

Examensarbete i geografisk informationsteknik nr 16

Samredovisning av BIM- och GIS-data

Anna Larsson

Civilingenjörsutbildningen i Lantmäteri

Lunds Tekniska Högskola

Institutionen för Naturgeografi och Ekosystemvetenskap

Lunds Universitet





LUNDS UNIVERSITET
Lunds Tekniska Högskola

Samredovisning av BIM- och GIS-data

EXTM05 Masteruppsats, 30hp
Civilingenjörsutbildningen i Lantmäteri

Anna Larsson

Handledare:

Lars Harrie

Institutionen för Naturgeografi och Ekosystemvetenskap

28 Augusti 2015

Opponent: Anton Lundkvist

Examinator: Micael Runnström

Copyright © Anna Larsson, LTH

Institutionen för Naturgeografi och Ekosystemvetenskaper

Lunds Universitet

Sölvegatan 12

223 62 Lund

Telefon: 046-222 30 30

Fax: 046-222 03 21

Hemsida: <http://www.nateko.lu.se>

Examensarbete i geografisk informationsteknik nr 16

Tryckt av E-tryck, E-huset, 2015

Abstract

Throughout the process of creating and maintaining our built environment, from planning to building to facility management, large amounts of data are produced and consumed. Much of these are geographic data, processed in some form of GIS (Geographical Information System). Others are building data, increasingly handled in different BIM (Building Information Model) software packages. To be able to integrate data from these two different worlds can be seen as a crucial step in improving communication, avoiding resource waste and creating new possibilities.

This master's thesis contains an investigation on ways of combining data from BIM and GIS sources, and displaying the combined models online. The study is conducted in two parts, the first using literature, interviews and technical documentation to provide a theoretical background, and the second using a case study to investigate the technical feasibility of the techniques described.

The background contains sections on possible application domains and an overview of representations of 3D geometry. It further details some of the most important 3D data formats currently in use, known issues with integration of BIM and GIS data, and options for visualizing 3D data online. The study goes on to describe possible system architectures for combining BIM and GIS data, and to define criteria against which these can be evaluated.

The case study uses data from the ongoing reconstruction of a Swedish pulp mill. A 3D model of the mill and a digital elevation model from the national land survey (Lantmäteriet) are combined. Four different system architectures are investigated, one using an online GIS platform (ArcGIS Online and the CityEngine Web Viewer), one using a BIM distribution tool (Maint3D), one where a viewer is developed using a JavaScript library (three.js), and one where the data is transformed to 2D (using WMS and ArcGIS Server).

Results show that there are still several issues that need to be resolved to achieve combined 3D visualizations. There are problems on a technical level, with geometric representations and format translations; on a data level, with differing information models and content handling; and on an organizational level, where data and data producers still seem divided into separate worlds. The study concludes that co-operation and standardization are necessary, as well as continued technical development, to make combinations of 3D data plausible in the future.

Sammanfattning

Under vägen från plan till bygge till förvaltning omsätter samhällsbyggnadsprocessen idag stora mängder data. Mycket av dessa data är geografiska data, som hanteras i någon form av GIS (geografiskt informationssystem). Andra är byggnads- och anläggningsdata, som i allt större utsträckning hanteras i olika BIM-program (byggnadsinformationsmodell). Att kunna integrera data från dessa två världar ses idag som ett väsentligt steg för att förbättra kommunikation mellan aktörer, uppnå nya nyttor, och minska resursslöseriet i processen.

Detta examensarbete i geografisk informationsteknik innehåller en undersökning kring sätt att samredovisa BIM- och GIS-data, och visa upp de kombinerade modellerna online. Arbetet utförs i två delar, där den första använder litteraturstudier, intervjuer och teknisk dokumentation för att skapa en teoretisk bakgrund, och den andra genom en fallstudie praktiskt undersöker de tekniska möjligheterna.

Bakgrundsdelen innehåller avsnitt om möjliga tillämpningar av samredovisningstekniker, och en översikt av geometriska representationer i tre dimensioner. Den beskriver de viktigaste dataformaten som används inom tredimensionella geodata respektive bygg- och anläggningsdata, samt vilka data som idag produceras i Sverige och av vem. Litteraturen kring kända integrationsproblem och föreslagna lösningar på dessa diskuteras, och möjliga lösningar för visualisering av 3D-data online beskrivs. Dessa olika trådar dras sen samman i en beskrivning av olika möjliga systemarkitekturer för att kombinera BIM- och GIS-data, och en diskussion kring kriterier för att utvärdera dessa arkitekturförslag.

Fallstudien använder data från den pågående ombyggnaden av ett pappermassabruk. En 3D-modell av bruket kombineras med en digital ortofotodraperad markmodell, och den kombinerade modellen redovisas på prov i fyra olika system. Dessa har valts för att representera en GIS-baserad plattform (ArcGIS Online och CityEngine Web Viewer), ett BIM-visualiseringsverktyg (Maint3D), en egenutvecklad JavaScript-baserad visualiseringsmiljö (byggd på three.js), och en 2D-miljö (som utnyttjar WMS och ArcGIS Server).

Resultaten visar att inget av de provade alternativen når riktigt ända fram, och att det fortfarande finns flera hinder i vägen för att åstadkomma bra samredovisningar i 3D. Det finns tekniska problem med överföringar mellan olika geometriska representationer och dataformat. Det finns dataproblem, med dåligt kompatibla informationsmodeller, kvalitetsbegrepp och attributhantering. Dessutom finns mänskliga och organisatoriska problem, i form av brist på kommunikation och gemensamma målbilder mellan datavärldarna. Slutsatsen blir att fortsatt standardisering och utökat samarbete krävs, utöver en fortsatt teknisk utveckling, för att utvidga möjligheterna till samredovisning av 3D-data i framtiden.

Förord

Detta examensarbete tillkom under den långa våren och kalla försommaren 2015. Det hade inte varit möjligt utan insatser från en lång rad fantastiska människor, som jag vill passa på att tacka här. Således, hjärtinnerliga tack till:

- Min akademiske handledare, Lars, som har varit ett otroligt stöd hela vägen, kommit med bra idéer, spännande artiklar och konstruktiv kritik, och som ALLTID svarar på mail.
- Min andre handledare, Fredrik Ekelund på Sweco, som kom med grundidén till det här examensarbetet, och som ställer rätt frågor och känner rätt människor och har varit entusiastisk och nyfiken och pådrivande från första början.
- Karin, Johan, Ulf, Mikael, Peter, Peter, Björn och alla andra på Sweco Positions Malmökontor, som bidragit med sin expertis, sitt stöd och ovärderligt lunchsällskap.
- Niklas, Sölve, Gil, Christoffer och Andreas från systerbolagen runtom i Sverige, som svarat på frågor och kommit med data och tagit sig tid att hjälpa till.
- Alla de anonyma genier i forumtrådar som vet vad jag gjort fel innan jag ens frågat.
- Min syster Sara, som är korrekturläsarnas drottning.
- Min far, som lärde mig att älska kartor, och min mor, som lärde mig vikten av att vara kreativ och nyfiken.
- Vänner, klasskompisar, familj, spexare och annat löst folk som gett guldkant åt våren och fått mig att tänka på annat ibland.

Utan er hade detta exjobb aldrig blivit till!

Innehållsförteckning

1. Introduktion.....	1
1.1 Bakgrund.....	1
1.2 Problemformulering	2
1.3 Syfte	2
1.4 Avgränsning	2
1.4 Metod	3
1.5 Disposition	3
2. Tillämpningar.....	4
2.1 Stadsplanering	4
2.2 Stora anläggningar.....	5
2.3 Undermarkskonstruktioner	5
2.4 Geografiskt utbredda anläggningar.....	6
2.5 3D-Fastighetsbildning	7
3. Geometrisk representation i tre dimensioner	8
3.1 Boundary representation (B-rep)	8
3.2 Voxlar	9
3.3 Constructive Solid Geometry (CSG).....	9
4. 3D-data och dataformat.....	10
4.1 Geodataformat	10
4.1.1 2,5D-data.....	11
4.1.2 CityGML.....	12
4.1.3 KML	14
4.2 Bygg- & Anläggningsdataformat.....	15
4.2.1 CAD-format	16
4.2.2 IFC.....	16
4.3 Generella 3D-dataformat	19
4.3.1 COLLADA	19
4.3.3 OBJ	20
4.4 3D-data i Sverige.....	20
5. Integrationsmetoder	22

5.1 Problematiken	22
5.2 Föreslagna lösningar	24
6. Visualisering online	27
6.1 ESRI's programsvit	28
6.2 BIM-baserade visualiseringslösningar	29
6.3 X3D/X3DOM.....	29
6.4 Three.js	30
6.5 Cesium	30
6.6 Swecos interna 3D-motor	30
6.7 2D-presentation av 3D-data	31
7. Arkitektur och informationsflöde.....	32
7.1 Kriterier.....	32
7.2 Arkitekturförslag.....	34
7.2.1 ESRI-baserad lösning	34
7.2.2 BIM-verktygslösning.....	35
7.2.3 Egen visualiseringslösning.....	37
7.2.4 2D-lösning	38
8. Fallstudie	40
8.1 Bakgrund.....	40
8.2 Data.....	40
8.2.1 Dataimport från PDMS.....	41
8.4 Kriterieanalys och upplägg	42
8.5 Alternativ 1: CityEngine Web Viewer	43
8.5.1 Implementering	43
8.5.2 Resultat	44
8.6 Alternativ 2: Maint 3D	45
8.6.1 Implementering	45
8.6.2 Resultat	46
8.7 Alternativ 3: Three.js	48
8.7.1 Implementering	48
8.7.2 Resultat	49
8.8 Alternativ 4: 2D-karta	49

8.8.1 Implementering	49
8.8.2 Resultat	50
8.9 Allmänna observationer	53
9. Diskussion.....	55
9.1 Den tekniska nivån.....	55
9.2 Datanivån.....	56
9.3 Organisations- och marknadsnivån	57
9.4 Alternativa upplägg.....	58
9.5 Framtida utveckling	58
10. Slutsatser	60
Källförteckning.....	63
Litteratur.....	63
Muntliga källor.....	67
Bilaga 1 – FME workspaces	68
Bilaga 2 - Ordlista	71

1. Introduktion

1.1 Bakgrund

Samhällsbyggnad i vid bemärkelse är en sektor som producerar och använder stora mängder information, där mycket har en spatial komponent och kan redovisas som två- eller tredimensionella digitala modeller. Det kan röra sig om kartor, stadsmodeller, översiktliga ritningar eller detaljerade 3D-modeller av enskilda byggnader. Spatiala data på alla skalnivåer spelar idag en fundamental roll i planering, genomförande och utvärdering av vitt skilda projekt inom vitt skilda sektorer (Abdul-Rahman & Pilouk, 2008).

Dessa spatiala datamängder är resurskrävande att framställa och hålla uppdaterade, och det är därför viktigt att det finns system på plats som kan optimera användningen. En grundläggande del av detta är att kunna kombinera, återanvända och samutnyttja data på genomtänkta och värdeskapande sätt. Information som redan finns ska inte behöva samlas in igen för ett nytt användningsområde, och kombinationer av data som ger upphov till nya värden och insikter ska vara lätta att göra. Tyvärr är vi i dagsläget ganska långt från denna beskrivning.

En del av svårigheten grundar sig i att data från olika sektorer och organisationer är uppbyggda och lagrade i olika system, med olika grundläggande datamodeller och olika format. Grovt kan man skilja på geodata, som lagras i olika former av GIS-format (geografiska informationssystem), och bygg- och anläggningsdata för individuella konstruktionsprojekt som allt oftare redovisas i BIM-format (byggnadsinformationsmodell). Eftersom modellerna har olika ursprung och olika grundläggande syfte finns stora skillnader i datainnehåll, detaljnivå och geometri, och det är ett långt ifrån trivialt problem att kombinera dem utan stora dataförluster.

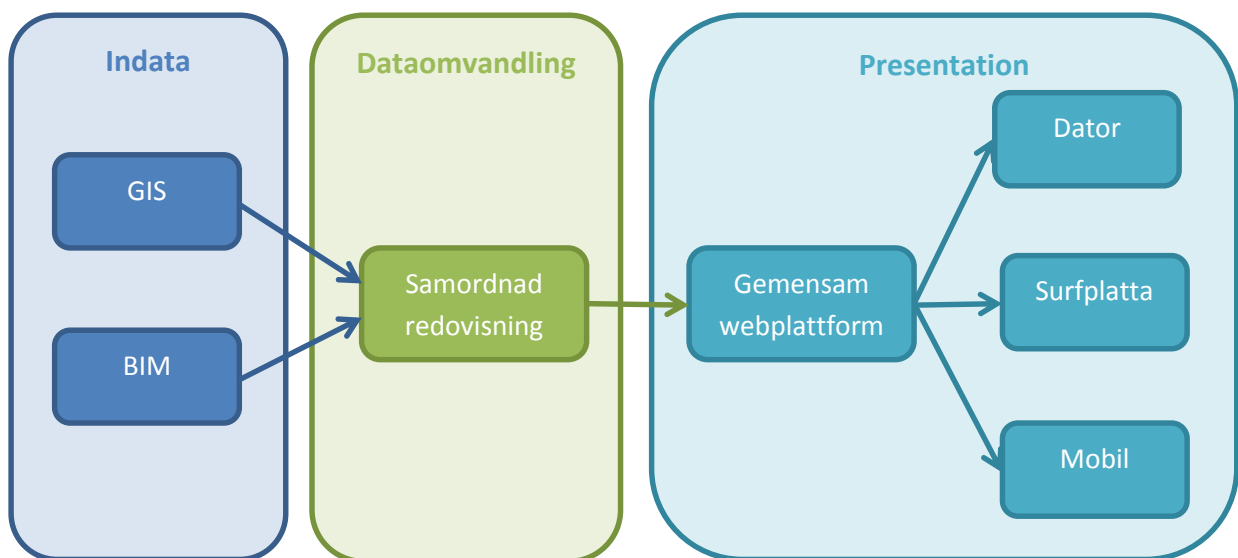
På senare år har det ägnats en hel del resurser åt att finna sätt att kombinera data från GIS- och BIM- världarna. En del av motivationen för detta härrör ur ett ökat intresse för 3D-data på GIS-sidan, där utvecklingen av bland annat detaljerade laserskannade höjddata och fotorealistiska stadsmodeller tagit fart. Byggnadsdata, som för det allra mesta redan är tredimensionella, ses där som en viktig datakälla. Dessutom ser man att byggnadsmodellerna skulle kunna berikas av tillgång till sådant som idag huvudsakligen finns redovisat i GIS-format, till exempel terrängmodeller och modeller av omgivande bebyggelse och infrastruktur.

Lantmäteriet pekar i ett nyligen redovisat regeringsuppdrag ut integration av GIS- och BIM-data som väsentligt för den framtida utvecklingen av 3D-dataanvändning i Sverige (Lantmäteriet, 2014). Det har under de senaste tio åren producerats en hel del forskning kring problemet, där olika aspekter belysts och olika lösningsförslag presenterats. Det är dock fortfarande i högsta grad en aktiv utveckling, och nya lösningar produceras kontinuerligt för olika tillämpningar.

1.2 Problemformulering

Den generella målsättningen med detta examensarbete är att ta fram en lämplig arkitektur för web-baserad samredovisning och visualisering av BIM- och GIS-data. Huvudsakligen behandlas tredimensionella data, även om vissa möjligheter kring tvådimensionella data kommer tas upp. En översiktlig bild av det tänkta flödet finns i Figur 1.1.

Samredovisning i det här sammanhanget kan antingen handla om att överföra data till ett gemensamt format, eller hitta en visualiseringsplattform som klarar av att kombinera format från olika källor direkt. Det rör sig inte om enbart samvisualisering, eftersom målsättningen är att bevara objektorientering och attributdata, men inte heller om en fullständig integration på datanivå, eftersom olika format och informationsmodeller fortfarande tillåts.



Figur 1.1 Tänkt informationsflöde för BIM- och GIS-data.

1.3 Syfte

Examensarbetet har två specifika syften.

1. Att kartlägga vilka alternativ som finns för de olika stegen i figur 1.1. Utredningen kommer täcka geometriska representationer av 3D-data, 3D-dataformat, integrationsmetoder för 3D-data, samt olika sätt att redovisa 3D-data på internet.
2. Att implementera ett eller några av de lösningsförslag som utretts. Fokus ligger på informationsflöde och användbarhet.

1.4 Avgränsning

Området är alltför stort och innehåller allt för många komplexa frågor för att rymmas helt inom ramen för ett examensarbete. Följande avgränsningar har därför gjorts:

Det utvecklade systemet ska kunna:

- Visualisera tredimensionella data från olika källor.

- Visa attributdata kopplat till geometrier.

Däremot kommer arbetet inte gå på djupet med:

- datalagringsfrågor
- uppdatering av data
- informationsmodeller och semantiska skillnader mellan datalagringsformat
- stöd för tredimensionella rumsliga analyser
- prestandaoptimering.

Även om målet är att utreda så många olika lösningar som möjligt kommer fokus ligga på de som är realistiska att genomföra med tanke på tidsramarna och tillgängliga resurser.

1.4 Metod

Målsättningen ska uppnås i två steg. Den första delen kommer att utgöras av en litteratur- och intervjustudie, som syftar till att utforska vilka olika lösningar som finns för samredovisning av GIS- och BIM-data idag och vad dessa har för styrkor och svagheter. Information ska samlas in från akademisk litteratur, intervjuer med sakkunniga, och teknisk dokumentation för de dataformat och standarder som kan bli aktuella.

Den andra delen av examensarbetet försöker verifiera slutsatserna från teoridelen genom att implementera ett par valda metoder i en specifik fallstudie. Ett antal önskvärda kriterier för det färdiga systemet sätts upp, och de olika alternativ som presenterats jämförs med dessa. Fallstudien introduceras i kapitel 8, som också innehåller mer detaljerade beskrivningar av hur de valda implementeringarna genomförts.

Arbetet utförs i samarbete med teknikonsultföretaget Sweco, och kommer i första hand att utnyttja kompetens och kunddata därifrån. De lösningar som utforskas kommer också primärt vara baserade på programvaror och tekniska plattformar som redan används inom koncernen.

1.5 Disposition

Examensarbetet är upplagt enligt följande. Kapitel 2 ger en motivation och bakgrund för examensarbetet genom att presentera ett antal användarfall och tillämpningar. Kapitel 3 beskriver den teoretiska bakgrunden för lagring av 3D-data. Kapitel 4 beskriver översiktligt GIS- och BIM-världarna, och de huvudsakliga dataformat som används för att lagra och redovisa tredimensionella data där. Kapitel 5 går mer detaljerat igenom vilka problem som uppstår när dessa data ska integreras och varför, samt presenterar en genomgång av befintlig litteratur kring hur problemet kan lösas. Kapitel 6 beskriver alternativ för visualisering av tredimensionella data online. Kapitel 7 diskuterar vilka möjliga arkitekturer och informationsflöden som är tänkbara att implementera i den praktiska delen. Kapitel 8 redogör för använda data, hur det praktiska problemet behandlats i fallstudien, samt resultaten av de olika försöken. Slutligen innehåller kapitel 9 diskussion och reflektioner och kapitel 10 sammanfattande slutsatser.

2. Tillämpningar

Det finns ett antal tänkbara tillämpningar för den här tekniken, där information från geodatavärlden och från bygg- och anläggningsvärlden tillför varandra värde. I grund och botten handlar det huvudsakligen om kommunikation, om att utbyta data mellan olika aktörer i olika delar av samhällsbyggnadsprocessen. På så vis kan man till exempel tidigt hitta missar, feltolkningar och problem som i ett senare skede kan bli mycket dyra att åtgärda. Visualisering är i sammanhanget ett kraftfullt verktyg för att ge en intuitiv förståelse för andras planer och tankar. Man kan också återanvända data och utnyttja dem för att underlätta den långsiktiga förvaltningen. Det handlar både om att sänka kostnaderna och att höja kvaliteten i hela samhällsbyggnadsprocessen, genom alla steg från plan till projektering till konstruktion till förvaltning.

2.1 Stadsplanering

En grundläggande tillämpning är presentation av planerade byggen i en befintlig miljö. Detta kan vara värdefullt vid till exempel stadsplanering och samrådsprocesser, där en tredimensionell visualisering av den planerade konstruktionen i sitt sammanhang ger en mycket mer begriplig och intuitiv bild av slutresultatet än en tvådimensionell plankarta. Speciellt i kommunikation med allmänheten, som generellt har mindre erfarenhet av att tillgodogöra sig kart- och ritningsinformation, blir skillnaden stor. Figur 2.1 visar ett exempel från Göteborg, där modeller över föreslagen bebyggelse kombinerats med en 3D-vy över den befintliga.



Figur 2.1. Exempel på en stadsplaneringstillämpning (Göteborgs stad, 2015)

I ett sådant scenario finns normalt två huvudsakliga aktörer. Den ena är kommunen, som generellt har tillgång till detaljerad GIS-data i två dimensioner, och i allt större utsträckning även till enklare 3D-modeller av den befintliga staden. Den andra är en arkitekt eller byggherre, som i snart sagt alla större projekt har en tredimensionell CAD- eller BIM-modell av det tänkta bygget.

Målet blir att kombinera dessa datakällor för att skapa orienterbarhet och igenkänningsfaktorer, och kanske även som ett verktyg för att samla in synpunkter och förslag från allmänheten. Här är det mycket viktigt att verktyget är tillgängligt och enkelt att använda även för de som saknar specialistkunskap.

2.2 Stora anläggningar

Ett samredovisningsverktyg kan också vara användbart vid planering och underhåll av större, utspridda anläggningar med många byggnadskroppar, där interaktionen med omgivningen och byggnadernas relation till varandra spelar stor roll. Exempel kan vara bland annat industri-anläggningar och sjukhusområden. I många fall finns även här befintlig bebyggelse att ta hänsyn till (Ekelund, 2015).

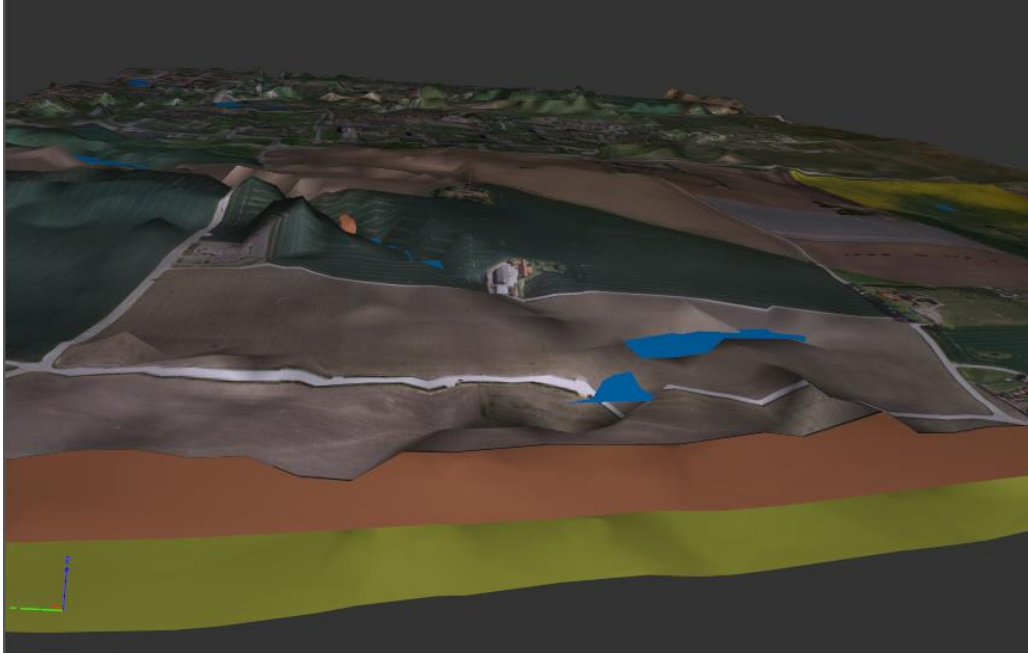
Stora projekt innebär generellt många olika parter och yrkesroller. Vissa, så som arkitekter och projektörer, tar fram egna data och modeller, och kan vara intresserade av att se sin del i förhållande till helheten och omgivningen. Andra är användare eller beställare, med insikter i vilka funktioner och egenskaper som ger verksamhetsnytta i framtiden, och som lättare kan sätta fingret på problem och möjligheter i en gemensam modell.

Det viktiga i en sådan användning är bland annat att modellen är enkel att ajourhålla allt eftersom projektet fortskrider, och att data visualiseras på en detaljnivå som gör den meningsfull utan att överväldiga den som inte är expert inom ett visst område. I denna typ av projekt finns det också stor potential i att återanvända byggnadsmodeller i förvaltningsskedet.

2.3 Undermarkskonstruktioner

En annan mer teknisk tillämpning skulle kunna vara vid undermarkskonstruktioner, där geologiska och geotekniska data över till exempel berggrundens uppbyggnad skulle kunna kombineras med anläggningsdata för att i ett tidigt skede kunna hitta potentiella problem eller risker. Geologiska data är en av de applikationer där behovet av 3D-GIS upplevts som stort, eftersom tvådimensionella representationer av geologi kräver mycket utbildning och erfarenhet att läsa och göra begripliga analyser på. Speciellt i kommunikation med icke-experten, både utanför och inom projektet, är tredimensionell visualisering betydelsefullt (Segerstedt, 2015). Figur 2.1 visar ett utsnitt ur en tredimensionell geologisk modell, med två olika bergytter, en grundvattenyta och markytan.

I underjordiska projekt är de huvudsakliga dataproducenterna geologer/geotekniker och de som designar och projekterar den tänkta anläggningen. Geologerna skapar tredimensionella GIS-modeller utifrån borrhningar och andra mätmetoder. Design sker normalt i någon form av CAD- eller BIM-program.



Figur 2.2 Exempel på geologisk data (Malmö geologiatlas, 2010).

I denna tillämpning är det viktigt med kraftfulla visualiseringsmöjligheter, eftersom det rör sig om objekt som vi inte ser eller har någon direkt relation till i våra vanliga liv. Det är viktigt att kunna kommunicera mellan yrkesgrupper och kanske framför allt med beställare och allmänhet, som saknar djupare kunskaper i hur undermarksföreteelser ser ut och fungerar. Också här finns det ett långsiktigt intresse i att bevara en kombinerad modell för att även i framtiden förstå hur till exempel berggrunden kring en viss anläggning ser ut.

2.4 Geografiskt utbredda anläggningar

Ett fjärde tänkbart användningsområde handlar om anläggningar med betydande geografisk utbredning, till exempel väg- eller järnvägsnät. Dessa projekteras normalt i CAD- eller BIM-program som tredimensionella kroppar. Däremot finns andra data som är relevanta för konstruktionen, som höjdmodeller över den befintliga terrängen eller exakta inmätta geografiska positioner för projekteringen, som snarast redovisas som geodata. Vilka aktörer som är inblandade och vem som har vilken data beror på projektstrukturen, men sannolika intressenter är trafikverket, projektörer, och entreprenörer. Även kommuner och andra myndigheter som ska planera för markanvändning och analysera påverkan kring nybygget kan vara inblandade.

Enkla kombinationer mellan datavärldarna skulle kunna vara ett stöd både i projektering, sikt- och bulleranalyser, och anläggningsplanering. En gemensam modell är också här ett stöd i kommunikationen med allmänhet och andra intressenter, och kan öka förståelsen för hur slutresultatet kan tänkas bli.

2.5 3D-Fastighetsbildning

Fastighetsbildning är också ett område där tredimensionell spatial information skulle kunna spela en mycket stor roll, speciellt vid bildande av 3D-fastigheter. Dessa har väldigt ofta gränser inuti byggnader, som inte markeras och som kan vara svåra att på ett otvetydigt vis beskriva med koordinater eller på en 2D-registerkarta. Idag redovisas detta normalt med en skriftlig beskrivning av gränsen, eventuellt kompletterad med ritningar och tvärsnitt. Samtidigt vet man att i väldigt många fall, framförallt vid ny- eller ombyggnadsprojekt, så har byggherren eller fastighetsägaren tillgång till en detaljerad 3D CAD- eller BIM-modell. Denna hade kunnat vara till stor hjälp med att tydligt bestämma gränserna, men lantmäteriet har i dagsläget varken teknisk möjlighet eller juridiskt stöd för att använda den (Lantmäteriet, 2014a).

