

"Would you tell me, please, which way I ought to go from here?"

"That depends a good deal on where you want to get to," said the Cat.

"I don't much care where—" said Alice.

"Then it doesn't matter which way you go," said the Cat.

"—so long as I get *somewhere*," Alice added as an explanation.

"Oh, you're sure to do that," said the Cat, "if you only walk long enough."

Revit Families en Data Handling

Martin Plasschaert, 18 december 2015

Colofon

Auteur: Martin Plasschaert
Datum: 18 december 2015
Plaats: KOEWACHT
Opleiding: HBO ACE Architectural Designer
Instituut: TEC / CAD College BV
Kerkenbos 1018 B
6546 BA NIJMEGEN
024-3565677
info@cadcollege.nl
Begeleider: Ir. Ronald Boeklagen
Beoordeelaars: Ing. Rinus de Boer
Ing. Wim de Goede

Samenvatting

Deze eindopdracht is tweeledig, enerzijds wordt het gebruik en de mogelijkheden van een tweetal families beschreven. Deze families zijn een uitwerking van standaard functionaliteit binnen Revit. Anderzijds wordt het proces beschreven om gegevens, buiten de basismogelijkheden die er reeds zijn binnen Revit, te linken aan objecten binnen het Revit-model.

Voorwoord

Allereerst wil ik beginnen met een motivatie voor de keuze van de voorpagina, een afbeelding uit Alice in Wonderland van Lewis Carroll.

De tekst van de voorpagina vertalend naar deze opdracht houdt in dat Revit, en soortgelijke pakketten, schier oneindig zijn in hun mogelijkheden, dat er veelal meer dan één manier is om tot een oplossing te komen. Deze constatering en het vervolgens benutten van deze mogelijkheden en omzetten naar een praktische invulling daarvan blijken twee verschillende zaken te zijn.

In aanvang heb ik gekozen voor een uitwerking waarvan ik op dat moment niet eens wist of dit mogelijk was binnen Revit. Het eindpunt bepalen, waar wil je naar toe, de gekozen oplossingen, de route naar het eindpunt, waarbij tegelijkertijd gezegd moet worden dat de ene oplossing niet beter of slechter hoeft te zijn dan de andere, zolang er maar naar het eindpunt wordt toegewerkt en het daadwerkelijk functioneel werkend maken van de gekozen opdracht is niet altijd een eenvoudige weg gebleken.

De afbeelding zelf staat voor een bepaalde grafische eigenschap die ik zelf niet heb en me daar ook bewust van ben.

Het verfraaien van afbeeldingen door middel van rendering is daarom binnen deze opdracht niet uitgewerkt.

De nadruk is zo veel mogelijk gelegd op het benutten van en waar nodig toevoegen van functionaliteit. Soms zijn verschillende methoden uitgewerkt die beiden hetzelfde realiseren, daar is dan op dat moment voor gekozen om zoveel mogelijk van het pakket te ontdekken. Dan weer zijn sommige onderdelen niet verder uitgewerkt, lees afgemaakt, omdat deze verdere uitwerking slechts een herhaling van reeds gedane handelingen zou zijn en niets meer zou toevoegen aan de uitdieping van het pakket.

Deze eindopdracht is het resultaat van de cursussen Autocad & VB.NET, Revit Basis en Revit Gevorderd in het kader van de HBO opleiding ACE Architectural Designer aan het TEC/CAD College te Nijmegen.

Martin Plasschaert

Martin Plasschaert

Inhoud:

1. Inleiding
 - 1.1 Gebruikte software en programmeertalen
2. Probleemstelling
 - 2.1 Probleemstelling Data Tool
3. Doel
 - 3.1 Doel family muurplaat
 - 3.2 Doel family dakelement
 - 3.3 Doel Data Tool
4. Plan van aanpak
5. Uitwerking family muurplaat
 - 5.1 Variabele afmetingen
 - 5.2 Variabele dakhelling
 - 5.3 Keuzemogelijkheid geneste families
6. Uitwerking family dakelement
 - 6.1 Base family dakelement
 - 6.2 Variabele afmeting
 - 6.2.1 Variabele panlatafstand
 - 6.3 Variabele dakhelling
 - 6.4 Parameters geneste families
 - 6.5 Visibility family onderdelen
 - 6.5.1 Visibility control door yes/no parameter
 - 6.5.2 Visibility control door object styles subcategory
7. Uitwerking Data Tool
 - 7.1 Geïntegreerde Revit menustructuur
 - 7.1.1 Tab
 - 7.1.2 Ribbonpanels
 - 7.1.3 Pushbuttons
 - 7.1.4 Images
 - 7.1.5 Tooltips
 - 7.1.6 Quick access toolbar
 - 7.2 Data handling
 - 7.2.1 Toekennen van data
 - 7.2.2 Uitlezen van data
 - 7.2.3 Verwijderen van data
 - 7.3 Selectie op basis van data
 - 7.3.1 Highlight op basis van data
 - 7.3.2 Hide op basis van data
 - 7.4 Database bewerkingen
 - 7.4.1 Export to database
 - 7.4.2 Show data from database
 - 7.4.2.1 Show data extern application
 - 7.4.2.2 Show data in browser
 - 7.4.2.3 Show data in Revit UI
 - 7.4.3 Edit data in database
 - 7.4.4 Update model data from database
 - 7.5 Weergave productie voortgang in virtueel model
 - 7.6 Toevoegen parameters aan family via API
 - 7.7 Parameters voorzien van extensible storage data
8. Nawoord
9. Bronnen

1 Inleiding

In de huidige werkmethode worden binnen Autocad Architecture de projecten de 3D-gemodelleerd. Aan deze Autocad objecten kan data worden toegekend, waarna vervolgens de data kan worden geëxporteerd naar een database. Eenmaal geëxporteerd naar de database kan de data zodanig worden bewerkt dat een efficiënte data-handling wordt gerealiseerd.

Dit houdt onder andere in dat de aantallen van de in het model uniek uitgetekende elementen wordt vermenigvuldigd naar het totaal aantal te leveren elementen, inclusief alle onderliggende data behorend bij dat specifieke element. Ook wordt van de 3D-objecten uit het model de leverancier opgezocht en worden er per artikelsoort gegroepeerde bestellingen aangemaakt. De geëxporteerde aantallen worden geanalyseerd naar soort en worden vervolgens ingepland in een gekoppeld planningssysteem.

Door al deze administratieve handelingen vergaand te automatisering in een gewaarborgd proces ontstaat de mogelijkheid om meer tijd te besteden aan het model waardoor het model intrinsiek naar een kwalitatief hoger niveau wordt getild. Vanzelfsprekend kan voornoemd proces slechts dan blijven bestaan indien de betrouwbaarheid van dat proces volkomen is.

Vanwege de duidelijke toename van BIM-projecten is het zinvol om, naast de reeds bestaande mogelijkheden, data-handling binnen Revit te realiseren.

Onder data-handling wordt hier verstaan het toekennen van vrije data aan objecten en deze vervolgens te exporteren naar een database.

In basis zijn hiervoor binnen Revit reeds voorzieningen aanwezig. Echter het grote nadeel is dat er totaal geen waarborging van de integriteit van de data is.

Voor een gegarandeerde werking van dit soort processen is een beveiligde omgeving een absolute noodzaak. Zowel voor de werking van het proces als voor de data zelf.

1.1 Gebruikte software en programmeertalen

De gebruikte software pakketten zijn: Autodesk Revit 2015, Visual Studio 2015, SQL Server en SQL Management Studio 2012.

De gebruikte talen zijn VB.NET (Visual Basic) binnen de Revit UI.

SQL (Structured Query Language) voor het aanspreken van de database.

ASP.NET (Active Server Page) voor dynamische webpagina's.

2. Probleemstelling

2.1 Probleemstelling family muurplaat

Het ontbreken van een family die het mogelijk maakt om te variëren in profiel en de variabele lengtemaat onderverdeelt in te kiezen muurplaatbeugels.

2.2 Probleemstelling family dakelement

Het ontbreken van een family die dermate flexibel is en toch het gewenste detailniveau in zich heeft zodanig dat tot op 'plank-niveau' snel voor een basis woning een dakelement te genereren.

2.3 Probleemstelling data tool

De onmogelijkheid om binnen de basis voorzieningen die er zijn een waarborging van toegekende data te realiseren.

3. Doel

3.1 Doel family muurplaat

Doel is om een family te ontwikkelen die:

- o een variabele breedte, hoogte en lengte kan hebben.
- o een variabele dakhelling kan hebben waarbij een vlak haaks op de dakhelling ontstaat met een vaste lengte van 44mm.
- o een keuze mogelijkheid heeft van geneste families die optimaal verdeeld worden op de lengte met een onderlinge variabele hart-op-hart afstand. Tevens moet in een schedule een count van de afzonderlijke aantallen van de geneste family worden weergegeven.

