aloja automaattisesti.

DE-9IM-mallissa kahden geometrian välinen relaatio ilmaistaan 9-merkkisellä mallilla (taulukko 3). VOOKA-projektissa ylivoimaisesti yleisin relaatiomuoto KTJ:n ja kunta-aineistojen välillä oli “212101212”.

Taulukko 3. Esimerkki DE-9IM relaatiomallista.

DE-9IM	Sisusta (Interior)	Raja (Boundary)	Ulkopuoli (Exterior)
Sisusta (Interior)	2	1	2
Raja (Boundary)	1	0	1
Ulkopuoli (Exterior)	2	1	2

Mallia luetaan seuraavasti:

- 2: Geometrioiden risteyskohta on alue (Polygon) eli kaksiulotteinen geometria
- 1: Geometrioiden risteyskohta on viiva (LineString) eli yksiulotteinen geometria
- 0: Geometrioiden risteyskohta on piste (Point) eli nollaulotteinen geometria

Muita DE-9IM-mallissa esiintyviä arvoja voivat olla:

- F: False - risteyskohtaa ei esiinny
- T: True - mikä tahansa ei-false arvo on sallittu (0,1 tai 2)
- *: kaikki sallitaan

Jos kahden geometrian välinen relaatiomalli vastaa muotoa “T*F**FFF*”, geometriat ovat täydellisesti samat. VOOKA-projektissa yhdelläkään KTJ:n ja kunta-aineiston vastaavalla kaavalla tämä ehto ei täyttynyt. Tämä johtunee siitä, että kaava-aineistojen digitointi on tapahtunut eri tavalla ja DE-9IM-malli ei salli pienintäkään murto-osan eroavaisuutta. **Tämän takia VOOKA-projektissa vastaavat kaavat tulkittiin geometris-topologisesti yhteneviksi, jos niiden topologinen tarkkuus oli vähintään 98 %.**

Topologinen tarkkuus ilmaistaan ETL-muuntimessa iou-arvolla. Lyhenne tulee sanoista intersection of union, geometrioiden yhtymäkohdan risteyskohta. Mikäli kaksi kaavaa ovat geometris-topologisesti tismalleen samanlaiset, niiden risteyskohdan sekä yhtymäkohdan pinta-alat ovat tismalleen samat. Näiden suhde on täten 100 %. Mikäli kaavoissa on geometris-topologista eroavaisuutta, niiden yhtymäkohta on erisuuri kuin risteyskohta. Tällöin yhtymäkohdan ja risteyskohdan pinta-alojen suhde myös eroaa 100 %:sta.

Geometris-topologista eheyttä ilmaistaan ETL-työkalussa kaiken kaikkiaan taulukossa 4 ilmaistuin kentin.

Taulukko 4. ETL-työkalussa tuotettavat, geometris-topologista eheyttä ilmaisevat tietokentät.

Ominaisuustieto	Selite
area_ha	Kaavan pinta-ala hehtaareina.
refe_area_ha	Vastaavan kaavan pinta-ala hehtaareina.
de9im_pattern	DE-9IM-mallin mukainen relaatio 9-merkkisellä mallilla.
topo_equal	Boolean-tieto. Onko vertailtavat kaavat topologisesti yhtenevät vai ei. Kyllä, jos iou-arvo on vähintään 98.
iou	Intersection of union -prosentti. Ilmaistaan vertailtavien kaavojen yhtymä- ja risteyskohdan suhteena.
refe_kaavatunnus	Vastaavan kaavan kaavatunnus.
a_delta_%	Vertailtavien kaavojen pinta-alojen suhteellinen virheprosentti.
false_a_delta	Boolean-tieto. 0 = virheprosentti ei ole false positive, 1 = virheprosentti on false positive.

VOOKA-projektin aikana kuntien datan ja KTJ-datan välillä ei ollut yksilöivää tunnistetta. Täten ominaisuustietojen pohjalta ei voitu tietää, mikä kunnan kaava vastaa KTJ-kaavaa ja päin vastoin. ETL-työkalussa ongelma on ratkaistu siten, että kaavan vastine toisessa datasetissa on se kaava, jonka iou-prosentti on suurin.