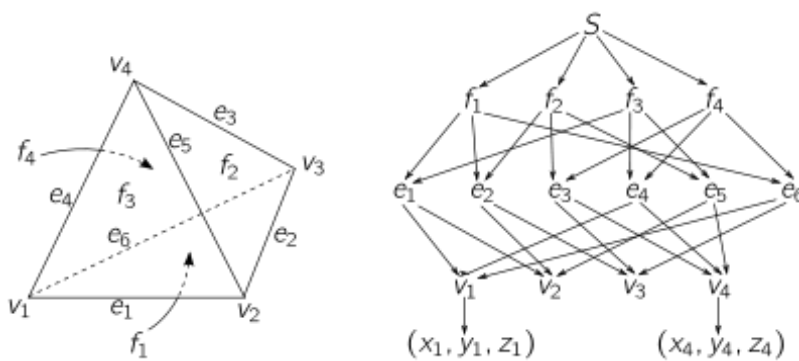
Internationellt pågår en hel del arbete kring tredimensionell fastighetsinformation. Det konstateras dock att det finns en lång rad på tekniska, juridiska och visualiseringsmässiga problem kring hanteringen, som inget land i världen ännu fullt ut lyckats lösa (Aien et al, 2013).

3. Geometrisk representation i tre dimensioner

En grundläggande del av alla typer av digitala tredimensionella data är den geometriska representationen, det vill säga hur ett tredimensionellt objekt beskrivs eller modelleras. Precis som tvådimensionella data kan representeras på olika sätt, vanligen antingen i en raster- eller vektorstruktur, så finns det ett par olika alternativ för beskrivning av 3D-data. Alla utgår från olika sätt att konceptualisera ett tredimensionellt objekt, vilket också ger upphov till helt olika datastrukturer.

3.1 Boundary representation (B-rep)

En vanlig modell är att definiera ett objekt genom att beskriva dess yttre avgränsning mot omvärlden, på engelska *boundary*. Detta görs med hjälp av punkter, linjer och ytor (eller i den engelska terminologin *vertex*, *edge* och *face*) i ett tredimensionellt rum. Till exempel kan en kub definieras som sex ytor, som i sin tur bestäms av tolv linjer, som binder samman åtta punkter. Den enda egentliga geometri som behöver lagras är punkternas, tillsammans med information om vilka punkter som skapar vilka linjer, vilka linjer som omsluter vilka ytor, och vilka ytor som omsluter vilken volym. Figur 3.1 visar ett exempel av en boundary representation av en tetraeder.



Figur 3.1. B-rep av en tetraeder (Wikipedia, 2015a).

Modellen kan också inkludera mer avancerade kurvor och ytor, för att modellera till exempel välvda strukturer, vilket gör att mängden objekt som kan modelleras på detta vis är mycket stort. Detta gör dock att komplexiteten ökar drastiskt, och för stora datamängder kan det resultera i orimligt långa processtider (Abdul-Rahman & Pilouk, 2008). Därför kräver en del implementeringar att enbart raka linjer och ytor i ett plan används. Detta är till exempel ett krav i CityGML (se avsnitt 4.1.2) (Gröger & Plumer, 2012).

Inom tredimensionell spelgrafik, där uppdateringshastigheten är av oerhörd vikt, är det mycket vanligt att man dessutom har kravet att alla ytor ska vara triangulära, eftersom en triangel alltid ligger i ett plan och därför blir matematiskt enkel att behandla (Angel & Shreiner, 2012). När topologin inte är viktig kan man också underlåta att lagra gränslinjerna, och knyta en yta direkt till tre ordnade gränspunkter. I sådana fall används ofta termen *mesh* snarare än *boundary representation*.

3.2 Voxlar

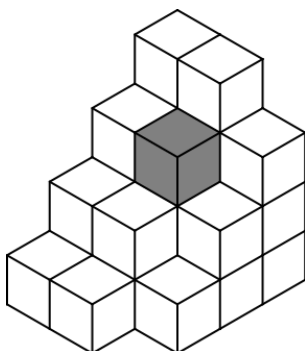
Om b-rep kan ses som den tredimensionella motsvarigheten till vektorrepresentationen, så är den tredimensionella motsvarigheten till raster voxlar (Angel & Shreiner, 2012). Istället för att dela upp ett tvådimensionellt rum i regelbundna celler och spara ett värde i varje cell delar man här upp ett tredimensionellt rum i regelbundna volymer och ger ett värde till varje volym. En principiell skiss på hur det kan se ut finns i figur 3.2. Ordet voxel är sammansatt från orden volym och pixel.

Modellen är intuitiv och lätt att använda, men genererar väldigt snabbt väldigt stora datavolymer. Speciellt i applikationer där många intilliggande volymer innehåller samma värde lagras ineffektivt mycket data. Därför har ett antal olika tekniker utvecklats där man varierar storleken på voxeln efter de data som ska modelleras. Ett exempel är octree-tekniken, som rekursivt delar upp varje kub som inte fullständigt innesluter objektet i åtta mindre kuber, till en bestämd upplösning är nådd. På så vis skapas en hierarkisk trädstruktur där områden som kräver högre upplösning modelleras av mindre kuber, och vice versa (Abdul-Rahman & Pilouk, 2008).

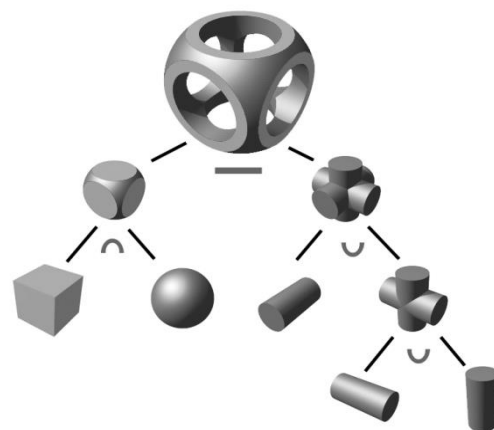
3.3 Constructive Solid Geometry (CSG)

Ett helt annat sätt att tänka sig ett tredimensionellt objekt är att se det som en sammansättning av ett begränsat antal fördefinierade volymer. Med utgångspunkt från till exempel rätblock, cylindrar och sfärer kan man genom att flytta, skala om, lägga ihop och dra ifrån volymer skapa mer avancerade objekt. Figur 3.3 visar ett exempel.

Detta är mycket vanligt inom till exempel CAD, och fungerar generallt väldigt bra för att modellera konstgjorda föremål. Modellen har däremot en del brister när det gäller att representera oregelbundna naturliga företeelser, och blir i sådana fall snabbt väldigt komplex och utrymmeskrävande (Abdul-Rahman & Pilouk, 2008).



Figur 3.2. Exempel på voxel-data (Wikipedia, 2015b).



Figur 3.3. CSG-modellering av ett komplext objekt från enkla primitiver (Wikipedia, 2015c).

4. 3D-data och dataformat

Mycket av problematiken kring integration av GIS- och BIM-data kan härledas till hur spatiala data modelleras och struktureras i olika dataformat. Även modeller som utåt sett verkar ganska lika kan ha helt olika intern representation, både vad gäller den geometriska representationen (se kapitel 3) och hanteringen av attribut, koordinatsystem och metadata. Skillnaderna kan huvudsakligen förklaras med historisk utveckling och skilda användningsområden. Något förenklat kan GIS ses som en digital vidareutveckling av papperskartan, som så tydligt som möjligt ska beskriva en existerande inmätt verklighet. BIM, å andra sidan, härstammar i grund och botten från byggnadsritningen, som är ett designverktyg och ska beskriva något som ännu inte finns och hur det ska konstrueras (Isikdag & Zlatanova, 2009). Dessa olika ursprung har lett till olika prioriteringar och olika grundläggande tankemodeller för vilka data som behöver lagras.

Generellt kan man säga att GIS-data jämfört med BIM bygger på enklare geometrier som möjliggör mer avancerade spatiala analyser. Visualisering i GIS-modeller handlar ofta om att skapa begripliga översikter över större områden, snarare än detaljerade modeller av enskilda byggnader eller utrymmen. Geografisk position och skalbarhet är viktigare i ett GIS än i en BIM-modell, medan BIM-världen har mycket större möjligheter till realistisk tredimensionell modellering (Mignard & Nicolle, 2014).

Både inom geodata och bygg- och anläggningsdata finns anledning att skilja på format för produktion och analys, format för datalagring och format för visualisering. När man ska producera eller arbeta med geografiska data är det viktigt med till exempel intuitiv behandling av geometrier, enkla uppdateringar och stöd för spatial analys, där prioriteringen naturligtvis skiftar beroende på arbetsområde. När data ska lagras och distribueras mellan användare är det viktigt med utrymmeseffektivitet, standardisering, och god hantering av metadata. Visualiseringsformat, å andra sidan, kräver stöd för just snabb och effektiv visualisering, men sällan för någon avancerad analys. Även om många format tekniskt sett klarar av alla tre uppgifterna, så är alla i någon mån specialiserade på någon del av processen.

4.1 Geodataformat

Att tala om geodata- eller GIS-format som samlade och homogena begrepp är i viss mån vilseledande. GIS-världen innehåller hundratals olika dataformat, där en del är standardiserade och allmängiltiga, och en del är proprietära och enbart fungerar i särskild programvara. Den absoluta majoriteten av dessa format hanterar dock världen i två dimensioner, projicerad på en plan yta (Mignard & Nicolle, 2014).

För tredimensionella analyser i GIS används dels höjdsatta data i så kallade 2,5D-modeller (se avsnitt 4.1.1 nedan), dels generella 3D-grafikformat som huvudsakligen hämtats från spelvärlden (se avsnitt 4.3). Det finns även ett par öppna format som utvecklats specifikt för hantering av geografisk information i tre dimensioner, där de viktigaste presenteras här nedan.

Det är viktigt att påpeka att även om både det upplevda behovet och den faktiska användningen av 3D-GIS växer, så finns fortfarande oklarheter och pågående diskussioner om vad som är lämpliga geometriska representationer, datamodeller och lagringsformat (Abdul-Rahman & Pilouk, 2008). Alla de dataformat som används idag har sina svagheter, och vilket format som väljs för ett visst projekt avgörs vanligen av de specifika krav som projektet ställer.

4.1.1 2,5D-data

Det enklaste sättet att inkludera höjd i en karta är att helt enkelt lagra det som ett attribut knutet till en cell i ett raster eller en punkt i en vektormodell. Detta benämns ofta 2,5D-data, eftersom det saknas en full tredimensionell beskrivning av geometrin (Zamyadi et al., 2014). På detta vis kan man på ett enkelt sätt skapa till exempel markmodeller i utrymmesnåla och väl beprövade dataformat. För många analyser är 2,5D fullt tillräckligt, och det används idag i stor utsträckning för terrängmodellering, hydrologi, siktanalyser, med mera.

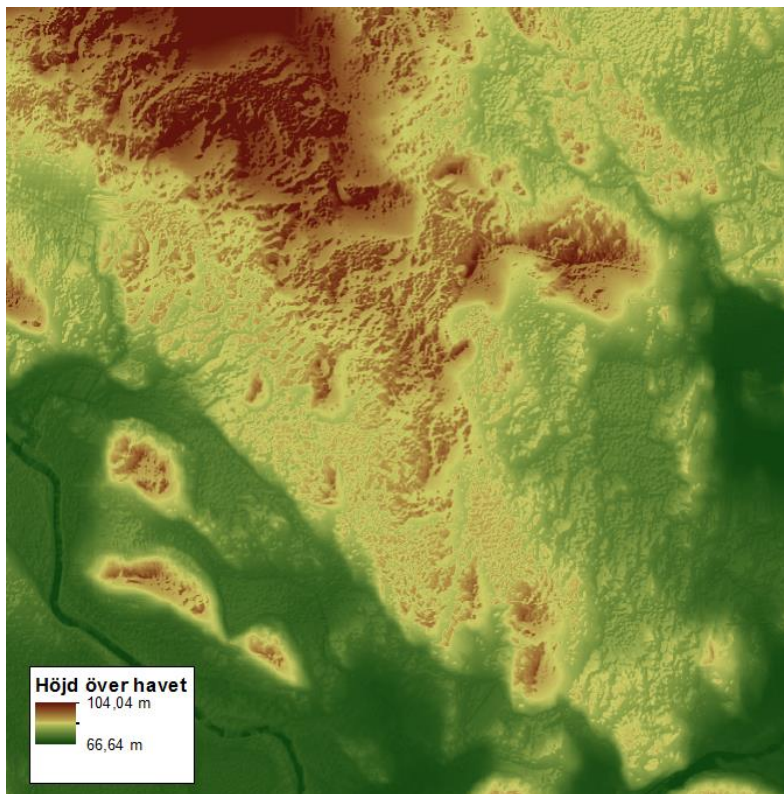
I sammanhanget skiljer man på en markmodell och en ytmodell. En markmodell ska representera den obebyggda och obeväxta markytan, ett slags teoretiskt golv som allt annat står på. En ytmodell inkluderar byggnader, anläggningar och vegetation, och representerar den övre ytan av både naturliga och konstruerade objekt. Ett annat begrepp som förekommer är terrängmodell, vilket i de flesta fall används synonymt till markmodell (Lantmäteriet, 2013a).

Lantmäteriets nationella höjdmodell, som baseras på laserskanning, är ett exempel på 2,5D-data. Modellen levereras bland annat som ASCII grid-fil, som redovisar höjder i ett rutnät, se figur 4.1. De första raderna definierar rastrets storlek, position och upplösning. Denna typ av data kan snabbt visualiseras som markmodell (se figur 4.2).

Förutom rasterdata, som i exemplet nedan, är ett vanligt förekommande format TIN (triangular irregular network), där höjdsatta punkter i ett oregelbundet nätverk binds samman till en yta. Det finns också vektormodeller som kan klassificeras som 2,5D, till exempel stadsmodeller där byggnader representeras av objekt som har ett höjdattribut. Dessa kan ofta visualiseras i 3D genom att byggnader extruderas till den angivna höjden, men geometri och topologi är fortfarande bara definierade i två dimensioner (Ledoux & Meijers, 2010).

```
ncols 1250
nrows 1250
xllcenter 595001.000
yllcenter 6725001.000
cellsize 2.0000
nodata_value -999
 93.85 94.04 94.00 93.93 94.02 94.09 94.15 94.11 94.07 93.84
93.97 94.10 94.04 94.00 94.07 94.10 94.13 94.05 93.97 93.89 93.96
93.80 93.79 93.84 93.81 93.80 93.63 93.73 93.60 93.48 93.57 93.65
93.64 93.21 93.07 93.03 93.07 93.11 93.12 93.02 93.03 93.04 93.00
92.97 92.93 92.96 92.98 92.80 92.76 92.85 92.81 92.79 92.76 92.66
92.62 92.70 92.54 92.34 92.17 92.29 92.14 91.94 91.74 91.56 91.45
91.37 91.30 91.19 91.08 90.94 90.78 90.72 90.26 90.12 90.04 89.99
89.93 89.79 89.66 89.62 89.61 89.54 89.36 89.38 89.21 89.14 89.14
89.06 89.10 89.01 89.08 89.14 89.01 89.07 89.09 88.99 89.01 89.16
89.11 89.04 89.24 88.85 88.68 88.95 88.98 89.07 89.10 89.16 89.18
89.05 89.18 89.25 89.33 89.44 89.45 89.61 89.62 89.62 89.65 89.74
```

Figur 4.1. Del av ASCII grid-fil (Lantmäteriet, 2013b).



Figur 4.2. Markmodell visualiserad i ArcMap från nationella höjddatamodellen (Lantmäteriet, 2013b).

Den stora nackdelen med 2,5D är att man endast kan lagra ett höjdvärde per punkt. Detta gör att modellering av till exempel olika våningsplan i ett hus, eller broar och tunnlar, inte kan göras på ett topologiskt korrekt vis. Det är också omöjligt att explicit lagra vertikala strukturer, till exempel väggar, och utstickande föremål, som balkonger. Utan slutna tredimensionella kroppar är det dessutom svårt att göra meningsfulla volymeräkningar (Gröger & Plümer, 2005). Dessa begränsningar blir framför allt problematiska i byggda miljöer och stadsmodellering, men det finns också applikationer inom till exempel geologi och riskanalyser där full 3D krävs för att framställa begripliga visualiseringar och genomföra viktiga analyser.

4.1.2 CityGML

CityGML är ett format som utvecklats specifikt för att lagra och transportera stadsmodeller i 3D. Det är ett applikationsschema av GML (geography markup language, ett standardiserat utbytesformat för geografiska data), som i sin tur baseras på XML. CityGML är antaget som officiell standard för utbyte av 3D-stadsmodeller av både OGC (Open Geospatial Consortium) och ISO (ISO TC 211), och även om det ännu inte finns några större datamängder tillgängliga i formatet så börjar det etableras som standard inom stadsmodellering världen över (Mignard & Nicolle, 2014). Det är också ett format som det forskats och skrivits mycket om under de senaste åren vilket gör att tillgången till dokumentation och applikationsexempel är god.

Målet med CityGML är att kunna representera allt som kan förekomma i en stadsmiljö på ett sätt som möjliggör inte bara visualisering utan även spatial analys. CityGML lagrar därför inte bara geometrier utan även topologi, semantiska beskrivningar och utseende för byggnader, vegetation, vatten, transportleder, med mera (OGC, 2012). Det finns också möjligheter för

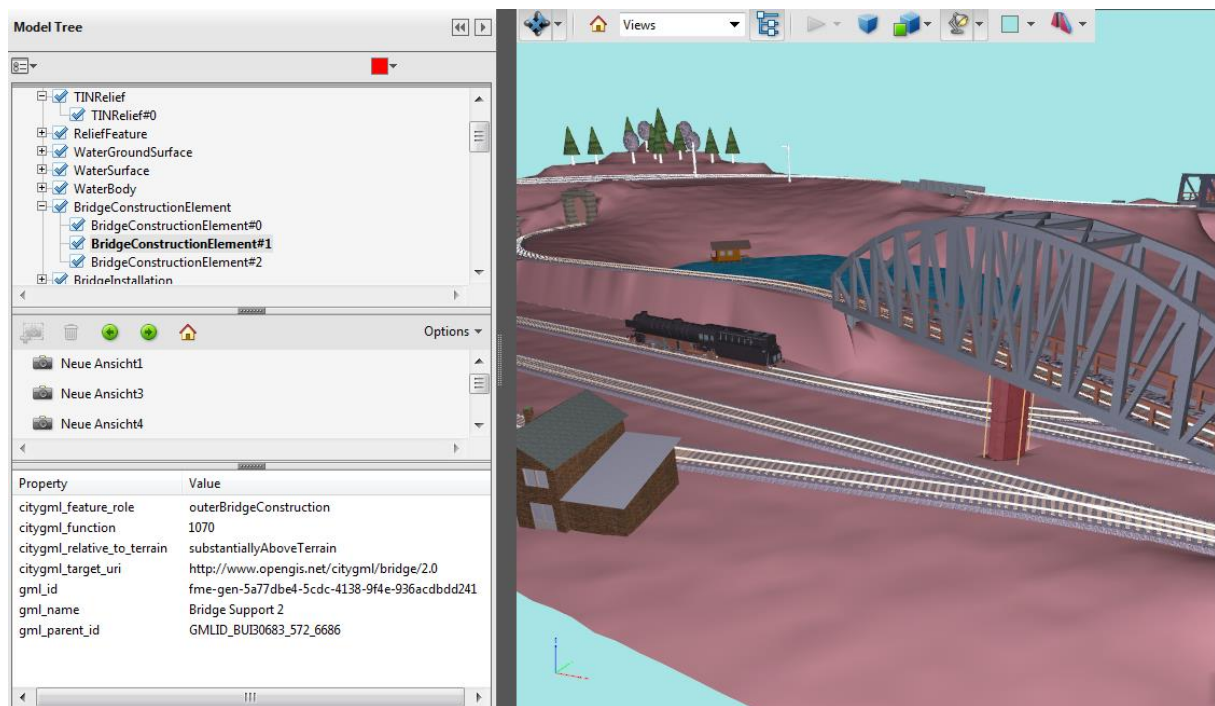
användare att bygga ut formatet med egna ADEs (application domain extensions) för användning i olika applikationer, till exempel geotekniska data (Tegtmeier et al., 2014) eller navigering inomhus (Kim et al., 2014). ADE-strukturen kan också användas för att skapa nationella profiler av CityGML för att ge bättre interoperabilitet med befintliga datamängder (van den Brink et al., 2013).

CityGML är definierat i fem olika detaljnivåer, LoD (Level of Detail) 0-4. LoD 0 är i princip en 2,5D terrängmodell. LoD 1 används för att visa upp hela städer där alla byggnader dragits upp till sin rätta höjd, men med platta tak, en så kallad boxmodell. På LoD 2 har taken form och fasaderna kan vara texturerade. LoD3 inkluderar till exempel fönster och dörrar som egna objekt, och LoD 4 har stöd för rumsindelning och interiörer. Ett objekt kan vara definierat på flera olika LoD samtidigt, vilket ger möjlighet till olika detaljerade representationer för olika användningar eller inzoomningsnivåer (Gröger & Plümer, 2012).

Figur 4.3 visar ett litet utklipp en CityGML-fil, där en del av en bropelare beskrivs. Som i alla XML-baserade format definieras varje element av en start- och en sluttagg, och kan i sin tur innehålla andra element i en hierarkisk struktur. I detta fall innehåller bropelar-elementet ett namn, sin relation till markytan, en funktionskod, och en geometri enligt GML-standarderna. Figur 4.4 visar en visualisering av samma fil, som innehåller bland annat broar, tunnlar, byggnader, vatten och enskilda träd. Denna demo-fil är uppbyggd för att inkludera exempel på alla typer av element som CityGML i sin nuvarande version klarar av.

```
<brid:outerBridgeConstruction>
  <brid:BridgeConstructionElement>
    <gml:name>Bridge Support 2</gml:name>
    <relativeToTerrain>substantiallyAboveTerrain</relativeToTerrain>
    <brid:function>1070</brid:function>
    <brid:lod3Geometry>
      <gml:MultiSurface>
        <gml:surfaceMember>
          <gml:Polygon gml:id="PolyID32737_1648_610211_332853">
            <gml:exterior>
              <gml:LinearRing gml:id="PolyID32737_1648_610211_332853_0">
                <gml:pos>8.08337940084355 2.08169752299809
                  8.36060602711122</gml:pos>
                <gml:pos>8.04874492569104 2.10170866155822
                  8.36058672492979</gml:pos>
                <gml:pos>8.0214223861279 2.09439749815595
                  8.36044410733769</gml:pos>
                <gml:pos>7.96139033108442 1.99049628686773
                  8.35964724945789</gml:pos>
                <gml:pos>7.96870216440647 1.96317376954023
                  8.3595239324782</gml:pos>
                <gml:pos>8.00333663955898 1.9431626309801
                  8.35954323465963</gml:pos>
                <gml:pos>8.03065917912213 1.95047379438238
                  8.35968585225172</gml:pos>
                <gml:pos>8.09069118135201 2.0543750306636
                  8.36048272191174</gml:pos>
                <gml:pos>8.08337940084355 2.08169752299809
                  8.36060602711122</gml:pos>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
      </gml:MultiSurface>
    </brid:lod3Geometry>
  </brid:BridgeConstructionElement>
</brid:outerBridgeConstruction>
```

Figur 4.3. Utdrag ur en CityGML-fil (citygml.org).



Figur 4.4. Visualisering av CityGML-fil som 3D-PDF. Brodelen som beskrivs ovan är markerad och dess attribut visas i nedre vänstra hörnet (citygml.org).

Filen ovan beskriver geometrin som ytor definierade av polygoner med koordinater i tre dimensioner, det vill säga en boundary representation (se avsnitt 3.1). CityGML innehåller även en helt tredimensionell datatyp, *solid*. Till varje geometri kan attributdata kopplas, liksom utseendeparametrar. Geometrimodellen är en delmängd av den som definieras i GML3, som i sin tur implementerar standarden ISO 19107 för spatiala beskrivningar av geografiska objekt (Gröger & Plümer, 2012).

Den stora fördelen med CityGML är detaljeringsgraden och den rika semantiken, som gör det möjligt att representera viktiga delar i stadsmodellen på ett sätt som ger stöd för avancerade analyser och visualisering. Baksidan av samma mynt är att CityGML-filer snabbt blir väldigt stora och svårhanterliga. I likhet med de flesta textbaserade dataformat är CityGML bäst lämpat för lagring och utbyte. Både modellering från grunden och visualisering är möjlig, men görs generellt sett bäst i andra verktyg och format (Mao et al., 2011).

4.1.3 KML

Ett annat format som används för geografiska data i 3D är KML. Förkortningen står för Keyhole Markup Language, och bygger precis som GML i grunden på XML. KML är från början knutet till Google Earth, och innehåller information om var och hur ett objekt ska visas upp på den digitala jordgloben. KML är numera, liksom CityGML, antaget som OGC-standard och fungerar i betydligt fler visualiseringsverktyg och digitala jordglober (OGC, 2008).

Den stora fördelen med KML är att det är enkelt och tillgängligt att arbeta med. Även oerfarna användare kan själva både skapa 3D-strukturer och exportera dem som KML, och importera KML-filer för att titta på i till exempel Google Earth. Det gör att det är ett användbart format

om man till exempel ska kommunicera med användare som inte har tillgång till dyra GIS-program (Ballagh et al., 2011).

En KML-fil lagrar en geografisk position för objektet, en geometri, en utsiktspunkt, och en del utseendeattribut. Geometrin kan lagras direkt som listor av 3D-koordinater, eller som COLLADA-fil (se avsnitt 4.3.1). Man kan lägga till en beskrivning, men till skillnad från CityGML lagrar den inte någon specificerad klassificering av objektet eller detaljeringsnivå. Det gör att formatet är lämpligt för visualisering och informationsutbyte, men det har inte stöd för några mer avancerade spatiala analyser.

4.2 Bygg- & Anläggningsdataformat

Precis som för geodata gäller att data för bygg- och anläggningsprojekt är en mycket heterogen grupp som innehåller allt från enkla inskannade linjeritningar till avancerade objektorienterade 4D informationsmodeller (dvs. 3D plus tid). Många dataformat är proprietära och knutna till speciella programvaror, vilket bidrar till mångfalden, men det finns även öppna format och format som fått spridning som de facto-standarder inom industrin.

Den stora spridningen bland dataformaten kan i stor utsträckning spåras till den snabba teknikutvecklingen som skett inom området. CAD har använts som verktyg inom bygg- och anläggningsindustrin för att skapa tvådimensionella ritningar i årtionden. Den första generationen innehöll i princip enbart geometriska primitiver, som linjer och cirkelsegment. Attributdata och teckenförklaringar kunde skrivas till på ritningen, men var inte knutna till geometrin på datanivå. I princip framställdes samma produkt som tidigare ritats för hand, fast nu med datorns hjälp (Akin, 2009).

I takt med att 3D-grafik började bli möjlig för datorerna att hantera framställdes allt fler 3D-modeller. Dessa har dels fördelar ur visualiseringsperspektiv, och gör dels att 2D-planer och sektioner som genereras ur modellen per automatik kommer stämma överens, vilket eliminerar dyra felkällor (Eastman et al., 2008). Ytterligare ett steg mer avancerade system börjar introducera objektorientering, det vill säga att linjer inte bara är linjer utan delar av komplexa objekt som inkluderar attribut och regler som styr beteenden. Vid den mognadsgraden börjar man övergå från att prata om CAD till att prata om BIM.

Det finns ingen exakt gemensam definition av vad som egentligen måste finnas för att en modell ska räknas som en BIM-modell snarare än 3D-CAD. I stora drag rör det sig om att det ska finnas en tredimensionell, objektorienterad CAD-modell av byggnaden, men också icke-spatial information kring till exempel förvaltning och användning som knyts till den geometriska modellen. Tanken är att alla som är inblandade i design, projektering, uppförande och förvaltning av en byggnad ska dela samma information, och att denna ska finnas och hållas uppdaterad under hela byggnadens livscykel (Eastman et al., 2008).

Det är viktigt att påpeka att data på alla dessa olika mognadsnivåer fortfarande finns och aktivt används. BIM som koncept är fortfarande relativt nytt, och många byggföretag har inte funnit det värt resurserna att övergå till det sättet att tänka. I mindre projekt, där kostnaderna och

riskerna vid informationsutbyte är relativt sett små, används enklare 3D-modeller och även traditionella 2D-ritningar fortfarande med gott resultat.

4.2.1 CAD-format

CAD-program har idag stöd för import och export av data i en stor mängd format i både två och tre dimensioner. För informationsutbyte används allt från vanliga rasterbaserade bildformat som JPG, till specialiserade objektbaserade 3D-format (Eastman et al, 2008). Det vanligaste är tredimensionella vektorbaserade format, där flera olika mer eller mindre officiella standarder finns. Ett tidigt standardformat var IGES, som utvecklades i USA redan i början av 1980-talet, men fortfarande används. Den officiella ISO-standarden för CAD-datautbyte heter STEP (ISO 10303), som har en mycket stor mängd applikationsspecifika implementationer för allt från elkretsar till skeppsbyggnad (Eastman et al, 2008).