3.2 Doel family dakelement

Doel is om een family te ontwikkelen die:

- o een variabele breedte kan hebben waarbij het aantal sporen onderling hart op hart 610 mm verdeeld worden.
- o een variabele spoorhoogte kan hebben waarbij de positie van onderkant panlatten 11mm boven de bovenzijde van de sporen blijft liggen.
- o een variabele spoorbreedte kan hebben.
- o een variabele startpositie van de bovenste panlat kan hebben.
- o een variabele onderlinge hart op hart panlatafstand kan hebben
- o een variabele startpositie van de panlat ter plaatse van de goot kan hebben.
- o een variabele dakhelling kan hebben.
- o een variabele basis kan hebben.
- o Een variabele voetplankdikte kan hebben

3.1 Doel data tool

Doel is om een mogelijkheid te maken die, buiten de standaard voorzieningen binnen Revit, het mogelijk maakt om data, op een gewaarborgde manier, toe te kennen aan Revit-objecten.

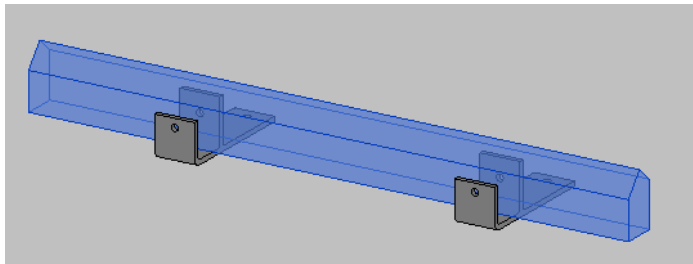
4. Plan van aanpak

Voor de families was vooraleerst een omschrijving van de functionele eisen benodigd.

Na het tot stand komen van die omschrijving kon begonnen worden met het maken van de family en het toevoegen van de gewenste functionaliteit.

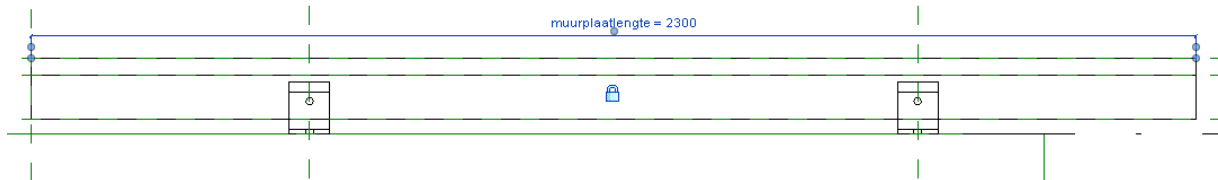
Voor wat betreft de data tool bleek al snel dat er geen uitwerkingen te vinden zijn die als leidraad zouden kunnen dienen. Zeker binnen vb.net zijn er geen voorbeelden van. Het plan van aanpak hier was om het doel te ontleden in veel kleine stappen. Die kleine stappen werden al gauw doelen op zich. Per stap werd via trial and error resultaat bereikt.

5. Uitwerking family muurplaat



5.1 Variabele afmetingen

De breedte, hoogte en lengte van de muurplaat zijn variabel door een in te vullen parameter die gelocked is aan de extrusie.



5.2 Variabele dakhelling

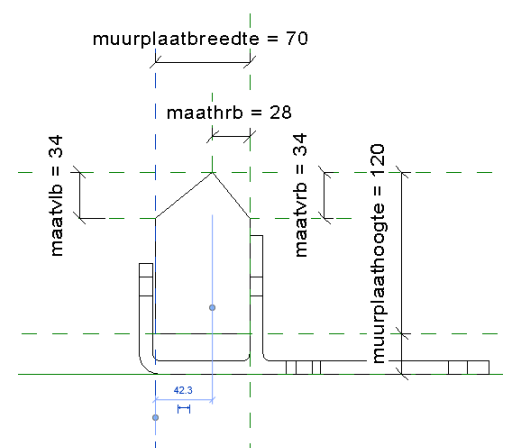
De knikpunten van de muurplaat-extrusie zijn wiskundig berekende waarden.

De knikpunten zijn knikpunten van een extrusie die gelocked zijn aan posities van referentievlakken.

De positie van de vlakken wordt bepaald door de berekende parameterwaarde.

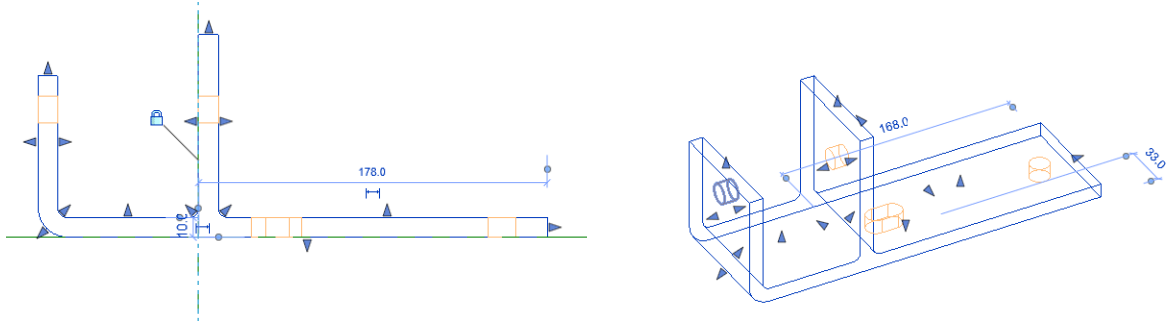
Zodanig dat een vlak ontstaat met een vaste lengte van 44 mm wat haaks op het dakvlak staat.

Parameter	Value	Formula	Lock
Dimensions			
startmaat beugel (default)	150.0	= if((muurplaatlengte - (hoh beugel * ar	<input checked="" type="checkbox"/>
muurplaathoogte (default)	120.0	=	<input checked="" type="checkbox"/>
muurplaatbreedte (default)	70.0	=	<input checked="" type="checkbox"/>
maatvrb (default)	34.2	= 44 mm * cos(dakhelling)	<input checked="" type="checkbox"/>
maatvrb (default)	34.3	= maathrb / tan(90° - dakhelling)	<input checked="" type="checkbox"/>
maathrb (default)	27.7	= 44 mm * sin(dakhelling)	<input checked="" type="checkbox"/>
maathrb (default)	42.3	= (muurplaatbreedte) - maathrb	<input checked="" type="checkbox"/>
hoh beugel (default)	1200.0	=	<input checked="" type="checkbox"/>
dakhelling (default)	39.000°	=	<input checked="" type="checkbox"/>
muurplaatlengte (default)	2300.0	=	<input checked="" type="checkbox"/>
Other			
type muurplaatbeugel <Generic Models>	mpitbgl_70	=	<input checked="" type="checkbox"/>
ansycount (default)	2	= roundup(muurplaatlengte / hoh beu	<input checked="" type="checkbox"/>
Identity Data			

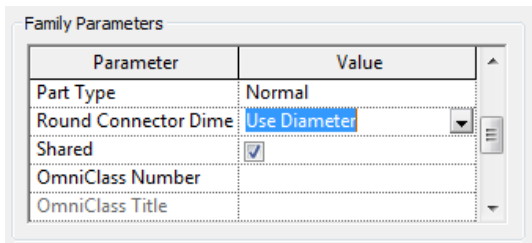


5.3 Keuzemogelijkheid geneste families

De muurplaat zelf bestaat uit een extrusion waarbij de gaten in de beugel ontstaan door voids. Omdat we deze beugel ook in een schedule willen laten weergeven is in deze family verder geen flexibiliteit opgenomen.

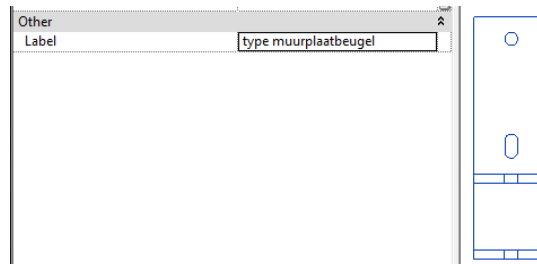
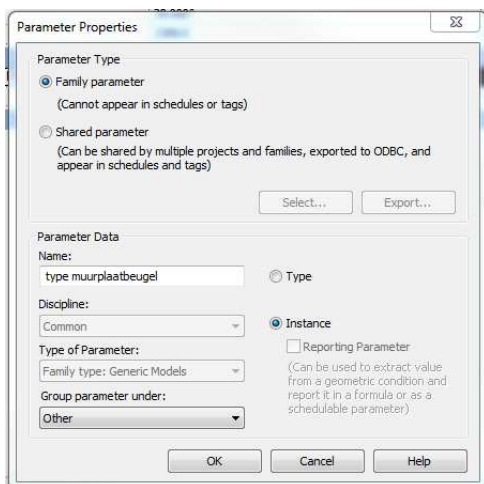


Om de nested family afzonderlijk in een schedule op te nemen moet deze 'Shared' zijn. Indien dit niet het geval is wordt enkel de host family in de schedule getoond.

