Jere Virolainen

Vantaan 3D-kaupunkimallin ylläpito tietokannassa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Maanmittaustekniikka

Insinöörityö

19.4.2018

Metropolia

Tekijä Otsikko Sivumäärä Aika	Jere Virolainen Vantaan 3D-kaupunkimallin ylläpito tietokannassa 41 sivua 19.4.2018
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	maanmittaustekniikka
Ohjaajat	lehtori Jussi Laari kaupunkimittausinsinööri Arsi Juote

Insinöörityön tarkoituksena oli kuvailla uusi ylläpitoprosessi Vantaan kaupungin kolmiulotteiselle kaupunkimallille sen siirtyessä tietokantaan. Tavoitteena työssä oli selvittää, miten malli siirretään tietokantaan sekä kuvailla uusi jatkuva ylläpitoprosessi aina kaupungin rakennusmallien luomisesta, sen julkaisuun asti.

Työ toteutettiin perehtymällä verkossa saatavilla oleviin aineistoihin, joita aiheesta löytyi mittavia määriä. Perehtymisen lisäksi suoritettiin empiirisiä kokeiluja ja niistä tehtyjen havaintojen pohjalta kartoitettiin ongelmakohtia, ratkaisut niille ja lopullinen prosessi kaupunkimallin ylläpidolle tietokannan kautta. Käytännön kokeilut liittyivät muun muassa kaupunkimallin tiedostoformaatin konvertointiin, tietokantaan siirtoon sekä kaupunkimallin julkaisuun esimerkiksi internet selaimeen pohjautuvalla ratkaisulla. Kokeilut ja ongelmien ratkaisut tehtiin eri paikkatietoalalla käytössä olevilla ohjelmistoilla.

Uutta ylläpitoprosessia ei vielä tutkielman lopussa otettu käyttöön. Tuloksina selvisi kuitenkin, että CityGML, 3D City Database ja Cesium ovat lupaavia ja hyviä ratkaisuja kolmiulotteisen kaupunkimallin tehokkaan jatkuvan ylläpidon sekä julkaisun mahdollistamiseksi. Tutkielmassa tehtyjen empiiristen kokeilujen tulosten pohjalta jatketaan uuden kolmiulotteisen kaupunkimallin tietokantapohjaisen ylläpidon kehitystä ja käyttöönottoa Vantaan kaupungissa yhteistyökumppanien kanssa.

Avainsanat	CityGML, 3D-kaupunkimalli, tietokanta, ylläpito

Author	Jere Virolainen
Title	Maintaining 3D City Model of Vantaa in Database
Number of Pages	41 pages
Date	19 April 2018
Degree	Bachelor of Engineering
Degree Programme	Land Surveying
Instructors	Jussi Laari, Senior Lecturer Arsi Juote, City Survey Engineer

The purpose of this Bachelor's thesis was to define a new method for maintaining the digital 3D city model of Vantaa as it was to be transferred to a database. The thesis focused on establishing a continuous maintenance process from the creation of digital building models to the publication of the model.

To achieve the goals, the possible problems of the transferring project were mapped, and solutions for them found with the help of the vast material online. Once the problems were mapped, practical experiments were done to find out how the city model could be published on a browser based solution. A number of geoinformatics software solutions were used.

The new maintenance method was not launched at the end of this study. The results of the study, however, indicated that the designated solutions and formats are suitable for the effective continuous maintenance of the Vantaa 3D city model. The study is an excellent basis for the future development of the maintenance process.

Keywords	CityGML, 3D City Model, database, maintenance

Sisällys

Lyhenteet

BIM	Building Information Model. Rakennuksen tietomalli eli rakennuksen koko elinkaaren ajan käytettävä tarkka digitaalinen kolmiulotteinen malli, joka sisältää yksityiskohtaista tietoa rakennuksen ominaisuuksista.
CAD	Computer Assisted Design. Tietokoneavusteinen suunnittelu.
CityGML	Kaupunkimaisia kohteita varten kehitelty GML:n sovellus.
DGN	Design file. Microstation-ohjelmiston oma tiedostomuoto.
FME	Feature Manipulation Engine. Safe Softwaresin kehittämä ohjelmisto datan yhdistämistä ja muokkausta varten.
GML	Geography Markup Language. Spatiaalisen datan esittämistä varten kehitetty merkintäkielioppi.
LOD	Level Of Detail. Kapunkimalleissa käytettävä mallin tarkkuustaso. Ilmaistaan yleensä viitenä tasona LOD0–LOD4.
OASIS	Organization for the Advancement of Structured Information Standards. Avoimien standardien kehittämiseen ja hyväksymiseen perustettu kansainvälinen liitto.
OGC	Open Geospatial Consortium. Kansainvälinen avoimia standardeja kehittävä liitto paikkatietoaineistoille.
SIG 3D	Special Interest Group 3D. Avoin kansainvälinen kolmiulotteista kaupunki– ja maastomallinnusta kehittävä ja tutkiva työryhmä.
W3C	World Wide Web Consortium. Kansainvälinen verkkostandardeja kehittävä liitto.
XML	Extensible Markup Language. Tietynlaisten merkintäkielien yläkäsite ja metakieli.

1 Johdanto

Tämä insinöörityö käsittelee Vantaan kaupungin kolmiulotteisen kaupunkimallin siirtämistä tietokantaan sekä sen myötä uutta hallintamenetelmää ja ylläpitoprosessia kolmiulotteiselle kaupunkimallille tietokantaa hyödyntäen. Vantaan kaupungin kaupunkimittauksen osastolla pohdittiin kaupunkimallin tiedonhallinnan ja ylläpidon parannuksia sekä siirtyä käyttämään rikasta semantiikkaa tukevaa kaupunkitietomallia. Vaihtoehdoiksi näille nousi esiin muun muassa CityGML-formaatti sekä 3d City Database -tietokanta. CityGML vaikutti hyvältä ”de facto” -tyyliseltä standardilta, ja 3d City Database sopivalta teknologialta muun muassa siksi, että se toimii jo Vantaalla käytössä olevan PostgreSQL-tietokanta-alustan kanssa. Insinöörityön tarkoituksena on selvittää ja kuvailla kokonaisvaltainen prosessi ajantasaisen kaupunkimallin tuottamiseen, ylläpitoon sekä julkaisuun kyseisten tietokannan ja tiedostoformaatin avulla. Kysymyksinä prosessiin liittyen olivat etenkin, miten jo nyt tuotettu kaupunkimalli muutetaan CityGML-muotoon, kuinka malli siirtyy ja tulostuu tietokannasta hallittavaksi ja miten malli julkaistaan?

Raportissa käsitellään sekä selvennetään ensiksi yleisesti kaupunkimalleja sekä niiden tuottamista. Lisäksi käsitellään CityGML sekä sen tuomat hyödyt ja miksi se on juuri oivallinen tiedostomuoto kaupunkimallille. Yhtenä käsittelyn aiheena on kaupunkimallin ylläpito sekä sen ongelmakohdat yleisesti. Raportissa kuvaillaan myös nykyinen prosessi ennen tietokantapohjaista kaupunkimallia sekä tuleva uusi, tietokantapohjainen prosessi ylläpidolle. Lisäksi esitellään kaupunkimallin julkaisuun liittyviä asioita sekä julkaisu osana ylläpidon prosessia. Lopuksi käydään läpi käytännössä törmättyihin ongelmiin ja ratkaisuihin sekä selostetaan tutkielman aikana tehtyjä käytännön kokeiluja yleisesti.

Työ koostui laajalti aineiston tutkimisella, sekä empiirisillä kokeiluilla, joiden kautta pyrin löytämään ratkaisuja työhön liittyviin haasteisiin. Aloitin työn perehtymällä CityGML-formaattiin, sekä kokeilemalla 3D City Database -tietokanta-alustaa. Kokeilut ja tutkimukset siirtyivät seuraavaksi internetjulkaisuun liittyviin asioihin. Esimerkiksi siihen, kuinka tietyt tiedostoformaatit voidaan luoda mallista ja miten ne määrittyvät sovellusympäristöissä oikein, jotta ne olisivat julkaisukelpoisia selaimessa.

2 Kolmiulotteinen kaupunkimalli

Kolmiulotteinen kaupunkimalli, eli 3D-kaupunkimalli, on visualisoitu esitys rakennetun ympäristön reaalimaailman kohteista kolmiulotteisessa ympäristössä (Billen ym. 2014: 10). 3D-kaupunkimalli esitetään nykyään lähes aina digitaalisena tietokoneella, tai muulla päätelaitteella, ja tässä työssä käsittelen digitaalista kaupunkimallia erotuksena niin sanottuun korkokuvasta tehtyyn malliin. Kaupunkimalli koostuu rakennuksista, kasvillisuudesta, infrastruktuurista ja muista kaupunkikohteista. (Lammi 2015: 2.) Yksinkertaisemmillaan malli koostuu vektoroiduista, kolmiulotteisista kaupungin rakennuksista ja maanpinnan mallista. Tarkempiin malleihin ollaan lisätty rakennusten lisäksi pienempiä yksityiskohtia kuten kasvillisuus, infrastruktuuri ja muita kohteita, kuten aidat ja kaiteet. (Ahokas 2014: 8.)

Kolmiulotteiseen kaupunkimalliin voidaan graafisen ulkoasun ja geometrian lisäksi lisätä semantiikkaa, jolloin voidaan puhua jo niin sanotusta kaupunkitietomallista erotuksena pelkkään vektorimuotoiseen geometriamalliin. Semantiikka on mallin kohteiden ominaisuustietojen sekä luokkien, suhteiden ja rakenteiden määrittelyä. Geometrian lisäksi mallin kohteiden ominaisuudet ja niiden väliset suhteet on myös kuvattu semanttisessa kaupunkimallissa. (Liukkonen 2015: 21, 34.) Toisin sanoen, esimerkiksi mallin rakennuksiin voidaan lisätä tietoja mm. rakennuksen osoitteesta, korkeudesta, rekisterinumeroista ynnä muusta halutuista tiedoista. Tällöin mallin avulla voidaan luoda erilaisia analyysejä ja simulaatioita ja mallia voidaan käyttää myös monissa sovelluksissa ja suunnittelutehtävissä sekä kohteiden tietojen ylläpidossa ja seurannassa. (Uuden sukupolven kaupunkimallit Helsinkiin 2016: 7; Kärkkäinen 2016: 4.) Kaupunkitietomalleille yksi avoin, laajasti käytetty ja dokumentoitu standardi on muun muassa CityGML. CityGML ratkaisee juurikin semantiikan mallintamiseen sekä erilaisia tiedon yhteensovittamiseen ja niiden toimivuuteen keskenään liittyviä haasteita. Luku 3 käsittelee laajemmin CityGML-standardia.

2.1 Kaupunkimallien käyttö

Kolmiulotteinen malli tuo esille reaalimaailman kohteita ja asioita kaksiulotteista karttaa tehokkaammin. Siksi malleja käytetään etenkin visualisoimiseen. Mallin avulla on helppo visualisoida kohteita ja niiden vaikutuksia olemassa olevaan ympäristöön, esimerkiksi näkymien ja maiseman kannalta, ennen kuin kohteet ovat valmiita reaalimaailmassa.

(Erving 2008: 4.) Samalla tehokas visualisointi heijastuu tehokkaaseen kaupunkisuunnitteluun. Kokonaisia uusia asuinalueita voidaan 3D-kaupunkimallien avulla suunnitella erittäin tarkasti. Näin säästytään yllätyksiltä toteutusvaiheessa, mikä edelleen madaltaa esimerkiksi kustannuksia ja toteutusaikaa.

Tietorakenteen omaavilla kaupunkitietomalleilla voidaan myös suorittaa erilaisia analyysejä ja simulaatioita, jotka ovat muuten käytännössä mahdottomia tehdä. Esimerkiksi auringon säteily ja rakennusten luomat varjot voidaan esittää mallissa tarkasti. Auringon säteilyn kannalta mallia voidaan lisäksi hyödyntää aurinkoenergian potentiaalisten keräyskohteiden kartoittamiseen ja analysointiin. Lammi mainitsee esitelmässään (2015: 9) kaupunkimallien käytön Keski-Euroopassa, jossa niitä käytetään kaupunkisuunnittelun lisäksi mm. pelastustoimien simulointiin, radioverkon kuuluvuuden suunnittelussa, melusuunnittelussa, poliisien simulaattoreissa, navigaatiossa ja kiinteistöjen hallinnassa ja ylläpidossa.

2.2 Tuotanto

Kolmiulotteisen kaupunkimallin tuottaminen on monivaiheinen prosessi, joka koostuu eri tuotantovaiheista kohteiden geometriatiedon hankinnasta aina datan käyttöön eri sovelluksineen. Yksinkertaisimmillaan mallin tuotanto koostuu geometriadatan hankinnasta, luokittelusta ja 3D-kohteiden luomisesta sekä käytöstä yksinkertaisessa sovelluksessa, esimerkiksi CAD -ohjelmistossa sellaisenaan. Kolbe selostaa esitelmässään (2009a: 11) kaupunkimallinnuksen tuotannon viisi vaihetta:

- datan hankinta
- datan luokittelu ja kelpuuttaminen
- datan jalostus
- muokkaus sovellusta varten
- valmis sovellus kaupunkimallista.

Kolmiulotteisen kaupunkimallin tuottamisen pitäisi Döllnerin ym. (2006: 3) mukaan perustua aina automaattisiin tai puoliautomaattisiin menetelmiin kun se on mahdollista. Manuaalinen mallinnus on hyväksyttävää pienen mittakaavan kaupunkimalleissa, mutta mallinnettaessa suuria alueita manuaalinen mallinnus ei ole kustannustehokasta ja vie paljon aikaa. Ainoat taloudellisesti kannattavat menetelmät 3D-kaupunkimallinnuksen

geometriadatan hankintaan perustuvat fotogrammetrisiin menetelmiin. Tällaisia ovat erilaiset kuvaus- ja laserkeilausmenetelmät, joilla pystytään tuottamaan kattavaa tietoa kohteiden sijainnista, korkeudesta sekä ulkoasusta. Muina tietolähteinä kaupunkimallinnuksessa voidaan käyttää mm. kantakarttaa, paikka- ja rekisteritietoja, eri tietokantoja, maastomalliaineistoja tai mitä tahansa muita käytettävissä olevia aineistoja, kuten jo valmiita kolmiulotteisia CAD- ja BIM-malleja. (Ahokas 2014: 14; Liukkonen 2015: 47–48.) Datan luokittelu- ja kelpuutusvaiheessa hankittua geometriadataa luokitellaan eri kohteiksi ja siitä luodaan 3D-malleja. Jalostusvaiheessa voidaan luotua mallia niin sanotusti rikastaa esimerkiksi lisäämällä siihen semanttisia tietoja ja yhdistää eri tietolähteiden aineistoja. Eri sovellukset kaupunkimallista vaativat usein myös omanlaista sovelluskohtaisia tietoja, joita voidaan muokkausvaiheessa lisätä sovelluksen vaatimuksien mukaan. Lopulta malli on valmis käytettäväksi halutussa sovelluksessa. Erving kuvailee selvitystyössään (2008) CityGML-muotoisen kolmiulotteisen kaupunkimallin tuotantoprosessin seuraavasti: aineiston keruu, 3D-mallinnus, georeferointi, semantiikan lisäys ja mallin tallentaminen lopulliseen CityGML-muotoon.

Kaupunkimalleja usein myös teksturoidaan tuotantovaiheessa. Teksturointi tarkoittaa kuvan tai kuvioinnin lisäämistä kaupunkimallin kohteiden pinnoille. Kun tekstuurit lisätään oikeista valokuvista kaupunkimallin kohteille, puhutaan fotorealistisesta kaupunkimallista. Valokuvien lisäksi tekstuureina voivat toimia myös yksinkertaiset väriarvot sekä keinotekoiset kuvioinnit. Tekstuurit lisäävät mallin realistisuutta mikä tuo enemmän tietoarvoa mallille. (Erving 2007: iii.)

2.3 Ohjelmistot

Esittelen tässä alaluvussa lyhyesti niitä kaupunkimallin tuotantoon käytettyjä ohjelmistoja, jotka ovat olennaisessa osassa Vantaan kaupunkimallin tuotantoa ja joita käytettiin työn aikana.

Microstation

Microstaton on Bentley Systemsin kehittämä tietokoneavusteinen suunnitteluohjelma, jolla voidaan luoda, dokumentoida sekä tallettaa ja tarkastella kaksi -ja kolmiulotteista tietoa. Ohjelmistoa käytetään muun muassa maankäytön, arkkitehtuurin ja infrastruktuurin suunnittelun työkaluna. (Microstation 2017.)