Geometris-topologinen vertailu on mahdollista toteuttaa VOOKA-pilotissa tuotetulla työkalulla joko kunta- tai KTJ-aineistopohjaisesti. On kuitenkin suositeltavaa, että KTJ-aineistoa käytetään vertailun pohjana, sillä aineisto on lähtökohtaisesti eheämpää monissa Suomen kunnissa!

4.2.1 Virheellisten aineistovertailutulosten tunnistaminen

DE-9IM-laskennnassa on mahdollista, että vertailun vastaava kaava voi olla ns. false positive. Tämä tarkoittaa käytännössä sitä, että pinta-alojen suhteelliset virheprosentit nousevat tuhansiin ja iou-prosentti on minimaalinen. Tällöin laskenta tuottaa virheellisen tuloksen, joka näyttäytyy oikeana. Käytännössä vastaavaksi kaavaksi ei hyväksytä kaavaa, jolla pinta-alojen suhteellinen virheprosentti ylittää 100 %, jolloin virheprosentti voidaan todeta false positiveksi. Pohjois-Savon toteutuksessa työkaluun kehitettiin osio, joka tunnistaa nämä virheelliset tiedot.

Tämä toteutettiin lisäämällä aineistoon uusi tietokenttä false_a_delta, joka kertoo, onko arvo false positive vai ei. Tietokenttä saa arvon 0, jos virheprosentti (a_delta_%) on alle 100. Jos virheprosentti on yli 100, tietokenttä saa arvon 1.

HUOM! Tämä ominaisuus kehitettiin vain KTJ-aineistopohjaiseen vertailuun, sillä kyseinen vertailutapa todettiin aineiston eheyden vuoksi kannattavammaksi.

5. Ominaisuustietojen vertailu

Siinä missä geometris-topologinen vertailu on mahdollista tehdä myös kunta-aineistoon pohjautuen, ominaisuustietojen vertailu on toteutettu VOOKA-pilotissa vain KTJ-pohjaisesti. KTJ- ja kunta-aineiston ominaisuustietojen yhdenmukaisuutta ilmaistaan ETL-työkalussa kaiken kaikkiaan Taulukossa 5 ilmaistuin kentin.

Taulukko 5. ETL-työkalussa tuotettavat, ominaisuustietojen yhteneväisyyttä ilmaisevat tietokentät.

Ominaisuustieto	Selite
kl_equal	Boolean-tieto. Onko KTJ-aineiston kaavalaji-tieto sama kuin kunta-aineistossa.
hyv_equal	Kertoo, onko KTJ- ja kunta-aineiston hyväksymispäivämäärä sama.
voim_equal	Kertoo, onko KTJ- ja kunta-aineiston voimaantulopäivämäärä sama.

Ensimmäinen ominaisuustietokenttä “kl_equal” kertoo boolean-tietona, onko vertailtavien aineistojen kaavalajit samat. Näissä on ajoittain paljonkin heittoa VOOKA-pilotin eri kunnissa, sillä:

1. KTJ kertoo asemakaavoissa usein, onko kaava ohjeellista tonttijakoa vai ei. Pilotin kunta-aineistoista tämä tieto puuttui.
2. KTJ:ssa ei ole tietoa oikeusvaikutuksettomista kaavoista. Tämä tieto oli myös VOOKA-pilotin kunta-aineistoissa harvinainen, mutta ajoittain esiintyvä.