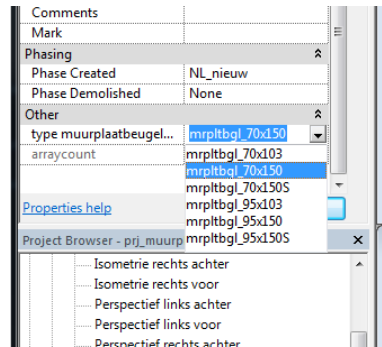


<muurplaatbeugels>	
A	B
muurplaatbeugel	aantal
mrpltbgl_70x103	4
mrpltbgl_70x150	16
mrpltbgl_95x150S	8

Om in de host-family de nested families in een pull-down menu te laten verschijnen, dienen de te nesten families in de host-family te worden geladen, maar niet geplaatst. De parameter die de keuzelijst vult is een 'family parameter'. De parameter wordt gevuld met alle 'Generic models' families (in dit geval) die in de host geladen zijn. Als laatste wordt de parameter gelinkt aan de plaatsingspositie door middel van het label van een geplaatste family.



Het resultaat: de keuzelijst.



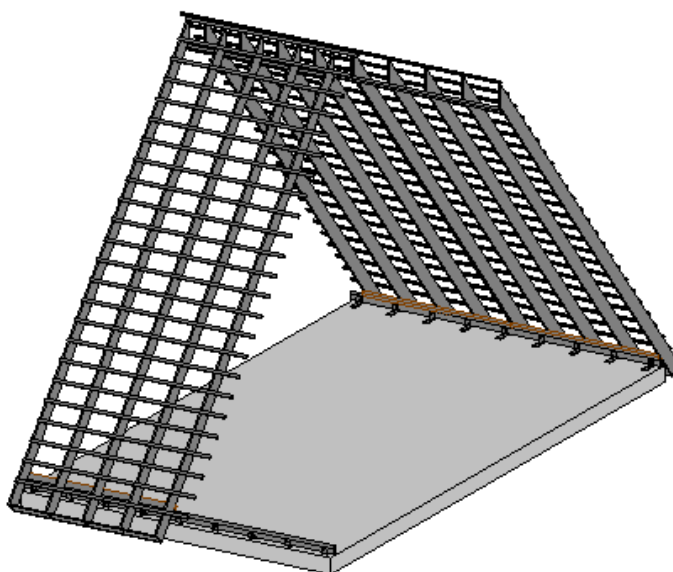
Om de beugels te verdelen zijn 2 in te vullen parameters benodigd: de muurplaatlengte en de hart-op-hart afstand van de beugels. En 2 berekende parameters: het aantal beugels (arraycount) en de startpositie van de eerste beugel (startmaat beugel).

arraycount (default) | 2 | = roundup(muurplaatlengte / hoh beugel)

Het aantal beugels wordt zodanig vanuit het midden over de muurplaat verdeeld dat de de restmaat aan de uiteinden altijd kleiner is dan de halve hart-op-hart afstand.

startmaat beugel (default) | 550.0 | = if((muurplaatlengte - (hoh beugel * arraycount)) / 2 > 0 mm, (muurplaatlengte - (hoh beugel * arraycount)) / 2, (muurplaatlengte - (hoh beugel * arraycount)) / 2 + (hoh beugel / 2))

6. Uitwerking family dakelement



6.1 Base family dakelement

Omdat het dakelement rust op de muurplaat is gekozen om ook in het model het dakelement te relateren aan een face van de muurplaat. Indien de dakhelling van de muurplaat wordt gewijzigd, wijzigt ook de helling van het dakelement mee.

6.2 Variabele afmetingen

Properties	
dakelement	
Generic Models (1) Edit Type	
Constraints	
Host	muurplaat : muurplaat
Elevation	130.3
Dimensions	
onderkantpanlat	275.0
spoorhoogte	220.0
spoordoorsneeknobbuiten	7236.5
spoordoorsneeknobbinnen	6974.3
spoordoorsneekgoot	500.0
spoorbreedte	30.0
positiescharnierregel	6948.3
panlatstartmaatnokcalculated	7262.9
panlatstartmaatnok	0.0
panlathohafstand	310.0
panlateindmaatgoot	200.0
elementbreedte	2440.0
diktevoetplank	19.0
dakhelling	50.000°
basis	4500.0
Volume	0.339 m ³
Identity Data	
Image	
Comments	
Mark	
Phasing	
Phase Created	NL_nieuw
Phase Demolished	None
Other	
panlatzichtbaar	<input checked="" type="checkbox"/>
spoorarraycount	4
panlatarraycount	24
Schedule Level	Zolder

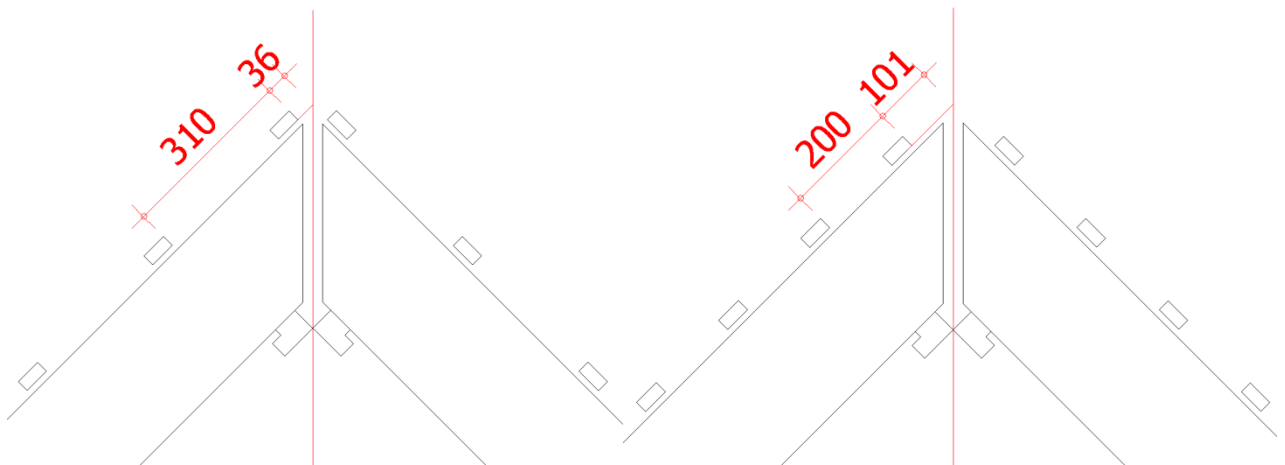
De family is voorzien van een aantal parameters die de flexibiliteit van de family verhogen. Een aantal parameters spreekt voor zich, zoals bijvoorbeeld 'spoorbreedte', hierbij wordt de spoorbreedte aangepast. Een aantal wordt hieronder uitgebreider besproken.

6.2.1 variabele panlatafstand

De positie van de panlat is gekoppeld aan de spoorhoogte. Waarbij tevens het startpunt van de bovenste panlat wordt bepaald door de parameter 'panlatstartmaatnok'. Deze afstand is het snijpunt van bovenzijde van de tengels. Het vervolgens verdelen van de panlatten zelf gebeurt door een array. Het te verdelen vlak wordt bepaald door onderstaande formule.

$$\text{panlatarraycount (default) } 23 = \text{rounddown}((\text{panlatstartmaatnokcalculated} + \text{spoordoorsneekgoot} - \text{panlateindmaatgoot}) / \text{panlathohafstand})$$

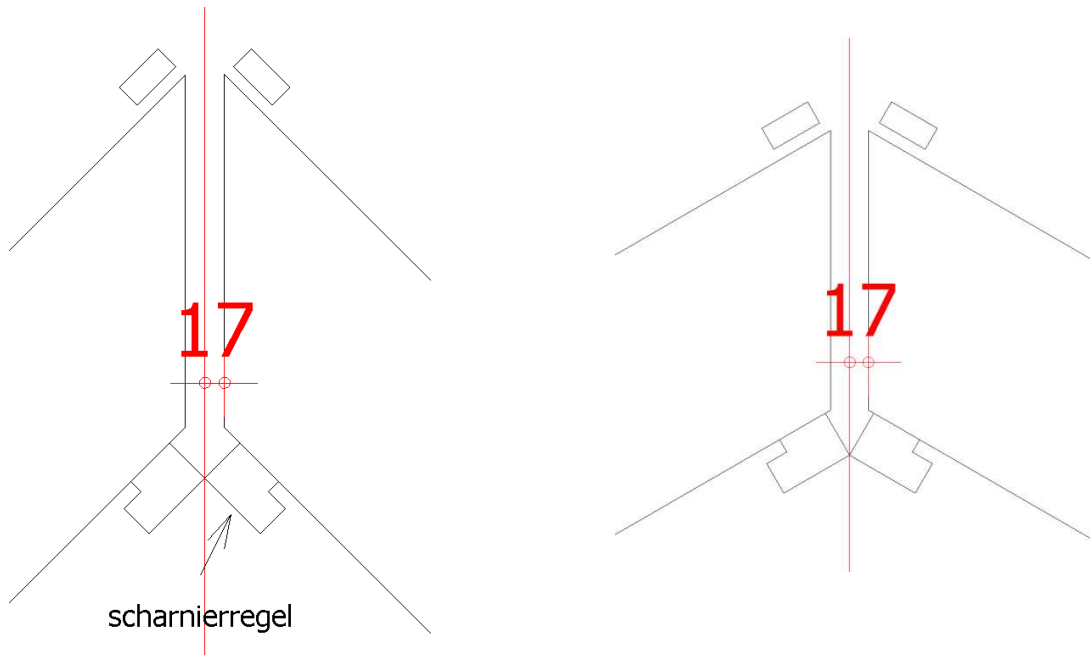
Onderstaand 2 afbeeldingen met een verschillende 'panlatstartmaatnok' en een verschillende 'panlathohafstand'.