FME

FME (sanoista Feature Manipulation Engine) on Safe Softwaresin kehittämä ohjelmisto, jonka avulla voidaan yhdistää ja muokata eri tietoja useista eri tiedostomuodoista. Sen avulla voi muun muassa muuntaa datan tiedostomuotoa formaatista toiseen hävittämättä tietoa. FME toimii niin sanotuilla työtiloilla, jossa tiedosto luetaan sisään, muokataan halutuksi FME:n transformer-muuntajatyökaluilla sekä kirjoitetaan ulos halutussa formaatissa. Työtilat voidaan automatisoida ja asettaa toimimaan esimerkiksi jatkuvana tai tietyn aikavälein ajettaviksi. (How FME works? 2018.)

Terrasolid -tuotteet

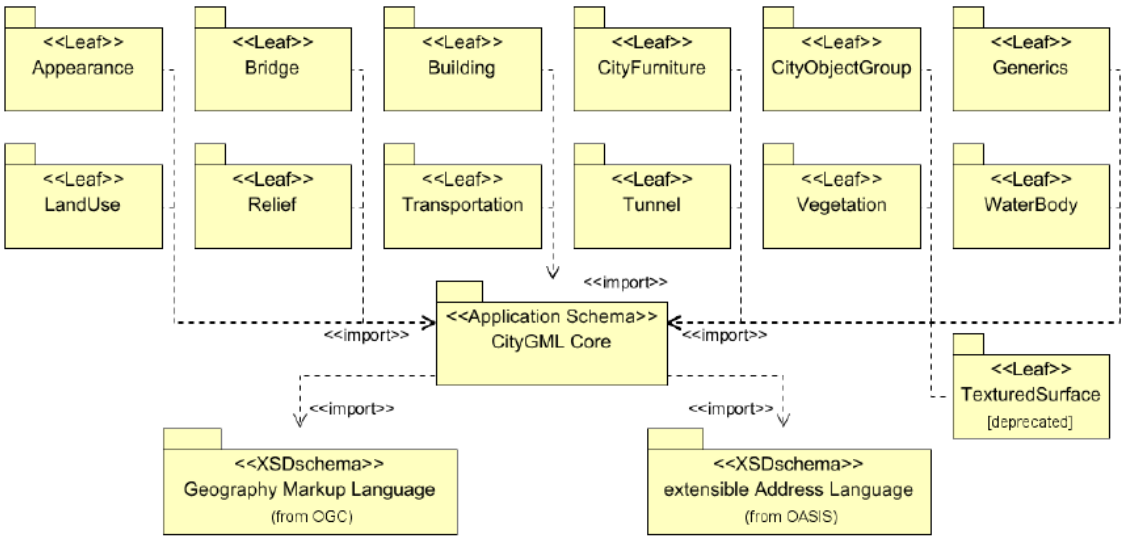
Terrasolid on suomalainen yritys, joka tuottaa ohjelmistoratkaisuja pistepilvi- ja ilmakuva-aineistojen prosessoinnille. Useimmat Terrasolid- tuotteet toimivat Bentley Systemsin ohjelmistojen ympäristössä, kuten Microstationissa. (Terrasolid products 2018.) Terrasolidin tuotteilla muun muassa tuotetaan kolmiulotteista kaupunkimallia ilmasta kerätyn pistepilviaineiston ja ilmakuvien pohjalta Vantaan kaupungissa.

3 CityGML

CityGML on Special Interest Group 3D:n (SIG 3D) vuodesta 2002 kehittelemä, Open Geospatial Consortiumin (OGC) hyväksymä Extensible Markup Languageen (XML) pohjautuva kolmiulotteisten virtuaalisten kaupunkimallien esitystä, varastointia ja jakamista varten kehitelty tiedostoformaatti. Se on yleinen semanttinen tietomalli, jonka avulla voidaan esittää ja jakaa kolmiulotteisia kaupunkimaisia kohteita eri sovellusalustojen välillä. CityGML on sovellusskeema Geography Markup Languagesta (GML), ja se perustuu useisiin ISO 19100 perheen, eli OGC:n, World Wide Web Consortiumin, (W3C) Web 3D Consortiumin ja Organization for the Advancement of Structured Information Standards (OASIS) määrittelemiin standardeihin. CityGML:n kehityksen tavoitteena on luoda yleiset määrittelyt 3D-kaupunkimallin perusrakenteille, ominaisuuksille ja suhteille. (OGC CityGML Encoding Standard 2012: xiv, 9; Liukkonen 2015: 28.)

3.1 Modularisointi ja geometria

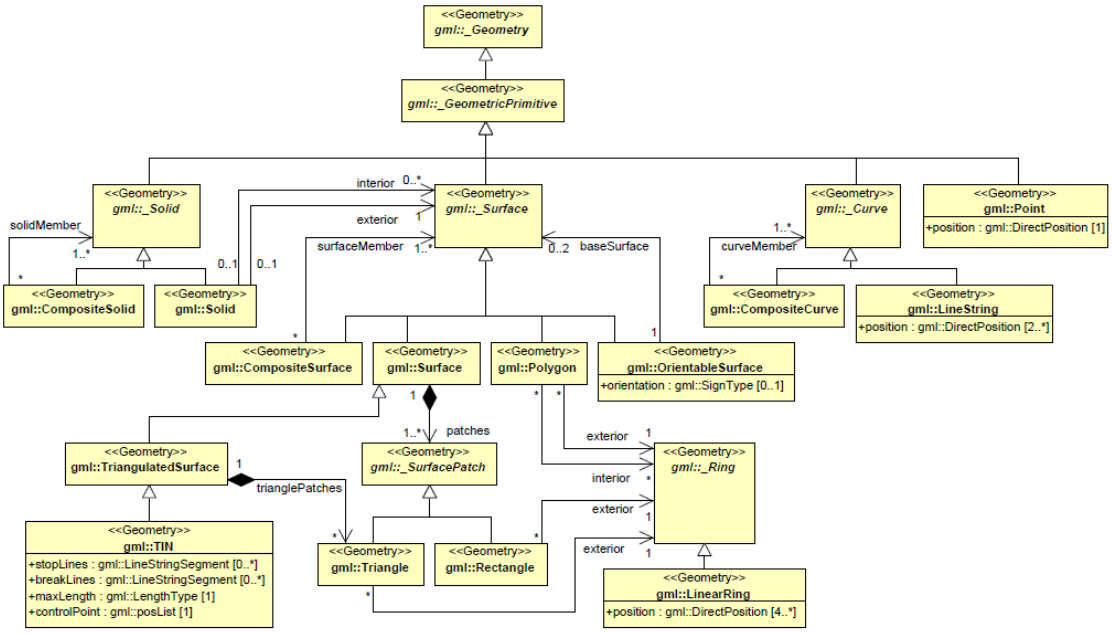
CityGML ei ole pelkästään kaupunkimallin graafisen ulkonäön esittämistä varten, vaan se osoittaa myös mallin semanttisia ja temaattisia ominaisuuksia, sekä niiden luokitusta ja aggregointia. Se määrittelee erilaisia luokkia ja suhteita kaupunkikohteisiin niiden geometristen, topologisten, semanttisten ja ulkonäöllisten ominaisuuksien mukaan. CityGML on jaettu kahteen osaan temaattisesti: ydinmoduuliin (core module) ja laajennusmoduuleihin (extension modules). Kaupunkikohteet ovat organisoituneena näihin laajennusmoduleeihin oman temaattisen aihealueensa mukaan, ja jokainen laajennusmoduuli sisältää yhden temaattisen aihealueen osat. Temaattisia aihealueita ovat mm. rakenteet ja rakennukset, ulkonäkö, maanpinnan malli, kasvillisuus, maankäyttö, vesistöt, liikenteelliset kohteet sekä niin sanotut kaupunkihuonekalut. Kaikki laajennusmoduulien temaattiset aihealueet ovat nähtävissä kuvassa 1. Laajennusmoduulit ovat riippuvaisia ydinmoduulista, joka sisältää CityGML:n peruskäsitteistön ja osat. (OGC CityGML Encoding Standard 2012: 9, 11, 18; Liukkonen 2015: 30.)



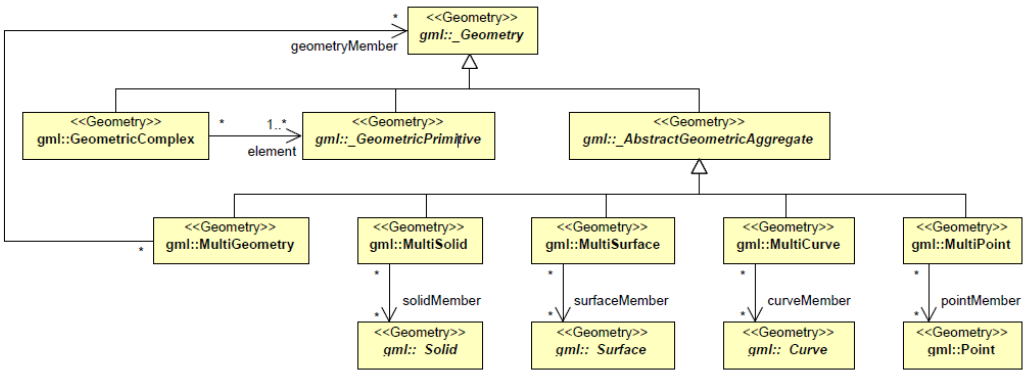
Kuva 1. UML-diagrammi CityGML:n modularisoinnista (OGC CityGML Encoding Standard 2012: 19).

CityGML:n kohteiden pinnat ja topologiat esitetään GML3-geometriamallin mukaisesti, josta CityGML käyttää vain osaa. GML3-geometriamalli koostuu perusalkioista, joita yhdistämällä voidaan luoda komplekseja (complex), yhdistelmägeometrioita (composite geometry) ja yhdistelmiä (aggregate). (Kuvat 2 ja 3.) Jokainen ulottuvuus käsittää oman

geometrisen perusalkionsa: piste (point) 0-ulottuvuudessa, kaari (curve) 1-ulottuvuudessa, pinta (surface) 2-ulotteisessa ja kappale (solid) 3-ulotteisessa. Kappale, esimerkiksi rakennus, tulee olla ympäröity pinnoilla ja pinnat kaarilla. Pinnat esitetään monikulmiona (polygon), jonka kaikki rajaavat kaaret ja sisältävät pisteet tulee olla samalla yhdellä tasolla. (OGC CityGML Encoding Standard 2012: 25.)

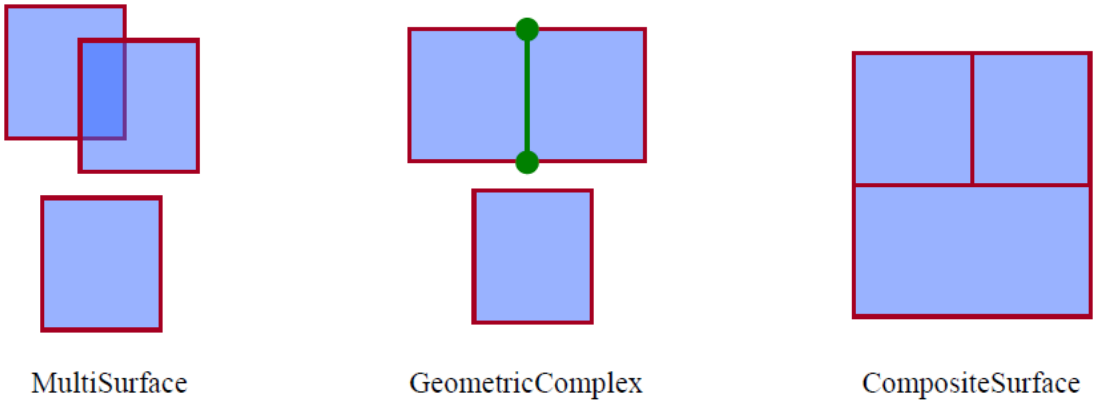


Kuva 2. UML-diagrammi CityGML:n geometriamallista (OGC CityGML Encoding Standard 2012: 25)



Kuva 3. UML-diagrammi CityGML:n geometriamallin komplekseista ja yhdistelmistä. (OGC CityGML Encoding Standard 2012: 26)

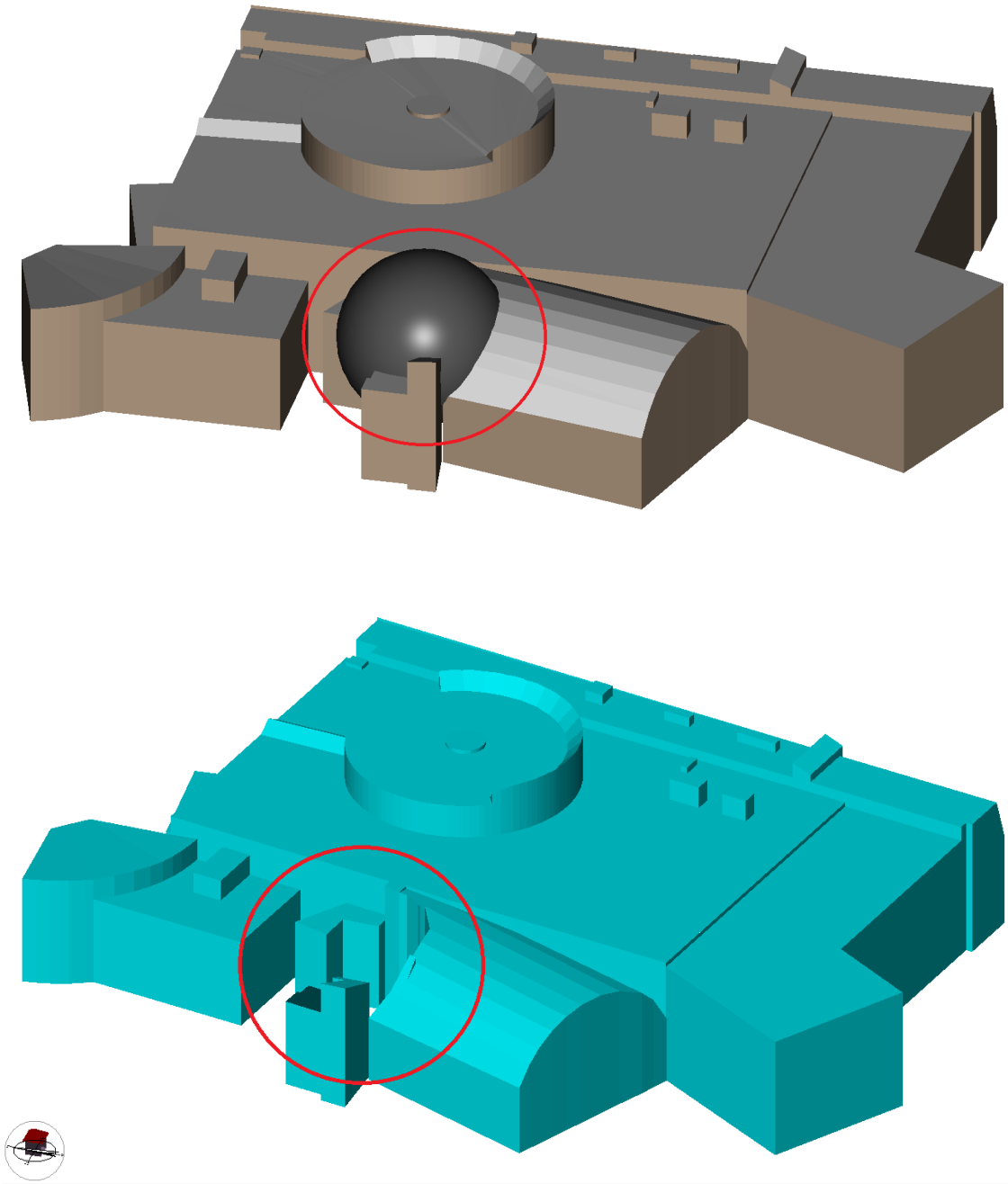
Useimmat kohteet CityGML-kaupunkimallissa koostuvat yhdistetyistä geometrioista, jotka voivat olla yhdistelmiä, komplekseja tai perusalkioiden yhdistelmiä. Yhdistelmien komponenttien välisille spatiaalisille suhteille ei ole rajoituksia. Yhdistelmät voivat siis läpäistä toisiaan, olla päällekkäin, koskettaa tai olla irrallaan toisistaan. Yhdistelmän komponenteille on jokaiselle ulottuvuudelle omat yhdistelmät: MultiPoint, MultiCurve, MultiSurface ja MultiSolid (kuva 3, alareuna). Esimerkki yhdistelmän (MultiSurface) spatiaalisille suhteille on esitetty kuvassa 4 vasemmalla. Komplekseilla on taas rakennettu topologia, ja täten spatiaaliset suhteetkin ovat rajatumpia. Osat eivät saa olla päällekkäin tai jatkuvia, ja ne saavat koskea toisiaan rajoilta tai jakaa rajoja keskenään. (kuva 4, keskellä). Komposiitti on erityiskompleksi, jonka kaikki osat ovat samassa ulottuvuudessa, ja jonka kaikki osat ovat yhdistyneinä rajoiltaan. Yksi komposiiteista, CompositeSurface, on esitetty kuvassa 4 oikealla. (OGC CityGML Encoding Standard 2012: 26; Liukkonen. 2015: 32.)