Toinen ja kolmas ominaisuustietokenttä “hyv_equal” sekä “voim_equal” ilmaisevat hyväksymis- ja voimaantulopäivämäärien yhteneväisyyden sekä niihin liittyvät mahdolliset puutteet. Kentissä voi esiintyä:

1. “Hyvpvm/voimpvm puuttuu molemmista”
   - Sekä KTJ- että kunta-aineistosta puuttuu tarkasteltava päivämäärä kaavan osalta.
2. “Hyvpvm/voimpvm puuttuu KTJ”
   - KTJ-aineistosta puuttuu tarkasteltava päivämäärä kaavan osalta.
3. “Hyvpvm/voimpvm puuttuu kunta”
   - Kunnan aineistosta puuttuu tarkasteltava päivämäärä kaavan osalta.
4. 1 - Tarkasteltavat päivämäärät ovat samat
5. 0 - Tarkasteltavat päivämäärät eroavat toisistaan

Vahvistamispäivämäärät eivät ole mukana vertailussa, sillä ne puuttuvat KTJ-aineistosta kokonaan. Yleisenä periaatteena vertailussa mukana olevat ominaisuustiedot ovat ainoita, joita VOOKA-pilotissa kyettiin vertailemaan.

5.1 Ominaisuustietovertailun kehittäminen

Ominaisuustietovertailua kehitettiin Pohjois-Savon toteutuksessa niin, että se perustuu Pythonin Difflib-moduuliin. Päivämäärien kirjoitusasut voivat vaihdella huomattavasti ja lisäksi ne voivat sisältää turhia merkkejä: esimerkiksi välilyöntejä tai ylimääräisiä nollia. Koodia kehitettiin niin, että tietyn raja-arvon perusteella päivämäärät voidaan katsoa yhteneväisiksi. Tähän käytettiin Difflib:in SequenceMatcher luokkaa. Aikaisemmin koodissa päivämäärävertailu ei antanut kovinkaan hyviä tuloksia. Syynä tähän on se, että esimerkiksi KTJ-kuntavertailussa KTJ-aineiston päivämäärät ovat muotoa VVVV-KK-PP. Kunnan aineistoissa päivämäärät saattoivat olla jotain muuta muotoa, jolloin pelkässä yhtäsuuruuden tarkastelussa jää huomioimatta se, että päivämäärät saattavat olla yhteneväiset, mutta vain eri kirjoitusasussa.

HUOM! Tämä ominaisuus kehitettiin vain KTJ-kuntapohjaiseen vertailuun. Syynä tähän on se, että kyseinen vertailutapa on aineiston eheyden vuoksi kannattavampi.

Tämän projektin kuntien ja KTJ:n kaava-aineistojen päivämäärät vastasivat toisiaan hyvin monen kunnan tapauksessa. Päivämäärien ominaisuustietojen muuttamiseen kehitettyä työkalua piti kuitenkin täsmentää erittäin kirjavan päivämäärien kirjaamistavan vuoksi. ETL-työkalun jatkohyödyntämistä ajatellen erilaisia kirjaamistapoja on valtakunnallisesti lukematon määrä, joten työkaluun pitäisi kehittää runsaasti erilaisia skenaarioita, jotka huomioisivat kaikki kirjaamistavat. Valtakunnallisesti olisi tarve yhtenäistää päivämäärien kirjaamistapaa, jotta aineistojen automaattinen käsittely olisi tehokasta.

6. PDF-linkityskonversio

6.1 Kaava-asiakirjatiedostojen uudelleennimeäminen

Kaava-asiakirjatiedostojen uudelleennimeämisen periaatteena on yhdenmukaisten nimien muodostus. ETL-työkalu muodostaa vakioidut nimet liitostauluun uuteen “new_filename” -ominaisuustietokenttään seuraavin periaattein:

1. Asiakirjanimi huomioi virallisen kuntakoodin
2. Asiakirjanimi huomioi Rakennetun ympäristön tietojärjestelmän (Ryhti) koodistojen mukaisesti kaavalajin
3. Asiakirjanimi huomioi Ryhtin koodistojen mukaisesti asiakirjan lajin
4. Asiakirjanimi huomioi kaavatunnuksen
5. Kaksi tai useampi asiakirja ei saa olla samanniminen

Esimerkki ETL-työkalun (tarkemmin ETL-työkalusta liitteessä 1) tuottamasta kaava-asiakirjan tiedostonimestä "402-31-05-95-2.pdf", jossa:

- 402 on Leppävirran virallinen kuntakoodi
- 31 on RYTJ:n kaavalaji-koodiston mukaisesti asemakaava
- 05 viittaa RYTJ:n asiakirjan laji -koodiston mukaisesti siihen, että liite kattaa sekä kaavakartan että määräykset
- 95 on kunnan indeksitunnus
- 2 on liukuva etuliite (402-31-05-95-1.pdf on jo olemassa, eli voimassa olevalle kaavalle on löytynyt kaksi samannimistä liitettä)

ETL-työkalu mahdollistaa asiakirjatiedoston uudelleennimeämisen joko KTJ-tunnukseen tai kunnan omaan indeksitunnukseen perustuen.