6.3 Variabele dakhelling

De variabele dakhelling zorgt ervoor dat het snijpunt onderkant scharnierregels op de noklijn komt te liggen.

En dat de sporen verticaal worden afgeschuind zodanig dat de verticale afschuining 17 mm terug ligt ten op zichte van de noklijn.

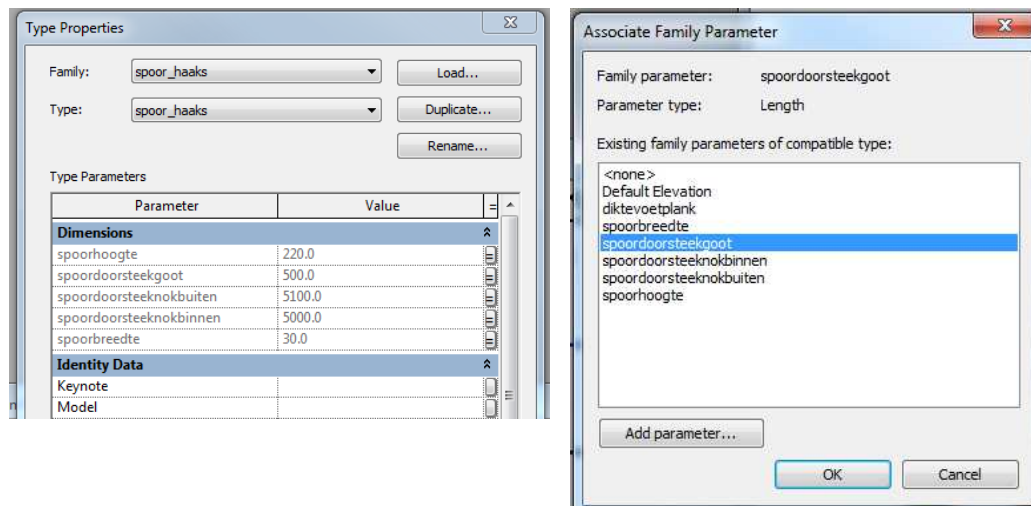


6.4 Parameters geneste families

Het nesten van families gebeurt zoals reeds is omschreven in hoofdstuk 5.3.

Echter in aanvulling hierop willen we parameters van de geneste family kunnen wijzigen in de host family. Om dit te realiseren koppelen we de parameter van de geneste family aan de parameter van de host. Voorwaarde hierbij is dat beide parameters van hetzelfde type zijn, dus tekst parameters koppelen aan tekst parameters. Maak in de te nesten family en in de host family de gewenste parameters aan. Laad de te nesten family in de host en selecteer een instance.

Klik nu op 'Edit type'. Klik op de '='-knop van de betreffende parameter en selecteer de te koppelen parameter. Doe dit voor alle te koppelen parameters.

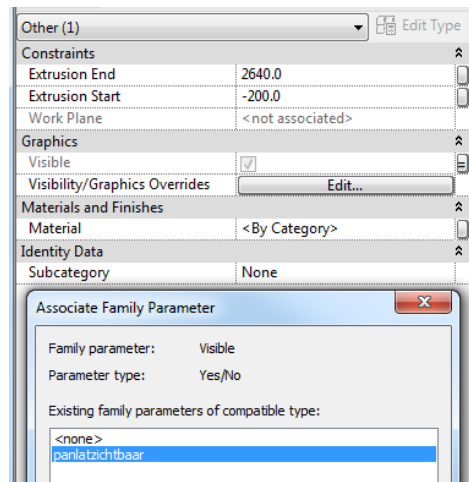


6.5 Visibility family onderdelen

Er zijn een aantal mogelijkheden om de zichtbaarheid/weergave te wijzigen van family onderdelen. We kiezen er 2 om uit te werken.

6.5.1 Visibility control door yes/no parameter

In de host-family is een yes/no parameter aangemaakt met de naam 'panlatzichtbaar'. Deze parameter linken we aan de panlat door de panlat te selecteren in de family en bij eigenschappen- visible de 'panlatzichtbaar'-parameter te selecteren. Deze methode werkt globaal in het project, dat wil zeggen voor alle views.

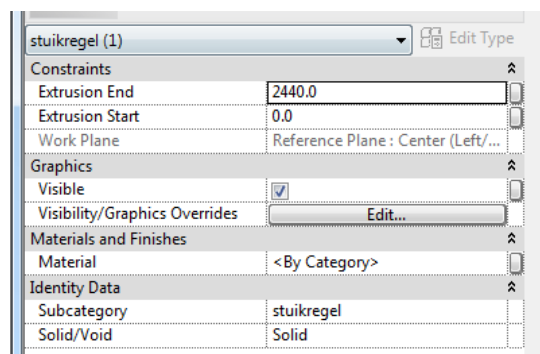


6.5.2 Visibility control door subcategory

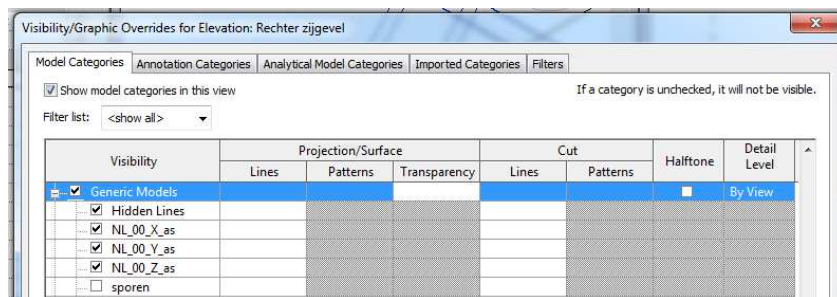
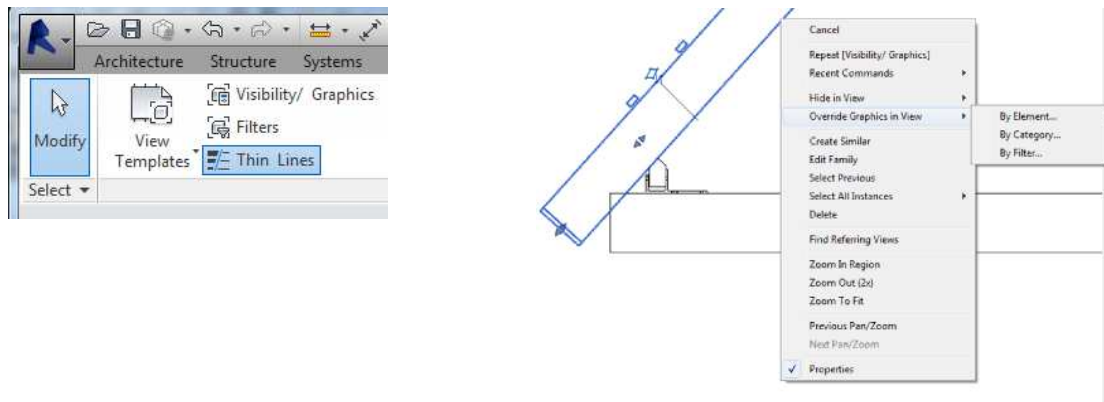
Bij 'Manage' – 'object styles' is het mogelijk om een nieuwe subcategory aan te maken. Waar dan ook de lijnkleur en of het lijntype, material kan worden ingesteld. Hier wordt de kleur bruin ingesteld voor de stuikregel.

Category	Line Weight		Line Color	Line Pattern	Material
	Projection	Cut			
Generic Models	1	1	Black		
Hidden Lines	1	1	Black	Dash	
stuikregel	1	1	RGB 128-064-000	Solid	

Door in de family het object te selecteren kan de subcategory worden gewijzigd.



Door 'View'- 'Visibility/ Graphics' of binnen een view een element te selecteren kan de visibility van de subcategorie worden ingesteld. Deze methode is view afhankelijk. Per view is het mogelijk om verschillende settings in te stellen.



7. Uitwerking Data Tool

7.1 Geïntegreerde Revit menustructuur

Algemeen:

Tijdens opstarten van Revit wordt het 'Data Tool' menu geladen.

De locatie van de DLL, Dynamic Link Library: het programmabestand met de eigenlijke verzameling software-routines, die wordt geladen alsmede de images die worden getoond is niet vast gecodeerd. Met andere woorden: er is geen vast gedefinieerde folder waarin de DLL moet staan, dit kan een willekeurige folder zijn.

Omdat het toekennen van data altijd moet werken vanuit een document, worden de pushbuttons pas geactiveerd bij een geopend document.

Taal: om een mengmoes van Nederlandse en Engelse woorden te voorkomen is gekozen om de dialogen in Revit in het Engels te doen, dit is ook de 'moedertaal' van Revit. De naamgeving van de datavelden is in het Nederlands, omdat dit de duidelijkheid verhoogt en de naamgeving hier juist nadrukkelijk vrij in te vullen is.