Kuva 4. Yhdistetyt geometriat (OGC CityGML Encoding Standard 2012: 26)

CityGML on rajoittanut kaaren pelkästään GML3:n luokka Linestringiksi, suoraksi viivaksi. Tämä tuottaa ongelmia kaarimaisten kohteiden mallintamisessa, joita ei voi CityGML-muotoiseksi mallintaa (Kuva 5). Tällöin monet kaupunkikohteet, jotka sisältävät kaarimaisia pintoja, kuten useimmat sillat ja vesitornit jäävät pois CityGML-muotoisesta kaupunkimallista. Kaarimaiset kohteet tulisikin mallintaa CityGML-muotoista mallia varten lähtökohtaisesti siten, että kaaret koostuvat varsinaisten kaarien sijaan useista pienistä suorista viivoista tai pallomaiset kohteet pienistä monikulmiopinnoista, jotka esityksessä tarkasteltuna näyttävät tarpeeksi kaarimaisilta tai pallomaisilta. Liukkosen (2015: 31) mukaan kaarimaisten geometrioiden pois rajoittaminen johtuu siitä, että useimmat

paikkatietojärjestelmät eivät tue kaarigeometrioita, joten rajoittamalla kaaret pois varmistetaan laajempi järjestelmätuki.



Kuva 5. Kaarimaisten kohteiden esitys. Ylemmässä kuvakaappauksessa 3D-mallinnus Vantaan Tikkurilassa sijaitsevasta rakennuksesta design file -muotoisena Microstation-ohjelmistossa. Alemmassa sama rakennus CityGML-muotoisena FZK-viewer-ohjelmistossa. Huomaa punaisella ympyröity pallomainen osa rakennusta, joka ei esiinny CityGML-muotoisessa esityksessä.

3.2 Rakennusten mallinnus CityGML:ssä

Kaupunkimallin tärkein ja näkyvin osa ovat kaupungin rakennukset. CityGML:ssä rakennukset on jaoteltu CityGML:n laajennusmoduulin ”Building”. Rakennuksen geometriana suositellaan käytettävän kappaletta (Solid), joka koostuu komposiittipinnasta (CompositeSurface), jonka pinnat taas koostuvat monikulmioista (polygon). LOD2 tasosta lähtien rakennuksilla suositellaan käytettävän ”rajaavaa esitystä”. Se tarkoittaa sitä, että solid elementin tulisi sisältää ulkoisia Xlink-viittauksia rajaaviin monikulmiopintoihin. Silloin Solid-elementille ei suoraan määritetä geometriaa, vaan geometria koostuu kappaleen, kuten rakennuksen rajaavista pinnoista, joiden geometria on määritelty vasta BoundedBy-elementeissä. Xlink-viittaukset on määritelty Solid-elementin sisään. Rakennusta rajaavat seinäpinnat, kattopinnat, maanvastaiset pinnat, ulkoiset katto- ja lattiapinnat ja LOD3-tasolta lähtien myös sulkupinnat, kuten ovet ja ikkunat. (Yli-Tainio & Savisalo 2015a: 14; Yli-Tainio & Savisalo 2015b: 15–16). Xlink-viittaukset ovat topologiamallin toteutustapa CityGML:ssä, ja sen avulla esimerkiksi kaksi kappaletta tai ominaisuutta voi jakaa saman geometrisen tiedon. Esimerkiksi rakennus ja vierekkäinen autotalli voidaan esittää omina kappaleinaan, jotka koskettavat toisiaan. Pinta, jossa molemmat kappaleet koskettavat, esitetään vain kerran ja molemmat kappaleet viittaavat kyseiseen pintaan. (OGC CityGML Encoding Standard 2012: 26.)

Kuvassa 6 on esimerkki Xlink-viittauksesta rakennuksessa: rakennus koostuu kappaleesta (Solid), joka koostuu komposiittipinnasta (CompositeSurface). Komposiittipinnan jokainen jäsen (surfaceMember) on määritelty kyseisen elementin sisälle ja jokaisella jäsenellä on Xlink-viittaus siihen monikulmioon, joka määrittää kyseisen jäsenpinnan geometrian. Xlink-viittaus tapahtuu ”href”-linkityksellä ”gml:id”-tunnisteeseen ja näiden tulee vastata toisiaan.

```
<cityObjectMember>
  <bldg:Building gml:id="GML_7b1a5a6f-ddad-4c3d-a507-3eb9ee0a8e68">
    <gml:name>Example Building LOD2 </gml:name>
    …  (further attributes see LOD1 example)
    <bldg:lod2Solid>
      <gml:Solid>
        <gml:exterior>
          <gml:CompositeSurface>
            <!-- Ground Slab -->
            <gml:surfaceMember xlink:href="#GML_d3981803-d4b0-4b5b-969c-53f657594757" />
            <!-- Wall South -->
            <gml:surfaceMember xlink:href="#GML_1d350a50-6acc-4d3c-8c28-326ca4305fd1" />
            <!-- Wall North -->
            <gml:surfaceMember xlink:href="#GML_d3909000-2f18-4472-8886-1c127ea67df1" />
            <!-- Wall East -->
            <gml:surfaceMember xlink:href="#GML_6286ffa9-3811-4796-a92f-3fd037c8e668" />
            <!-- Wall West -->
            <gml:surfaceMember xlink:href="#GML_5cc4fd92-d5de-4dd8-971e-892c91da2d9f" />
            <!-- Roof North -->
            <gml:surfaceMember xlink:href="#GML_ec6a8966-58d9-4894-8edd-9aceb91b923f" />
            <!-- Roof South -->
            <gml:surfaceMember xlink:href="#GML_b41dc792-5da6-4cd9-8f85-247583f305e3" />
          </gml:CompositeSurface>
        </gml:exterior>
      </gml:Solid>
    </bldg:lod2Solid>
    <bldg:boundedBy>
      <bldg:GroundSurface>
        <gml:name>Ground Slab</gml:name>
        <bldg:lod2MultiSurface>
          <gml:MultiSurface>
            <gml:surfaceMember>
              <gml:Polygon gml:id="GML_d3981803-d4b0-4b5b-969c-53f657594757">
                <gml:exterior>
                  <gml:LinearRing>
                    <gml:posList>458875.0 5438350.0 112.0 458875.0 5438355.0 112.0 458885.0 5438355.0 112.0 458885.0 5438350.0
                    112.0 458875.0 5438350.0 112.0 </gml:posList>
                  </gml:LinearRing>
                </gml:exterior>
              </gml:Polygon>
            </gml:surfaceMember>
          </gml:MultiSurface>
        </bldg:lod2MultiSurface>
      </bldg:GroundSurface>
    </bldg:boundedBy>
```

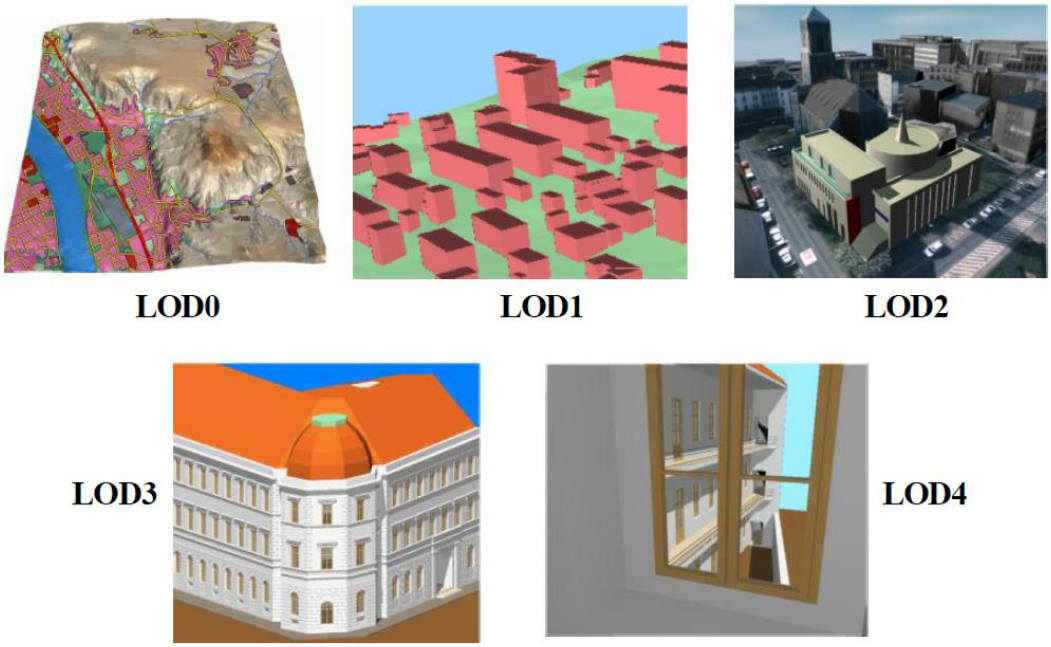
Kuva 6. Esimerkki LOD2-tasoisen rakennuksen CityGML koodista (OGC CityGML Encoding Standard 2012: 275).

Edellä esitetty tapa ei kuitenkaan ole pakollinen, eikä ainoa mahdollinen tapa mallintaa rakennuksia, sillä CityGML:n instanssitiedot validoituvat oikein millä tahansa GML-validilla geometrialla. Kyseinen tapa on hyödyksi, jos mallia halutaan käyttää hyödyksi kaikkine semanttisine ominaisuuksineen, mutta tehokkaaseen visualisointiin ja ylläpitoon riittää myös yksinkertaisempi geometriaesitys. (OGC CityGML Encoding Standard 2012: 27; Yli-Tainio & Savisalo 2015b: 15.)

3.3 LOD-tarkkuustasot

CityGML:ssa kaupunkimalli on jaettu viiteen eri Level Of Detail (LOD) -tarkkuustasoon. Alin taso on LOD0 ja ylin LOD4, joista LOD0 on kaikista yksinkertaisin ja LOD4 on tarkin.

LOD-tasot helpottavat kaupunkimallin käyttöä tehokkaampaan visualisointiin ja analyysien tekemiseen. Tärkeämmät kohteet voidaan esittää tarkempina ja vähemmän tärkeämmät epätarkemmalla tasolla. Sama kohde voi esiintyä CityGML-tiedostossa useammalla eri tarkkuustasolla, jolloin tarkkuustason voi valita tietyn käyttötarpeen mukaan. Visualisoinnissa voidaan käyttää tarkempia tasoja, ja analyyseissä, esimerkiksi lukumäärien laskemisessa, riittävät epätarkemmat tasot.



Kuva 7. Havainnekuva viidestä LOD-tasosta (OGC CityGML Encoding Standard 2012: 11)

Tarkkuustasojen alin taso LOD0 on periaatteessa vain maastomalli, jonka päälle on lisätty esimerkiksi ilmakuva tai kartta alueesta. Rakennukset voidaan esittää monikulmioina tasolla rakennuksen pohjapiirroksen tai ulkonevien kattorakenteiden perusteella. LOD1-tasolla rakennukset ovat palikkamaisia. Katot ovat tasaisia, eivätkä sisällä tarkkaa geometriaa kattorakenteen, kuten esimerkiksi katon harjan muodoista. LOD2-tasosta lähtien rakennuksilla on jo kattomuotoja. LOD3 on jo käytännössä arkkitehtuuri tason malli, jossa on yksityiskohtaiset piirteet rakennuksen katosta ja seinistä. LOD3 sisältää jo esimerkiksi ikkunat sisennyksineen ja sadevesirännit. LOD4-taso täydentää LOD3-mallin sisällyttämällä myös rakennuksen sisäiset piirteet, kuten huoneet ovineen, portaikot, ja joskus jopa huonekalut. Jokaiselle tasolle voidaan vielä lisätä tekstuurit rakenteiden pinnoille. (OGC CityGML Encoding Standard 2012: 11.) Tuotettaessa kaupunkimallia LOD2-tarkkuustaso on riittävä tarkimmaksi tasoksi. Useimmat analyysit, sovellukset ja simulaatiot eivät vaadi sen tarkempaa tasoa, ja kaupunkimallin tuottaminen LOD2-

tarkkuustasolle on mahdollista tuottaa helposti automaattisin tai puoliautomaattisin menetelmin. (Liukkonen 2015: 48; Ahokas 2014: 23.)

3.4 CityGML:n hyödyt

Kaupunkimalleja, varsinkin semanttisia, käytetään yhä enenevin määrin erilaisissa sovelluksissa ja eri tahojen toimesta. Lisäksi semanttisten kaupunkimallien tarpeellisuus kasvaa koko ajan ja sovelluskohteita syntyy jatkuvasti lisää. Tällöin semanttisen kaupunkimallin tuotanto olisi taloudellisesti kannattavaa vain, jos sitä voidaan käyttää useiden eri tahojen toimesta useissa eri sovellusalustoissa. CityGML pyrkiikin juuri olemaan yleinen määritys kaupunkimallille, ja se on jo hyvin pitkälle standardisoitu. Lisäksi CityGML on ilmainen sekä kaikille avoin, ja se on mahdollista ottaa käyttöön käytännössä kaikkialla, missä kaupunkimalleja käytetään tai tuotetaan. Standardisoituna se mahdollistaa useiden eri aineistojen ja ohjelmistojen yhteiskäytön. Tiedon käsittely, tallennus, esittäminen ja jakaminen helpottuvat silloin huomattavasti. (Kolbe 2009b: 15–16, Erving 2008: 4.) Ervingin listaamia (2008: 4) CityGML:n hyötyjä ovat seuraavat ominaisuudet:

- Tietosisällölle saadaan yhdenmukaisempi esitys- ja tallennusmuoto.
- Tiedon hakeminen helpottuu, kun semantiikka on mukana standardimuodossa.
- Aineiston monikäyttöisyys erilaisissa sovelluksissa lisääntyy.
- Aineistojen käsittelyvaiheiden automatisointi yhdenmukaistuu.
- Ollaan riippumattomia tietystä ohjelmistotoimittajasta.
- Tiedon pitkäaikaissäilyvyys parantuu.
- Tietojen yhdistäminen helpottuu.

Toisaalta CityGML ei ole varsinaisesti paras mahdollinen tiedostomuoto mallin esittämiseen, vaan ikään kuin pohjamalli, laaja lähtötieto kaupunkimalleille, joka muuntautuu helposti muihin kaupunkimallin esitystä varten kehitettyihin formaatteihin.

4 Kaupunkimallin ylläpito

Kaupunkimallin ylläpito on käytännössä kaupunkimallin tuottamista, mutta pienemmällä alueella ja tiheämmällä välillä. Ylläpidossa malliin tuotetaan yksittäisiä kohteita tai alueita, jotka puuttuvat jo tuotetusta mallista kaupunkirakenteen muuttuessa. Ylläpitämisellä pyritään pitämään tuotettu kaupunkimalli mahdollisimman ajantasaisena.

Kaupunkimallin ylläpitoon sisältyy haasteita. Muutokset kaupunkirakenteessa on haasteellista tuoda ylläpitoprosessien läpi riittävän nopeasti. Kaupunkirakenne muuttuu jatkuvasti, kun uusia kohteita rakennetaan tai puretaan reaalimaailmassa. Malli ”rappeutuu” ja erkaantuu reaalimaailmasta, jolloin sitä ei voida käyttää kaikista ajankohtaisinta dataa vaativissa sovelluksissa. Richterin sanoin (2015: 4) 3D-kaupunkimalli on kuin pikainen valokuva reaalimaailmasta, joka alkaa vanhentua heti datan hankintahetkeltä alkaen. Automaattisen tai puoliautomaattisen kaupunkimallinnuksen tärkein lähtödatan keruu, laserkeilaus, suoritetaan isoista alueista yleensä vain muutaman vuoden välein, jolloin malli päivittyy pelkästään sen perusteella liian hitaasti. Vantaan kaupungilla vaihtoehtoisia menetelmiä isolle laserkeilaukselle LOD2-tason kaupunkimallin tuottamista varten ei käytännössä vielä ole, eikä pienempienkään alueiden tai tärkeimpien kohteiden manuaalinen mallinnus ole mahdollista, vaikka niin haluttaisiinkin tehdä.