Inom byggnads- och anläggningsdata har DWG (från *drawing*) fått viss ställning som standardformat (Akin, 2009). DWG är ett två- och tredimensionellt vektorformat som i första hand utvecklas och upprätthålls av företaget Autodesk, och ligger till grund för deras AutoCAD-mjukvara. Eftersom formatet är proprietärt finns det ingen offentligt tillgänglig dokumentation, och detaljer kring exakt hur det fungerar är inte allmänt kända. Det är ett binärt format, vilket innebär att filerna inte är direkt läsbara av människor, och innehåller i första hand geometri och metadata. Exakt vilka geometrier som stöds beror på vilken version som används, och utvecklas kontinuerligt. Geometrier kan packas ihop i återanvändbara block, om man till exempel vill göra flera väggar med exakt samma konstruktion, men de innehåller ingen semantisk eller topologisk information. DWG-formatet är kopplat till ett textbaserat utbytesformat som heter DXF (drawing exchange format). Det innehåller i princip samma information, men med taggar som talar om vilken typ av data som representeras (Autodesk, 2015).

4.2.2 IFC

Även BIM-världen innehåller en stor mängd dataformat, där i princip varje större programvaruleverantör utvecklat ett eget, tätt knutet till interna processer och datamodeller. Däremot finns det här ett ganska tydligt och allt mer betydande standardformat för utbyte, nämligen IFC (Industry Foundation Classes).

I samband med utvecklingen av BIM noterades behovet av ett öppet standardiserat dataformat som kan hantera både detaljerade 3D-geometrier och objektorientering. Den stora variationen av dataformat inom CAD-världen sågs som ett problem för interoperabilitet och datautbyte mellan professioner och program (Eastman et al, 2008). Man ville undvika dataformat som var knutna till enskilda företag och mjukvarupaket, eftersom det gör den framtida utvecklingen osäker och okontrollerbar för alla andra intressenter. Det finns också en speciell problematik inom branschen som härrör ur det faktum att ritningar skapas när en byggnad eller anläggning uppförs, men potentiellt behöver användas under byggnadens hela livslängd. Dataformat har under de senaste decennierna blivit omoderna och utbytta betydligt

oftare än byggnader, vilket kan skapa stora problem vid till exempel underhåll och ombyggnader (Harr, 2015). Med tanke på dessa behov utvecklades IFC.

Tanken med IFC är att formatet ska utgöra en stabil, heltäckande grund som täcker in allt som kan tänkas behöva modelleras hos en byggnad eller anläggning. Modellen definierar breda basklasser för objekt och datatyper som sedan kan användas för att skapa mer och mer detaljerade subklasser i hierarkiska träd. Antalet klasser som erbjuds i modellen utvidgas kontinuerligt allt eftersom CAD- och BIM-programmen utvecklar nya tjänster och funktioner.

IFC utvecklas och underhålls av en internationell ideell organisation som heter BuildingSMART och består av industrirepresentanter från hela världen. Modellen är antagen som ISO-standard (ISO 16739), och växer i betydelse för varje år. IFC är precis som CityGML i första hand uppbyggd för informationsutbyte, inte analys eller presentation, och oavsett vilket filformat som används internt kan ett stort och växande antal CAD- och BIM-program importera och exportera IFC-data (BuildingSMART, 2015). Filformatet är baserat på modelleringspråket EXPRESS, men finns också i en XML-version, för att öka antalet program som kan hantera det.

IFC är uppbyggt i fyra succesivt mer detaljerade lager. Det finns ett resurslager (*resource layer*), med generella klasser för baskoncept som geometri, aktörer och kostnader. Sen finns ett baslager (*core layer*) som innehåller basen för alla enskilda entiteter i modellen. Interoperabilitetslagret (*interoperability layer*) innehåller moduler för koncept som delas av flera olika applikationsdomäner. Slutligen finns domänlagret (*domain layer*) som delar upp schemat i olika delar beroende på applikation. Tanken är att en programvara bara behöver implementera en del av modellen för att vara kompatibelt med andra delar. Till exempel kan el- eller VVS-installationer projekteras i specialiserad och invand programvara för att sedan exporteras till IFC för att samredovisas med arkitektmodellen och fastighetsförvaltarens underhållskalkyler (BuildingSMART, 2013).

En IFC-fil lagrar geometri i två eller tre dimensioner, i form av till exempel B-rep eller CSG (se kapitel 3). Till geometrin kan knytas attribut, länkar, relationer till andra objekt, med mera. För att öka flexibiliteten kring vilka attribut som kan knytas till olika objekt kan man själv definiera så kallade *property sets*, som definierar attribut och deras datatyper. Klasstrukturen möjliggör sökning och analys i modellen.

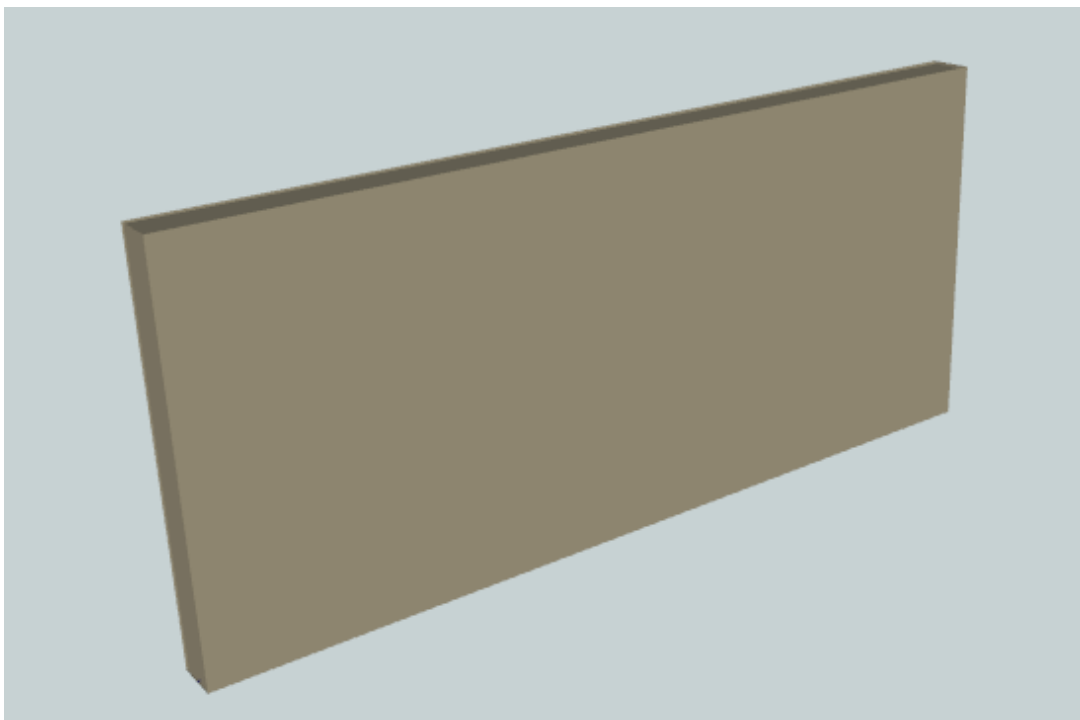
Enskilda IFC-element lagrar, i likhet med de allra flesta CAD- och BIM-format, sin position i ett lokalt koordinatsystem. En IFC-fil har dock möjlighet att knyta en geografisk position till *IfcSite*-elementet, som beskriver platsen eller tomten som byggnaden står på. Denna definieras då med latitud och longitud i WGS84 och höjd över havet (BuildingSMART, 2013). Detta gör det möjligt att beräkna den geografiska positionen för varje element i modellen, även om den inte lagras explicit (se Rafiee et al, 2014). Attributet är dock inte obligatoriskt, och finns inte i alla filer. Det finns inte heller några krav på noggrannheten.

IFC innehåller klasser för väldigt många olika byggnadsdelar på väldigt hög detaljnivå. För objekt som ännu inte har en egen klass i specifikationen finns en generisk *BuildingElementProxy*-klass. Det pågår också en hel del forskning och utveckling kring utvidgningar till IFC, även om det kan ta lång tid från det att en förändring föreslagits till dess att den ingår i den formella standarden. Något som är speciellt tydligt i det här sammanhanget är att det helt saknas klasser för mark, vägar, anläggningar, det vill säga allt som inte kan hierarkiskt underordnas själva byggnaden. Det finns planer på en integrering med LandXML, som hanterar just mark- och anläggningsdata, men dessa ligger ännu i framtiden (Ekholm et al., 2013).

Figur 4.5 visar ett litet urklipp ur en IFC-fil, som definierar geometrin för den vägg som syns i figur 4.6. Varje nummerad rad beskriver ett objekt, som i sin tur kan byggas upp av referenser till andra objekt på andra rader. I detta fall beskrivs väggen som en extrusion, där rektangeln som definieras i rad 41 längs axeln som definieras i rad 40 med hjälp av punkten i rad 39 dras upp 3500 mm i rad 42. Den får också utseende och färg. Hela filen, som alltså bara innehåller geometrin för väggen i figur 4.6 och dess beskrivning, är på 75 rader, och innehåller bland annat definitionen av koordinatsystemet, och alla egenskaper som knutits till väggen. Filen som innehåller hela våningsplanet som väggen ingår i har nästan en halv miljon rader.

```
#36=IFCCARTESIANPOINT((7900.,0.));
#37=IFCPOLYLINE((#3,#36));
#38=IFCSHAPEREREPRESENTATION(#18,'Axis','Curve2D',(#37));
#39=IFCCARTESIANPOINT((3950.,0.));
#40=IFCAXIS2PLACEMENT2D(#39,#11);
#41=IFCRECTANGLEPROFILEDEF(.AREA.,$, #40,7900.,400.);
#42=IFCEXTRUDEDAREASOLID(#41,#16,#9,3500.);
#43=IFCCOLOURRGB($,0.866667,0.8,0.615686);
#44=IFCSURFACESTYLERENDERING(#43,0.,IFCNORMALISEDRAIOMEASURE(1.))
#45=IFCSURFACESTYLE('Cembrit 3',.BOTH.,(#44));
#46=IFCPRESENTATIONSTYLEASSIGNMENT((#45));
#47=IFCSTYLEDITEM(#42, (#46), $);
#48=IFCSHAPEREREPRESENTATION(#19,'Body','SweptSolid',(#42));
#49=IFCPRODUCTDEFINITIONSHAPE($,$, (#38,#48));
#50=IFCCARTESIANPOINT((17900.,9800.,36700.));
#51=IFCAXIS2PLACEMENT3D(#50,$,#6);
#52=IFCLOCALPLACEMENT(#34,#51);
#53=IFCWALLSTANDARDCASE('0hi5tJtk5538ZNPPoJvjqv',#23,'Basic Wall:');
```

Figur 4.5. Utdrag ur IFC-fil.



Figur 4.6 Väggen som beskrivs i IFC-filen ovan.

4.3 Generella 3D-dataformat

Förutom de dataformat som utvecklats specifikt för geo- och byggnadsdata finns det en hel del 3D-dataformat som utvecklats för andra eller mer generella syften men som ändå används i betydande utsträckning inom GIS- och BIM-världarna. Många har uppkommit i spel- eller animationsvärlden, som är en stor pådrivare av 3D-grafikutvecklingen, eller mer generellt i 3D-modellering. Det rör sig allmänt om ganska renodlade grafikformat, utan några avancerade semantiska eller topologiska egenskaper, men som stöder snabb visualisering eller effektivt datautbyte. De format som har direkt koppling till olika visualiseringsmiljöer kommer att behandlas i kapitel 6.

4.3.1 COLLADA

COLLADA är ett öppet XML-baserat filformat som används inom många olika 3D-applikationer. Det är varken optimerat för att skapa innehåll eller visualisera det, utan i första hand tänkt som ett utbytes- och transportformat. Formatet lagrar geometri, texturer och material, ljus och animationer på ett sätt som kan skrivas och läsas av ett stort antal olika program inom olika applikationer. Det utvecklas av Khronos-gruppen, som är ett konsortium av olika intressenter inom datorgrafik, och antaget som ISO-standard (ISO/PAS 17506) sedan 2012.

Av speciellt intresse i sammanhanget är att formatet ger möjlighet att definiera en geografisk position för ett objekt, och därför kan användas för att definiera geografiska 3D-modeller. COLLADA används även som bärare av tredimensionella geometrier i KML-filer.

4.3.2 OBJ

OBJ är ett förhållandevis icke-komplicerat filformat som lagrar geometri och vissa utseende-parametrar. I det enklaste utförandet finns enbart en lista med punktkoordinater och en lista med vilka punkter som definierar vilka polygoner. Man kan också lagra texturkoordinater, normaler och referenser till materialbeskrivningar.

Formatet är utvecklat av företaget Wavefront, ursprungligen till programmet Advanced Visualizer, men har spridits och används nu i många olika 3D-applikationer samt som utbytesformat.

4.4 3D-data i Sverige

Det finns ett antal viktiga aktörer inom produktionen av spatiala 3D-data i Sverige. Generellt kommer geodata oftast från offentliga aktörer, och byggnads- och anläggningsdata oftast från privata, även om det naturligtvis finns undantag. Nationellt sett skapas och lagras 3D-data ännu inte på något systematiskt eller standardiserat vis (Lantmäteriet, 2014b).

Lantmäteriet producerar den nationella höjddatamodellen som presenterades i avsnitt 4.1.1, och tillhandahåller också de laserskanningsdata den bygger på, klassificerade som mark, vatten och övrigt (Lantmäteriet, 2015). Sjöfartsverket har en liknande laserskannad modell av havs- och sjöbottnar, som täcker stora delar av de svenska farvattnen. Andra stora offentliga geodataproducenter är Trafikverket, Sveriges geologiska undersökning (SGU) och Försvarsmakten. Dessa har inga heltäckande datamängder, men jobbar med 3D-data inom enskilda projekt (Lantmäteriet, 2014b). Trafikverket har börjat kräva att BIM används i upphandlingar, och ser långsiktigt att modellerna även ska användas för förvaltning och underhåll (Trafikverket, 2015).

På lokal nivå är det kommunerna som står för den största produktionen av 3D-geodata. Det finns en stor spridning i landet, från kommuner som inte hanterar 3D-data alls, till kommuner som har fullständiga modeller över åtminstone tätortsområden. Generellt sett gäller att större kommuner med större resurser har kommit längre i sin 3D-mognad (Lantmäteriet, 2014b). Vilka tekniska lösningar och datainsamlingsmetoder som använts skiljer sig åt från kommun till kommun. Många använder kombinationer av laserskanningar och flygfoton för att få fram draperade 3D-bildmodeller. Andra använder parametriska modelleringsverktyg för att till exempel extrudera byggnaders fotavtryck till en laserskannad höjd. Det finns också stora skillnader kring hur data används och görs tillgängliga. Göteborgs kommun använder en 3D-modell online för att till exempel samla medborgarförslag och visa upp planerad bebyggelse (Göteborgs stad, 2015). I Stockholm kan man beställa utsnitt ur 3D-modellen för egna användningar, medan till exempel Malmö än så länge bara använder sin 3D-modell internt.

BIM-data framställs huvudsakligen vid planering och projektering av nya byggnads- och anläggningsprojekt, även om BIM av befintliga byggnader börjat undersökas (Volk et al., 2013). Det rör sig sällan om en enskild dataproducent, tvärtom är en av de stora fördelarna med angreppssättet att man kan kombinera data från många olika håll. Till exempel kan

arkitekter, konstruktörer, VVS-projektörer och elektriker alla bidra med sin del i samma projekt. För att modellen efter byggnadsfasen ska kunna användas vidare i förvaltningsskedet krävs att den i någon form ägs av eller görs tillgänglig för förvaltaren. De juridiska frågorna kring detta är inte fullt utredda, vilket sannolikt till viss del fördröjer utvecklingen (Ekholm et al., 2013).

Förutom bristen på samordning och gemensamma standarder är det sannolikt ett problem för 3D-utvecklingen att så lite data är offentligt tillgänglig. Det är svårt att argumentera för vikten av kombinerbarhet och datautbyte, när det finns så pass lite data tillgängliga att faktiskt kombinera med. Mycket 3D-data framställs inom enskilda projekt, och när projektet väl är slut finns det idag få processer på plats för att återanvända och ajourhålla informationen.

5. Integrationsmetoder

På en helt abstrakt nivå skulle man kunna säga att BIM och GIS egentligen är samma sak. Båda är system som representerar en befintlig eller tänkt verklighet som objektorienterade modeller där attribut av olika slag knyts till geometriska representationer. Det finns de som hävdar att GIS är en typ av tematiskt utvidgat BIM, och de som hävdar att BIM är en typ av specialiserat GIS. Trots det principiella släktskapet finns idag stora problem med interoperabilitet mellan de olika datasfärerna.

Historiskt kan man säga att geodata och bygg- och anläggningsdata länge utvecklades i helt separata världar med helt olika verktyg och prioriteringar. Att man idag kan tala om integrationsproblematik är egentligen snarast en konsekvens av att världarna för första gången kommit så nära varandra att integration är möjlig överhuvudtaget. Geodatavärlden har börjat tänka i 3D, och övergången mot BIM har infört objektorientering med sökbara attribut och analysmöjligheter till bygg- och anläggningsdata. Decennier av åtskillnad har dock lämnat en hel del barriärer.

5.1 Problematiken

Integrationsproblemet är i själva verket inte ett utan flera relaterade problem. Självklart varierar de i relevans över olika tillämpningar och olika dataformat, men på en generell nivå kan man identifiera ett antal som nästan alltid har betydelse.

För det första handlar det om koordinatsystem och positionering. Många CAD- och BIM-format saknar helt möjlighet till geografisk positionering, eller behandlar det som valfritt. Även om geografiska positioner förekommer finns de ofta bara i en eller ett par punkter i modellen, ofta med dålig precision och okänd mätosäkerhet. Att hantera en hel BIM-modell med tusentals komponenter i ett geografiskt koordinatsystem blir helt enkelt för tungrott, eftersom geografiska koordinater behöver lagras med ett stort antal signifikanta siffror på den detaljeringsnivå BIM kräver (Harr, 2015). Dessutom blir det vid projekt med stor geografisk utberedning problematiskt att BIM-data är definierade i ett rätvinkligt kartesiskt koordinatsystem, medan GIS-data normalt arbetar antingen i ett projicerat geografiskt referenssystem, eller som geografiska koordinater på en definierad ellipsoid. Enkelt uttryckt antar BIM-data för enkelhetens skull att jorden över en liten yta kan ses som platt, medan GIS-data till priset av en del matematisk komplexitet försöker hantera det faktum att den inte är det. Detta gör det svårt att integrera BIM-modellen i en GIS-värld utan svårlösta fel och mycket manuella åtgärder, och vice-versa. Det finns inte heller någon allmän standard för hur höjder anges i BIM-system.

Det finns även andra skillnader i vilken data som lagras som kan ställa till problem vid översättningar. BIM-modeller innehåller till exempel ofta komplicerad geometri i form av bland annat matematiskt definierade ytor som inte GIS-mjukvara idag klarar av (Isikdag & Zlatanova, 2009). Dessa kan i bästa fall approximeras och ersättas med enklare geometrier för

att kunna läsas i ett GIS-program, med viss förlust i utseende och lagringseffektivitet. I värsta fall läses de inte in över huvudtaget, om ingen förenklingsalgoritm finns implementerad.

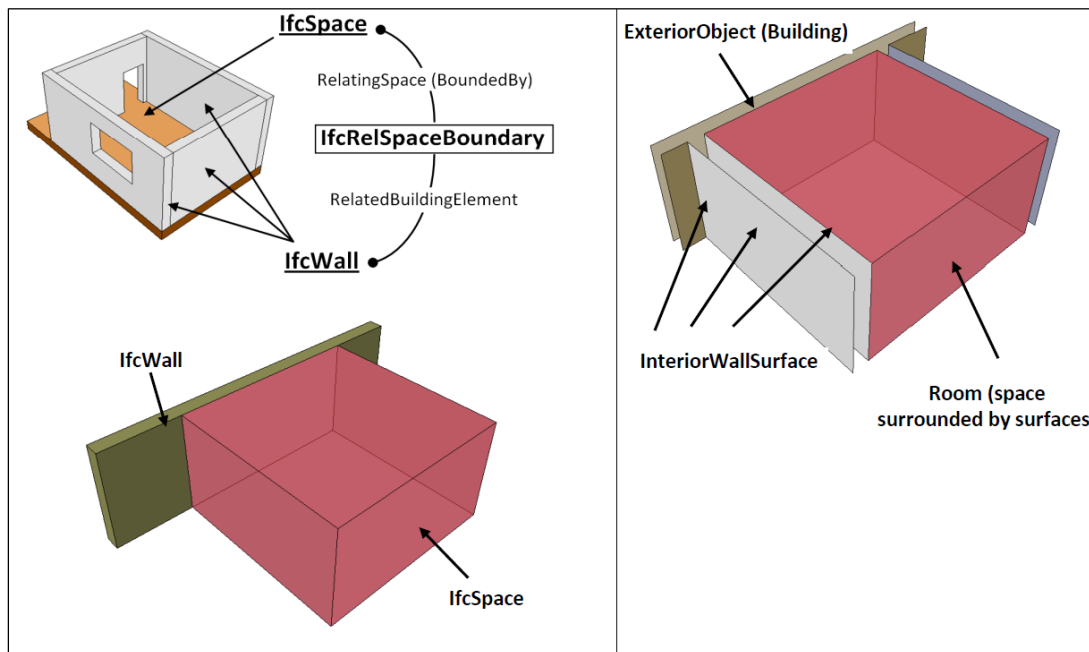
Ett relaterat problem är detaljeringsgraden. En BIM-modell kan innehålla information ner på millimeternivå, och en enskild modell kan innehålla tusentals objekt. Däremot är den geografiska utberedningen sällan särskilt stor, eftersom filstorleken i sådant fall lätt skulle bli ohanterlig. För en GIS-datamängd gäller ofta det motsatta. Dessa skillnader måste på något vis hanteras vid en dataintegration, oftast genom att förenkla BIM-modellen (Isikdag & Zlatanova, 2009). Eftersom GIS-data generellt rör inmätta existerande objekt snarare än designade sådana måste man också här på något vis hantera problem som mätosäkerhet, datakvalitet och aktualitet.

Detaljrikedom i GIS-data hanteras dessutom ofta med någon form av skalbarhet eller LoD-begrepp, det vill säga att data förenklas eller döljs helt vid utzoomning. Detta används i praktiken idag inte alls i BIM-världen, även om man kan tänka sig applikationer, kanske framförallt inom anläggningsbranschen, där det skulle kunna ha en ganska naturlig plats. Borrmann et al. (2014) presenterar en implementering av LoD-begreppet i en IFC-utvidgning för tunnelkonstruktioner. De konstaterar att det är fullt görbart, men kräver en annan typ av implementation än den som finns i till exempel CityGML. Det beror på att en BIM-konstruktör, på ett helt annat sätt än en stadsmodellerare, rör sig fram och tillbaka mellan komplexitetsnivåerna medan modellen skapas. Att hålla informationen konsekvent mellan detaljnivåerna kräver då en hel del eftertanke.

Den kanske allra svåraste frågan när man talar om integration mellan BIM- och GIS-data är den som handlar om informationsmodeller och schemaöversättningar. Detta är framförallt ett problem när man diskuterar översättningar mellan semantiskt rika format som IFC och CityGML. En del objektclasser har helt enkelt inte någon motsvarighet i den andra modellen, och översätts i bästa fall som generiska objekt, i sämsta fall inte alls. Till exempel innehåller IFC en stor mängd specialiserade klasser för installationer och el som inte har någon motsvarighet i CityGML, och CityGML beskriver trafikytor och vegetation på ett sätt som inte IFC hanterar. Kanske ännu mer svårlösta är de fall där olika dataformat har konceptualiserat samma reella företeelse på olika sätt. Informationen går att utläsa, men kräver specialbehandling för att extraheras och översättas.

Ett illustrativt exempel kan vara en enkel vägg i IFC respektive CityGML, vilket visas i figur 5.1. I IFC lagras en vägg som en *IfcWall*, som är en subclass av *IfcBuildingElement*. En speciell klass håller reda på väggens förhållande till omgivande utrymmen (*IfcSpace*). I CityGML, däremot, finns ingen direkt klass för väggar. Istället definieras ett hus med hjälp av bland annat en samling väggytor och ett rum med en samling innerväggsytor. Det som i IFC är en solid vägggeometri blir här utrymmet mellan två ytor, som är knutna till olika objekt på olika platser i hierarkin. Detta är inte någon omvandling som enkelt låter sig automatiseras (El-Mekawy et al., 2012).

Utöver de rent tekniska problemen finns en påtaglig mänsklig och organisatorisk aspekt som till viss del bromsar utvecklingen. De branscher och organisationer som använder GIS och de som använder BIM är idag ofta helt åtskilda. Man har egna verktyg, egna prioriteringar och ser olika framtidsutveckling. Kunskapen om vilka datamängder som finns i den andra världen och hur de fungerar är ofta ganska liten, och det finns väldigt få individer som är kunniga inom båda världarna.



Figur 5.1. En vägg i IFC respektive CityGML (El-Mekawy et al., 2012).

5.2 Föreslagna lösningar

Sen ett par år tillbaka finns det en växande forskning kring hur man teoretiskt gör för att flytta data mellan BIM- och GIS-världarna. Eftersom det på många sätt är det svåraste problemet har mycket av forskningen fokuserat på de semantiska skillnaderna, och specifikt på översättning mellan IFC och CityGML. Till exempel har Isikdag och Zlatanova (2009), utvecklat ett ramverk för översättning av IFC till alla nivåer av CityGML, i termer av både semantik och geometri.

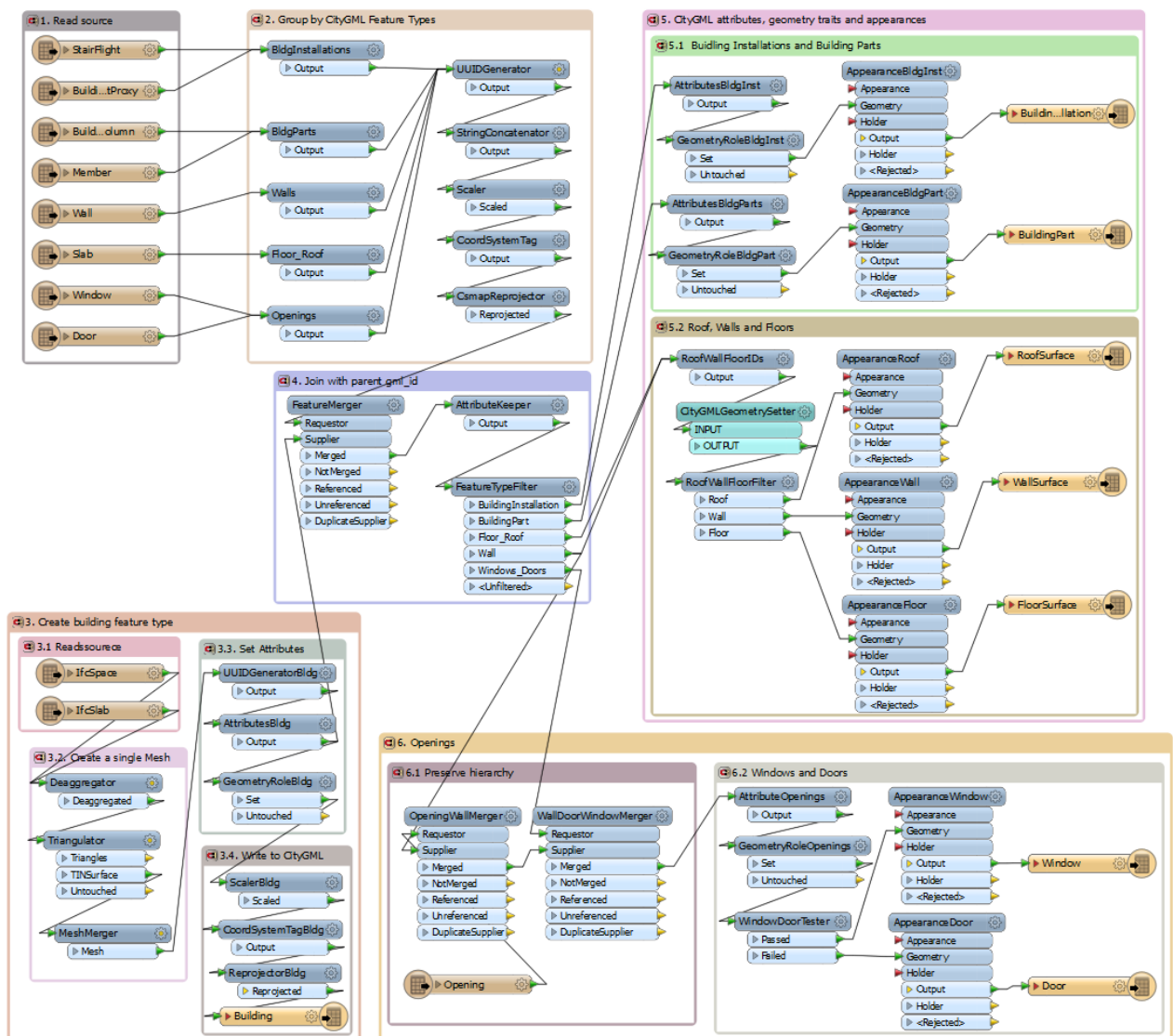
El-Mekawy (2013) gör en översikt över de försök som gjorts hittills, och konstaterar dels att huvuddelen av de metoder som föreslagits handlar om enkelriktad översättning från IFC till CityGML, och dels att diskussionerna kring vad som borde översättas och hur ännu inte kommit fram till någon slutgiltig översättningsmodell. Hans avhandling fortsätter sedan med att beskriva en UBM (*Unified Building Model*), som ska kunna användas som mellansteg för att gå från IFC till CityGML och vice versa, utan att tappa i informationsinnehåll. Modellen begränsas dock till byggnader, och täcker inte in till exempel markdata.