7.1.1 Tab

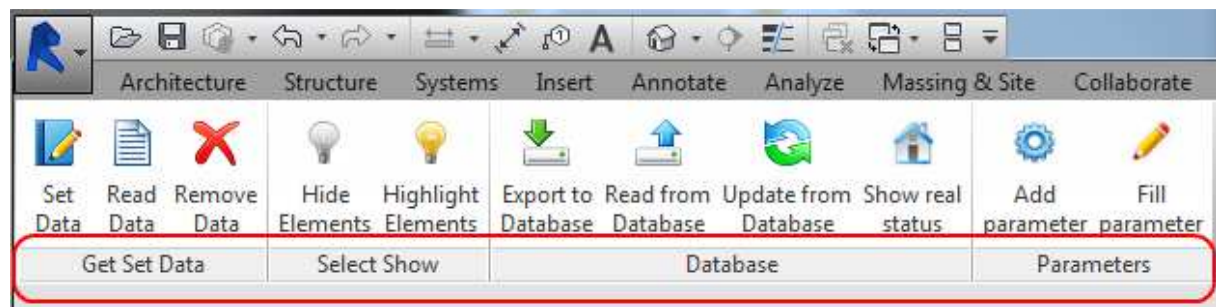
Tijdens het opstarten van Revit wordt een tab in het menu toegevoegd.

De tab is ingedeeld in 4 Ribbonpanels: Get Set Data, Select Show, Database en Parameters.



7.1.2 RibbonPanel

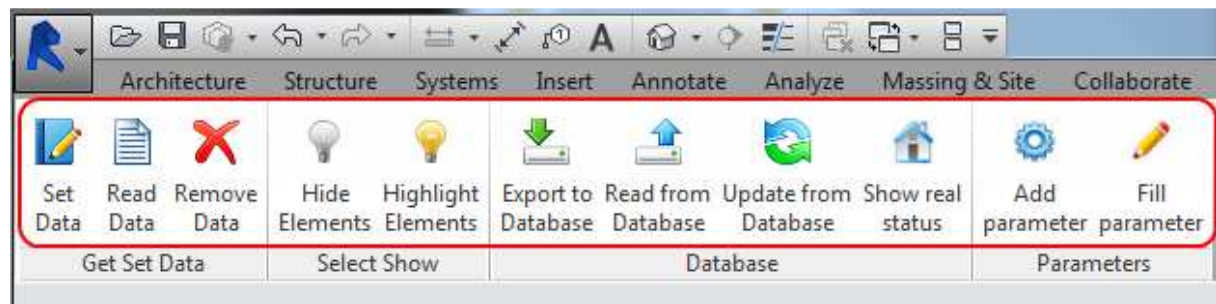
Ribbonpanels groeperen de bij elkaar horende commando's.



7.1.3 Pushbuttons

Pushbuttons realiseren het uitvoeren van een commando.

Aan de tekst bijbehorend bij de button is de eigenschap 'multiline' toegekend.

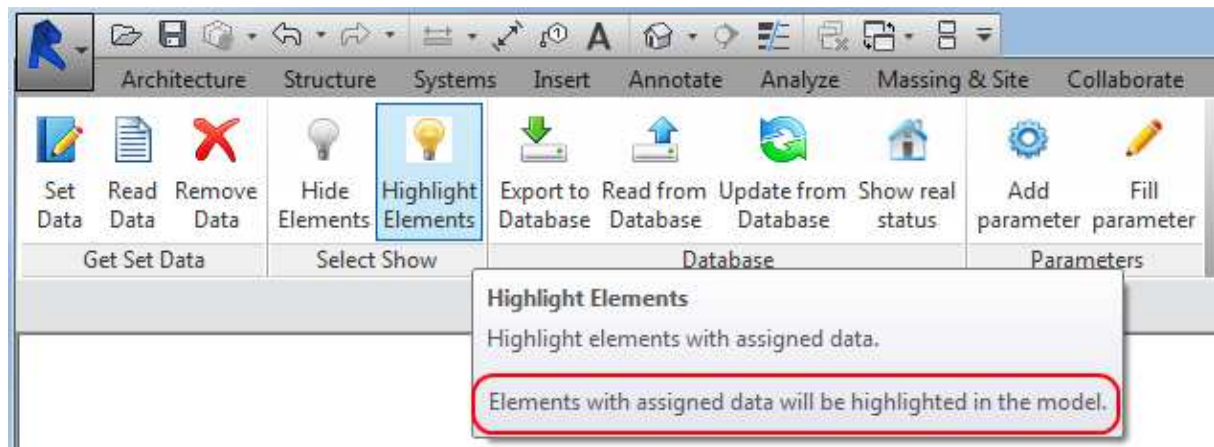
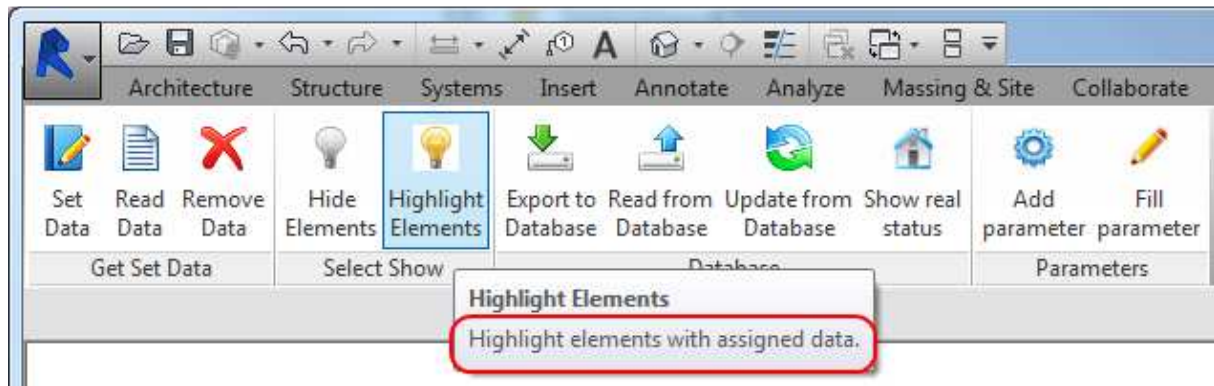


7.1.4 Images

Om de pushbuttons en de bijbehorende commando's meer herkenbaar/aansprekender te laten zijn kunnen deze worden voorzien van een afbeelding. Indien de pushbutton naar het snelmenu wordt geslept, wordt aan de button een andere, kleiner in pixels, afbeelding toegekend.

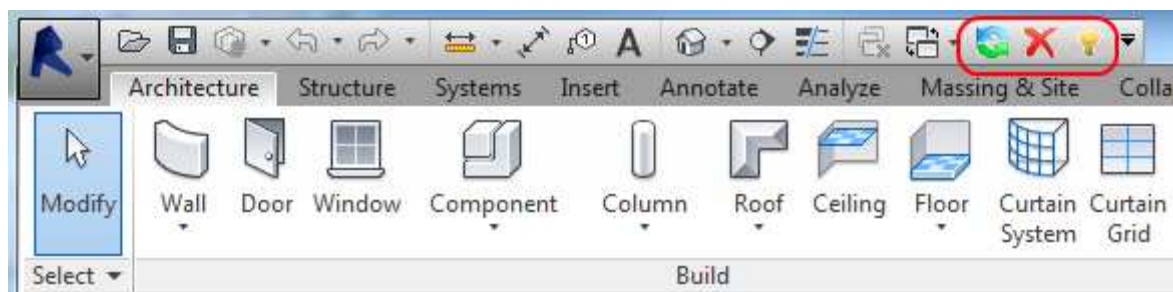
7.1.5 Tooltips

Door met de muisaanwijzer over de pushbutton te gaan wordt de tooltip geactiveerd. Door nu even te blijven hangen met de muisaanwijzer boven de pushbutton verschijnt een uitgebreidere tooltip.



7.1.6 Quick access toolbar

De pushbuttons hebben de mogelijkheid om te worden toegevoegd aan de 'Quick access toolbar'. In dat geval wordt de krijgt de pushbutton een alternatieve afbeelding. De overige eigenschappen blijven ongewijzigd. Het voordeel is dat bij het kiezen van een andere tab de pushbutton toch direct aan te roepen is.



7.2 Data handling

Algemeen:

Toekennen van data aan Revit (roof)objecten middels extensible storage geprogrammeerd in vb.net.

Er zijn binnen Revit bestaande mogelijkheden om aan objecten data toe te kennen. Dit kan binnen vastomlijnde kaders bijvoorbeeld de identity-data parameters.

Daarnaast zijn er mogelijkheden te maken die meer vrijheid bieden om dit te kunnen realiseren, bv parameters. Echter moeten dan wel alle objecten reeds zijn voorzien van deze parameters.

Wat we nu willen realiseren is om met een volledige vrijheid diverse soorten data, string, numeriek, boolean, enz. aan de objecten toe te kennen.