4.1 Vaihtoehtoja ylläpitoon

Ylläpitoa varten mallia voidaan myös tuottaa muilla keinoilla yksittäisistä, valmistuvista rakennuksista, jotka perustuvat esimerkiksi rakennusvaiheessa kerättyyn geometriadataan tai suunnitteluvaiheessa tehtyihin malleihin. Toisaalta puutteina tai rajoitteina voi näillä menetelmillä olla muun muassa muuta mallia alempi tarkkuustaso, fotorealistisuuden puute, yhteensovittamiseen liittyviä haasteita tai muita käytännön ongelmia.

4.1.1 Ylläpito kaksiulotteisen datan avulla

Kaupunkimallia voi päivittää ilman pistepilvidataa kaksiulotteisen rakennusten pohjapiirrosten avulla. Kun rakennuksen kulmapaikkojen koordinaatit tiedetään ja valmistunut rakennus viedään kantakartalle monikulmiona pohjapiirroksena, siitä voidaan pursottaa LOD1-tarkkuustason laatikkomalli haluttuun korkeuteen. (Kuva 8.) Korkeus voidaan asettaa tai arvioida pursotukselle esimerkiksi kiinteistöjärjestelmän tietojen avulla kerroslukumäärän tai jonkun muun korkeutta määrittävän ominaisuustiedon perusteella. (Biljecki 2017: 126.)

Kuva 8. LOD1-tarkkuustasoisen rakennuksen pursotus pohjapiirroksesta. Vasemmalla pohjapiirros, keskellä pohjapiirrokset verrattuna olemassa olevaan kaupunkimalliin ja oikealla vaaleammalla pohjapiirroksista pursotetut rakennukset. (Biljecki 2017: 126.)

Vastaavanlaisella menetelmällä onnistuu myös LOD2-tasoisen kaupunkimallin tuotanto, jos kantakartassa tai kiinteistöjärjestelmässä on tieto rakennuksen katon muodoista. Harjakattoisille rakennuksille tämä vaatisi esimerkiksi tietoa myös harja- ja räystäskorkeuksista.

Menetelmä on oivallinen esimerkiksi kunnan kaupunkimallin ylläpitoon, sillä kunnissa kerätään menetelmää varten vaadittavat rakennusten kulmapisteiden geometrioita joka tapauksessa muun muassa rakennusten tarkistusmittauksissa. Tällöin LOD1-tasoisen mallin tuottamista varten vaaditut geometriset tiedot uusista rakennuksista päivittyvät ikään kuin automaattisesti, ja tuotanto voi olla koko aika käynnissä.

4.1.2 Mallinnus suunnitelmapiirustuksista

Esimerkiksi Helsingin kaupungilla on käytössään menetelmä, jossa rakennus mallinnetaan LOD2-tasolle suunnitelmapiirustuksista. Tarkistusmittauksilla saaduista rakennuksen kulmapistetiedoista pursotetaan yllä mainittuun tapaan seinäpiirteet, ja rakennusten kattomuodot mallinnetaan ohjelmallisesti suunnitelmapiirustuksista. Rakennus muodostuu tällöin LOD2-tason malliksi. Menetelmällä malli ajantasaistuu alusta lähtein LOD2-tasolla, ja LOD1-, sekä LOD0-tason mallit tulevat ikään kuin sivutuotteena automaattisesti. (Hårdh 2018.)

4.1.3 BIM/IFC-mallien vieminen kaupunkimalliin

Rakennuksen tietomalli (BIM, Building Information Model) on kolmiulotteinen malli rakennuksesta, ja se sisältää myös yksityiskohtaisia tietoja rakennuksesta. BIM-malliin on lisätty rakennuksen geometrian lisäksi myös rakennuksen semantiikkaa ja tietoja, joita tarvitaan rakentamisen, osien valmistuksen ja hankintatoimien tukena rakennusvaiheessa. Sen avulla voidaan esittää reaaliaikaisesti rakennuksen eri vaiheet koko sen elinkaaren ajan digitaalisessa muodossa, jota voidaan myös reaaliaikaisesti jakaa eri toimijoiden kesken. BIM-malleja käytetään laajalti arkkitehtuurin, suunnittelun ja rakentamisen alalla. Avoimia tietomalleja BIM-malleille on esimerkiksi standardisoitu IFC (Industry Foundation Classes). (Mitä on BIM? 2018; Liukkonen 2015: 24.)

IFC-malleja voidaan liittää kaupunkimalliin, mutta liittämisessä on omat haasteet CityGML-yhteensopivuuden kannalta. Lisäksi ajantasaisen ylläpidon kannalta, IFC-malleja tulisi saada kaupunkimallia tuottavan osaston käyttöön jatkuvalla syötöllä. Tämä voitaisiin ratkaista esimerkiksi vaatimalla IFC-malli osaksi rakennuslupaprosessia. Lisää IFC-mallien käytännön liittämisestä kerrotaan seuraavassa luvussa.

4.2 Tietokanta ja ylläpito

Kaupunkimallien tehokkaan hallinnan, käytön ja varastoinnin vuoksi, malli on syytä siirtää tietokantaan. Tietokannassa mallia voidaan käsitellä tehokkaammin esimerkiksi tietokantojen suomien kyselyiden avulla. CityGML-muotoisessa mallissa mallin kohteet ovat myös yksilöityjä. Näin tietokannan avulla voidaan hallita, tarkastella ja muokata esimerkiksi yksittäisiä kohteita, mutta myös laajempia alueita. Alueetkin voidaan rajata helposti siihen, mitä käyttötarkoitus vaatii, vaikka kaupunginosan tai itse vapaasti piirretyn rajauksen mukaan. Esimerkiksi Vantaalla kaupunkimalli on tällä hetkellä jaoteltu karttalehdittäin osiin ja mallin korjailu, ylläpito sekä tarkastelu suoritetaan karttalehti kerrallaan. Ylläpidon kannalta tietokanta mahdollistaisi muun muassa tehokkaan tavan tunnistaa ja irrottaa muokattavaksi esimerkiksi kaikki puretut rakennukset tietystä ajankohdasta eteenpäin yhdellä kyselyllä, sen sijaan että tutkittaisiin mallia karttalehti kerrallaan ja koitettaisiin löytää puretut rakennukset vertaamalla niitä kantakarttaan tai ilmakuviin.

3D City Database (3DCityDB) on ilmainen avoimeen lähdekoodiin perustuva tietokantaskeema ja sarja sovellustyökaluja kolmiulotteisten kaupunkimallien tietokantaan viemiseen ja sieltä uloskirjoittamiseen, hallintaan, analysointiin sekä visualisointiin CityGML-

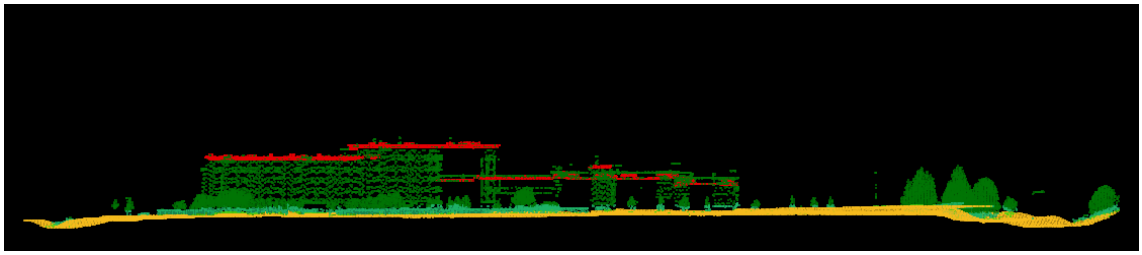
standardissa. Se on kehitetty Münchenin teknillisen yliopiston, virtualcitySYSTEMS:in ja M.O.S.S Computer grafik systemen yhteistyönä. Se toimii joko Oracle- tai PostGIS-tietokantahallintajärjestelmissä ja pystyy käsittelemään hyvinkin suuria, miljoonista 3D-objekteista ja sadoista miljoonista geometrioista ja tekstuurikuvista koostuvia malleja monella eri tarkkuustasolla. Se soveltuu erittäin tehokkaasti kolmiulotteisen kaupunkimallin hallintaan sekä ylläpitämiseen ja on jo täydessä käytössä useassa eri kaupungissa Euroopassa. (3D City Database for CityGML 2016: 13.)

5 Ylläpitomenetelmät Vantaalla

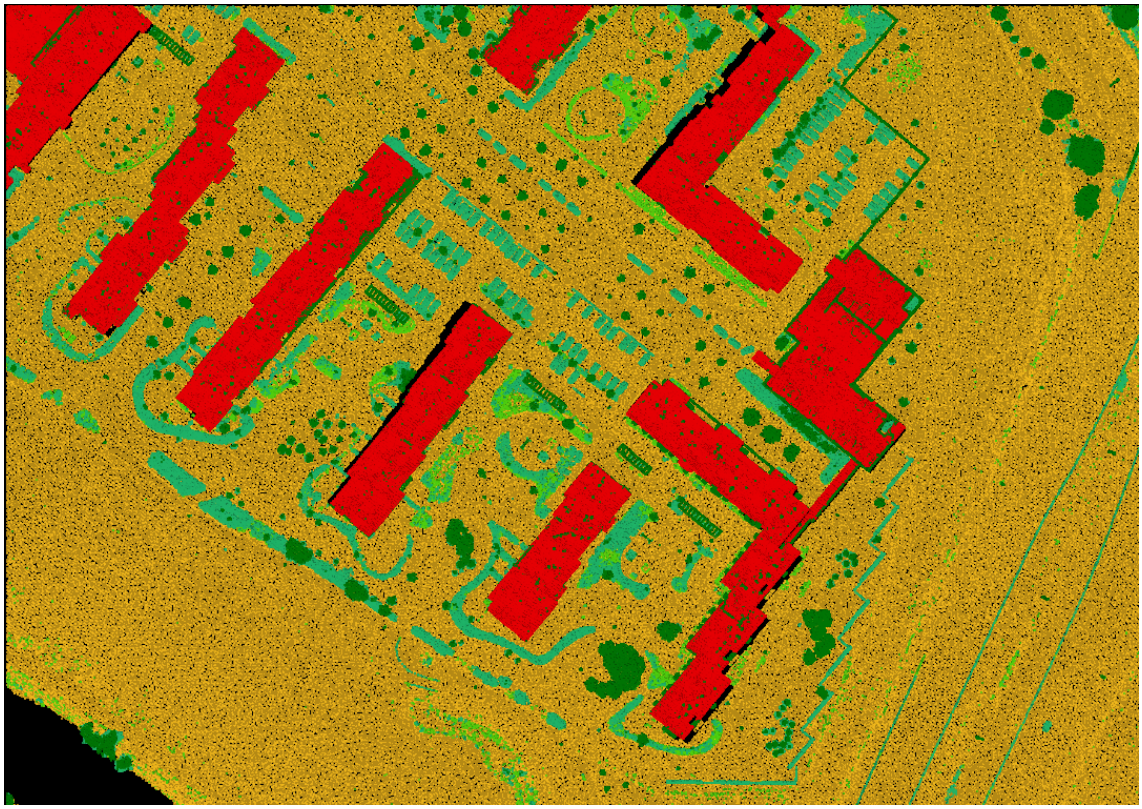
Vantaan kaupungilla ei ole vielä varsinaista jatkuvaa ja yhtenäistä, selvää 3D-kaupunkimallin ylläpitomenetelmää. Kaupunkimallia tuotetaan LOD2-tarkkuustasolle muutaman vuoden välein kerätystä laserkeilaus- ja ilmakuva-aineistoista. Vanhemmasta aineistosta tuotettu malli päivitetään uudemman aineiston pohjalta. Vantaa on suorittanut laserkeilaukset koko kaupungista vuosina 2012 ja 2016, ja keilauksia tullaan tulevaisuudessa suorittamaan kolmen vuoden välein. Ohjelmistoina Vantaalla käytetään kaupunkimallin tuotannossa Microstationia sekä Microstation-ympäristössä toimivia TerraSolid-tuoteperheen ohjelmistoja kuten TerraScania.

5.1 Nykyinen prosessi

Kaupunkimallin tuotantoprosessi alkaa lähtödatan, eli laserkeilausaineiston keruusta. Vantaa kilpailuttaa laserkeilauksen ulkoisille yrityksille, jotka tuottavat luokittelematonta pistepilviaineistoa käytettäväksi Vantaalle. Saatu pistepilviaineisto luokitellaan eri kohteiksi. (Kuvat 9 ja 10.) Kohteita ovat mm. rakennukset, kasvillisuus, sekä maanpinta. Kasvillisuus on erikseen vielä luokiteltu korkean, keskitason ja matalan kasvillisuuden luokkiin. Luokittelu voidaan suorittaa TerraScanin automaattisella luokittelulla. Vuoden 2012 laserkeilausaineisto oli kuitenkin verrattain harvaa (10 pistettä neliömetrillä), joten rakennusten luokittelua parannettiin rakennusten pohjapiirrosten avulla. Luokittelussa rakennuksiksi luokiteltiin vain ne pisteet, jotka osuivat kantakartasta irrotettujen rakennusten pohjapiirrosten sisään. Samalla vältyttiin myös ylimääräisiltä ”roskilta”, eli kohteilta, jotka eivät kuulu malliin mutta joita automaattinen luokittelu saattaa luulla rakennuksiksi.



Kuva 9. Poikkileikkaus luokitellusta pistepilvidatasta. Punaisella rakennus, keltaisella maanpinta ja vihreällä kasvillisuus.



Kuva 10. Luokiteltu pistepilvidata yläperspektiivistä.

TerraScan luo luokitelluista pisteistä automaattisesti LOD2-tason malleja rakennuksista. Yksinkertaisemmat rakennukset mallintuvat automaattisella menetelmällä yleensä hyvin ja tarkasti, mutta etenkin isoimpien ja monimutkaisten kattorakenteiden omaavien rakennusten automaattinen mallinnus vaatii tarkastelua ja usein myös manuaalista käsin korjailua, jotta malli olisi riittävän tarkka ja totuudenmukainen.

Tuotetun malli päivittäminen alkoi, kun uusi laserkeilausaineisto saatiin käytettäväksi. Vuoden 2016 pistepilviaineisto luokiteltiin pelkästään TerraScanin automaattisella luokit-

telulla, sillä tällä kertaa aineisto oli tiheämpää, jonka myötä myös automaattinen luokittelu on luotettavampaa. Jotta vältytään samojen kohteiden kaksinkertainen mallinnus, uudesta laseraineistosta tulisi erotella vain ne mallinnettavat kohteet, joita ei ole aiemmassa mallissa. Tätä varten luotiin automaattinen työnkulku, niin sanottu makro- prosessi TerraScanissa. Macro ryhmittelee ensiksi pisteet rakennusten kattopintojen mukaisesti ryhmiin ja vertaa näitä ryhmiä rakennuksen pohjapiirroksesta luotuun monikulmioon. Jos ryhmitelty pistepilvi on monikulmion sisällä, se siirtyy väliaikaiseen pisteluokkaan. Monikulmiot on luotu keilausajankohdan kantakartasta, jolloin ne pisteet, jotka eivät ole jo monikulmion sisällä, ovat mallista puuttuvia uusien rakennuksien pisteitä. Kun pisteet oli eroteltu, suoritettiin taas TerraScanin automaattinen mallien luonti sekä rakennuksien korjailu minkä jälkeen uusi ja vanha malli yhdistettiin päivitetyksi malliksi. (Kalso 2018.)