Mignard & Nicolle (2014) har byggt upp ett bredare ramverk, som förutom IFC och CityGML också kan ta in till exempel DWG- och shape-filer, georeferera, bygga en gemensam geometri

och även få med sig all hierarkisk och semantisk information i modellerna. Som kommersiell produkt finns ramverket tillgängligt som ACTIVE3D, med den stora nackdelen att i princip all dokumentation är på franska (ACTIVE3D, 2015).

Utanför den akademiska sfären har mjukvaruproducenter i både GIS- och BIM-världen gradvis börjat inse fördelarna med att kunna ta in data från andra sidan. Vissa CAD-produkter för till exempel väg- och vattenbyggnad kan ta in geografiska datalager, och många har möjlighet till KML-export. På motsvarande sätt har GIS-program börjat kunna ta in vanliga CAD-exportformat. Översättningarna är dock sällan helt problemfria, och begränsas ofta till ren geometri. De kräver också att det någonstans finns specificerat hur koordinatsystemen förhåller sig till varandra.

Ett program som specialiserat sig på området är FME (*Feature Manipulation Engine*) från företaget Safe Software, som är ett universalverktyg för att läsa, omvandla och skriva data från och till hundratals olika spatiala dataformat. Listan på dataformat inkluderar både GIS-, CAD- och BIM-format, och är växande, även om den är långt ifrån komplett. Det finns till exempel stöd för att översätta mellan IFC och CityGML, även om det inte är någon automatisk process utan kräver en hel del arbete. Figur 5.2 visar ett FME-workspace som flyttar en byggnad från IFC till LoD3 CityGML, för att ge en bild av hur många operationer som faktiskt krävs.



Figur 5.2 FME-workspace som omvandlar IFC till LoD3 CityGML (safe.com, 2015a).

6. Visualisering online

Även om 3D-modeller av byggnader och städer relativt länge har kunnat konstrueras och visas upp på enskilda datorer, så är det inte förrän de senaste åren som dessa modeller enkelt kunnat distribueras över internet. Tidigare har tillgängligheten begränsats av bland annat bandbredd och behovet av plug-in program. Nu finns det dock ett ganska stort urval av metoder som kan användas för att med relativ lätthet visa upp tredimensionellt innehåll direkt via nätet (Evans et al., 2014).

Det grundläggande genombrottet kom i samband med spridningen av en ny version av HTML (Hyper Text Markup Language), det grundläggande märkspråket på internet. Till skillnad från sina föregångare har HTML 5, som antogs som rekommenderad standard av W3C (World Wide Web Consortium) i oktober 2014, direkt stöd för 3D-grafik. Ett nytt element, <canvas>, har introducerats för grafik som ska genereras snarare än statiskt visas upp. I detta element kan man välja att använda WebGL (Web Graphics Language), för att generera tredimensionell grafik. Eftersom WebGL använder datorns grafikkort för parallella beräkningar har det möjliggjort även ganska krävande renderingar i realtid (Angel & Shreiner, 2012). Alla stora moderna webbläsare har nu stöd för HTML5 och WebGL, även om en del begränsningar fortfarande finns avseende hårdvara. Det finns också WebGL-stöd i ett växande antal webbläsare för mobiler.

Precis som för dataformaten har olika 3D-lösningar för webben olika ursprung och olika styrkor. Många är helt generella, och har framförallt sitt ursprung i spelvärlden, även om användningsområdena de senaste åren vidgats. Det finns lösningar som är skrivna direkt i WebGL, men då detta är ett API på relativt låg nivå kräver det stor kunskap hos utvecklarna. För att underlätta finns flera specialiserade javascript-bibliotek, och även lösningar som baseras på spelmotorer (Evans et al., 2014). Eftersom dessa lösningar har tillgång till hårdvaru-accelerering från datorns grafikkort har de stora möjligheter till visuella effekter och animationer, men är i regel inte byggda med stöd för till exempel geografisk position eller analyser.

Andra lösningar är utvecklade speciellt för visualisering av byggnader eller stadsmodeller. CityPlanner från Agency 9 är ett sådant verktyg som används i en hel del svenska kommuner för att bygga interaktiva 3D-kartor online. Den tillåtna komplexiteten i modellerna är dock begränsad, och verktyget fungerar bäst på översiktsnivå. Det finns också ett antal webbapplikationer som är specialiserade på att läsa och visa upp BIM-modeller, i form av IFC eller i proprietära format, till exempel BIMSurfer och IFCWebViewer.

Utöver det finns lösningar som bygger på plug-in program men som fått sådan spridning att de ändå får ses som ganska allmängiltiga. Ett exempel är 3D-PDF, som kan läsas i nyare versioner av Adobe Reader, och därmed på miljontals datorer. Eftersom alla data måste paketeras i en enda fil och läsas in på en gång fungerar detta dock bäst för begränsade datamängder. En annan publiceringsväg för geografiska 3D-data online är via KML och Google Earth, som också har mycket stor spridning. Detta ger en möjlighet att georeferera objekt och

sätta dem i sitt geografiska sammanhang, men har en begränsad detaljeringsnivå och attributhantering. Dessutom är bakgrundskartan och 3D-byggnaderna som redan finns i Google Earth av mycket skiftande kvalitet och aktualitet.

Nedan följer en genomgång av de lösningar som ses som möjliga alternativ i just detta projekt. Det finns många fler som skulle kunna tas upp, men just dessa bedöms som genomförbara med tanke på svårighetsgraden och tillgången till programvara.

6.1 ESRI:s programsvit

ESRI:s ArcGIS-programsvit har de senaste åren byggt upp kapacitet för att både hantera 3D-data och visa upp den i webbklienter. Hörnstenarna i denna process är just nu CityEngine, ArcScene och ArcGIS Online. CityEngine är ett verktyg som i första hand bygger stadsmodeller. Programmet kan både importera färdiga 3D-modeller i till exempel OBJ- och DWG-format, eller generera dem proceduralt, det vill säga efter definierade regler, från 2D-kartor och ritningar. Programmet köptes upp av ESRI 2011, och har sedan dess delvis integrerats i ArcGIS-sviten (ESRI, 2011). ArcScene är i första hand ett visualiseringsverktyg för ArcGIS 3D-analysfunktion, men kan också behandla importerad 3D-data. Båda kan exportera tredimensionella scener i ett speciellt Web Scene-format (.3ws), som sedan kan publiceras via ArcGIS Online till en klient som heter CityEngine Web Viewer.

Web Scene-formatet är proprietärt och ingen fullständig dokumentation är tillgänglig, men det är känt att det utnyttjar WebGL för renderingen, och får med både geometri, texturer och attribut från modellen. CityEngine Web Viewer kan bland annat tända och släcka lager, visa två alternativa vyer parallellt, söka på attribut och visa scenen vid olika tidpunkter på dagen och året (ESRI, 2015a).

Nackdelen med lösningen är dels att den bygger på proprietär programvara, som kräver licenser och där den framtida utvecklingen är okänd och okontrollerbar, och dels så fungerar den i nuläget bäst för begränsade datamängder. Alla data läses in samtidigt, och det finns inget stöd för olika detaljeringsnivåer eller cacheade data.

Ett annat sätt att utnyttja ESRI-produkter är att bygga en scen i ArcGlobe, som liksom ArcScene är en del av Desktop-sviten och används för att visualisera 3D-analyser, fast på en större skala. Här finns möjlighet att bygga in regler för till exempel vilken data som visas på vilken inzoomnings-nivå. Scenen kan sen driftas som en *globe service* på ArcGIS Server. Problemet med denna lösning är att det inte finns någon färdig webbklient som kan konsumera dessa tjänster, utan de är i nuläget enbart användbara i andra ESRI-program (ESRI, 2015b).

I sammanhanget bör också nämnas att ett tredje ESRI-spår håller på att utvecklas, som sannolikt kan komma att bli en bra lösning framöver men som inte är så långt implementerat att det går att använda i detta projekt. Det handlar om att kunna skapa *scene services* i ArcGIS Pro, publicera dem till ArcGIS Server, och sedan konsumera dem i den nya Scene Viewern i ArcGIS Online. Till skillnad från City Engine Web Viewer arbetar denna med cacher och laddar upp data efterhand, och har därmed potential för större datamängder och mer avancerade

datalagringsstrukturer (ESRI, 2015c). Det finns också ett javascript-API under utveckling som kommer möjliggöra specialiserade 3D-webbapplikationer.

6.2 BIM-baserade visualiseringslösningar

Om ESRI:s program representerar en GIS-baserad lösning på visualiseringsproblemet, så finns det också möjliga lösningar med utgångspunkt i BIM-världen. En variant är ett verktyg som heter Maint3D, som används för visualisering av BIM-data i IFC-format. Detta är också ett proprietärt verktyg som kräver licens, och där de exakta stegen i arbetskedjan är okända. Dessutom fungerar det inte direkt i en webbläsare, utan kräver installation av en speciell plugin.

Den stora fördelen med lösningen är snabbheten och direktstödet för IFC, utan mellansteg. Programmet översätter IFC till ett egenutvecklat 3D-format (F3D) som komprimerar informationen och gör det möjligt att visa komplicerade modeller även över mobildatauppkopplingar. Klienten klarar av relativt avancerade visualiseringar, och kan till exempel markera hela system i byggnaden, skala av enskilda väggar och visa tvärsnitt (SVSI, 2015).

Det finns även andra onlinebaserade klienter för visualisering av i första hand BIM-data som skulle kunna vara aktuella. En kandidat är Autodesk 360, som är en molntjänst för delning av projektdata, inklusive tredimensionella modeller i en speciell webbaserad viewer. A360 kan importera data i flera viktiga BIM-format, och finns i en gratisversion. Den har inget direkt stöd för GIS-dataformat eller geografisk position.

6.3 X3D/X3DOM

X3D är en ISO-standard (ISO 19775) för tredimensionella scener och objekt, som används främst för visualiseringsändamål på internet. Det är en vidareutveckling av det tidigare VRML (Virtual Reality Modeling Language), och baseras på XML. Det är ett format som det forskats ganska mycket på, och det finns en hel del litteratur kring exempelvis visualisering av CityGML med hjälp av X3D (Mao et al., 2011; Prieto, 2012). Det finns dock en del som tyder på att formatet är populärare bland forskare än bland utvecklare i stort (Evans et al., 2012).

Modellen har en geospatial komponent och kan hantera data i olika LoD, vilket gör den väl lämpad för redovisning av geodata (Prieto, 2012). Den har också en speciell CAD-geometrikomponent. Formatet är en öppen standard, som utvecklas av ett internationellt konsortium av företag och forskare, i samarbete med bland annat ISO, OGC och W3C (Web3D Consortium, 2015).

En X3D-fil bygger upp en virtuell värld i form av en hierarkisk scenograf, bestående av noder som representerar objekt i modellen. Modellen hanterar bland annat ljus, animationer, texturer och användarinteraktioner. Geometrin kan representeras som geometriska grundformer i två eller tre dimensioner, till exempel cirklar, linjer och cylindrar. X3D har också stöd för höjdsatta ytor (*elevation grids*) och 3D-geometri definierad av ytor, *indexed face sets*.

I sig själv är inte X3D läsbart i webbläsare, utan kräver en plug-in. För att lösa detta, och skapa en bro mellan X3D och HTML, har X3DOM utvecklats. X3DOM är ett Javascript-baserat gränssnitt, som gör att X3D-scener kan renderas direkt i en HTML5-sida. Målsättningen på lång sikt är att göra X3D till en naturlig del av HTML-standarderna, som till exempel SVG blivit (Web3D Consortium, 2015).

6.4 Three.js

Ett mindre akademiskt utforskat men populärt format för 3D på nätet är three.js, eller 3JS som det också skrivs. Det är ett öppet javascript-bibliotek som gör att användaren slipper mycket av de programmeringsmässiga svårigheterna med att skriva WebGL. Tanken är att även relativt oerfarna utvecklare ska kunna skriva 3D-världar direkt i en HTML-sida.

Three.js arbetar med modeller i en egen JSON-specifikation, men kan också importera 3D-modeller som bland annat COLLADA och OBJ, eller skapa egna utifrån geometriska basobjekt. Det innehåller också möjligheter att ställa in ljus, material, och animationer. Det finns en LoD-nod i specifikationen, men den är inte implementerad i dagsläget. Det finns inte heller något direkt stöd för geografisk positionering (Three.js, 2015).

Den stora fördelen med att utveckla en applikation i three.js är att det är ett spritt och väldokumenterat format, med många tillgängliga exempel och aktiva användarforum.

6.5 Cesium

Cesium är också ett öppet javascript-bibliotek som byggts för att underlätta 3D direkt i webbläsaren, men med den skillnaden att det är specialiserat på geografiska data. Det betyder att det finns väl utbyggt stöd för geografisk positionering, import av kartlager från olika källor, och visualisering på en virtuell glob. Det finns också ett världsomspännande terrängdataset man kan använda, och stöd för data som förändras över tid. De geometrier man kan skapa är ganska lika de som finns tillgängliga i X3D och three.js, men möjligheterna att texturera, ljussätta och använda grafiska effekter är något mindre (Cesium, 2015).

Cesium kan importera 3D-modeller, men enbart som glTF (GL Transmission Format). glTF är ett relativt nytt format, som fortfarande är under utveckling och ännu inte finns i någon officiell rekommendation eller standard. Det utvecklas av COLLADA-gruppen inom Khronos, och är tänkt att komplettera COLLADA som presentationsformat, optimerat för användande med WebGL (Khronos group, 2015). Cesium har ett eget verktyg som konverterar COLLADA-modeller till glTF. Än så länge har formatet inget stöd för geolokalisering eller detaljnivåhantering.

6.6 Swecos interna 3D-motor

Sweco har internt utvecklat en egen 3D-motor för webbaserad visualisering, som kallas elements. Tanken är att den ska fungera som en plattform som kan skräddarsys till specifika projektapplikationer, och den används just nu bland annat för att visualisera en laserscannad modell av Lunds Domkyrka. Elements är skriven från grunden i javascript, och utnyttjar

WebGL-funktioner för att ladda in och visa upp data. Den hanterar shaders, animering och användarinput via ett enkelt GUI. I nuläget stöds import av modeller i OBJ- och JSON-format (Brand, 2015).

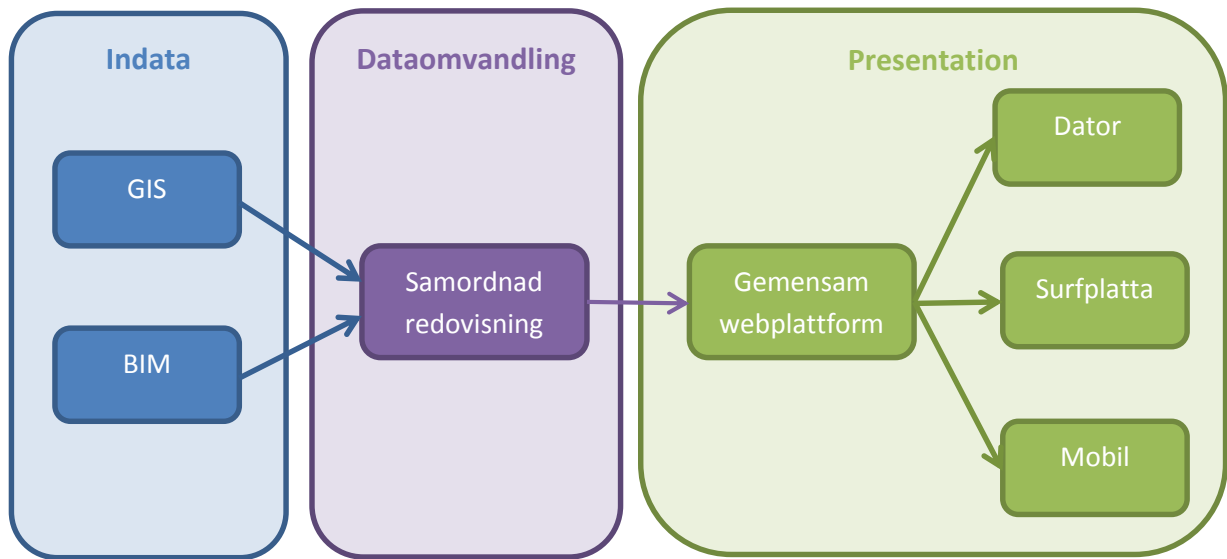
6.7 2D-presentation av 3D-data

Det kan också finnas situationer där det bästa sättet att visualisera 3D-data är i 2D. Detta kan till exempel vara om man vill kombinera 3D-data med 2D-lager i en existerande kartplattform, eller om man enkelt vill kunna plocka ut begripliga modellredovisningar i 2D från webbmiljön. Detta blir egentligen huvudsakligen ett kartografiskt problem, att kunna förenkla och symbolisera data så att samband och kategorier blir begripliga även i två dimensioner. Denna lösning skiljer sig en hel del från de övriga, och utreds inte i samma detalj eftersom redovisning av 2D-data online redan är ett jämförelsevis väletablerat och utrett område. Det finns också alldeles för många potentiella lösningar för att inom ramen för detta examensarbete utreda dem alla.

Den kanske viktigaste lösningen för att publicera tvådimensionella kartdata online har under senare år blivit med hjälp av servertjänster. OGC har publicerat ett antal standarder för att hämta data till webbaserade kartklienter från geografiska databaser via HTTP-anrop. Av stor betydelse i modern webb-GIS är WMS (*Web Map Service*), som hämtar specificerade lager inom specificerade koordinater och skickar tillbaka en bild av den önskade kartan. WMS hanterar enbart 2D-data (OGC, 2006). Om man vill ge användaren tillgång till grunddata, snarare än en bild av den, får man istället använda en WFS (*Web Feature Service*). I sammanhanget kan också nämnas att OGC även presenterat ett utkast till en standard för 3D-webbtjänster, W3DS (*Web 3D Service*). Det är en tredimensionell analogi till WMS, men i stället för en bild levereras en scen. Utkastet talar om att scenen ska kunna visas i webbläsare med plug-in, eller i virtuella jordglober, och det baseras alltså inte på WebGL (OGC, 2010). Utvecklingen verkar dock inte ha gått framåt sedan 2010, och det finns väldigt få vetenskapliga artiklar eller andra exempel som beskriver implementationer av W3DS.

7. Arkitektur och informationsflöde

I inledningskapitlet presenterades figur 7.1 som ett skissartat utkast till informationsflödet i det färdiga systemet. De olika formerna av indata har gått igenom i kapitel 4, dataomvandlingssteget har presenterats i kapitel 5 och olika versioner av presentationsmekanismer har redovisats i kapitel 6. Utifrån dessa diskussioner är det nu möjligt att definiera ett antal kriterier som den slutliga lösningen bör uppfylla. Olika möjliga arkitekturer jämförs sen med dessa kriterier för att komma fram till en genomförbar lösning.



Figur 7.1 Tänkt informationsflöde.

7.1 Kriterier

Den grundläggande målsättningen med det här projektet är att hitta en lösning som faktiskt fungerar, och alla andra egenskaper är i någon mån förhandlingsbara. För att utgöra ett underlag för valet av implementering har dock ett antal önskvärda egenskaper hos ett samvisualiseringssystem satts upp. Med tanke på den snabba teknikutvecklingen inom området är det mycket sannolikt att förutsättningarna ser helt annorlunda ut om bara något år. Värdet i den här genomgången kommer då snarare ligga i identifierade problemområden och förbättringsförslag än i en specifik föreslagen lösning. Kriterierna hålls med avsikt på en generell nivå, och är tänkta att kunna användas som grund för mer specificerade och kontextberoende krav i ett enskilt projekt.

Ur indatakapitlet (kapitel 4) framgår att både BIM- och GIS-världen är väldigt heterogena med många aktörer och många dataformat. I BIM-världen finns IFC som en växande standard, men den omfattas ännu inte av alla och täcker inte in alla användningsområden. I GIS-världen är det ännu svårt att tala om någon 3D-standard, och olika format används i olika applikationer. Det är också här viktigt att ta med sig att utvecklingen just nu går väldigt snabbt, och att vilka

format som faktiskt används kan förändras drastiskt på bara ett par år. Med detta i åtanke ställs följande kriterier upp:

K1: Systemet ska kunna hantera indata från många olika källor i många olika format.

K2: Systemet ska vara enkelt att bygga ut med fler format och standarder i framtiden, och bygga på delar som med stor sannolikhet kommer följa med i utvecklingen.

Dessa kriterier innebär också att alla steg i processen måste vara så pass öppna och väldokumenterade att det är möjligt att justera dem, både för att hantera olika filformat och för de olika variationer inom samma filformat som med all säkerhet kommer att uppstå mellan olika datamängder.

Vad gäller dataomvandling gäller i stor utsträckning att slutformatet blir beroende av vilket slutredovisningssteg som väljs. Det som är viktigt i processen är att data kan samordnas med avseende på till exempel detaljnivå och geografisk position, vilket oavsett slutformat med stor sannolikhet kommer kräva en del bearbetning. Det är också viktigt att data i mellanstegen behandlas i format som kan bibehålla tillräcklig geometrisk precision och hålla kvar kopplingen till sina attribut. Vad som är rätt nivå på detaljer och attributmängder kommer vara beroende av indata och användning, och måste därför vara möjligt att justera.

K3: Systemet ska kunna samordna data och lämna till valt visualiseringsformat med lämplig detaljnivå, korrekt positionering och bibehållna attribut.

En mer långtgående sammanslagning av data som även tillåter analyser skulle även kräva att attributdata var semantiskt och formmässigt harmoniserade. Eftersom detta arbete enligt avgränsningarna i avsnitt 1.3 bara berör själva visualiseringen behandlas inte sådana frågor vidare.

Vad gäller visualiseringssteget så har undersökningen fokuserat på webbaserade, HTML5-drivna lösningar. Det finns andra vägar att gå, men det är svårt att hitta något annat alternativ som tillhandahåller samma spridning och lättillgänglighet över olika plattformar. Även om det självklart är fördelaktigt så ställs inga explicita krav på hastighet, lagring eller optimerad datahantering, i enlighet med de generella avgränsningarna i inledningen. Det kommer inte heller finnas några explicita avgränsningar kring vilka funktionaliteter som ska vara inbyggda, eftersom svårigheten att implementera dessa kommer att vara mycket beroende av vilket alternativ som väljs. Fokus ligger på följande kriterier:

K4: Systemet ska visualisera samordnade data på ett enkelt och användarvänligt vis.

K5: Systemet ska inte kräva installation av programvara utöver en webbläsare.

K6: Systemet ska kunna användas på såväl dator som surfplatta som mobil.

Det är också viktigt att tillstå att en del av kriterierna för den slutliga lösningen redan implicit byggts in i urvalet av de alternativ som diskuterats. Till exempel har de tidigare kapitlen bara

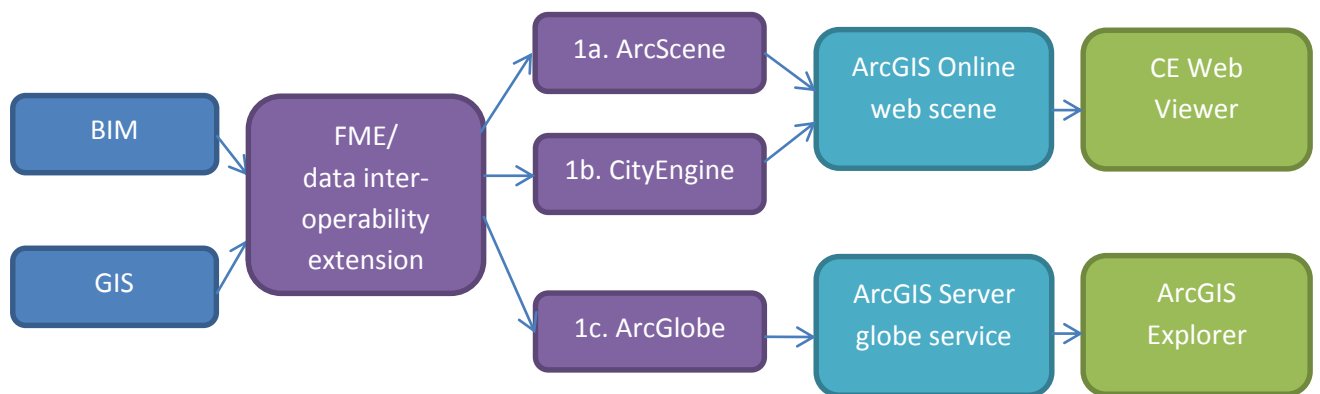
tagit upp lösningar som baseras antingen på öppen programvara eller på program som Sweco har och kommer fortsätta ha licenser på av andra skäl. Det kan naturligtvis finnas andra lösningar som på det viset helt fallit bort.

7.2 Arkitekturförslag

Arkitekturförslagen i detta avsnitt har grupperats efter det sista steget, det vill säga visualiseringslösningen, eftersom denna i stor utsträckning påverkar hur resten ser ut. För varje underalternativ görs en snabb genomgång av för- och nackdelar, och en jämförelse med de kriterier som definierats i avsnitt 7.1 ovan.

7.2.1 ESRI-baserad lösning

Den första gruppen av lösningar är de som baseras på ESRI-produkter. Här kan FME-steget ersättas av en *data interoperability extension* till ArcGIS, som bygger på samma mjukvara. En principiell illustration finns i figur 7.2.



Figur 7.2. Arkitekturförslag 1a-1c.

Fördelen med den här gruppen av lösningar är framförallt att mycket funktionalitet redan finns implementerad, och uppgiften handlar här mest om att konfigurera parametrar och justera hur indata hanteras. Attributhantering stöds hela vägen så länge objekten har attribut med in. I likhet med de flesta GIS-program stöds dock huvudsakligen ganska enkla geometrityper.

De första alternativen handlar om att skapa ett web scene-dokument som kan publiceras via ArcGIS Online och visas i CityEngine Web Viewer. Detta kan göras antingen från ArcScene (alternativ 1a), eller från CityEngine (1b). CityEngine kan direktimportera något fler format och har en del utseendeparametrar som ArcScene saknar, medan ArcScene har betydligt fler analys- och omvandlingsverktyg. I just detta sammanhang är skillnaden mellan programmen inte speciellt stor, även om de från början är ganska olika programvaror.

CityEngine Web Viewer har en hel del funktionalitet och en inbyggd attributhantering som gör det lätt att till exempel söka i modellen. Den är däremot inte byggd för att hantera några större datamängder. Hela datamängden laddas in när ett scen-dokument öppnas, och dokumentationen rekommenderar att .3ws-dokumentet begränsas till 15 MB. Programmet

fungerar i nuläget inte heller på mobiltelefoner. Om man av någon anledning inte vill använda ArcGIS Online som bas finns också möjlighet att installera och använda CityEngine Web Viewer på en egen server.

Det andra i nuläget fungerande ESRI-spåret är 1c, att via ArcGlobe skapa en scen som man sedan kan publicera till ArcGIS Server som en *globe service*. Ett globe service-dokument innehåller datalagrings- och visualiseringsstrukturer som till exempel gör det möjligt att visa olika detaljnivå vid olika inzoomning och ladda in data efter hand från servern snarare än allt på en gång. Nackdelen här ligger på klientsidan, eftersom en globe service bara kan konsumeras av andra ESRI-produkter. Det finns en gratis viewer, ArcGIS Explorer, som man kan ladda ner, men i dagsläget är det bara desktopversionen för Windows som kan hantera 3D. Det finns även en möjlighet att via ArcGIS Engine bygga en egen applikation som konsumerar globe services.