De uitgewerkte methode is om dit via extensible storage te realiseren. Voor deze oplossing is gekozen omdat dit de meest directe vorm van dataopslag is. Tevens biedt dit de benodigde flexibiliteit om de doelstelling efficiënt te realiseren.

Omdat de mogelijkheden schier oneindig zijn is het zaak om vooraleerst een duidelijk afbakening te maken die de gewenste doelstelling realiseert, maar toch niet verzanden in een te uitgebreide functionaliteit.

Eigen data kan zijn informatie met betrekking tot kavelnummers, bouwblokken, woningtypes, routing, elementcoderingen, artikelnummers enz.

De velden van het type string die we aan de objecten toe gaan kennen zijn: project-id, bouwnummer, bouwblok, woningtype en elementcodering. Van het type numeriek: routingnummer.

En van het type datum: elementgereeddatum.

De data kan worden toegekend aan objecten die geselecteerd worden door de user interface.

Nadat data is toegekend aan objecten realiseren we de mogelijkheid tot het opvragen en weergeven van deze data.

Op basis van een selectiefilter van de toegekende data objecten kunnen bewerken, bv pin, hide, unhide, toon alleen enz. alle objecten met elementnummer E03.

Het vervolgens exporteren van deze data naar een database-omgeving, bv SQL server.

Het bewerken van de data in de database-omgeving en terugplaatsen op de oorspronkelijke objecten. Zo ontstaat een één-op-één situatie tussen de data in de database en het model.

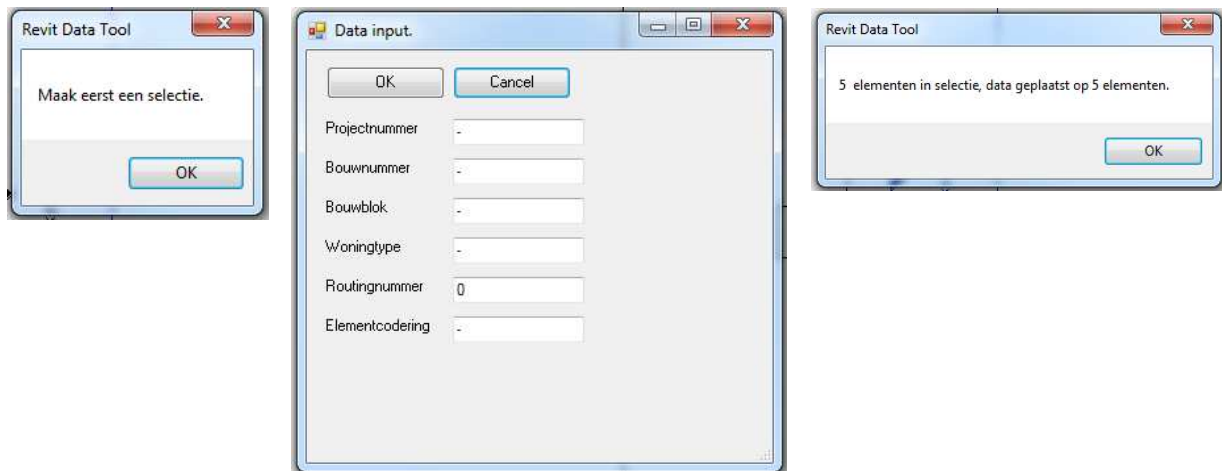
7.2.1 Toekennen van data



Set
Data

Indien het aantal geselecteerde elementen groter is dan 0 wordt het data inputscherm geopend, indien er geen selectie is volgt een melding dat er eerst een selectie moet worden gemaakt. Na invullen van het data inputscherm wordt de ingevulde data als extensible storage schema's aan de geselecteerde objecten toegekend.

Achtereenvolgens de schermen indien er geen elementen zijn geselecteerd, het data input scherm en de melding bij een geslaagde plaatsing van data op de objecten.



7.2.2 Uitlezen van data



Read
Data

Binnen de selectie wordt bepaald of elementen geldige data bevatten, deze wordt uitgelezen en getoond. Tijdens het tonen van de uitgelezen data wordt ingezoomd op het betreffende element. De annuleer knop maakt het mogelijk om het uitlezen te onderbreken.

Werkwijze code:

Eerst wordt een collectie gedefinieerd van de huidige selectie in het actieve document.

Indien het aantal groter is dan nul wordt voor elk element binnen de collectie bekeken of het element een (geldig) data schema bevat. Indien dit het geval is dan wordt de waarde uit elk veld uitgelezen en getoond. Op het element waarvan de data wordt getoond wordt ingezoomd in beeld gebracht. Indien tussentijds het tonen van de data van de selectie moet worden onderbroken kan op annuleren worden gedrukt. Bij geen aanwezig (geldig) schema wordt een scherm getoond met een melding dat er geen data bij het betreffende element is gevonden.

Achtereenvolgens de schermen indien er geen elementen zijn geselecteerd, data word uitgelezen uit van een element met een geldig schema en als het element geen geldig schema heeft.



7.2.3 Verwijderen van data



Binnen de selectie wordt bepaald of de elementen geldige data bevatten, indien ja dan wordt deze van het element verwijderd. Na de actie volgt een melding met het aantal geselecteerde elementen en het aantal elementen waarvan de data is verwijderd.

Werkwijze code:

Inhoudelijk gelijk aan 7.2.2, met het verschil dat in plaats van uitlezen van het (geldige) schema, het schema wordt verwijderd. Na doorlopen van alle elementen in de selectie wordt een scherm getoond met het aantal geselecteerd elementen, en het aantal elementen waarvan het data schema is verwijderd.



7.3 Selectie op basis van data

Voor het selecteren van objecten bestaan een aantal mogelijkheden, ieder met zijn eigen voor en nadelen. Voor het specifiek filteren van gegevens binnen een extensible storage schema wordt binnen de selectie de gegevens van het schema doorlopen. Voldoet de data aan het in het filter gestelde criteria dan wordt deze toegekend aan een collectie. Vervolgens kan met deze collectie een bewerking worden gedaan.

Omdat dit eigenlijk een variatie op de combinatie van hetgeen omschreven is in hoofdstuk 7.3.2 en hoofdstuk 7.5 is dit niet verder uitgewerkt.

7.3.1 Hide op basis van data



Hide elements doet het tegenovergestelde van Highlight elements, namelijk elementen binnen de selectie die geen geldig data schema bevatten worden gehide.

7.3.2 Highlight op basis van data



Highlight elements laat binnen een selectie zien welke elementen een geldig data schema bevatten. Met andere woorden welke elementen zijn voorzien van data.

Werkwijze code: alle elementen binnen de huidige selectie worden doorlopen. Indien een element een geldig data schema bevat wordt het element toegevoegd aan een nieuwe elementset. De aanvankelijke selectieset wordt vervangen door de nieuwe selectieset en het scherm wordt ververst.

7.4 Database bewerkingen

Databasebewerkingen kunnen omvatten: lezen, toevoegen, verwijderen en bewerken van data in de database.

Eerst realiseren we een connectie met de database, we maken een connectie-object.

Indien we data willen lezen uit de database hebben we iets nodig om deze data in te bewaren.

Hiervoor gebruiken we een dataset-object. Deze dataset bestaat in het geheugen van de computer en is te vergelijken met een tabel. De verbinding en de dataset zijn nu gecreëerd maar hebben nog geen onderlinge verbinding met elkaar. Hiervoor gebruiken we het data-adapter object.

De data-adapter communiceert via de connectie met de database, voert daar een SQL-instructie uit en slaat het resultaat van die instructie op in de dataset. Onderstaand vult de data-adapter 'da' de dataset 'ds'. De toevoeging "Projectoverzicht" dient als identifier. Voorbeeld om gegevens uit dataset te halen: `ds.Tables("Projectoverzicht").Rows(0).Item(1)` geeft de waarde uit de 2^e kolom in de eerste rij

```
Dim Con = New SqlConnection(GlobalVar.connectionString)
Dim ds As New dataset
Dim da As New SqlDataAdapter
Dim SQL As String = "SELECT * from TableProject"
Con.Open()
da = New SqlDataAdapter(sql, Connection)
da.Fill(ds, "Projectoverzicht")
Con.Close()
```

7.4.1 Export to database



Export to database exporteert de data van geldige schema's binnen de selectie.

Per element wordt gekeken of dit element een schema bevat. Vervolgens wordt de data in het schema uitgelezen en toegevoegd aan de database.

Omdat we hier records willen toevoegen aan de database maken we gebruik van het Data-adapter object. Middels een SQL-instructie worden de records aan de database toegevoegd. Waarna de connectie met de database wordt gesloten. Een dataset is hier niet van toepassing, we lezen immers niets uit de database.

Achtereenvolgens de schermen indien er geen elementen zijn geselecteerd, elementdata wordt geëxporteerd en wanneer een element geen geldig schema heeft.



7.4.2 Show data from database

Nadat de data is geëxporteerd naar de database is het mogelijk om deze via verschillende kanalen weer te geven en de bewerken. Die kanalen kunnen zijn: een externe op zich zelf staande applicatie, een internetbrowser of binnen Revit.