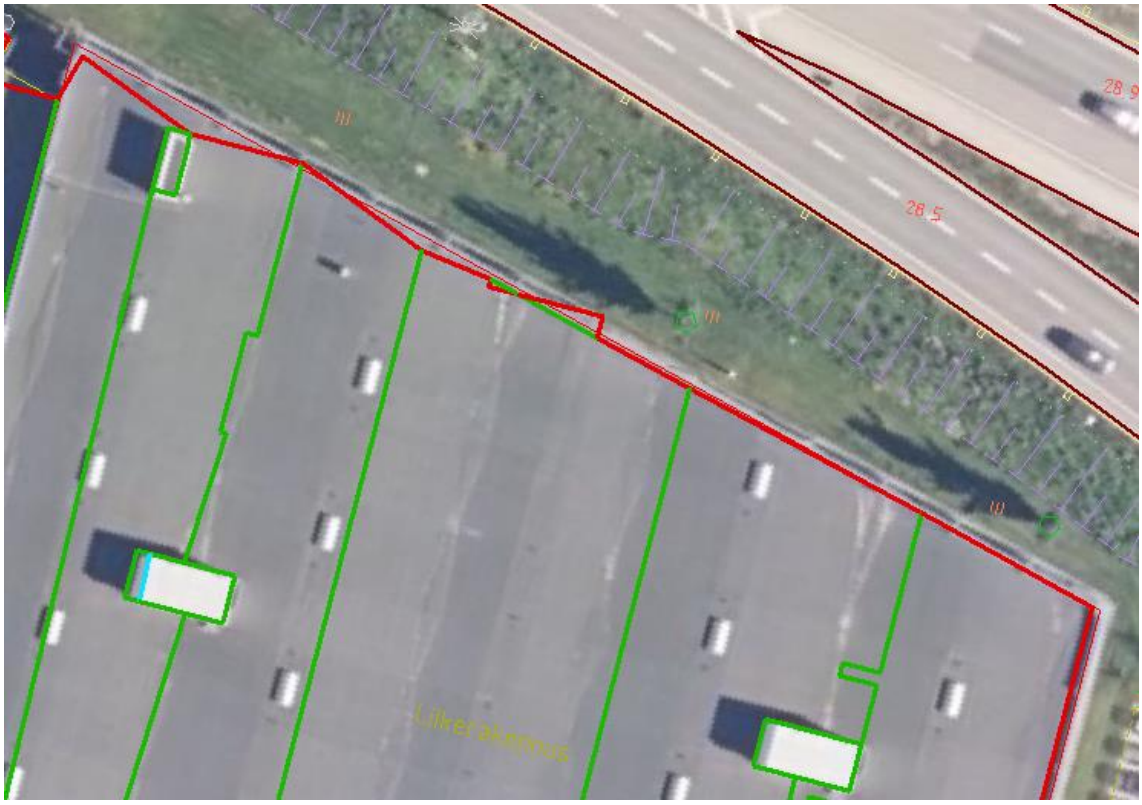
Automaattisen mallinnuksen manuaalinen korjailu

Mallien korjailu tapahtuu TerraScanin omilla korjaustyökaluilla, ja on koko kaupungin mittakaavassa erittäin paljon aikaa vievä prosessi. Ennen kuin malli on kokonaan korjailtu, ja valmis käytettäväksi sekä jaettavaksi, se on jo pahasti jäänyt jälkeen verrattuna reaalimaailmaan. Korjailua varten MicroStationiin avataan automaattisesti mallinnetut rakennukset mallinnusvaiheessa tehdyn jaottelun mukaisesti, Vantaalla yhden neliökilometrin kokoinen karttalehti kerrallaan. Päälle ladataan TerraScan, ja alkuasetusten tarkastamisen jälkeen korjailu voi alkaa. TerraScan tunnistaa ja luettelee luodut rakennusmallit listaan, jonka avulla mallia tarkastellaan ja korjaillaan yksi rakennus kerrallaan. Listatyökalulla (kuva 11.) voi hyväksyä mallin, poistaa sen tai laskea uudelleen eli toisin sanoen, mallintaa uudestaan esimerkiksi pienemmillä yksityiskohtien määrittelyarvoilla. Listalta voidaan valita yksittäinen rakennus tai tunnistaa yksittäinen rakennus poimimalla se koko mallista.

Kuva 11. Kuvakaappaus TerraScan korjailutyökalun rakennusmallien tarkastelulistasta (Check Building Models).

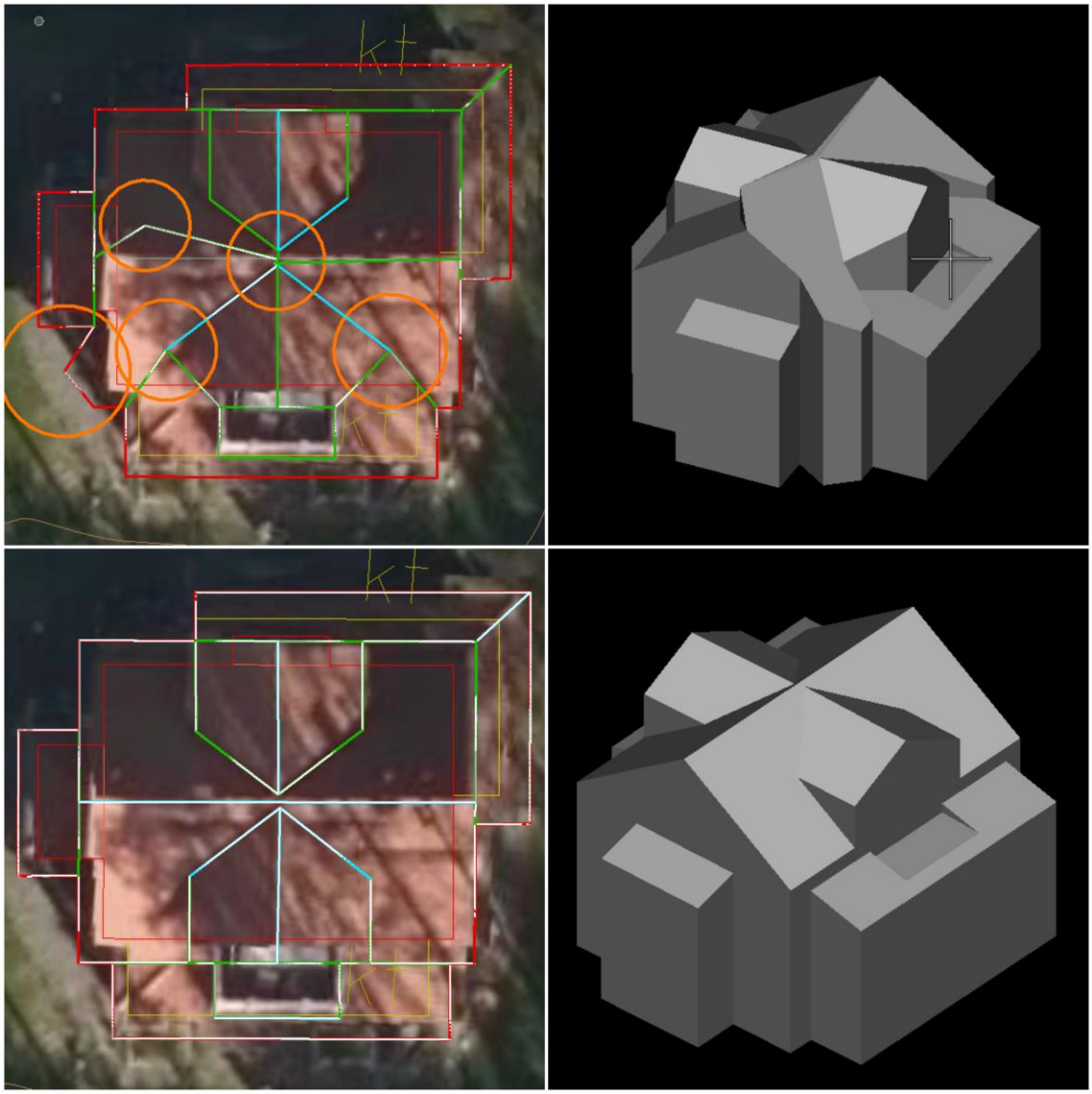
Kun listalla on valittuna rakennusmalli, MicroStation lähentää automaattisesti näkymän kyseisen rakennuksen kohdalle ja mallin ääriviivat korostuvat. Esimerkiksi kuvassa 11 valitun mallin ääriviivat näkyvät vaaleansinisenä, vihreinä sekä punaisina. Muiden, ei-valittujen mallien ääriviivat ovat joko vaaleanruskeita tai harmaita. Punaisella kuvataan rakennuksen ulkoreunat, eli käytännössä seinät, joihin rakennus päättyy. Vihreällä kuvataan niiden seinien reunat, jotka ovat päättyvien seinien sisäpuolella. Siniset ovat ”kohtaamislinjoja” eli esimerkiksi rakennuksen harjat.

Valitun rakennuksen ääriviivoja vertaillaan korjailussa ilmakuviin sekä kantakarttaan, jotka ladataan taustatasoiksi MicroStationiin korjailun ajaksi. Useimmat automaattisen mallinnuksen tekemät virheet liittyvät ääriviivojen virheellisyyteen verrattuna ilmakuviin ja kantakarttaan. (Kuva 12.) Rakennuksen jotkin osat eivät toisin sanoen ole oikeissa kohdissaan ja seinien, sekä kattorakenteiden muodot ovat virheellisiä. Lisäksi yleistä on, että rakennus saa ylimääräisiä ”paloja” automaattisessa mallinnuksessa.



Kuva 12. Kuvakaappaus korjaamattomasta ja virheellisestä esimerkkirakennusmallista. Seinämuodot poikkeavat todellisesta kuvassa rakennuksen vasemmassa yläreunassa ja rakennuksella on ylimääräisiä paloja keskellä rakennuksen seinämää. Ohuempi punainen viiva on taustalla olevan kantakartan rakennuksen reunaviiva.

Rakennuksen reunoja ja kattomuotoja asetellaan oikeille kohdilleen TerraScanin korjailutyökaluilla (Soininen 2016: 209–253) ja oikeat kohdat selviävät ilmakuvista ja kantakartasta. Ääriviivat asetellaan kattojen reunojen mukaisesti kohdilleen. Yläperspektiivin lisäksi mallia tarkastellaan korjailuvaiheessa samanaikaisesti myös isometrisessä näkymässä ilman taustakarttoja, korostettuja ääriviivoja tai ilmakuvia, jotta rakennusmallin oikeellisuudesta saadaan parempi kuva myös kolmiulotteisessa ympäristössä. (Kuva 13.)



Kuva 13. Korjaamaton ja korjailtu esimerkkirakennus Microstationissa. Ylhäällä kuvat korjaamattomasta rakennuksesta yläperspektiivistä, jossa virheet on korostettu oranssein ympyröin, sekä isometrisessä näkymässä. Alhaalla kuvat korjaillusta rakennuksesta samoista perspektiiveistä. Virheitä on käsin liioiteltu havainnollistamisen parantamiseksi

Jos mallista puuttuu osia tai esimerkiksi kattomuodot eivät ole riittävän yksityiskohtaisia tai malli ei muuten vain ole järkevän näköinen kyseiselle rakennukselle, yksittäinen rakennus voidaan mallintaa uudestaan. Uudestaan mallinnuksessa rakennuksen ympärille ladataan pistepilviaineistosta luokitellut pisteet. Pisteitä voidaan väliaikaisesti luokitella tarkimmillaan jopa pistekohtaisesti uudelleen mallinnuksen ajaksi, esimerkiksi tarkkuuden parantamiseksi tai jos osa rakennuksen pisteitä on luokiteltu väärään pisteluokkaan. Lisäksi uudelleen laskennalle voidaan asettaa arvoja, esimerkiksi sille, kuinka pienet pisteiden luomat alueet ja yksityiskohdat otetaan huomioon mallinnuksessa tai kuinka iso

aukko rakennuksiksi luokiteltujen pisteiden välillä saa olla, jotta pisteet huomioidaan samaksi rakennukseksi. Arvoja säätelemällä sekä luokittelemalla pisteitä väliaikaisesti rakennuksesta voidaan saada todenmukaisempi ja puuttuvia osia tuotua esiin. Valmiiksi korjailtu rakennus hyväksytään listassa, ja lista siirtyy automaattisesti seuraavaan korjaamattomaan rakennukseen. (Soininen 2016: 216–218.)

Yksittäisen rakennuksen korjailu kestää rakennuksen monimutkaisuudesta ja koosta riippuen alle minuutista jopa tunteihin. Isot rakennukset, kuten kauppakeskukset ja voimalaitokset, sisältävät yleensä paljon virheitä, joiden korjailu vie suhteellisen runsaasti aikaa. Virheitä täytyy tarkastella jokaisen rakennuksen kohdalla erikseen ja tarvittaessa korjailla. Korjailussa ääriviivojen asettelu ei aina ole helppoa, sillä jokainen rakennuksen ääriviiva, eli vektori, on mallissa kiinnittyneenä toisiinsa kulmapistein. Yhden viivan siirtäminen voi siten horjuttaa muita vektoriin kiinnittyneitä kulmapisteitä ja ääriviivavektoreita, jolloin yhden ääriviivavektorin siirtäminen oikealle kohdalleen saattaa vaatia useamman muun kulmapisteen ja vektorin siirtämistä ja muokkausta. Tämä ongelma korostuu etenkin silloin, kun rakennuksessa on useampi harja kiinnittyneenä samaan kulmapisteeseen. Yksittäisen rakennusmallin uudelleenlaskenta voi viedä tietokoneen tehoista ja uudelleen laskentaa varten ladatun pistepilven koosta riippuen jopa kymmenen minuuttia, ja uudelleen laskennankin jälkeen on edelleen syytä tarkastella mallin oikeellisuutta.

Rakennusten korjailu muodostuu ylläpidossa yhdeksi pullonkaulaksi. Yhden karttalehden korjailuun saattaa kulua aikaa karttalehden sisältämien rakennusten määrästä ja laadusta riippuen jopa kaksi työpäivää yhden korjailevan työntekijän toimesta. Harjaantunut työntekijä huomaa virheitä ja korjailee rakennuksia nopeammin, ja harjaantuminen sekä korjailutyökalujen opettelu ja ymmärtäminen vie oman aikansa, mikä myös hidastaa prosessia. Lisäksi esimerkiksi vuoden 2016 laserkeilauksen korjauksen yhteydessä tarkasteltiin myös kuntarekisterin rakennuskohteiden suhdetta rakennuksen 3D-malliin. Jos yhdellä rakennuksella on useita rakennusnumeroja, tulee rakennusnumeroiden mukaiset osat erottaa mallissa omiksi rakennuksikseen. Tämän takia myös jo aiemmin, vuoden 2012 laserkeilauksen, korjatut rakennukset tuli joka tapauksessa tarkastaa uudestaan siltä varalta, ettei yhdellä yksittäisellä mallilla ole useampaa rakennusnumeroa, vaikka olisikin ollut mahdollista tarkastaa ja korjailla vain uusia luotuja rakennuksia.

5.2 Uusi prosessi

Uudessa prosessissa kaupunkimallin ylläpito tapahtuu tietokannan kautta. Tietokannan avulla on mahdollista päivittää mallin rakennuksia esimerkiksi rakennuskohtaisesti jo olemassa olevaan malliin. Menetelmä parantaa ylläpidon ja kaupunkimallin päivityksen mahdollisuuksia, sillä yksittäinen, valmistuva tai päivitystä kaipaava rakennus voidaan erotella tietokannasta sen sijaan, että rakennus ”etsittäisiin” esimerkiksi karttalehtijakoisesta kaupunkimallista. Lisäksi ennen kaupunkimallin rakennuksilla ei ollut varsinaisesti kiinteistöjärjestelmää vastaavaa erottelua esimerkiksi rakennustunnuksittain. CityGML-pohjainen kaupunkimalli mahdollistaa muun muassa kiinteistöjärjestelmän tietojen lisäämisen rakennuskohtaisesti rakennuksille.

Vuoden 2016 laserkeilauksesta tuotettu sekä korjailtu malli konvertoidaan teksturoituna CityGML-muotoon ja tallennetaan 3D City Database -tietokantaan. Konvertoinnissa rakennuksille lisätään myös ominaisuustietoja kiinteistöjärjestelmästä. Tämä kaupunkimalli muodostaa pohjan, jonka päälle aletaan päivittää mallia.

Kun uusi laserkeilaus- ja ilmakuvausaineisto saapuu käsiteltäväksi, alkaa mallin päivitys. Mallia varten pistepilvi edelleen luokitellaan ja käytetään hyväksi TerraScanin automaattista mallien luontia. Tietokantapohjaisessa päivityksessä voidaan käyttää hyväksi tietokantakyselyjä, joiden perusteella uusien, mallintamattomien rakennusten erottelu tehostuu ja roskien määrä automaattisessa mallinnuksessa vähenee. Lisäksi on mahdollista tunnistaa automaattisesti myös laajennetut rakennukset, jolloin tietokannasta voidaan poistaa tai siirtää eri tasolle vanhentunut rakennusmalli ja luoda uusi laajennettu rakennus yhtenä rakennuksena. Aiemmin laajennukset mallintuivat omiksi rakennuksikseen rakennuksen viereen, ja mallit jouduttiin laskemaan yhdeksi uudeksi rakennukseksi manuaalisesti.