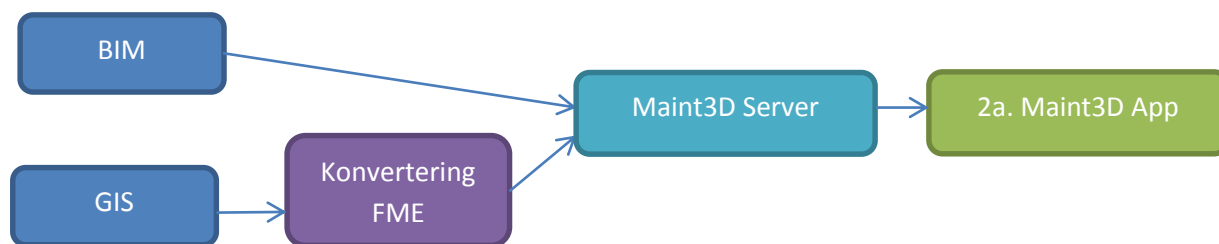
Eftersom lösningen med *scene services* inte ännu finns tillgänglig utvärderas den inte vidare i detta projekt. Den är dock sannolikt en stark kandidat för framtiden.

Alla alternativ i denna kategori uppfyller mer eller mindre de första två kriterierna, så länge det finns en framkomlig väg från det aktuella BIM-formatet till FME. Det tredje kriteriet är mycket beroende av indata, men eftersom attributdatahanteringen är inbyggd som en naturlig del av GIS-världen är det antagligen här dessa arkitekturförslag har fördelar över de andra. Skillnaden mellan de olika underalternativen är inte heller särskilt stor vad gäller datamanipulationen, utom i vissa utseendehänseenden. CityEngine Web Viewer är kompetent och lättanvänd och kräver inga installationer, men funkar bäst för mindre dataset. Globe service-lösningen kräver tillgång till en ESRI-baserad produkt för uppvisning, och möter därmed inte kriterium 5. Det finns ännu ingen av ESRI-lösningarna som täcker in mobiler och surfplattor, även om 3D-stöd för ArcGIS explorer i mobilen sägs vara planerat, så det sjätte kriteriet möts inte.

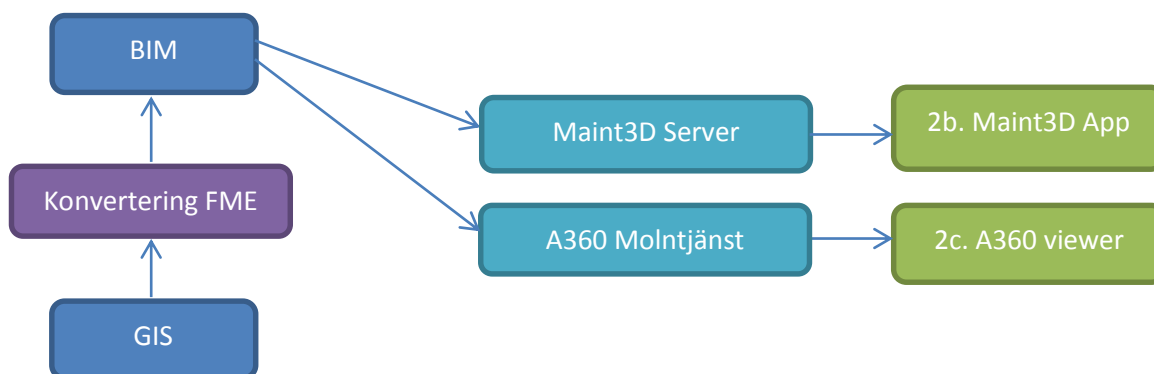
7.2.2 BIM-verktygslösning

Den andra gruppen lösningar handlar om att föra in GIS-data i produkter som från början designats för att visa upp BIM-modeller. Den översiktliga arkitekturen illustreras i figur 7.3 och 7.4. De flesta BIM-verktyg har inte stöd för direktimport av GIS-dataformat, så indata måste först behandlas och omvandlas till format som går att läsa in. Eftersom GIS-program än så länge har ganska begränsade alternativ för tredimensionella exportformat är det sannolikt att viss databehandling i FME kommer att vara nödvändig som mellansteg.

Databehandlingsteget kompliceras av att data i en BIM-modell normalt sett ligger i ett eget koordinatsystem, och det kan vara svårt att placera GIS-data rätt. Dataformaten är också grovt sett antingen rent geometriska, eller så bygger de på väldigt specialiserade klassificeringar. I det första fallet tappar man lätt viktiga data, och i det andra fallet kan det vara svårt att hitta meningsfulla översättningar.



Figur 7.3. Arkitekturförslag 2a.



Figur 7.4. Arkitekturförslag 2b-2c.

Arkitekturförslag 2a bygger på Maint3D. Fördelen är viewern, som har bra funktioner för att granska modeller och attributdata på olika vis. Lösningen med filsystem, server och det komprimerade dataformatet ger också en bra prestanda. Nackdelen är kanske främst att programmet importerar modeller i mycket få format, där egentligen bara IFC är attributbärande, vilket gör det svårt att få in rika GIS-data. Programmet är dessutom proprietärt och har licenser knutna till enskilda användare vilket gör det dyrt i större projekt. Man behöver också ladda ner en plug-in för att granska modeller i webbläsaren.

2b bygger också på Maint 3D som visualiseringssteg, men med skillnaden att GIS-data först tagits in i själva BIM-modellen, innan den exporteras, snarare än att laddas rakt in i viewern. Detta kan vara en fördel då programmet är optimerat för exporter från BIM-program.

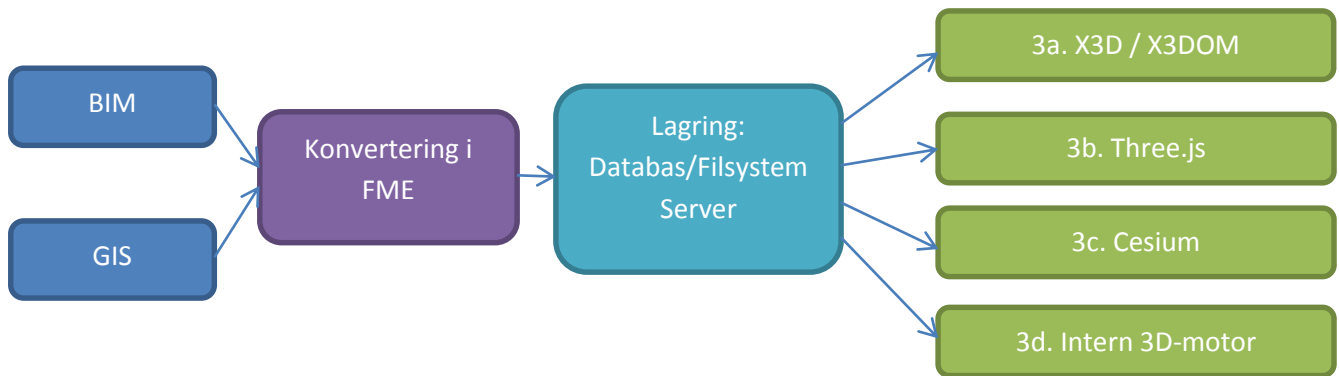
2c handlar om att redovisa i Autodesk 360. Material från olika källor måste först kombineras till en enda modell, som sen kan laddas upp i ett projekt i molnlösningen. Även här är fördelen en bra viewer och fungerande datalagringslösning. Nackdelen är bland annat att gratisversionen bara kan lagra en begränsad mängd data.

I förhållande till kriterierna kan man säga att kompetensen i de här lösningarna för att hantera BIM-data är väldigt stor, men att det finns en hel del frågetecken kring GIS-data. Databehandlingen kommer i stor utsträckning behöva lösas innan data läses in, och det finns ingen uppenbar kandidat till utbytesformat, vilket gör att de tre första kriterierna är tveksamt mötta. Båda de program som diskuterats har bra användargränssnitt och lagringslösningar,

och båda har dessutom tillhörande mobilappar. Maint3D kräver nedladdning av en plug-in vid första användningen, vilket motsäger kriterium 5.

7.2.3 Egen visualiseringslösning

Denna grupp av lösningar baseras på konvertering i FME och någon form av egenutvecklad viewer i slutsteget, och illustreras översiktligt i figur 7.5.



Figur 7.5 Arkitekturförslag 3a-3d.

Indata behandlas i FME, antingen automatiserat via FME server, eller om det visar sig omöjligt att standardisera processen med tanke på de stora variationerna i indata, manuellt i ett förberett men anpassningsbart FME workspace.

Presentationsformaten som accepteras skiljer sig något från alternativ till alternativ, men generellt rör det sig om ganska enkla geometriformat (X3D, COLLADA, OBJ. Se kapitel 6). Det innebär att attributen behöver kopplas till rätt objekt inom ramen för visualiseringsverktyget, vilket också kan vara svårt att automatisera och kan kräva viss anpassning under datahanteringssteget.

Mellan databehandling och visualisering finns ett lagringssteg, som också kan se ut på ett par olika sätt. Även här är FME Server ett alternativ, där kommunikation med klienten kan ske via till exempel HTTP-anrop eller websockets. Det finns även flera olika varianter av databas/server-kombinationer som har olika fördelar i termer av kapacitet, flexibilitet och snabbhet. Lagring och klientkommunikation av spatiala data är ett stort område i sig, och här finns många aspekter som inte kommer behandlas i detta examensarbete.

För själva klienten behandlades fyra huvudsakliga alternativ i kapitel 6. Alla har sina för- och nackdelar. Gemensamt har de fördelen att man genom att sköta utvecklingen själv har kontroll över riktning och framtid och själv kan avgöra vilka utökningar och förbättringar som behövs. Motsvarande nackdel är att man inte utnyttjar det utvecklingsarbete andra redan lagt ner på att lösa de specifika problem som uppstår. Speciellt kan det med de tillgängliga verktygen visa sig mycket svårt att skriva något själv som kan hantera stora datamängder med acceptabel prestanda.

Alternativet med kombinationen X3D/X3DOM har fördelen att det bygger på en officiell standard, och har en del inbyggda metoder för att hantera geografiska data. Nackdelen är den relativa bristen på exempel och utvecklarstöd.

Three.js har till viss del det omvända problemet: det är ett open source-projekt med en enda huvudsaklig författare, och därmed i ständig förändring och med en svårbedömd framtid. Däremot har det stor spridning, med många exempel och aktiva användarforum.

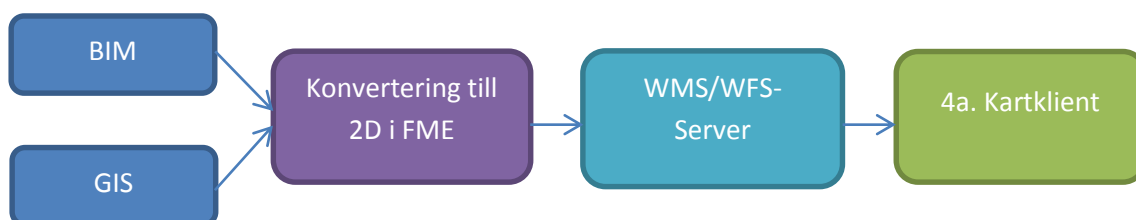
Cesiumbiblioteket är specialiserat på geografiska data, och har därför en del användbara funktionaliteter. Däremot brister biblioteket något i vilka format som kan importeras, och vad man kan göra med modellerna.

Den internt utvecklade 3D-motorn är möjligen den mest flexibla lösningen av alla, eftersom den bygger på ren WebGL, och det finns en direkt dialog med den ursprungliga utvecklaren. Däremot är tillgången på exempel och dokumentation väldigt begränsad.

Vad gäller kriterierna har dessa lösningar samma möjligheter för indata som i avsnitt 7.2.1. ovan, och kan därmed sägas täcka de första två kriterierna rimligt väl. Det tredje kriteriet är mer svårbedömt. Korrekt geografisk placering och geometrisk detaljnivå kommer att handla mycket om kvalitet och omfattning på indata och är svårt att säga något generellt om. Attributhantering kommer inte följa automatiskt med dataformaten i något fall och kommer att kräva en speciallösning. Inget av alternativen kräver plug-in eller specialprogram, och på en tillräckligt modern mobiltelefon eller surfplatta med uppdaterad webläsare kommer tillgängligheten på olika plattformar vara tillgodosedd, vilket gör att kriterium 5 och 6 täcks. Det är däremot omöjligt att i detta skede säga hur långt man inom ramen för detta projekt kommer hinna komma med användarvänlighet och smarta verktyg, vilket gör kriterium 4 osäkert.

7.2.4 2D-lösning

För att omvandla en 3D-modell till en 2D-karttjänst följs den huvudsakliga arkitekturen i figur 7.6 nedan. Data konverteras i FME till en 2D-karta som kan publiceras med hjälp av en OGC-karttjänst.



Figur 7.6 Arkitekturförslag 4a.

Lösningen skiljer sig som tidigare nämnts en hel del från de övriga, men kan ändå huvudsakligen bedömas efter samma kriterier. Vad gäller indata och formathantering kommer möjligheterna inte skilja sig nämnvärt från de som presenterats ovan. Dataomvandlingssteget

kommer att kräva en del manuella justeringar och anpassning efter den tänkta användningen för att göra resultatet läsbart. Slutstegen kommer också vara oerhört beroende av den tänkta användningen och användaren, och här finns väldigt många alternativ. Generellt kan dock sägas att de tre sista kriterierna uppfylls, eftersom OGC-tjänster hanteras direkt i webbläsaren och fungerar även på mobila plattformar. Det finns också många tillgängliga kartklienter som ger tillgång till bra användarvänliga funktionaliteter.

8. Fallstudie

De föreslagna arkitekturerna utvärderas med hjälp av en fallstudie från pappersindustrin. Denna kan ses som representativ för det typfall som presenterades i avsnitt 2.2, det vill säga stora anläggningar. Målsättningen är att med utgångspunkt i ett riktigt användarfall med riktiga data testa de teoretiska upplägg som presenterats i kapitel 7, och dokumentera uppkomna problem och lärdomar.

8.1 Bakgrund

Skogs- och massabolaget Södra håller på att bygga om sitt pappersmassabruk i Värö utanför Varberg. I ombyggt skick ska anläggningen kunna producera 700 000 ton pappersmassa om året, plus el och fjärrvärme, och blir därmed en av världens största och modernaste massaanläggningar (Södra, 2014). Projekteringen av bygget har utförts av Sweco, och har bland annat resulterat i en komplett 3D-modell över bruket.

Modellen är uppbyggd i en programvara som heter PDMS (Plant Design Management System), från företaget Aveva. Det är ett verktyg för modellering av fabriker och anläggningar snarare än byggnader per se, och används i stor utsträckning inom den industriella sektorn. Även om det inte brukar räknas upp som ett av de typiska BIM-programmen, så delar det många av deras egenskaper: det bygger på objektorienterad 3D-modellering där attribut knyts till objekten, med automatisk kollisionskontroll och rapportgenerering. PDMS-modellen i sig innehåller ingen förvaltnings- eller kostnadsinformation, men den kan knytas till andra programvaror från samma tillverkare som gör det.

Södra är, som många skogsbolag, redan en aktiv GIS-användare. De har en egen onlinebaserad GIS-tjänst som används för förvaltning av skogsbeståndet, transportplanering, med mera. Frågan uppstår därför om och hur man kan kombinera den nya BIM-modellen av bruket med befintliga GIS-data, för att uppnå nya användarnyttor i till exempel förvaltning och underhåll. Det är också intressant att undersöka hur 3D-data kan användas inom organisationen.

8.2 Data

Data från PDMS-modellen exporteras i första hand i ett filformat som heter RVM, som är proprietärt och maskinläsbart format kopplat direkt till programvaran. Det är även möjligt att exportera geometrin som DXF, det vill säga i AutoCAD-format. Attribut exporteras parallellt, i en vanlig textfil med samma namn, eller som Excel-fil. Geometrierna är från början modellerade med geometriska primitiver, en form av CSG (Karlsson, 2015). RVM-exporten bibehåller denna struktur, medan DXF-exporten approximerar kropparna med ytor och omvandlar modellen till ett mesh. Till skillnad från till exempel IFC-modeller är inte varje enskild byggnadsdel modellerad som ett enskilt objekt. Istället läggs de ihop i funktionella grupper: en grupp kan vara en tank, en annan alla stegar, en tredje en hel huskropp, med väggar och golv och tak. Attributen knyts huvudsakligen till dessa grupper, snarare än till enskilda geometrier. Modellen ligger i ett lokalt koordinatsystem, och y-axeln är vriden i förhållande till geografiskt norr.

I detta examensarbete har bara delar av 3D-modellen använts i de olika implementeringarna, för hanterlighetens skull. Framförallt har ett dataset som innehåller byggnaderna, och ett som innehåller ett antal tankar med utförligare attribut använts.

Från GIS-datavärlden finns ett utdrag ur lantmäteriets höjddatamodell för det aktuella området, levererat som ett georefererat raster i TIFF-format. Detta kan kombineras med ett ortofoto med samma utsträckning för att skapa en tredimensionell omgivning.

Utöver dessa har en del andra kompletterande dataset använts, i jämförelsesyfte och för att testa användbarhet för andra områden och typanvändningar. Till exempel har en komplett IFC-modell av en skola, och en geologisk modell med markyta, grundvattenyta och två geologiska ytor i DWG-format testats.

8.2.1 Dataimport från PDMS

Nästan alla de föreslagna arkitekturerna i avsnitt 7.2 förutsätter att data kan importeras till FME för att behandlas och översättas till lämpligt mellanstegsformat. Detta har visat sig innebära en hel del svårigheter.

RVM-filerna med byggnadsdata från Värö som exporteras från PDMS kan läsas in i NavisWorks, som är Autodesk's program för samredovisning och kollisionshantering. NavisWorks läser också automatiskt in .txt-filen med attribut och tilldelar dessa till rätt objekt, vilket är en stor fördel. Detta är också den arbetsgång som används och är invand bland projektörerna, och den bevarar den ingående geometrin väldigt väl. Dock är exportmöjligheterna från NavisWorks mycket begränsade. Detta är gjort med avsikt: meningen är att kollisioner och problem som upptäcks i NavisWorks ska korrigeras i grundmodellen som sen importeras på nytt, för att undvika versionsproblem och dylikt.

De tre exportformat som finns tillgängliga är DWF, FBX och KML. Alla tre bevarar objektstrukturen, det vill säga läser in hela hierarkin, med skilda geometrier som skilda objekt.

- DWF (Design Web Format) är ett komprimerat redovisningsformat som är optimerat för webbredovisning. Det bevarar både geometri och attribut, men eftersom det är designat som slutredovisningsformat finns det väldigt få program som kan läsa det utanför Autodesk-världen. FME kan läsa DWF, men bara i två dimensioner, och alla 3D-modeller läses in förenklade till en enda 2D-punkt i origo.
- FBX (från FilmBoX) är ett rent grafikformat som används i modelleringsprogram, spelmotorer etc. Det är också proprietärt och ägt av Autodesk. Det bevarar geometri och även avancerade utseendeparametrar, men bevarar inte modellens attribut och går inte att läsa in i FME.
- KML (se avsnitt 4.1.3) har fördelen att modellen kan geolokaliseras, antingen manuellt eller med hjälp av ett importerat platsmärke. Det kan också läsas in i FME. Däremot bevaras inga attribut i importen. Formatet får också med sig ett par i sammanhanget onödiga geometrier, som till exempel utsiktspunkt, som komplicerar hanteringen, filerna blir mycket stora, och höjderna blir inte korrekt överförda.

Eftersom inga av exportalternativen från NavisWorks gav tillfredställande resultat har istället DXF-exporter direkt från PDMS använts som indata. Dessa har också inneburit vissa problem, och det har krävts en hel del arbete för att hitta en struktur på exportfilerna som gör att de går att använda vidare. DXF-filen i sig innehåller ingen attributinformation alls, utan enbart geometri, uppdelad i lager och block. För att i FME kunna dela upp objekten och knyta rätt attribut till dem, behöver varje objekt antingen ligga i ett eget lager, eller i ett block som inte innehåller andra block (FME sparar bara den lägsta blocktillhörigheten i en hierarki). Lager- eller blocknamnet måste gå att koppla till objektnamnet i Excel-filen med attribut. Det ska också noteras att omvandlingen av geometrin från solida kroppar till mesh medför minskad precision och en del problem med validering senare i processen (se till exempel avsnitt 8.6.2).

Geografiska koordinater för modellens origo ingår inte i exporten, och har visat sig svåra att få tag på även senare. Detsamma gäller för modellens vridning gentemot geografiskt norr. Den geografiska positioneringen i det här projektet har istället fått utgå från att byggnader med igenkännbar form passats in över ortofotot. Detta ger en precision i placeringen som inte alls motsvarar precisionen i modellen.

8.4 Kriterieanalys och upplägg

Målet med fallstudien är att skapa ett proof of concept, det vill säga att visa vad som är möjligt med ett par olika verktyg. Detta är tänkt att användas som underlag till en diskussion kring vilka konkreta användarfall som kan finnas i framtiden. Med tanke på detta är det svårt att specificera kriterierna från avsnitt 7.1 speciellt mycket. Det första kriteriet kan naturligtvis specificeras till att verktyget ska kunna hantera indata från PDMS och ortofotodraperade höjddata. Det tredje kan delvis förtydligas med att detaljnivån ska vara så hög som de valda redovisningsstegen klarar av, och att alla attribut som kommer in ska bibehållas, eftersom det ännu är oklart exakt vilka som behövs. Vad gäller de sista tre kriterierna är deras relativa betydelse väldigt avhängig slutanvändningen. Ett verktyg som ska användas för aktiv förvaltning i fält måste självklart fungera på mobil och platta, men installationen av en app är där oftast inte någon stor sak. I en desktopmiljö där systemadministratören måste vara delaktig i varje programinstallation gäller kanske det motsatta. Därför gäller i detta skede att alla kriterier behöver beaktas och kommenteras.

Det finns inte utrymme inom ramen för detta examensarbete att prova alla de föreslagna arkitekturerna i avsnitt 7.2. Ett alternativ i varje huvudspår har valts ut för att testas, inklusive en 2D-lösning som framkommit som önskemål specifikt i fallstudien.

Alternativ 1. Det första spåret här handlar om att föra över BIM-data i en tredimensionell GIS-värld. Här används höjddata och ortofoto för att bygga upp en omgivning, och själva BIM-modellen förenklas så långt det är nödvändigt för att komma ner i en acceptabel datamängd. Huvudsakligen kommer arkitekturförslag 1a att följas, det vill säga via ArcScene till ArcGIS Online och City Engine Web Viewer. Detta förslag täcker kriterierna 1,2,3,4 och 5 lika bra som eller bättre än alternativen i denna kategori, och kriterium 6 missas av samtliga. Jämfört med

arkitekturförslag 1c bedöms den enkla tillgängligheten över internet överväga nackdelen med den begränsade datamängden.

Alternativ 2. En andra väg är att få in geografiska tredimensionella data i en BIM-viewer. Här ska så mycket av originaldata som möjligt bevaras, och modellen placeras in i en omgivning som omvandlats för att passa koordinatsystem och orientering. Denna lösning baseras på arkitekturförslag 2a. Kriterium 1,2 och 3 är lika svårbedömda som för alternativen. Kriterium 4 och 6 möts väl, då Maint 3D innehåller den troligen bästa av de viewers som undersökts, och har hög kapacitet och bra tillgänglighet för mobil och platta. Dessa fördelar bedöms här väga upp nackdelen med kravet på plug-in, som motsäger kriterium 5.

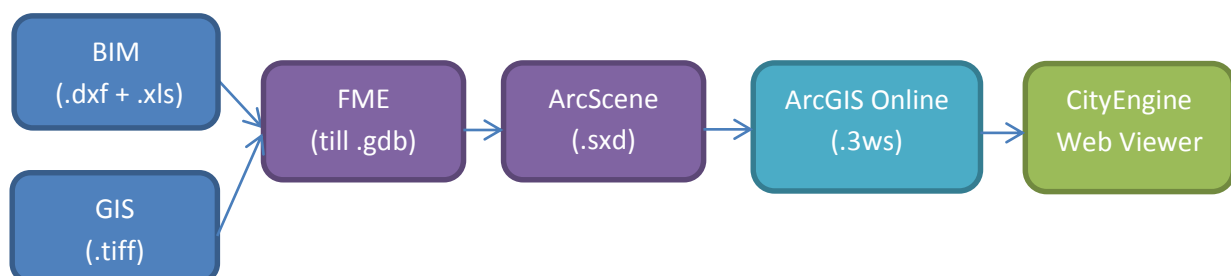
Alternativ 3. Ett försök kommer också att göras med att skapa en egen enkel WebGL- och JavaScript-baserad visualiseringsmiljö, för att testa kapaciteten och ytterligare identifiera och undersöka problem och möjligheter. Eftersom FME i sin senaste version har infört stöd för att skriva direkt till three.js-JSON, och det formatet är så spritt, baseras denna lösning på arkitekturförslag 3b. Kriterieuppfyllnaden bedöms vara ganska likvärdig mellan alternativen i denna kategori, där kriterium 1, 2, 5 och 6 möts förhållandevis väl, och kriterium 3 och 4 är svårbedömda.

Alternativ 4. Till sist finns ett önskemål om att kombinera data från BIM-modellen direkt med den GIS-tjänst som Södra redan har på plats. Eftersom denna bygger på WMS-standard, och WMS i dagsläget bara hanterar 2D-geometrier, så handlar det om att med bevarade attribut förenkla och platta till 3D-geometrin, positionera den geografiskt och omvandla den till ett format som kan publiceras som en karttjänst på ArcGIS Server. Denna lösning har efterfrågats eftersom infrastrukturen för att stödja den redan finns på plats och används. Den uppfyller också samtliga kriterier.

8.5 Alternativ 1: CityEngine Web Viewer

8.5.1 Implementering

Den GIS-baserade lösningen som valts följer flödet i figur 8.1.



Figur 8.1. Implementering av omvandling till CityEngine Web Scene.

BIM-modellen läses in, delas upp i objekt, och får sina attribut från Excel-filen. Detta kompliceras något av olika filstrukturer på indata och det faktum att namnen i attributtabellen bara är nästan identiska med objektsnamnen på geometrin. Modellen måste skalas om från

millimeter till meter, roteras, och placeras rätt. Sedan definieras det lokala koordinatsystemet, och transformeras till SWEREF99, eftersom det är i det systemet höjdmodellen och ortofotot läses in. Hela modellen, inklusive höjdmodellen, flyttas sedan till ett lokalt koordinatsystem. Detta är nödvändigt för att få tillräcklig geometrisk precision i koordinaterna i den slutliga modellen (se avsnitt 8.9). Modellen färgläggs, för att delarna ska vara lättare att urskilja. Modellgeometrin skrivs sedan ut till en ESRI geodatabas-fil, som 3D-ytor (*multipatch*). Det FME-workspace som använts finns i bilaga 1.

Det finns ett par olika sätt att behandla höjdmodellen. Bäst visuellt resultat får man om både höjdmodellen och ortofotot läses in i FME, transformeras till det lokala koordinatsystemet och sedan skrivs ut till TIFF igen. Dessa båda raster kan sedan läsas in i ArcScene, och i ortofotots *properties*-dialog kan man sätta höjdmodellen som bashöjd för fotot. Något kantigare, men mindre utrymmeskrävande, är det om man i FME bygger en TIN-yta och draperar rastret på den, och sedan skriver TIN-filen direkt till geodatabasen med den övriga geometrin. Detta kräver att man skriver till geodatabas snarare än shape-filer, eftersom den exporten inte har stöd för texturer.

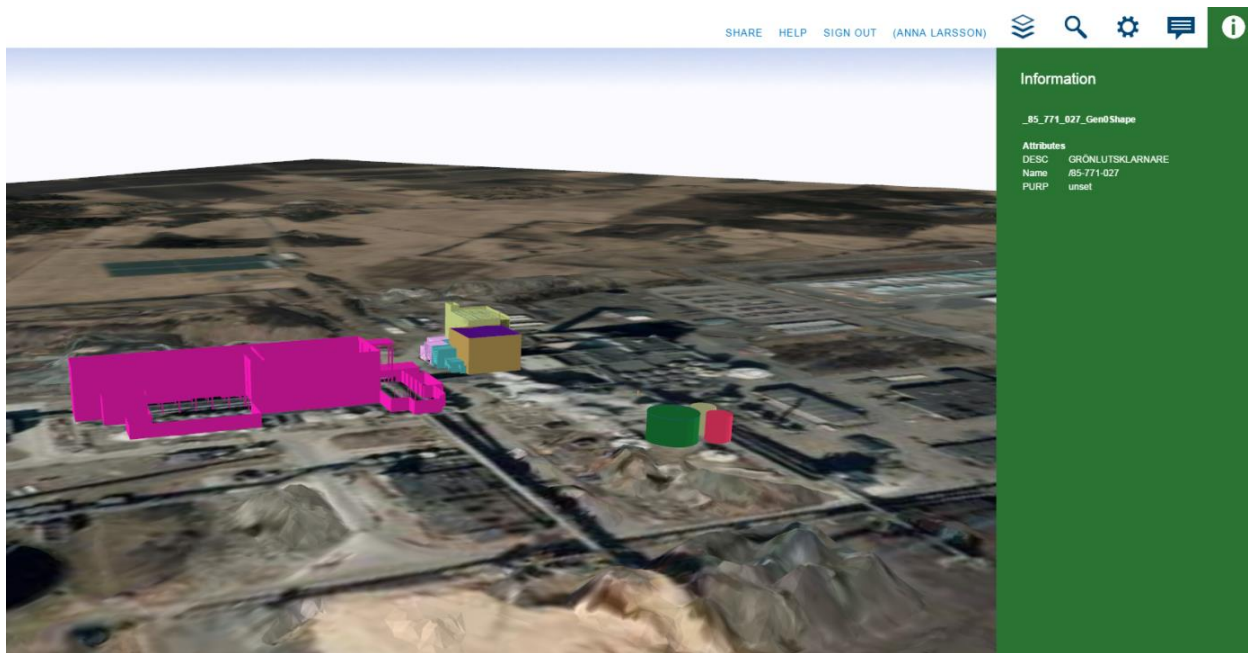
I ArcScene öppnas en ny tom scen, och filerna från geodatabasen läggs till. Om markmodellen är importerad som TIN behöver i princip inget göras, mer än att spara filen. Sedan öppnar man *ArcToolbox -> 3D Analyst Tools -> CityEngine -> Export to 3D Web Scene*, anger vilken scen som ska exporteras och var den nya filen ska läggas. I ArcGIS Online kan man sedan ladda upp den resulterande .3ws-filen, och får alternativet att öppna den med CityEngine Web Viewer (menyvalet *view application*).