7.4.2.1 Show data extern application

Een mogelijkheid is om via een externe applicatie een connectie te maken met de database en daarbinnen de data weer te geven. Deze optie wordt enkel illustratief vermeld en wordt hier verder niet uitgewerkt.

7.4.2.2 Show data in browser

Een andere mogelijkheid is om de data via een browser weer te geven. Het voordeel hiervan is dat men niet gebonden is aan een bepaalde locatie of aan een bepaald netwerk om de data te bekijken. Dit soort webpagina's, die interactief zijn met de user, worden dynamische pagina's genoemd, dit in tegenstelling tot statische pagina's die enkel hun vast gedefinieerde lay-out tonen.

Feitelijk reageren dynamische pagina's met een antwoord op een bepaalde vraag. Bijvoorbeeld: laat alle elementen zien van bouwnummer 04, etc.



The screenshot shows a web browser window with the URL www.vankerckhovenprefab.nl. The page features a navigation menu with links for Home, Details, Projecten, CSV, Standhoek, BIM, BIM-data, Hout%, and Revit data. Below the menu, there is a dropdown menu labeled 'Selecteer een project:' with '15701-01' selected. The main content area displays a table titled 'Overzicht Revit elementdata per project' with the following data:

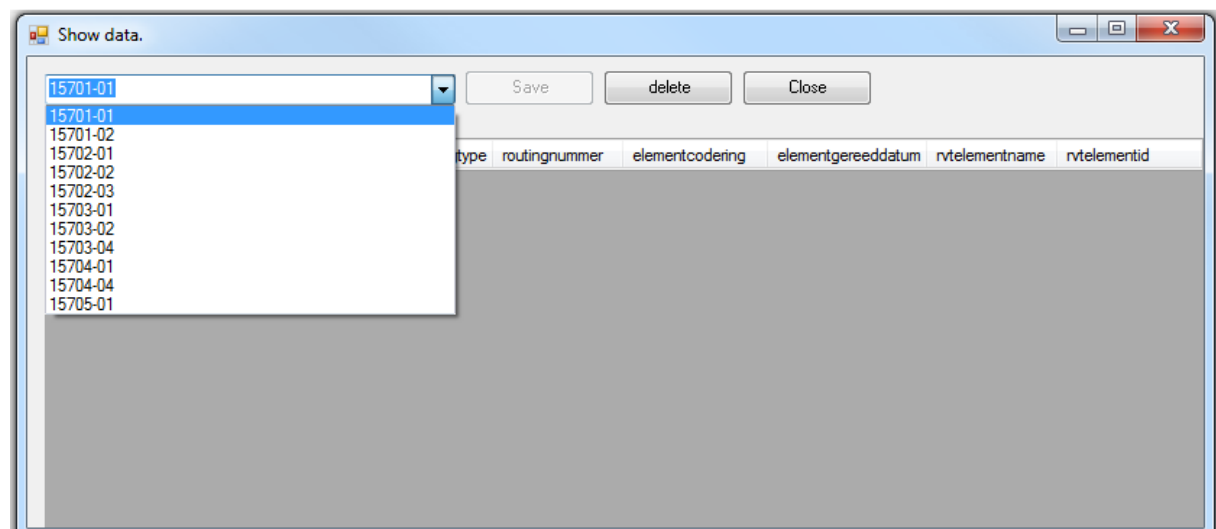
projectnr	bouwnr	bouwblok	woningtype	routingnummer	elementcodering	elementgereeddatum	rvtelementname	rvtelementid
15701-01	01	A	C	01	E01		dakelement	435426
15701-01	01	A	C	01	E01		dakelement	435426
15701-01	01	A	C	01	E01		dakelement	435426
15701-01	01	A	C	01	E02		dakelement	435426
15701-01	01	A	C	01	E02		dakelement	435426
15701-01	01	A	C	01	E02		dakelement	435426
15701-01	01	A	C	01	E02		dakelement	923664
15701-01	01	A	C	01	E03		dakelement	965940
15701-01	01	A	C	01	E03		dakelement	965940
15701-01	01	A	C	01	E03		dakelement	435480

7.4.2.1 Show data in Revit UI



De 'Read from database' button opent een scherm.

Tijdens het openen van het scherm wordt op de achtergrond de pulldown gevuld met de in de database aanwezige projectnummers.



The screenshot shows a dialog box titled 'Show data.' with a dropdown menu containing a list of project numbers: 15701-01, 15701-02, 15702-01, 15702-02, 15702-03, 15703-01, 15703-02, 15703-04, 15704-01, 15704-04, and 15705-01. The '15701-01' option is selected. To the right of the dropdown are three buttons: 'Save', 'delete', and 'Close'. Below the dropdown, a table header is visible with columns: 'type', 'routingnummer', 'elementcodering', 'elementgereeddatum', 'rvtelementname', and 'rvtelementid'.

Na selectie van een project wordt de bij dit project, de uit het Revit-model geëxporteerde, behorende data getoond.

The screenshot shows a window titled "Show data." with a dropdown menu set to "15701-01". Below the menu are three buttons: "Save" (disabled), "delete", and "Close". The main area contains a table with the following data:

projectnummer	bouwnummer	bouwblok	woningtype	routingnummer	elementcodering	elementgereeddatum	rvtelementname	rvtelementid
15701-01	01	A	C	01	E01		dakelement	435426
15701-01	01	A	C	01	E01		dakelement	435426
15701-01	01	A	C	01	E01		dakelement	435426
15701-01	01	A	C	01	E02		dakelement	435426
15701-01	01	A	C	01	E02		dakelement	435426
15701-01	01	A	C	01	E02		dakelement	435426
15701-01	01	A	C	01	E02		dakelement	923664
15701-01	01	A	C	01	E03		dakelement	965940
15701-01	01	A	C	01	E03		dakelement	965940
15701-01	01	A	C	01	E03		dakelement	435480
15701-01	01	A	C	01	E03		dakelement	435729

7.4.3 Edit data in database

Data wordt getoond in een datagridview. Standaard is bij openen van het formulier de save button niet geactiveerd. Bij wijzigen van data wordt de knop geactiveerd. Bij save wordt de data in de database bijgewerkt en wordt de 'save' button op niet actief gezet. Dit om te duiden dat er nog geen niet gesavede wijzigingen zijn aangebracht. Na saven van de gewijzigde data wordt de button weer inactief, totdat er weer wijzigingen in de data zijn aangebracht, enz.

The screenshot shows the same "Show data." window, but the "Save" button is now active. The table data is identical to the previous screenshot, but the "elementgereeddatum" column now contains the date "18-12-2015" for the first two rows.

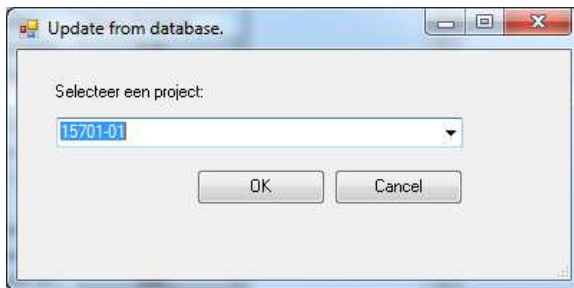
projectnummer	bouwnummer	bouwblok	woningtype	routingnummer	elementcodering	elementgereeddatum	rvtelementname	rvtelementid
15701-01	01	A	C	01	E01	18-12-2015	dakelement	435426
15701-01	01	A	C	01	E01	18-12-2015	dakelement	435426
15701-01	01	A	C	01	E01		dakelement	435426
15701-01	01	A	C	01	E02		dakelement	435426
15701-01	01	A	C	01	E02		dakelement	435426
15701-01	01	A	C	01	E02		dakelement	435426
15701-01	01	A	C	01	E02		dakelement	923664
15701-01	01	A	C	01	E03		dakelement	965940
15701-01	01	A	C	01	E03		dakelement	965940
15701-01	01	A	C	01	E03		dakelement	435480

7.4.4 Update model data from database



Wijzigingen aangebracht in de database kunnen vanuit de database gesynchroniseerd worden met de data in het model. Hier betekent dat de toegevoegde datums in het veld 'elementgereeddatum' worden bijgewerkt in de betreffende extensible storage schema's in het model.

Werkwijze code: Na het openen van de database connectie wordt een SQL-instructie uitgevoerd die alle records uit de tabel ophaalt die bij het geselecteerde projectnummer horen en die een ingevulde elementdatum hebben. De gemaakte dataset wordt doorlopen waarbij het schema van iedere element-id uit het model die voorkomt in de dataset wordt bijgewerkt. Na het doorlopen van de volledige dataset volgt een melding met het aantal bijgewerkte elementen.