HUOM! Kaava-asiakirjatiedostojen uudelleennimeämiskäytäntö on ainoastaan ehdotus liitteiden nimeämiselle Ryhti-järjestelmässä! Vakioitu nimeämiskäytäntö voi muuttua Ryhti-toteutusprojektin aikana!

6.2 Asiakirjatiedostojen linkitys paikkatietoaineistoon

ETL-työkalu linkittää uudelleennimetyt tiedostonimet linkitystaulusta paikkatietoaineistoon kaavalajin, kuntakoodin sekä kaavan indeksitunnuksen perusteella (joko KTJ tai kunnan tunnus, riippuen kumman mukaan liitteet on uudelleennimetty). Kaavalaji ja kuntakoodi tarvitaan, koska:

1. Kunnan sisällä esim. ranta-asemakaavoilla voi olla sama kunnan indeksitunnus kuin asemakaavoilla
2. Eri kunnilla voi olla tismalleen sama kaavatunnus.

Kukin asiakirja löytyy ajon jälkeen paikkatietoaineistosta omasta kentästään (kaavakartta).

6.3 PDF/A-konversio

ETL-työkalu konvertoi kunnilta saadut asiakirja-PDF-tiedostot automatisoidusti Ghostscript-muuntimella PDF/A-arkistoformaattiin. Muunnin perustuu soveltuvin osin GitHubista avoimesti saatavilla olevaan Python-Ghostscript-konvertteriin. VOOKA-projektin tarpeisiin muokattu muunnin löytyy omana versionaan projektin GitHub-sivuilta.

Jos saatu PDF-formaatti on toteutettu versiolla, jota ei saada PDF/A-muotoon, se tallennetaan sellaisenaan. Mikäli jotkin kunnan PDF-tiedostot eivät käänny kunnan aineistosta johtuvista teknisistä syistä PDF/A-muotoon, tallennetaan nämä normaalissa PDF-muodossa.

7. Kaavatietomallin implementointi

ETL-työkalussa kaavatietomallin implementointi tapahtuu muuntamalla GeoPandas GeoDataFrame -muotoinen kaava-aineisto GeoJSON-formaattiin. ETL-muuntimelle ilmoitetaan aineistolähteeksi joko

“kunta” tai “KTJ” sekä molempien aineistolähteiden kaavatunnukset. Osana aineistojen vertailua kohdassa 4.2 on luotu tietokenttä “refe_kaavatunnus”, joka kertoo esim. KTJ-pohjaisessa aineistossa kaavan vastaavan kuntatunnuksen. Mikäli vastaavaa tunnusta ei löydy (esim. kaavakohde on vain KTJ:n aineistossa), ETL-muunnin jättää kunnan indeksitunnus-tiedon GeoJSON:ssa tyhjäksi. Kaavatietomallin soveltamisessa on hyvä huomioida kaavatietomallin jatkuva muuttuminen, joten syötettyjen tietojen määrä tai nimeämistapa sekä tallennettavan tiedon hierarkinen rakenne kannattaa tarkastaa kaavatietomallia ylläpitävältä taholta ennen tietojen syöttämistä. Kuntien aineistot saattavat olla joko polygon tai multipolygon muotoisia. Lopulliseen JSON-formaattiin kaikki geometriat kuitenkin tallennettiin multipolygoineina.