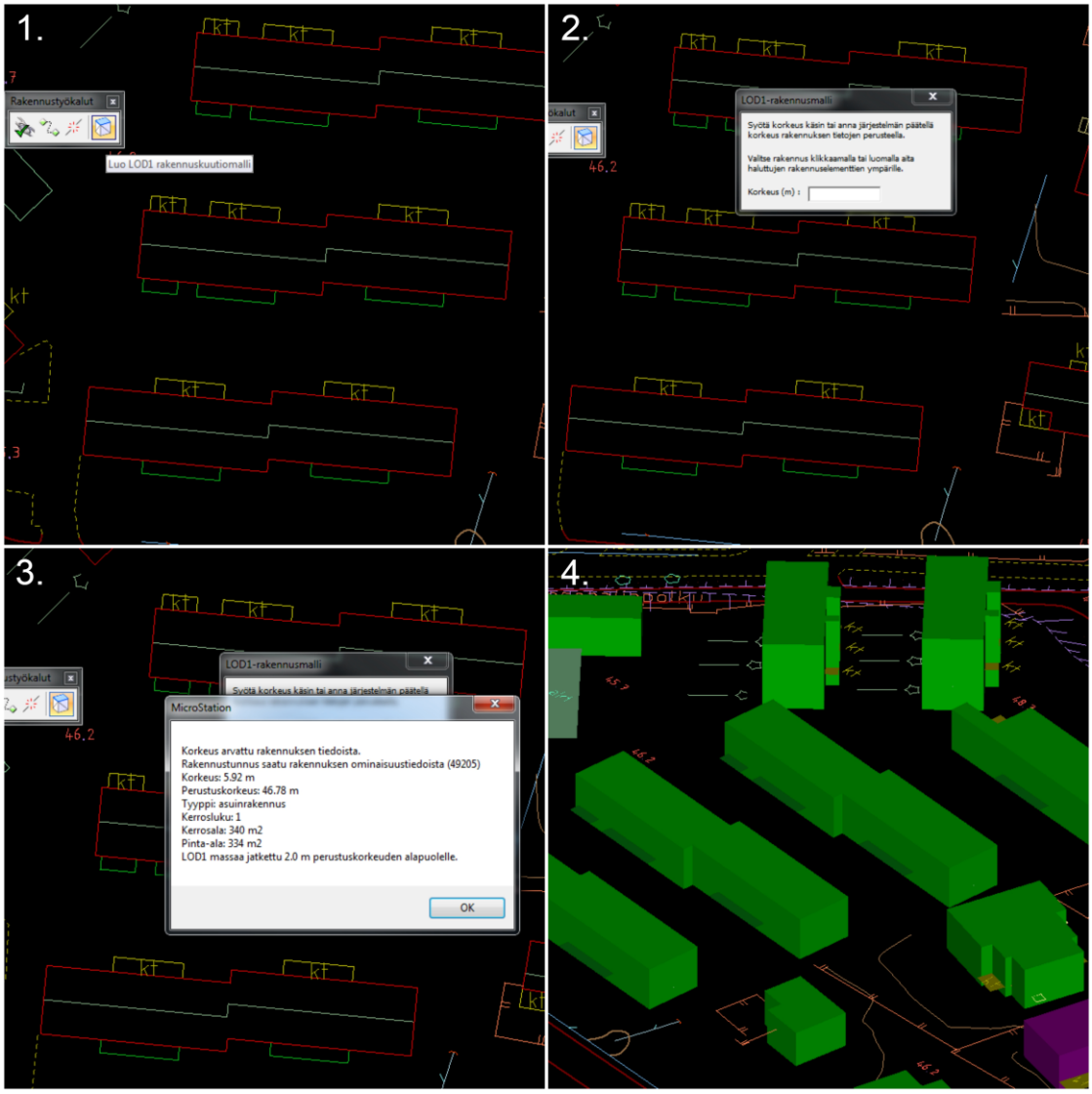
Mallien luonnin jälkeen on edelleen edessä uusien rakennusten manuaalinen tarkistus ja korjailu. Kaikki uudet rakennukset luodaan ja teksturoidaan sekä viedään korjaamattomana tietokantaan. Korjailu suoritetaan rakennuskohtaisesti tunnistamalla korjaamattomat rakennukset tietokannasta esimerkiksi päivämäärä ominaisuustiedon avulla ja tuomalla rakennus tietokannasta muokattavaksi. Muokattu malli palautetaan tietokantaan korjattuna. Malli päivittyisi heti tietokannassa uusien rakennusten kohdalla. Rakennukset

olisivat alkuun korjaamattomia ja osittain virheellisiä, mutta korjatut rakennukset päivittyvät tietokannassa malliin jatkuvalla syötöllä, ja kokonaan korjailtu, valmis malli saavutettaisiin suhteellisen nopeasti.

Edellä mainittu korjailu kuitenkin vaatisi tekstuurien uudestaan asettelun, sillä ensimmäinen teksturointi osuu mahdollisesti virheelliseen rakennukseen, jonka korjailu muuttaa geometriaa ja tekstuurit olisivat vääränlaisia. Ongelma voidaan ratkaista esimerkiksi siten, että korjailtavaksi haetusta rakennuksesta poistetaan tekstuurit kokonaan ja rakennus teksturoidaan uudestaan korjailemisen jälkeen.

Valmistuvien rakennusten päivitys ennen laserdataa

Pistepilveä kerätään suhteellisen harvoin verrattuna valmistuviin rakennuksiin. Neljännessä kappaleessa esiteltiin vaihtoehtoja ongelman ratkaisemiseksi. Varteenotettavin vaihtoehto kirjoitushetkellä on LOD1-tasoisen kuutiomallin lisääminen malliin valmistuvan rakennuksen osalta, sillä Vantaalla luodaan jo LOD1-mallia valmistuvien rakennusten osalta. Mallit luodaan pursottamalla kuutio arvattuun korkeuteen kiinteistöjärjestelmästä saatujen tietojen perusteella siinä vaiheessa, kun rakennuksesta on tehty sijaintikatselmus ja tämän tiedon perusteella luotua pohjapiirrosta ollaan viemässä kantakarttaan. Menetelmää varten on luotu puoliautomaattinen prosessi, joka toimii Microstationissa. Kantakarttaan viedyn rakennuksen pohjapiirros valitaan, ja prosessi hakee automaattisesti vastaavan rakennuksen korkeustiedot kuntarekisteristä sekä pursottaa siitä kuutiomaisen rakennusmallin, jonka sivut ovat pohjapiirroksen mukaiset. Korkeus rakennukselle arvioidaan menetelmällä, jossa valittua rakennusta vertaillaan sen käyttötarkoituksen, sijainnin ja muiden tekijöiden perusteella vastaavanlaisiin rakennuksiin. Menetelmä valitsee parhaimman arvioidun korkeuden rakennukselle. Korkeuden voi myös syöttää käsin, jos se on tiedossa. (Kuva 14.)



Kuva 14. LOD1-tasoisen kaupunkimallin luontiprosessi Vantaalla. 1) Avataan työkalu mallin luontia varten. 2) Valitaan halutun rakennuksen pohjapiirros, ja syötetään korkeus joko käsin, tai annetaan prosessin arvata korkeus kiinteistöjärjestelmän tietojen perusteella. 3) Prosessi luo mallin ja tulostaa raportin. 4) Valmiita LOD1-malleja kantakartan päällä.

Esimerkkinä Helsingin kaupungilta, voidaan tutkia mahdollisuuksia mallintaa valmistuvia rakennuksia jo alusta alkaen LOD2-tasolle. Tällöin kaupunkimallin ylläpidosta vastaavalla osastolla pitäisi olla käytössään suunnitelmapiirrokset rakennuksista, joista ilmenee kattomuodot. Muodot lisättäisiin LOD1-tasoisiin malleihin esimerkiksi manuaalisesti mallintamalla. Jos rakennusmalleja luodaan ja siirretään tietokantaan osaksi kaupunkimallia suunnitelmien pohjalta, on järkevää laittaa rakennuksille ominaisuustieto sen suunnitelmapohjasta. Ominaisuustiedosta selvenee, että kyseinen malli on tehty suunnitelmista erotuksena valmiista rakennuksista, joiden geometriat on hankittu laserkeilaamalla. Helsingissä rakennusmalleilla on ominaisuustietoina ”suunnitelma”, jos rakennus on mallin-

nettu suunnitelmapiirrosten tai IFC-mallien pohjalta. Ominaisuus vaihdetaan arvoksi ”toteutunut”, kun laserkeilauksen pohjalta tehty malli kyseisestä rakennuksesta on päivitetty kaupunkimalliin. (Hårdh 2018.)

Tulevaisuuden näkymissä Vantaalla on keskusteltu myös rakennustietomallien, IFC-mallien viemisestä kaupunkimalliin. IFC-malleja kerättäisiin osana rakennuslupaprosessia, jota kautta ne saataisiin käyttöön kaupunkimalliin lisäystä varten. Haasteena tässä on kuitenkin IFC-mallien oikea koordinaatisto ja sijoittuminen suhteessa todellisuuteen sekä mallin muuntuminen CityGML-muotoiseksi. Rakennustietomallit lisäksi elävät koko rakennuksen ajan aina vaihtuvien tilanteiden mukaan, joten ensiksi saatu suunnitelmamalli, sekä valmis IFC-malli voivat erota toisistaan. Rakennuslupaprosessia halutaankin kehittää niin, että luvan hyväksymisvaiheessa kaupungille toimitetaan suunnitelmamalli ja loppukatselmuksen yhteydessä toimitetaan toteumamalli. Toteumamallia verrataan sijaintikatselmuksessa mitattuun pohjasijaintiin ja myöhemmin tapahtuvassa laserkeilauksessa varmistetaan, että IFC-malli on toleranssien sisällä, jolloin IFC-malleista luodut kaupunkimalliin vietävät versiot ovat oikeissa koordinaateissaan. IFC-mallien muuntaminen CityGML-malliksi on hankalaa, sillä malleilla on eroja muun muassa semantiikan rakenteessa ja geometrian esityksessä ja muuntaminen vaatii monivaiheisen, monimutkaisen prosessin, jossa kartoitetaan semantiikat ja oikeanlaiset geometriat IFC-mallista CityGML-muotoista esitystä varten. (Donkers 2017: 23–30, 98–101.)

6 Vantaan kaupunkimallin julkaisu

Kaupunkimallin julkaisu, eli visuaalisen lopputuotteen saattaminen loppukäyttäjien käyttöön, tulisi tapahtua siten, että malli saataisiin käytettäväksi loppukäyttäjilleen ajantasaisena. Mallin julkaisun tulisi koskettaa niitä alustoja sekä formaatteja, joita loppukäyttäjät käyttävät tai voisivat käyttää. Mahdolliset tulevaisuuden hankkeet tulee myös ottaa huomioon mahdollisina loppukäyttäjien alustoina. Ajantasaisuuden kannalta ylläpitoprosessin tulee myös ottaa huomioon kaupunkimallin julkaisu niin sanottuna viimeisenä askeleena prosessia.

6.1 Julkaisumuodot

Tällä hetkellä Vantaan kaupunkimallia julkaistaan muun muassa Helsinki Region Infoshare -palvelussa avoimena KML- SketchUp- sekä CityGML-muotoisena mallina. Tavoitteena on saada kaupunkimalli vuoden 2018 aikana selaimeen interaktiiviseksi malliksi, jossa käyttäjä voi tutkia kaupunkimallia kolmiulotteisessa virtuaalisessa ympäristössä internet selaimen avulla. Lisäksi selainpohjaisessa mallissa voisi myös tutkia tiettyjä rakennuksen ominaisuustietoja klikkailemalla rakennuksia selaimessa. Selainpohjainen palvelu olisi tarkoitettu niin kaupungin omien työntekijöiden, kuin myös kuntalaisten käyttöön.

6.1.1 Cesium ja selainpohjainen julkaisu

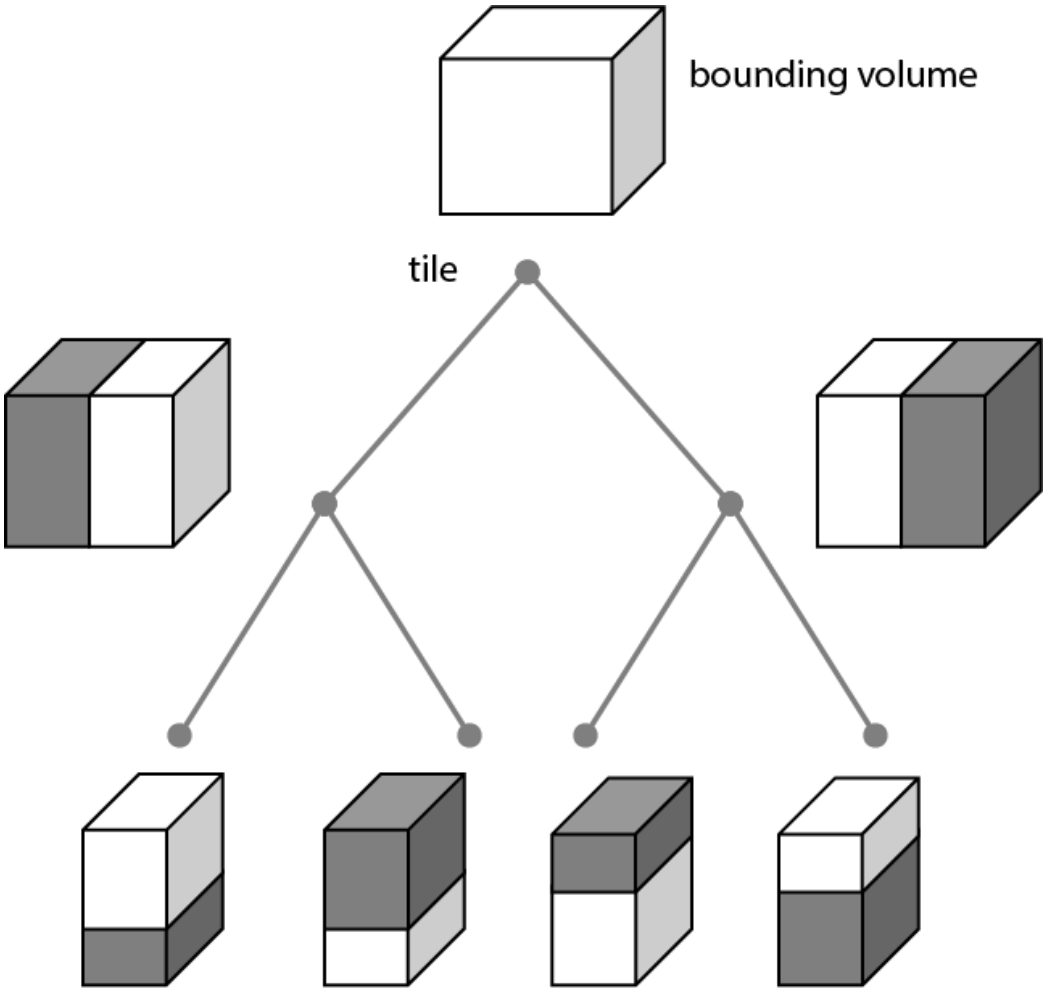
Eräs ratkaisu selainpohjaisen kaupunkimallin esittämiseen on AGI:in (Analytical Graphics, Inc.) kehittämä Cesium. Cesium on avoin JavaScript-kirjasto, jolla voi esittää kolmiulotteista dataa esimerkiksi maapallon pinnalla suoraan internet selaimessa. (Cesium: about 2018.) (Kuva 15.) Vastaavanlaista selainpohjaista kaupunkimallia tuotetaan muun muassa Helsingin kaupungissa, jota voi tarkastella kartta.hel.fi-palvelussa internet selaimella. (Hårdh 2018.)

Kuva 15. Kuvakaappaus testivaiheessa olevasta selainpohjaisesta Cesium-esityksestä osasta Vantaan kaupunkimallia. Lähellä olevat rakennukset esiintyvät korkeimmalla tasolla, teksturoituna LOD2-tason mallina. Kauempana olevat kohteet ilman teksturointia punakattoisena LOD2-tasoisena. Aivan kauimmaiset kohtaat ovat harmaita LOD1-tason laatikkomalleja.

Cesium toimii niin sanotulla tiilitys (englanniksi "tileset") -tekniikalla (kuva 16). Haluttu data jaetaan ”tiileihin” eli datapalasiin, jotka tuodaan selaimessa näkyviin käyttäjälle perinteisen kartan lähestymiskuvien tavoin. Mittakaavan ollessa pienempi, eli kun ollaan niin sanotusti kauempana kohteista, näytetään vähemmän kohteita, ja kohteiden tarkkuus on pienempi. Lähestyttäessä kohteiden määrä sekä kohteiden tarkkuus kasvavat. Kyseiset määrittelyt määritellään erillisessä JSON-muotoisessa tileset-tiedostossa jokaiselle tiilelle erikseen geometrisenä virheenä, jonka arvo on positiivinen ja määritellään metreissä. Virheen ollessa käyttäjän päätelaitteen näytöllä suurempi kuin määritelty arvo kyseistä tiiltä ei näytetä. Jokaisella tiilellä on rajaavan tilavuuden määrittelyt boundingVolume-ominaisuudella, jolla määritellään tiilien sijainti ja koko. Lisäksi tiilillä voi olla ”lapsi”-tiiliä, jolloin tiilitykselle voidaan määritellä esimerkiksi tarkkuustasoista hierarkiaa. Selaimella näytettävä tileset.json-tiedosto määritellään Cesiumin JavaScript-tiedostossa.