8.5.2 Resultat

Figur 8.2 nedan visar ett utsnitt från CityEngine Web Viewer, med en av tankarna markerad. Som synes finns både byggnader, markmodell och ortofoto med. Alla ligger som olika lager, vilket innebär att man kan dölja och plocka fram olika delar, och även söka på attribut.

Den stora bristen med den här lösningen är att den möjliga datamängden är begränsad. Den .3ws-filen som visas ovan är på drygt 11 MB, komprimerad från en geodatabas på 19 MB. En liknande fil på 13 MB får viewern att krascha ungefär varannan gång, och ytterligare en på 17 MB går inte överhuvudtaget att öppna. Detta begränsar användningsområdet. Hela 3D-modellen över Värö Bruk går till exempel omöjligt att få in.

Modellen flimrar också en hel del när man navigerar runt i den, vilket är visuellt något störande.

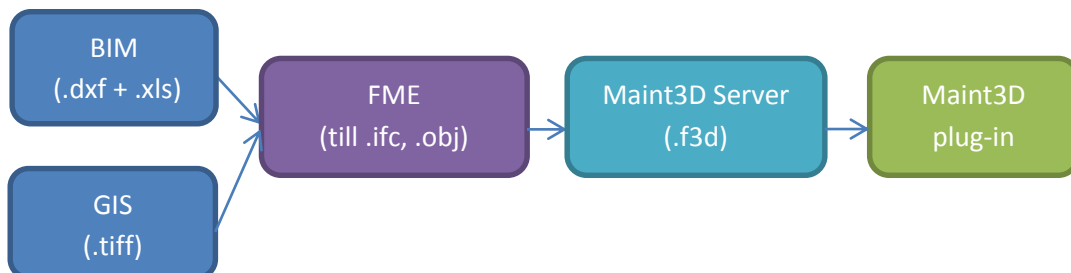


Figur 8.2 Modell i CityEngine Web Viewer.

8.6 Alternativ 2: Maint 3D

8.6.1 Implementering

Den BIM-baserade lösningen som valts följer principiellt flödet i figur 8.3.



Figur 8.3. Implementering av import till Maint3D.

De första stegen är i princip identiska med dem i alternativet ovan: data importerats, behandlas och koordinatsätts på samma sätt. Höjdmodellen byggs upp som TIN med draperat raster. För att kunna skriva till IFC kräver FME att geometrin är solid snarare än mesh, så modellen definieras om till en b-rep. IFC-exporten i sig är något invecklad, och kräver ganska många steg. Attribut måste sättas på geometrier snarare än objekt, och geometriernas namn måste stämma överens med det som heter *representation identifier* i IFC, i detta fall "body". För att få med attribut måste man först definiera *property sets*, det vill säga bestämma vilka attribut som ska ingå och vilken datatyp de har, vilket skrivs till IFC-filen separat. Sedan måste man skapa dessa property sets, det vill säga fylla dem med data från attribut, och koppla dem till geometrierna. Det workspace som använts finns i bilaga 1.

Även här finns minst två olika sätt att behandla markmodellen. Man kan skriva ut den separat, till exempel som en OBJ-fil. Eftersom OBJ definierar vilken axel som pekar uppåt annorlunda än IFC, måste man då rotera den -90 grader runt x-axeln först. Ett annat alternativ är att ta in markmodellen som en del av IFC-filen, och knyta geometrin till *IfcSite*-noden.

De exporterade filerna laddas upp direkt till Maint3D Server. Det krävs att man är administratör i ett projekt för att kunna ladda upp filer, öppna dem kan däremot vem som helt med inloggningsrättigheter göra. Första gången en fil öppnas kommer man behöva installera deras plug-in, men efter det öppnas den automatiskt.

8.6.2 Resultat

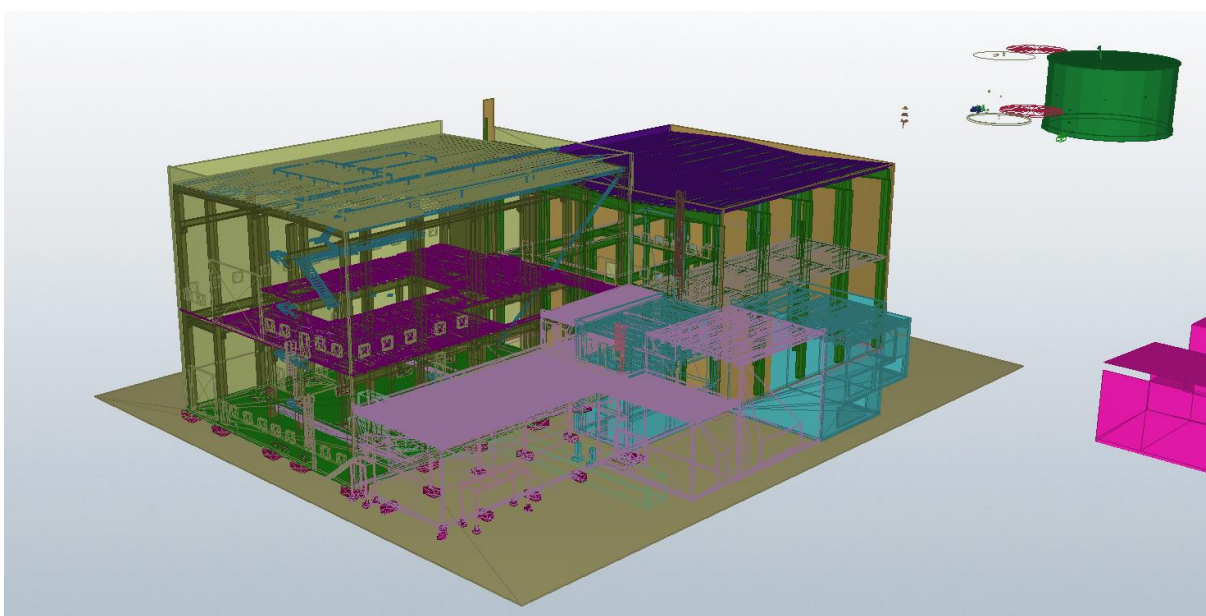
Eftersom det enda filformatet som Maint3D kan läsa som bevarar attribut är IFC, är det IFC som varit huvudspåret i denna implementering. Tidiga försök med andra format ger vid handen att geometrin blir helt acceptabel med till exempel OBJ eller COLLADA, med det finns vissa problem med att bevara utseenden.

Svårigheterna kring IFC-implementeringen har framförallt rört mismatchningen mellan indata och IFC, både vad gäller klasstruktur och vad gäller geometri. Eftersom det inte konsekvent gått att dela upp byggnaderna i till exempel golv och tak, har mycket fått hamna i *IfcBuildingElementProxy*-noder, det vill säga den nod som är tänkt för småsaker som inte ännu har någon relevant klass i IFC-specifikationen. Och även om det finns en del klasser för tankar och rör, är de designade för avlopp snarare än industriutrustning. Detta gör att en del funktioner i Maint3D inte gått att använda, till exempel kan man bara gå runt i modellen om man har en *IfcSlab* att gå runt på.

Geometriproblemet är på många sätt ännu mer svårhanterligt. Någonstans i översättningskedjan skadas eller förvanskas geometrin, och det är mycket svårt att avgöra exakt var. Felen är inte på en synlig nivå, det handlar sannolikt om till exempel ytor som inte längre är helt omslutna eller ligger exakt i ett plan. Tänkbara anledningar är att geometrin korrumpas redan vid DXF-exporten från PDMS, eller så sker det vid importen till FME, eller så finns det kompatibilitetsproblem mellan det FME definierar som en solid kropp och det Maint3D definierar som en solid kropp, eller så finns det avrundningsfel någonstans i kedjan. Oavsett vilket så är det svårt att hitta en lösning som ger ett konsekvent bra resultat. Det finns två olika verktyg i FME som kan omvandla en yta till en solid, *SolidBuilder* och *GeometryCoercer*. Det förstnämnda ger betydligt bättre resultat, men kan inte användas på alla delar av indatamodellen, då vissa geometrier har för mycket fel och kraschar programmet. Den sistnämnda släpper igenom allt, men bygger geometrier som på olika sätt visas fel i viewern.



Figur 8.4 Maint3D lokalt visar bara delar av geometrin. Den texturerade OBJ-filen med markmodellen visas utan problem.



Figur 8.5 Maint3D i webbläsaren visar hela byggnader, men geometrierna blir utan yttre skal vid inzoomning, och markmodellen försvinner.

Det visar sig också att resultaten ser olika ut beroende på om man använder Maint3Ds desktop viewer lokalt, eller om man laddar upp sina filer på Maint3D Server och använder deras webbläsar-plugin. Lokalt får man ett resultat som i figur 8.4, där bara delar av byggnaden syns, av okänd anledning. På servern ser man hela byggnaden, men vid inzoomning försvinner det yttre skalet, och bara inre strukturer syns, som i figur 8.5. Ingen riktig förklaring på detta fenomen har hittats.

Vad gäller markmodellen är det också stora skillnader mellan den lokala viewern och lösningen med server och plug-in. Lokalt fungerar både markmodellen som OBJ-fil med draperad textur, och markmodellen som *IfcSite*-geometri, dock utan texturen (det finns stöd i IFC-

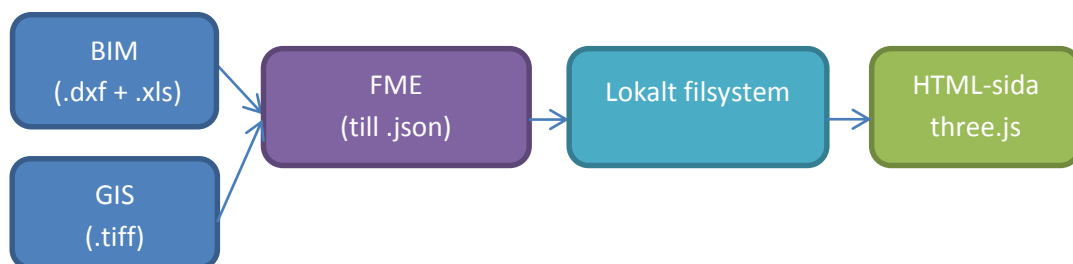
specifikationen för texturer, men det verkar inte vara inbyggt i FMEs IFC-stöd ännu, så texturen följer inte med i exporten). I online-lösningen fungerar inget av det riktigt som det ska. OBJ-filen kan laddas upp på servern, men inte utseende-filen, eftersom den inte har någon godkänd filändelse. Den otexturerade filen hamnar även på fel ställe i förhållande till byggnaderna, trots att de låg i ett gemensamt koordinatsystem från början. *IfcSite*-geometrin dyker inte överhuvudtaget upp. Båda dessa problem härrör gissningsvis ur hur programmet översätter indata till det proprietära F3D-formatet, och är därför svåra att göra något åt utifrån.

Programmet och F3D-formatet är ganska uppenbart uppbyggt för att hantera IFC i första hand, och inte tänkt för att kunna kombinera olika format. Även om OBJ- eller 3DS-filer läses in hamnar de på fel ställe, vilket tyder på att koordinaterna hanteras olika, och kompressionen är mycket sämre än för IFC-filerna. I detta fall kunde till exempel 17 MB OBJ-fil komprimeras till 8 MB F3D, medan 43 MB IFC blev mindre än 4 MB i F3D-format.

8.7 Alternativ 3: Three.js

8.7.1 Implementering

Försöken med three.js är mindre kompletta än de andra, och ska mer ses som ett preliminärt test än en genomförd fallstudie. Flödet har i detta försök sett ut som i figur 8.6. Dock borde lagringssteget i en fullskalig implementation ersättas av en serverlösning.



Figur 8.6. Implementering av export till three.js.

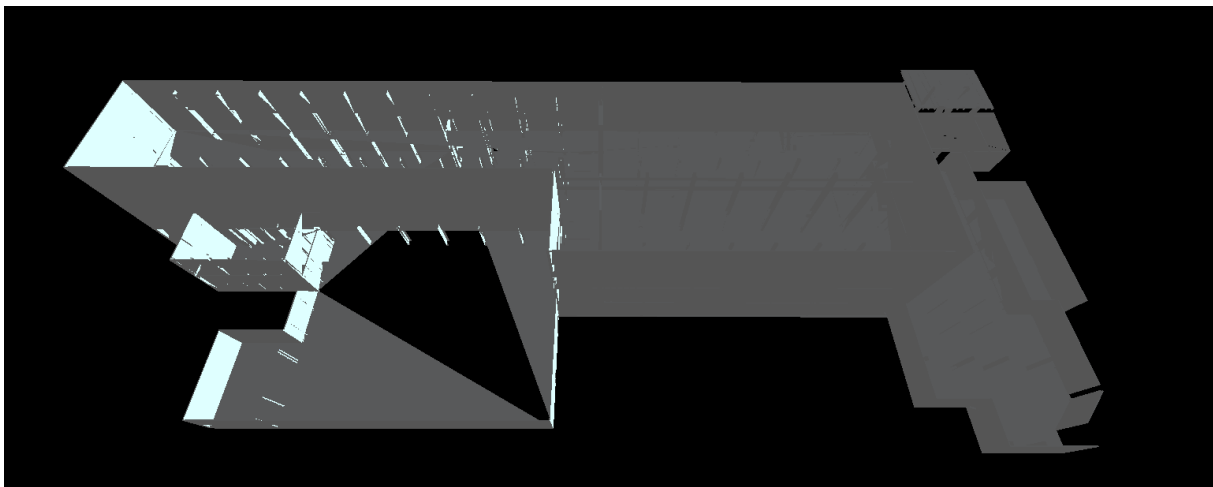
Grundprincipen är att modellen läses in i FME och aggregeras till objekt. Sen används en *JSONTemplater*-transformer för att skriva ut geometrin till JSON-strängar enligt den specifikation som three.js använder internt. Dessa skrivs sedan ut till individuella textfiler, med filändelsen .json.

För att visa upp modellen krävs en HTML-sida, som innehåller ett skript som använder three.js-funktioner för att skapa en scen, ljus, kameror, etc. Detta försök har utgått från exempelkod från en FME-demo (Safe Software, 2015b). I scriptet kan man lägga in en uppladdningsfunktion (*JSONLoader*), som laddar in JSON-filerna. Dessa innehåller då enbart geometri. För att få in attribut också kan man i FME förutom geometrin skriva ut en annan JSON-fil som innehåller attributinformation, och sedan använda en jQuery-funktion för att ladda denna i HTML-sidan och länka den till geometrin.

Demo-koden innehåller också funktionalitet för att ta reda på var en användare klickat, och plocka fram det objekt som klickats på. Detta kan användas för att ta fram det klickade objektet, och skicka motsvarande attributinformation till en textruta i HTML-koden.

8.7.2 Resultat

Att skapa och läsa in geometrier fungerar relativt väl, även om det även här finns en del indataproblem. Framförallt spelar orienteringen av trianglarna i modellen roll här, vilket inte PDMS DXF-export tar hänsyn till. Det gör att det uppstår en del hål och flimmer och andra visuella störningar. IFC-modellen blir betydligt bättre, men även den har en del flimmerproblem. Ett exempel på hur det kan se ut finns i figur 8.7.



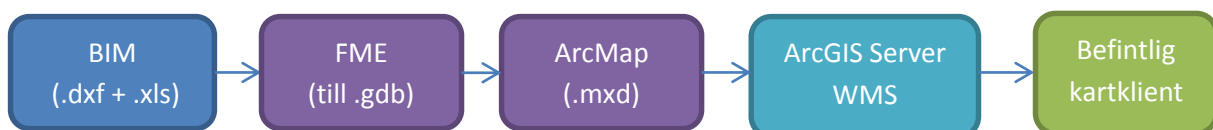
Figur 8.7 Exempel på three.js-geometri.

Ett annat och relaterat problem är att JSON-filerna snabbt blir väldigt stora. Ett våningsplan ur IFC-filen, som från början upptar 23 MB, blir som three.js-JSON 154 MB stor. Laddnings- och uppdateringstiden är fortfarande acceptabel, men det sätter helt klart en övre gräns för hur komplicerade scener man kan visualisera.

8.8 Alternativ 4: 2D-karta

8.8.1 Implementering

För att skapa en 2D-karta av området utifrån 3D-modellen följdes arbetsgången som illustreras i figur 8.8. Två olika varianter skapades, en förenklad för att använda till exempel i översiktskartor, och en med all tillgänglig geometri bevarad. Eftersom tanken är att kartan ska läggas in som lager i en redan befintlig GIS-tjänst finns det ingen anledning att lägga till ytterligare GIS-data.



Figur 8.8. Implementering av omvandlingen till 2D-WMS.

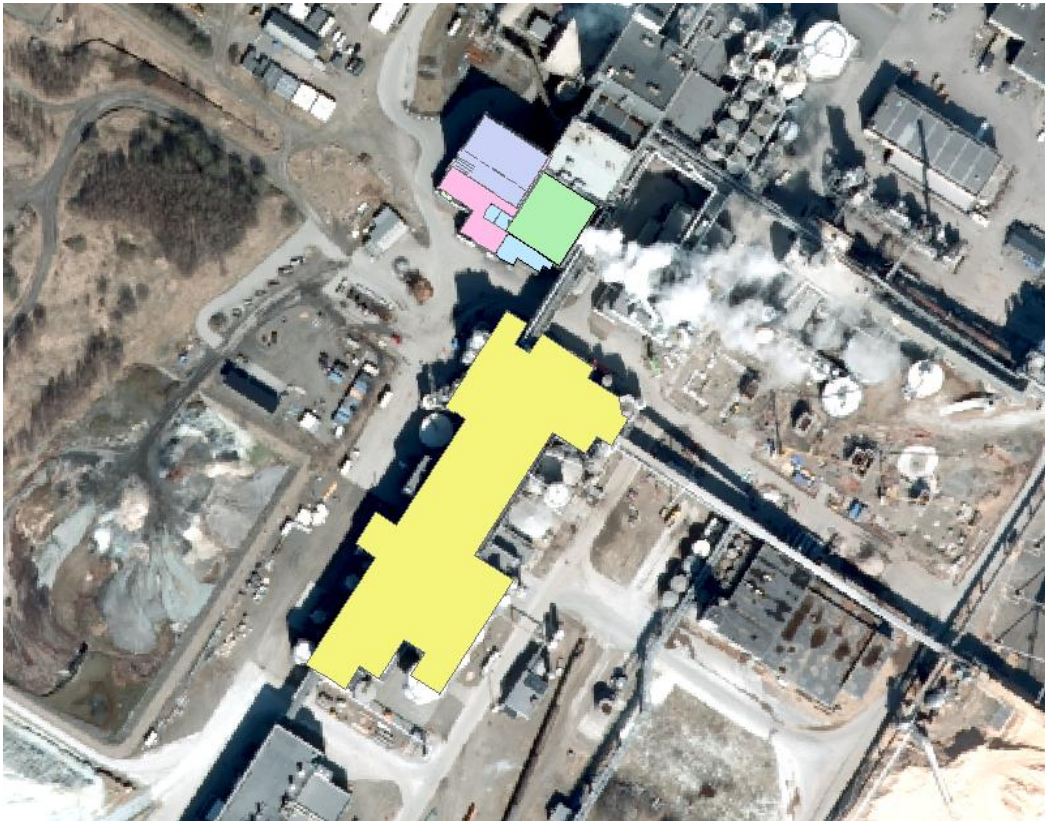
3D-modellen läses in i FME som DXF, tillsammans med Excelbladet med attributdata. För den förenklade modellen filtreras vissa delar bort, till exempel inre strukturer och våningsplan. Geometrin grupperas till objekt, plattas till 2D, och inre gränser löses upp inom objekten. För den förenklade modellen beräknas areor för alla geometridelar, och de som understiger en viss gräns filteras bort. Sedan kombineras geometrierna med attributdata, roteras och flyttas på plats. Koordinatsystemet definieras, och hela modellen transformeras sedan för att passa bakgrundskartan. Filerna skrivs till en ESRI geodatabas, som polygoner. Det FME-workspace som använts finns i bilaga 1.

I ArcMap öppnas ett nytt kartdokument, och geodatabas-filerna läggs till. Symbologin ändras så att varje lager får en egen distinkt färg, och ordningen mellan lagren ändras, så att inte allt döljs under golvet. Sedan används menyverket *File -> Share as -> Service* för att starta guiden för att publicera till ArcGIS Server. För att tjänsten ska följa WMS-standarderna får man välja det under *Capabilities*.

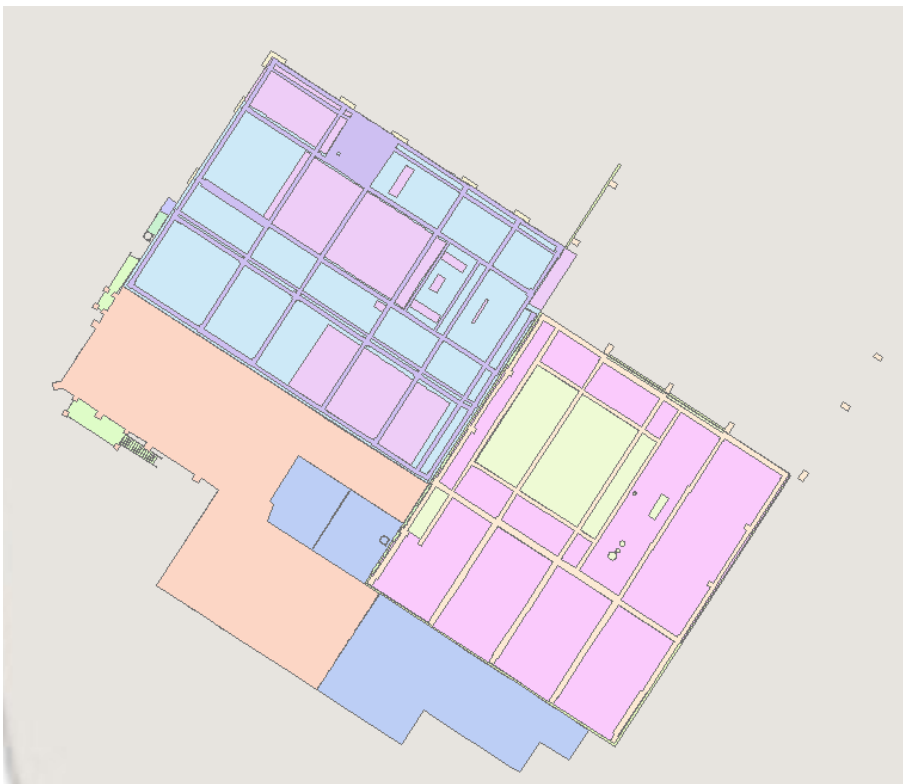
I jämförelsesyfte har också ett försök gjorts med en IFC-modell av en helt annan byggnad. Här kan man använda FMEs Revit-läsare för att direkt få ut golvplaner, vilket gör processen betydligt enklare.

8.8.2 Resultat

Att göra 2D-karta av 3D-modell är inte tekniskt speciellt svårt. För att skapa områdeskartor eller dylikt kan en bild som den i figur 8.9, som visar byggnaderna på Värö bruk utplacerade på ortofotot, till exempel vara praktisk. I mindre skala är det dock en kartografisk utmaning att göra resultatet tillräckligt tydligt för att bli användbart. Speciellt gäller detta när klassifikationerna av indata är så pass varierade. Figur 8.10 visar en del av den detaljerade 2D-kartan över fabriksbyggnaderna, med de tak som går att ta bort borttagna. Geometrierna och placeringen är korrekta, men det är svårt att utläsa några relevanta detaljer. Vid ett mer specificerat användarfall är det möjligt att göra ett mer relevant urval, och då kanske också en tydligare karta. I de fall där till exempel tak och väggar ligger i samma objekt är det dock fortfarande svårt att få till ett bra resultat.

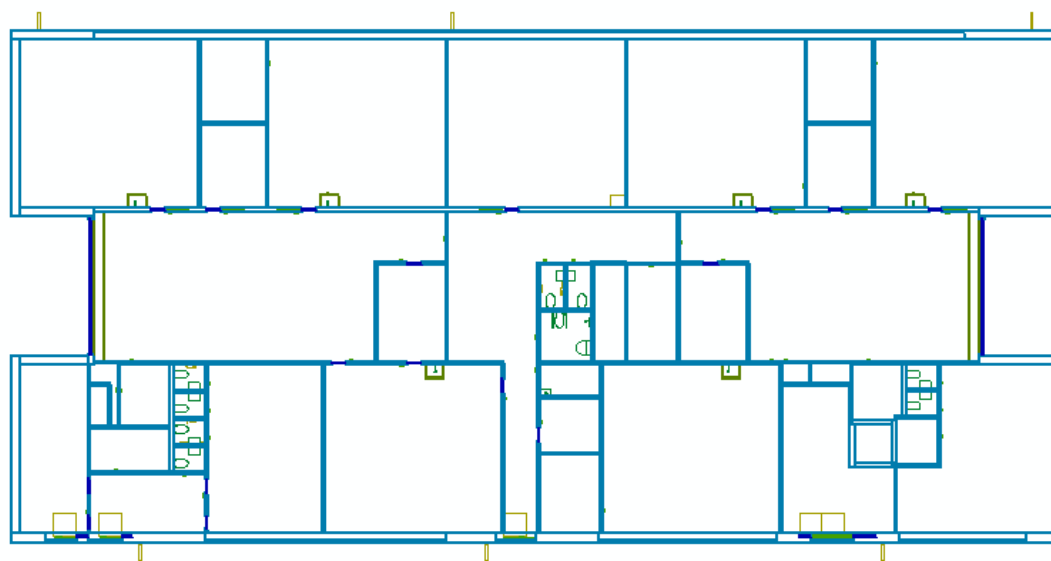


Figur 8.9. Översiktlig 2D-modell inplacerad på ortofotot.



Figur 8.10. 2D-karta Värö bruk.

Som jämförelse är 2D-modellen skapad av IFC-indata både bättre strukturerad och enklare att få fram. Ett exempel, som visar ett våningsplan i en skola, finns i figur 8.11.

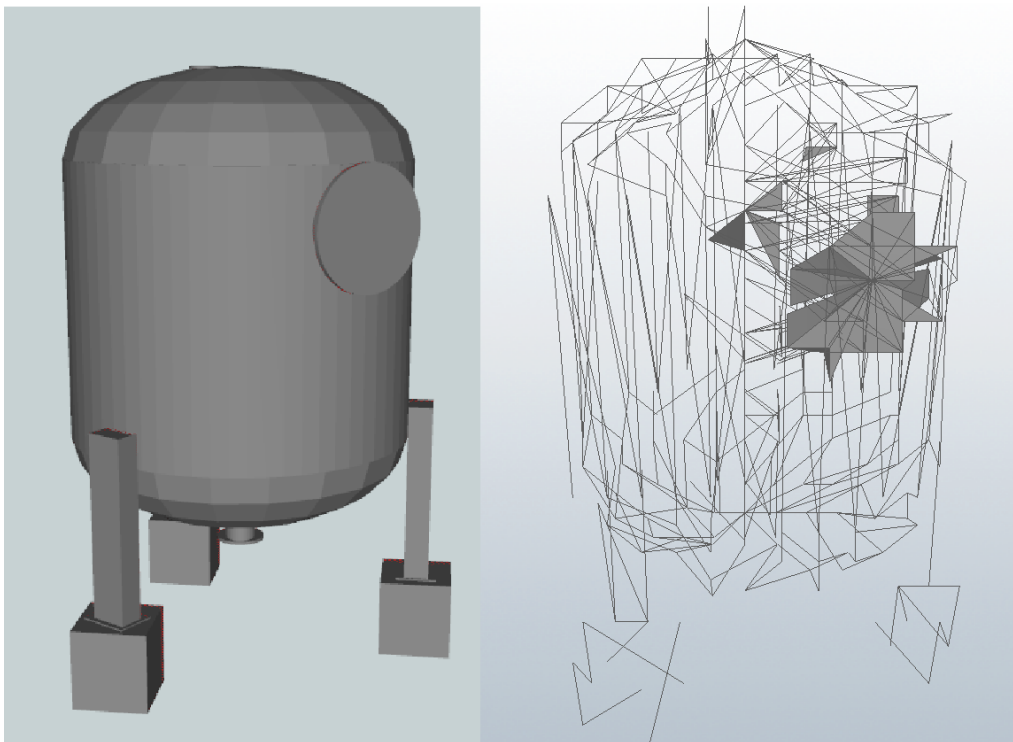


Figur 8.11. 2D-karta från IFC-modell.