ETL-työkalun avulla on lisäksi mahdollista normalisoida kaavatietomallin mukainen JSON-aineisto takaisin geopackage-formaattiin, jotta aineisto on kunnille helpommin luettavissa sekä tarkasteltavissa paikkatieto-ohjelmistoissa.

Kaavatietomalliin on määritelty pakolliseksi tiedoksi kirjata päätöksentekijän laji. Projektissa ei kerätty tietoa, onko kaavan hyväksynyt esimerkiksi kaupungin lautakunta vai yksittäinen viranhaltija. Tieto vaihtelee kaavakohtaisesti, mutta todennäköisesti yksittäisen viranhaltijan tekemiä päätöksiä on vain muutamia, jos ollenkaan. Siksi Pohjois-Savon toteutuksessa päätöksentekijän lajiksi määriteltiin kaikkiin kaavoihin 02 - Monijäseninen päätöksentekoelin.

8. Jatkokehitystarpeet

8.1 OGC API Features maantieteellinen rajaus

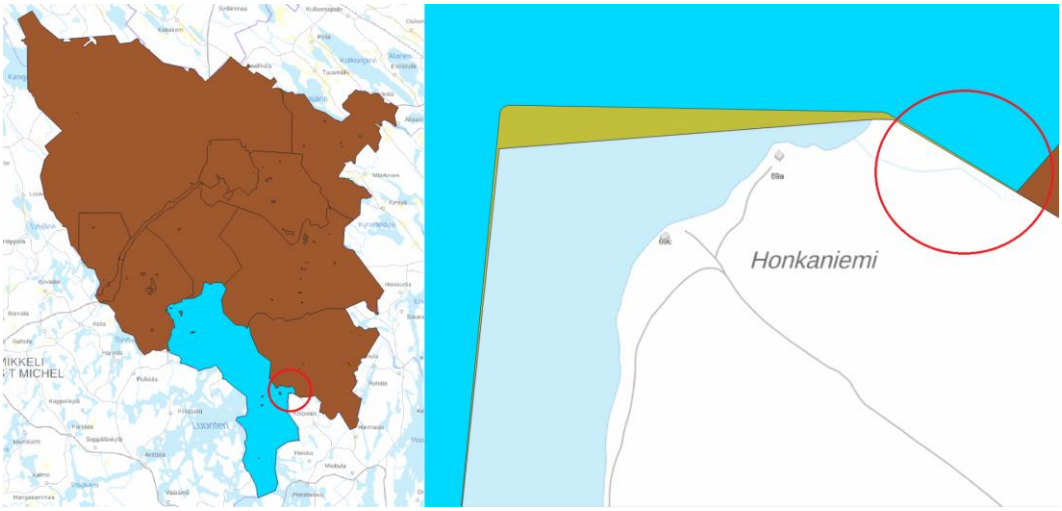
Sekä Etelä-Savon pilotissa että Pohjois-Savon VOOKA-projektissa tarkasteltavan maakunnan maantieteelliset rajat on kovakoodattu bounding boxina OGC API Features getter-skriptiin. Jatkossa muilla maantieteellisillä alueilla bounding box tulee korvata halutuilla koordinaateilla.

8.2 Geopandasin versiopäivitys

Pohjois-Savon projektissa ETL-prosessin vaiheessa 1.1 havaittiin, ettei ETL-työkalu toimi kaikkiin WFS-rajapinnoista tehtäviin aineiston irrotuksiin. Virhe johtuu luultavasti ETL-työkalussa käytetystä Geopandasin versiosta, joka ei pysty käsittelemään rajapinnassa olevia geometrioita oikein. ETL-työkalussa käytetty Geopandasin ja sen liitännäisten versiot voisi päivittää jatkokehityksessä uudempiin, mikä vaatii muutoksia ja testausta kaikkiin ETL-työkalun vaiheisiin.

8.3 Yleiskaavojen kuntarajatörmäys

Yleiskaavojen kuntarajatörmäyksen korjauslogiikka toimii hyvin, muttei aukottomasti. VOOKA-pilottiprojektin aikana tunnistettiin yksi selkeä tapaus, johon logiikan avulla ei ole vielä annettu vastausta: kun kaavaindeksi jää toisen kaavaindeksin rajalla “auki”, jolloin laskennassa muodostuva differencen osageometria ei filteröidy pinta-alavertailulla pois (kuva 7).