Tileset.json-tiedostoissa on kansiorakenneviittaukset kohteiden kolmiulotteisille geometrioille. Tiilitykset ja sen sisältämät geometriat voidaan jakaa esimerkiksi karttalehdittäin omiksi tileset.json-tiedostoiksi ja näille tileset.json-tiedostoille voidaan luoma oma,

esimerkiksi koko kaupungin käsittävä ”emo” tileset.json, jossa on geometriaviittausten sijaan viittaukset karttalehdittäin luotuihin tileset-tiedostoihin. (Anderson ym. 2017.) Tiilitys ratkaisu vähentää suorituskyvyllisiä vaatimuksia, sillä vain tarpeellinen data ladataan näytettäväksi. Testivaiheessa Vantaan kaupunkimallilla on kolme hierarkiatasoa, jotka ovat alkaen tarkimmasta LOD2-tason teksturoitu malli, LOD2-tason malli ja viimeinen eli LOD1-tason malli. (Kuva 15.) Tulevaisuudessa tasoja tullaan lisäämään esimerkiksi tekstuurien osalta, siten että tekstuureillakin esiintyy tasoja korkealaatuisesta huonompilaatuiseen teksturointiin. Kaupunkimalli säilyy tiilityksen ansioista visuaalisesti näyttävänä vaatimatta liikaa suorituskykyä.



Kuva 16. Havainnekuva tileset-rakenteesta. Tummennetut palaset kuvaavat tiilen sisällä olevien sisältöjen tilarajauksia. (Anderson ym. 2017.)

6.1.2 MATTI ja kaupunkimalli uusissa ohjelmistoissa

Yhtenä tulevaisuuden suurista muutoksista Vantaan kaupungin maankäytön toimialan yksiköissä on uusi maankäytön toimintamalli ja tietojärjestelmä, eli lyhennettynä MATTI. Sen tavoitteina on luoda uusi kokonaisvaltainen ratkaisu, joka sisältää maankäytön suunnittelun, rekisteröinnin sekä karttatuotannon. Osana MATTI-hanketta, Vantaalla aletaan käyttää ArcGIS-tuoteperheen ohjelmistoja, joihin myös kaupunkimallin on sitä myötä sovelluttava. (MATTI – Maankäytön toimintamalli ja tietojärjestelmä 2016: 4, 9.)

Koska kaupunkimalli siirtyy CityGML-muotoon ja tietokantaan, on julkaisussa otettava huomioon yhteensopivuudet näiden formaattien ja tietokannan sekä ArcGIS-ohjelmistojen välillä. Indexed 3D Scene Layers, lyhennettynä i3s, on Esrin kehittelemä tiedostoformaatti, joka on tarkoitettu massiivisten maantieteellisten, kolmiulotteisten datojen esittämiseen. OGC on hiljattain hyväksynyt i3s:n standardiksi, ja se on julkaistu avoimena. I3s:llä voidaan esittää erilaista dataa, muun muassa kolmiulotteisia objekteja, pistepilvidataa, yhdistettyjä verkkomalleja sekä yksittäisiä pistemäisiä kohteita. I3s toimii laajalti ArcGis-sovelluksissa. (I3s-spec. 2017.)

CityGML muuntautuu i3s:ksi ArcGISiin tehdyn laajennussovellus Data Interoperability -laajennuksen kautta. Data Interoperability laajennus on ArcGIS:ille suunniteltu yhdistetty työkalusarja, joka nojaa FME-sovelluksen teknologiaan ja jolla muunnosprosessi voidaan automatisoida ArcGIS-sovelluksissa. FME-sovellukseen i3s-formaatti on tulossa käyttöön vuoden 2018 aikana ja julkaistaan FME:n vuoden 2018 beetaversiossa. (Dahmen & Satish 2017: 12–23; What is the Data Interoperability extension? 2017.)

6.2 Julkaisu osana jatkuvan ylläpidon prosessia

Jatkuvan ylläpidon seurauksena myös julkaisun olisi suotava olla jatkuvasti päivittyvää sekä ajantasaista. Huomioon tulee kuitenkin ottaa päivitysten tarve julkaisussa. Esimerkiksi vuonna 2016 Vantaalla valmistui 616 rakennusta, eli noin 12 rakennusta viikossa tai 1–2 rakennusta päivässä (taulukko 1). Oletettavaa on, että kolmiulotteisia rakennusmalleja tuotetaan suurin piirtein samaan tahtiin ylläpitoketjun ollessa jatkuvaa valmistuvien rakennusten osalta. Kaupunkimalli siis päivittyy miltei päivittäin, jolloin julkaistavan mallin päivitystarve on myös tiheää. Kaupunkimalli jaetaan Cesiumin tileset-rakenteessa 250*250 metrin paloihin. Kun kaupunkimallista luodaan tileset.json-tiedostot Cesium-esitystä varten, niille annetaan luomispäivämäärä ominaisuudeksi. Uusien rakennusmallien

valmistuessa voidaan tietokantakyselyllä tunnistaa ne 250*250 metrin alueet, joihin uusi rakennusmalli on lisätty, kun kyselyyn laitetaan ehto, joka tunnistaa päivitetyt alueet päivämäärän perusteella. Jos kyseiselle alueelle on lisätty malli viimeksi luodun tileset.json-tiedoston luomispäivämäärän jälkeen tai jos alueelle on lisätty rakennusmalli viimeisen alueelle tehdyn päivityksen jälkeen, se erotellaan päivitettäväksi.

Taulukko 1. Vuonna 2016 valmistuneet rakennukset käyttötarkoituksen mukaan (Rakennus- ja asuntotuotanto vuonna 2016. 2017)

Alue	VALMISTUNEET RAKENNUSHANKKEET YHTEENSÄ				
Talotyyppi/	rak.	tilavuus	kerrosala	asuntojen	asuin
Käyttötarkoitus	lkm	m3	m2	lkm	h-ala m2
Koko Vantaa					
Kaikki rakennukset	616	2272918	420147	2955	167903
Asuinrakennukset	384	868951	253587	2954	167815
Omakotitalot	223	119595	29826	181	24247
Muut erilliset pientalot	59	42702	10810	108	9297
Rivi- ja ketjutalot	28	21306	5616	78	5055
Asuinkerrostalot	74	685348	207335	2587	129217
Vapaa-ajan asuinrakennukset	-	-	-	-	-
Muut kuin asuinrakennukset	232	1403967	166560	1	88
Liikerakennukset	9	199566	34479	-	-
Toimistorakennukset	3	6776	784	-	-
Liikenteen rakennukset	25	251160	26742	-	-
Hoitoalan rakennukset	13	34027	8619	-	-
Kokoontumisrakenukset	3	144420	13319	-	-
Opetusrakennukset	4	45677	9159	-	-
Teollisuusrakennukset	15	30979	6175	1	88
Varastorakennukset	16	662854	60813	-	-
Palo- ja pelastus rakennukset	-	-	-	-	-
Maatalousrakennukset	2	5610	1243	-	-
Muut rakennukset	142	22898	5227	-	-

Päivitettäväksi tunnistetut alueet syötetään FME-työtilaan, joka luo CityGML-tiedostosta tileset.json-tiedoston. Tämän jälkeen luodut, päivitetyt tileset-tiedostot päivitetään osaksi koko kaupungin käsittävää emotiedostoa. Päivitetty koko kaupungin käsittävä emo tileset-tiedosto päivitetään lopuksi palvelimeen, jossa Cesium-esitys sijaitsee. Päivitys prosessia varten tehtävä kysely ja loput prosessista asetetaan tapahtumaan automaattisesti esimerkiksi päivittäin, jolloin kaupunkimalli päivittyy selaimeen joka päivä.

7 Käytännön prosessit

Tässä kappaleessa käydään läpi tutkielman aikana tehtyjen empiiristen kokeilujen yhteydessä törmättyihin ongelmiin ja niiden ratkaisuihin sekä selostetaan käytännön prosessit yleisesti. Kokeilujen pohjalta koitettiin löytää optimaaliset keinot konvertoida teksturoitua vektorimuotoista aineistoa CityGML-muotoon, lisätä ominaisuustietoja kiinteistörekisteristä rakennuksille sekä kuinka konvertoitu aineisto siirtyy ja poistuu tietokannasta. Lisäksi Cesium-pohjaiseen julkaisuun liittyy muita tiedostomuotoja, sekä esimerkiksi maanpinnan mallin lisäys, jotka aiheuttivat ongelmia. Ongelmien ratkaisemiseksi käytettiin etenkin FME-sovellusta.

7.1 CityGML-muunnos

Jo tuotettu ja korjailtu LOD2-tason kaupunkimalli muutetaan CityGML-muotoiseksi käyttämällä hyväksi TerraSolidin TerraPhoton kaupunkimallin teksturointiin tarkoitetun työkalun valmiina olevaa CityGML-uloskirjoitusominaisuutta. Samalla kaupunkimalliin saadaan fotorealistiset tekstuurit.

Kirjoitushetkellä sovellus kuitenkin kirjoittaa vanhentunutta CityGML 1.0-versiota, joka ei enää ole validia esimerkiksi 3D City Database -tietokantaa varten. Lisäksi ulos kirjoittaessa joidenkin kohteiden geometriat olivat vääränlaisia. Virheellisten kohteiden rajaavien pintojen jotkin monikulmiot eivät täyttäneet monikulmion GML-geometrian mukaisia vaatimuksia: monikulmion muodostavan aloitus- ja lopetuspisteiden koordinaatit eivät olleet samat. TerraPhotosta ulos kirjoitettu CityGML joutuu siis vielä toistaiseksi käymään läpi prosessin, jossa geometriat korjataan oikeiksi ja jossa CityGML muuttuu 2.0-versioksi.

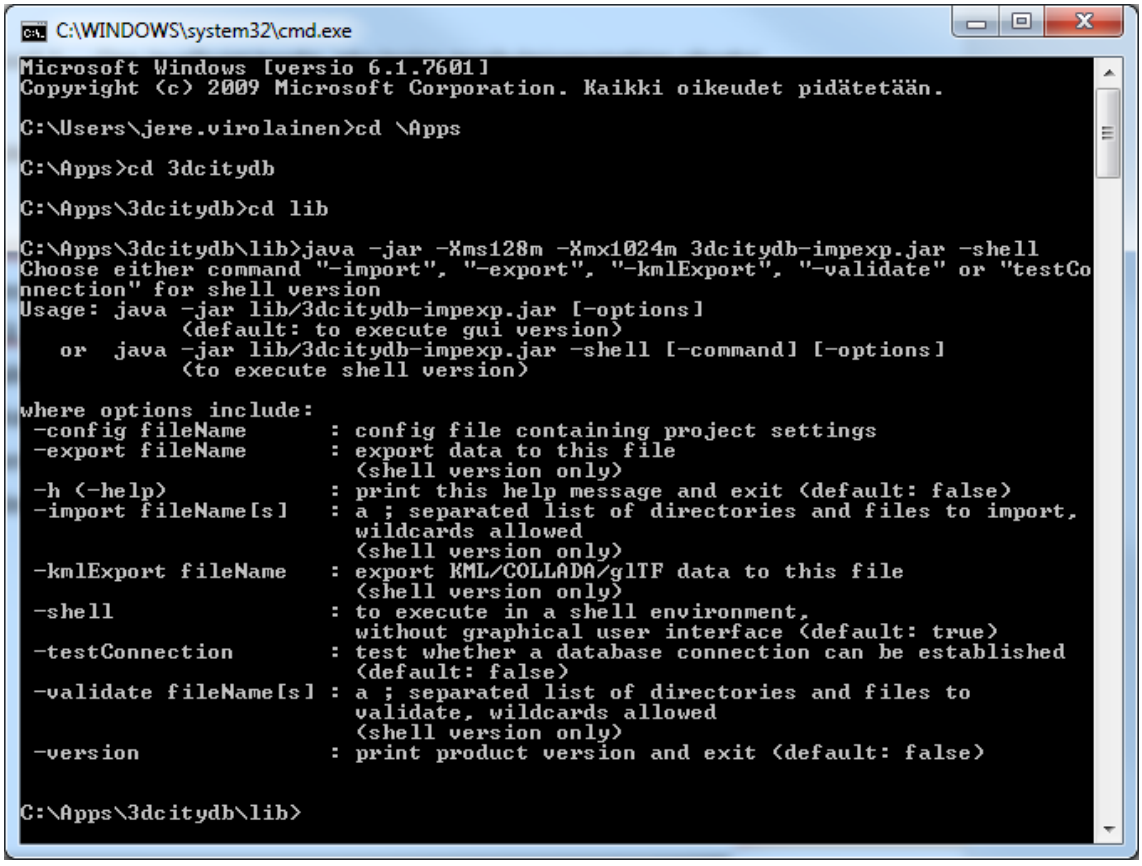
Uloskirjoitettu CityGML-tiedosto käy läpi JavaScriptillä ohjelmoidun prosessin, joka etsii virheellisen monikulmion ja korjaa alkavan ja päättyvän pisteen koordinaatit kyseisessä monikulmiossa samaksi. (Esimerkkikoodi 1.) Korjauksen jälkeen CityGML-tiedosto syötetään yksinkertaiseen FME-työtilaan, jossa syötetty tiedosto pelkästään kirjoitetaan ulos CityGML-muotoisena, mutta uudempana 2.0-versiona. FME-sovellus korjaa vanhentuneen CityGML 1.0 -muodon CityGML 2.0:ksi automaattisesti.

```
function fixPolygonEndpoints(cityobject){
     var building_id = cityobject.Building[0].$['gml:id']
     traverse(cityobject).forEach(function(key, val){
          if (this.key==='gml:posList') {
               var geomcrdstring = this.node[0]._;
               var crds = geomcrdstring.split('\r\n');
               // console.log(geomcrdstring.split('\r\n'))
               if (crds[1] != crds[crds.length-2]) {
                    console.log(building_id + ': trying to fix invalid poly-
gon.')
                    crds[1] = crds[crds.length-2];
                    this.node[0]._ = crds.join('\r\n');
               }
          };
     });
     return cityobject;
}
```

Esimerkkikoodi 1. Osa JavaScript-koodia, joka korjaa monikulmiogeometrian oikeaksi.

7.2 Tietokantaan vieminen ja uloskirjoitus

3D City Databasen rinnalle on kehitetty graafinen Importer/Exporter-työkalu, jolla dataa voidaan viedä ja tuoda ulos tietokannasta. Työkalulla myös otetaan yhteys esimerkiksi PostgreSQL-tietokantaan sekä hallitaan ulos- ja sisäänkirjoitusten asetuksia. Asetukset sekä määritykset yhteyden luontiin voidaan tallentaa työkalun avulla XML-muotoiseksi tiedostoksi. Sen lisäksi että 3DCityDB-tietokantaa voidaan hallita Importer/Exporter-työkalun avulla, sitä voidaan operoida myös komentorivien ja komentotulkin avulla. (Kuva 17.) Tämä puolestaan mahdollistaa automaattisen yhteyden luomisen, datan haun sekä sisäänkirjoituksen, sillä komentorivit voidaan syöttää osaksi automaattisia prosesseja, jolloin sisään- ja uloskirjoitus, yhteyden luominen ja oikeiden asetusten hakeminen, tapahtuvat automaattisesti ilman graafisen Importer/Exporter-työkalun avaamista ja käyttämistä. Prosessien automatisointi on tärkeässä osassa tehokasta tietokannan käyttöä ja sitä myötä myös tehokasta ylläpitoa.



Kuva 17. 3D City Databasen operointi komentorivein Windowsin Command Prompt-komentotulkissa.