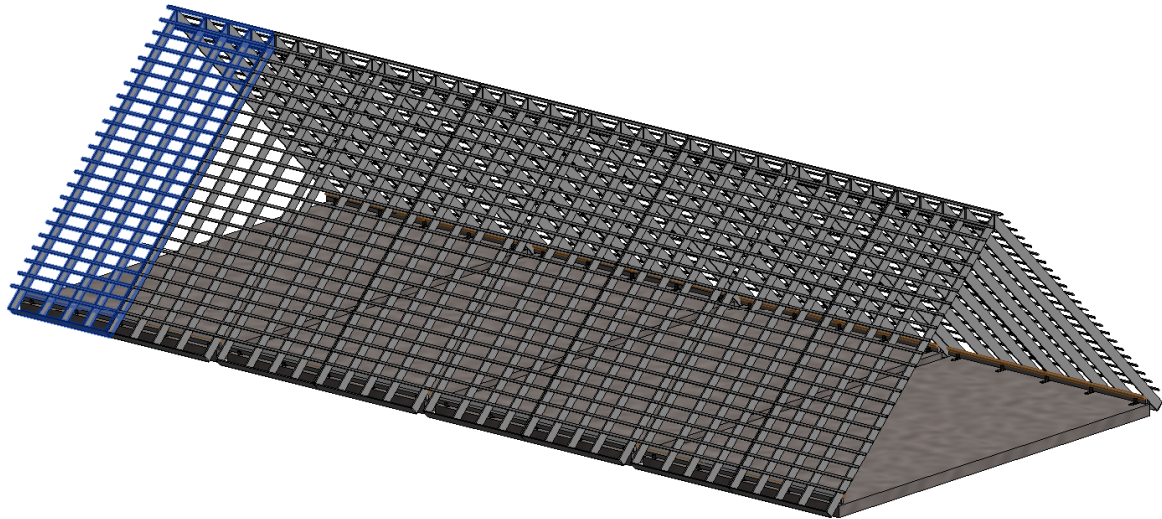


7.5 Weergave productie voortgang in virtueel model

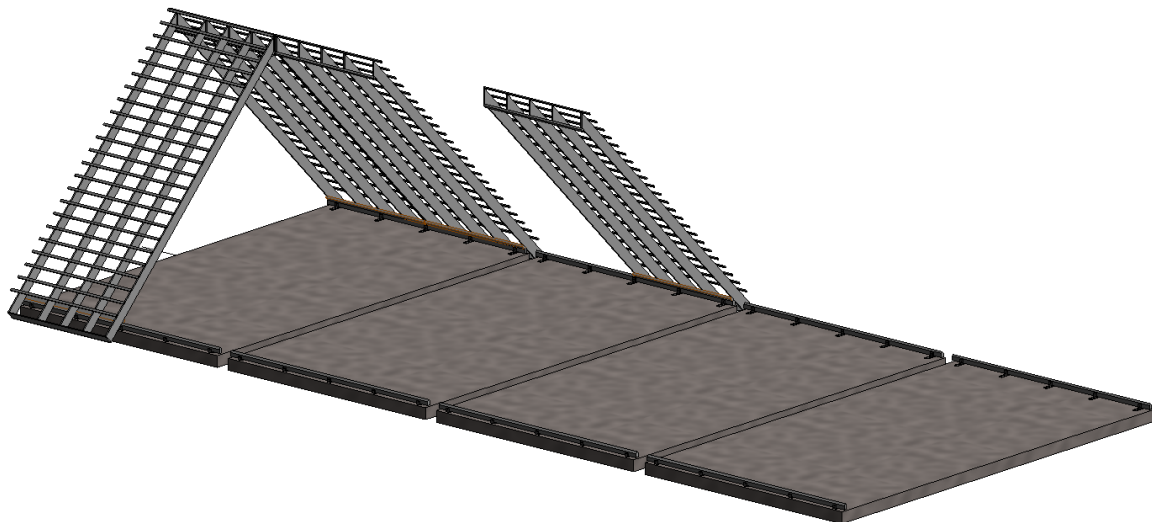


Show real
status

Na synchronisatie met de externe database zijn binnen het model elementen die een ingevulde 'elementgereeddatum'. Hierin is het mogelijk om een selectie te maken. Vervolgens kunnen we deze selectie laten weergeven in een view. Zodoende kunnen we in nagenoeg real-time het virtuele model de actuele stand van de productie laten weergeven. Met andere woorden hetgeen nu zichtbaar is in het model is gereed gemeld in de productie.



Afbeelding van de weergave van het volledig model.




Afbeelding van de weergave van het gereed gemelde model.

7.6 Toevoegen parameters aan family via API




De tot nu toe toegekende data aan de elementen is enkel bij het desbetreffende element weer te geven door middel van de functie 'Read data'. Om nu ook de data weer te geven aan derden die niet over de applicatie beschikken om de data te tonen en om de extensible storage data mee te exporteren naar IFC-modellen wordt de extensible storage data toegekend aan de desbetreffende elementen. Hiertoe worden de benodigde parameters eerst toegekend aan de betreffende family. Indien de huidige selectie meer dan één object bevat wordt een melding getoond dat de selectie uit één object dient te bestaan. Dit omdat de wijziging per family wordt gedaan. De family wordt op de achtergrond geopend in de family-editor. Indien de family al een parameter een parameter bezit met dezelfde naam van de aan te maken parameter zou een foutmelding worden gegenereerd, hierop wordt dus eerst gecontroleerd. Vervolgens wordt de parameter met de property 'instance' aan de family toegevoegd. Dit gebeurt voor alle benodigde aan te maken parameters. Het family-document wordt nu gesloten en vervolgens geladen in het huidige project.

Properties	
	dakelement
Constraints ⌵	
Text ⌶	
Bouwblok	
Bouwnummer	
Elementcodering	
Elementgereed	
Projectnummer	
Routingnummer	
Woningtype	

7.7 Parameters voorzien van extensible storage data

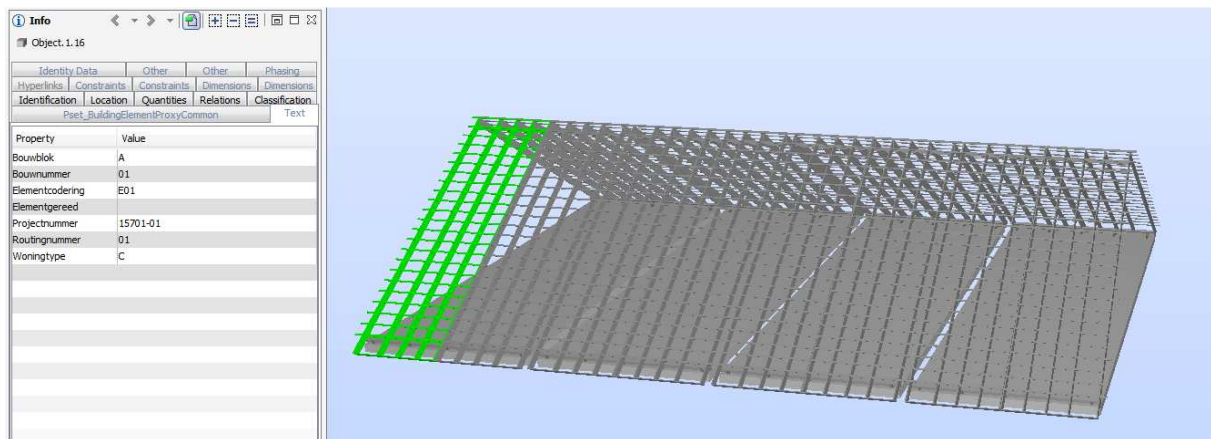


Nadat de benodigde parameters aan de betreffende family zijn toegevoegd kunnen de parameters voorzien worden van de relevante data. In het model wordt voor het project de relevante instantie van de family geselecteerd. Vervolgens wordt de data geplaatst in de daarvoor bestemde parameter.

Properties	
	dakelement
Constraints	
Text	
Bouwblok	A
Bouwnummer	01
Elementcodering	E01
Elementgereed	
Projectnummer	15701-01
Routingnummer	01
Woningtype	C

De nu met data gevulde parameters worden mee geëxporteerd met het betreffende element.

Weergave van het IFC-model met de geëxporteerde data.



8. Nawoord

Alle facetten zoals omschreven in mijn allereerste concept voor deze opdracht zijn eigenlijk uitgewerkt. Voornoemde is voor mij niet een afgerond geheel, het is slechts een begin om de functionele mogelijkheden binnen Revit te gaan benutten.

Als jarenlange Autocad gebruiker is het voor de hand liggend om een vergelijking te maken tussen beide pakketten. Revit biedt meer in zich vanuit de BIM-gedachte op een meer toegankelijke wijze, terwijl Autocad 'consequenter' en meer gestructureerd lijkt te zijn voor wat betreft het tekenen op zich.

Om vanuit Revit direct een productie-omgeving aan te sturen, op basis van data uit het model, is extensible storage een noodzaak. Hoewel Revit in basis de mogelijkheden biedt om wellicht hetzelfde te realiseren, bieden deze mogelijkheden onvoldoende zekerheden voor een gewaarborgd proces.

9. Bronnen

Kris Riemslagh, Software Architect HSBCAD, inspirator voorblad.

Ronald Boeklagen, Revit 2015. Nijmegen, Nederland: TEC / CAD College.

Kean Walmsley, Software Architect voor een aantal AutoCAD-based producten, diverse artikelen verzameld onder: through-the-interface.

Jeremy Tammik, lid van het Autodesk Developer Network ADN team, diverse artikelen verzameld onder: The Building Coder.