Kuva 7. Esimerkki tilanteesta, jossa yleiskaavaindeksin kuntarajatörmäyksen korjaus ei toimi aukottomasti. Kuntarajamaskin ja kaavaindeksin differencea laskettaessa indeksi jää toisen kaavaindeksin rajalla “auki” suhteessa maskiin, jolloin eroavaisuus ei filteröidy pinta-alavertailulla pois. Sininen alue (kaavaindeksi) ei yhdisty vihreään (kiinteistökuntaraja), vaikka pitäisi.

Pinta-alan suuruuteen pohjautuva difference-geometrioiden filtteröinti perustuu kokeiluun, sillä kunta-aineistojen spatiaalisissa tarkkuuksissa on paljon eroja. VOOKA-projektin osalta jokaisen kunnan toimiva raja-arvo on testattu erikseen ja listattu referenssiksi jatkokehitystä varten. Vaikka kuntien toleranssiarvoissa onkin yhdenmukaisuutta, niihin ei voi luottaa sokeasti eri maantieteellisillä alueilla, vaan jokaiselle kunnalle tulee löytää toimiva raja-arvo testauksen kautta erikseen. Jatkokehitysideana tähän tarpeeseen voisi myös kouluttaa AI/ML-toimintoja arvioimaan sopivinta toleranssiarvoa kullekin kunnalle.

8.4 Kohderekisteritunnukset palstatiedoista

Kiinteistötunnusten sijaintitiedot ovat pistetason lisäksi tarjolla MML:n OGC API Features -rajapinnasta aluetasona osana palstojen laajoja sijaintitietoja. Aluemaisen referenssitason hyödyntäminen kohderekisteritunnusten linkittämisessä kaavaindekseihin olisi tarkempi tapa kuin pistetason hyödyntäminen. VOOKA-projektin toteutuksessa osa kiinteistötunnuksista voi jäädä linkittämättä kaavaindekseihin, jotka todellisuudessa sivuavat kiinteistötunnusta, sillä kiinteistötunnuksen sijaintitieto on vaikutusalueen sijasta sidottu vain yhteen pisteeseen.

Lisäksi rakennetun ympäristön tietojärjestelmän kaavatietomallissa halutaan tietää kohderekisteritunnusten osalta, että kuuluvatko ne kokonaan kaavan sisälle sekä sisältyvän alueen pinta-ala. Tätä tietoa ei ole mahdollista saada tietoon ainoastaan pistetasoa hyödyntämällä.

Huom. Pohjois-Savon toteutuksen aikana ilmeni, ettei pilotin jälkeen päivittyneeseen kaavatietomalliin tarvita enää näitä ominaisuustietoja, joten tämän kehittäminen jää vapaaehtoiseksi vaiheeksi.

8.5 Kiinteistöpalstojen ja kaavaindeksien vertailu

VOOKA-projektissa kehitetty ETL-työkalu ei tunnista sitä, onko kiinteistörajat muodostettu ennen kaavan voimaantuloa vai sen jälkeen. Näin ollen se ei pysty indikoimaan, onko kaavaindeksin rajan tarkoitus olla yhtenevä kiinteistörajan kanssa vai ei.

8.6 Ominaisuustietojen vertailu

VOOKA-pilotissa ominaisuustietojen vertailu on mahdollista vain kolmelle tietokentälle (kaavalaji, hyväksymispäivämäärä ja voimaantulopäivämäärä) KTJ-aineistopohjaisesti. Muilla maantieteellisillä alueilla ominaisuustiedot voivat olla kattavampia, jolloin vertailuun voi tuoda mukaan myös muita tietokenttiä. Lisäksi KTJ-pohjainen vertailuperiaate on mahdollista laajentaa myös kuntapohjaiseksi.