8.9 Allmänna observationer

En del problem som dykt upp har varit gemensamma för de olika implementeringsförsöken, och behandlas därför gemensamt i det följande avsnittet.

Ett sådant är att det visat sig att det med nuvarande verktyg och dataformat i princip är omöjligt att lagra tredimensionella data i geografiska koordinater med tillräcklig precision. Figur 8.12 visar tidiga försök, där en tank i COLLADA-format lagts i ett geografiskt koordinatsystem. Den vänstra bilden visar modellen internt i FME, den andra hur exporten ser ut i Maint3D. Även vid liten datamängd blir resultatet uppenbart oacceptabelt, och det blir värre ju större utsträckningen är. Problemen upprepas även i ArcGIS och three.js. Lösningen har i detta projekt varit att skapa ett lokalt koordinatsystem, med origo mitt i den gemensamma bounding-boxen, innan modellen exporteras. Detta fungerar visuellt väl, men har en del principiella problem. Dels kan data inte utan vidare kombineras, om något ska läggas till i modellen måste allt göras om från början. Dels har detta system samma problem som noterades i avsnitt 5.1 om den antaget platta och rätvinkliga jorden. För mindre områden, som i denna fallstudie, är felet för litet för att vara något problem, men i större projekt kan det bli bekymmersamt.



Figur 8.12 Tank i geografiska koordinater internt i FME och exporterad till Maint3D.

Ett annat problem som varit återkommande i många av de undersökta implementeringarna har varit förenkling av 3D-geometrier. Det finns en hel del tänkbara användningsfall där den existerande modellen är onödigt detaljerad, och i flera av visualiseringslösningarna har datamängden varit en begränsande faktor. Det har dock visat sig svårt att förenkla geometrierna med ett acceptabelt resultat. Detta har både data- och verktygsrelaterade

orsaker. Med mer detaljerat klassificerad indata hade geometrier under en viss volym kunnat filteras ut, vilket är ett vanligt angreppssätt i litteratur och teknikexempel. Utan sådana klassificeringar är det närmsta man kommer att filtera ut trianglar under en viss storlek. Detta ger dock ett visuellt ganska märkligt resultat, mer som att modellen är trasig än att den är förenklad. Inga av de förenklingsalgoritmer som är inbyggda i FME ger heller något riktigt acceptabelt resultat, märkliga artefakter och hål i modellen uppstår långt innan man nått någon väsentlig datamängdsreduktion. För att kunna kombinera data över större områden eller implementera någon form av LoD-begrepp är detta en fråga som kommer behöva hitta en lösning.

Ett annat mindre men i vissa fall störande visuellt problem är att markytan med ortofotot legat helt eller delvis över golvytan i byggnaderna, och därmed syns istället för denna. Eftersom bottenytan av byggnaderna är plan, medan marken inte är det, kommer det alltid att vara svårt att passa in exakt. Lösningen på detta problem borde vara att klippa markytan efter byggnaderna i modellen. Tyvärr stöter detta på en hel del problem med FMEs interna datatyper. En fungerande lösning har implementerats, som i grova drag går ut på att byggnadernas fotavtryck först draperas på ytan, görs om till linjer och läggs in som brytlinjer i TIN-modellen. De enskilda trianglarna klipps sedan mot fotavtrycken, aggregeras till en yta, och draperas med ortofotot. Detta fungerar för mindre datamängder, men tar väldigt lång tid att köra, och har inte kunnat genomföras på hela modellen samtidigt eftersom den dator som använts för projektet då fått slut på RAM-minne. Eftersom koordinatproblemet som diskuterades ovan gör att hela modellen måste köras på en gång om den ska hamna på samma plats är detta inte riktigt tillfredställande. Utvidgningen till FME-workspacet finns i bilaga 1.

9. Diskussion

Fallstudien ovan har illustrerat flera av de stora problem som ännu står i vägen för möjligheterna att fritt kombinera 3D-data från olika källor inom GIS- och BIM-världarna. Ingen av de provade lösningarna har nått riktigt ända fram, snarare har fler nya fallgropar som måste hanteras upptäckts. Grovt kan man dela in de uppkomna problemen i en teknisk nivå, en datanivå, och en organisatorisk nivå. Dessa går naturligtvis in i och förstärker varandra, men var och en innehåller viktiga pusselbitar som måste falla på plats.

9.1 Den tekniska nivån

I princip alla de klassiska problem som beskrevs i avsnitt 5.1 har varit aktuella i fallstudien. BIM-data ligger i lokala koordinater, som det visat sig vara svårt att få fram en geografisk utgångspunkt för. Detaljerade geometrier och stora datamängder har gjort många lösningar tungrodda eller rent av ogenomförbara. Problemen med olika geometriska representationer har tagit en lite annan form än väntat, eftersom de snarare rör dataöverföring mellan BIM-program än mellan BIM och GIS. Även datamodellproblemen har snarast handlat om olika paradigmer i olika BIM-modelleringsmiljöer, kanske framförallt eftersom inga data i schematiskt avancerade GIS-format använts.

Geometriska representationer av 3D-data visar sig vara långt mer komplicerade att hantera än den inledande översikten i kapitel 3 ger vid handen. I princip alla program arbetar med en egen intern representation, och även geometrier som utåt sett verkar väldigt lika visas helt olika i olika miljöer. Trianglarnas orientering i ett mesh påverkar till exempel ibland utseendet och ibland inte, och är mycket svåra att göra någonting åt. I FME behandlas ett objekt helt annorlunda beroende på om det är definierat som en yta eller som en kropp som beskrivs av en yta. I nuläget finns ingen riktig lösning på dessa problem, mer än att lära sig exakt hur varje program man arbetar i hanterar geometrin och vad det innebär. Det är dock en uppenbar framtidsfråga hur man ska harmonisera och skapa konsekventa översättningar mellan olika representationer.

Även formatspecifikationer, som utåt är standardiserade, verkar tolkas olika i olika miljöer och beroende på ursprung. Detta är speciellt tydligt i de mer komplicerade formaten, som IFC, men syns även i till exempel DXF och KML-filer. Speciellt när de används som exportformat kan detta bli besvärligt, eftersom de flesta program helt har automatiserat exporter, och som användare har man väldigt begränsat inflytande över var data hamnar och hur filen är strukturerad. Vad gäller IFC finns en mycket stor kvalitetsspridning beroende på hur väl informationsmodellen utnyttjats. Det går att göra validerbara, formellt korrekta IFC-filer där all geometri ligger i ett enda objekt. För att fullt ut ta vara på formatet ska inte bara objekt utan även relationer och system vara inbyggda, vilket i nuläget i princip kräver att filen skrivs i specialprogram. IFC producerad i FME verkar vara fullt läsbar i vissa program men får stora fel i andra, vilket inte funnit någon riktigt tillfredställande förklaring.

På visualiseringssidan har fallstudien också visat att stora datamängder fortfarande är ett reellt tekniskt problem. De lösningar som provats bygger alla på att hela datamängden laddas upp när en fil öppnas, och det ger ganska snabbt ohanterliga väntetider. För att kunna skapa 3D-modeller av större områden, speciellt med den detaljeringsnivå man får från BIM-modellerna, så kommer det att krävas andra lösningar. Inom 2D webb-GIS är det idag vanligt förekommande att man kan hämta valda delar ur stora datamängder med hjälp av servertjänster, och något liknande för 3D-data skulle kunna vara en möjlig väg framåt. Implementeringar av LoD-koncept är också en befintlig teknologi som behöver ta steget över i fler format och omgivningar, men som har potential att bli väldigt viktigt. Inom mobil data är problemet med stora datamängder än värre, på grund av den mycket mer begränsade bandbredden, och att göra 3D-data mobilt utan stora begränsningar kommer sannolikt att vara svårt ännu en tid.

9.2 Datanivån

Fallstudien har visat på den otroliga skillnad kvaliteten på indata kan göra för slutresultatet, och även hur mångfacetterat begreppet "kvalitet" i sammanhanget är. Det kan konstateras att geometrisk precision och validerbarhet ganska snabbt blir lidande av översättningar mellan format, och att olika program och format är olika känsliga för sådana problem. Speciellt omvandlingar mellan olika grundläggande 3D-representationer (se kapitel 3) är svåra, och resultaten långt ifrån perfekta. Det kan också konstateras att om datamodellen från början är datarik, med detaljerade klassificeringar och uppdelningar, är den mycket lättare att jobba vidare med. Att förenkla och göra urval är bara möjligt om det finns tillräckligt med kriterier att göra det utifrån. För GIS-data handlar kvalitet mycket om precision och ajourhållning. Det ortofoto som följde med den geologiska datamängden var till exempel bara ett par år gammalt, men eftersom det är ett område där det byggts mycket är det redan helt inaktuellt.

Bland de dataformat som studerats kan man konstatera att de människoläsbara öppna standarder som hanterar attributdata snabbt blir opraktiskt stora, svårhanterliga och i många fall olämpliga för visualiseringar. De komprimerade binära format som provats kan minska filstorleken rejält, i något fall över 10 gånger, men de är generellt sett proprietära och knutna till en viss programvara. Bra 3D-komprimeringsalgoritmer verkar fortfarande vara något som är så pass specialiserat att man kan och vill ta betalt för det. Ett möjligt undantag är glTF, som också lanserats i en binär form sen detta examensarbete inleddes, och som kan komma att utvecklas till en betydelsefull standard. Bland de öppna visualiseringsformaten finns det inte något som idag hanterar attributdata på ett riktigt tillfredställande sätt. Med de verktyg som finns idag är det därför svårt att själv bygga något som kommer i närheten av de färdiga kommersiella produkterna i termer av snabbhet och kapacitet.

De klassificerade, meningsbärande formaten som diskuterats, som CityGML och IFC, har uppenbara fördelar vad gäller informationsutbyte, standardisering och analys. Däremot är det en avancerad process att skapa dem utifrån andra format, och de strikta scheman, hierarkier och relationer som finns gör det svårt att automatisera och anpassa processerna. De är även

stora och därmed svåra att hantera i visualiserings-sammanhang. De enkla, rena geometriformaten är å andra sidan för enkla för många av de applikationer som diskuterats. Det som just nu helt saknas, både på BIM-sidan och i visualiseringsvärlden, är ett generiskt, objektorienterat 3D-dataformat som kan hantera både geometri och attribut, men utan fastställt schema. Detta är närmast norm i GIS-världen, där attributdata och objektorientering varit viktiga långt innan 3D blev det. Detta talar för att det i framtiden kan vara verktyg med rötterna i GIS snarare än BIM som blir viktiga för mer generisk, automatiserad visualisering. De verktyg som skrivits specifikt för att visa upp BIM-modeller bygger ofta helt på IFC-datamodellen, och är svåra att föra in andra data i.

Det bör poängteras att de datafrågor som diskuterats ovan rör ren visualisering, eftersom det varit fokus för detta examensarbete. Det finns dock en mycket stor sannolikhet för att de tjänster som efterfrågas i framtiden kommer gå förbi rena visualiseringar mot mer avancerade analyser, analogt med utvecklingen hos 2D-data. Då kommer det att behövas antingen programmiljöer som på ett bra sätt kan hantera inte bara utseende utan egenskaper hos flera filformat samtidigt, eller dataformat som är så genomtänkta och heltäckande att de kan modellera en sammanhängande verklighet på ett flexibelt men standardiserat sätt. Inget av dessa alternativ är enkla att åstadkomma, och inget av dem finns fullt ut idag.

9.3 Organisations- och marknadsnivån

En annan sak som blivit uppenbar under examensarbetets gång är att tanken på att data ska kunna användas vidare ännu inte har något starkt fäste bland projektörer och modellerare. De BIM-experters jag haft kontakt med under detta projekt är väldigt insatta i sina egna verktyg och informationsmodeller, men de har generellt sett inte tidigare reflekterat kring hur deras data fungerar i andra sammanhang. Det hade absolut inte varit omöjligt att modellera i till exempel PDMS på ett sätt som underlättade export till både GIS- och andra BIM-format, med bland annat enhetlig uppdelning i olika byggnadsdelar och explicit geografisk positioneringsinformation. Det krävs dock både en tydlig efterfrågan och etablerade rutiner för att det ska ske. Även om det definitivt fortfarande finns en teknisk osmidighet i dataöverföringar mellan program, så hade många svårigheter kunnat förebyggas om framtida återanvändningsmöjligheter beaktades redan på ett tidigt stadium. Att det inte sker idag beror sannolikt till viss del på att man inte är van vid att tänka i dessa banor, och även på den fortfarande ganska tydliga professionella och organisatoriska uppdelningen mellan de som jobbar med byggnadsmodeller och de som jobbar med kartor. Om man aldrig tänkt i termer av att ens 3D-modeller ska kunna leva vidare i kartor och stadsmodeller är det knappast anmärkningsvärt att man inte bygger dem för det.

Det kan nog också misstänkas att vissa mer eller mindre medvetna leverantörlåsningar finns på plats. Mjukvaruleverantörerna har kortsiktigt inget egenintresse av att data från deras program går att använda i konkurrenternas. Inom 2D-geodata har utvecklingen under det senaste decenniet gått bort från en liknande situation, med proprietära format och leverantörlåsningar, mot fler och bättre öppna standarder. Det kommer att bli mycket

spännande att se om 3D-data följer i samma spår framöver, trots den högre tekniska komplexiteten. En aktivt utvecklad öppen OGC-standard för 3D-servertjänster hade till exempel kunnat bli en viktig del av den utvecklingen.

9.4 Alternativa upplägg

Om man tittar på de föreslagna arkitekturer som inte implementerats är det svårt att se att någon av dem skulle lyckats lösa problemet signifikant mycket bättre än de testade just nu. De andra Javascript-baserade 3D-ramverk som diskuterats har på många sätt samma problem som three.js hade, med stora filer och utan inbyggd attributhantering. Sedan examensarbetet inleddes har Cesium-teamet kommit med en del intressanta demonstrationsexempel, som tyder på att det kan bli en bra plattform framöver. Swecos internt utvecklade 3D-motor hade sannolikt kunnat utvecklas för att lösa problemet, men det hade troligen krävt en stor insats av situationsanpassad WebGL-kodning. Av de andra ESRI-lösningarna hade den via CityEngine sannolikt inte förändrat resultatet nämnvärt. Att utnyttja ArcGlobe hade säkert kunnat ge ett bra resultat rent visuellt, men att behöva ladda ner ett helt program, inte bara en plug-in, för att läsa filen vore ett ganska stort avsteg från den ursprungliga målsättningen. Slutligen var den provade BIM-baserade lösningen den enda av de föreslagna som visade sig praktiskt möjlig.

Kriterierna som användes för att utvärdera arkitekturerna formulerades utifrån en ganska allmänt hållen kravbild, och visade sig vara tämligen svåra att specificera och prioritera sinsemellan utan ett specifikt användarfall. Kriteriernas inflytande på vilka arkitekturer som valdes för implementering har därför inte varit helt avgörande. Ofta har kriterieanalysen gett ungefär samma resultat för liknande lösningar, och det som fällt avgörandet har istället varit författarens vana vid olika program, och deras uppskattade användarvänlighet. En annan och mer specificerad kriterieuppsättning hade säkert kunnat ge ett annat resultat.

Utöver de arkitekturer som föreslogs i kapitel 7.2 finns det med all säkerhet fler som skulle kunna vara gångbara alternativ. Urvalet här har påverkats av vilka programvaror som är viktiga för Sweco, vad som föreslagits av intervjupersoner och vad som lyfts fram i de vetenskapliga artiklar som lästs. Samma undersökning gjord i en annan miljö hade säkert till viss del resulterat i andra lösningar och rekommendationer. Ett alternativ som inte undersökts vidare är till exempel att gå via CityGML, till en onlinemiljö anpassad efter det formatet.

9.5 Framtida utveckling

Trots att inget provat alternativ svarat mot alla kriterier, och trots alla teknikutmaningar som kvarstår, visar den utförda undersökningen ändå att det finns skäl för en viss optimism kring den framtida utvecklingen. Behovet av kombinerbarhet och dataintegration har uppfattats, och det finns en vilja, kanske framförallt inom GIS-världen, att driva fram fungerande integrationslösningar. BIM i allmänhet och IFC i synnerhet växer i betydelse, och med ett öppet standardiserat utbytesformat att utgå från är det lättare att bygga långsiktigt fungerande lösningar. Det finns även ett visst tryck på att utöka IFC, med till exempel fler mark- och

väglklasser, även om processen är långsam. Lösningar för servertjänster och strömmade 3D-data är på väg, och kommer utvidga den övre gränsen för möjliga datamängder betydligt.

Det som krävs nu är kanske framförallt att den organisatoriska samarbets- och förändringsviljan följer med den tekniska utvecklingen. BIM- och GIS-användare behöver utöka sina kunskaper om och användning av varandras data. De som skapar BIM-modeller behöver börja knyta geografisk position till sina koordinatsystem och tänka på hur informationsmodellen fungerar i andra sammanhang än sitt eget. De som arbetar med GIS behöver börja lära sig att förstå och använda IFC, eftersom det verkar vara det som kommer gälla som standard under en överskådlig framtid. Interaktiva 3D-modeller på nätet kommer att komma, kommer behöva data från olika källor, och branschen behöver vara beredd.

10. Slutsatser

Detta examensarbete hade från början två specifika syften. Det första handlade om att utreda vilka alternativ som finns för samredovisning av 3D-data över internet, och det andra om att skapa provimplementationer för ett eller ett par lösningsförslag. Det första syftet definierades ytterligare till att utreda geometriska representationer av 3D-data, 3D-dataformat, integrationsmetoder för 3D-data, samt olika sätt att redovisa 3D-data på internet. Dessa två syften har i huvudsak mötts, och gett upphov till en rad slutsatser.

De första delarna av utredningen, kring representationer av 3D-data och 3D-dataformat, visar kanske framförallt på hur stort och utspritt utbudet av olika lösningar är. Olika representationer och format har olika fördelar och olika tänkta användningsområden, även om de till viss del överlappar, och det finns idag inget format som täcker in alla identifierade behov. Detta gör att korrekta översättningar mellan representationer och format är otroligt viktiga för utvecklingen, och kommer att vara viktiga under en överskådlig framtid.

Utredningen kring integrationsproblematik och -metoder mellan BIM- och GIS-data visar att det finns ett antal välkända problemområden, och en aktiv forskning kring hur de ska lösas. Det finns däremot inga enkla svar, och de lösningar som finns är fortfarande beroende av både indata och slutanvändning. Standardisering av hur dataformat används och fortsatt utveckling av formaten kan vara en del av lösningen av dessa problem. Det finns dock en inbyggd paradox i att försöka modellera en komplex verklighet i format som både ska kunna fånga upp den komplexiteten och ändå vara enkla att hantera, och efterhand som formaten utvecklas kommer med stor sannolikhet nya integrationsproblem att dyka upp. Slutsatsen blir därmed att fullständig integration av BIM- och GIS-data på datanivå ännu är svårt och kräver en hel del manuellt arbete, och en universell lösning kommer antagligen inte finnas i den närmaste framtiden.

De delar som behandlar redovisning av 3D-data på internet visar också på stor spridning, och många olika utgångspunkter. De verktyg som specialiserat sig på BIM- eller GIS-data är väldigt specialiserade, och i dagsläget svåra att använda till andra datatyper. De plattformar som har sitt ursprung i mer generell 3D-hantering eller spelgrafik har begränsningar i attributhantering och inbyggda funktioner, och kräver en hel del manuellt arbete för att nå upp till samma funktionalitetsnivå som de färdigbyggda verktygen. Inom detta område finns dock exempel på en oerhört snabb utveckling, och nästan alla de online-visualiseringsverktyg som undersökts har på något vis uppgraderats under examensarbetets gång.

I det ursprungliga syftet specificerades att systemarkitekturer skulle testas med fokus på informationsflöde och användbarhet, det vill säga hur data med fördel flyttas mellan olika program och vad man kan göra med dessa data. Det har visat sig att samtliga undersökta system i någon mån uppvisat begränsningar och problem på båda fronter.

Det är svårt att med utgångspunkt i denna undersökning komma med några rättvisande generella uttalanden kring de lösningsförslag som provats. Resultaten är starkt beroende av indata och hantering, och med programmen i ständig förändring blir de också snabbt inaktuella. Med dessa förbehåll sammanfattar tabell 10.1 hur resultaten sett ut i just detta försök.

Lösningsförslag	Viktiga resultat
CityEngine Web Viewer	Acceptabel visualisering, en del flimmar Attributhantering inbyggd och smidig Begränsad kapacitet Ingen tillgång från mobil
Maint3D	Märkliga fel i visualiseringen Dålig hantering av icke-IFC-data Bäst komprimering och tillgång från mobil
Three.js	Flimrig visualisering Mycket eget utvecklingsarbete krävs för att hantera attribut Osmidigt att utvidga utöver grundfunktioner Stora filer
2D-karta	Svårt att skapa tydlig kartografi utan klassificerad data Webbmiljöerna betydligt bättre utbyggda än för 3D

Tabell 10.1. Sammanfattning av resultaten från fallstudien.

Mycket av problemen som visat sig handlar om verktygens kompatibilitet med indata, och om vilka format och datamängder visualiseringsverktygen kunnat hantera. Detta handlar i grund och botten mycket om att det inte ursprungligen är tänkt att man ska kunna samredovisa data på det här viset i de befintliga verktygen. GIS-programmen är inte tänkta att hantera detaljerade BIM-modeller. BIM-programmen är inte tänkta att hantera geografiska koordinater eller andra dataformat än sina egna. De generiska online-visualiseringsverktygen är inte tänkta att hantera attributdata. Datamängderna är inte uppbyggda för att kunna återanvändas i andra sammanhang. Ingen av dessa svårigheter är i grunden tekniskt olösliga, även om det absolut finns utmaningar, de ingår bara inte i mängden av problem som de befintliga verktygen är designade för att lösa.

Detta tar sig uttryck på många sätt, som här delats upp i en teknisk nivå, en datanivå och en organisatorisk nivå. På den tekniska nivån finns framförallt översättningsproblem mellan olika

miljöer och format. Även det som på pappret borde vara möjligt blir omöjligt, eftersom de olika verktygen inte är tänkta att kombineras, och därmed inte testats och utvecklats för det. På datanivån handlar det om att data är felstrukturerad eller innehåller fel information för att fungera väl i samredovisningsform. Återigen kan problemet hänföras till att inget enskilt format egentligen är designat för att hantera alla de här typerna av information, eller tänkt att kunna kombineras med andra. På organisationsnivån handlar det om att produktionsmiljöer och organisationer för BIM- och GIS-data i stor utsträckning fortfarande är åtskilda, och att data inte är tänkta att leva vidare i andra sammanhang.

För att komma framåt och närma sig en fungerande lösning på samredovisningsproblematiken behövs insatser på alla tre nivåerna. De tekniska problemen med att flytta data behöver lösas, vilket bland annat kommer att kräva samarbete mellan de stora mjukvaruleverantörerna. Arbetet med att utveckla bra standardiserade dataformat behöver fortsätta. Och människor och företag inom BIM- och GIS-världarna måste börja kommunicera och definiera en gemensam målbild och en gemensam väg framåt.

Källförteckning

Litteratur

- Abdul-Rahman, A. och Pilouk, M. 2008. *Spatial Data Modelling for 3D GIS*. Springer.
- ACTIVE3D. 2015. Hämtad 2015-06-08 från <http://www.active3d.net/>
- Aien, A., M. Kalantari, A. Rajabifard, I. Williamson, och J. Wallace. Towards integration of 3D legal and physical objects in cadastral data models. *Land Use Policy* 35:140-154. doi:10.1016/j.landusepol.2013.05.014
- Akin, O. 2009. Current Trends and Future Directions in CAD. I *CAD and GIS Integration*, red. H. A. Karimi och B. Akinci, pp. 1-21. CRC Press.
- Angel, E. och D. Shreiner. 2012. *Interactive Computer Graphics*. 6:e upplagan. Boston: Pearson.
- Autodesk. 2015. *AutoCAD 2000 DXF Reference*. Hämtad 2015-02-27 från <http://www.autodesk.com/techpubs/autocad/acad2000/dxf/>
- Ballagh, L. M., B. H. Raup, R. E. Duerr, S. J. S. Khalsa, C. Helm, D. Fowler och A. Gupte. 2011. Representing scientific data sets in KML: Methods and challenges. *Computers & Geosciences* 37:57–64.
- van den Brink, L., J. Stoter och S. Zlatanova. 2013. Establishing a national standard for 3D topographic data compliant to CityGML. *International Journal of Geographical Information Science* 27:92-113. doi: 10.1080/13658816.2012.667105.
- BuildingSMART. 2013. *IFC4 Official release*. Hämtad 2015-03-03 från <http://www.buildingsmart-tech.org/ifc/IFC4/final/html/index.htm>.
- Cesium. 2015. Hämtat 2015-03-24 från <https://cesiumjs.org/index.html>
- CityGML.org. 2012. Hämtad 2015-02-19 från <http://www.citygml.org/index.php?id=1523>.
- Eastman, C., et al. 2008. *BIM handbook : a guide to building information modeling for owners, managers, designers, engineers, and contractors*. Hoboken, N.J. : Wiley.
- Ekholm, A., H. Blom, K. Eckerberg, K. Löwnertz, V. Tarandi. 2013. *BIM – Standardiseringsbehov*. Hämtad 2015-06-12 från http://www.bimalliance.se/Natverk_och_utveckling/Projekt/Projektkatalog/BIM_standardiseringsbehov

- El-Mekawy, M., A. Östman och I. Hijazi. 2012. An Evaluation of IFC-CityGML Unidirectional Conversion. *International Journal of Advanced Computer Science and Applications* 3:159-171.
- El-Mekawy, M. 2013. Unified Building Model for 3D Cities - the Integration of IFC and CityGML. Doktorsavhandling. Stockholm: KTH.
- ESRI. 2011. *Esri Acquires 3D Software Company Procedural*. Hämtad 2015-04-14 från <http://www.esri.com/news/releases/11-3qtr/esri-acquires-3d-software-company-procedural.html>
- ESRI. 2015a. *About CityEngine Web Viewer*. Hämtad 2015-04-15 från <https://doc.arcgis.com/en/arcgis-online/reference/about-cityengine-web-viewer.htm>
- ESRI. 2015b. *What is a globe service?* Hämtad 2015-04-15 från <http://server.arcgis.com/en/server/latest/publish-services/windows/what-is-a-globe-service-.htm>
- ESRI. 2015c. *Scene services*. Hämtad 2015-06-12 från <http://server.arcgis.com/en/server/latest/publish-services/windows/scene-services.htm>
- Evans, A., M. Romeo, A. Bahrehmand, J. Agenjo, och J. Blat. 2014. 3D graphics on the web: A survey. *Computers & Graphics* 41:43–61.
- Gröger, G. och L. Plümer. 2005. How to Get 3-D for the Price of 2-D - Topology and Consistency of 3-D Urban GIS. *GeoInformatica* 9: 139-158. doi: 10.1007/s10707-005-6431-2.
- Gröger, G. och L. Plümer. 2012. CityGML – Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing* 71:12–33.
- Göteborgs Stad. 2015. *Min Stad*. Hämtad 2015-06-12 från <http://minstad.goteborg.se/minstad/index.do>
- Isikdag, U. och S. Zlatanova. 2009. Towards Defining a Framework for Automatic Generation of Buildings in CityGML Using Building Information Models. I *3D Geo-Information Sciences, Lecture Notes in Geoinformation and Cartography*, red. J. Lee och S. Zlatanova, pp 79-96. Springer.
- Khronos group. 2015. *glTF*. Hämtad 2015-03-24 från <http://gltf.gl/>
- Kim, Y., H. Kang och J. Lee. 2014. Developing CityGML Indoor ADE to Manage Indoor Facilities. I *Innovations in 3D Geo-Information Sciences, Lecture Notes in Geoinformation and Cartography*, red. U. Isikdag. Springer.