Tutkimalla 3D City Databasen latauksen mukana tullutta ohjekirjaa löydettiin tarvittavat komennot ja kokeilemalla oikea komentojen suoritusjärjestys, jolla yhteys luotiin ja dataa voitiin tallettaa ja kirjoittaa ulos tietokannasta. Lisäksi graafisessa työkalussa muokattiin ensin asetukset ja yhteyden luonti halutuiksi sekä tallennettiin ne omaksi XML -tiedostoksi. Valmis, oikean versioinen ja validi teksturoitu CityGML-muotoinen kaupunkimalli vietiin tietokantaan. Tietokannasta uloskirjoitusta varten luotiin FME-työtila, jolla voidaan hakea rakennukset tietokannasta halutun aluerajauksen mukaan, jossa käytetään muun muassa hyväksi tallennettua tietokanta-asetusten XML-tiedostoa. Haluttu aluerajaus syötetään FME-työtilassa, ja FME-työtila vie kyseisen syötetyn rajauksen 3DCityDB:n XML-asetustiedoston siihen kohtaan, jossa aluerajaus määritellään. Jos aluerajaus on monimutkaisen muotoinen ja koska 3DCityDB ymmärtää vain suorakaiteen muotoisia aluerajauksia, uloskirjoitettuja kohteita verrataan vielä aluerajaukseen, jotta mahdolliset rajauksen ulkopuoliset kohteet voidaan poistaa uloskirjoituksessa. Tällainen menetelmä on tarpeellinen, jos halutaan tulostaa esimerkiksi tietyn kaupunginosan rakennukset, jolloin aluerajaus usein on suorakaidetta monimutkaisempi.

7.3 Cesium-julkaisuun liittyvät ongelmat

Internetjulkaisu Cesium ympäristössä osoittautui koko projektin hankalimmaksi ja eniten tutkimusta ja työtä vaativaksi osioksi. Ongelmia ilmeni muun muassa maanpinnan mallin lisäämisessä, sekä oikeanlaisen tileset-tiedoston kirjoituksessa kolmiulotteisten kohteiden esittämistä varten.

Cesium käyttää oletuksena esityksien maanpintana tasaista WGS84-ellipsoidin mukaista pintaa. Vantaan kaupungin kohteiden korkeuskoordinaatit ovat valtakunnallisessa N2000-järjestelmässä. Vertausellipsoidin sekä korkeusjärjestelmän välillä on eroja ja tästä syystä kohteet ”leijuivat” esityksessä ilmassa eikä maanpinnan korkeuseroja ollut havaittavissa. Cesiumiin on saatavilla ulkopuolisten luomia, koko maapallon kattavia maanpintamalleja, jotka kuitenkaan eivät esityksessä sointuneet yhteen Vantaan rakennusten kanssa korkeusjärjestelmäerojen takia (Terrain 2017). Rakennukset ”upposivat” maanpintamallin sisälle tai jäivät edelleen leijumaan maanpinnan yläpuolelle. Lisäksi Vantaa haluaa oman maanpintamallin osaksi kaupunkimallia ja päivitystä. Oman pintamallin luominen Cesiumiin on mahdollista cesium-terrain-builder -nimisen C++ -kirjaston kautta. Cesium-terrain-builder luo korkeusmallista, esimerkiksi pistepilvestä luodusta kolmioverkkomallista, sekä ilmakuvarastereista tiilitys-rakenteen omaavan maanpintamallin, joka soveltuu esitettäväksi Cesium-ympäristössä. Se on kuitenkin luotu ensisijaisesti Linux OS -käyttöjärjestelmälle ja vaatii toimiakseen Windows-käyttöjärjestelmissä Docker-virtuaalitietokoneen rinnalleen. (Zwaagsta 2018.) Cesium-terrain-builderin ja Dockerin varsinaista käyttöä ja toteutustapaa ei tässä insinöörityössä eritellä enempää työn aiherajauksen vuoksi.

FME-sovellus mahdollistaa tileset.json tiedoston kirjoittamisen rakennuksille. Sovelluksen avulla voidaan kirjoittaa tileset.json-tiedostoa, jonka lähtötietona on CityGML-muotoinen malli, ja se luo samalla myös tarvittavat Cesiumin tukemat b3dm-tiedostoformaatit malleille. Sovelluksen kirjoittama tileset.json-tiedosto ei kuitenkaan sellaisenaan omaa haluttua tarkkuustasoihin perustuvaa hierarkiarakennetta. Haluttu hierarkiarakenne kokeiltiin ensiksi rakentaa JavaScript-koodilla, joka asettaisi FME:n kirjoittamat tileset.json:it haluttuun rakenteeseen, kunnes huomattiin, että FME:ssä on myös mahdollista muokata rakenne halutun kaltaiseksi tiettyjen FME-muuntajien avulla helpommin ja paremmin kuin JavaScript-koodilla. FME-työtilaan lisättiin tarvittavat muuntajat, ja oikeanlaisen tileset.json tiedoston kirjoitus onnistui, mikä mahdollistaa rakennusten esittämisen selaimessa tarkkuustasoihin perustuvassa hierarkiarakenteessa.

8 Yhteenveto

Työn tavoitteena oli selvittää kokonaisvaltainen prosessi Vantaan kolmiulotteisen kaupunkimallin ylläpidolle mallin tuotannosta sen julkaisuun asti tietokantapohjaisen ratkaisun kautta. Etenkin vastauksia haluttiin jo aiemmin tuotetun mallin konvertoinnista CityGML-muotoon, tietokannan toimivuudesta sekä kuinka mallia voidaan julkaista. Työn aikana kokeiltiin mallin konvertointia, tietokannan käyttöä sekä mallin julkaisua käytännössä, joiden pohjalta löydettiin vastauksia kysymyksiin.

Työssä havaittiin muun muassa, että jo nyt tuotetun kaupunkimallin muuntaminen oikeanlaiseksi CityGML-muotoiseksi malliksi vaatii kehitystä. Vaikka tuotettu malli onnistuttiin muuntamaan oikeaan muotoon, on siinä tällä hetkellä suhteellisen monta väliprosessia esimerkiksi mallin geometrian oikeellisuuden ja CityGML:n oikean version varmistamiseksi. Itse mallin tuottaminen nykyisillä sovelluksilla on hyvällä tolalla, mutta sovellusten ja tietokannan, sekä CityGML-konvertoinnin yhteentoimivuutta on kehitettävä. CityGML-muotoon muuntaminen, sekä mallin tietokantaan viemisen ja tuomisen olisi suotava toimia mallin luomiseen tarkoitettujen sovellusten yhteydessä, ilman että konvertointia ja tietokantaan tallettamista tarvitsisi tehdä enää erikseen muiden sovellusten suomien ratkaisujen avulla. Nämä puutteet ja parannukset ovatkin ensisijaisia kehityskohteita yhdessä Vantaan kaupunkimittausosaston ja sen yhteistyökumppanien parissa tulevaisuudessa tietokantapohjaisen ylläpidon käyttöönotossa.

3DCityDB-tietokanta osoittautui työn aikana toimivaksi ja hyväksi, tehokkaaksi ratkaisuksi CityGML-muotoisen kaupunkimallin hallintaa ja ylläpitoa varten eikä varsinaisia ongelmia sen käytössä esiintynyt. Myös Cesium-pohjainen internetselaimeen nojautuva julkaisualusta osoittautui lupaavaksi vaihtoehdoksi kaupunkimallin julkaisua varten, joskin sen täysimittainen käyttöönotto on edelleen työn kirjoitushetkellä vielä testivaiheessa. Työn aikana tehtyjen havaintojen ja kokeilujen pohjalta on kuitenkin saavutettu tarvittava lähtötieto julkaisualustan jatkokehitykselle, joka mahdollistaa kaupunkimallin selainpohjaisen julkaisun vuoden 2018 aikana. CityGML-tiedostomuoto itsessään havaittiin hyväksi lähtötietoformaatiksi kaupunkimallille, joka mahdollistaa semanttisesti rikkaan kaupunkimallin käytön ja joka muuntautuu helposti muiksi tarvittaviksi tiedostoformaateiksi.

Vuoden 2016 laserkeilauksen pohjalta luodun kaupunkimallin manuaalinen korjailu on edelleen kesken. Korjailu saatetaan loppuun nykyisellä menetelmällä ja työssä kuvailtu uusi tietokantapohjainen ylläpitomenetelmä otetaan käyttöön myöhemmin. Työssä esitelty IFC-mallien lisääminen kaupunkimalliin eräänä ylläpidon osana on keskustelun aiheena kirjoitushetkellä Vantaalla ja se tullaan mitä todennäköisemmin toteuttamaan. Siihen liittyen jatkokehityksen kohteena edelleen on ratkaista IFC-mallien CityGML-muotoon muuttaminen.

Lähteet

3D City Database for CityGML Version 3.3.0. 2016. Asiakirja. Saatavilla: < https://www.3dcitydb.org/3dcitydb/documentation/>

Ahokas, Niila. 2014. 3D-Kaupunkimallin tuottaminen ja ylläpito. Opinnäytetyö. Lapin ammattikorkeakoulu. Theseus-tietokanta.

Anderson, Erik; Brown, Dylan; Chow, Sarah; Fini, Leesa; Ring, Kevin. 2017. 3D-tiles. Verkkoaineisto. <https://github.com/AnalyticalGraphicsInc/3d-tiles>. 23.02.2017. Luettu 4.4.2018.

Biljecki, Filip. 2017. Level of detail in 3D city models. Väitöskirja. E–kirja. Delft University of Technology.

Billen, Ronald; Caglioni, Matteo; Cutting-Decelle, Anne-Francoise; de Almeida, José-Paulo; Falquet, Gilles; Leduc, Thomas; Marina, Ognen; Métral, Claudine; Moreau, Guillaume; Perret, Julien; Rabino, Giovanni; San Jose, Roberto; Yatskiv, Irina; Zlatanova, Sisi. 2014. 3D City Models and urban information: Current issues and perspectives. Les Ulis, Ranska: EDP Sciences. 17.2.2014.

Cesium: about. Verkkoaineisto. <https://cesiumjs.org/about/>. Luettu 30.01.2018.

Dahmen, Christian & Sankaran Satish. 2017. Data Conversion to I3S for 3D Modeling from CityGML. Esityskalvot. Verkkoaineisto. <http://proceedings.esri.com/library/userconf/proc17/tech-workshops/tw_2582-457.pdf>. 16.08.2017. Luettu 6.4.2018.

Döllner, Jürgen; Falko, Liecke; Kolbe, Thomas; Sqouros Takis; Teichmann Karin 2006. The virtual 3D city model of Berlin - Managing, integrating, and communicating complex urban information. Proceedings of the 25th Urban Data Management Symposium UDMS.

Donkers, Sjors. 2013. Automatic generation of CityGML LoD3 building models from IFC models. Diplomityö. Delft University of Technology. Alankomaat. TUDelft Repository-tietokanta.

Erving, Anna. 2007. Julkisivutekstuurin liittäminen 3D-malliin. Diplomityö. Teknillinen korkeakoulu. Saatavilla: <https://foto.aalto.fi/publications/diplomityot.html>

Erving, Anna. 2008. Paikkatiedoista kaupunkimalleihin: CityGML selvitystyö. Fotogrammetrian ja kaukokartoituksen laboratorio. Teknillinen korkeakoulu. Saatavilla: <https://foto.aalto.fi/publications/publications.php>

How FME works? 2018. Verkkoaineisto. <https://www.safe.com/how-it-works/>. 30.3.2018. Luettu 16.4.2018.

Hårdh, Jarkko. 2018. Paikkatietoasiantuntija, Helsingin kaupunki. Sähköpostikeskustelu. 6.4.2018.

I3s-spec. 2017. Scene Layers: Service and Package Standard. Verkkoaineisto. <https://github.com/Esri/i3s-spec>. 28.7.2017. Luettu 6.4.2018.

Jokela, Joonas. 2016. CityGML building model production from airborne laser scanning. Diplomityö. Aalto-yliopisto. Aaltodoc-tietokanta.

Kalso, Markus 2018. Paikkatietoinsinööri, Vantaan kaupunki. Haastattelu 16.2.2018.

Kolbe, Thomas H. 2009a. CityGML–OGC standard for photogrammetry? 10.9.2009. Photogrammetric Week 2009, Stuttgart. Saatavilla: <http://www.ifp.uni-stuttgart.de/publications/phowo09/270Kolbe.pdf>

Kolbe, Thomas H. 2009b. Representing and Exchanging 3D City Models with CityGML. 3D Geo-Information Sciences. Springer-Verlag Berlin Heidelberg, Saksa.

Kolbe, Thomas H. 2016. 3D City Database for CityGML; A Hands-on Tutorial for Beginners Version 3.3.0. Chair of Geoinformatics. Technical University of Munich. Verkkoaineisto. <https://www.3dcitydb.org/3dcitydb/documentation/>

Kärkkäinen, Risto. 2016. Locus-kantakartasta SketchUp-3D-kaupunkimalli, 3D-mallinnus ja visualisointi kunnan suunnittelun tukena. Opinnäytetyö. Metropolia Ammattikorkeakoulu. Theseus-tietokanta.

Lammi, Hannu. 2015. Kaupunkimallit ja CityGML. Esityskalvo SFS-seminaarissa. 14.4.2015. Saatavilla: <https://www.sfs.fi/files/7740/Lammi_Kaupunkimallit_ja_CityGML.pdf>

Liukkonen, Oskari. 2015. Kuntien paikkatiedon polku kantakartasta 3D-kaupunkimalliin. Diplomityö. Aalto-yliopisto. Aaltodoc-tietokanta.

MATTI – Maankäytön toimintamalli ja tietojärjestelmä. 2016. Vantaan kaupunki. Esityskalvot. 12.10.2016.

Microstation. 2017. Verkkoaineisto. <https://www.bentley.com/en/products/brands/microstation> 04.12.2017. Luettu 16.4.2018.

Mitä on BIM? Verkkoaineisto. Tekla Oyj <https://www.tekla.com/fi/tietoa-meist%C3%A4/mit%C3%A4-bim> Luettu 26.02.2018.

OGC City Geography Markup Language (CityGML) Encoding Standard. Version 2.0.0. 2012. Verkkoaineisto. 04.04.2012. Saatavilla: <http://www.opengeospatial.org/standards/citygml#downloads>

Rakennus- ja asuntotuotanto vuonna 2016. 2017. Excel -taulukko. <https://www.vantaa.fi/hallinto_ja_talous/tietoa_vantaasta/tilastot_ja_tutkimukset/tilasto-_ja_tutkimusjulkaisut>.

Richter, Rico 2015. 3D Point Cloud Analytics for Updating 3D City Models. Seminaarikalvot. 25.3.2015. Geospatial World Forum 2015. <https://geospatialworldforum.org/speaker/SpeakersImages/Rico%20Richter.pdf>

Terrain. 2017. Verkkoaineisto. Cesium. <https://cesiumjs.org/tutorials/Terrain-Tutorial/> 1.10.2017. Luettu 12.4.2018.

Terrasolid products. 2018. Verkkoaineisto. Terrasolid. <http://www.terrasolid.com/products.php> 08.01.2018. Luettu 16.04.2018.

Soininen, Arttu. 2016. TerraScan User's Guide. <http://www.terrasolid.com/guides/user_guides.php>

Uuden sukupolven kaupunkimallit Helsinkiin. 2016. Verkkoaineisto. Helsingin kaupunki. <https://www.hel.fi/static/kanslia/Helsinki3D/Uuden-sukupolven-kaupunkimallit-Helsinkiin.pdf>. Luettu 3.1.2018.

What is the Data Interoperability extension? 2017. Verkkoaineisto. ArcGIS. <http://desktop.arcgis.com/en/arcmap/latest/extensions/data-interoperability/what-is-the-data-interoperability-extension-.htm>. 4.12.2017. Luettu 6.4.2018.

Xun Xu, Lieyun Ding, Hanbin Luo, Ling Ma. 2014. From building information modeling to city information modeling, Journal of Information Technology in Construction (ITcon), Special Issue BIM Cloud-Based Technology in the AEC Sector: Present Status and Future Trends, Vol. 19. Sivut: 292–307. <http://www.itcon.org/2014/17>

Yli-Tainio, Antti & Savisalo Anssi. 2015a. Opas 3D-esineiden mallintamiseen. Osa 1: Perusteet. (Säännöt GML-geometrioiden validointiin CityGML:ssä). Special Interest Group SIG3D, German Spatial Data Infrastructure (GDI-DE). Verkkoaineisto. <https://buildingsmart.fi/wp-content/uploads/2016/11/>

Yli-Tainio, Antti & Savisalo Anssi. 2015b. Opas 3D–esineiden mallintamiseen. Osa 2: Rakennusten mallintaminen (LoD1, LoD2 ja LoD3). Special Interest Group SIG3D, German Spatial Data Infrastructure (GDI-DE). Verkkoaineisto. <https://buildingsmart.fi/wp-content/uploads/2016/11/>

Zwaagsta, Homme. 2018. Cesium Terrain Builder. Verkkoaineisto. <https://github.com/geo-data/cesium-terrain-builder> 23.3.2018. Luettu 12.4.2018.