8.7 Aineistojen kansiolajittelun seuranta

Pohjois-Savon projektin aikana huomattiin, että kunnat päivittävät usein aineistoja Sharepoint-palveluun osissa, mikä tekee aineistosaannon seurannan haastavaksi. Aineistosaannon seuraamiseksi olisi tarpeen päivittää kansiolajitteluun työkalu, joka tunnistaa uudesta ladatusta aineistosta duplikaatit. Tämän avulla voisi seurata saadun aineiston määrää.

8.8 PDF-linkityskonversiotaulun luonti ja päivitys

Tällä hetkellä linkitystaulun luonti tapahtuu niin, että prosessissa luodaan kansiosta jokaisessa ajossa erillinen csv-tiedosto, josta tiedot siirretään pilvessä olevaan taulukkoon. Projektin aikana keskusteltiin, voisiko taulukkoa päivittää suoraan pilveen, mutta suurin este tälle oli tietoturvakysymykset. Ohjelmalle ei voida antaa oikeuksia muokata yrityksen pilvessä olevia tiedostoja.

Kuitenkin olisi mahdollista luoda toteutus, jossa tiedot päivittyvät paikallisesti koneella olevaan csv-taulukkoon. Tätä varten olisi hyvä kehittää ohjelmakoodia niin, että sen avulla voidaan päivittää tiedostoa ilman, että jo luodut arvot poistuvat. Lisäksi tätä varten pitäisi kehittää kattavampaa virheiden käsittelyä.

Tämä vaihe luotiin Pohjois-Savon projektissa, joten todennäköisesti vaiheen ohjelmakoodia ja sen toimintalogiikkaa olisi tarve jatkokehittää, jotta se toimisi muidenkin maakuntien osalta.

8.9 PDF-linkityskonversiotaulun luonti - dokumentin tyyppi

Linkitystaulun luomiseen kehitettiin työkalu, joka päättelee tiedoston nimestä sen tyypin. Tällä hetkellä työkalu palauttaa dokumentin tyypiksi määräyksen, jos tiedoston nimi sisältää määräykseen viittaavaa sanastoa. Jos kyseistä sanastoa ei löydy, niin työkalu antaa dokumentin tyypin oletusarvoksi kaavakartan sekä määräykset. Työkalu ei tällä hetkellä ole täydellinen, joten dokumentin tyyppi täytyy varmistaa oikeaksi manuaalisesti tarkistamalla. Työkalu antaa tällä hetkellä osittain virheellisiä tuloksia, joten jatkokehityksenä olisi tarpeen luoda varmempi tapa tunnistaa dokumentin tyyppi. Käytännössä tämä tarkoittaa sitä, että olisi tarpeen käyttää esimerkiksi Python Difflib-moduulia tai muuta vastaavaa menetelmää.

8.10 Sisäkkäisten kaavarajojen tunnistaminen koneoppimismenetelmällä

ETL-prosessin vaiheessa 3.2.1 kuntien kaavarajat rajattiin todellisiin kuntarajoihin. Prosessin automatisoinnissa käytettävien toleranssiarvojen testaus toteutettiin kokeillen, mutta koska Pohjois-Savon aineistossa esiintyvien kunnan ulkorajojen ulkopuolisten kaavojen tai kaavan osien määrä oli varsin pieni, noudattelivat kaavojen ulkorajat varsin hyvin voimassa olevia kunnan ulkorajoja. Samaa logiikkaa voisi

käyttää esimerkiksi asemakaavojen rajojen yhdenmukaistamiseen viereisten palstojen rajojen kanssa. Tämän prosessin automatisoinnissa pitäisi kuitenkin hyödyntää koneoppimismenetelmiä, joka osaisi lukea sisäkkäisten kaavojen rajoja ja etsiä lähimpiä tai sopivimpia pisteitä tai viivoja, joihin kunnan kaavarajoja rinnastaa. Koneoppimismenetelmässä voisi kokeilla myös käyttää vertailuaineistona esimerkiksi Maanmittauslaitoksen kiinteistörajoja, mikäli ne ovat yhdenmukaisia vaikkapa asemakaavojen ulkorajojen kanssa.