- Lantmäteriet. 2013a. *HMK Höjddata*. Hämtad 2015-04-03 från www.lantmateriet.se/globalassets/om-lantmateriet/var-samverkan-med-andra/handbok-mat--och-kartfragor/hmk-geodatainsamling/ad_hmk_hojddata.pdf
- Lantmäteriet. 2013b. *Demodata - GSD-Höjddata, grid 2+*. Hämtad 2015-02-19 från <https://www.lantmateriet.se/sv/Kartor-och-geografisk-information/Hojddata/GSD-Hojddata-grid-2/Demodata/>
- Lantmäteriet. 2014a. *Förutsättningar för att tillhandahålla kart- och bildinformation i tre dimensioner (3D)*. Redovisning av uppdrag från regeringen, dnr 505-2013/3895.
- Lantmäteriet. 2014b. *Tilläggsuppdrag - Förutsättningar för att tillhandahålla kart- och bildinformation i tre dimensioner (3D)*. Redovisning av uppdrag från regeringen, dnr 505-2013/3895.
- Lantmäteriet. 2015. *Höjddata*. Hämtad 2015-06-12 från <http://www.lantmateriet.se/sv/Kartor-och-geografisk-information/Hojddata/>
- Ledoux, H. och M. Meijers. 2010. Topologically consistent 3D city models obtained by extrusion. *International Journal of Geographical Information Science*, 25: 557–574.
- Mignard, C. och C. Nicolle. 2014. Merging BIM and GIS using ontologies application to urban facility management in ACTIVE3D. *Computers in Industry*, 65: 1276–1290.
- Malmö Geologiatlas. 2010. ©Kartunderlag: Malmö Stadsbyggnadskontor.
- Mao, B., Y. Ban och L. Harrie. 2011. A multiple representation data structure for dynamic visualisation of generalised 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing* 66:198–208.
- OGC. 2006. *OpenGIS® Web Map Server Implementation Specification*. Hämtad 2015-06-12 från <http://www.opengeospatial.org/standards/wms>
- OGC. 2008. *OGC® KML*. Hämtad 2015-02-27 från <http://www.opengeospatial.org/standards/kml>
- OGC. 2010. *Draft for Candidate OpenGIS® Web 3D Service Interface Standard*. Hämtad 2015-06-10 från <http://www.opengeospatial.org/standards/dp>
- OGC. 2015. *OGC City Geography Markup Language (CityGML) Encoding Standard*. Hämtad 2015-02-27 från <http://www.opengis.net/spec/citygml/2.0>
- Prieto I., J.L. Izgara, F.J. Delgado. 2012. From point cloud to Web 3D through CityGML. I *18th International Conference on Virtual Systems and Multimedia*. IEEE.

- Rafiee, A., E. Dias, S. Fruijtier, H. Scholten. 2014. From BIM to geo-analysis: view coverage and shadow analysis by BIM/GIS integration. *12th International Conference on Design and Decision Support Systems in Architecture and Urban Planning*. Procedia Environmental Sciences 22:397 – 402.
- Safe Software. 2015a. *BIM to GIS - IFC Building to CityGML – Advanced*. Hämtad 2015-04-21 från https://knowledge.safe.com/articles/Samples_and_Demos/BIM-to-GIS-IFC-Building-to-CityGML-Advanced
- Safe Software. 2015b. *FME/threejs*. Hämtad 2015-05-04 från <https://www.safe.com/fme/new/demo.html>
- SVSI. 2015. *Maint3D User Manual*. Hämtad 2015-04-16 från <http://www.svsi.fi/maint3d/manual.html>
- Södra. 2014. *Södra Cell Värö byggs ut 2014-2016*. Hämtad 2015-05-13 från <http://www.sodra.com/sv/Massa/Vara-massabruk/Sodra-Cell-Varo/Expansion-Sodra-Cell-Varo/>
- Tegtmeier, W., S. Zlatanova, P.J.M. van Oosterom, H.R.G.K. Hack. 2014. 3D-GEM: Geotechnical extension towards an integrated 3D information model for infrastructural development. *Computers & Geosciences* 64:126–135.
- Three.js. 2015. Hämtad 2015-03-25 från <http://threejs.org/>
- Trafikverket. 2015. *Trafikverket beställer BIM*. Hämtad 2015-06-12 från <http://www.trafikverket.se/Foretag/Bygga-och-underhalla/Teknik/trafikverket-bestaller-bim/>
- Volk, R., J. Stengel och F. Schultmann. 2014. Building Information Modeling (BIM) for existing buildings — Literature review and future needs. *Automation In Construction*, 38:109-127. doi:10.1016/j.autcon.2013.10.023
- Web3D Consortium. 2015. Hämtad 2015-03-24 från <http://www.web3d.org/>
- Wikipedia, 2015a. *B-rep*. Hämtad 2015-04-07 från <https://fr.wikipedia.org/wiki/B-Rep>
- Wikipedia, 2015b. *Voxel*. Hämtad 2015-04-07 från <https://en.wikipedia.org/wiki/Voxel>
- Wikipedia, 2015c. *Constructive Solid Geometry*. Hämtad 2015-04-07 från https://en.wikipedia.org/?title=Constructive_solid_geometry
- Zamyadi, A., J. Pouliot och Y. Bédard. 2014. Towards 3D Metadata for Discovering 3D Geospatial Models. I *Innovations in 3D Geo-Information Sciences, Lecture Notes in Geoinformation and Cartography*, red. U. Isikdag. Springer.

Muntliga källor

Brand, Christoffer. 3D-Systemutvecklare, Sweco Position. Telefon- och mailkontakt under februari-mars 2015 angående Sweco elements och 3D-modeller i allmänhet.

Ekelund, Fredrik. Verksamhetsutvecklare, Sweco Position. Kontinuerlig kontakt kring användande av och kundnytta från 3D-data.

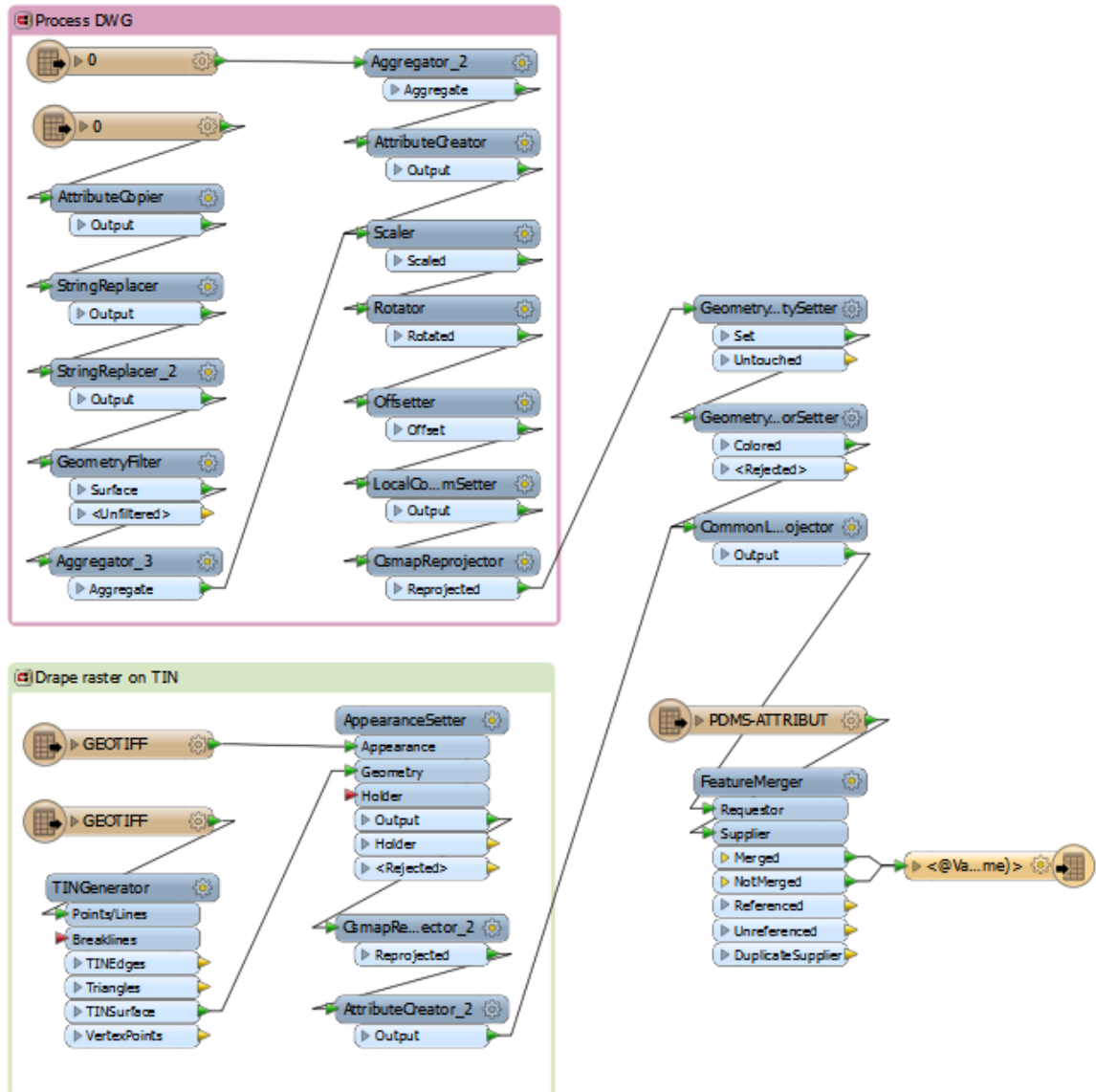
Harr, Sölve. CAD/BIM Ansvarig, Sweco Systems. Telefon- och mailkontakt kontinuerligt under våren 2015 kring BIM-data och Maint3D.

Karlsson, Niklas. Systemspecialist, Sweco Industry. Telefon- och mailkontakt kontinuerligt under våren 2015 kring PDMS-exporter och Värö bruk-modellen.

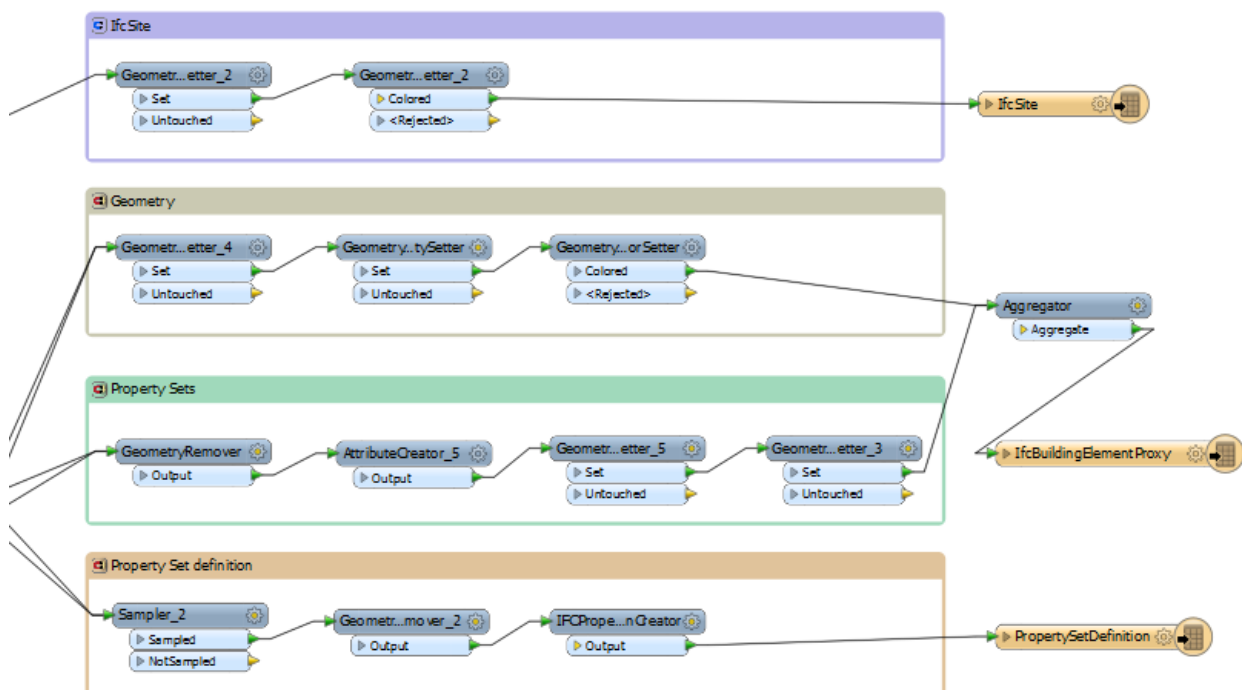
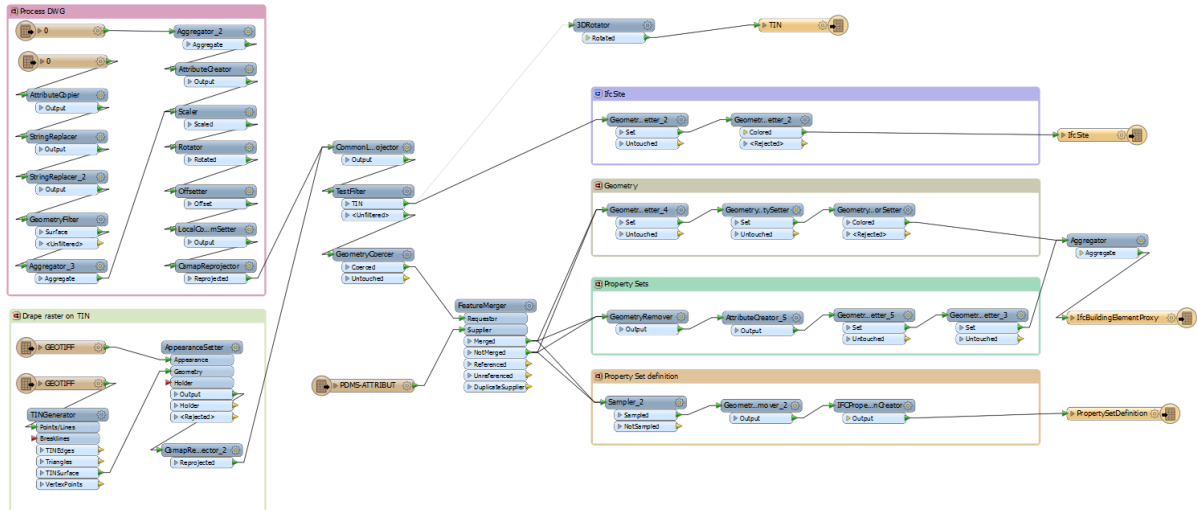
Rothstein, Peter. Projektledare, Sweco Position. Kontakt under mars-april 2015 kring geologiska 3D-data.

Bilaga 1 – FME workspaces

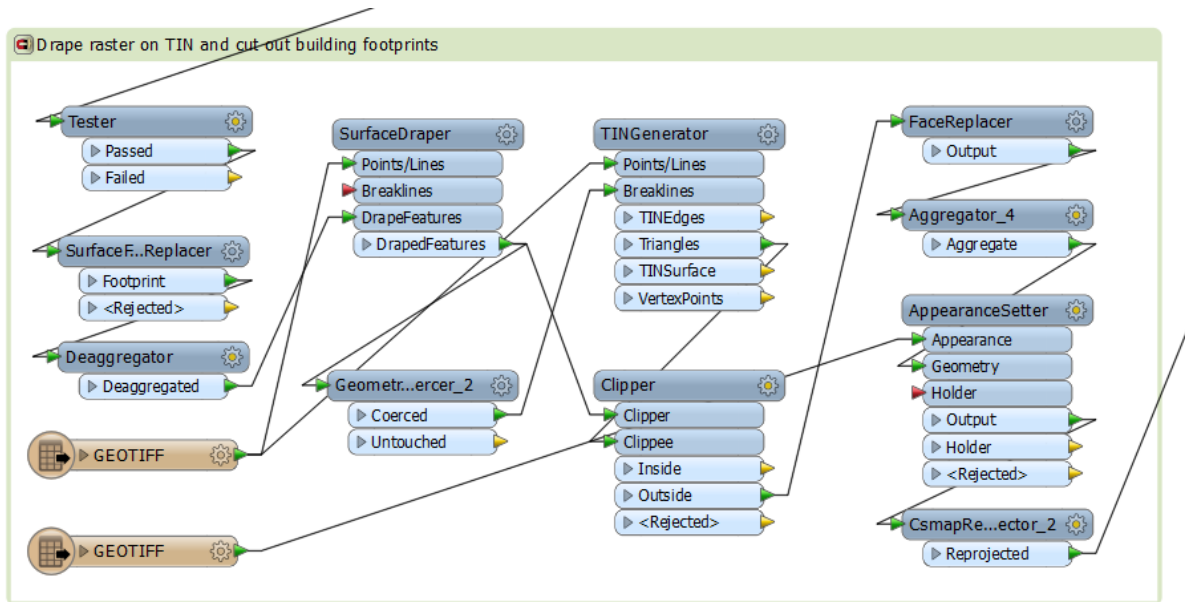
För omvandlingen till en geodatabas, som senare användes för att skapa en CityEngine 3D Web Scene, användes detta workspace:



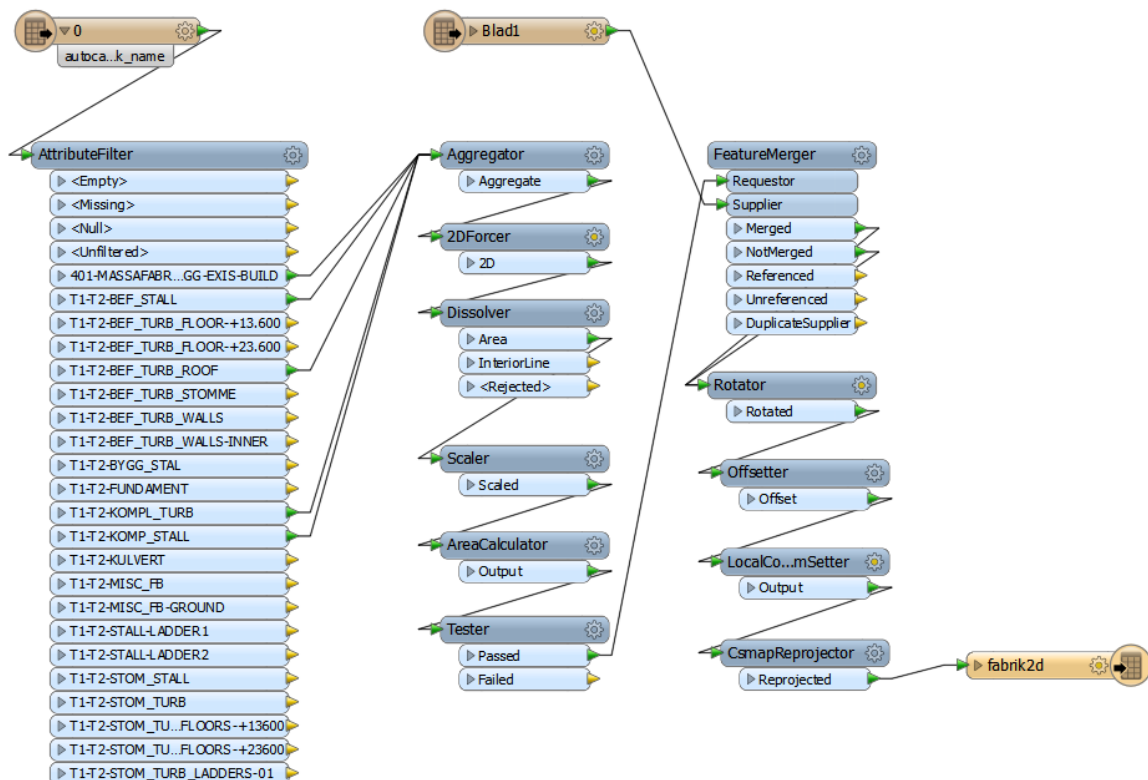
Det workspace som användes för att skriva IFC för att använda i Maint3D är för stort för att bli synligt på en sida i sin helhet. Den första delen, som behandlar indata, är väldigt lik den för att skriva till geodatabasen i exemplet ovan. Den andra delen, som skriver till IFC, visas i förstoring nedan.



För att klippa markmodellen efter byggnaderna ersattes den gröna "Drape raster on TIN"-rutan i workspacen ovan med följande:



För omvandlingen till 2D-karta användes detta workspace:



Bilaga 2 - Ordlista

Denna lista innehåller definitioner av termer och förkortningar som används i texten.

3D-fastighet	En fastighet som till skillnad från en vanlig fastighet avgränsats även i höjddled. Kan t.ex. innefatta en del av en byggnad.
API	<i>Application Program Interface</i> . Beskriver hur applikationsprogram kan interagera med en viss grundprogramvara, t.ex. vilka funktioner som finns tillgängliga.
Attribut	Data om egenskaper hos objekt. Ett byggnadsobjekt kan t.ex. ha attributen adress, höjd, ägare, fasadfärg, m.fl.
BIM	I detta arbete: <i>Building Information Model</i> . En informationsmodell som knyter information om byggnader och byggnadsdelar till en geometrisk representation av dessa. Alternativ betydelse: <i>Building Information Modelling</i> , som är processen att framställa och använda en sådan modell.
Bounding box	Det minsta axelparallella rätblock som innefattar all geometri hos ett objekt eller en grupp objekt.
Bygg- och anläggningsdata	Data som är knutna till ett specifikt konstruktionsprojekt. Normalt sett definierade i ett lokalt koordinatsystem.
CAD	Computer Aided Design. Designverktyg i datorn. Används inom många olika områden, bland annat arkitektur och fysisk planering.
Drapera	Lägga en bild över en geometri så att den följer ytan, t.ex. ett flygfoto över en terrängmodell.
Ellipsoid	Matematisk modell för att approximera jordklotet. Används bland annat för definition av geodetiska referenssystem.
ESRI	<i>Environmental Science Research Institute</i> . Amerikanskt företag som producerar programvarorna inom ArcGIS-sviten.
Extrudera	En metod för att skapa 3D-data genom att "dra ut" en tvådimensionell geometri i höjddled.
Fotavtryck	Den yta en 3D-struktur täcker på marken. T.ex. ytan som täcks av en byggnad.
Geodata	Data som på något vis knutits till en plats i ett geografiskt koordinatsystem.

GIS	<i>Geographic Information System</i> . Ett datorbaserat system för att hantera geografisk information.
GML	<i>Geography Markup Language</i> . Ett XML-baserat märkspråk för geografiska data. Innehåller bland annat standarder för beskrivning av geometri och koordinater.
GUI	<i>Graphical User Interface</i> . Grafiskt användargränssnitt, d.v.s. det en användare möts av och använder för att kontrollera funktionerna i ett datorprogram.
Hierarkisk scenograf	Ett sätt att strukturera innehållet i en 3D-scen. Beskrivs som en trädstruktur där scenen i sig är rot-noden, och kameror, ljus, geometriska objekt o.s.v. är barn. Transformationer som appliceras på någon nod ärvs av alla objekt på lägre nivåer.
HTML	<i>Hyper Text Markup Language</i> . Det märkspråk som används för att skapa internetsidor. Byggs upp av olika element med start- och sluttaggar.
HTTP	<i>Hyper Text Transfer Protocol</i> . Det kommunikationsprotokoll som används för att dela data över internet, med hjälp av anrop mellan server och klient.
ISO	<i>International Standardisation Organisation</i> . Publicerar internationella standarder för produkter och processer inom ett brett spektrum av områden.
JavaScript	Ett skriptspråk som huvudsakligen används för att göra HTML-sidor interaktiva. Utökas ofta med olika områdesspecifika funktionsbibliotek.
jQuery	Ett bibliotek som förenklar många javascriptfunktioner.
JSON	<i>JavaScript Object Notation</i> . Ett utbytesformat för data som kan organiseras antingen som värde/nyckel-par eller listor.
LoD	<i>Level of Detail</i> . Beskriver detaljnivån i ett dataset enligt fördefinierade kriterier. Kan vara olika implementerat i olika format.
Mesh	En datastruktur som beskriver ett 3D-objekt som en yta uppbyggd av orienterade trianglar. Används i många 3D-dataformat.
Metadata	Data om data. Kan t.ex. ange vem som skapat data, när, hur och med vilken kvalitet.
OGC	<i>Open Geospatial Consortium</i> . Ideell organisation som bland annat utvecklar öppna standarder för geodata.

Ortofoto	Flygfoto som korrigeras geometriskt så att alla objekt förefaller vara avbildade rakt uppifrån och i samma skala.
Parametrisk modellering	Modellering byggd på användarangivna regler och parametrar. En regel kan t.ex. vara att väggen ska gå upp till taket. Om parametern "takhöjd" ändras så ändras automatiskt då också väggen.
Plug-in	Ett program som utökar funktionaliteten hos t.ex. en webb-browser. Kräver installation.
Rekursiv	Beskriver en metod eller process som kallar på sig själv tills ett visst stoppvärde uppnåtts.
Shader	Inom datorgrafiken: ett program som beskriver hur ett objekt färgsätts. Kallas varje gång en scen renderas.
Shape-fil	Ett filformat för vektordata som används i ESRI-produkter, och i viss utsträckning som utbytesformat. Lagrar geometri och attribut i separata filer.
Spatiala data	Alla data som har en spatial komponent, d.v.s. som på något vis kan lokaliseras. Delas här grovt in i geodata och bygg- och anläggningsdata.
Textur	Inom datorgrafiken: en bild som läggs utanpå en geometri för att ge ett visst utseende.
W3C	<i>World Wide Web Consortium</i> . Organisation som utvecklar standarder för Internet.
W3DS	<i>Web 3D Service</i> . Utkast till OGC-standard för 3D-servertjänster över nätet.
WebGL	<i>Web Graphics Language</i> . En version av grafik-APIt OpenGL som utvecklats speciellt för 3D-grafik på webben.
Websockets	Protokoll som möjliggör kontinuerlig kommunikation mellan klient och server, istället för att vänta på HTTP-anrop.
WMS	<i>Web Map Service</i> . OGC-standard för 2D-karttjänster över internet.
XML	<i>eXtensible Markup Language</i> . Märkspråk som organiserar information med start- och sluttaggar. Kan användas med ett schema som definierar vilken information som kan eller ska finnas i en viss dokumenttyp.

Institutionen för naturgeografi och ekosystemvetenskap, Lunds Universitet.

Student examensarbete (Seminarieuppsatser) i geografisk informationsteknik. Uppsatserna finns tillgängliga på institutionens geobibliotek, Sölvegatan 12, 223 62 LUND. Serien startade 2010. Hela listan och själva uppsatserna är även tillgängliga på LUP student papers och via Geobiblioteket (www.geobib.lu.se)

Serie examensarbete i geografisk informationsteknik

- 1 *Patrik Carlsson och Ulrik Nilsson* (2010) Tredimensionella GIS vid fastighetsförvaltning.
- 2 *Karin Ekman och Anna Felleson* (2010) Att välja grundläggande karttjänst – Utveckling av jämförelsemodell och testverktyg för utvärdering
- 3 *Jakob Mattsson* (2011) Synkronisering av vägdata med KML och GeoRSS - En fallstudie i Trafikverkets verksamhet
- 4 *Patrik Andersson and Anders Jürisoo* (2011) Effective use of open source GIS in rural planning in South Africa
- 5 *Nariman Emamian och Martin Fredriksson* (2012) Visualisering av bygglovsärenden med hjälp av Open Source-verktyg - En undersökning kring hur man kan effektivisera ärendehantering med hjälp av en webbapplikation
- 6 *Gustav Ekstedt and Torkel Endoff* (2012) Design and Development of a Mobile GIS Application for Municipal Field Work
- 7 *Karl Söderberg* (2012) Smartphones and 3D Augmented Reality for disaster management - A study of smartphones ability to visualise 3D objects in augmented reality to aid emergency workers in disaster management
- 8 *Viktoria Strömberg* (2012) Volymberäkning i samhällsbyggnadsprojekt
- 9 *Daniel Persson* (2013) Lagring och webbaserad visualisering av 3D-stadsmodeller - En pilotstudie i Kristianstad kommun
- 10 *Danebjer Lisette och Nyberg Magdalena* (2013) Utbyte av geodata - studie av leveransstrukturer enligt Sveriges kommuner och landstings objekttypskatalog
- 11 *Alexander Quist* (2013) Undersökning och utveckling av ett mobilt GIS-system för kommunal verksamhet
- 12 *Nariman Emamian* (2014) Visning av geotekniska provborrningar i en webbmiljö
- 13 *Martin Fredriksson* (2014) Integrering av BIM och GIS med spatiala databaser – En prestandaanalys
- 14 *Niklas Krave* (2014) Utveckling av en visualiseringsapplikation för solinstrålningsdata
- 15 *Magdalena Nyberg* (2015) Designing a generic user interface for distribution of open geodata: based on FME server technology
- 16 *Anna Larsson* (2015) Samredovisning av BIM- och GIS-data